66AK2L06 Multicore DSP+ARM KeyStone II System-on-Chip (SoC)

1 66AK2L06 Features and Description

1.1 Features

- Four TMS320C66x DSP Core Subsystems (C66x CorePacs), Each With
  - 1.0 GHz or 1.2 GHz C66x Fixed/Floating-Point DSP Core
    - 38.4 GMacs/Core for Fixed Point @ 1.2 GHz
    - 19.2 GFlops/Core for Floating Point @ 1.2 GHz
  - Memory
    - 32K Byte L1P Per CorePac
    - 32K Byte L1D Per CorePac
    - 1024K Byte Local L2 Per CorePac
- ARM CorePac
  - Two ARM® Cortex®-A15 MPCore™ Processors at Up to 1.2 GHz
  - 1MB L2 Cache Memory Shared by Two ARM 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 for Low Latency Access to Shared MSMC SRAM
- Multicore Shared Memory Controller (MSMC)
  - 2 MB SRAM Memory Shared by Four DSP CorePacs and One ARM CorePac
  - Memory Protection Unit for Both MSM SRAM and DDR3_EMIF
- On-chip Standalone RAM (OSR) - 1MB On-Chip SRAM for Additional Shared Memory
- Hardware Coprocessors
  - Two Fast Fourier Transform Coprocessors
    - Support Up to 1200 Msps at FFT Size 1024
    - Support Max FFT Size 8192
- Multicore Navigator
  - 8k Multi-Purpose Hardware Queues with Queue Manager
  - Packet-Based DMA for Zero-Overhead Transfers
- Network Coprocessor
  - Packet Accelerator Enables Support for
    - 1 Gbps Wire Speed Throughput at 1.5 MPackets Per Second
  - Security Accelerator Engine Enables Support for
    - IPSec, SRTP, and SSL/TLS Security
    - ECB, CBC, CTR, F8,CCM, GCM, HMAC, CMAC, GMAC, AES, DES, 3DES, SHA-1, SHA-2 (256-bit Hash), MD5
    - Up to 6.4 Gbps IPSec
  - Ethernet Subsystem
    - Four SGMII Port Switch
- Peripherals
  - Digital Front End (DFE) Subsystem
    - Support up to Four Lane JESD204A/B (7.37 Gbps Line Rate Max.) Interface to Multiple Data Converters
    - Integration of Digital Down/Up-Conversion (DDC/DUC) Modules
  - IQNet Subsystem
    - Transporting data streams to an integrated Digital Front End (DFE)
  - Two One-Lane PCIe Gen2 Interfaces
    - Supports Up to 5 GBaud
  - Three Enhanced Direct Memory Access (EDMA) Controllers
  - 72-Bit DDR3 Interface, Speeds Up to 1600 MHz
  - EMIF16 Interface
  - USB 3.0 Interface
  - USIM Interface
  - Four UART Interfaces
  - Three I^2C Interfaces
  - 64 GPIO Pins
  - Three SPI Interfaces
  - Semaphore Module
  - Fourteen 64-Bit Timers
- Commercial Case Temperature:
  - 0ºC to 100ºC
- Extended Case Temperature:
  - -40ºC to 100ºC

1.2 Applications

- Medical
- Test and Measurement
- Avionics and Defense
- Industrial

1.3 KeyStone Architecture

TI's KeyStone Multicore Architecture provides a high-performance structure for integrating RISC and DSP cores with application-specific coprocessors and I/O. KeyStone is the first of its kind in that it provides adequate internal bandwidth for non-blocking access to all processing cores, peripherals, coprocessors, and I/O. This is achieved with four main hardware elements: Multicore Navigator, TeraNet, and Multicore Shared Memory Controller.

Multicore Navigator is an innovative packet-based manager that controls 8K queues. When tasks are allocated to the queues, Multicore Navigator provides hardware-accelerated dispatch that directs tasks to the appropriate available hardware. The packet-based system on a chip (SoC) uses the 2-Tbps capacity of the TeraNet switched central resource to move packets. The Multicore Shared Memory Controller enables processing cores to access shared memory directly without drawing from the TeraNet's capacity, so packet movement cannot be blocked by memory access.

1.4 Device Description

The 66AK2L06 KeyStone SoC is a member of the C66x family based on TI's new KeyStone II Multicore SoC Architecture and is a low-power solution with integrated JESD204B lanes that meets the more stringent power, size, and cost requirements of applications requiring connectivity with ADC and DAC based applications. The device's ARM and DSP cores deliver exceptional processing power on platforms requiring high signal and control processing.

TI's KeyStone II Architecture provides a programmable platform integrating various subsystems (ARM CorePac, C66x CorePacs, IP network, Digital Front End, and FFT processing) and uses a queue-based communication system that allows the SoC resources to operate efficiently and seamlessly. This unique SoC architecture also includes a TeraNet switch that enables the wide mix of system elements, from programmable cores to dedicated coprocessors and high-speed IO, to each operate at maximum efficiency with no blocking or stalling.

The addition of the ARM CorePac in the 66AK2L06 device enables the ability for complex control code processing on-chip. Operations such as housekeeping and management processing can be performed with the Cortex-A15 processor.

TI's new C66x core launches a new era of DSP technology by combining fixed-point and floating-point computational capability in the processor without sacrificing speed, size, or power consumption. The raw computational performance is an industry-leading 38.4 GMACS/core and 19.2 Gflops/core (@ 1.2 GHz operating frequency). The C66x is also 100% backward compatible with software for C64x+ devices. The C66x CorePac incorporates 90 new instructions targeted for floating point (FPi) and vector math oriented (VPi) processing.

The 66AK2L06 contains many coprocessors to offload the bulk of the processing demands of higher layers of application. This keeps the cores free for algorithms and other differentiating functions. The SoC contains multiple copies of key coprocessors such as the FFTC. The architectural elements of the SoC (Multicore Navigator) ensure that data is processed without any CPU intervention or overhead, allowing the system to make optimal use of its resources.

TI's scalable multicore SoC architecture solutions provide developers with a range of software-compatible and hardware-compatible devices to minimize development time and maximize reuse.

The 66AK2L06 device has a complete set of development tools that includes: a C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows and Linux debugger interface for visibility into source code execution.

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 higher layer processing. A Digital Front End (DFE) and IQNet (IQN) subsystem have been added to help meet the more stringent power, size and bill of materials (BOM) cost requirements. MSMC internal memory bandwidth is quadrupled with MSMC V2 architecture improvements. Multicore Navigator supports 8K 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 2.0/3.0 controller, USIM interface controller, and Asynchronous EMIF controller for NAND/NOR memory access. The 2-port Gigabit Ethernet switch in KeyStone I has been replaced with a 4-port Gigabit Ethernet 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

Figure 1-1 shows the functional block diagram of the device.

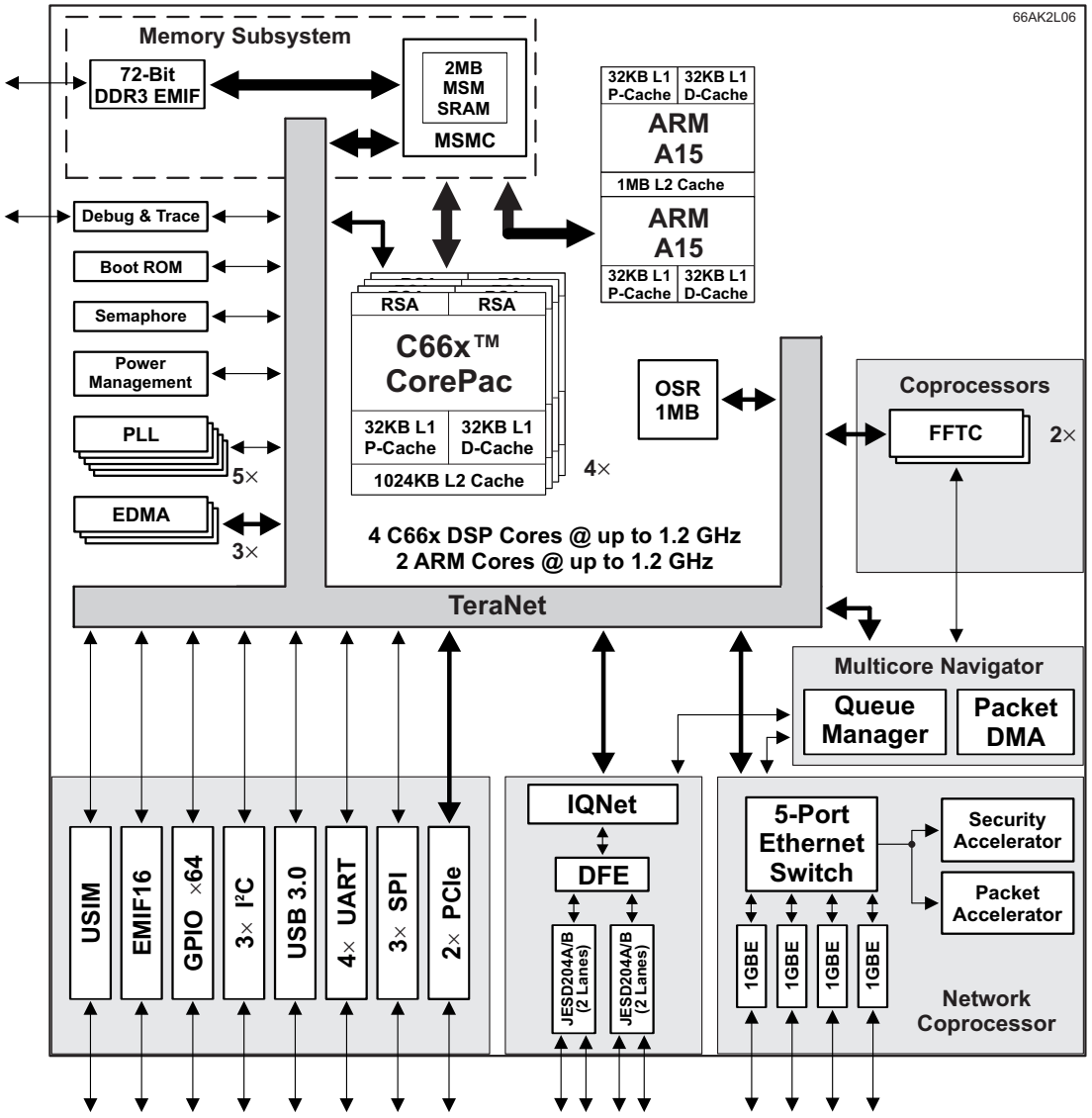


Figure 1-1. Functional Block Diagram

Table of Contents

1 66AK2L06 Features and Description 1
1.1 Features 1
1.2 Applications 2
1.3 KeyStone Architecture 2
1.4 Device Description 2
1.5 Enhancements in KeyStone II 3
1.6 Functional Block Diagram 4
2 Revision History 5
3 Device Characteristics 6
3.1 C66x DSP CorePac 7
3.2 ARM CorePac 7
3.3 Development Tools 8
3.4 Device Nomenclature 8
3.5 Related Documentation from Texas Instruments 9
3.6 Community Resources 10
3.7 Trademarks 10
3.8 Electrostatic Discharge Caution 10
3.9 Glossary 10
4 C66x CorePac 11
4.1 Memory Architecture 12
4.2 Memory Protection 15
4.3 Bandwidth Management 16

4.4 Power-Down Control 16
4.5 C66x CorePac Revision 17
4.6 C66x CorePac Register Descriptions 17
5 ARM CorePac 18
5.1 Features 19
5.2 System Integration 19
5.3 ARM Cortex-A15 Processor 19
5.4 CFG Connection 21
5.5 Main TeraNet Connection 21
5.6 Clocking and Reset 21
6 Terminals 22
6.1 Package Terminals 22
6.2 Pin Map 22
6.3 Terminal Functions 27
6.4 Pullup/Pulldown Resistors 56
7 Memory, Interrupts, and EDMA for 66AK2L06 57
7.1 Memory Map Summary for 66AK2L06 57
7.2 Memory Protection Unit (MPU) 67
7.3 Interrupts for 66AK2L06 80
7.4 Enhanced Direct Memory Access (EDMA3) Controller for 66AK2L06 128
8 System Interconnect 135
8.1 Internal Buses and Switch Fabrics 135
8.2 Switch Fabric Connections Matrix - Data Space 135
8.3 TeraNet Switch Fabric Connections Matrix - Configuration Space 147
8.4 Bus Priorities 155
9 Device Boot and Configuration 156
9.1 Device Boot 156
9.2 Device Configuration 174
10 Device Operating Conditions 211
10.1 Absolute Maximum Ratings 211
10.2 Recommended Operating Conditions 212
10.3 Electrical Characteristics 213
10.4 Power Supply to Peripheral I/O Mapping 214
11 66AK2L06 Peripheral Information and Electrical Specifications 215
11.1 Recommended Clock and Control Signal Transition Behavior 215
11.2 Power Supplies 215
11.3 Power Sleep Controller (PSC) 222
11.4 Reset Controller 229
11.5 PLLs 233
11.6 DDR3A PLL 250
11.7 NETCP PLL 252
11.8 DFE PLL 253
11.9 External Interrupts 257
11.10 On-Chip Standalone RAM (OSR) 258
11.11 DDR3A Memory Controller 258
11.12 I^2C Peripheral 260
11.13 SPI Peripheral 264
11.14 UART Peripheral 267
11.15 PCIe Peripheral 268
11.16 Packet Accelerator 268
11.17 Security Accelerator 269
11.18 Network Coprocessor Gigabit Ethernet (GbE) Switch Subsystem 269
11.19 SGMII Management Data Input/Output (MDIO) 271
11.20 Timers 272
11.21 Rake Search Accelerator (RSA) 273
11.22 General-Purpose Input/Output (GPIO) 274
11.23 Semaphore2 275
11.24 IQNet (IQN) 275
11.25 Digital Front End (DFE) 277
11.26 Fast Fourier Transform Coprocessor (FFTC) 278
11.27 Universal Serial Bus 3.0 (USB 3.0) 278
11.28 Universal Subscriber Identity Module (USIM) 278
11.29 EMIF16 Peripheral 278
11.30 Emulation Features and Capability 281
11.31 Debug Port (EMUx) 284
12 Mechanical Data 294
12.1 Thermal Data 294
12.2 Packaging Information 294

2 Revision History

DATE	REVISION	NOTES
March 2015	*	Initial Release

3 Device Characteristics

Table 3-1 provides an overview of the 66AK2L06 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 66AK2L06 Processor

HARDWARE FEATURES		66AK2L06
Peripherals	DDR3 memory controller (72-bit bus width) (clock source = DDRREFCLKN|P)	1
	16-bit ASYNC EMIF	1
	EDMA3 (64 independent channels) [CPU/3 clock rate]	3
	DFE	1
	IQNet Antenna Interface	1
	I^2C	3
	SPI	3
	PCIe (1 lane)	2
	USB 3.0	1
	USIM(1)	1
	UART	4
	10/100/1000 Ethernet	4 (external ports)
	Management Data Input/Output (MDIO)	1
	64-bit timers (configurable) (internal clock source = CPU/6 clock frequency)	Fourteen 64-bit ***or*** Twenty eight 32-bit
	General-Purpose Input/Output port (GPIO)	64
Encoder/Decoder Coprocessors	FFTC (clock source = CPU/3 clock frequency)	2
Accelerators	Packet Accelerator	1
	Security Accelerator(2)	1
On-Chip Memory Organization	On-chip Standalone Ram (OSR)	1024KB
	L1 program memory controller (C66x)	128KB
	L1 data memory controller (C66x)	128KB
	Shared L2 Cache (C66x)	1024KB
	L3 ROM (C66x)	128KB
	L1 program memory controller (ARM Cortex-A15)	64KB
	L1 data memory controller (ARM Cortex-A15)	64KB
	Shared L2 Cache (ARM Cortex-A15)	1024KB
	L3 ROM (ARM Cortex-A15)	256KB
	MSMC	2MB
C66x CorePac Revision ID	CorePac Revision ID Register (address location: 0x01812000)	0x00090003
JTAG BSDL_ID	JTAGID Register (address location: 0x02620018)	0x0b9a702f
Frequency	DSP	1.0 GHz 1.2 GHz
	ARM	1.0 GHz 1.2 GHz
Voltage	Core (V)	SmartReflex variable supply
	I/O (V)	.85 V, 1.0 V, 1.8 V and 3.3 V
BGA Package	25 mm × 25 mm	900-Pin Flip-Chip Plastic BGA (CMS)
Process Technology	µm	0.028 µm

(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.

Table 3-1. Characteristics of the 66AK2L06 Processor (continued)

HARDWARE FEATURES		66AK2L06
Product Status(3)	Product Preview (PP), Advance Information (AI), or Production Data (PD)	PD

(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 C66x DSP CorePac

The C66x DSP CorePac extends the performance of the C64x+ and C674x CPUs through enhancements and new features. Many of the new features target increased performance for vector processing. The C64x+ and C674x DSPs support 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. On the C66x DSP, the vector processing capability is improved by extending the width of the SIMD instructions. C66x DSPs can execute instructions that operate on 128-bit vectors. The C66x CPU also supports SIMD for floating-point operations. Improved vector processing capability (each instruction can process multiple data in parallel) combined with the natural instruction level parallelism of C6000™ architecture (e.g., execution of up to 8 instructions per cycle) results in a very high level of parallelism that can be exploited by DSP programmers through the use of TI's optimized C/C++ compiler.

Each C66x DSP CorePac has two Rake and Search Accelerators (RSA) integrated on-chip which can perform Reed Muller decoding.

For more details on the C66x CPU and its enhancements over the C64x+ and C674x architectures, see the following documents:

- *TMS320C66x DSP CPU and Instruction Set Reference Guide* (SPRUGH7)
- *TMS320C66x DSP Cache User's Guide* (SPRUGY8)
- *TMS320C66x DSP CorePac User's Guide* (SPRUGW0)

3.2 ARM CorePac

The ARM CorePac of the 66AK2L06 integrates a Cortex-A15 Cluster (2 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.

3.3 Development Tools

3.3.1 Development Support

In case the customer would like to develop their own features and software on the 66AK2L06 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 (DSP/BIOS™), which provides the basic run-time target software needed to support any DSP application
- **Hardware Development Tools**:
  - Extended Development System (XDS™) Emulator (supports multiprocessor system debug)
  - EVM (Evaluation Module)

3.4 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.

Device development evolutionary flow:

- **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, CMS), 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 66AK2L06 in the CMS package type, see the TI website www.ti.com or contact your TI sales representative.

Figure 3-1 provides a legend for reading the complete device name for any C66x+ DSP generation member.

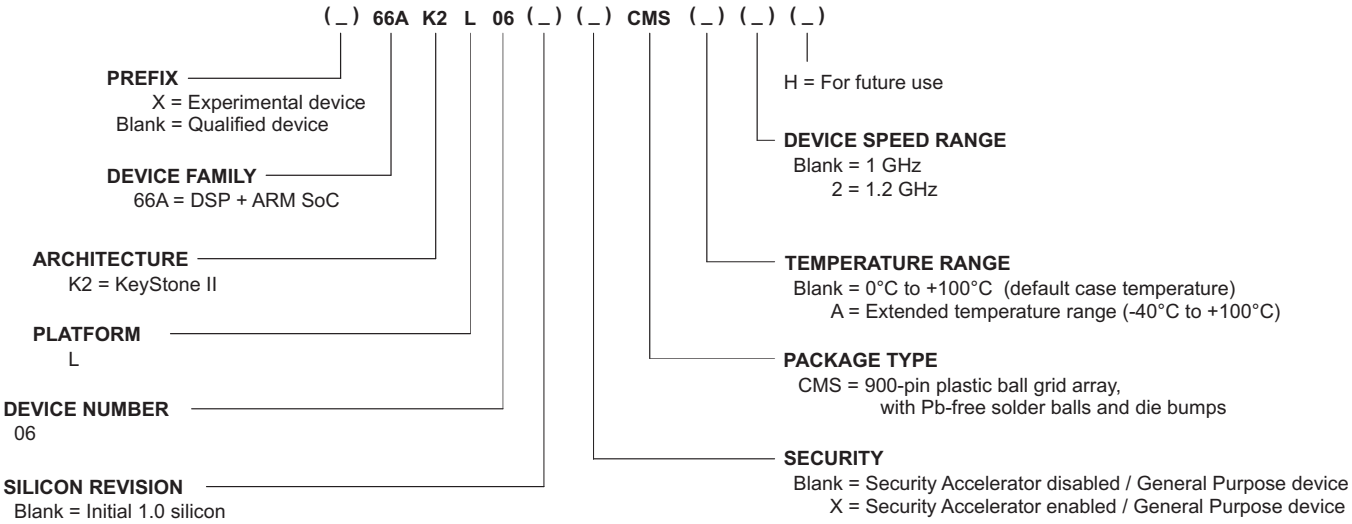


Figure 3-1. C66x DSP Device Nomenclature (including the 66AK2L06)

3.5 Related Documentation from Texas Instruments

These documents describe the 66AK2L06 Multicore ARM+DSP 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 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
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 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
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 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

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

C6000, Code Composer Studio, DSP/BIOS, 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 C66x CorePac

The C66x CorePac consists of several components:

- Level-one and level-two memories (L1P, L1D, L2)
- Data Trace Formatter (DTF)
- Embedded Trace Buffer (ETB)
- Interrupt controller
- Power-down controller
- External memory controller
- Extended memory controller
- A dedicated local power/sleep controller (LPSC)

The C66x CorePac also provides support for big and little endianness, memory protection, and bandwidth management (for resources local to the CorePac). Figure 4-1 shows a block diagram of the C66x CorePac.

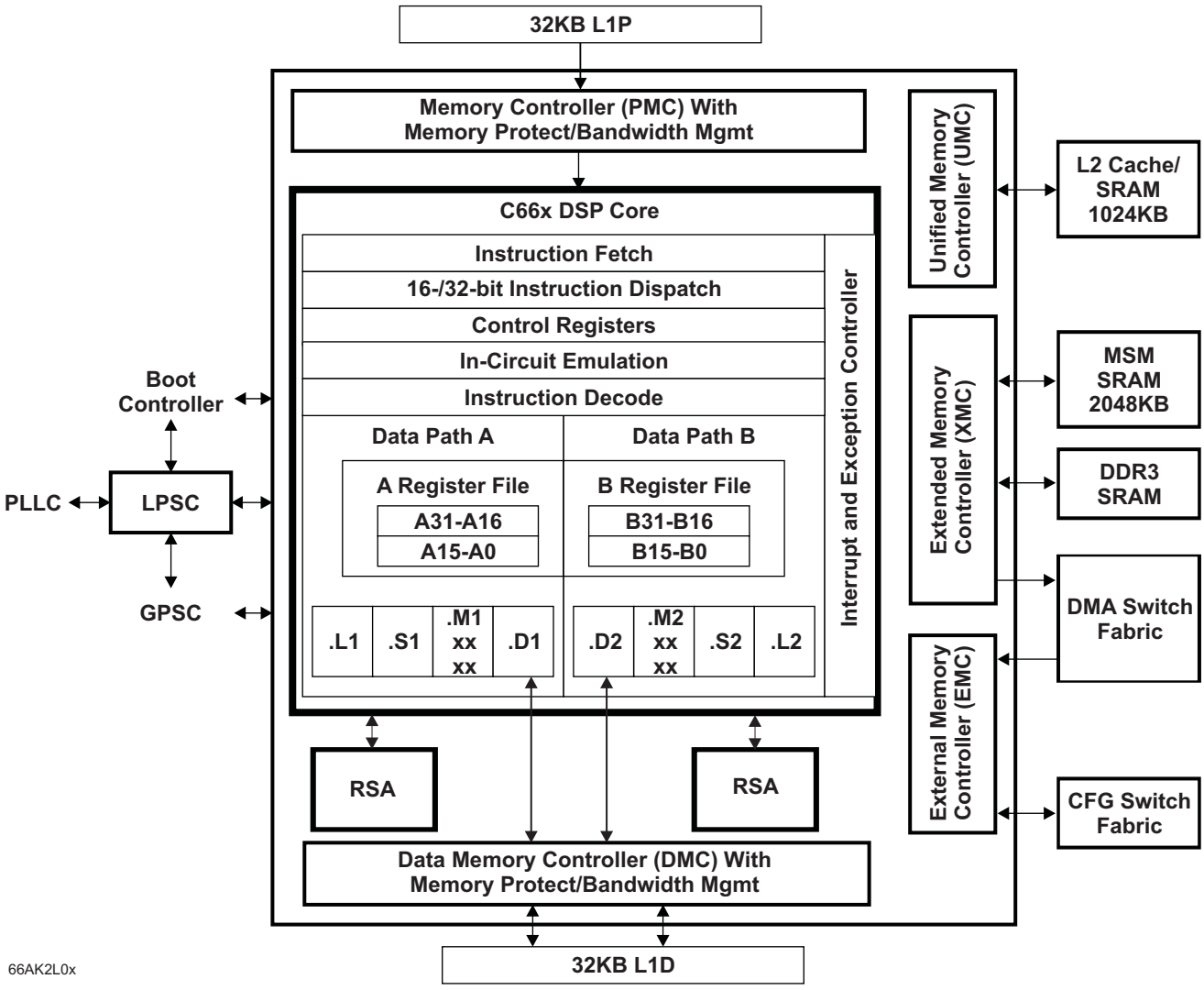


Figure 4-1. C66x CorePac Block Diagram

For more detailed information on the C66x CorePac in the 66AK2L06 device, see the*TMS320C66x DSP CorePac User's Guide* (SPRUGW0).

4.1 Memory Architecture

Each C66x CorePac of the 66AK2L06 device contains a 1024KB level-2 memory (L2), a 32KB level-1 program memory (L1P), and a 32KB level-1 data memory (L1D). The device also contains a 2048KB multicore shared memory (MSM). All memory on the 66AK2L06 has a unique location in the memory map (see Section 7).

After device reset, L1P and L1D cache are configured as all cache, by default. The L1P and L1D cache can be reconfigured via software through the L1PMODE field of the L1P Configuration Register (L1PMODE) and the L1DMODE field of the L1D Configuration Register (L1DCFG) of the C66x CorePac. L1D is a two-way set-associative cache, while L1P is a direct-mapped cache.

The on-chip bootloader changes the reset configuration for L1P and L1D. For more information, see the *KeyStone Architecture DSP Bootloader User's Guide* (SPRUGY5).

For more information on the operation L1 and L2 caches, see the *TMS320C66x DSP Cache User's Guide* (SPRUGY8).

4.1.1 L1P Memory

The L1P memory configuration for the 66AK2L06 device is as follows:

- Region 0 size is 0K bytes (disabled)
- Region 1 size is 32K bytes with no wait states

Figure 4-2 shows the available SRAM/cache configurations for L1P.

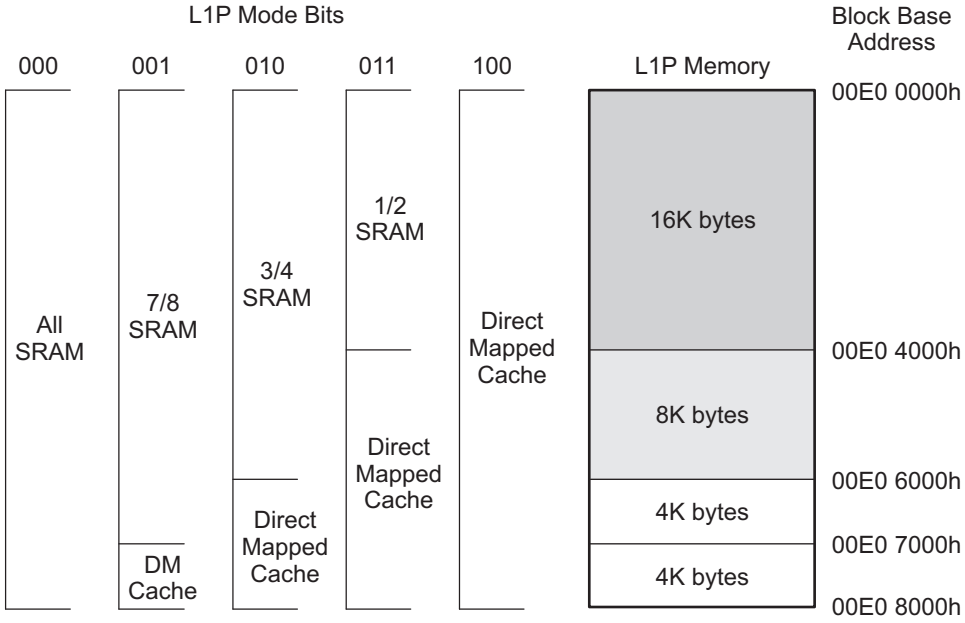


Figure 4-2. L1P Memory Configurations

4.1.2 L1D Memory

The L1D memory configuration for the 66AK2L06 device is as follows:

- Region 0 size is 0K bytes (disabled)
- Region 1 size is 32K bytes with no wait states

Figure 4-3 shows the available SRAM/cache configurations for L1D.

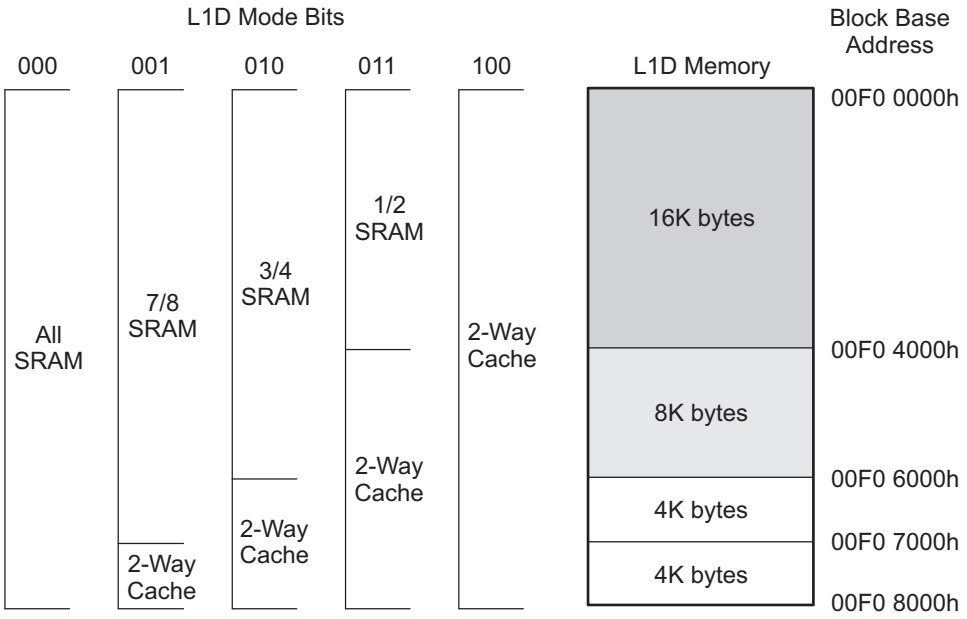


Figure 4-3. L1D Memory Configurations

4.1.3 L2 Memory

The L2 memory configuration for the 66AK2L06 device is as follows:

- Total memory size is 4096KB
- Each CorePac contains 1024KB of memory
- Local starting address for each CorePac is 0080 0000h

L2 memory can be configured as all SRAM, all 4-way set-associative cache, or a mix of the two. The amount of L2 memory that is configured as cache is controlled through the L2MODE field of the L2 Configuration Register (L2CFG) of the C66x CorePac. Figure 4-4 shows the available SRAM/cache configurations for L2. By default, L2 is configured as all SRAM after device reset.

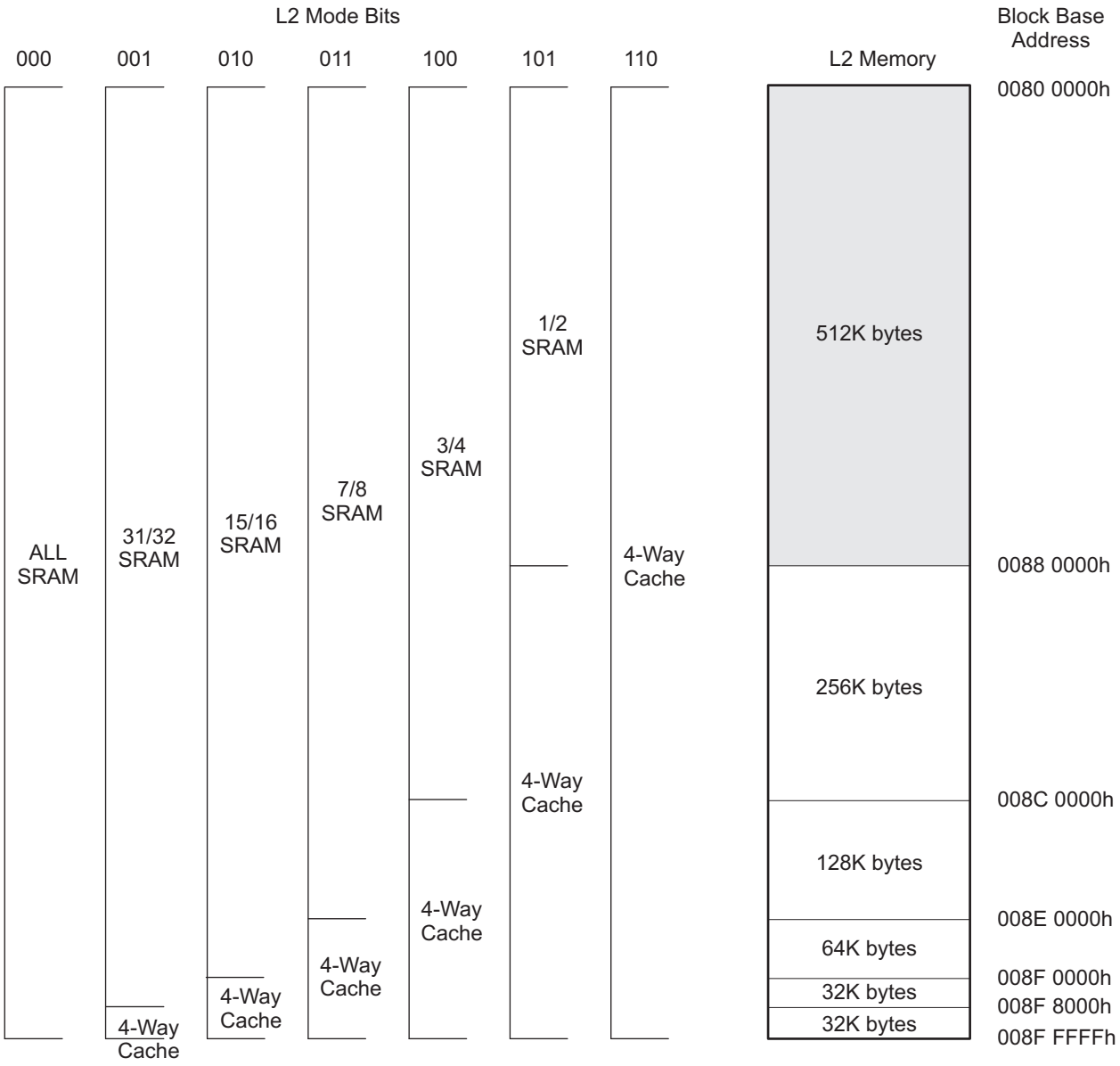


Figure 4-4. L2 Memory Configurations

Global addresses that are accessible to all masters in the system are in all memory local to the processors. In addition, local memory can be accessed directly by the associated processor through aliased addresses, where the eight MSBs are masked to 0. The aliasing is handled within the CorePac and allows for common code to be run unmodified on multiple cores. For example, address location 0x10800000 is the global base address for CorePac0's L2 memory. CorePac0 can access this location by either using 0x10800000 or 0x00800000. Any other master on the device must use 0x10800000 only. Conversely, 0x00800000 can by used by any of the C66x CorePacs as their own L2 base addresses. For CorePac0, as mentioned, this is equivalent to 0x10800000, for CorePac1 this is equivalent to 0x11800000, and for CorePac2 this is equivalent to 0x12800000. Local addresses should be used only for shared code or data, allowing a single image to be included in memory. Any code/data targeted to a specific core, or a memory region allocated during run-time by a particular CorePac should always use the global address only.

4.1.4 Multicore Shared Memory SRAM

The MSM SRAM configuration for the 66AK2L06 device is as follows:

- Memory size of 2048KB
- Can be configured as shared L2 or shared L3 memory
- Allows extension of external addresses from 2GB up to 8GB
- Has built-in memory protection features

The MSM SRAM is always configured as all SRAM. When configured as a shared L2, its contents can be cached in L1P and L1D. When configured in shared L3 mode, it's contents can be cached in L2 also. For more details on external memory address extension and memory protection features, see the *KeyStone Architecture Multicore Shared Memory Controller (MSMC) User's Guide* (SPRUGW7).

4.1.5 L3 Memory

The L3 ROM on the device is 128KB. The ROM contains software used to boot the device. There is no requirement to block accesses from this portion to the ROM.

4.2 Memory Protection

Memory protection allows an operating system to define who or what is authorized to access L1D, L1P, and L2 memory. To accomplish this, the L1D, L1P, and L2 memories are divided into pages. There are 16 pages of L1P (2KB each), 16 pages of L1D (2KB each), and 32 pages of L2 (32KB each). The L1D, L1P, and L2 memory controllers in the C66x CorePac are equipped with a set of registers that specify the permissions for each memory page.

Each page may be assigned with fully orthogonal user and supervisor read, write, and execute permissions. In addition, a page may be marked as either (or both) locally accessible or globally accessible. A local access is a direct DSP access to L1D, L1P, and L2, while a global access is initiated by a DMA (either IDMA or the EDMA3) or by other system masters. Note that EDMA or IDMA transfers programmed by the DSP count as global accesses. On a secure device, pages can be restricted to secure access only (default) or opened up for public, non-secure access.

The DSP and each of the system masters on the device are all assigned a privilege ID. It is possible to specify only whether memory pages are locally or globally accessible.

The AIDx and LOCAL bits of the memory protection page attribute registers specify the memory page protection scheme, see Table 4-1.

Table 4-1. Available Memory Page Protection Schemes

AIDx BIT(1)	LOCAL BIT	DESCRIPTION
0	0	No access to memory page is permitted.
0	1	Only direct access by DSP is permitted.
1	0	Only accesses by system masters and IDMA are permitted (includes EDMA and IDMA accesses initiated by the DSP).
1	1	All accesses permitted.

(1) x = 0, 1, 2, 3, 4, 5

Faults are handled by software in an interrupt (or an exception, programmable within the CorePac interrupt controller) service routine. A DSP or DMA access to a page without the proper permissions will:

- Block the access — reads return 0, writes are ignored
- Capture the initiator in a status register — ID, address, and access type are stored
- Signal the event to the DSP interrupt controller

The software is responsible for taking corrective action to respond to the event and resetting the error status in the memory controller. For more information on memory protection for L1D, L1P, and L2, see the *TMS320C66x DSP CorePac User's Guide* (SPRUGW0).

4.3 Bandwidth Management

When multiple requestors contend for a single C66x CorePac resource, the conflict is resolved by granting access to the highest priority requestor. The following four resources are managed by the bandwidth management control hardware:

- Level 1 Program (L1P) SRAM/Cache
- Level 1 Data (L1D) SRAM/Cache
- Level 2 (L2) SRAM/Cache
- Memory-mapped registers configuration bus

The priority level for operations initiated within the C66x CorePac are declared through registers in the CorePac. These operations are:

- DSP-initiated transfers
- User-programmed cache coherency operations
- IDMA-initiated transfers

The priority level for operations initiated outside the CorePac by system peripherals is declared through the Priority Allocation Register (PRI_ALLOC). System peripherals with no fields in PRI_ALLOC have their own registers to program their priorities.

More information on the bandwidth management features of the CorePac can be found in the*TMS320C66x DSP CorePac User's Guide* (SPRUGW0).

4.4 Power-Down Control

The C66x CorePac supports the ability to power-down various parts of the CorePac. The power-down controller (PDC) of the CorePac can be used to power down L1P, the cache control hardware, the DSP, and the entire CorePac. These power-down features can be used to design systems for lower overall system power requirements.

NOTE

The 66AK2L06 device does not support power-down modes for the L2 memory at this time.

More information on the power-down features of the C66x CorePac can be found in the *TMS320C66x DSP CorePac User's Guide* (SPRUGW0).

4.5 C66x CorePac Revision

The version and revision of the C66x CorePac can be read from the CorePac Revision ID Register (MM_REVID) located at address 0181 2000h. The MM_REVID register is shown in Figure 4-5 and described in Table 4-2. The C66x CorePac revision is dependent on the silicon revision being used.

Figure 4-5. CorePac Revision ID Register (MM_REVID)

31	16	15	0
VERSION		REVISION	
R-n		R-n	

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

Table 4-2. CorePac Revision ID Register (MM_REVID) Field Descriptions

Bit	Name	Value	Description
31-16	VERSION	xxxxh	Version of the C66x CorePac implemented on the device will depend on the silicon being used.
15-0	REVISION	0000h	Revision of the C66x CorePac version implemented on this device.

4.6 C66x CorePac Register Descriptions

See the *TMS320C66x DSP CorePac User's Guide* (SPRUGW0) for register offsets and definitions.

5 ARM CorePac

The ARM CorePac is added in the 66AK2L06 to enable the ability for data processing on-chip. Operations such as housekeeping and management processing can all be performed with the Cortex-A15 processor core.

The ARM CorePac of the 66AK2L06 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 a 1MB 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 figure shows the ARM CorePac.

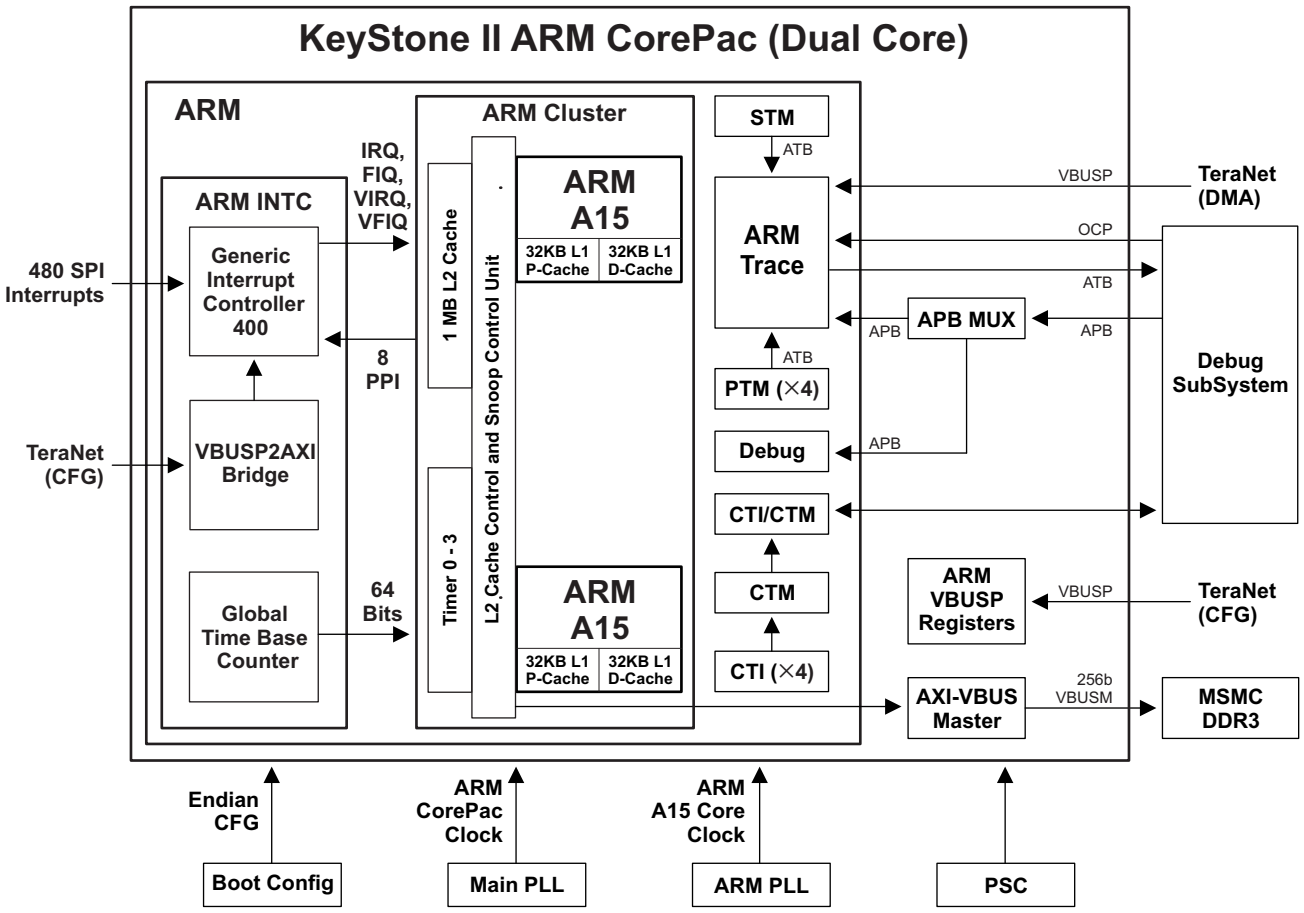


Figure 5-1. ARM CorePac Block Diagram

5.1 Features

The key features of the 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 1MB, 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 CORECLK divided by 6)
- Emulation/debug
  - Compatible with CoreSight™ architecture

5.2 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 5.3.3).
- **Clock Divider:** Provides the required divided clocks to the internal modules of the ARM CorePac and has a clock input from the ARM PLL and the Main PLL
- **In-Circuit Emulator:** Fully compatible with CoreSight™ architecture and enables debugging capabilities.

5.3 ARM Cortex-A15 Processor

5.3.1 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.

5.3.2 Features

Table 5-1 shows the features supported by the Cortex-A15 processor core.

Table 5-1. 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 5-1. 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	1024KB, 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 DDR3A 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

5.3.3 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 7.3 for more details.

The figure below shows an overall view of the ARM CorePac Interrupt Controller.

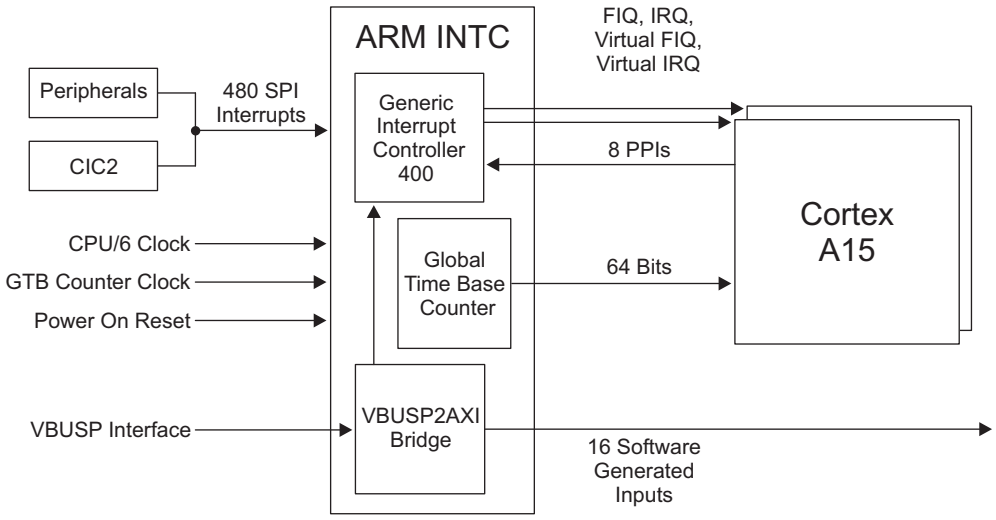


Figure 5-2. ARM Interrupt Controller for Two Cortex-A15 Processor Cores

5.3.4 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.

5.4 CFG Connection

The ARM CorePac has two slave ports. The 66AK2L06 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.

5.5 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.

5.6 Clocking and Reset

5.6.1 Clocking

The Cortex-A15 processor core clocks are sourced from this ARM PLL 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.

5.6.2 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).

6 Terminals

6.1 Package Terminals

Figure 6-1 shows the CMS 900-ball grid array package (bottom view).

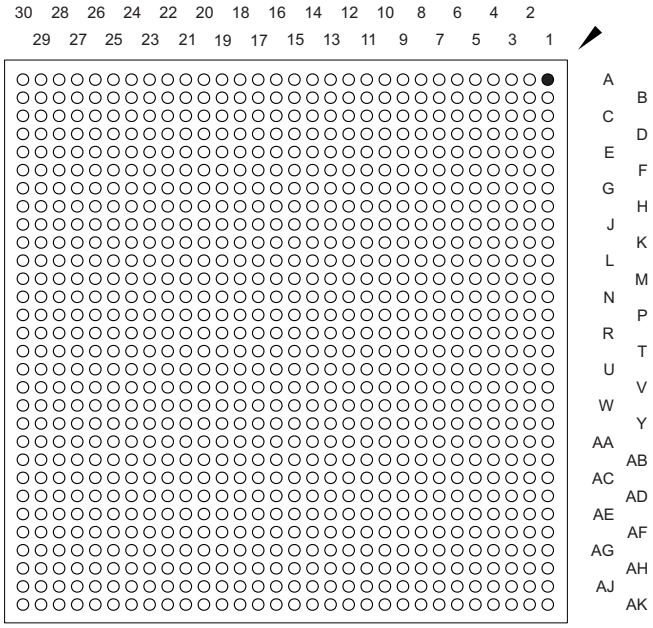


Figure 6-1. CMS 900-Pin BGA Package (Bottom View)

6.2 Pin Map

The following figures show the 66AK2L06 pin assignments in four panels (A, B, C, and D).

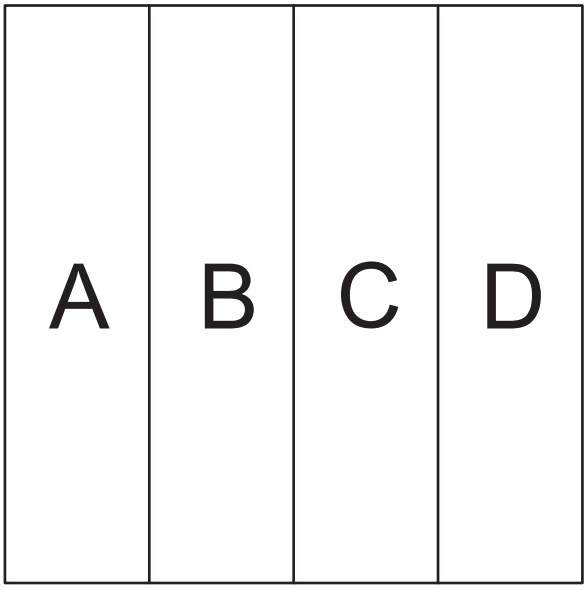


Figure 6-2. Pin Map Panels (Bottom View)

	30	29	28	27	26	25	24	23
A	VSS	DVDDR	DDR3AD63	DDR3AD51	DDR3ADQS6N	DDR3AD50	DDR3ADQS5N	DDR3ADQM5
B	DVDDR	DDR3ADQS7N	DDR3AD62	DDR3AD55	DDR3ADQS6P	DDR3AD49	DDR3ADQS5P	DDR3AD47
C	DDR3AD60	DDR3ADQS7P	DVDDR	DDR3ADQM6	VSS	DDR3AD48	DVDDR	DDR3AD42
D	DDR3AD59	DDR3AD61	VSS	DDR3AD54	DVDDR	DDR3AD41	VSS	DDR3AD45
E	DDR3AD56	DDR3AD57	DDR3ADQM7	DDR3AD53	DDR3AD52	DDR3AD43	DDR3AD44	DDR3AD38
F	DDR3ACLKP	DDR3AD58	GPIO04	GPIO01	GPIO02	DDR3AD40	DDR3AD46	DDR3AD39
G	DDR3ACLKN	GPIO03	GPIO12	GPIO05	GPIO00	DVDDR	VSS	DVDDR
H	GPIO06	GPIO09	GPIO11	GPIO13	GPIO08	RSV012	DVDDR	VSS
J	GPIO14	DVDD18	VSS	GPIO10	GPIO07	RSV011	VSS	AVDDA1
K	SPI0SCS2	SPI0SCS3	SPI0SCS1	GPIO15	GPIO16	VSS	DVDD18	VSS
L	SPI0SCS0	SPI0CLK	SPI1SCS1	SPI0DOUT	SPI1DIN	DVDD18	VSS	VNWA2
M	EMIFRnW	SPI0SCS4	SPI1SCS0	SPI1SCS2	SPI1CLK	VSS	DVDD18	RSV003
N	$\overline{\text{EMIFCE0}}$	VSS	DVDD18	SPI0DIN	SPI1DOUT	DVDD18	VSS	RSV002
P	$\overline{\text{EMIFCE3}}$	$\overline{\text{EMIFBE1}}$	EMIFA03	EMIFA10	EMIFA04	VSS	DVDD18	VSS
R	EMIFA02	EMIFA01	EMIFA09	EMIFA18	EMIFA05	DVDD18	VSS	DVDD18
T	$\overline{\text{EMIFOE}}$	EMIFA08	EMIFA14	EMIFA17	EMIFA16	VSS	DVDD18	VSS
U	$\overline{\text{EMIFCE1}}$	VSS	DVDD18	EMIFA21	EMIFD00	AVDDA2	VSS	DVDD18
V	$\overline{\text{EMIFWE}}$	EMIFA06	EMIFA12	EMIFA23	EMIFD07	VSS	DVDD18	VSS
W	EMIFA00	$\overline{\text{EMIFCE2}}$	EMIFA22	EMIFD02	EMIFD10	DVDD18	VSS	DVDD18
Y	EMIFWAIT0	EMIFA13	EMIFA15	EMIFD04	EMIFD14	VSS	DVDD18	VSS
AA	EMIFA07	VSS	DVDD18	EMIFD06	EMIFD13	DVDD18	VSS	DVDD18
AB	$\overline{\text{EMIFBE0}}$	EMIFA20	EMIFD01	EMIFD05	EMIFD15	VSS	DVDD18	VSS
AC	EMIFWAIT1	EMIFA19	EMIFD03	EMIFD09	EMIFD12	DVDD18	VSS	DVDD18
AD	EMIFA11	VSS	DVDD18	EMIFD08	EMIFD11	VSS	DVDD18	VSS
AE	DFESYSREFP	RSV004	RSV005	CORECLKSEL0	AVDDA4	DVDD18	VSS	SHARED_SERDES_2_REFRES
AF	DFESYSREFN	SYSCLKP	SYSCLKOUT	CORECLKSEL1	AVDDA3	VSS	SGMIICLKN	SGMIICLKP
AG	ALTCORECLKP	SYSCLKN	VSS	VCNTL3	VSS	SHARED_SERDES_2_TXP1	SHARED_SERDES_2_TXN1	VSS
AH	ALTCORECLKN	RSV006	RADSYNC	VCNTL4	VCNTL5	VSS	SHARED_SERDES_2_TXP0	SHARED_SERDES_2_TXN0
AJ	DVDD18	PHYSYNC	VCNTL2	VCNTL0	VSS	SHARED_SERDES_2_RXN0	SHARED_SERDES_2_RXP0	VSS
AK	VSS	DVDD18	VCNTL1	VSS	SHARED_SERDES_2_RXN1	SHARED_SERDES_2_RXP1	VSS	SHARED_SERDES_3_RXN1
	30	**29**	**28**	**27**	**26**	**25**	**24**	**23**

Figure 6-3. 66AK2L06 Left End Panel (A) — Bottom View

	22	21	20	19	18	17
A	DDR3ADQS4P	DDR3AD32	DDR3ADQS8P	DDR3ADQM8	DDR3AA02	DDR3AA05
B	DDR3ADQS4N	DDR3AD33	DDR3ADQS8N	DDR3ACB07	DDR3AA11	DDR3AA08
C	VSS	DDR3AD34	DVDDR	DDR3ACB06	VSS	DDR3AA12
D	DVDDR	DDR3AD35	VSS	DDR3ACB05	DVDDR	DDR3AA09
E	DDR3AD36	DDR3AD37	DDR3ACB03	DDR3ACB04	DDR3AA14	DDR3ACKE0
F	DDR3ADQM4	DDR3ARZQ2	DDR3ACB01	DDR3ACB00	DDR3ACB02	DDR3ACKE1
G	VSS	DVDDR	VSS	DVDDR	VSS	DVDDR
H	DVDDR	VSS	DVDDR	VSS	DVDDR	VSS
J	VSS	AVDDA5	VSS	DVDDR	VSS	AVDDA4
K	CVDD	VSS	CVDD	VSS	CVDD	VSS
L	VSS	CVDD	VSS	CVDD	VSS	CVDD
M	CVDD	VSS	CVDD	VSS	CVDD	VSS
N	VSS	CVDD1	VSS	CVDD1	VSS	CVDD
P	CVDD	VSS	CVDD1	VSS	CVDD	VSS
R	VSS	CVDD	VSS	CVDD	VSS	CVDD
T	CVDD	VSS	CVDD	VSS	CVDD	VSS
U	VSS	CVDD	VSS	CVDD	VSS	CVDD
V	CVDD	VSS	CVDD	VSS	CVDD	VSS
W	VSS	CVDD	VSS	CVDD	VSS	CVDD
Y	CVDD	VSS	CVDD	VSS	CVDD	VSS
AA	VSS	CVDD	VSS	CVDD	VSS	CVDD
AB	CVDD	VSS	CVDD	VSS	CVDDS	VSS
AC	VSS	VNWA3	VSS	CVDDS	VSS	CVDDS
AD	AVDDAS	VSS	AVDDAS	VSS	AVDDAS	VSS
AE	SHARED_SERDES_3_REFRES	RSV014	PCIECLKP	PCIECLKN	VSS	SHARED_SERDES_0_REFRES
AF	RSV013	VSS	RSV017	VSS	SHARED_SERDES_0_REFCLKN	SHARED_SERDES_0_REFCLKP
AG	SHARED_SERDES_3_TXP1	SHARED_SERDES_3_TXN1	VSS	SHARED_SERDES_0_TXP1	SHARED_SERDES_0_TXN1	VSS
AH	VSS	SHARED_SERDES_3_TXP0	SHARED_SERDES_3_TXN0	VSS	SHARED_SERDES_0_TXP0	SHARED_SERDES_0_TXN0
AJ	SHARED_SERDES_3_RXN0	SHARED_SERDES_3_RXP0	VSS	SHARED_SERDES_0_RXN0	SHARED_SERDES_0_RXP0	VSS
AK	SHARED_SERDES_3_RXP1	VSS	SHARED_SERDES_0_RXN1	SHARED_SERDES_0_RXP1	VSS	SHARED_SERDES_1_RXN1
	22	21	20	19	18	17

Figure 6-4. 66AK2L06 Left Center Panel (B) — Bottom View

16	15	14	13	12	11	10	9	
DDR3ACLKOUTP0	DDR3ACLKOUTN0	DDR3ACE0	DDR3ACAS	DDR3AODT0	DDR3ACE1	DDR3ADQS3P	DDR3AD28	**A**
DDR3AA07	DDR3ACLKOUTP1	DDR3ACLKOUTN1	DDR3ARAS	DDR3AA10	DDR3AA13	DDR3ADQS3N	DDR3AD30	**B**
DVDDR	DDR3AA04	DDR3AA00	VSS	RSV001	DVDDR	DDR3AD27	VSS	**C**
VSS	DDR3AA06	DDR3AA01	DVDDR	DDR3AWE	VSS	DDR3AD25	DVDDR	**D**
DDR3AA15	DDR3ARESET	DDR3AA03	RSV015	DDR3ABA2	DDR3ABA0	DDR3AD26	DDR3AD31	**E**
VSS	DDR3AVREFSSTL	RSV016	DDR3ARZQ0	DDR3ABA1	DDR3AODT1	DDR3AD24	DDR3ARZQ1	**F**
VSS	DVDDR	VSS	DVDDR	VSS	DVDDR	VSS	DVDDR	**G**
DVDDR	VSS	DVDDR	VSS	DVDDR	VSS	DVDDR	VSS	**H**
VSS	DVDDR	AVDDA3	DVDDR	VSS	AVDDA2	VSS	DVDDR	**J**
CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDCMON	VSSCMON	**K**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	**L**
CVDD1	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS	**M**
VSS	CVDD1	VSS	CVDD	VSS	CVDD1	VSS	CVDD	**N**
CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	**P**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	DVDD33	**R**
CVDD	VSS	CVDD	VSS	CVDD	VSS	VPH	VSS	**T**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDUSB	**U**
CVDD	VSS	CVDD	VSS	CVDD	VSS	VPTX	VSS	**V**
VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD	**W**
CVDD1	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	**Y**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	**AA**
CVDDS	VSS	CVDDS	VSS	CVDD	VSS	CVDD	VSS	**AB**
VSS	CVDDS	VSS	CVDDS	VSS	DVDD18	VSS	DVDD18	**AC**
AVDDAS	VSS	AVDDAS	VSS	AVDDAS	VSS	DVDD18	VSS	**AD**
VSS	RSV018	VSS	SHARED_SERDES_1_REFRES	VSS	DVDD18	VSS	DVDD18	**AE**
VSS	SHARED_SERDES_1_REFCLKN	SHARED_SERDES_1_REFCLKP	VSS	AVDDA5	DFESYNCINN1	DFESYNCINP1	DFEIO14	**AF**
SHARED_SERDES_1_TXP1	SHARED_SERDES_1_TXN1	VSS	DFESYNCINN0	DFESYNCINP0	VSS	DFEIO10	DFEIO16	**AG**
VSS	SHARED_SERDES_1_TXP0	SHARED_SERDES_1_TXN0	VSS	DFESYNCOUTN1	DFESYNCOUTP1	VSS	DFEIO17	**AH**
SHARED_SERDES_1_RXN0	SHARED_SERDES_1_RXP0	VSS	TSRXCLKOUT0N	TSRXCLKOUT0P	VSS	DFESYNCOUTN0	DFESYNCOUTP0	**AJ**
SHARED_SERDES_1_RXP1	VSS	TSREFCLKN	TSREFCLKP	VSS	DFEIO15	DFEIO12	DFEIO13	**AK**
16	**15**	**14**	**13**	**12**	**11**	**10**	**9**	

Figure 6-5. 66AK2L06 Right Center Panel (C) — Bottom View

8	7	6	5	4	3	2	1	
DDR3AD29	DDR3ADQS2P	DDR3ADQM2	DDR3AD08	DDR3ADQS1N	DDR3ADQS0P	DVDDR	VSS	**A**
DDR3ADQM3	DDR3ADQS2N	DDR3AD19	DDR3AD11	DDR3ADQS1P	DDR3ADQS0N	DDR3AD02	DVDDR	**B**
DDR3AD23	DVDDR	DDR3AD18	VSS	DDR3AD14	DVDDR	DDR3AD01	DDR3AD03	**C**
DDR3AD22	VSS	DDR3AD09	DVDDR	DDR3AD13	VSS	DDR3AD05	DDR3AD04	**D**
DDR3AD21	DDR3AD17	DDR3AD10	DDR3ADQM1	DDR3ADQM0	DDR3AD07	DDR3AD06	DDR3AD00	**E**
DDR3AD16	DDR3AD20	DDR3AD12	DDR3AD15	VSS	USIMCLK	USIMIO	USIMRST	**F**
VSS	DVDDR	VSS	VSS	$\overline{\text{POR}}$	VSS	MDCLK	VSS	**G**
DVDDR	VSS	DVDDR	VSS	VSS	TIMO0	VSS	MDIO	**H**
VSS	DVDDR	VSS	UART1TXD	UART1RTS	TIMO1	TIMI0	TIMI1	**J**
AVDDA1	VSS	VSS	UART0RXD	UART1CTS	UART0TXD	UART0RTS	VSS	**K**
VSS	VNWA1	VSS	UART1RXD	UART0CTS	SCL1	VSS	USBTX0P	**L**
USBRESREF	VSS	USBDRVVBUS	SDA1	SDA2	SCL2	SDA0	USBTX0M	**M**
VSS	VSS	VSS	USBID0	USBVBUS	VSS	SCL0	VSS	**N**
VSS	VSS	VSS	VSS	VSS	USBCLKP	VSS	USBDP	**P**
VSS	DVDD18	VSS	VSS	VSS	USBCLKM	USBRX0P	USBDM	**R**
DVDD18	VSS	DVDD18	GPIO29	GPIO31	GPIO27	USBRX0M	VSS	**T**
VSS	DVDD18	VSS	GPIO30	GPIO23	GPIO28	VSS	GPIO26	**U**
VPH	VSS	DVDD18	GPIO24	GPIO25	GPIO19	GPIO22	GPIO21	**V**
VSS	DVDD18	VSS	RSV010	VSS	DVDD18	GPIO17	GPIO20	**W**
VNWA4	VSS	DVDD18	VSS	RSV009	EMU17	EMU18	GPIO18	**Y**
VSS	DVDDR	VSS	EMU16	EMU13	EMU15	EMU14	EMU12	**AA**
DVDD18	VSS	DVDD18	EMU10	EMU09	EMU11	DVDD18	VSS	**AB**
VSS	DVDD18	VSS	EMU06	EMU05	EMU00	EMU07	EMU08	**AC**
DVDD18	VSS	DVDD18	EMU01	DVDD18	VSS	EMU03	EMU02	**AD**
VSS	DVDD18	VSS	CORESEL0	$\overline{\text{RESETSTAT}}$	EXTFRAMEEVENT	$\overline{\text{RESETFULL}}$	EMU04	**AE**
RSV0B	VSS	RSV0A	RSV007	RSV008	$\overline{\text{RESET}}$	$\overline{\text{LRESETNMIEN}}$	TSPUSHEVT0	**AF**
DFEIO7	DFEIO2	CORESEL1	DVDD18	VSS	BOOTCOMPLETE	TSSYNCEVT	TSCOMPOUT	**AG**
DVDD18	DFEIO4	CORESEL2	DFEIO0	TDO	TMS	HOUT	TSPUSHEVT1	**AH**
VSS	DFEIO6	DFEIO11	DFEIO1	TDI	TCK	$\overline{\text{NMI}}$	DVDD18	**AJ**
DFEIO8	DFEIO9	DFEIO5	DFEIO3	$\overline{\text{TRST}}$	$\overline{\text{LRESET}}$	DVDD18	VSS	**AK**
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**	

Figure 6-6. 66AK2L06 Right End Panel (D) — Bottom View

6.3 Terminal Functions

The terminal functions table (Table 6-2) 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 6-3) lists the various power supply pins and ground pins and gives functional pin descriptions. Table 6-4 shows all pins arranged by signal name. Some pins have additional functions beyond their primary functions. There are pins that have a secondary function and pins that have a bootstrap function. Secondary functions are indicated with a superscript 2 (^{2}), and bootstrap functions are indicated with a superscript B (^{B}).

Table 6-5 shows all pins arranged by ball number.

For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see Section 9.2.

Use the symbol definitions in Table 6-1 when reading Table 6-2.

Table 6-1. I/O Functional Symbol Definitions

FUNCTIONAL SYMBOL	DEFINITION	Table 6-2 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. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see the *Hardware Design Guide for KeyStone II Devices* application report SPRABV0.	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 6-2. Terminal Functions — Signals and Control by Function

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
AVS Interface				
AVSIFSEL0^{B}	J2	IOZ	Down	AVS interface select 0 (^{B} pin is a secondary function and is shared with TIMI0)
AVSIFSEL1^{B}	J1	IOZ	Down	AVS interface select 1 (^{B} pin is a secondary function and is shared with TIMI1)
Common Serial Interface				
CSISC2_0_CLKCTL^{B}	J3	IOZ	Up	Selection of reference clock sharing scheme for CSISC2_0 and CSISC2_1 (^{B} pin is a secondary function and is shared with TIMO1)
CSISC2_0_MUX^{B}	H3	IOZ	Down	Selection between AIL and JESD (^{B} pin is a secondary function and is shared with TIMO0)
CSISC2_3_MUX^{B}	K26	IOZ	Down	Selection between SGMII and PCIe (^{B} pin is a secondary function and is shared with GPIO16)
Boot Configuration Pins				
BOOTMODE00^{B}	F27	IOZ	Down	User-defined boot mode pins. (^{B} pins are secondary functions and are shared with GPIO[01:08])
BOOTMODE01^{B}	F26	IOZ	Down	
BOOTMODE02^{B}	G29	IOZ	Down	
BOOTMODE03^{B}	F28	IOZ	Down	
BOOTMODE04^{B}	G27	IOZ	Down	
BOOTMODE05^{B}	H30	IOZ	Down	
BOOTMODE06^{B}	J26	IOZ	Down	
BOOTMODE07^{B}	H26	IOZ	Down	

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
BOOTMODE08[B]	H29	IOZ	Down	User-defined boot mode pins. ([B] pins are secondary functions and are shared with GPIO[09:13])
BOOTMODE09[B]	J27	IOZ	Down	
BOOTMODE10[B]	H28	IOZ	Down	
BOOTMODE11[B]	G28	IOZ	Down	
BOOTMODE12[B]	H27	IOZ	Down	
BOOTMODE13[B]	AE5	I	Down	Select for the target core for LRESET and NMI. ([B] pin is a secondary function and is shared with CORESEL0)
BOOTMODE14[B]	AG6	I	Down	User-defined boot mode pin. ([B] pin is a secondary function and is shared with CORESEL1)
BOOTMODE15[B]	AH6	I	Down	User-defined boot mode pin. ([B] pin is a secondary function and is shared with CORESEL2)
LENDIAN[B]	G26	IOZ	Up	Little endian configuration pin. ([B] pin is a secondary function and is shared with GPIO00)
MAINPLL_OD_SEL[B]	J30	IOZ	Down	Main PLL output divider select. ([B] pin is a secondary function and is shared with GPIO14)
Clock / Reset				
ALTCORECLKN	AH30	I		System clock input to antenna interface and main PLL (Main PLL optional vs. ALTCORECLK)
ALTCORECLKP	AG30	I		
BOOTCOMPLETE	AG3	O	Down	Boot progress indication output
CORECLKSEL0	AE27	I	Down	Ref clock select for core/ARM/PA PLL
CORECLKSEL1	AF27	I	Down	
CORESEL0	AE5	I	Down	Select for the target core for LRESET and NMI
CORESEL1	AG6	I	Down	
CORESEL2	AH6	I	Down	
DDR3ACLKN	G30	I		DDR3A reference clock input to DDR PLL
DDR3ACLKP	F30	I		
HOUT	AH2	O	Up	Interrupt output pulse created by IPCGRH
$\overline{\text{LRESET}}$	AK3	I	Up	Warm reset
$\overline{\text{LRESETNMIEN}}$	AF2	I	Up	Enable for core selects
$\overline{\text{NMI}}$	AJ2	I	Up	Non-maskable interrupt
PCIECLKN	AE19	I		PCIe reference clock to drive the PCIe SerDes. Not used when PCIe is not selected
PCIECLKP	AE20	I		
$\overline{\text{POR}}$	G4	I		Power-on reset
$\overline{\text{RESET}}$	AF3	I	Up	Warm reset of non isolated portion on the IC
$\overline{\text{RESETFULL}}$	AE2	I	Up	Full reset
$\overline{\text{RESETSTAT}}$	AE4	O	Up	Reset status output. Drives low during power-on reset (no HHV override). Available after core and IOs are are completely powered-up.
SGMIICLKN	AF24	I		SGMII reference clock to drive the SGMII SerDes
SGMIICLKP	AF23	I		
SYSCLKN	AG29	I		System clock input to antenna interface and main PLL (Main PLL optional vs. ALTCORECLK)
SYSCLKP	AF29	I		
SYSCLKOUT	AF28	O	Down	System clock output to be used as a general purpose output clock for debug purposes
TSREFCLKN	AK14	I		Clock from external OCXO/VCXO for SyncE
TSREFCLKP	AK13	I		
TSRXCLKOUT0N	AJ13	O		SerDes recovered clock output for SyncE.
TSRXCLKOUT0P	AJ12	O		

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDR3A				
DDR3ADQM0	E4	OZ		DDR3A EMIF data masks
DDR3ADQM1	E5	OZ		
DDR3ADQM2	A6	OZ		
DDR3ADQM3	B8	OZ		
DDR3ADQM4	F22	OZ		
DDR3ADQM5	A23	OZ		
DDR3ADQM6	C27	OZ		
DDR3ADQM7	E28	OZ		
DDR3ADQM8	A19	OZ		
DDR3ADQS0P	A3	IOZ	Up/Dn	DDR3A EMIF data strobe.
DDR3ADQS0N	B3	IOZ	Up/Dn	
DDR3ADQS1P	B4	IOZ	Up/Dn	
DDR3ADQS1N	A4	IOZ	Up/Dn	
DDR3ADQS2P	A7	IOZ	Up/Dn	
DDR3ADQS2N	B7	IOZ	Up/Dn	
DDR3ADQS3P	A10	IOZ	Up/Dn	
DDR3ADQS3N	B10	IOZ	Up/Dn	
DDR3ADQS4P	A22	IOZ	Up/Dn	
DDR3ADQS4N	B22	IOZ	Up/Dn	DDR3A EMIF data strobe.
DDR3ADQS5P	B24	IOZ	Up/Dn	
DDR3ADQS5N	A24	IOZ	Up/Dn	
DDR3ADQS6P	B26	IOZ	Up/Dn	
DDR3ADQS6N	A26	IOZ	Up/Dn	
DDR3ADQS7P	C29	IOZ	Up/Dn	
DDR3ADQS7N	B29	IOZ	Up/Dn	
DDR3ADQS8P	A20	IOZ	Up/Dn	
DDR3ADQS8N	B20	IOZ	Up/Dn	
DDR3ACB00	F19	IOZ		DDR3A EMIF Check Bits
DDR3ACB01	F20	IOZ		
DDR3ACB02	F18	IOZ		
DDR3ACB03	E20	IOZ		
DDR3ACB04	E19	IOZ		
DDR3ACB05	D19	IOZ		
DDR3ACB06	C19	IOZ		
DDR3ACB07	B19	IOZ		
DDR3AD00	E1	IOZ		DDR3A EMIF data bus
DDR3AD01	C2	IOZ		
DDR3AD02	B2	IOZ		
DDR3AD03	C1	IOZ		
DDR3AD04	D1	IOZ		
DDR3AD05	D2	IOZ		
DDR3AD06	E2	IOZ		
DDR3AD07	E3	IOZ		

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDR3AD08	A5	IOZ		DDR3A EMIF data bus
DDR3AD09	D6	IOZ		
DDR3AD10	E6	IOZ		
DDR3AD11	B5	IOZ		
DDR3AD12	F6	IOZ		
DDR3AD13	D4	IOZ		
DDR3AD14	C4	IOZ		
DDR3AD15	F5	IOZ		
DDR3AD16	F8	IOZ		DDR3A EMIF data bus
DDR3AD17	E7	IOZ		
DDR3AD18	C6	IOZ		
DDR3AD19	B6	IOZ		
DDR3AD20	F7	IOZ		
DDR3AD21	E8	IOZ		
DDR3AD22	D8	IOZ		
DDR3AD23	C8	IOZ		
DDR3AD24	F10	IOZ		DDR3A EMIF data bus
DDR3AD25	D10	IOZ		
DDR3AD26	E10	IOZ		
DDR3AD27	C10	IOZ		
DDR3AD28	A9	IOZ		
DDR3AD29	A8	IOZ		
DDR3AD30	B9	IOZ		
DDR3AD31	E9	IOZ		
DDR3AD32	A21	IOZ		DDR3A EMIF data bus
DDR3AD33	B21	IOZ		
DDR3AD34	C21	IOZ		
DDR3AD35	D21	IOZ		
DDR3AD36	E22	IOZ		
DDR3AD37	E21	IOZ		
DDR3AD38	E23	IOZ		
DDR3AD39	F23	IOZ		
DDR3AD40	F25	IOZ		DDR3A EMIF data bus
DDR3AD41	D25	IOZ		
DDR3AD42	C23	IOZ		
DDR3AD43	E25	IOZ		
DDR3AD44	E24	IOZ		
DDR3AD45	D23	IOZ		
DDR3AD46	F24	IOZ		
DDR3AD47	B23	IOZ		
DDR3AD48	C25	IOZ		DDR3A EMIF data bus
DDR3AD49	B25	IOZ		
DDR3AD50	A25	IOZ		
DDR3AD51	A27	IOZ		
DDR3AD52	E26	IOZ		
DDR3AD53	E27	IOZ		
DDR3AD54	D27	IOZ		
DDR3AD55	B27	IOZ		

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDR3AD56	E30	IOZ		DDR3A EMIF data bus
DDR3AD57	E29	IOZ		
DDR3AD58	F29	IOZ		
DDR3AD59	D30	IOZ		
DDR3AD60	C30	IOZ		
DDR3AD61	D29	IOZ		
DDR3AD62	B28	IOZ		
DDR3AD63	A28	IOZ		
$\overline{\text{DDR3ACE0}}$	A14	OZ		DDR3A EMIF chip enable
$\overline{\text{DDR3ACE1}}$	A11	OZ		
DDR3ABA0	E11	OZ		DDR3A EMIF bank address
DDR3ABA1	F12	OZ		
DDR3ABA2	E12	OZ		
DDR3AA00	C14	OZ		DDR3A EMIF address bus
DDR3AA01	D14	OZ		
DDR3AA02	A18	OZ		
DDR3AA03	E14	OZ		
DDR3AA04	C15	OZ		
DDR3AA05	A17	OZ		
DDR3AA06	D15	OZ		
DDR3AA07	B16	OZ		
DDR3AA08	B17	OZ		DDR3A EMIF address bus
DDR3AA09	D17	OZ		
DDR3AA10	B12	OZ		
DDR3AA11	B18	OZ		
DDR3AA12	C17	OZ		
DDR3AA13	B11	OZ		
DDR3AA14	E18	OZ		
DDR3AA15	E16	OZ		
$\overline{\text{DDR3ACAS}}$	A13	OZ		DDR3A EMIF column address strobe
$\overline{\text{DDR3ARAS}}$	B13	OZ		DDR3A EMIF row address strobe
$\overline{\text{DDR3AWE}}$	D12	OZ		DDR3A EMIF write enable
DDR3ACKE0	E17	OZ		DDR3A EMIF clock enable0
DDR3ACKE1	F17	OZ		DDR3A EMIF clock enable1
DDR3ACLKOUTP0	A16	OZ		DDR3A EMIF output clocks to drive SDRAMs (one clock pair per SDRAM)
DDR3ACLKOUTN0	A15	OZ		
DDR3ACLKOUTP1	B15	OZ		
DDR3ACLKOUTN1	B14	OZ		
DDR3AODT0	A12	OZ		DDR3A EMIF on-die termination outputs used to set termination on the SDRAMs
DDR3AODT1	F11	OZ		
$\overline{\text{DDR3ARESET}}$	E15	OZ		DDR3A reset signal
DDR3ARZQ0	F13	A		PTV Compensation Reference Resistor PAD for DDR3A
DDR3ARZQ1	F9	A		PTV Compensation Reference Resistor PAD for DDR3A
DDR3ARZQ2	F21	A		PTV Compensation Reference Resistor PAD for DDR3A

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DFE				
DFEIO0	AH5	IOZ	Down	DFE GPIO
DFEIO1	AJ5	IOZ	Down	
DFEIO2	AG7	IOZ	Down	
DFEIO3	AK5	IOZ	Down	
DFEIO4	AH7	IOZ	Down	
DFEIO5	AK6	IOZ	Down	
DFEIO6	AJ7	IOZ	Down	
DFEIO7	AG8	IOZ	Down	
DFEIO8	AK8	IOZ	Down	
DFEIO9	AK7	IOZ	Down	DFE GPIO
DFEIO10	AG10	IOZ	Down	
DFEIO11	AJ6	IOZ	Down	
DFEIO12	AK10	IOZ	Down	
DFEIO13	AK9	IOZ	Down	
DFEIO14	AF9	IOZ	Down	
DFEIO15	AK11	IOZ	Down	
DFEIO16	AG9	IOZ	Down	
DFEIO17	AH9	IOZ	Down	
DFESYNCINN0	AG13	I		JESD sync input A
DFESYNCINP0	AG12	I		
DFESYNCINN1	AF11	I		JESD sync input B
DFESYNCINP1	AF10	I		
DFESYNCOUTN0	AJ10	O		JESD sync output A
DFESYNCOUTP0	AJ9	O		
DFESYNCOUTN1	AH12	O		JESD sync output B
DFESYNCOUTP1	AH11	O		
DFESYSREFN	AF30	I		DFE sys ref data
DFESYSREFP	AE30	I		
EMIF16				
$\overline{\text{EMIFBE0}}$	AB30	IOZ	Up	EMIF control signals
$\overline{\text{EMIFBE1}}$	P29	IOZ	Up	
$\overline{\text{EMIFCE0}}$	N30	IOZ	Up	
$\overline{\text{EMIFCE1}}$	U30	IOZ	Up	
$\overline{\text{EMIFCE2}}$	W29	IOZ	Up	
$\overline{\text{EMIFCE3}}$	P30	IOZ	Up	
$\overline{\text{EMIFOE}}$	T30	IOZ	Up	
EMIF$\overline{\text{RW}}$	M30	IOZ	Up	
EMIFWAIT0	Y30	IOZ	Down	
EMIFWAIT1	AC30	IOZ	Down	
$\overline{\text{EMIFWE}}$	V30	IOZ	Up	
EMIFA00	W30	IOZ	Down	EMIF address
EMIFA01	R29	IOZ	Down	
EMIFA02	R30	IOZ	Down	
EMIFA03	P28	IOZ	Down	
EMIFA04	P26	IOZ	Down	
EMIFA05	R26	IOZ	Down	
EMIFA06	V29	IOZ	Down	
EMIFA07	AA30	IOZ	Down	

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
EMIFA08	T29	IOZ	Down	EMIF address
EMIFA09	R28	IOZ	Down	
EMIFA10	P27	IOZ	Down	
EMIFA11	AD30	IOZ	Down	
EMIFA12	V28	IOZ	Down	
EMIFA13	Y29	IOZ	Down	
EMIFA14	T28	IOZ	Down	
EMIFA15	Y28	IOZ	Down	
EMIFA16	T26	IOZ	Down	EMIF address
EMIFA17	T27	IOZ	Down	
EMIFA18	R27	IOZ	Down	
EMIFA19	AC29	IOZ	Down	
EMIFA20	AB29	IOZ	Down	
EMIFA21	U27	IOZ	Down	
EMIFA22	W28	IOZ	Down	
EMIFA23	V27	IOZ	Down	
EMIFD00	U26	IOZ	Down	EMIF data
EMIFD01	AB28	IOZ	Down	
EMIFD02	W27	IOZ	Down	
EMIFD03	AC28	IOZ	Down	
EMIFD04	Y27	IOZ	Down	
EMIFD05	AB27	IOZ	Down	
EMIFD06	AA27	IOZ	Down	
EMIFD07	V26	IOZ	Down	
EMIFD08	AD27	IOZ	Down	EMIF data
EMIFD09	AC27	IOZ	Down	
EMIFD10	W26	IOZ	Down	
EMIFD11	AD26	IOZ	Down	
EMIFD12	AC26	IOZ	Down	
EMIFD13	AA26	IOZ	Down	
EMIFD14	Y26	IOZ	Down	
EMIFD15	AB26	IOZ	Down	
EMU				
EMU00	AC3	IOZ	Up	Emulation and trace port
EMU01	AD5	IOZ	Up	
EMU02	AD1	IOZ	Up	
EMU03	AD2	IOZ	Up	
EMU04	AE1	IOZ	Up	
EMU05	AC4	IOZ	Up	
EMU06	AC5	IOZ	Up	
EMU07	AC2	IOZ	Up	
EMU08	AC1	IOZ	Up	Emulation and trace port
EMU09	AB4	IOZ	Up	
EMU10	AB5	IOZ	Up	
EMU11	AB3	IOZ	Up	
EMU12	AA1	IOZ	Up	
EMU13	AA4	IOZ	Up	
EMU14	AA2	IOZ	Up	
EMU15	AA3	IOZ	Up	

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
EMU16	AA5	IOZ	Up	Emulation and trace port
EMU17	Y3	IOZ	Up	
EMU18	Y2	IOZ	Up	
EMU19[2]	W2	IOZ	Down	Emulation and Trace Port ([2] pins are secondary functions and are shared with GPIO[17:24])
EMU20[2]	Y1	IOZ	Down	
EMU21[2]	V3	IOZ	Down	
EMU22[2]	W1	IOZ	Down	
EMU23[2]	V1	IOZ	Down	
EMU24[2]	V2	IOZ	Down	
EMU25[2]	U4	IOZ	Down	
EMU26[2]	V5	IOZ	Down	
EMU27[2]	V4	IOZ	Down	Emulation and Trace Port ([2] pins are secondary functions and are shared with GPIO[25:31])
EMU28[2]	U1	IOZ	Down	
EMU29[2]	T3	IOZ	Down	
EMU30[2]	U3	IOZ	Down	
EMU31[2]	T5	IOZ	Down	
EMU32[2]	U5	IOZ	Down	
EMU33[2]	T4	IOZ	Down	
General Purpose Input/Output (GPIO)				
GPIO00	G26	IOZ	Up	GPIO
GPIO01	F27	IOZ	Down	
GPIO02	F26	IOZ	Down	
GPIO03	G29	IOZ	Down	
GPIO04	F28	IOZ	Down	
GPIO05	G27	IOZ	Down	
GPIO06	H30	IOZ	Down	
GPIO07	J26	IOZ	Down	
GPIO08	H26	IOZ	Down	GPIO
GPIO09	H29	IOZ	Down	
GPIO10	J27	IOZ	Down	
GPIO11	H28	IOZ	Down	
GPIO12	G28	IOZ	Down	
GPIO13	H27	IOZ	Down	
GPIO14	J30	IOZ	Down	
GPIO15	K27	IOZ	Down	
GPIO16	K26	IOZ	Down	GPIO
GPIO17	W2	IOZ	Down	
GPIO18	Y1	IOZ	Down	
GPIO19	V3	IOZ	Down	
GPIO20	W1	IOZ	Down	
GPIO21	V1	IOZ	Down	
GPIO22	V2	IOZ	Down	
GPIO23	U4	IOZ	Down	

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
GPIO24	V5	IOZ	Down	GPIO
GPIO25	V4	IOZ	Down	
GPIO26	U1	IOZ	Down	
GPIO27	T3	IOZ	Down	
GPIO28	U3	IOZ	Down	
GPIO29	T5	IOZ	Down	
GPIO30	U5	IOZ	Down	
GPIO31	T4	IOZ	Down	
GPIO32[2]	W30	IOZ	Down	GPIO ([2] pins are secondary functions and are shared with EMIFA[00:07])
GPIO33[2]	R29	IOZ	Down	
GPIO34[2]	R30	IOZ	Down	
GPIO35[2]	P28	IOZ	Down	
GPIO36[2]	P26	IOZ	Down	
GPIO37[2]	R26	IOZ	Down	
GPIO38[2]	V29	IOZ	Down	
GPIO39[2]	AA30	IOZ	Down	
GPIO40[2]	T29	IOZ	Down	GPIO ([2] pins are secondary functions and are shared with EMIFA[08:10], EMIFA[13:17])
GPIO41[2]	R28	IOZ	Down	
GPIO42[2]	P27	IOZ	Down	
GPIO43[2]	Y29	IOZ	Down	
GPIO44[2]	T28	IOZ	Down	
GPIO45[2]	Y28	IOZ	Down	
GPIO46[2]	T26	IOZ	Down	
GPIO47[2]	T27	IOZ	Down	
GPIO48[2]	AG7	IOZ	Down	GPIO ([2] pins are secondary functions and are shared with DFEIO[02:09]
GPIO49[2]	AK5	IOZ	Down	
GPIO50[2]	AH7	IOZ	Down	
GPIO51[2]	AK6	IOZ	Down	
GPIO52[2]	AJ7	IOZ	Down	
GPIO53[2]	AG8	IOZ	Down	
GPIO54[2]	AK8	IOZ	Down	
GPIO55[2]	AK7	IOZ	Down	
GPIO56[2]	AG10	IOZ	Down	GPIO ([2] pins are secondary functions and are shared with DFEIO[10:17]
GPIO57[2]	AJ6	IOZ	Down	
GPIO58[2]	AK10	IOZ	Down	
GPIO59[2]	AK9	IOZ	Down	
GPIO60[2]	AF9	IOZ	Down	
GPIO61[2]	AK11	IOZ	Down	
GPIO62[2]	AG9	IOZ	Down	
GPIO63[2]	AH9	IOZ	Down	
I²C				
SCL0	N2	IOZ		I²C0 clock
SCL1	L3	IOZ		I²C1 clock
SCL2	M3	IOZ		I²C2 clock
SDA0	M2	IOZ		I²C0 data
SDA1	M5	IOZ		I²C1 data
SDA2	M4	IOZ		I²C2 data

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
IQN				
EXTFRAMEEVENT	AE3	O	Down	CPRI, OBSAI or radio external frame reference
PHYSYNC	AJ29	I	Down	Non-OBSAI CPRI input sync
RADSYNC	AH28	I	Down	Non-OBSAI radio sync
RP1CLKN[2]	AG13	I		OBSAI RP1 sync clock ([2] pin is a secondary function and is shared with DFESYNCINN0)
RP1CLKP[2]	AG12	I		OBSAI RP1 sync clock ([2] pin is a secondary function and is shared with DFESYNCINP0)
RP1FBN[2]	AF11	I		OBSAI RP1 sync ([2] pin is a secondary function and is shared with DFESYNCINN1)
RP1FBP[2]	AF10	I		OBSAI RP1 sync ([2] pin is a secondary function and is shared with DFESYNCINP1)
JTAG				
TCK	AJ3	I	Up	JTAG clock input
TDI	AJ4	I	Up	JTAG data input
TDO	AH4	OZ	Up	JTAG data output
TMS	AH3	I	Up	JTAG test mode input
$\overline{\text{TRST}}$	AK4	I	Down	JTAG reset
MDIO				
MDCLK	G2	O	Down	MDIO Clock
MDIO	H1	IOZ	Up	MDIO Data
JESD				
SHARED_SERDES_0_REFCLKN	AF18	I		Clock for CSISC2_0 B4 SerDes Marco
SHARED_SERDES_0_REFCLKP	AF17	I		
SHARED_SERDES_0_REFRES	AE17	A		CSISC2_0 SerDes reference resistor input (3 kΩ ±1%)
SHARED_SERDES_1_REFCLKN	AF15	I		Clock for CSISC2_1
SHARED_SERDES_1_REFCLKP	AF14	I		
SHARED_SERDES_1_REFRES	AE13	A		CSISC2_1 macro reference resistor input (3 kΩ ±1%)
SHARED_SERDES_1_RXN0	AJ16	I		CSISC2_1 RX
SHARED_SERDES_1_RXN1	AK17	I		
SHARED_SERDES_1_RXP0	AJ15	I		
SHARED_SERDES_1_RXP1	AK16	I		
SHARED_SERDES_1_TXN0	AH14	O		CSISC2_1 TX
SHARED_SERDES_1_TXN1	AG15	O		
SHARED_SERDES_1_TXP0	AH15	O		
SHARED_SERDES_1_TXP1	AG16	O		
SHARED_SERDES_2_RXN0	AJ25	I		Ethernet MAC SGMII receive data
SHARED_SERDES_2_RXP0	AJ24	I		
SHARED_SERDES_2_TXN0	AH23	O		Ethernet MAC SGMII transmit data
SHARED_SERDES_2_TXP0	AH24	O		

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
SHARED_SERDES_2_RXN1	AK26	I		Ethernet MAC SGMII receive data
SHARED_SERDES_2_RXP1	AK25	I		
SHARED_SERDES_2_TXN1	AG24	O		Ethernet MAC SGMII transmit data
SHARED_SERDES_2_TXP1	AG25	O		
SHARED_SERDES_2_REFRES	AE23	A		SGMII SerDes reference resistor input (3 kΩ ±1%)
SHARED_SERDES_0_RXN0	AJ19	I		CSISC2_0 RX
SHARED_SERDES_0_RXN1	AK20	I		
SHARED_SERDES_0_RXP0	AJ18	I		
SHARED_SERDES_0_RXP1	AK19	I		
SHARED_SERDES_0_TXN0	AH17	O		CSISC2_0 TX
SHARED_SERDES_0_TXN1	AG18	O		
SHARED_SERDES_0_TXP0	AH18	O		
SHARED_SERDES_0_TXP1	AG19	O		
SGMII/PCIe				
SHARED_SERDES_3_RXN0	AJ22	I		Ethernet MAC SGMII or PCIe receive data
SHARED_SERDES_3_RXP0	AJ21	I		
SHARED_SERDES_3_TXN0	AH20	O		Ethernet MAC SGMII or PCIe transmit data
SHARED_SERDES_3_TXP0	AH21	O		
SHARED_SERDES_3_RXN1	AK23	I		Ethernet MAC SGMII or PCIe receive data
SHARED_SERDES_3_RXP1	AK22	I		
SHARED_SERDES_3_TXN1	AG21	O		Ethernet MAC SGMII or PCIe transmit data
SHARED_SERDES_3_TXP1	AG22	O		
SHARED_SERDES_3_REFRES	AE22	A		SGMII/PCIe SerDes reference resistor input (3 kΩ ±1%)
SmartReflex				
VCL[2]	AH27	IOZ		Voltage Control I[2]C Clock ([2] pin is a secondary function and is shared with VCNTL4)
VCNTL0	AJ27	IOZ		Voltage Control Outputs to variable core power supply
VCNTL1	AK28	IOZ		
VCNTL2	AJ28	OZ		
VCNTL3	AG27	OZ		
VCNTL4	AH27	OZ		
VCNTL5	AH26	OZ		
VD[2]	AH26	IOZ		Voltage Control I[2]C Data ([2] pin is a secondary function and is shared with VCNTL5)
SPI0				
SPI0CLK	L29	IOZ	Down	SPI0 clock
SPI0DIN	N27	IOZ	Down	SPI0 data In
SPI0DOUT	L27	IOZ	Down	SPI0 data out

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
SPI0SCS0	L30	IOZ	Up	SPI0 interface enable 0
SPI0SCS1	K28	IOZ	Up	SPI0 interface enable 1
SPI0SCS2	K30	IOZ	Up	SPI0 interface enable 2
SPI0SCS3	K29	IOZ	Up	SPI0 interface enable 3
SPI0SCS4	M29	IOZ	Up	SPI0 interface enable 4
SPI1				
SPI1CLK	M26	IOZ	Down	SPI1 clock
SPI1DIN	L26	IOZ	Down	SPI1 data in
SPI1DOUT	N26	IOZ	Down	SPI1 data out
SPI1SCS0	M28	IOZ	Up	SPI1 interface enable 0
SPI1SCS1	L28	IOZ	Up	SPI1 interface enable 1
SPI1SCS2	M27	IOZ	Up	SPI1 interface enable 2
SPI2				
SPI2CLK[2]	L4	IOZ	Up	SPI2 clock ([2] pin is a secondary function and is shared with UART0CTS)
SPI2SCS0[2]	K2	IOZ	Up	SPI2 interface enable 0 ([2] pin is a secondary function and is shared with UART0RTS)
SPI2SCS1[2]	G26	IOZ	Up	SPI2 interface enable 1 ([2] pin is a secondary function and is shared with GPIO00)
SPI2SCS2[2]	F27	IOZ	Down	SPI2 interface enable 2 ([2] pin is a secondary function and is shared with GPIO01)
SPI2SCS3[2]	F26	IOZ	Down	SPI2 interface enable 3 ([2] pin is a secondary function and is shared with GPIO02)
SPI2SCS4[2]	G29	IOZ	Down	SPI2 interface enable 4 ([2] pin is a secondary function and is shared with GPIO03)
SPI2DIN[2]	J4	IOZ	Up	SPI2 data in ([2] pin is a secondary function and is shared with UART1RTS)
SPI2DOUT[2]	K4	IOZ	Up	SPI2 data out ([2] pin is a secondary function and is shared with UART1CTS)
Sync-Ethernet / IEEE1588				
TSCOMPOUT	AG1	O	Down	IEEE1588 compare output
TSPUSHEVT0	AF1	I	Down	PPS push event from GPS for IEEE1588
TSPUSHEVT1	AH1	I	Down	Push event from BCN for IEEE1588
TSSYNCEVT	AG2	O	Down	IEEE1588 sync event output
Timer				
TIMI0	J2	IOZ	Down	Timer inputs
TIMI1	J1	IOZ	Down	
TIMI2[2]	F28	IOZ	Down	Timer inputs ([2] pins are secondary functions and are shared with GPIO[04:07])
TIMI3[2]	G27	IOZ	Down	
TIMI4[2]	H30	IOZ	Down	
TIMI5[2]	J26	IOZ	Down	
TIMI6[2]	H26	IOZ	Down	
TIMI7[2]	H29	IOZ	Down	
TIMO0	H3	IOZ	Down	Timer outputs
TIMO1	J3	IOZ	Up	
TIMO2[2]	J27	IOZ	Down	Timer outputs ([2] pins are secondary functions and are shared with GPIO[10:15])
TIMO3[2]	H28	IOZ	Down	
TIMO4[2]	G28	IOZ	Down	
TIMO5[2]	H27	IOZ	Down	
TIMO6[2]	J30	IOZ	Down	
TIMO7[2]	K27	IOZ	Down	
UART0				
UART0CTS	L4	I	Up	UART0
UART0RTS	K2	O	Up	
UART0RXD	K5	I	Down	
UART0TXD	K3	O	Down	
UART1				

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
UART1CTS	K4	I	Up	UART1
UART1RTS	J4	O	Up	
UART1RXD	L5	I	Up	
UART1TXD	J5	O	Up	
UART2				
UART2CTS[2]	AC29	I	Down	UART2 ([2] pin is a secondary function and is shared with EMIFA19)
UART2RTS[2]	R27	O	Down	UART2 ([2] pin is a secondary function and is shared with EMIFA18)
UART2RXD[2]	AB29	I	Down	UART2 ([2] pin is a secondary function and is shared with EMIFA20)
UART2TXD[2]	U27	O	Down	UART2 ([2] pin is a secondary function and is shared with EMIFA21)
UART3				
UART3CTS[2]	P30	I	Up	UART3 ([2] pin is a secondary function and is shared with $\overline{\text{EMIFCE3}}$)
UART3RTS[2]	W29	O	Up	UART3 ([2] pin is a secondary function and is shared with $\overline{\text{EMIFCE2}}$)
UART3RXD[2]	W28	I	Down	UART3 ([2] pin is a secondary function and is shared with EMIFA22)
UART3TXD[2]	V27	O	Down	UART3 ([2] pin is a secondary function and is shared with EMIFA23)
USB 3.0				
USBCLKM	R3	I		USB 3.0 ref clock
USBCLKP	P3	I		
USBDM	R1	IOZ		USB D-
USBDP	P1	IOZ		USB D+
USBDRVVBUS	M6	O	Down	USB DRVVBUS output
USBID0	N5	A		USB ID
USBRESREF	M8	A		Reference resistor connection for USB PHY 200 Ω ±1%
USBRX0M	T2	I		USB 3.0 receive data
USBRX0P	R2	I		
USBTX0M	M1	O		USB 3.0 transmit data
USBTX0P	L1	O		
USBVBUS	N4	A		USB 5-V line presence detect
USIM				
USIMCLK	F3	O	Down	USIM clock
USIMIO	F2	IOZ	Up	USIM data
USIMRST	F1	O	Down	USIM reset
Reserved				
RSV0A	AF6			Unconnected
RSV0B	AF8			Unconnected
RSV001	C12	A		Unconnected
RSV002	N23	P		Unconnected
RSV003	M23	P		Unconnected
RSV004	AE29	O		Unconnected
RSV005	AE28	O		Unconnected
RSV006	AH29	O	Down	Unconnected
RSV007	AF5	A		Connect to GND
RSV008	AF4	A		Unconnected
RSV009	Y4	A		Unconnected
RSV010	W5	A		Unconnected
RSV011	J25	A		Unconnected
RSV012	H25	A		Unconnected
RSV013	AF22	A		Unconnected
RSV014	AE21	A		Unconnected
RSV015	E13	OZ		Unconnected
RSV016	F14	OZ		Unconnected

Table 6-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
RSV017	AF20	A		Unconnected
RSV018	AE15	A		Unconnected

Table 6-3. Terminal Functions — Power and Ground

SUPPLY	BALL NO.	VOLTS	DESCRIPTION
AVDDA1	J23	1.8	SYS_CLK PLL power supply
AVDDA2	U25	1.8	DDR3A_CLK PLL power supply
AVDDA3	AF26	1.8	PS_SS_CLK PLL power supply
AVDDA4	AE26	1.8	ARM CLK PLL power supply
AVDDA5	AF12	1.8	DFE PLL power supply
AVDDA6	K8	1.8	DDRA DLL supply
AVDDA7	J11	1.8	DDRA DLL supply
AVDDA8	J14	1.8	DDRA DLL supply
AVDDA9	J17	1.8	DDRA DLL supply
AVDDA10	J21	1.8	DDRA DLL supply
AVDDAS	AD12, AD14, AD16, AD18, AD20, AD22	1.8	SerDes IO supply
CVDD	K12, K14, K16, K18, K20, K22, L9, L11, L13, L15, L17, L19, L21, M14, M18, M20, M22, N9, N13, N17, P10, P12, P14, P18, P22, R11, T12, , T14, T16, T18, T20, T22, U11, U13, U15, U17, U19, U21, V12, V14, V16, V18, V20, V22, W9, W11, W13, W17, W19, W21, Y10, Y12, Y18, Y20, Y22, AA9, AA11, AA13, AA15, AA17, AA19, AA21, AB10, AB12, AB20, AB22	AVS	SmartReflex DSP core supply voltage
CVDD1	M10, M12, M16, N11, N15, N19, N21, P16, P20, W15, Y14, Y16	0.95	Core supply voltage for memory array
CVDDS	AB14, AB16, AB18, AC13, AC15, AC17, AC19	0.85	SerDes low voltage
DDR3AVREFSSTL	F15	0.75	0.75-V DDR3 reference voltage
DVDD18	J29, K24, L25, M24, N25, N28, P24, R7, R23, R25, T6, T8, T24, U7, U23, U28, V6, V24, W3, W7, W23, W25, Y6, Y24, AA23, AA25, AA28, AB2, AB6, AB8, AB24, AC7, AC9, AC11, AC23, AC25, AD4, AD6, AD8, AD10, AD24, AD28, AE7, AE9, AE11, AE25, AG5, AH8, AJ1, AJ30, AK2, AK29	1.8	1.8-V IO supply
DVDD33	R9	3.3	3.3-V USB supply
DVDDR	A2, A29, B1, B30, C3, C7, C11, C16, C20, C24, C28, D5, D9, D13, D18, D22, D26, G7, G9, G11, G13, G15, G17, G19, G21, G23, G25, H6, H8, H10, H12, H14, H16, H18, H20, H22, H24, J7, J9, J13, J15, J19, AA7	1.5 or 1.35	1.5-V/1.35-V DDR IO supply
VDDCMON	K10	0.95	Connect to CVDDS
VDDUSB	U9	0.85	USB0 PHY analog and digital highspeed supply
VNWA1	L7	0.95	Nwell bias supply voltage for core memories
VNWA2	L23	0.95	Nwell bias supply voltage for core memories
VNWA3	AC21	0.95	Nwell bias supply voltage for core memories
VNWA4	Y8	0.95	Nwell bias supply voltage for core memories
VSSCMON	K9	GND	Ground
VP	V8	0.85	PHY analog and digital SuperSpeed supply. Filtered 0.85-V supply.
VPH	T10	3.3	PHY high supply for SuperSpeed. Filtered 3.3-V USB supply.
VPTX	V10	0.85	PHY transmit supply. Filtered 0.85-V supply. Connect to CVDDS through a ferrite bead.
VSS	A1, A30, C5, C9, C13, C18, C22, C26, D3, D7, D11, D16, D20, D24, D28, F4, F16, G1, G3, G5, G6, G8, G10, G12, G14, G16, G18, G20, G22, G24, H2, H4, H5, H7, H9, H11, H13, H15, H17, H19, H21, H23, J6, J8, J10, J12, J16, J18, J20, J22, J24, J28, K1, K6, K7, K11, K13, K15, K17, K19, K21, K23, K25, L2, L6, L8, L10, L12, L14, L16, L18, L20, L22, L24, M7, M9, M11, M13, M15, M17, M19, M21, M25, N1, N3, N6, N7, N8, N10, N12, N14, N16, N18, N20, N22, N24, N29, P2, P4, P5, P6, P7, P8, P9, P11, P13, P15, P17, P19, P21, P23, P25, R4, R5, R6, R8, R10, R12, R14, R16, R18, R20, , R22, R24, T1, T7, T9, T11, T13, T15, T17, T19, T21, T23, T25, U2, U6, U8, U10, U12, U14, U16, U18, U20, U22, U24, U29, V7, V9, V11, V13, V15, V17, V19, V21, V23, V25, W4, W6, W8, W10, W12, W14, W16, W18, W20, W22, W24, Y5, Y7, Y9, Y11, Y13, , Y15, Y17, Y19, Y21, Y23, Y25, AA6, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AA24, AA29, AB1, AB7, AB9, AB11, AB13, AB15, AB17, AB19, AB21, AB23, AB25, AC6, AC8, AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AD3, AD7, AD9, AD11, AD13, AD15, AD17, AD19, AD21, AD23, AD25, AD29, AE6, AE8, AE10, AE12, AE14, AE16, AE18, AE24, AF7, AF13, AF16, AF19, AF21, AF25, AG4, AG11, AG14, AG17, AG20, AG23, AG26, AG28, AH10, AH13, AH16, AH19, AH22, AH25, AJ8, AJ11, AJ14, AJ17, AJ20, AJ23, AJ26, AK1, AK12, AK15, AK18, AK21, AK24, AK27, AK30	GND	Ground

Table 6-4. Terminal Functions — By Signal Name

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
AVDDAS	AD12, AD14, AD16, AD18, AD20, AD22	CVDD1	M10, M12, M16, N11, N15, N19, N21, P16, P20, W15, Y14, Y16	DDR3ACLKP	F30
				DDR3AD00	E1
AVSIFSEL0 [B]	J2	CVDDS	AB14, AB16, AB18, AC13, AC15, AC17, AC19	DDR3AD01	C2
AVSIFSEL1 [B]	J1			DDR3AD02	B2
BOOTCOMPLETE	AG3	DDR3AA00	C14	DDR3AD03	C1
BOOTMODE00 [B]	F27	DDR3AA01	D14	DDR3AD04	D1
BOOTMODE01 [B]	F26	DDR3AA02	A18	DDR3AD05	D2
BOOTMODE02 [B]	G29	DDR3AA03	E14	DDR3AD06	E2
BOOTMODE03 [B]	F28	DDR3AA04	C15	DDR3AD07	E3
BOOTMODE04 [B]	G27	DDR3AA05	A17	DDR3AD08	A5
BOOTMODE05 [B]	H30	DDR3AA06	D15	DDR3AD09	D6
BOOTMODE06 [B]	J26	DDR3AA07	B16	DDR3AD10	E6
BOOTMODE07 [B]	H26	DDR3AA08	B17	DDR3AD11	B5
BOOTMODE08 [B]	H29	DDR3AA09	D17	DDR3AD12	F6
BOOTMODE09 [B]	J27	DDR3AA10	B12	DDR3AD13	D4
BOOTMODE10 [B]	H28	DDR3AA11	B18	DDR3AD14	C4
BOOTMODE11 [B]	G28	DDR3AA12	C17	DDR3AD15	F5
BOOTMODE12 [B]	H27	DDR3AA13	B11	DDR3AD16	F8
BOOTMODE13 [B]	AE5	DDR3AA14	E18	DDR3AD17	E7
BOOTMODE14 [B]	AG6	DDR3AA15	E16	DDR3AD18	C6
BOOTMODE15 [B]	AH6	DDR3ABA0	E11	DDR3AD19	B6
CORECLKSEL0	AE27	DDR3ABA1	F12	DDR3AD20	F7
CORECLKSEL1	AF27	DDR3ABA2	E12	DDR3AD21	E8
CORESEL0	AE5	$\overline{\text{DDR3ACAS}}$	A13	DDR3AD22	D8
CORESEL1	AG6	DDR3ACB00	F19	DDR3AD23	C8
CORESEL2	AH6	DDR3ACB01	F20	DDR3AD24	F10
CSISC2_0_CLKCTL[B]	J3	DDR3ACB02	F18	DDR3AD25	D10
CSISC2_0_MUX [B]	H3	DDR3ACB03	E20	DDR3AD26	E10
CSISC2_3_MUX [B]	K26	DDR3ACB04	E19	DDR3AD27	C10
CVDD	K12, K14, K16, K18, K20, K22, L9, L11, L13, L15, L17, L19, L21, M14, M18, M20, M22, N9, N13, N17, P10, P12, P14, P18, P22, R11, T12, , T14, T16, T18, T20, T22, U11	DDR3ACB05	D19	DDR3AD28	A9
		DDR3ACB06	C19	DDR3AD29	A8
		DDR3ACB07	B19	DDR3AD30	B9
		$\overline{\text{DDR3ACE0}}$	A14	DDR3AD31	E9
CVDD	U13, U15, U17, U19, U21, V12, V14, V16, V18, V20, V22, W9, W11, , W13, W17, W19, W21, Y10, Y12, Y18, Y20, Y22, AA9, AA11, AA13, AA15, AA17, AA19, AA21	$\overline{\text{DDR3ACE1}}$	A11	DDR3AD32	A21
		DDR3ACKE0	E17	DDR3AD33	B21
		DDR3ACKE1	F17	DDR3AD34	C21
		DDR3ACLKN	G30	DDR3AD35	D21
		DDR3ACLKOUTN0	A15	DDR3AD36	E22
		DDR3ACLKOUTN1	B14	DDR3AD37	E21
		DDR3ACLKOUTP0	A16	DDR3AD38	E23
		DDR3ACLKOUTP1	B15	DDR3AD39	F23
CVDD	AB10, AB12, AB20, AB22			DDR3AD40	F25

Table 6-4. Terminal Functions — By Signal Name (continued)

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
DDR3AD41	D25	DDR3ADQS5N	A24	DFESYNCOUTP1	AH11
DDR3AD42	C23	DDR3ADQS5P	B24	DFESYSREFN	AF30
DDR3AD43	E25	DDR3ADQS6N	A26	DFESYSREFP	AE30
DDR3AD44	E24	DDR3ADQS6P	B26	DVDD18	J29, K24, L25, M24, N25, N28, P24, R7, R23, R25, T6, T8, T24, U7, U23, U28, V6, V24, W3, W7, W23, W25, Y6, Y24, AA23, AA25, AA28, AB2, AB6, AB8, AB24, AC7, AC9
DDR3AD45	D23	DDR3ADQS7N	B29		
DDR3AD46	F24	DDR3ADQS7P	C29		
DDR3AD47	B23	DDR3ADQS8N	B20		
DDR3AD48	C25	DDR3ADQS8P	A20		
DDR3AD49	B25	DDR3AODT0	A12		
DDR3AD50	A25	DDR3AODT1	F11	DVDD18	AC11, AC23, AC25, AD4, AD6, AD8, AD10, AD24, AD28, AE7, AE9, AE11, AE25, AG5, AH8, AJ1, AJ30, AK2, AK29
DDR3AD51	A27	$\overline{\text{DDR3ARAS}}$	B13		
DDR3AD52	E26	$\overline{\text{DDR3ARESET}}$	E15		
DDR3AD53	E27	DDR3ARZQ0	F13		
DDR3AD54	D27	DDR3ARZQ1	F9	DVDD33	R9
DDR3AD55	B27	DDR3ARZQ2	F21	DVDDR	A2, A29, B1, B30, C3, C7, C11, C16, C20, C24, C28, D5, D9, D13, D18, D22, D26, G7, G9, G11, G13, G15, G17, G19, G21, G23, G25, H6, H8, H10, H12, H14, H16
DDR3AD56	E30	DDR3AVREFSSTL	F15		
DDR3AD57	E29	$\overline{\text{DDR3AWE}}$	D12		
DDR3AD58	F29	DFEIO0	AH5		
DDR3AD59	D30	DFEIO1	AJ5		
DDR3AD60	C30	DFEIO10	AG10		
DDR3AD61	D29	DFEIO11	AJ6	DVDDR	H18, H20, H22, H24, J7, J9, J13, J15, J19, AA7
DDR3AD62	B28	DFEIO12	AK10		
DDR3AD63	A28	DFEIO13	AK9	EMIFA00	W30
DDR3ADQM0	E4	DFEIO14	AF9	EMIFA01	R29
DDR3ADQM1	E5	DFEIO15	AK11	EMIFA02	R30
DDR3ADQM2	A6	DFEIO16	AG9	EMIFA03	P28
DDR3ADQM3	B8	DFEIO17	AH9	EMIFA04	P26
DDR3ADQM4	F22	DFEIO2	AG7	EMIFA05	R26
DDR3ADQM5	A23	DFEIO3	AK5	EMIFA06	V29
DDR3ADQM6	C27	DFEIO4	AH7	EMIFA07	AA30
DDR3ADQM7	E28	DFEIO5	AK6	EMIFA08	T29
DDR3ADQM8	A19	DFEIO6	AJ7	EMIFA09	R28
DDR3ADQS0N	B3	DFEIO7	AG8	EMIFA10	P27
DDR3ADQS0P	A3	DFEIO8	AK8	EMIFA11	AD30
DDR3ADQS1N	A4	DFEIO9	AK7	EMIFA12	V28
DDR3ADQS1P	B4	DFESYNCINN0	AG13	EMIFA13	Y29
DDR3ADQS2N	B7	DFESYNCINN1	AF11	EMIFA14	T28
DDR3ADQS2P	A7	DFESYNCINP0	AG12	EMIFA15	Y28
DDR3ADQS3N	B10	DFESYNCINP1	AF10	EMIFA16	T26
DDR3ADQS3P	A10	DFESYNCOUTN0	AJ10	EMIFA17	T27
DDR3ADQS4N	B22	DFESYNCOUTN1	AH12	EMIFA18	R27
DDR3ADQS4P	A22	DFESYNCOUTP0	AJ9	EMIFA19	AC29

Table 6-4. Terminal Functions — By Signal Name (continued)

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
EMIFA20	AB29	EMU11	AB3	GPIO18	Y1
EMIFA21	U27	EMU12	AA1	GPIO19	V3
EMIFA22	W28	EMU13	AA4	GPIO20	W1
EMIFA23	V27	EMU14	AA2	GPIO21	V1
EMIFBE0	AB30	EMU15	AA3	GPIO22	V2
EMIFBE1	P29	EMU16	AA5	GPIO23	U4
EMIFCE0	N30	EMU17	Y3	GPIO24	V5
EMIFCE1	U30	EMU18	Y2	GPIO25	V4
EMIFCE2	W29	EMU19 [2]	W2	GPIO26	U1
EMIFCE3	P30	EMU20 [2]	Y1	GPIO27	T3
EMIFD00	U26	EMU21 [2]	V3	GPIO28	U3
EMIFD01	AB28	EMU22 [2]	W1	GPIO29	T5
EMIFD02	W27	EMU23 [2]	V1	GPIO30	U5
EMIFD03	AC28	EMU24 [2]	V2	GPIO31	T4
EMIFD04	Y27	EMU25 [2]	U4	GPIO32[2]	W30
EMIFD05	AB27	EMU26 [2]	V5	GPIO33[2]	R29
EMIFD06	AA27	EMU27 [2]	V4	GPIO34[2]	R30
EMIFD07	V26	EMU28 [2]	U1	GPIO35[2]	P28
EMIFD08	AD27	EMU29 [2]	T3	GPIO36[2]	P26
EMIFD09	AC27	EMU30 [2]	U3	GPIO37[2]	R26
EMIFD10	W26	EMU31 [2]	T5	GPIO38[2]	V29
EMIFD11	AD26	EMU32 [2]	U5	GPIO39[2]	AA30
EMIFD12	AC26	EMU33 [2]	T4	GPIO40[2]	T29
EMIFD13	AA26	EXTFRAMEEVENT	AE3	GPIO41[2]	R28
EMIFD14	Y26	GPIO00	G26	GPIO42[2]	P27
EMIFD15	AB26	GPIO01	F27	GPIO43[2]	Y29
EMIFOE	T30	GPIO02	F26	GPIO44[2]	T28
EMIFR W	M30	GPIO03	G29	GPIO45[2]	Y28
EMIFWAIT0	Y30	GPIO04	F28	GPIO46[2]	T26
EMIFWAIT1	AC30	GPIO05	G27	GPIO47[2]	T27
EMIFWE	V30	GPIO06	H30	GPIO48[2]	AG7
EMU00	AC3	GPIO07	J26	GPIO49[2]	AK5
EMU01	AD5	GPIO08	H26	GPIO50[2]	AH7
EMU02	AD1	GPIO09	H29	GPIO51[2]	AK6
EMU03	AD2	GPIO10	J27	GPIO52[2]	AJ7
EMU04	AE1	GPIO11	H28	GPIO53[2]	AG8
EMU05	AC4	GPIO12	G28	GPIO54[2]	AK8
EMU06	AC5	GPIO13	H27	GPIO55[2]	AK7
EMU07	AC2	GPIO14	J30	GPIO56[2]	AG10
EMU08	AC1	GPIO15	K27	GPIO57[2]	AJ6
EMU09	AB4	GPIO16	K26	GPIO58[2]	AK10
EMU10	AB5	GPIO17	W2	GPIO59[2]	AK9

Table 6-4. Terminal Functions — By Signal Name (continued)

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
GPIO60[2]	AF9	SDA0	M2	SHARED_SERDES_1_TXP1	AG16
GPIO61[2]	AK11	SDA1	M5	SHARED_SERDES_2_REFRES	AE23
GPIO62[2]	AG9	SDA2	M4	SHARED_SERDES_2_RXN1	AK26
GPIO63[2]	AH9	SGMIICLKN	AF24	SHARED_SERDES_2_RXP1	AK25
HOUT	AH2	SGMIICLKP	AF23	SHARED_SERDES_2_TXN1	AG24
LENDIAN[B]	G26	SHARED_SERDES_0_REFCLKN	AF18	SHARED_SERDES_2_TXP1	AG25
$\overline{\text{LRESETNMIEN}}$	AF2	SHARED_SERDES_0_REFCLKP	AF17	SHARED_SERDES_3_REFRES	AE22
$\overline{\text{LRESET}}$	AK3	SHARED_SERDES_0_REFRES	AE17	SHARED_SERDES_3_RXN0	AJ22
MAINPLL_OD_SEL[B]	J30	SHARED_SERDES_0_RXN0	AJ19	SHARED_SERDES_3_RXN1	AK23
MDCLK	G2	SHARED_SERDES_0_RXN1	AK20	SHARED_SERDES_3_RXP0	AJ21
MDIO	H1	SHARED_SERDES_0_RXP0	AJ18	SHARED_SERDES_3_RXP1	AK22
$\overline{\text{NMI}}$	AJ2	SHARED_SERDES_0_RXP1	AK19	SHARED_SERDES_3_TXN0	AH20
PCIECLKN	AE19	SHARED_SERDES_0_TXN0	AH17	SHARED_SERDES_3_TXN1	AG21
PCIECLKP	AE20	SHARED_SERDES_0_TXN1	AG18	SHARED_SERDES_3_TXP0	AH21
PHYSYNC	AJ29	SHARED_SERDES_0_TXP0	AH18	SHARED_SERDES_3_TXP1	AG22
$\overline{\text{POR}}$	G4	SHARED_SERDES_0_TXP1	AG19	SPI0CLK	L29
RADSYNC	AH28	SHARED_SERDES_1_REFCLKN	AF15	SPI0DOUT	L27
$\overline{\text{RESETFULL}}$	AE2	SHARED_SERDES_1_REFCLKP	AF14	SPI0SCS1	K28
$\overline{\text{RESETSTAT}}$	AE4	SHARED_SERDES_1_REFRES	AE13	SPI0SCS2	K30
$\overline{\text{RESET}}$	AF3	SHARED_SERDES_1_RXN0	AJ16	SPI0SCS3	K29
RP1CLKN[2]	AG13	SHARED_SERDES_1_RXN1	AK17	SPI0SCS4	M29
RP1CLKP[2]	AG12	SHARED_SERDES_1_RXP0	AJ15	SPI1CLK	M26
RP1FBN[2]	AF11	SHARED_SERDES_1_RXP1	AK16	SPI1DIN	L26
RP1FBP[2]	AF10	SHARED_SERDES_1_TXN0	AH14	SPI1DOUT	N26
RSV001	C12	SHARED_SERDES_1_TXN1	AG15	SPI1SCS0	M28
RSV002	N23	SHARED_SERDES_1_TXP0	AH15		
RSV003	M23				
RSV004	AE29				
RSV005	AE28				
RSV006	AH29				
RSV007	AF5				
RSV008	AF4				
RSV009	Y4				
RSV010	W5				
RSV011	J25				
RSV012	H25				
RSV013	AF22				
RSV014	AE21				
RSV015	E13				
RSV016	F14				
RSV017	AF20				
RSV018	AE15				
RSV0A	AF6				
RSV0B	AF8				
SCL0	N2				
SCL1	L3				
SCL2	M3				

Table 6-4. Terminal Functions — By Signal Name (continued)

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
SPI1SCS1	L28	TIMO6[2]	J30	USBDM	R1
SPI1SCS2	M27	TIMO7[2]	K27	USBDP	P1
SPI2CLK[2]	L4	TMS	AH3	USBDRVVBUS	M6
SPI2DIN[2]	K4	$\overline{\text{TRST}}$	AK4	USBID0	N5
SPI2DOUT[2]	J4	TSCOMPOUT	AG1	USBRESREF	M8
SPI2SCS0[2]	K2	TSPUSHEVT0	AF1	USBRX0M	T2
SPI2SCS1[2]	G26	TSPUSHEVT1	AH1	USBRX0P	R2
SPI2SCS2[2]	F27	TSREFCLKN	AK14	USBTX0M	M1
SPI2SCS3[2]	F26	TSREFCLKP	AK13	USBTX0P	L1
SPI2SCS4[2]	G29	TSRXCLKOUT0N	AJ13	USBVBUS	N4
SYSCLKN	AG29	TSRXCLKOUT0P	AJ12	USIMCLK	F3
SYSCLKOUT	AF28	TSSYNCEVT	AG2	USIMIO	F2
SYSCLKP	AF29	UART0CTS	L4	USIMRST	F1
TCK	AJ3	UART0RTS	K2	VCL[2]	AH27
TDI	AJ4	UART0RXD	K5	VCNTL0	AJ27
TDO	AH4	UART0TXD	K3	VCNTL1	AK28
TIMI0	J2	UART1CTS	K4	VCNTL2	AJ28
TIMI1	J1	UART1RTS	J4	VCNTL3	AG27
TIMI2[2]	F28	UART1RXD	L5	VCNTL4	AH27
TIMI3[2]	G27	UART1TXD	J5	VCNTL5	AH26
TIMI4[2]	H30	UART2CTS[2]	AC29	VD[2]	AH26
TIMI5[2]	J26	UART2RTS[2]	R27	VDDCMON	K10
TIMI6[2]	H26	UART2RXD[2]	AB29	VDDUSB	U9
TIMI7[2]	H29	UART2TXD[2]	U27	VNWA1	L7
TIMO0	H3	UART3CTS[2]	P30	VNWA2	L23
TIMO1	J3	UART3RTS[2]	W29	VNWA3	AC21
TIMO2[2]	J27	UART3RXD[2]	W28	VNWA4	Y8
TIMO3[2]	H28	UART3TXD[2]	V27	VPH	T10
TIMO4[2]	G28	USBCLKM	R3	VP	V8
TIMO5[2]	H27	USBCLKP	P3	VPTX	V10

Table 6-4. Terminal Functions — By Signal Name (continued)

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
VSS	A1, A30, C5, C9, C13, C18, C22, C26, D3, D7, D11, D16, D20, D24, D28, F4, F16, G1, G3, G5, G6, G8, G10, G12, G14, G16, G18, G20, G22, G24, H2, H4, H5, H7, H9, H11, H13, H15	VSS	P21, P23, P25, R4, R5, R6, R8, R10, R12, R14, R16, R18, R20, , R22, R24, T1, T7, T9, T11, T13, T15, T17, T19, T21, T23, T25, U2, U6, U8, U10, U12, U14, U16, U18, U20, U22	VSS	AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AD3, AD7, AD9, AD11, AD13, AD15, AD17, AD19, AD21, AD23, AD25, AD29, AE6, AE8, AE10, AE12
VSS	H17, H19, H21, H23, J6, J8, J10, J12, J16, J18, J20, J22, J24, J28, K1, K6, K7, K11, K13, K15, K17, K19, K21, K23, K25, L2, L6, L8, L10, L12, L14, L16, L18, L20, L22, L24	VSS	U24, U29, V7, V9, V11, V13, V15, V17, V19, V21, V23, V25, W4, W6, W8, W10, W12, W14, W16, W18, W20, W22, W24, Y5, Y7, Y9, Y11, Y13, , Y15, Y17, Y19, Y21, Y23, Y25	VSS	AE14, AE16, AE18, AE24, AF7, AF13, AF16, AF19, AF21, AF25, AG4, AG11, AG14, AG17, AG20, AG23, AG26, AG28, AH10, AH13, AH16, AH19, AH22, AH25
VSS	M7, M9, M11, M13, M15, M17, M19, M21, M25, N1, N3, N6, N7, N8, N10, N12, N14, N16, N18, N20, N22, N24, N29, P2, P4, P5, P6, P7, P8, P9, P11, P13, P15, P17, P19	VSS	AA6, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AA24, AA29, AB1, AB7, AB9, AB11, AB13, AB15, AB17, AB19, AB21, AB23, AB25, AC6, AC8	VSS	AJ8, AJ11, AJ14, AJ17, AJ20, AJ23, AJ26, AK1, AK12, AK15, AK18, AK21, AK24, AK27, AK30
				VSSCMON	K9

Table 6-5. Terminal Functions — By Ball Number

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
A1	VSS	B20	DDR3ADQS8N	C4	DDR3AD14
A10	DDR3ADQS3P	B21	DDR3AD33	C5	VSS
A11	DDR3ACE1	B22	DDR3ADQS4N	C6	DDR3AD18
A12	DDR3AODT0	B23	DDR3AD47	C7	DVDDR
A13	DDR3ACAS	B24	DDR3ADQS5P	C8	DDR3AD23
A14	DDR3ACE0	B25	DDR3AD49	C9	VSS
A15	DDR3ACLKOUTN0	B26	DDR3ADQS6P	D1	DDR3AD04
A16	DDR3ACLKOUTP0	B27	DDR3AD55	D10	DDR3AD25
A17	DDR3AA05	B28	DDR3AD62	D11	VSS
A18	DDR3AA02	B29	DDR3ADQS7N	D12	DDR3AWE
A19	DDR3ADQM8	B3	DDR3ADQS0N	D13	DVDDR
A2	DVDDR	B30	DVDDR	D14	DDR3AA01
A20	DDR3ADQS8P	B4	DDR3ADQS1P	D15	DDR3AA06
A21	DDR3AD32	B5	DDR3AD11	D16	VSS
A22	DDR3ADQS4P	B6	DDR3AD19	D17	DDR3AA09
A23	DDR3ADQM5	B7	DDR3ADQS2N	D18	DVDDR
A24	DDR3ADQS5N	B8	DDR3ADQM3	D19	DDR3ACB05
A25	DDR3AD50	B9	DDR3AD30	D2	DDR3AD05
A26	DDR3ADQS6N	C1	DDR3AD03	D20	VSS
A27	DDR3AD51	C10	DDR3AD27	D21	DDR3AD35
A28	DDR3AD63	C11	DVDDR	D22	DVDDR
A29	DVDDR	C12	RSV001	D23	DDR3AD45
A3	DDR3ADQS0P	C13	VSS	D24	VSS
A30	VSS	C14	DDR3AA00	D25	DDR3AD41
A4	DDR3ADQS1N	C15	DDR3AA04	D26	DVDDR
A5	DDR3AD08	C16	DVDDR	D27	DDR3AD54
A6	DDR3ADQM2	C17	DDR3AA12	D28	VSS
A7	DDR3ADQS2P	C18	VSS	D29	DDR3AD61
A8	DDR3AD29	C19	DDR3ACB06	D3	VSS
A9	DDR3AD28	C2	DDR3AD01	D30	DDR3AD59
B1	DVDDR	C20	DVDDR	D4	DDR3AD13
B10	DDR3ADQS3N	C21	DDR3AD34	D5	DVDDR
B11	DDR3AA13	C22	VSS	D6	DDR3AD09
B12	DDR3AA10	C23	DDR3AD42	D7	VSS
B13	DDR3ARAS	C24	DVDDR	D8	DDR3AD22
B14	DDR3ACLKOUTN1	C25	DDR3AD48	D9	DVDDR
B15	DDR3ACLKOUTP1	C26	VSS	E1	DDR3AD00
B16	DDR3AA07	C27	DDR3ADQM6	E10	DDR3AD26
B17	DDR3AA08	C28	DVDDR	E11	DDR3ABA0
B18	DDR3AA11	C29	DDR3ADQS7P	E12	DDR3ABA2
B19	DDR3ACB07	C3	DVDDR	E13	RSV015
B2	DDR3AD02	C30	DDR3AD60	E14	DDR3AA03

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
E15	DDR3ARESET	F26	BOOTMODE01 [B]	G28	BOOTMODE11 [B]
E16	DDR3AA15	F26	GPIO02	G28	GPIO12
E17	DDR3ACKE0	F26	SPI2SCS3 [2]	G28	TIMO4 [2]
E18	DDR3AA14	F27	BOOTMODE00 [B]	G29	BOOTMODE02 [B]
E19	DDR3ACB04	F27	GPIO01	G29	GPIO03
E2	DDR3AD06	F27	SPI2SCS2 [2]	G29	SPI2SCS4 [2]
E20	DDR3ACB03	F28	BOOTMODE03 [B]	G3	VSS
E21	DDR3AD37	F28	GPIO04	G30	DDR3ACLKN
E22	DDR3AD36	F28	TIMI2[2]	G4	POR
E23	DDR3AD38	F29	DDR3AD58	G5	VSS
E24	DDR3AD44	F3	USIMCLK	G6	VSS
E25	DDR3AD43	F30	DDR3ACLKP	G7	DVDDR
E26	DDR3AD52	F4	VSS	G8	VSS
E27	DDR3AD53	F5	DDR3AD15	G9	DVDDR
E28	DDR3ADQM7	F6	DDR3AD12	H1	MDIO
E29	DDR3AD57	F7	DDR3AD20	H10	DVDDR
E3	DDR3AD07	F8	DDR3AD16	H11	VSS
E30	DDR3AD56	F9	DDR3ARZQ1	H12	DVDDR
E4	DDR3ADQM0	G1	VSS	H13	VSS
E5	DDR3ADQM1	G10	VSS	H14	DVDDR
E6	DDR3AD10	G11	DVDDR	H15	VSS
E7	DDR3AD17	G12	VSS	H16	DVDDR
E8	DDR3AD21	G13	DVDDR	H17	VSS
E9	DDR3AD31	G14	VSS	H18	DVDDR
F1	USIMRST	G15	DVDDR	H19	VSS
F10	DDR3AD24	G16	VSS	H2	VSS
F11	DDR3AODT1	G17	DVDDR	H20	DVDDR
F12	DDR3ABA1	G18	VSS	H21	VSS
F13	DDR3ARZQ0	G19	DVDDR	H22	DVDDR
F14	RSV016	G2	MDCLK	H23	VSS
F15	DDR3AVREFSSTL	G20	VSS	H24	DVDDR
F16	VSS	G21	DVDDR	H25	RSV012
F17	DDR3ACKE1	G22	VSS	H26	BOOTMODE07 [B]
F18	DDR3ACB02	G23	DVDDR	H26	GPIO08
F19	DDR3ACB00	G24	VSS	H26	TIMI6[2]
F2	USIMIO	G25	DVDDR	H27	BOOTMODE12 [B]
F20	DDR3ACB01	G26	GPIO00	H27	GPIO13
F21	DDR3ARZQ2	G26	LENDIAN [B]	H27	TIMO5 [2]
F22	DDR3ADQM4	G26	SPI2SCS1 [2]	H28	BOOTMODE10 [B]
F23	DDR3AD39	G27	BOOTMODE04 [B]	H28	GPIO11
F24	DDR3AD46	G27	GPIO05	H28	TIMO3 [2]
F25	DDR3AD40	G27	TIMI3[2]	H29	BOOTMODE08 [B]

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
H29	GPIO09	J3	TIMO1	K7	VSS
H29	TIMI7[2]	J30	GPIO14	K8	AVDDA6
H3	CSISC2_0_MUX[B]	J30	MAINPLL_OD_SEL[B]	K9	VSSCMON
H3	TIMO0	J30	TIMO6[2]	L1	USBTX0P
H30	BOOTMODE05[B]	J4	SPI2DIN[2]	L10	VSS
H30	GPIO06	J4	UART1RTS	L11	CVDD
H30	TIMI4[2]	J5	UART1TXD	L12	VSS
H4	VSS	J6	VSS	L13	CVDD
H5	VSS	J7	DVDDR	L14	VSS
H6	DVDDR	J8	VSS	L15	CVDD
H7	VSS	J9	DVDDR	L16	VSS
H8	DVDDR	K1	VSS	L17	CVDD
H9	VSS	K10	VDDCMON	L18	VSS
J1	AVSIFSEL1[B]	K11	VSS	L19	CVDD
J1	TIMI1	K12	CVDD	L2	VSS
J10	VSS	K13	VSS	L20	VSS
J11	AVDDA7	K14	CVDD	L21	CVDD
J12	VSS	K15	VSS	L22	VSS
J13	DVDDR	K16	CVDD	L23	VNWA2
J14	AVDDA8	K17	VSS	L24	VSS
J15	DVDDR	K18	CVDD	L25	DVDD18
J16	VSS	K19	VSS	L26	SPI1DIN
J17	AVDDA4	K2	SPI2SCS0[2]	L27	SPI0DOUT
J18	VSS	K2	UART0RTS	L28	SPI1SCS1
J19	DVDDR	K20	CVDD	L29	SPI0CLK
J2	AVSIFSEL0[B]	K21	VSS	L3	SCL1
J2	TIMI0	K22	CVDD	L30	SPI0SCS0
J20	VSS	K23	VSS	L4	SPI2CLK[2]
J21	AVDDA10	K24	DVDD18	L4	UART0CTS
J22	VSS	K25	VSS	L5	UART1RXD
J23	AVDDA1	K26	CSISC2_3_MUX[B]	L6	VSS
J24	VSS	K26	GPIO16	L7	VNWA1
J25	RSV011	K27	GPIO15	L8	VSS
J26	BOOTMODE06[B]	K27	TIMO7[2]	L9	CVDD
J26	GPIO07	K28	SPI0SCS1	M1	USBTX0M
J26	TIMI5[2]	K29	SPI0SCS3	M10	CVDD1
J27	BOOTMODE09[B]	K3	UART0TXD	M11	VSS
J27	GPIO10	K30	SPI0SCS2	M12	CVDD1
J27	TIMO2[2]	K4	SPI2DOUT[2]	M13	VSS
J28	VSS	K4	UART1CTS	M14	CVDD
J29	DVDD18	K5	UART0RXD	M15	VSS
J3	CSISC2_0_CLKCTL[B]	K6	VSS		

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
M16	CVDD1	N27	SPI0DIN	P7	VSS
M17	VSS	N28	DVDD18	P8	VSS
M18	CVDD	N29	VSS	P9	VSS
M19	VSS	N3	VSS	R1	USBDM
M2	SDA0	N30	$\overline{\text{EMIFCE0}}$	R10	VSS
M20	CVDD	N4	USBVBUS	R11	CVDD
M21	VSS	N5	USBID0	R12	VSS
M22	CVDD	N6	VSS	R13	CVDD
M23	RSV003	N7	VSS	R14	VSS
M24	DVDD18	N8	VSS	R15	CVDD
M25	VSS	N9	CVDD	R16	VSS
M26	SPI1CLK	P1	USBDP	R17	CVDD
M27	SPI1SCS2	P10	CVDD	R18	VSS
M28	SPI1SCS0	P11	VSS	R19	CVDD
M29	SPI0SCS4	P12	CVDD	R2	USBRX0P
M3	SCL2	P13	VSS	R20	VSS
M30	EMIFR $\overline{\text{W}}$	P14	CVDD	R21	CVDD
M4	SDA2	P15	VSS	R22	VSS
M5	SDA1	P16	CVDD1	R23	DVDD18
M6	USBDRVVBUS	P17	VSS	R24	VSS
M7	VSS	P18	CVDD	R25	DVDD18
M8	USBRESREF	P19	VSS	R26	EMIFA05
M9	VSS	P2	VSS	R26	GPIO37[2]
N1	VSS	P20	CVDD1	R27	EMIFA18
N10	VSS	P21	VSS	R27	UART2RTS[2]
N11	CVDD1	P22	CVDD	R28	EMIFA09
N12	VSS	P23	VSS	R28	GPIO41[2]
N13	CVDD	P24	DVDD18	R29	EMIFA01
N14	VSS	P25	VSS	R29	GPIO33[2]
N15	CVDD1	P26	EMIFA04	R3	USBCLKM
N16	VSS	P26	GPIO36[2]	R30	EMIFA02
N17	CVDD	P27	EMIFA10	R30	GPIO34[2]
N18	VSS	P27	GPIO42[2]	R4	VSS
N19	CVDD1	P28	EMIFA03	R5	VSS
N2	SCL0	P28	GPIO35[2]	R6	VSS
N20	VSS	P29	$\overline{\text{EMIFBE1}}$	R7	DVDD18
N21	CVDD1	P3	USBCLKP	R8	VSS
N22	VSS	P30	$\overline{\text{EMIFCE3}}$	R9	DVDD33
N23	RSV002	P30	UART3CTS[2]	T1	VSS
N24	VSS	P4	VSS	T10	VPH
N25	DVDD18	P5	VSS	T11	VSS
N26	SPI1DOUT	P6	VSS	T12	CVDD

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
T13	VSS	U17	CVDD	V22	CVDD
T14	CVDD	U18	VSS	V23	VSS
T15	VSS	U19	CVDD	V24	DVDD18
T16	CVDD	U2	VSS	V25	VSS
T17	VSS	U20	VSS	V26	EMIFD07
T18	CVDD	U21	CVDD	V27	EMIFA23
T19	VSS	U22	VSS	V27	UART3TXD [2]
T2	USBRX0M	U23	DVDD18	V28	EMIFA12
T20	CVDD	U24	VSS	V29	EMIFA06
T21	VSS	U25	AVDDA2	V29	GPIO38 [2]
T22	CVDD	U26	EMIFD00	V3	EMU21[2]
T23	VSS	U27	EMIFA21	V3	GPIO19
T24	DVDD18	U27	UART2TXD [2]	V30	$\overline{\text{EMIFWE}}$
T25	VSS	U28	DVDD18	V4	EMU27 [2]
T26	EMIFA16	U29	VSS	V4	GPIO25
T26	GPIO46[2]	U3	EMU30 [2]	V5	EMU26[2]
T27	EMIFA17	U3	GPIO28	V5	GPIO24
T27	GPIO47[2]	U30	$\overline{\text{EMIFCE1}}$	V6	DVDD18
T28	EMIFA14	U4	EMU25[2]	V7	VSS
T28	GPIO44[2]	U4	GPIO23	V8	VP
T29	EMIFA08	U5	EMU32[2]	V9	VSS
T29	GPIO40[2]	U5	GPIO30	W1	EMU22 [2]
T3	EMU29[2]	U6	VSS	W1	GPIO20
T3	GPIO27	U7	DVDD18	W10	VSS
T30	$\overline{\text{EMIFOE}}$	U8	VSS	W11	CVDD
T4	EMU33[2]	U9	VDDUSB	W12	VSS
T4	GPIO31	V1	EMU23[2]	W13	CVDD
T5	EMU31[2]	V1	GPIO21	W14	VSS
T5	GPIO29	V10	VPTX	W15	CVDD1
T6	DVDD18	V11	VSS	W16	VSS
T7	VSS	V12	CVDD	W17	CVDD
T8	DVDD18	V13	VSS	W18	VSS
T9	VSS	V14	CVDD	W19	CVDD
U1	EMU28[2]	V15	VSS	W2	EMU19 [2]
U1	GPIO26	V16	CVDD	W2	GPIO17
U10	VSS	V17	VSS	W20	VSS
U11	CVDD	V18	CVDD	W21	CVDD
U12	VSS	V19	VSS	W22	VSS
U13	CVDD	V2	EMU24[2]	W23	DVDD18
U14	VSS	V2	GPIO22	W24	VSS
U15	CVDD	V20	CVDD	W25	DVDD18
U16	VSS	V21	VSS	W26	EMIFD10

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
W27	EMIFD02	Y5	VSS	AB15	VSS
W28	EMIFA22	Y6	DVDD18	AB16	CVDDS
W28	UART3RXD [2]	Y7	VSS	AB17	VSS
W29	$\overline{\text{EMIFCE2}}$	Y8	VNWA4	AB18	CVDDS
W29	UART3RTS [2]	Y9	VSS	AB19	VSS
W3	DVDD18	AA1	EMU12	AB2	DVDD18
W30	EMIFA00	AA10	VSS	AB20	CVDD
W30	GPIO32[2]	AA11	CVDD	AB21	VSS
W4	VSS	AA12	VSS	AB22	CVDD
W5	RSV010	AA13	CVDD	AB23	VSS
W6	VSS	AA14	VSS	AB24	DVDD18
W7	DVDD18	AA15	CVDD	AB25	VSS
W8	VSS	AA16	VSS	AB26	EMIFD15
W9	CVDD	AA17	CVDD	AB27	EMIFD05
Y1	EMU20[2]	AA18	VSS	AB28	EMIFD01
Y1	GPIO18	AA19	CVDD	AB29	EMIFA20
Y10	CVDD	AA2	EMU14	AB29	UART2RXD [2]
Y11	VSS	AA20	VSS	AB3	EMU11
Y12	CVDD	AA21	CVDD	AB30	$\overline{\text{EMIFBE0}}$
Y13	VSS	AA22	VSS	AB4	EMU09
Y14	CVDD1	AA23	DVDD18	AB5	EMU10
Y15	VSS	AA24	VSS	AB6	DVDD18
Y16	CVDD1	AA25	DVDD18	AB7	VSS
Y17	VSS	AA26	EMIFD13	AB8	DVDD18
Y18	CVDD	AA27	EMIFD06	AB9	VSS
Y19	VSS	AA28	DVDD18	AC1	EMU08
Y2	EMU18	AA29	VSS	AC10	VSS
Y20	CVDD	AA3	EMU15	AC11	DVDD18
Y21	VSS	AA30	EMIFA07	AC12	VSS
Y22	CVDD	AA30	GPIO39[2]	AC13	CVDDS
Y23	VSS	AA4	EMU13	AC14	VSS
Y24	DVDD18	AA5	EMU16	AC15	CVDDS
Y25	VSS	AA6	VSS	AC16	VSS
Y26	EMIFD14	AA7	DVDDR	AC17	CVDDS
Y27	EMIFD04	AA8	VSS	AC18	VSS
Y28	EMIFA15	AA9	CVDD	AC19	CVDDS
Y28	GPIO45[2]	AB1	VSS	AC2	EMU07
Y29	EMIFA13	AB10	CVDD	AC20	VSS
Y29	GPIO43[2]	AB11	VSS	AC21	VNWA3
Y3	EMU17	AB12	CVDD	AC22	VSS
Y30	EMIFWAIT0	AB13	VSS	AC23	DVDD18
Y4	RSV009	AB14	CVDDS	AC24	VSS

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
AC25	DVDD18	AD6	DVDD18	AF12	AVDDA5
AC26	EMIFD12	AD7	VSS	AF13	VSS
AC27	EMIFD09	AD8	DVDD18	AF14	SHARED_SERDES_1_REFCLKP
AC28	EMIFD03	AD9	VSS	AF15	SHARED_SERDES_1_REFCLKN
AC29	EMIFA19	AE1	EMU04	AF16	VSS
AC29	UART2CTS [2]	AE10	VSS	AF17	SHARED_SERDES_0_REFCLKP
AC3	EMU00	AE11	DVDD18	AF18	SHARED_SERDES_0_REFCLKN
AC30	EMIFWAIT1	AE12	VSS	AF19	VSS
AC4	EMU05	AE13	SHARED_SERDES_1_REFRES	AF2	$\overline{\text{LRESETNMIEN}}$
AC5	EMU06	AE14	VSS	AF20	RSV017
AC6	VSS	AE15	RSV018	AF21	VSS
AC7	DVDD18	AE16	VSS	AF22	RSV013
AC8	VSS	AE17	SHARED_SERDES_0_REFRES	AF23	SGMIICLKP
AC9	DVDD18	AE18	VSS	AF24	SGMIICLKN
AD1	EMU02	AE19	PCIECLKN	AF25	VSS
AD10	DVDD18	AE2	$\overline{\text{RESETFULL}}$	AF26	AVDDA3
AD11	VSS	AE20	PCIECLKP	AF27	CORECLKSEL1
AD12	AVDDAS	AE21	RSV014	AF28	SYSCLKOUT
AD13	VSS	AE22	SHARED_SERDES_3_REFRES	AF29	SYSCLKP
AD14	AVDDAS	AE23	SHARED_SERDES_2_REFRES	AF3	$\overline{\text{RESET}}$
AD15	VSS	AE24	VSS	AF30	DFESYSREFN
AD16	AVDDAS	AE25	DVDD18	AF4	RSV008
AD17	VSS	AE26	AVDDA4	AF5	RSV007
AD18	AVDDAS	AE27	CORECLKSEL0	AF6	RSV0A
AD19	VSS	AE28	RSV005	AF7	VSS
AD2	EMU03	AE29	RSV004	AF8	RSV0B
AD20	AVDDAS	AE3	EXTFRAMEEVENT	AF9	DFEIO14
AD21	VSS	AE30	DFESYSREFP	AF9	GPIO60 [2]
AD22	AVDDAS	AE4	$\overline{\text{RESETSTAT}}$	AG1	TSCOMPOUT
AD23	VSS	AE5	BOOTMODE13 [B]	AG10	DFEIO10
AD24	DVDD18	AE5	CORESEL0	AG10	GPIO56[2]
AD25	VSS	AE6	VSS	AG11	VSS
AD26	EMIFD11	AE7	DVDD18	AG12	DFESYNCINP0
AD27	EMIFD08	AE8	VSS	AG12	RP1CLKP [2]
AD28	DVDD18	AE9	DVDD18	AG13	DFESYNCINN0
AD29	VSS	AF1	TSPUSHEVT0	AG13	RP1CLKN[2]
AD3	VSS	AF10	DFESYNCINP1	AG14	VSS
AD30	EMIFA11	AF10	RP1FBP[2]	AG15	SHARED_SERDES_1_TXN1
AD4	DVDD18	AF11	DFESYNCINN1	AG16	SHARED_SERDES_1_TXP1
AD5	EMU01	AF11	RP1FBN[2]		

Table 6-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
AG17	VSS	AH25	VSS	AJ5	DFEIO1
AG18	SHARED_SERDES_0_TXN1	AH26	VCNTL5	AJ6	DFEIO11
AG19	SHARED_SERDES_0_TXP1	AH26	VD [2]	AJ6	GPIO57[2]
AG2	TSSYNCEVT	AH27	VCL[2]	AJ7	DFEIO6
AG20	VSS	AH27	VCNTL4	AJ7	GPIO52[2]
AG21	SHARED_SERDES_3_TXN1	AH28	RADSYNC	AJ8	VSS
AG22	SHARED_SERDES_3_TXP1	AH29	RSV006	AJ9	DFESYNCOUTP0
AG23	VSS	AH3	TMS	AK1	VSS
AG24	SHARED_SERDES_2_TXN1	AH30	ALTCORECLKN	AK10	DFEIO12
AG25	SHARED_SERDES_2_TXP1	AH4	TDO	AK10	GPIO58[2]
AG26	VSS	AH5	DFEIO0	AK11	DFEIO15
AG27	VCNTL3	AH6	BOOTMODE15 [B]	AK11	GPIO61[2]
AG28	VSS	AH6	CORESEL2	AK12	VSS
AG29	SYSCLKN	AH7	DFEIO4	AK13	TSREFCLKP
AG3	BOOTCOMPLETE	AH7	GPIO50[2]	AK14	TSREFCLKN
AG30	ALTCORECLKP	AH8	DVDD18	AK15	VSS
AG4	VSS	AH9	DFEIO17	AK16	SHARED_SERDES_1_RXP1
AG5	DVDD18	AH9	GPIO63 [2]	AK17	SHARED_SERDES_1_RXN1
AG6	BOOTMODE14 [B]	AJ1	DVDD18	AK18	VSS
AG6	CORESEL1	AJ10	DFESYNCOUTN0	AK19	SHARED_SERDES_0_RXP1
AG7	DFEIO2	AJ11	VSS	AK2	DVDD18
AG7	GPIO48[2]	AJ12	TSRXCLKOUT0P	AK20	SHARED_SERDES_0_RXN1
AG8	DFEIO7	AJ13	TSRXCLKOUT0N	AK21	VSS
AG8	GPIO53[2]	AJ14	VSS	AK22	SHARED_SERDES_3_RXP1
AG9	DFEIO16	AJ15	SHARED_SERDES_1_RXP0	AK23	SHARED_SERDES_3_RXN1
AG9	GPIO62[2]	AJ16	SHARED_SERDES_1_RXN0	AK24	VSS
AH1	TSPUSHEVT1	AJ17	VSS	AK25	SHARED_SERDES_2_RXP1
AH10	VSS	AJ18	SHARED_SERDES_0_RXP0	AK26	SHARED_SERDES_2_RXN1
AH11	DFESYNCOUTP1	AJ19	SHARED_SERDES_0_RXN0	AK27	VSS
AH12	DFESYNCOUTN1	AJ2	$\overline{\text{NMI}}$	AK28	VCNTL1
AH13	VSS	AJ20	VSS	AK29	DVDD18
AH14	SHARED_SERDES_1_TXN0	AJ21	SHARED_SERDES_3_RXP0	AK3	$\overline{\text{LRESET}}$
AH15	SHARED_SERDES_1_TXP0	AJ22	SHARED_SERDES_3_RXN0	AK30	VSS
AH16	VSS	AJ23	VSS	AK4	$\overline{\text{TRST}}$
AH17	SHARED_SERDES_0_TXN0	AJ24	SHARED_SERDES_2_RXP0	AK5	DFEIO3
AH18	SHARED_SERDES_0_TXP0	AJ25	SHARED_SERDES_2_RXN0	AK5	GPIO49[2]
AH19	VSS	AJ26	VSS	AK6	DFEIO5
AH2	HOUT	AJ27	VCNTL0	AK6	GPIO51[2]
AH20	SHARED_SERDES_3_TXN0	AJ28	VCNTL2	AK7	DFEIO9
AH21	SHARED_SERDES_3_TXP0	AJ29	PHYSYNC	AK7	GPIO55[2]
AH22	VSS	AJ3	TCK	AK8	DFEIO8
AH23	SHARED_SERDES_2_TXN0	AJ30	DVDD18	AK8	GPIO54[2]
AH24	SHARED_SERDES_2_TXP0	AJ4	TDI	AK9	DFEIO13
				AK9	GPIO59[2]

6.4 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 9-25), 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 66AK2L06 device, see Section 10.3. To determine which pins on the device include internal pullup/pulldown resistors, see Table 6-2.

7 Memory, Interrupts, and EDMA for 66AK2L06

7.1 Memory Map Summary for 66AK2L06

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

Table 7-1. Device Memory Map Summary for 66AK2L06

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0000 0000	00 0003 FFFF	256K	ARM ROM	Reserved	ARM ROM
00 0004 0000	00 007F FFFF	8M-256K	Reserved	Reserved	Reserved
00 0080 0000	00 008F FFFF	1M	Reserved	L2 SRAM	L2 SRAM
00 0090 0000	00 00DF FFFF	5M	Reserved	Reserved	Reserved
00 00E0 0000	00 00E0 7FFF	32K	Reserved	L1P SRAM	L1P SRAM
00 00E0 8000	00 00EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 00F0 0000	00 00F0 7FFF	32K	Reserved	L1D SRAM	L1D SRAM
00 00F0 8000	00 00FF FFFF	1M-32K	Reserved	Reserved	Reserved
00 0100 0000	00 0100 FFFF	64K	ARM AXI2VBUSM Master Registers	C66x CorePac registers	C66x CorePac registers
00 0101 0000	00 010F FFFF	1M-64K	Reserved	C66x CorePac registers	C66x CorePac registers
00 0110 0000	00 0110 FFFF	64K	ARM STM Stimulus Ports	C66x CorePac registers	C66x CorePac registers
00 0111 0000	00 01BF FFFF	11M-64K	Reserved	C66x CorePac registers	C66x CorePac registers
00 01C0 0000	00 01CF FFFF	1M	Reserved	Reserved	Reserved
00 01D0 0000	00 01D0 007F	128	Tracer CFG0	Tracer CFG0	Tracer CFG0
00 01D0 0080	00 01D0 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D0 8000	00 01D0 807F	128	Tracer CFG1	Tracer CFG1	Tracer CFG1
00 01D0 8080	00 01D0 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D1 0000	00 01D1 007F	128	Tracer CFG2	Tracer CFG2	Tracer CFG2
00 01D1 0080	00 01D1 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D1 8000	00 01D1 807F	128	Tracer CFG3	Tracer CFG3	Tracer CFG3
00 01D1 8080	00 01D1 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D2 0000	00 01D2 007F	128	Tracer CFG23	Tracer CFG23	Tracer CFG23
00 01D2 0080	00 01D2 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D2 8000	00 01D2 807F	128	Tracer CFG8	Tracer CFG8	Tracer CFG8
00 01D2 8080	00 01D2 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D3 0000	00 01D3 007F	128	Tracer CFG20	Tracer CFG20	Tracer CFG20
00 01D3 0080	00 01D3 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D3 8000	00 01D3 807F	128	Tracer CFG21	Tracer CFG21	Tracer CFG21
00 01D3 8080	00 01D3 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D4 0000	00 01D4 007F	128	Tracer CFG25	Tracer CFG25	Tracer CFG25
00 01D4 0080	00 01D4 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D4 8000	00 01D4 807F	128	Tracer CFG09	Tracer CFG09	Tracer CFG09
00 01D4 8080	00 01D4 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D5 0000	00 01D5 007F	128	Tracer CFG10	Tracer CFG10	Tracer CFG10
00 01D5 0080	00 01D5 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D5 8000	00 01D5 807F	128	Tracer CFG11	Tracer CFG11	Tracer CFG11
00 01D5 8080	00 01D5 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D6 0000	00 01D6 007F	128	Tracer CFG12	Tracer CFG12	Tracer CFG12
00 01D6 0080	00 01D6 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D6 8000	00 01D6 807F	128	Reserved	Reserved	Reserved
00 01D6 8080	00 01D6 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D7 0000	00 01D7 007F	128	Reserved	Reserved	Reserved
00 01D7 0080	00 01D7 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D7 8000	00 01D7 807F	128	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 01D7 8080	00 01D7 FFFF	32K-128	Reserved	Reserved	Reserved
00 01D8 0000	00 01D8 007F	128	Reserved	Reserved	Reserved
00 01D8 0080	00 01D8 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D8 8000	00 01D8 807F	128	Tracer CFG26	Tracer CFG26	Tracer CFG26
00 01D8 8080	00 01D8 8FFF	32K-128	Reserved	Reserved	Reserved
00 01D9 0000	00 01D9 007F	128	Tracer CFG27	Tracer CFG27	Tracer CFG28
00 01D9 0080	00 01D9 7FFF	32K-128	Reserved	Reserved	Reserved
00 01D9 8000	00 01D9 807F	128	Tracer CFG28	Tracer CFG28	Tracer CFG28
00 01D9 8080	00 01D9 FFFF	32K-128	Reserved	Reserved	Reserved
00 01DA 0000	00 01DA 007F	128	Tracer CFG22	Tracer CFG22	Tracer CFG22
00 01DA 0080	00 01DA 7FFF	32K-128	Reserved	Reserved	Reserved
00 01DA 8000	00 01DA 807F	128	Reserved	Reserved	Reserved
00 01DA 8080	00 01DA FFFF	32K-128	Reserved	Reserved	Reserved
00 01DB 0000	00 01DB 007F	128	Tracer CFG31	Tracer CFG31	Tracer CFG31
00 01DB 0080	00 01DB 7FFF	32K-128	Reserved	Reserved	Reserved
00 01DB 8000	00 01DB 807F	128	Reserved	Reserved	Reserved
00 01DB 8080	00 01DB 8FFF	32K-128	Reserved	Reserved	Reserved
00 01DC 0000	00 01DC 007F	128	Tracer CFG17	Tracer CFG17	Tracer CFG17
00 01DC 0080	00 01DC 7FFF	32K-128	Reserved	Reserved	Reserved
00 01DC 8000	00 01DC 807F	128	Tracer CFG18	Tracer CFG18	Tracer CFG18
00 01DC 8080	00 01DC FFFF	32K-128	Reserved	Reserved	Reserved
00 01DD 0000	00 01DD 007F	128	Tracer CFG19	Tracer CFG19	Tracer CFG19
00 01DD 0080	00 01DD 7FFF	32K-128	Reserved	Reserved	Reserved
00 01DD 8000	00 01DD 807F	128	Tracer CFG4	Tracer CFG4	Tracer CFG4
00 01DD 8080	00 01DD FFFF	32K-128	Reserved	Reserved	Reserved
00 01DE 0000	00 01DE 007F	128	Tracer CFG5	Tracer CFG5	Tracer CFG5
00 01DE 0080	00 01DE 03FF	1K-128	Reserved	Reserved	Reserved
00 01DE 0400	00 01DE 047F	128	Tracer CFG6	Tracer CFG6	Tracer CFG6
00 01DD 0480	00 01DD 07FF	1K-128	Reserved	Reserved	Reserved
00 01DE 0800	00 01DE 087F	128	Tracer CFG7	Tracer CFG7	Tracer CFG7
00 01DE 0880	00 01DE 7FFF	30K-128	Reserved	Reserved	Reserved
00 01DE 8000	00 01DE 807F	128	Tracer CFG24	Tracer CFG24	Tracer CFG24
00 01DE 8080	00 01DF FFFF	64K-128	Reserved	Reserved	Reserved
00 01E0 0000	00 01E3 FFFF	256K	Reserved	Reserved	Reserved
00 01E4 0000	00 01E4 3FFF	16K	Reserved	Reserved	Reserved
00 01E4 4000	00 01E7 FFFF	240k	Reserved	Reserved	Reserved
00 01E8 0000	00 01E8 3FFF	16K	ARM CorePac VBUSP Memory Mapped Registers	ARM CorePac VBUSP Memory Mapped Registers	ARM CorePac VBUSP Memory Mapped Registers
00 01E8 4000	00 01EB FFFF	240k	Reserved	Reserved	Reserved
00 01EC 0000	00 01EF FFFF	256K	Reserved	Reserved	Reserved
00 01F0 0000	00 01F7 FFFF	512K	Reserved	Reserved	Reserved
00 01F8 0000	00 01F8 FFFF	64K	Reserved	Reserved	Reserved
00 01F9 0000	00 01F9 FFFF	64K	Reserved	Reserved	Reserved
00 01FA 0000	00 01FB FFFF	128K	Reserved	Reserved	Reserved
00 01FC 0000	00 01FD FFFF	128K	Reserved	Reserved	Reserved
00 01FE 0000	00 01FF FFFF	128K	Reserved	Reserved	Reserved
00 0200 0000	00 020F FFFF	1M	Reserved	Reserved	Reserved
00 0210 0000	00 0210 FFFF	64K	Reserved	Reserved	Reserved
00 0211 0000	00 0211 FFFF	64K	Reserved	Reserved	Reserved
00 0212 0000	00 0213 FFFF	128K	Reserved	Reserved	Reserved
00 0214 0000	00 0215 FFFF	128K	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0216 0000	00 0217 FFFF	128K	Reserved	Reserved	Reserved
00 0218 0000	00 0218 7FFF	32k	Reserved	Reserved	Reserved
00 0218 8000	00 0218 FFFF	32k	Reserved	Reserved	Reserved
00 0219 0000	00 0219 FFFF	64k	Reserved	Reserved	Reserved
00 021A 0000	00 021A FFFF	64K	Reserved	Reserved	Reserved
00 021B 0000	00 021B FFFF	64K	Reserved	Reserved	Reserved
00 021C 0000	00 021C 03FF	1K	Reserved	Reserved	Reserved
00 021C 0400	00 021C 3FFF	15K	Reserved	Reserved	Reserved
00 021C 4000	00 021C 43FF	1K	Reserved	Reserved	Reserved
00 021C 4400	00 021C 5FFF	7K	Reserved	Reserved	Reserved
00 021C 6000	00 021C 63FF	1K	Reserved	Reserved	Reserved
00 021C 6400	00 021C 7FFF	7K	Reserved	Reserved	Reserved
00 021C 8000	00 021C 83FF	1K	Reserved	Reserved	Reserved
00 021C 8400	00 021C FFFF	31K	Reserved	Reserved	Reserved
00 021D 0000	00 021D 00FF	256	Reserved	Reserved	Reserved
00 021D 0100	00 021D 3FFF	16K	Reserved	Reserved	Reserved
00 021D 4000	00 021D 40FF	256	Reserved	Reserved	Reserved
00 021D 4100	00 021D 7FFF	16K	Reserved	Reserved	Reserved
00 021D 8000	00 021D 80FF	256	Reserved	Reserved	Reserved
00 021D 8100	00 021D BFFF	16K	Reserved	Reserved	Reserved
00 021D C000	00 021D C0FF	256	Reserved	Reserved	Reserved
00 021D C100	00 021D EFFF	12K-256	Reserved	Reserved	Reserved
00 021D F000	00 021D F07F	128	Reserved	Reserved	Reserved
00 021D F080	00 021D FFFF	4K-128	Reserved	Reserved	Reserved
00 021E 0000	00 021E FFFF	64K	Reserved	Reserved	Reserved
00 021F 0000	00 021F 07FF	2K	FFTC_0 configuration	FFTC_0 configuration	FFTC_0 configuration
00 021F 0800	00 021F 0FFF	2K	Reserved	Reserved	Reserved
00 021F 1000	00 021F 17FF	2K	Reserved	Reserved	Reserved
00 021F 1800	00 021F 3FFF	10K	Reserved	Reserved	Reserved
00 021F 4000	00 021F 47FF	2K	FFTC_1 configuration	FFTC_1 configuration	FFTC_1 configuration
00 021F 4800	00 021F 7FFF	14K	Reserved	Reserved	Reserved
00 021F 8000	00 021F 87FF	Reserved	Reserved	Reserved	Reserved
00 021F 8800	00 021F BFFF	Reserved	Reserved	Reserved	Reserved
00 021F C000	00 021F C7FF	Reserved	Reserved	Reserved	Reserved
00 021F C800	00 021F FFFF	14K	Reserved	Reserved	Reserved
00 0220 0000	00 0220 007F	128	Timer0	Timer0	Timer0
00 0220 0080	00 0220 FFFF	64K-128	Reserved	Reserved	Reserved
00 0221 0000	00 0221 007F	128	Timer1	Timer1	Timer1
00 0221 0080	00 0221 FFFF	64K-128	Reserved	Reserved	Reserved
00 0222 0000	00 0222 007F	128	Timer2	Timer2	Timer2
00 0222 0080	00 0222 FFFF	64K-128	Reserved	Reserved	Reserved
00 0223 0000	00 0223 007F	128	Timer3	Timer3	Timer3
00 0223 0080	00 0223 FFFF	64K-128	Reserved	Reserved	Reserved
00 0224 0000	00 0224 007F	128	Reserved	Reserved	Reserved
00 0224 0080	00 0224 FFFF	64K-128	Reserved	Reserved	Reserved
00 0225 0000	00 0225 007F	128	Reserved	Reserved	Reserved
00 0225 0080	00 0225 FFFF	64K-128	Reserved	Reserved	Reserved
00 0226 0000	00 0226 007F	128	Reserved	Reserved	Reserved
00 0226 0080	00 0226 FFFF	64K-128	Reserved	Reserved	Reserved
00 0227 0000	00 0227 007F	128	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0227 0080	00 0227 FFFF	64K-128	Reserved	Reserved	Reserved
00 0228 0000	00 0228 007F	128	Timer 8	Timer 8	Timer 8
00 0228 0080	00 0228 FFFF	64K-128	Reserved	Reserved	Reserved
00 0229 0000	00 0229 007F	128	Timer 9	Timer 9	Timer 9
00 0229 0080	00 0229 FFFF	64K-128	Reserved	Reserved	Reserved
00 022A 0000	00 022A 007F	128	Timer 10	Timer 10	Timer 10
00 022A 0080	00 022A FFFF	64K-128	Reserved	Reserved	Reserved
00 022B 0000	00 022B 007F	128	Timer 11	Timer 11	Timer 11
00 022B 0080	00 022B FFFF	64K-128	Reserved	Reserved	Reserved
00 022C 0000	00 022C 007F	128	Timer 12	Timer 12	Timer 12
00 022C 0080	00 022C FFFF	64K-128	Reserved	Reserved	Reserved
00 022D 0000	00 022D 007F	128	Timer 13	Timer 13	Timer 13
00 022D 0080	00 022D FFFF	64K-128	Reserved	Reserved	Reserved
00 022E 0000	00 022E 007F	128	Timer 14	Timer 14	Timer 14
00 022E 0080	00 022E FFFF	64K-128	Reserved	Reserved	Reserved
00 022F 0000	00 022F 007F	128	Timer 15	Timer 15	Timer 15
00 022F 0080	00 022F 00FF	128	Timer 16	Timer 16	Timer 16
00 022F 0100	00 022F 017F	128	Timer 17	Timer 17	Timer 17
00 022F 0180	00 022F 01FF	128	Reserved	Reserved	Reserved
00 022F 0200	00 022F 027F	128	Reserved	Reserved	Reserved
00 0230 0000	00 0230 FFFF	64K	Reserved	Reserved	Reserved
00 0231 0000	00 0231 01FF	512	PLL Controller	PLL Controller	PLL Controller
00 0231 0200	00 0231 9FFF	40K-512	Reserved	Reserved	Reserved
00 0231 A000	00 0231 BFFF	8K	Reserved	Reserved	Reserved
00 0231 C000	00 0231 DFFF	8K	Reserved	Reserved	Reserved
00 0231 E000	00 0231 FFFF	8K	Reserved	Reserved	Reserved
00 0232 0000	00 0232 1FFF	8K	CSISC2 SerDes Config 3	CSISC2 SerDes Config 3	CSISC2 SerDes Config 3
00 0232 2000	00 0232 3FFF	8K	Reserved	Reserved	Reserved
00 0232 4000	00 0232 5FFF	8K	CSISC2 SerDes Config 0	CSISC2 SerDes Config 0	CSISC2 SerDes Config 0
00 0232 6000	00 0232 7FFF	4K	CSISC2 SerDes Config 1	CSISC2 SerDes Config 1	CSISC2 SerDes Config 1
00 0232 8000	00 0232 8FFF	8K	Reserved	Reserved	Reserved
00 0232 9000	00 0232 9FFF	4K	DDRA PHY Config	DDRA PHY Config	DDRA PHY Config
00 0232 A000	00 0232 BFFF	8K	CSISC2 SerDes Config 2	CSISC2 SerDes Config 2	CSISC2 SerDes Config 2
00 0232 C000	00 0232 CFFF	4K	Reserved	Reserved	Reserved
00 0232 D000	00 0232 DFFF	4K	Reserved	Reserved	Reserved
00 0232 E000	00 0232 EFFF	4K	Reserved	Reserved	Reserved
00 0232 F000	00 0232 FFFF	4K	Reserved	Reserved	Reserved
00 0233 0000	00 0233 03FF	1K	SmartReflex0	SmartReflex0	SmartReflex0
00 0233 0400	00 0233 07FF	1K	Reserved	Reserved	Reserved
00 0233 0400	00 0233 FFFF	62K	Reserved	Reserved	Reserved
00 0234 0000	00 0234 03FF	1K	Memory protection unit (MPU) 15	Memory protection unit (MPU) 15	Memory protection unit (MPU) 15
00 0234 0400	00 0234 07FF	1K	Reserved	Reserved	Reserved
00 0234 0800	00 0234 087F	128	Tracer CFG30	Tracer CFG30	Tracer CFG30
00 0234 0880	00 0234 0BFF	1K-128	Reserved	Reserved	Reserved
00 0234 0C00	00 0234 3FFF	13K	Reserved	Reserved	Reserved
00 0234 4000	00 0234 7FFF	16K	Reserved	Reserved	Reserved
00 0234 8000	00 0234 80FF	256	GPIO1 configuration	GPIO1 configuration	GPIO1 configuration
00 0234 8100	00 0234 83FF	768	Reserved	Reserved	Reserved
00 0234 8400	00 0234 843F	64	UART2 configuration	UART2 configuration	UART2 configuration
00 02348440	00 0234 87FF	1K-64	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0234 8800	00 0234 883F	64	UART3 configuration	UART3 configuration	UART3 configuration
00 0234 8840	00 0234 8BFF	1K-64	Reserved	Reserved	Reserved
00 0234 8C00	00 0234 8FFF	1K	OSR configuration	OSR configuration	OSR configuration
00 0234 9000	00 0234 FFFF	28K	Reserved	Reserved	Reserved
00 0235 0000	00 0235 0FFF	4K	Power sleep controller (PSC)	Power sleep controller (PSC)	Power sleep controller (PSC)
00 0235 1000	00 0235 FFFF	64K-4K	Reserved	Reserved	Reserved
00 0236 0000	00 0236 03FF	1K	Memory protection unit (MPU) 0	Memory protection unit (MPU) 0	Memory protection unit (MPU) 0
00 0236 0400	00 0236 7FFF	31K	Reserved	Reserved	Reserved
00 0236 8000	00 0236 83FF	1K	Memory protection unit (MPU) 1	Memory protection unit (MPU) 1	Memory protection unit (MPU) 1
00 0236 8400	00 0236 FFFF	31K	Reserved	Reserved	Reserved
00 0237 0000	00 0237 03FF	1K	Memory protection unit (MPU) 2	Memory protection unit (MPU) 2	Memory protection unit (MPU) 2
00 0237 0400	00 0237 7FFF	31K	Reserved	Reserved	Reserved
00 0237 8000	00 0237 83FF	1K	Memory protection unit (MPU) 3	Memory protection unit (MPU) 3	Memory protection unit (MPU) 3
00 0237 8400	00 0237 FFFF	31K	Reserved	Reserved	Reserved
00 0238 0000	00 0238 03FF	1K	Memory protection unit (MPU) 4	Memory protection unit (MPU) 4	Memory protection unit (MPU) 4
00 0238 8000	00 0238 83FF	1K	Memory protection unit (MPU) 5	Memory protection unit (MPU) 5	Memory protection unit (MPU) 5
00 0238 8400	00 0238 87FF	1K	Memory protection unit (MPU) 6	Memory protection unit (MPU) 6	Memory protection unit (MPU) 6
00 0238 8800	00 0238 8BFF	1K	Memory protection unit (MPU) 7	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	Memory protection unit (MPU) 8
00 0238 9000	00 0238 93FF	1K	Memory protection unit (MPU) 9	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	Memory protection unit (MPU) 10
00 0238 9800	00 0238 9BFF	1K	Memory protection unit (MPU) 11	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	Memory protection unit (MPU) 12
00 0238 A000	00 0238 A3FF	1K	Memory protection unit (MPU) 13	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	Memory protection unit (MPU) 14
00 0238 A800	00 023F FFFF	471K	Reserved	Reserved	Reserved
00 0240 0000	00 0243 FFFF	256K	Reserved	Reserved	Reserved
00 0244 0000	00 0244 3FFF	16K	DSP trace formatter 0	DSP trace formatter 0	DSP trace formatter 0
00 0244 4000	00 0244 FFFF	48K	Reserved	Reserved	Reserved
00 0245 0000	00 0245 3FFF	16K	DSP trace formatter 1	DSP trace formatter 1	DSP trace formatter 1
00 0245 4000	00 0245 FFFF	48K	Reserved	Reserved	Reserved
00 0246 0000	00 0246 3FFF	16K	DSP trace formatter 2	DSP trace formatter 2	DSP trace formatter 2
00 0246 4000	00 0246 FFFF	48K	Reserved	Reserved	Reserved
00 0247 0000	00 0247 3FFF	16K	DSP trace formatter 3	DSP trace formatter 3	DSP trace formatter 3
00 0247 4000	00 0247 FFFF	48K	Reserved	Reserved	Reserved
00 0248 0000	00 0248 3FFF	16K	Reserved	Reserved	Reserved
00 0248 4000	00 0248 FFFF	48K	Reserved	Reserved	Reserved
00 0249 0000	00 0249 3FFF	16K	Reserved	Reserved	Reserved
00 0249 4000	00 0249 FFFF	48K	Reserved	Reserved	Reserved
00 024A 0000	00 024A 3FFF	16K	Reserved	Reserved	Reserved
00 024A 4000	00 024A FFFF	48K	Reserved	Reserved	Reserved
00 024B 0000	00 024B 3FFF	16K	Reserved	Reserved	Reserved
00 024B 4000	00 024B FFFF	48K	Reserved	Reserved	Reserved
00 024C 0000	00 024C 01FF	512	Reserved	Reserved	Reserved
00 024C 0200	00 024C 03FF	1K-512	Reserved	Reserved	Reserved
00 024C 0400	00 024C 07FF	1K	Reserved	Reserved	Reserved
00 024C 0800	00 024C FFFF	62K	Reserved	Reserved	Reserved
00 024D 0000	00 024F FFFF	192K	Reserved	Reserved	Reserved
00 0250 0000	00 0250 007F	128	Reserved	Reserved	Reserved
00 0250 0080	00 0250 7FFF	32K-128	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0250 8000	00 0250 FFFF	32K	Reserved	Reserved	Reserved
00 0251 0000	00 0251 FFFF	64K	Reserved	Reserved	Reserved
00 0252 0000	00 0252 03FF	1K	Reserved	Reserved	Reserved
00 0252 0400	00 0252 FFFF	64K-1K	Reserved	Reserved	Reserved
00 0253 0000	00 0253 007F	128	I^2C0	I^2C0	I^2C0
00 0253 0080	00 0253 03FF	1K-128	Reserved	Reserved	Reserved
00 0253 0400	00 0253 047F	128	I^2C1	I^2C1	I^2C1
00 0253 0480	00 0253 07FF	1K-128	Reserved	Reserved	Reserved
00 0253 0800	00 0253 087F	128	I^2C2	I^2C2	I^2C2
00 0253 0880	00 0253 0BFF	1K-128	Reserved	Reserved	Reserved
00 0253 0C00	00 0253 0C3F	64	UART0	UART0	UART0
00 0253 0C40	00 0253 FFFF	1K-64	Reserved	Reserved	Reserved
00 0253 1000	00 0253 103F	64	UART1	UART1	UART1
00 0253 1040	00 0253 FFFF	60K-64	Reserved	Reserved	Reserved
00 0254 0000	00 0255 FFFF	128K	Reserved	Reserved	Reserved
00 0256 0000	00 0257 FFFF	128K	ARM CorePac INTC (GIC400) Memory Mapped Registers	ARM CorePac INTC (GIC400) Memory Mapped Registers	ARM CorePac INTC (GIC400) Memory Mapped Registers
00 0258 0000	00 025B FFFF	256K	Reserved	Reserved	Reserved
00 025C 0000	00 025CFFFF	256K	Reserved	Reserved	Reserved
00 0260 0000	00 0260 1FFF	8K	Secondary interrupt controller (INTC) 0	Secondary interrupt controller (INTC) 0	Secondary interrupt controller (INTC) 0
00 0260 2000	00 0260 3FFF	8K	Reserved	Reserved	Reserved
00 0260 4000	00 0260 5FFF	8K	Reserved	Reserved	Reserved
00 0260 6000	00 0260 7FFF	8K	Reserved	Reserved	Reserved
00 0260 8000	00 0260 9FFF	8K	Secondary interrupt controller (INTC) 2	Secondary interrupt controller (INTC) 2	Secondary interrupt controller (INTC) 2
00 0260 A000	00 0260 BEFF	8K-256	Reserved	Reserved	Reserved
00 0260 BF00	00 0260 BFFF	256	GPIO Config	GPIO Config	GPIO Config
00 0260 C000	00 0261 BFFF	64K	Reserved	Reserved	Reserved
00 0261 C000	00 0261 FFFF	16K	Reserved	Reserved	Reserved
00 0262 0000	00 0262 0FFF	4K	BOOTCFG chip-level registers	BOOTCFG chip-level registers	BOOTCFG chip-level registers
00 0262 1000	00 0262 FFFF	60K	Reserved	Reserved	Reserved
00 0263 0000	00 0263 FFFF	64K	USB PHY Config	USB PHY Config	USB PHY Config
00 0264 0000	00 0264 07FF	2K	Semaphore Config	Semaphore Config	Semaphore Config
00 0264 0800	00 0264 FFFF	62K	Reserved	Reserved	Reserved
00 0265 0000	00 0267 FFFF	192K	Reserved	Reserved	Reserved
00 0268 0000	00 0268 FFFF	512K	USB MMR Config	USB MMR Config	USB MMR Config
00 0270 0000	00 0270 7FFF	32K	EDMA channel controller (TPCC) 0	EDMA channel controller (TPCC) 0	EDMA channel controller (TPCC) 0
00 0270 8000	00 0270 FFFF	32K	Reserved	Reserved	Reserved
00 0271 0000	00 0271 FFFF	64K	Reserved	Reserved	Reserved
00 0272 0000	00 0272 7FFF	32K	EDMA channel controller (TPCC) 1	EDMA channel controller (TPCC) 1	EDMA channel controller (TPCC) 1
00 0272 8000	00 0272 FFFF	32K	Reserved	Reserved	Reserved
00 0273 0000	00 0273 FFFF	64K	Reserved	Reserved	Reserved
00 0274 0000	00 0274 7FFF	32K	EDMA channel controller (TPCC) 2	EDMA channel controller (TPCC) 2	EDMA channel controller (TPCC) 2
00 0274 8000	00 0275 FFFF	96K	Reserved	Reserved	Reserved
00 0276 0000	00 0276 03FF	1K	EDMA TPCC0 transfer controller (TPTC) 0	EDMA TPCC0 transfer controller (TPTC) 0	EDMA TPCC0 transfer controller (TPTC) 0
00 0276 0400	00 0276 7FFF	31K	Reserved	Reserved	Reserved
00 0276 8000	00 0276 83FF	1K	EDMA TPCC0 transfer controller (TPTC) 1	EDMA TPCC0 transfer controller (TPTC) 1	EDMA TPCC0 transfer controller (TPTC) 1
00 0276 8400	00 0276 FFFF	31K	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0277 0000	00 0277 03FF	1K	EDMA TPCC1 transfer controller (TPTC) 0	EDMA TPCC1 transfer controller (TPTC) 0	EDMA TPCC1 transfer controller (TPTC) 0
00 0277 0400	00 0277 7FFF	31K	Reserved	Reserved	Reserved
00 0277 8000	00 0277 83FF	1K	EDMA TPCC1 transfer controller (TPTC) 1	EDMA TPCC1 transfer controller (TPTC) 1	EDMA TPCC1 transfer controller (TPTC) 1
00 0278 0400	00 0277 FFFF	31K	Reserved	Reserved	Reserved
00 0278 0000	00 0278 03FF	1K	EDMA TPCC1 transfer controller (TPTC) 2	EDMA TPCC1 transfer controller (TPTC) 2	EDMA TPCC1 transfer controller (TPTC) 2
00 0278 0400	00 0278 7FFF	31K	Reserved	Reserved	Reserved
00 0278 8000	00 0278 83FF	1K	EDMA TPCC1 transfer controller (TPTC) 3	EDMA TPCC1 transfer controller (TPTC) 3	EDMA TPCC1 transfer controller (TPTC) 3
00 0278 8400	00 0278 FFFF	31K	Reserved	Reserved	Reserved
00 0279 0000	00 0279 03FF	1K	EDMA TPCC2 transfer controller (TPTC) 0	EDMA TPCC2 transfer controller (TPTC) 0	EDMA TPCC2 transfer controller (TPTC) 0
00 0279 0400	00 0279 7FFF	31K	Reserved	Reserved	Reserved
00 0279 8000	00 0279 83FF	1K	EDMA TPCC2 transfer controller (TPTC) 1	EDMA TPCC2 transfer controller (TPTC) 1	EDMA TPCC2 transfer controller (TPTC) 1
00 0279 8400	00 0279 FFFF	31K	Reserved	Reserved	Reserved
00 027A 0000	00 027A 03FF	1K	EDMA TPCC2 transfer controller (TPTC) 2	EDMA TPCC2 transfer controller (TPTC) 2	EDMA TPCC2 transfer controller (TPTC) 2
00 027A 0400	00 027A 7FFF	31K	Reserved	Reserved	Reserved
00 027A 8000	00 027A 83FF	1K	EDMA TPCC2 transfer controller (TPTC) 3	EDMA TPCC2 transfer controller (TPTC) 3	EDMA TPCC2 transfer controller (TPTC) 3
00 027A 8400	00 027A FFFF	31K	Reserved	Reserved	Reserved
00 027B 0000	00 027B 03FF	1K	Reserved	Reserved	Reserved
00 027B 0400	00 027B 7FFF	31K	Reserved	Reserved	Reserved
00 027B 8000	00 027B 83FF	1K	Reserved	Reserved	Reserved
00 027B 8400	00 027B 87FF	1K	Reserved	Reserved	Reserved
00 027B 8800	00 027B 8BFF	1K	Reserved	Reserved	Reserved
00 027B 8C00	00 027B FFFF	29K	Reserved	Reserved	Reserved
00 027C 0000	00 027C 03FF	1K	Reserved	Reserved	Reserved
00 027C 0400	00 027C FFFF	63K	Reserved	Reserved	Reserved
00 027D 0000	00 027D 3FFF	16K	TI embedded trace buffer (TETB) - CorePac0	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	TBR ARM CorePac - Trace buffer - ARM CorePac
00 027D 8000	00 027D FFFF	32K	Reserved	Reserved	Reserved
00 027E 0000	00 027E 3FFF	16K	TI embedded trace buffer (TETB) - CorePac1	TI embedded trace buffer (TETB) - CorePac1	TI embedded trace buffer (TETB) - CorePac1
00 027E 4000	00 027E FFFF	48K	Reserved	Reserved	Reserved
00 027F 0000	00 027F 3FFF	16K	TI embedded trace buffer (TETB) - CorePac2	TI embedded trace buffer (TETB) - CorePac2	TI embedded trace buffer (TETB) - CorePac2
00 027F 4000	00 027F FFFF	48K	Reserved	Reserved	Reserved
00 0280 0000	00 0280 3FFF	16K	TI embedded trace buffer (TETB) - CorePac3	TI embedded trace buffer (TETB) - CorePac3	TI embedded trace buffer (TETB) - CorePac3
00 0280 4000	00 0280 FFFF	48K	Reserved	Reserved	Reserved
00 0281 0000	00 0281 3FFF	16K	Reserved	Reserved	Reserved
00 0281 4000	00 0281 FFFF	48K	Reserved	Reserved	Reserved
00 0282 0000	00 0282 3FFF	16K	Reserved	Reserved	Reserved
00 0282 4000	00 0282 FFFF	48K	Reserved	Reserved	Reserved
00 0283 0000	00 0283 3FFF	16K	Reserved	Reserved	Reserved
00 0283 4000	00 0283 FFFF	48K	Reserved	Reserved	Reserved
00 0284 0000	00 0284 3FFF	16K	Reserved	Reserved	Reserved
00 0284 4000	00 0284 FFFF	48K	Reserved	Reserved	Reserved
00 0285 0000	00 0285 7FFF	32K	TBR_SYS-Trace Buffer -System	TBR_SYS-Trace Buffer -System	TBR_SYS-Trace Buffer -System

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 0285 8000	00 0285 FFFF	32K	Reserved	Reserved	Reserved
00 0286 0000	00 028F FFFF	640K	Reserved	Reserved	Reserved
00 0290 0000	00 0293 FFFF	256K	Reserved	Reserved	Reserved
00 0294 0000	00 029F FFFF	768K	Reserved	Reserved	Reserved
00 02A0 0000	00 02AF FFFF	1M	Navigator configuration	Navigator configuration	Navigator configuration
00 02B0 0000	00 02BF FFFF	1M	Navigator linking RAM	Navigator linking RAM	Navigator linking RAM
00 02C0 0000	00 02C0 FFFF	64K	Reserved	Reserved	Reserved
00 02C1 0000	00 02C1 FFFF	64K	Reserved	Reserved	Reserved
00 02C2 0000	00 02C3 FFFF	128K	Reserved	Reserved	Reserved
00 02C4 0000	00 02C5 FFFF	128K	Reserved	Reserved	Reserved
00 02C6 0000	00 02C7 FFFF	128K	Reserved	Reserved	Reserved
00 02C8 0000	00 02C8 FFFF	64K	Reserved	Reserved	Reserved
00 02C9 0000	00 02C9 FFFF	64K	Reserved	Reserved	Reserved
00 02CA 0000	00 02CB FFFF	128K	Reserved	Reserved	Reserved
00 02CC 0000	00 02CD FFFF	128K	Reserved	Reserved	Reserved
00 02CE 0000	00 02EF FFFF	15M-896K	Reserved	Reserved	Reserved
00 02F0 0000	00 02FF FFFF	1M	Reserved	Reserved	Reserved
00 0300 0000	00 030F FFFF	1M	Debug_SS Configuration	Debug_SS Configuration	Debug_SS Configuration
00 0310 0000	00 07FF FFFF	79M	Reserved	Reserved	Reserved
00 0800 0000	00 0801 FFFF	128K	Extended memory controller (XMC) configuration	Extended memory controller (XMC) configuration	Extended memory controller (XMC) configuration
00 0802 0000	00 0BBF FFFF	60M-128K	Reserved	Reserved	Reserved
00 0BC0 0000	00 0BCF FFFF	1M	Multicore shared memory controller (MSMC) config	Multicore shared memory controller (MSMC) config	Multicore shared memory controller (MSMC) config
00 0BD0 0000	00 0BFF FFFF	3M	Reserved	Reserved	Reserved
00 0C00 0000	00 0C1F FFFF	2M	Multicore shared memory (MSM)	Multicore shared memory (MSM)	Multicore shared memory (MSM)
00 0C20 0000	00 0FFF FFFF	62M	Reserved	Reserved	Reserved
00 1000 0000	00 107F FFFF	8M	Reserved	Reserved	Reserved
00 1080 0000	00 108F FFFF	1M	CorePac0 L2 SRAM	CorePac0 L2 SRAM	CorePac0 L2 SRAM
00 1090 0000	00 10DF FFFF	5M	Reserved	Reserved	Reserved
00 10E0 0000	00 10E0 7FFF	32K	CorePac0 L1P SRAM	CorePac0 L1P SRAM	CorePac0 L1P SRAM
00 10E0 8000	00 10EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 10F0 0000	00 10F0 7FFF	32K	CorePac0 L1D SRAM	CorePac0 L1D SRAM	CorePac0 L1D SRAM
00 10F0 8000	00 117F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1180 0000	00 118F FFFF	1M	CorePac1 L2 SRAM	CorePac1 L2 SRAM	CorePac1 L2 SRAM
00 1190 0000	00 11DF FFFF	5M	Reserved	Reserved	Reserved
00 11E0 0000	00 11E0 7FFF	32K	CorePac1 L1P SRAM	CorePac1 L1P SRAM	CorePac1 L1P SRAM
00 11E0 8000	00 11EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 11F0 0000	00 11F0 7FFF	32K	CorePac1 L1D SRAM	CorePac1 L1D SRAM	CorePac1 L1D SRAM
00 11F0 8000	00 127F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1280 0000	00 128F FFFF	1M	CorePac2 L2 SRAM	CorePac2 L2 SRAM	CorePac2 L2 SRAM
00 1290 0000	00 12DF FFFF	5M	Reserved	Reserved	Reserved
00 12E0 0000	00 12E0 7FFF	32K	CorePac2 L1P SRAM	CorePac2 L1P SRAM	CorePac2 L1P SRAM
00 12E0 8000	00 12EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 12F0 0000	00 12F0 7FFF	32K	CorePac2 L1D SRAM	CorePac2 L1D SRAM	CorePac2 L1D SRAM
00 12F0 8000	00 137F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1380 0000	00 1388 FFFF	1M	CorePac3 L2 SRAM	CorePac3 L2 SRAM	CorePac3 L2 SRAM
00 1390 0000	00 13DF FFFF	5M	Reserved	Reserved	Reserved
00 13E0 0000	00 13E0 7FFF	32K	CorePac3 L1P SRAM	CorePac3 L1P SRAM	CorePac3 L1P SRAM
00 13E0 8000	00 13EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 13F0 0000	00 13F0 7FFF	32K	CorePac3 L1D SRAM	CorePac3 L1D SRAM	CorePac3 L1D SRAM

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 13F0 8000	00 147F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1480 0000	00 148F FFFF	1M	Reserved	Reserved	Reserved
00 1490 0000	00 14DF FFFF	5M	Reserved	Reserved	Reserved
00 14E0 0000	00 14E0 7FFF	32K	Reserved	Reserved	Reserved
00 14E0 8000	00 14EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 14F0 0000	00 14F0 7FFF	32K	Reserved	Reserved	Reserved
00 14F0 8000	00 157F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1580 0000	00 158F FFFF	1M	Reserved	Reserved	Reserved
00 1590 0000	00 15DF FFFF	5M	Reserved	Reserved	Reserved
00 15E0 0000	00 15E0 7FFF	32K	Reserved	Reserved	Reserved
00 15E0 8000	00 15EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 15F0 0000	00 15F0 7FFF	32K	Reserved	Reserved	Reserved
00 15F0 8000	00 167F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1680 0000	00 168F FFFF	1M	Reserved	Reserved	Reserved
00 1690 0000	00 16DF FFFF	5M	Reserved	Reserved	Reserved
00 16E0 0000	00 16E0 7FFF	32K	Reserved	Reserved	Reserved
00 16E0 8000	00 16EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 16F0 0000	00 16F0 7FFF	32K	Reserved	Reserved	Reserved
00 16F0 8000	00 177F FFFF	9M-32K	Reserved	Reserved	Reserved
00 1780 0000	00 178F FFFF	1M	Reserved	Reserved	Reserved
00 1790 0000	00 17DF FFFF	5M	Reserved	Reserved	Reserved
00 17E0 0000	00 17E0 7FFF	32K	Reserved	Reserved	Reserved
00 17E0 8000	00 17EF FFFF	1M-32K	Reserved	Reserved	Reserved
00 17F0 0000	00 17F0 7FFF	32K	Reserved	Reserved	Reserved
00 17F0 8000	00 1FFF FFFF	129M-32K	Reserved	Reserved	Reserved
00 2000 0000	00 200F FFFF	1M	System trace manager (STM) configuration	System trace manager (STM) configuration	System trace manager (STM) configuration
00 2010 0000	00 201F FFFF	1M	Reserved	Reserved	Reserved
00 2020 0000	00 205F FFFF	4M	Reserved	Reserved	Reserved
00 2060 0000	00 206F FFFF	1M	Reserved	Reserved	Reserved
00 2070 0000	00 207F FFFF	1M	Reserved	Reserved	Reserved
00 2080 0000	00 208F FFFF	1M	Reserved	Reserved	Reserved
00 2090 0000	00 209F FFFF	1M	Reserved	Reserved	Reserved
00 20A0 0000	00 20A3 FFFF	256K	Reserved	Reserved	Reserved
00 20A4 0000	00 20A4 FFFF	64K	Reserved	Reserved	Reserved
00 20A5 0000	00 20AF FFFF	704K	Reserved	Reserved	Reserved
00 20B0 0000	00 20B3 FFFF	256K	Boot ROM	Boot ROM	Boot ROM
00 20B4 0000	00 20BE FFFF	704K	Reserved	Reserved	Reserved
00 20BF 0000	00 20BF 01FF	64K	Reserved	Reserved	Reserved
00 20C0 0000	00 20FF FFFF	4M	Reserved	Reserved	Reserved
00 2100 0000	00 2100 03FF	1K	Reserved	Reserved	Reserved
00 2100 0400	00 2100 05FF	512	SPI0	SPI0	SPI0
00 2100 0600	00 2100 07FF	512	SPI1	SPI1	SPI1
00 2100 0800	00 2100 09FF	512	SPI2	SPI2	SPI2
00 2100 0A00	00 2100 0AFF	256	AEMIF Config	AEMIF Config	AEMIF Config
00 2100 0B00	00 2100 FFFF	62K-768	Reserved	Reserved	Reserved
00 2101 0000	00 2101 01FF	512	DDR3A EMIF Config	Reserved	DDR3A EMIF Config
00 2101 0200	00 2101 07FF	2K-512	Reserved	Reserved	Reserved
00 2101 0800	00 2101 09FF	512	Reserved	Reserved	Reserved
00 2101 0A00	00 2101 0FFF	2K-512	Reserved	Reserved	Reserved
00 2101 1000	00 2101 FFFF	60K	Reserved	Reserved	Reserved

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
00 2102 0000	00 2102 7FFF	32K	PCIe1 config	PCIe1 config	PCIe1 config
00 2102 8000	00 217F FFFF	4M-192K	Reserved	Reserved	Reserved
00 2140 0000	00 2140 00FF	256	Reserved	Reserved	Reserved
00 2140 0100	00 2140 01FF	256	Reserved	Reserved	Reserved
00 2140 0400	00 217F FFFF	4M-512	Reserved	Reserved	Reserved
00 2180 0000	00 2180 7FFF	32K	PCIe0 config	PCIe0 config	PCIe0 config
00 2180 8000	00 21BF FFFF	4M-32K	Reserved	Reserved	Reserved
00 21C0 0000	00 21FF FFFF	4M	Reserved	Reserved	Reserved
00 2200 0000	00 229F FFFF	10M	Reserved	Reserved	Reserved
00 22A0 0000	00 22A0 FFFF	64K	Reserved	Reserved	Reserved
00 22A1 0000	00 22AF FFFF	1M-64K	Reserved	Reserved	Reserved
00 22B0 0000	00 22B0 FFFF	64K	Reserved	Reserved	Reserved
00 22B1 0000	00 22BF FFFF	1M-64K	Reserved	Reserved	Reserved
00 22C0 0000	00 22C0 FFFF	64K	Reserved	Reserved	Reserved
00 22C1 0000	00 22CF FFFF	1M-64K	Reserved	Reserved	Reserved
00 22D0 0000	00 22D0 FFFF	64K	Reserved	Reserved	Reserved
00 22D1 0000	00 22DF FFFF	1M-64K	Reserved	Reserved	Reserved
00 22E0 0000	00 22E0 FFFF	64K	Reserved	Reserved	Reserved
00 22E1 0000	00 22EF FFFF	1M-64K	Reserved	Reserved	Reserved
00 22F0 0000	00 22F0 FFFF	64K	Reserved	Reserved	Reserved
00 22F1 0000	00 22FF FFFF	1M-64K	Reserved	Reserved	Reserved
00 2300 0000	00 2300 FFFF	64K	Reserved	Reserved	Reserved
00 2301 0000	00 230F FFFF	1M-64K	Reserved	Reserved	Reserved
00 2310 0000	00 2310 FFFF	64K	Reserved	Reserved	Reserved
00 2311 0000	00 231F FFFF	1M-64K	Reserved	Reserved	Reserved
00 2320 0000	00 2323 FFFF	256K	Reserved	Reserved	Reserved
00 2324 0000	00 239F FFFF	8M-256K	Reserved	Reserved	Reserved
00 23A0 0000	00 23BF FFFF	2M	Navigator	Navigator	Navigator
00 23C0 0000	00 23CF FFFF	1M	Reserved	Reserved	Reserved
00 23D0 0000	00 23FF FFFF	3M	Reserved	Reserved	Reserved
00 2400 0000	00 25FF FFFF	32M	DFE configuration	DFE configuration	DFE configuration
00 2600 0000	00 26FF FFFF	16M	NetCP configuration	NetCP configuration	NetCP configuration
00 2700 0000	00 273 F FFFF	4M	IQNet configuration	IQNet configuration	IQNet configuration
00 274 0000	00 2FFF FFFF	140M	Reserved	Reserved	Reserved
00 3000 0000	00 33FF FFFF	64M	EMIF16 CE0	EMIF16 CE0	EMIF16 CE0
00 3400 0000	00 37FF FFFF	64M	EMIF16 CE1	EMIF16 CE1	EMIF16 CE1
00 3800 0000	00 3BFF FFFF	64M	EMIF16 CE2	EMIF16 CE2	EMIF16 CE2
00 3C00 0000	00 3FFF FFFF	64M	EMIF16 CE3	EMIF16 CE3	EMIF16 CE3
00 4000 0000	00 4FFF FFFF	256M	Reserved	Reserved	Reserved
00 5000 0000	00 5FFF FFFF	256M	PCIe 0 data	PCIe 0 data	PCIe 0 data
006000 0000	00 6FFF FFFF	256M	PCIe 1 data	PCIe 1 data	PCIe 1 data
00 7000 0000	00 700F FFFF	1M	OSR data	OSR data	OSR data
00 7010 0000	00 7FFF FFFF	255M	Reserved	Reserved	Reserved
00 8000 0000	00 FFFF FFFF	2G	DDR3A data	DDR3A data	DDR3A data
01 0000 0000	01 2100 FFFF	528M+64K	Reserved	Reserved	Reserved
01 2101 0000	01 2101 01FF	512	DDR3A EMIF configuration(1)	DDR3A EMIF configuration(2)	DDR3A EMIF configuration(3)
01 2101 0200	07 FFFF FFFF	32G-512	Reserved	Reserved	Reserved

(1) This region is aliased to 00 2101 0000-00 2101 01FF.
(2) Access to 40-bit address requires XMC MPAX programmation.
(3) Access to 40-bit address requires MSMC MPAX programmation. MPAX from SES port need to re-map the region of 00 2101 0000-00 2101 01FF to this region.

Table 7-1. Device Memory Map Summary for 66AK2L06 (continued)

PHYSICAL 40 BIT ADDRESS		BYTES	ARM VIEW	DSP VIEW	SOC VIEW
START	END				
08 0000 0000	09 FFFF FFFF	8G	DDR3A data	DDR3A data(2)	DDR3A data(3)
0A 0000 0000	FF FFFF FFFF	984G	Reserved	Reserved	Reserved

7.2 Memory Protection Unit (MPU)

CFG (configuration) space of all slave devices on the TeraNet is protected by the MPU. The 66AK2L06 contains sixteen MPUs:

- MPU0 is used for main TeraNet_3P_B (SCR_3P (B)) CFG.
- MPU1/2/5 are used for QM_SS (one for VBUSM port and one each for the two configuration VBUSP ports).
- MPU3 is reserved.
- MPU4 is reserved.
- MPU6 is reserved.
- MPU7 is used for OSR data.
- MPU8 is used for EMIF16.
- MPU9 is used for interrupt controllers (GIC, CIC0 and CIC2) 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 DFE, IQNet and NetCP CFG.

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 7-2. 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 7-3. MPU6-MPU11 Default Configuration

SETTING	MPU6	MPU7 OSR	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 7-4. MPU12-MPU15 Default Configuration

SETTING	MPU12 SPI0	MPU13 SPI1	MPU14 SPI2	MPU15 DFE, IQNet, NetCP
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	16
Compare width	1KB granularity	1KB granularity	1KB granularity	1KB granularity

Table 7-5. 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	0x027C_0000	0x027C_03FF
MPU4	Reserved	0x0210_0000	0x0215_FFFF
MPU5	QM_SS CFG2 PORT	0x02A0_4000	0x02BF_FFFF
MPU6	Reserved	0x02C0_0000	0x02CD_FFFF
MPU7	OSR	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
MPU15	DFE, IQNet, NetCP	0x2400_0000	0x2508_FFFF

Table 7-6 shows the unique Master ID assigned to each CorePac and peripherals on the device.

Table 7-6. Master ID Settings

MASTER ID	66AK2L06
0	C66x CorePac0 Data
1	C66x CorePac1 Data
2	C66x CorePac2 Data
3	C66x CorePac3 Data
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	ARM CorePac 0
9	ARM CorePac 1
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	C66x CorePac0 CFG
17	C66x CorePac1 CFG

Table 7-6. Master ID Settings (continued)

MASTER ID	66AK2L06
18	C66x CorePac2 CFG
19	C66x CorePac3 CFG
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	EDMA0_TC0 read
26	EDMA0_TC0 write
27	EDMA0_TC1 read
28	PCIe 1
29	Reserved
30	Reserved
31	PCIe 0
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
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	Reserved
50	Reserved
51	Reserved
52	Reserved
53	Reserved
54-55	NETCP_GLOBAL1
56	USB
57	FFTC_1
58	Reserved
59	Reserved
60	Reserved
61	Reserved
62	EDMA3CC0
63	EDMA3CC1
64	EDMA3CC2
65	Reserved

Table 7-6. Master ID Settings (continued)

MASTER ID	66AK2L06
66	Reserved
67	Reserved
68-71	Queue Manager Second
72-79	IQNet
80	Reserved
81	Reserved
82	Reserved
83	Reserved
84-87	Reserved
88	Reserved
89	Reserved
90	Reserved
91	Reserved
92-95	Reserved
96-99	Packet Coprocessor MST1
100-101	Reserved
102	Reserved
103	Reserved
104	Reserved
105	Reserved
106	FFTC_0_CDMA
107	DBG_DAP
108-111	Reserved
112-119	NETCP_LOCAL
120-139	Reserved
140	CPT_L2_0
141	CPT_L2_1
142	CPT_L2_2
143	CPT_L2_3
144	Reserved
145	Reserved
146	Reserved
147	Reserved
148	CPT_MSMC0
149	CPT_MSMC1
150	CPT_MSMC2
151	CPT_MSMC3
152	CPT_DDR3A
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_OSR_PCIe1

Table 7-6. Master ID Settings (continued)

MASTER ID	66AK2L06
162	Reserved
163	Reserved
164	CPT_EDMA3CC0
165	CPT_EDMA3CC1_2
166	CPT_INTC
167	CPT_SPI_ROM_EMIP16
168	Reserved
168	Reserved
169	Reserved
170	Reserved
171	Reserved
172	Reserved
173	Reserved
174	CPT_MSMC5
175	CPT_MSMC6
176	CPT_MSMC7
177	CPT_MSMC4
178	CPT_CFG_3P_U
179	Reserved
180-183	NETCP_GLOBAL0
184-255	Reserved

NOTE

There are two master ID values assigned to the Queue Manager_second master port, one master ID for external linking RAM and the other one for the PDSP/MCDM accesses.

Table 7-7 shows the privilege ID of each C66x CorePac and every 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 7-7. Privilege ID Settings

PRIVILEGE ID	MASTER	PRIVILEGE LEVEL	SECURITY LEVEL	ACCESS TYPE
0	C66x CorePac0	SW dependent, driven by MSMC	Non-secure	DMA
1	C66x CorePac1	SW dependent, driven by MSMC	Non-secure	DMA
2	C66x CorePac2	SW dependent, driven by MSMC	Non-secure	DMA
3	C66x CorePac3	SW dependent, driven by MSMC	Non-secure	DMA
4	Reserved			
5	Reserved			
6	Reserved			
7	Reserved			
8	ARM CorePac	SW dependent	Non-secure	DMA

Table 7-7. Privilege ID Settings (continued)

PRIVILEGE ID	MASTER	PRIVILEGE LEVEL	SECURITY LEVEL	ACCESS TYPE
9	All Packet DMA masters (NetCP, QM_CDMA, FFTC, IQNet_CDMA, and USB	User mode and supervisor mode is determined by per transaction basis. Only the transaction with source ID matching the value in SupervisorID register is granted supervisor mode.	Non-secure	DMA
10	QM_Second(1)	User	Non-secure	DMA
11	PCIe 0	Supervisor	Non-secure	DMA
12	DAP	Driven by Emulation SW	Driven by Emulation SW	DMA
13	Reserved			
14	PCIe 1	Supervisor	Non-secure	DMA
15	Reserved			

(1) QM_Second provides a path that PDSP uses to access the system memory.

7.2.1 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.

7.2.1.1 MPU Register Map

Table 7-8. 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
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

Table 7-8. MPU Registers (continued)

OFFSET	NAME	DESCRIPTION
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

7.2.1.2 Device-Specific MPU Registers

7.2.1.2.1 Configuration Register (CONFIG)

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

Figure 7-1. 66AK2L06 Configuration Register (CONFIG)

		31 24	23 20	19 16	15 12	11 1	0
		ADDR_WIDTH	NUM_FIXED	NUM_PROG	NUM_AIDS	Reserved	ASSUME_ALLOWED
Reset Values	MPU0	R-0	R-0	R-16	R-16	R-0	R-1
	MPU1	R-0	R-0	R-16	R-16	R-0	R-1
	MPU2	R-0	R-0	R-16	R-16	R-0	R-1
	MPU3	Reserved					
	MPU4	Reserved					
	MPU5	R-0	R-0	R-16	R-16	R-0	R-1
	MPU6	Reserved					
	MPU7	R-0	R-0	R-16	R-16	R-0	R-1
	MPU8	R-0	R-0	R-8	R-16	R-0	R-1
	MPU9	R-0	R-0	R-4	R-16	R-0	R-1
	MPU10	R-0	R-0	R-2	R-16	R-0	R-1
	MPU11	R-0	R-0	R-16	R-16	R-0	R-1
	MPU12	R-0	R-0	R-2	R-16	R-0	R-1
	MPU13	R-0	R-0	R-2	R-16	R-0	R-1
	MPU14	R-0	R-0	R-2	R-16	R-0	R-1
	MPU15	R-0	R-0	R-16	R-16	R-0	R-1

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

Table 7-9. 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

Figure 7-2. 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 7-10. 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 7-11. 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	0x027C_0000	Reserved	0x02A0_4000
PROG1_MPSAR	0x01F0_0000	0x23A0_2000	0x02A0_2000	N/A	Reserved	0x02A0_5000
PROG2_MPSAR	0x02F0_0000	0x023A_6000	0x02A0_6000	N/A	Reserved	0x02A0_6400
PROG3_MPSAR	0x0200_0000	0x23A0_6800	0x02A0_6800	N/A	Reserved	0x02A0_7400
PROG4_MPSAR	0x020C_0000	0x23A0_7000	0x02A0_7000	N/A	Reserved	0x02A0_A000
PROG5_MPSAR	0x021C_0000	0x23A0_8000	0x02A0_8000	N/A	Reserved	0x02A0_D000
PROG6_MPSAR	0x021D_0000	0x23A0_C000	0x02A0_C000	N/A	Reserved	0x02A0_E000
PROG7_MPSAR	0x021F_0000	0x23A0_E000	0x02A0_E000	N/A	Reserved	0x02A0_F000
PROG8_MPSAR	0x0234_0000	0x23A0_F000	0x02A0_F000	N/A	Reserved	0x02A0_F800
PROG9_MPSAR	0x0254_0000	0x23A0_F800	0x02A0_F800	N/A	Reserved	0x02A1_2000
PROG10_MPSAR	0x0258_0000	0x23A1_0000	0x02A1_0000	N/A	Reserved	0x02A1_C000
PROG11_MPSAR	Reserved	0x23A1_C000	0x02A2_0000	N/A	Reserved	0x02A2_8000
PROG12_MPSAR	0x0290_0000	0x23A4_0000	0x02A4_0000	N/A	Reserved	0x02A6_0000
PROG13_MPSAR	0x01E8_0000	0x23A8_0000	0x02A8_0000	N/A	Reserved	0x02AA_0000
PROG14_MPSAR	0x01E8_0800	0x23B0_0000	0x02AC_0000	N/A	Reserved	0x02B0_0000
PROG15_MPSAR	0x01E0_0000	0x23B8_0000	0x02AE_0000	N/A	Reserved	0x02B8_0000

Table 7-12. 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	Reserved	0x0231_0000
PROG2_MPSAR	Reserved	0x0800_0000	0x3400_0000	Reserved	N/A	0x0231_A000
PROG3_MPSAR	Reserved	0x1000_0000	0x3600_0000	N/A	N/A	0x0233_0000
PROG4_MPSAR	Reserved	0x1800_0000	0x3800_0000	N/A	N/A	0x0235_0000
PROG5_MPSAR	Reserved	0x2000_0000	0x3A00_0000	N/A	N/A	0x0263_0000
PROG6_MPSAR	Reserved	0x2800_0000	0x3C00_0000	N/A	N/A	0x0244_0000
PROG7_MPSAR	Reserved	0x3000_0000	0x2100_0800	N/A	N/A	0x024C_0000
PROG8_MPSAR	Reserved	0x3800_0000	N/A	N/A	N/A	0x0250_0000
PROG9_MPSAR	Reserved	0x4000_0000	N/A	N/A	N/A	0x0253_0000
PROG10_MPSAR	Reserved	0x4800_0000	N/A	N/A	N/A	0x0253_0C00
PROG11_MPSAR	Reserved	0x5000_0000	N/A	N/A	N/A	0x0260_B000
PROG12_MPSAR	Reserved	0x5800_0000	N/A	N/A	N/A	0x0262_0000
PROG13_MPSAR	Reserved	0x6000_0000	N/A	N/A	N/A	0x0300_0000
PROG14_MPSAR	Reserved	0x6800_0000	N/A	N/A	N/A	0x021E_0000
PROG15_MPSAR	Reserved	0x7000_0000	N/A	N/A	N/A	0x0268_0000

Table 7-13. 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	Reserved	Reserved	Reserved	0x2600_0000
PROG2_MPSAR	N/A	N/A	N/A	0x2700_0000
PROG3_MPSAR	N/A	N/A	N/A	Reserved
PROG4_MPSAR	N/A	N/A	N/A	N/A
PROG5_MPSAR	N/A	N/A	N/A	N/A
PROG6_MPSAR	N/A	N/A	N/A	N/A
PROG7_MPSAR	N/A	N/A	N/A	N/A

Table 7-13. MPU12-MPU15 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values (continued)

REGISTER	MPU12	MPU13	MPU14	MPU15
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

7.2.1.3 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 7-3. Programmable Range *n* End Address Register (PROG*n*_MPEAR)

31	10	9	0
END_ADDR		Reserved	
R/W		R	

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

Table 7-14. 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 7-15. MPU0-MPU5 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values

Table 7-16. 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	Reserved	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	Reserved	N/A	0x0235_0FFF
PROG5_MPEAR	Reserved	0x27FF_FFFF	0x3BFF_FFFF	Reserved	N/A	0x0263_FFFF
PROG6_MPEAR	Reserved	0x2FFF_FFFF	0x3FFF_FFFF	Reserved	N/A	0x024B_3FFF
PROG7_MPEAR	Reserved	0x37FF_FFFF	0x2100_0AFF	Reserved	N/A	0x024C_0BFF
PROG8_MPEAR	Reserved	0x3FFF_FFFF	N/A	Reserved	N/A	0x0250_7FFF
PROG9_MPEAR	Reserved	0x47FF_FFFF	N/A	Reserved	N/A	0x0253_0BFF
PROG10_MPEAR	Reserved	0x4FFF_FFFF	N/A	Reserved	N/A	0x0253_FFFF
PROG11_MPEAR	Reserved	0x57FF_FFFF	N/A	Reserved	N/A	0x0260_BFFF
PROG12_MPEAR	Reserved	0x5FFF_FFFF	N/A	Reserved	N/A	0x0262_0FFF
PROG13_MPEAR	Reserved	0x67FF_FFFF	N/A	Reserved	N/A	0x03FF_FFFF

Table 7-16. MPU6-MPU11 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values (continued)

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

Table 7-17. 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	0x25FF_FFFF
PROG1_MPEAR	Reserved	Reserved	Reserved	0x26FF_FFFF
PROG2_MPEAR	N/A	N/A	N/A	0x273FF_FFFF
PROG3_MPEAR	N/A	N/A	N/A	Reserved
PROG4_MPEAR	N/A	N/A	N/A	N/A
PROG5_MPEAR	N/A	N/A	N/A	N/A
PROG6_MPEAR	N/A	N/A	N/A	N/A
PROG7_MPEAR	N/A	N/A	N/A	N/A
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

7.2.1.4 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 7-4. 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 7-18. 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 7-18. 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 7-18. 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 7-19. 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 7-20. 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
PROG3_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB0

Table 7-20. MPU6-MPU11 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR) Reset Values (continued)

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
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 7-21. 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	Reserved	Reserved	Reserved	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	0x03FF_FCB6
PROG9_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG10_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG11_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG12_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG13_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG14_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG15_MPPAR	N/A	N/A	N/A	0x03FF_FCB6

7.3 Interrupts for 66AK2L06

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

7.3.1 Interrupt Sources and Interrupt Controller

The CPU interrupts on the 66AK2L06 device are configured through the C66x CorePac Interrupt Controller. The Interrupt Controller allows for up to 128 system events to be programmed to any of the 12 CPU interrupt inputs (CPUINT4 - CPUINT15), the CPU exception input (EXCEP), or the advanced emulation logic. The 128 system events consist of both internally-generated events (within the CorePac) and chip-level events.

Additional system events are routed to each of the C66x CorePacs to provide chip-level events that are not required as CPU interrupts/exceptions to be routed to the Interrupt Controller as emulation events. In addition, error-class events or infrequently used events are also routed through the system event router to offload the C66x CorePac interrupt selector. This is accomplished through the two CorePac Interrupt Controller blocks, CIC0 and CIC2. These CIC are clocked using CPU/6.

The event controllers consist of simple combination logic to provide additional events to each C66x CorePac, ARM GIC (ARM Generic Interrupt Controller) plus the EDMA3CC. CIC0 has 104 event outputs which provides 20 broadcast events and 18 additional events to each of the C66x CorePacs, 0 through 3. CIC1 is reserved. CIC2 has 103 event outputs which provides 8, 20, and 8 events to EDMA3CC0, EDMA3CC1, and EDMA3C2 respectively.

The events that are routed to the C66x CorePacs for Advanced Event Triggering (AET) purposes, from those EDMA3CC and FSYNC events that are not otherwise provided to each C66x CorePac.

Modules such as FFTC, CP_MPU (Coprocessor Memory Protection Unit), 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.

For FFTC:

- the EOI value is 0 for FFTC_x_INTD_INTR0,
- the EOI value is 1 for FFTC_x_INTD_INTR1,
- the EOI value is 2 for FFTC_x_INTD_INTR2
- the EOI value is 3 for FFTC_x_INTD_INTR3 (where FFTC_x can be FFTC_0 or FFTC_1)

Figure 7-5 shows the 66AK2L06 interrupt topology.

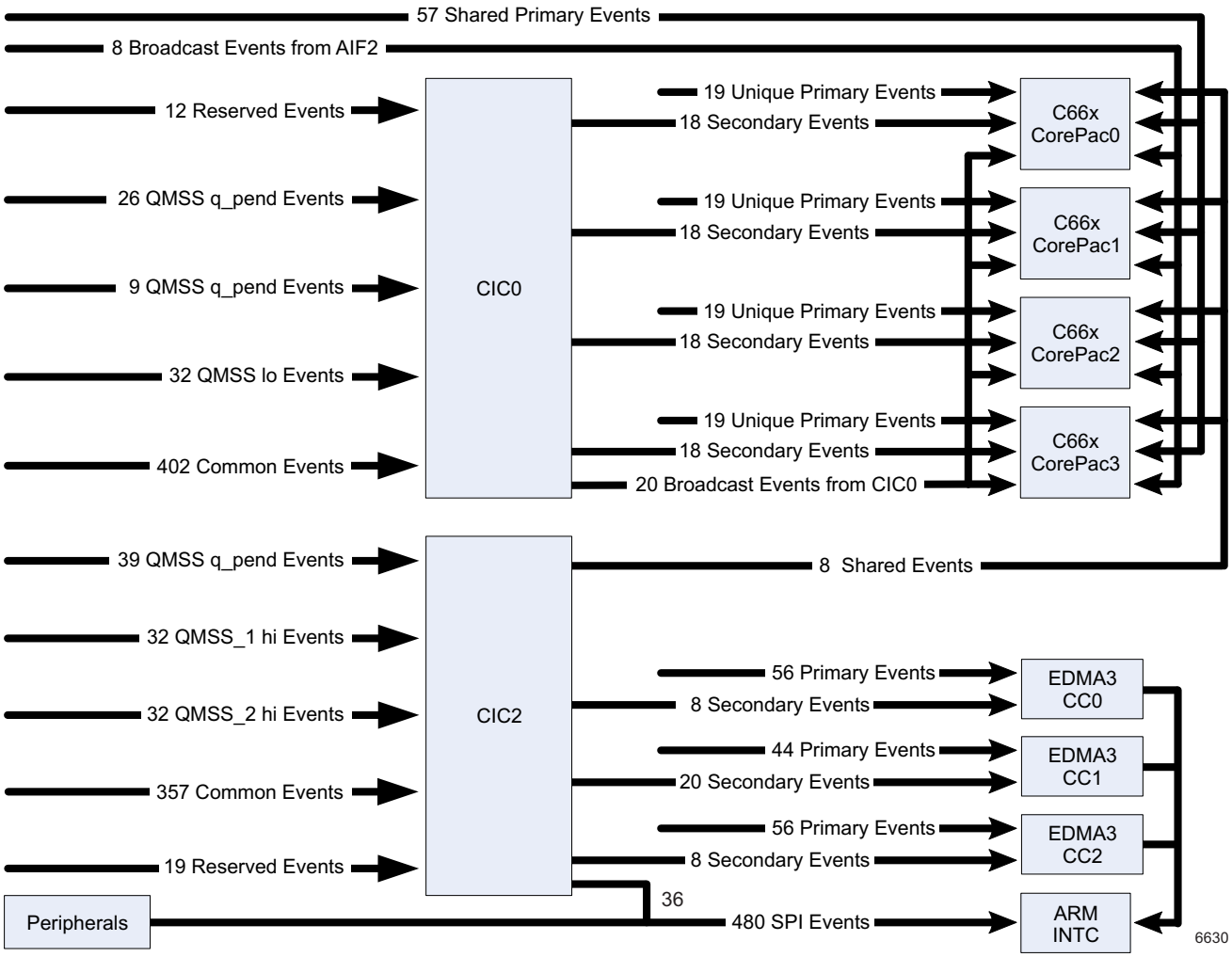


Figure 7-5. Interrupt Topology

Table 7-22 shows the mapping of primary events to C66x Corepac

Table 7-22. System Event Mapping — C66x CorePac Primary Interrupts

EVENT NO.	EVENT NAME	DESCRIPTION
0	EVT0	Event combiner 0 output
1	EVT1	Event combiner 1 output
2	EVT2	Event combiner 2 output
3	EVT3	Event combiner 3 output
4	TETB_HFULLINTN	TETB is half full
5	TETB_FULLINTN	TETB is full
6	TETB_ACQINTN	TETB Acquisition complete interrupt
7	TETB_OVFLINTN	TETB Overflow condition interrupt
8	TETB_UNFLINTN	TETB Underflow condition interrupt
9	EMU_DTDMA	Emulation interrupt for host scan, DTDMA transfer complete and AET
10	MSMC_MPF_ERRORN	Memory protection fault indicators for system master PrivID = 0 (C66x CorePac)
11	EMU_RTDXRX	Reserved
12	EMU_RTDXTX	Reserved

Table 7-22. System Event Mapping — C66x CorePac Primary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
13	IDMA0	IDMA channel 0 interrupt
14	IDMA1	IDMA channel 1 interrupt
15	SEM_ERRN	Semaphore error interrupt
16	SEM_INTN	Semaphore interrupt
17	PCIE_0_INT4_PLUS_N	PCIE_0 MSI interrupt
18	Reserved	
19	Reserved	
20	IQNET_INT0	IQNET interrupt
21	IQNET_INT1	IQNET interrupt
22	CIC_2_OUT98_PLUS_N	CIC Interrupt Controller output
23	CIC_OUT35	CIC Interrupt Controller output(1)
24	CIC_2_OUT102	CIC Interrupt Controller output
25	CIC_2_OUT94_PLUS_N	CIC Interrupt Controller output
26	CIC_OUT68_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
27	CIC_OUT69_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
28	CIC_OUT70_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
29	CIC_OUT71_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
30	CIC_OUT72_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
31	CIC_OUT73_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
32	CIC_OUT16	CIC Interrupt Controller output(1)
33	CIC_OUT17	CIC Interrupt Controller output(1)
34	CIC_OUT18	CIC Interrupt Controller output(1)
35	CIC_OUT19	CIC Interrupt Controller output(1)
36	CIC_OUT20	CIC Interrupt Controller output(1)
37	CIC_OUT21	CIC Interrupt Controller output(1)
38	CIC_OUT22	CIC Interrupt Controller output(1)
39	CIC_OUT23	CIC Interrupt Controller output(1)
40	CIC_OUT32	CIC Interrupt Controller output(1)
41	CIC_OUT33	CIC Interrupt Controller output(1)
42	CIC_OUT13_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
43	CIC_OUT14_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
44	CIC_OUT15_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
45	CIC_OUT64_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
46	CIC_OUT65_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
47	CIC_OUT66_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
48	QMSS_INTD_1_HIGH_N	Navigator 1 accumulated hi-priority interrupt 0
49	QMSS_INTD_1_HIGH_8_PLUS_N	Navigator 1 accumulated hi-priority interrupt 8
50	QMSS_INTD_1_HIGH_16_PLUS_N	Navigator 1 accumulated hi-priority interrupt 16
51	QMSS_INTD_1_HIGH_24_PLUS_N	Navigator 1 accumulated hi-priority interrupt 24
52	QMSS_INTD_2_HIGH_N	Navigator 2 accumulated hi-priority interrupt 0
53	QMSS_INTD_2_HIGH_8_PLUS_N	Navigator 2 accumulated hi-priority interrupt 8
54	QMSS_INTD_2_HIGH_16_PLUS_N	Navigator 2 accumulated hi-priority interrupt 16
55	QMSS_INTD_2_HIGH_24_PLUS_N	Navigator 2 accumulated hi-priority interrupt 24
56	CIC_OUT0	CIC Interrupt Controller output(1)
57	CIC_OUT1	CIC Interrupt Controller output(1)
58	CIC_OUT2	CIC Interrupt Controller output(1)

(1) For C66x CorePac[0-3], this generic primary interrupt comes from CIC0

Table 7-22. System Event Mapping — C66x CorePac Primary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
59	CIC_OUT3	CIC Interrupt Controller output(1)
60	CIC_OUT4	CIC Interrupt Controller output(1)
61	CIC_OUT5	CIC Interrupt Controller output(1)
62	CIC_OUT6	CIC Interrupt Controller output(1)
63	CIC_OUT7	CIC Interrupt Controller output(1)
64	TIMER_N_INTL	Local timer interrupt low
65	TIMER_N_INTH	Local timer interrupt high
66	TIMER_8_INTL	Timer interrupt low
67	TIMER_8_INTH	Timer interrupt high
68	TIMER_9_INTL	Timer interrupt low
69	TIMER_9_INTH	Timer interrupt high
70	TIMER_10_INTL	Timer interrupt low
71	TIMER_10_INTH	Timer interrupt high
72	TIMER_11_INTL	Timer interrupt low
73	TIMER_11_INTH	Timer interrupt high
74	CIC_OUT8_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
75	CIC_OUT9_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
76	CIC_OUT10_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
77	CIC_OUT11_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
78	TIMER_14_INTL	Timer interrupt low
79	TIMER_14_INTH	Timer interrupt high
80	TIMER_15_INTL	Timer interrupt low
81	TIMER_15_INTH	Timer interrupt high
82	GPIO_INT8	Local GPIO interrupt
83	GPIO_INT9	Local GPIO interrupt
84	GPIO_INT10	Local GPIO interrupt
85	GPIO_INT11	Local GPIO interrupt
86	GPIO_INT12	Local GPIO interrupt
87	IQNET_ATEVT0	IQNET Timer event
88	IQNET_ATEVT1	IQNET Timer event
89	IQNET_ATEVT2	IQNET Timer event
90	IQNET_ATEVT3	IQNET Timer event
91	IQNET_ATEVT4	IQNET Timer event
92	IQNET_ATEVT5	IQNET Timer event
93	IQNET_ATEVT6	IQNET Timer event
94	IQNET_ATEVT7	IQNET Timer event
95	CIC_OUT67_PLUS_10_MUL_N	CIC Interrupt Controller output(1)
96	INTERR	Dropped C66x CorePac interrupt event
97	EMC_IDMAERR	Invalid IDMA parameters
98	Reserved	
99	CIC_2_SPECIAL_BROADCAST	CIC Interrupt Controller output
100	EFIINT0	EFI interrupt from Side A
101	EFIINT1	EFI interrupt from Side B
102	GPIO_INT13	Local GPIO interrupt
103	GPIO_INT14	Local GPIO interrupt
104	GPIO_INT15	Local GPIO interrupt
105	IPC_GRN	Boot CFG

Table 7-22. System Event Mapping — C66x CorePac Primary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
106	GPIO_INTN	GPIO interrupt
107	CIC_OUT12_PLUS_16_MUL_N	CIC Interrupt Controller output(1)
108	CIC_OUT34	CIC Interrupt Controller output(1)
109	CIC_2_OUT13	CIC Interrupt Controller output(1)
110	MDMAERREVT	DMA internal bus error event
111	Reserved	
112	EDMACC_0_TC_AET_INT	EDMA3CC0 AET event
113	PMC_ED	Single bit error detected during DMA read
114	EDMACC_1_TC_AET_INT	EDMA3CC1 AET event
115	EDMACC_2_TC_AET_INT	EDMA3CC2 AET event
116	UMC_ED1	Corrected bit error detected
117	UMC_ED2	Uncorrected bit error detected
118	PDC_INT	Power down sleep interrupt
119	SYS_CMPA	SYS CPU MP fault event
120	PMC_CMPA	CPU memory protection fault
121	PMC_DMPA	DMA memory protection fault
122	DMC_CMPA	CPU memory protection fault
123	DMC_DMPA	DMA memory protection fault
124	UMC_CMPA	CPU memory protection fault
125	UMC_DMPA	DMA memory protection fault
126	EMC_CMPA	CPU memory protection fault
127	EMC_BUSERR	Bus error interrupt

NOTE

Event No. 0 is identical to ARM GIC interrupt ID 0.

Table 7-23 lists the ARM CorePac event inputs

Table 7-23. 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	Reserved	
3	Reserved	
4	IPC_GR8	Boot config IPCG
5	IPC_GR9	Boot config IPCG
6	Reserved	
7	Reserved	
8	SEM_INT8	Semaphore interrupt
9	SEM_INT9	Semaphore interrupt
10	Reserved	
11	Reserved	
12	SEM_ERR8	Semaphore error interrupt
13	SEM_ERR9	Semaphore error interrupt
14	Reserved	

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
15	Reserved	
16	Reserved	
17	Reserved	
18	Reserved	
19	Reserved	
20	ARM_NPMUIRQ0	ARM performance monitoring unit interrupt request
21	ARM_NPMUIRQ1	ARM performance monitoring unit interrupt request
22	Reserved	
23	Reserved	
24	ARM_NINTERRIRQ	ARM internal memory ECC error interrupt request
25	ARM_NAXIERRIRQ	ARM bus error interrupt request
26	PCIE_0_INT0	PCIe_0 legacy INTA interrupt
27	PCIE_0_INT1	PCIe_0 legacy INTB interrupt
28	PCIE_0_INT2	PCIe_0 legacy INTC interrupt
29	PCIE_0_INT3	PCIe_0 legacy INTD interrupt
30	PCIE_0_INT4	PCIe_0 MSI interrupt
31	PCIE_0_INT5	PCIe_0 MSI interrupt
32	PCIE_0_INT6	PCIe_0 MSI interrupt
33	PCIE_0_INT7	PCIe_0 MSI interrupt
34	PCIE_0_INT8	PCIe_0 MSI interrupt
35	PCIE_0_INT9	PCIe_0 MSI interrupt
36	PCIE_0_INT10	PCIe_0 MSI interrupt
37	PCIE_0_INT11	PCIe_0 MSI interrupt
38	PCIE_0_INT12	PCIe_0 error interrupt
39	PCIE_0_INT13	PCIe_0 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

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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
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	TIMER_0_INTL	Timer interrupt low
81	TIMER_0_INTH	Timer interrupt high
82	TIMER_1_INTL	Timer interrupt low
83	TIMER_1_INTH	Timer interrupt high
84	TIMER_2_INTL	Timer interrupt low
85	TIMER_2_INTH	Timer interrupt high
86	TIMER_3_INTL	Timer interrupt low
87	TIMER_3_INTH	Timer interrupt high
88	Reserved	
89	Reserved	
90	Reserved	
91	Reserved	
92	Reserved	
93	Reserved	
94	Reserved	
95	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

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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
114	TIMER_17_INTL	Timer interrupt low
115	TIMER_17_INTH	Timer interrupt high
116	Reserved	
117	Reserved	
118	Reserved	
119	Reserved	
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	GPIO_INT32	GPIO interrupt
153	GPIO_INT33	GPIO interrupt
154	GPIO_INT34	GPIO interrupt
155	GPIO_INT35	GPIO interrupt

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
156	GPIO_INT36	GPIO interrupt
157	GPIO_INT37	GPIO interrupt
158	GPIO_INT38	GPIO interrupt
159	GPIO_INT39	GPIO interrupt
160	GPIO_INT40	GPIO interrupt
161	GPIO_INT41	GPIO interrupt
162	GPIO_INT42	GPIO interrupt
163	GPIO_INT43	GPIO interrupt
164	GPIO_INT44	GPIO interrupt
165	GPIO_INT45	GPIO interrupt
166	GPIO_INT46	GPIO interrupt
167	GPIO_INT47	GPIO interrupt
168	GPIO_INT48	GPIO interrupt
169	GPIO_INT49	GPIO interrupt
170	GPIO_INT50	GPIO interrupt
171	GPIO_INT51	GPIO interrupt
172	GPIO_INT52	GPIO interrupt
173	GPIO_INT53	GPIO interrupt
174	GPIO_INT54	GPIO interrupt
175	GPIO_INT55	GPIO interrupt
176	GPIO_INT56	GPIO 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

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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
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	
228	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

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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
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

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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
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 subsystem MDIO interrupt
309	NETCP_MDIO_LINK_INT1	Packet Accelerator subsystem MDIO interrupt
310	NETCP_MDIO_USER_INT0	Packet Accelerator subsystem MDIO interrupt
311	NETCP_MDIO_USER_INT1	Packet Accelerator subsystem MDIO interrupt
312	NETCP_MISC_INT	Packet Accelerator subsystem MDIO interrupt
313	NETCP_SWITCH_INT	Packet Accelerator Packet DMA starvation interrupt
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	QMSS_QUE_PEND_637	Navigator transmit queue pending event for indicated queue
342	QMSS_QUE_PEND_638	Navigator transmit queue pending event for indicated queue
343	QMSS_QUE_PEND_639	Navigator transmit queue pending event for indicated queue

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
344	QMSS_QUE_PEND_640	Navigator transmit queue pending event for indicated queue
345	QMSS_QUE_PEND_641	Navigator transmit queue pending event for indicated queue
346	QMSS_QUE_PEND_642	Navigator transmit queue pending event for indicated queue
347	QMSS_QUE_PEND_643	Navigator transmit queue pending event for indicated queue
348	QMSS_QUE_PEND_644	Navigator transmit queue pending event for indicated queue
349	QMSS_QUE_PEND_645	Navigator transmit queue pending event for indicated queue
350	QMSS_QUE_PEND_646	Navigator transmit queue pending event for indicated queue
351	QMSS_QUE_PEND_647	Navigator transmit queue pending event for indicated queue
352	QMSS_QUE_PEND_648	Navigator transmit queue pending event for indicated queue
353	QMSS_QUE_PEND_649	Navigator transmit queue pending event for indicated queue
354	QMSS_QUE_PEND_650	Navigator transmit queue pending event for indicated queue
355	QMSS_QUE_PEND_651	Navigator transmit queue pending event for indicated queue
356	Reserved	
357	Reserved	
358	Reserved	
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	NETCP_SWITCH_STAT_INT0	NetCP Switch Status interrupt
374	NETCP_SWITCH_STAT_INT1	NetCP Switch Status interrupt
375	NETCP_SWITCH_STAT_INT2	NetCP Switch Status interrupt
376	NETCP_SWITCH_STAT_INT3	NetCP Switch Status interrupt
377	NETCP_SWITCH_STAT_INT4	NetCP Switch Status interrupt
378	NETCP_GLOBAL_STARVE_INT	NetCP Global Starve interrupt
379	NETCP_LOCAL_STARVE_INT	NetCP Local Starve interrupt
380	USB_INT05	USB event ring 05 interrupt
381	USB_INT06	USB event ring 06 interrupt
382	USB_INT07	USB event ring 07 interrupt
383	USB_INT08	USB event ring 08 interrupt
384	USB_INT09	USB event ring 09 interrupt
385	USB_INT10	USB event ring 10 interrupt
386	USB_INT11	USB event ring11 interrupt
387	Reserved	

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
388	Reserved	
389	ARM_NCTIIRQ0	ARM cross trigger (CTI) IRQ interrupt
390	ARM_NCTIIRQ1	ARM cross trigger (CTI) IRQ interrupt
391	Reserved	
392	Reserved	
393	USB_INT00	USB event ring 0 interrupt
394	USB_INT01	USB event ring 1 interrupt
395	USB_INT02	USB event ring 2 interrupt
396	USB_INT03	USB event ring 3 interrupt
397	USB_INT04	USB event ring 4 interrupt
398	USB_OABSINT	USB OABS interrupt
399	USB_MISCINT	USB miscellaneous interrupt
400	Reserved	
401	GPIO_INT57	GPIO interrupt
402	GPIO_INT58	GPIO interrupt
403	GPIO_INT59	GPIO interrupt
404	GPIO_INT60	GPIO interrupt
405	GPIO_INT61	GPIO interrupt
406	GPIO_INT62	GPIO interrupt
407	GPIO_INT63	GPIO interrupt
408	IQNET_ATEVT0	IQNET Timer event
409	IQNET_ATEVT1	IQNET Timer event
410	IQNET_ATEVT2	IQNET Timer event
411	IQNET_ATEVT3	IQNET Timer event
412	IQNET_ATEVT4	IQNET Timer event
413	IQNET_ATEVT5	IQNET Timer event
414	IQNET_ATEVT6	IQNET Timer event
415	IQNET_ATEVT7	IQNET Timer event
416	IQNET_ATEVT16	IQNET Timer event
417	IQNET_ATEVT17	IQNET Timer event
418	IQNET_ATEVT18	IQNET Timer event
419	IQNET_ATEVT19	IQNET Timer event
420	IQNET_ATEVT20	IQNET Timer event
421	IQNET_ATEVT21	IQNET Timer event
422	IQNET_ATEVT22	IQNET Timer event
423	IQNET_ATEVT23	IQNET Timer event
424	USIM_PONIRQ	USIM interrupt
425	USIM_RREQ	USIM read DMA event
426	USIM_WREQ	USIM write DMA event
427	Reserved	
428	Reserved	
429	Reserved	
430	Reserved	
431	Reserved	
432	UART_2_UARTINT	UART2 interrupt
433	UART_2_URXEVT	UART2 receive event
434	UART_2_UTXEVT	UART2 transmit event

Table 7-23. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
435	UART_3_UARTINT	UART3 interrupt
436	UART_3_URXEVT	UART3 receive event
437	UART_3_UTXEVT	UART3 transmit event
438	Reserved	
439	Reserved	
440	Reserved	
441	Reserved	
442	Reserved	
443	Reserved	
444	Reserved	
445	IQNET_PKTDMA_STARVE	IQNET Starve interrupt
446	IQNET_INT0	IQNET interrupt
447	IQNET_INT1	IQNET interrupt
448	CIC_2_OUT29	CIC2 interrupt controller output
449	CIC_2_OUT30	CIC2 interrupt controller output
450	CIC_2_OUT31	CIC2 interrupt controller output
451	CIC_2_OUT32	CIC2 interrupt controller output
452	CIC_2_OUT33	CIC2 interrupt controller output
453	CIC_2_OUT34	CIC2 interrupt controller output
454	CIC_2_OUT35	CIC2 interrupt controller output
455	CIC_2_OUT36	CIC2 interrupt controller output
456	CIC_2_OUT37	CIC2 interrupt controller output
457	CIC_2_OUT38	CIC2 interrupt controller output
458	CIC_2_OUT39	CIC2 interrupt controller output
459	CIC_2_OUT40	CIC2 interrupt controller output
460	CIC_2_OUT41	CIC2 interrupt controller output
461	CIC_2_OUT42	CIC2 interrupt controller output
462	CIC_2_OUT43	CIC2 interrupt controller output
463	CIC_2_OUT44	CIC2 interrupt controller output
464	CIC_2_OUT45	CIC2 interrupt controller output
465	CIC_2_OUT46	CIC2 interrupt controller output
466	CIC_2_OUT47	CIC2 interrupt controller output
467	CIC_2_OUT18	CIC2 interrupt controller output
468	CIC_2_OUT19	CIC2 interrupt controller output
469	CIC_2_OUT22	CIC2 interrupt controller output
470	CIC_2_OUT23	CIC2 interrupt controller output
471	CIC_2_OUT50	CIC2 interrupt controller output
472	CIC_2_OUT51	CIC2 interrupt controller output
473	CIC_2_OUT66	CIC2 interrupt controller output
474	CIC_2_OUT67	CIC2 interrupt controller output
475	CIC_2_OUT88	CIC2 interrupt controller output
476	CIC_2_OUT89	CIC2 interrupt controller output
477	CIC_2_OUT90	CIC2 interrupt controller output
478	CIC_2_OUT91	CIC2 interrupt controller output
479	CIC_2_OUT92	CIC2 interrupt controller output

Table 7-24 and Table 7-25 list the CIC0 and CIC2 event inputs.

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts

EVENT NO.	EVENT NAME	DESCRIPTION
0	EDMACC_1_ERRINT	EDMA3CC1 error interrupt
1	EDMACC_1_MPINT	EDMA3CC1 memory protection interrupt
2	EDMACC_1_TC_0_ERRINT	EDMA3CC1 TPTC0 error interrupt
3	EDMACC_1_TC_1_ERRINT	EDMA3CC1 TPTC1 error interrupt
4	EDMACC_1_TC_2_ERRINT	EDMA3CC1 TPTC2 error interrupt
5	EDMACC_1_TC_3_ERRINT	EDMA3CC1 TPTC3 error interrupt
6	EDMACC_1_GINT	EDMA3CC1 GINT
7	QMSS_QUE_PEND_637	Navigator transmit queue pending event for indicated queue
8	EDMACC_1_TC_0_INT	EDMA3CC1 individual completion interrupt
9	EDMACC_1_TC_1_INT	EDMA3CC1 individual completion interrupt
10	EDMACC_1_TC_2_INT	EDMA3CC1 individual completion interrupt
11	EDMACC_1_TC_3_INT	EDMA3CC1 individual completion interrupt
12	EDMACC_1_TC_4_INT	EDMA3CC1 individual completion interrupt
13	EDMACC_1_TC_5_INT	EDMA3CC1 individual completion interrupt
14	EDMACC_1_TC_6_INT	EDMA3CC1 individual completion interrupt
15	EDMACC_1_TC_7_INT	EDMA3CC1 individual completion interrupt
16	EDMACC_2_ERRINT	EDMA3CC2 error interrupt
17	EDMACC_2_MPINT	EDMA3CC2 memory protection interrupt
18	EDMACC_2_TC_0_ERRINT	EDMA3CC2 TPTC0 error interrupt
19	EDMACC_2_TC_1_ERRINT	EDMA3CC2 TPTC1 error interrupt
20	EDMACC_2_TC_2_ERRINT	EDMA3CC2 TPTC2 error interrupt
21	EDMACC_2_TC_3_ERRINT	EDMA3CC2 TPTC3 error interrupt
22	EDMACC_2_GINT	EDMA3CC2 GINT
23	QMSS_QUE_PEND_638	Navigator transmit queue pending event for indicated queue
24	EDMACC_2_TC_0_INT	EDMA3CC2 individual completion interrupt
25	EDMACC_2_TC_1_INT	EDMA3CC2 individual completion interrupt
26	EDMACC_2_TC_2_INT	EDMA3CC2 individual completion interrupt
27	EDMACC_2_TC_3_INT	EDMA3CC2 individual completion interrupt
28	EDMACC_2_TC_4_INT	EDMA3CC2 individual completion interrupt
29	EDMACC_2_TC_5_INT	EDMA3CC2 individual completion interrupt
30	EDMACC_2_TC_6_INT	EDMA3CC2 individual completion interrupt
31	EDMACC_2_TC_7_INT	EDMA3CC2 individual completion interrupt
32	EDMACC_0_ERRINT	EDMA3CC0 error interrupt
33	EDMACC_0_MPINT	EDMA3CC0 memory protection interrupt
34	EDMACC_0_TC_0_ERRINT	EDMA3CC0 TPTC0 error interrupt
35	EDMACC_0_TC_1_ERRINT	EDMA3CC0 TPTC1 error interrupt
36	EDMACC_0_GINT	EDMA3CC0 global completion interrupt
37	QMSS_QUE_PEND_639	Navigator transmit queue pending event for indicated queue
38	EDMACC_0_TC_0_INT	EDMA3CC0 individual completion interrupt
39	EDMACC_0_TC_1_INT	EDMA3CC0 individual completion interrupt
40	EDMACC_0_TC_2_INT	EDMA3CC0 individual completion interrupt
41	EDMACC_0_TC_3_INT	EDMA3CC0 individual completion interrupt
42	EDMACC_0_TC_4_INT	EDMA3CC0 individual completion interrupt
43	EDMACC_0_TC_5_INT	EDMA3CC0 individual completion interrupt
44	EDMACC_0_TC_6_INT	EDMA3CC0 individual completion interrupt

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
45	EDMACC_0_TC_7_INT	EDMA3CC0 individual completion interrupt
46	QMSS_QUE_PEND_640	Navigator transmit queue pending event for indicated queue
47	QMSS_QUE_PEND_652	Navigator transmit queue pending event for indicated queue
48	UART_2_UARTINT	UART_2interrupt
49	UART_2_URXEVT	UART_2 receive event
50	UART_2_UTXEVT	UART_2 transmit event
51	UART_3_UARTINT	UART_3 interrupt
52	UART_3_URXEVT	UART_3 receive event
53	UART_3_UTXEVT	UART_3 transmit event
54	SPI_0_INT0	SPI0 interrupt0
55	SPI_0_INT1	SPI0 interrupt1
56	SPI_0_XEVT	SPI0 transmit event
57	SPI_0_REVT	SPI0 receive event
58	I2C_0_INT	I2C0 interrupt
59	I2C_0_REVT	I2C0 receive event
60	I2C_0_XEVT	I2C0 transmit event
61	Reserved	
62	QMSS_QUE_PEND_641	Navigator transmit queue pending event for indicated queue
63	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
64	MPU_12_INT	MPU12 addressing violation interrupt and protection violation interrupt
65	DBGTBR_ACQCOMP	Debug trace buffer (TBR) acquisition has been completed
66	MPU_13_INT	MPU13 addressing violation interrupt and protection violation interrupt
67	MPU_14_INT	MPU14 addressing violation interrupt and protection violation interrupt
68	NETCP_MDIO_LINK_INT0	Packet Accelerator 0 subsystem MDIO interrupt
69	NETCP_MDIO_LINK_INT1	Packet Accelerator 0 subsystem MDIO interrupt
70	NETCP_MDIO_USER_INT0	Packet Accelerator 0 subsystem MDIO interrupt
71	NETCP_MDIO_USER_INT1	Packet Accelerator 0 subsystem MDIO interrupt
72	NETCP_MISC_INT	Packet Accelerator 0 subsystem misc interrupt
73	TRACER_CORE_0_INT	Tracer sliding time window interrupt for DSP0 L2
74	TRACER_CORE_1_INT	Tracer sliding time window interrupt for DSP1 L2
75	TRACER_CORE_2_INT	Tracer sliding time window interrupt for DSP2 L2
76	TRACER_CORE_3_INT	Tracer sliding time window interrupt for DSP3 L2
77	TRACER_DDR_INT	Tracer sliding time window interrupt for MSMC-DDR3A
78	TRACER_MSMC_0_INT	Tracer sliding time window interrupt for MSMC SRAM bank0
79	TRACER_MSMC_1_INT	Tracer sliding time window interrupt for MSMC SRAM bank1
80	TRACER_MSMC_2_INT	Tracer sliding time window interrupt for MSMC SRAM bank2
81	TRACER_MSMC_3_INT	Tracer sliding time window interrupt for MSMC SRAM bank3
82	TRACER_CFG_INT	Tracer sliding time window interrupt for CFG0 TeraNet
83	TRACER_QMSS_QM_CFG1_INT	Tracer sliding time window interrupt for Navigator CFG1 slave port
84	TRACER_QMSS_DMA_INT	Tracer sliding time window interrupt for Navigator DMA internal bus slave port
85	TRACER_SEM_INT	Tracer sliding time window interrupt for Semaphore
86	PSC_ALLINT	Power & Sleep Controller interrupt
87	MSMC_SCRUB_CERROR	Correctable (1-bit) soft error detected during scrub cycle
88	BOOTCFG_INT	Chip-level MMR Error Register
89	SR_0_PO_VCON_SMPSERR_INT	SmartReflex SMPS error interrupt
90	MPU_0_INT	MPU0 addressing violation interrupt and protection violation interrupt
91	QMSS_QUE_PEND_653	Navigator transmit queue pending event for indicated queue

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
92	MPU_1_INT	MPU1 addressing violation interrupt and protection violation interrupt.
93	QMSS_QUE_PEND_654	Navigator transmit queue pending event for indicated queue
94	MPU_2_INT	MPU2 addressing violation interrupt and protection violation interrupt.
95	QMSS_QUE_PEND_655	Navigator transmit queue pending event for indicated queue
96	PCIE_1_INT0	PCIe_1 MSI interrupt
97	QMSS_QUE_PEND_656	Navigator transmit queue pending event for indicated queue
98	MSMC_DEDC_CERROR	Correctable (1-bit) soft error detected on SRAM read
99	MSMC_DEDC_NC_ERROR	Non-correctable (2-bit) soft error detected on SRAM read
100	MSMC_SCRUB_NC_ERROR	Non-correctable (2-bit) soft error detected during scrub cycle
101	MSMC_MPF_ERROR4	Memory protection fault indicators for system master PrivID = 4
102	MSMC_MPF_ERROR8	Memory protection fault indicators for system master PrivID = 8
103	MSMC_MPF_ERROR9	Memory protection fault indicators for system master PrivID = 9
104	MSMC_MPF_ERROR10	Memory protection fault indicators for system master PrivID = 10
105	MSMC_MPF_ERROR11	Memory protection fault indicators for system master PrivID = 11
106	MSMC_MPF_ERROR12	Memory protection fault indicators for system master PrivID = 12
107	MSMC_MPF_ERROR13	Memory protection fault indicators for system master PrivID = 13
108	MSMC_MPF_ERROR14	Memory protection fault indicators for system master PrivID = 14
109	MSMC_MPF_ERROR15	Memory protection fault indicators for system master PrivID = 15
110	DDR3_0_ERR	DDR3A_EMIF error interrupt
111	GPIO_INT40	GPIO interrupt
112	GPIO_INT41	GPIO interrupt
113	GPIO_INT42	GPIO interrupt
114	GPIO_INT43	GPIO interrupt
115	GPIO_INT44	GPIO interrupt
116	GPIO_INT45	GPIO interrupt
117	GPIO_INT46	GPIO interrupt
118	GPIO_INT47	GPIO interrupt
119	GPIO_INT48	GPIO interrupt
120	GPIO_INT49	GPIO interrupt
121	GPIO_INT50	GPIO interrupt
122	GPIO_INT51	GPIO interrupt
123	GPIO_INT52	GPIO interrupt
124	GPIO_INT53	GPIO interrupt
125	GPIO_INT54	GPIO interrupt
126	GPIO_INT55	GPIO interrupt
127	GPIO_INT56	GPIO interrupt
128	AEMIF_EASYNCERR	Asynchronous EMIF16 error interrupt
129	TRACER_CORE_4_INT	Tracer sliding time window interrupt for DSP4 L2
130	TRACER_CORE_5_INT	Tracer sliding time window interrupt for DSP5 L2
131	TRACER_CORE_6_INT	Tracer sliding time window interrupt for DSP6 L2
132	TRACER_CORE_7_INT	Tracer sliding time window interrupt for DSP7 L2
133	QMSS_INTD_1_PKTDMA_0	Navigator interrupt for Packet DMA starvation
134	QMSS_INTD_1_PKTDMA_1	Navigator interrupt for Packet DMA starvation
135	GPIO_INT57	GPIO interrupt
136	NETCP_SWITCH_INT	Packet Accelerator0 Packet DMA starvation interrupt
137	SR_0_SMARTREFLEX_INTREQ0	SmartReflex controller interrupt
138	SR_0_SMARTREFLEX_INTREQ1	SmartReflex controller interrupt

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
139	SR_0_SMARTREFLEX_INTREQ2	SmartReflex controller interrupt
140	SR_0_SMARTREFLEX_INTREQ3	SmartReflex controller interrupt
141	SR_0_VPNOSMPSACK	SmartReflex VPVOLTUPDATE has been asserted but SMPS has not been responded to in a defined time interval
142	SR_0_VPEQVALUE	SmartReflex SRSINTERUPT is asserted, but the new voltage is not different from the current SMPS voltage
143	SR_0_VPMAXVDD	SmartReflex. The new voltage required is equal to or greater than MaxVdd
144	SR_0_VPMINVDD	SmartReflex. The new voltage required is equal to or less than MinVdd
145	SR_0_VPINIDLE	SmartReflex indicating that the FSM of voltage processor is in idle
146	SR_0_VPOPPCHANGEDONE	SmartReflex indicating that the average frequency error is within the desired limit
147	Reserved	
148	UART_0_UARTINT	UART0 interrupt
149	UART_0_URXEVT	UART0 receive event
150	UART_0_UTXEVT	UART0 transmit event
151	QMSS_QUE_PEND_657	Navigator transmit queue pending event for indicated queue
152	QMSS_QUE_PEND_658	Navigator transmit queue pending event for indicated queue
153	QMSS_QUE_PEND_659	Navigator transmit queue pending event for indicated queue
154	QMSS_QUE_PEND_660	Navigator transmit queue pending event for indicated queue
155	QMSS_QUE_PEND_661	Navigator transmit queue pending event for indicated queue
156	QMSS_QUE_PEND_662	Navigator transmit queue pending event for indicated queue
157	QMSS_QUE_PEND_663	Navigator transmit queue pending event for indicated queue
158	QMSS_QUE_PEND_664	Navigator transmit queue pending event for indicated queue
159	QMSS_QUE_PEND_665	Navigator transmit queue pending event for indicated queue
160	SR_0_VPSMPSACK	SmartReflex VPVOLTUPDATE asserted and SMPS has acknowledged in a defined time interval
161	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
162	ARM_TBR_ACQ	ARM trace buffer (TBR) acquisition has been completed
163	ARM_NINTERRIRQ	ARM internal memory ECC error interrupt request
164	ARM_NAXIERRIRQ	ARM bus error interrupt request
165	SR_0_SR_TEMPSENSOR	SmartReflex temperature threshold crossing interrupt
166	SR_0_SR_TIMERINT	SmartReflex internal timer expiration interrupt
167	IQNET_ATEVT8	IQNET timer event
168	IQNET_ATEVT9	IQNET timer event
169	IQNET_ATEVT10	IQNET timer event
170	IQNET_ATEVT11	IQNET timer event
171	IQNET_ATEVT12	IQNET timer event
172	IQNET_ATEVT13	IQNET timer event
173	IQNET_ATEVT14	IQNET timer event
174	IQNET_ATEVT15	IQNET timer event
175	QMSS_QUE_PEND_589	Navigator transmit queue pending event for indicated queue
176	QMSS_QUE_PEND_631	Navigator transmit queue pending event for indicated queue
177	QMSS_QUE_PEND_632	Navigator transmit queue pending event for indicated queue
178	QMSS_QUE_PEND_633	Navigator transmit queue pending event for indicated queue
179	QMSS_QUE_PEND_634	Navigator transmit queue pending event for indicated queue
180	QMSS_QUE_PEND_635	Navigator transmit queue pending event for indicated queue
181	QMSS_QUE_PEND_636	Navigator transmit queue pending event for indicated queue
182	QMSS_QUE_PEND_642	Navigator transmit queue pending event for indicated queue

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
183	TIMER_3_INTL	Timer interrupt low
184	TIMER_3_INTH	Timer interrupt high
185	TIMER_2_INTL	Timer interrupt low
186	TIMER_2_INTH	Timer interrupt high
187	TIMER_1_INTL	Timer interrupt low
188	TIMER_1_INTH	Timer interrupt high
189	TIMER_0_INTL	Timer interrupt low
190	TIMER_0_INTH	Timer interrupt high
191	Reserved	
192	Reserved	
193	QMSS_QUE_PEND_643	Navigator transmit queue pending event for indicated queue
194	QMSS_QUE_PEND_644	Navigator transmit queue pending event for indicated queue
195	Reserved	
196	Reserved	
197	Reserved	
198	Reserved	
199	IQNET_INT0	IQNET interrupt
200	IQNET_INT11	IQNET interrupt
201	IQNET_PKDMA_STARVE	IQNET interrupt
202	PCIE_0_INT0	PCIe_0 interrupt
203	Reserved	
204	Reserved	
205	Reserved	
206	QMSS_QUE_PEND_645	Navigator transmit queue pending event for indicated queue
207	QMSS_QUE_PEND_646	Navigator transmit queue pending event for indicated queue
208	QMSS_QUE_PEND_647	Navigator transmit queue pending event for indicated queue
209	QMSS_QUE_PEND_648	Navigator transmit queue pending event for indicated queue
210	QMSS_QUE_PEND_649	Navigator transmit queue pending event for indicated queue
211	QMSS_QUE_PEND_650	Navigator transmit queue pending event for indicated queue
212	QMSS_QUE_PEND_651	Navigator transmit queue pending event for indicated queue
213	QMSS_QUE_PEND_605	Navigator transmit queue pending event for indicated queue
214	QMSS_QUE_PEND_606	Navigator transmit queue pending event for indicated queue
215	QMSS_QUE_PEND_607	Navigator transmit queue pending event for indicated queue
216	QMSS_QUE_PEND_608	Navigator transmit queue pending event for indicated queue
217	QMSS_QUE_PEND_609	Navigator transmit queue pending event for indicated queue
218	QMSS_QUE_PEND_610	Navigator transmit queue pending event for indicated queue
219	QMSS_QUE_PEND_611	Navigator transmit queue pending event for indicated queue
220	QMSS_QUE_PEND_612	Navigator transmit queue pending event for indicated queue
221	QMSS_QUE_PEND_613	Navigator transmit queue pending event for indicated queue
222	QMSS_QUE_PEND_614	Navigator transmit queue pending event for indicated queue
223	QMSS_QUE_PEND_615	Navigator transmit queue pending event for indicated queue
224	QMSS_QUE_PEND_616	Navigator transmit queue pending event for indicated queue
225	QMSS_QUE_PEND_617	Navigator transmit queue pending event for indicated queue
226	QMSS_QUE_PEND_618	Navigator transmit queue pending event for indicated queue
227	QMSS_QUE_PEND_619	Navigator transmit queue pending event for indicated queue
228	QMSS_QUE_PEND_620	Navigator transmit queue pending event for indicated queue
229	QMSS_QUE_PEND_621	Navigator transmit queue pending event for indicated queue

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
230	QMSS_QUE_PEND_622	Navigator transmit queue pending event for indicated queue
231	QMSS_QUE_PEND_623	Navigator transmit queue pending event for indicated queue
232	QMSS_QUE_PEND_624	Navigator transmit queue pending event for indicated queue
233	UART_1_UARTINT	UART1 interrupt
234	UART_1_URXEVT	UART1 receive event
235	UART_1_UTXEVT	UART1 transmit event
236	I2C_1_INT	I2C1 interrupt
237	I2C_1_REVT	I2C1 receive event
238	I2C_1_XEVT	I2C1 transmit event
239	SPI_1_INT0	SPI1 interrupt0
240	SPI_1_INT1	SPI1 interrupt1
241	SPI_1_XEVT	SPI1 transmit event
242	SPI_1_REVT	SPI1 receive event
243	MPU_5_INT	MPU5 addressing violation interrupt and protection violation interrupt
244	MPU_8_INT	MPU8 addressing violation interrupt and protection violation interrupt
245	MPU_9_INT	MPU9 addressing violation interrupt and protection violation interrupt
246	MPU_11_INT	MPU11 addressing violation interrupt and protection violation interrupt
247	Reserved	
248	MPU_15_INT	MPU15 addressing violation interrupt and protection violation interrupt
249	MPU_7_INT	MPU7 addressing violation interrupt and protection violation interrupt
250	MPU_10_INT	MPU10 addressing violation interrupt and protection violation interrupt
251	SPI_2_INT0	SPI2 interrupt0
252	SPI_2_INT1	SPI2 interrupt1
253	SPI_2_XEVT	SPI DMA TX event
254	SPI_2_REVT	SPI DMA RX event
255	I2C_2_INT	I2C2 interrupt
256	I2C_2_REVT	I2C2 receive event
257	I2C_2_XEVT	I2C2 transmit event
258	GPIO_INT58	GPIO interrupt
259	GPIO_INT59	GPIO interrupt
260	GPIO_INT60	GPIO interrupt
261	GPIO_INT61	GPIO interrupt
262	GPIO_INT62	GPIO interrupt
263	GPIO_INT63	GPIO interrupt
264	USIM_PONIRQ	USIM interrupt
265	USIM_RREQ	USIM read DMA event
266	USIM_WREQ	USIM write DMA event
267	Reserved	
268	Reserved	
269	Reserved	
270	Reserved	
271	Reserved	
272	Reserved	
273	QMSS_QUE_PEND_625	Navigator transmit queue pending event for indicated queue
274	QMSS_QUE_PEND_626	Navigator transmit queue pending event for indicated queue
275	QMSS_QUE_PEND_627	Navigator transmit queue pending event for indicated queue
276	QMSS_QUE_PEND_628	Navigator transmit queue pending event for indicated queue

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
277	QMSS_QUE_PEND_629	Navigator transmit queue pending event for indicated queue
278	QMSS_QUE_PEND_630	Navigator transmit queue pending event for indicated queue
279	PCIE_0_INT4	PCIe_0 MSI interrupt
280	PCIE_0_INT5	PCIe_0 MSI interrupt
281	PCIE_0_INT6	PCIe_0 MSI interrupt
282	PCIE_0_INT7	PCIe_0 MSI interrupt
283	MSMC_MPF_ERROR5	Memory protection fault indicators for system master PrivID = 5
284	MSMC_MPF_ERROR6	Memory protection fault indicators for system master PrivID = 6
285	MSMC_MPF_ERROR7	Memory protection fault indicators for system master PrivID = 7
286	PCIE_1_INT4	PCIe_1 MSI interrupt
287	PCIE_1_INT5	PCIe_1 MSI interrupt
288	PCIE_1_INT6	PCIe_1 MSI interrupt
289	PCIE_1_INT7	PCIe_1 MSI interrupt
290	Reserved	
291	Reserved	
292	QMSS_QUE_PEND_666	Navigator transmit queue pending event for indicated queue
293	QMSS_QUE_PEND_667	Navigator transmit queue pending event for indicated queue
294	QMSS_QUE_PEND_668	Navigator transmit queue pending event for indicated queue
295	QMSS_QUE_PEND_669	Navigator transmit queue pending event for indicated queue
296	QMSS_QUE_PEND_670	Navigator transmit queue pending event for indicated queue
297	QMSS_QUE_PEND_671	Navigator transmit queue pending event for indicated queue
298	QMSS_QUE_PEND_672	Navigator transmit queue pending event for indicated queue
299	QMSS_QUE_PEND_673	Navigator transmit queue pending event for indicated queue
300	QMSS_QUE_PEND_674	Navigator transmit queue pending event for indicated queue
301	QMSS_QUE_PEND_675	Navigator transmit queue pending event for indicated queue
302	QMSS_QUE_PEND_676	Navigator transmit queue pending event for indicated queue
303	QMSS_QUE_PEND_677	Navigator transmit queue pending event for indicated queue
304	QMSS_QUE_PEND_678	Navigator transmit queue pending event for indicated queue
305	QMSS_QUE_PEND_679	Navigator transmit queue pending event for indicated queue
306	QMSS_QUE_PEND_680	Navigator transmit queue pending event for indicated queue
307	QMSS_QUE_PEND_681	Navigator transmit queue pending event for indicated queue
308	QMSS_QUE_PEND_682	Navigator transmit queue pending event for indicated queue
309	QMSS_QUE_PEND_683	Navigator transmit queue pending event for indicated queue
310	QMSS_QUE_PEND_684	Navigator transmit queue pending event for indicated queue
311	QMSS_QUE_PEND_685	Navigator transmit queue pending event for indicated queue
312	QMSS_QUE_PEND_686	Navigator transmit queue pending event for indicated queue
313	QMSS_QUE_PEND_687	Navigator transmit queue pending event for indicated queue
314	QMSS_QUE_PEND_590	Navigator transmit queue pending event for indicated queue
315	QMSS_QUE_PEND_591	Navigator transmit queue pending event for indicated queue
316	QMSS_QUE_PEND_592	Navigator transmit queue pending event for indicated queue
317	QMSS_QUE_PEND_593	Navigator transmit queue pending event for indicated queue
318	QMSS_INTD_2_PKTDMA_0	Navigator ECC error interrupt
319	QMSS_INTD_2_PKTDMA_1	Navigator ECC error interrupt
320	QMSS_INTD_1_LOW_0	Navigator interrupt low
321	QMSS_INTD_1_LOW_1	Navigator interrupt low
322	QMSS_INTD_1_LOW_2	Navigator interrupt low
323	QMSS_INTD_1_LOW_3	Navigator interrupt low

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
324	QMSS_INTD_1_LOW_4	Navigator interrupt low
325	QMSS_INTD_1_LOW_5	Navigator interrupt low
326	QMSS_INTD_1_LOW_6	Navigator interrupt low
327	QMSS_INTD_1_LOW_7	Navigator interrupt low
328	QMSS_INTD_1_LOW_8	Navigator interrupt low
329	QMSS_INTD_1_LOW_9	Navigator interrupt low
330	QMSS_INTD_1_LOW_10	Navigator interrupt low
331	QMSS_INTD_1_LOW_11	Navigator interrupt low
332	QMSS_INTD_1_LOW_12	Navigator interrupt low
333	QMSS_INTD_1_LOW_13	Navigator interrupt low
334	QMSS_INTD_1_LOW_14	Navigator interrupt low
335	QMSS_INTD_1_LOW_15	Navigator interrupt low
336	QMSS_INTD_2_LOW_0	Navigator second interrupt low
337	QMSS_INTD_2_LOW_1	Navigator second interrupt low
338	QMSS_INTD_2_LOW_2	Navigator second interrupt low
339	QMSS_INTD_2_LOW_3	Navigator second interrupt low
340	QMSS_INTD_2_LOW_4	Navigator second interrupt low
341	QMSS_INTD_2_LOW_5	Navigator second interrupt low
342	QMSS_INTD_2_LOW_6	Navigator second interrupt low
343	QMSS_INTD_2_LOW_7	Navigator second interrupt low
344	QMSS_INTD_2_LOW_8	Navigator second interrupt low
345	QMSS_INTD_2_LOW_9	Navigator second interrupt low
346	QMSS_INTD_2_LOW_10	Navigator second interrupt low
347	QMSS_INTD_2_LOW_11	Navigator second interrupt low
348	QMSS_INTD_2_LOW_12	Navigator second interrupt low
349	QMSS_INTD_2_LOW_13	Navigator second interrupt low
350	QMSS_INTD_2_LOW_14	Navigator second interrupt low
351	QMSS_INTD_2_LOW_15	Navigator second interrupt low
352	TRACER_EDMACC_0	Tracer sliding time window interrupt for EDMA3CC0
353	TRACER_EDMACC_12_INT	Tracer sliding time window interrupt for EDMA3CC1 and EDMA3CC2
354	TRACER_CIC_INT	Tracer sliding time window interrupt for interrupt controllers (CIC)
355	TRACER_MSMC_4_INT	Tracer sliding time window interrupt for MSMC SRAM bank4
356	TRACER_MSMC_5_INT	Tracer sliding time window interrupt for MSMC SRAM bank5
357	TRACER_MSMC_6_INT	Tracer sliding time window interrupt for MSMC SRAM bank6
358	TRACER_MSMC_7_INT	Tracer sliding time window interrupt for MSMC SRAM bank7
359	TRACER_SPI_ROM_EMIF_INT	Tracer sliding time window interrupt for SPI/ROM/EMIF16 modules
360	TRACER_QMSS_QM_CFG2_INT	Tracer sliding time window interrupt for QM2
361	Reserved	
362	TRACER_CFG_3P_U_INT	Tracer CFG_3P_U interrupt
363	Reserved	
364	PCIE_0_INT1	PCIe_0 interrupt
365	NETCP_SWITCH_STAT_INT4	NetCP interrupt
366	Reserved	
367	Reserved	
368	Reserved	
369	Reserved	
370	QMSS_QUE_PEND_594	Navigator transmit queue pending event for indicated queue

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
371	QMSS_QUE_PEND_595	Navigator transmit queue pending event for indicated queue
372	QMSS_QUE_PEND_596	Navigator transmit queue pending event for indicated queue
373	QMSS_QUE_PEND_597	Navigator transmit queue pending event for indicated queue
374	Reserved	
375	Reserved	
376	Reserved	
377	Reserved	
378	Reserved	
379	Reserved	
380	Reserved	
381	Reserved	
382	QMSS_QUE_PEND_598	Navigator transmit queue pending event for indicated queue
383	QMSS_QUE_PEND_599	Navigator transmit queue pending event for indicated queue
384	QMSS_QUE_PEND_600	Navigator transmit queue pending event for indicated queue
385	QMSS_QUE_PEND_601	Navigator transmit queue pending event for indicated queue
386	QMSS_QUE_PEND_602	Navigator transmit queue pending event for indicated queue
387	QMSS_QUE_PEND_603	Navigator transmit queue pending event for indicated queue
388	QMSS_QUE_PEND_604	Navigator transmit queue pending event for indicated queue
389	QMSS_QUE_PEND_570	Navigator transmit queue pending event for indicated queue
390	FFTC_0_INT0	FFTC interrupt
391	FFTC_0_INT1	FFTC interrupt
392	FFTC_0_INT2	FFTC interrupt
393	FFTC_0_INT3	FFTC interrupt
394	FFTC_1_INT0	FFTC interrupt
395	FFTC_1_INT1	FFTC interrupt
396	FFTC_1_INT2	FFTC interrupt
397	FFTC_1_INT3	FFTC interrupt
398	QMSS_QUE_PEND_571	Navigator transmit queue pending event for indicated queue
399	QMSS_QUE_PEND_572	Navigator transmit queue pending event for indicated queue
400	QMSS_QUE_PEND_573	Navigator transmit queue pending event for indicated queue
401	QMSS_QUE_PEND_574	Navigator transmit queue pending event for indicated queue
402	QMSS_QUE_PEND_575	Navigator transmit queue pending event for indicated queue
403	QMSS_QUE_PEND_576	Navigator transmit queue pending event for indicated queue
404	QMSS_QUE_PEND_577	Navigator transmit queue pending event for indicated queue
405	QMSS_QUE_PEND_578	Navigator transmit queue pending event for indicated queue
406	QMSS_QUE_PEND_579	Navigator transmit queue pending event for indicated queue
407	QMSS_QUE_PEND_580	Navigator transmit queue pending event for indicated queue
408	QMSS_QUE_PEND_581	Navigator transmit queue pending event for indicated queue
409	QMSS_QUE_PEND_582	Navigator transmit queue pending event for indicated queue
410	QMSS_QUE_PEND_583	Navigator transmit queue pending event for indicated queue
411	QMSS_QUE_PEND_584	Navigator transmit queue pending event for indicated queue
412	QMSS_QUE_PEND_585	Navigator transmit queue pending event for indicated queue
413	QMSS_QUE_PEND_586	Navigator transmit queue pending event for indicated queue
414	IQNET_ATEVT16	IQNET timer event
415	IQNET_ATEVT17	IQNET timer event
416	IQNET_ATEVT18	IQNET timer event
417	IQNET_ATEVT19	IQNET timer event

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
418	IQNET_ATEVT20	IQNET timer event
419	IQNET_ATEVT21	IQNET timer event
420	IQNET_ATEVT22	IQNET timer event
421	IQNET_ATEVT23	IQNET timer event
422	USB_INT00	USB interrupt
423	USB_INT04	USB interrupt
424	USB_INT05	USB interrupt
425	USB_INT06	USB interrupt
426	USB_INT07	USB interrupt
427	USB_INT08	USB interrupt
428	USB_INT09	USB interrupt
429	USB_INT10	USB interrupt
430	USB_INT11	USB interrupt
431	USB_MISCINT	USB miscellaneous interrupt
432	USB_OABSINT	USB OABS interrupt
433	TIMER_12_INTL	Timer interrupt low
434	TIMER_12_INTH	Timer interrupt high
435	TIMER_13_INTL	Timer interrupt low
436	TIMER_13_INTH	Timer interrupt high
437	PCIE_0_INT2	PCIe_0 MSI interrupt
438	PCIE_0_INT3	PCIe_0 MSI interrupt
439	TIMER_17_INTH	Timer interrupt high
440	TIMER_16_INTH	Timer interrupt high
441	TIMER_16_INTL	Timer interrupt low
442	TIMER_17_INTL	Timer interrupt low
443	QMSS_QUE_PEND_587	Navigator transmit queue pending event for indicated queue
444	PCIE_1_INT5	PCIe_1 MSI interrupt
445	PCIE_1_INT6	PCIe_11 MSI interrupt
446	GPIO_INT16	GPIO interrupt
447	GPIO_INT17	GPIO interrupt
448	GPIO_INT18	GPIO interrupt
449	GPIO_INT19	GPIO interrupt
450	GPIO_INT20	GPIO interrupt
451	GPIO_INT21	GPIO interrupt
452	GPIO_INT22	GPIO interrupt
453	GPIO_INT23	GPIO interrupt
454	GPIO_INT24	GPIO interrupt
455	GPIO_INT25	GPIO interrupt
456	GPIO_INT26	GPIO interrupt
457	GPIO_INT27	GPIO interrupt
458	GPIO_INT28	GPIO interrupt
459	GPIO_INT29	GPIO interrupt
460	GPIO_INT30	GPIO interrupt
461	GPIO_INT31	GPIO interrupt
462	GPIO_INT32	GPIO interrupt
463	GPIO_INT33	GPIO interrupt
464	GPIO_INT34	GPIO interrupt

Table 7-24. CIC0 Event Inputs — C66x CorePac Secondary Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
465	GPIO_INT35	GPIO interrupt
466	GPIO_INT36	GPIO interrupt
467	GPIO_INT37	GPIO interrupt
468	GPIO_INT38	GPIO interrupt
469	GPIO_INT39	GPIO interrupt
470	NETCP_SWITCH_STAT_INT2	NetCP interrupt
471	NETCP_SWITCH_STAT_INT3	NetCP interrupt
472	PCIE_1_INT7	PCIe_1 interrupt
473	PCIE_1_INT12	PCIe_1 interrupt
474	PCIE_1_INT13	PCIe_1 interrupt

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2)

EVENT NO.	EVENT NAME	DESCRIPTION
0	GPIO_INT8	GPIO interrupt
1	GPIO_INT9	GPIO interrupt
2	GPIO_INT10	GPIO interrupt
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	
10	Reserved	
11	TETB_FULLINT0	TETB0 is full
12	TETB_HFULLINT0	TETB0 is half full
13	TETB_ACQINT0	TETB0 acquisition has been completed
14	TETB_FULLINT1	TETB1 is full
15	TETB_HFULLINT1	TETB1 is half full
16	TETB_ACQINT1	TETB1 acquisition has been completed
17	TETB_FULLINT2	TETB2 is full
18	TETB_HFULLINT2	TETB2 is half full
19	TETB_ACQINT2	TETB2 acquisition has been completed
20	TETB_FULLINT3	TETB3 is full
21	TETB_HFULLINT3	TETB3 is half full
22	TETB_ACQINT3	TETB3 acquisition has been completed
23	Reserved	
24	QMSS_INTD_1_HIGH_16	Navigator hi interrupt
25	QMSS_INTD_1_HIGH_17	Navigator hi interrupt
26	QMSS_INTD_1_HIGH_18	Navigator hi interrupt
27	QMSS_INTD_1_HIGH_19	Navigator hi interrupt
28	QMSS_INTD_1_HIGH_20	Navigator hi interrupt
29	QMSS_INTD_1_HIGH_21	Navigator hi interrupt
30	QMSS_INTD_1_HIGH_22	Navigator hi interrupt
31	QMSS_INTD_1_HIGH_23	Navigator hi interrupt
32	QMSS_INTD_1_HIGH_24	Navigator hi interrupt
33	QMSS_INTD_1_HIGH_25	Navigator hi interrupt

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
34	QMSS_INTD_1_HIGH_26	Navigator hi interrupt
35	QMSS_INTD_1_HIGH_27	Navigator hi interrupt
36	QMSS_INTD_1_HIGH_28	Navigator hi interrupt
37	QMSS_INTD_1_HIGH_29	Navigator hi interrupt
38	QMSS_INTD_1_HIGH_30	Navigator hi interrupt
39	QMSS_INTD_1_HIGH_31	Navigator hi interrupt
40	NETCP_MDIO_LINK_INT0	Packet Accelerator 0 subsystem MDIO interrupt
41	NETCP_MDIO_LINK_INT1	Packet Accelerator 0 subsystem MDIO interrupt
42	NETCP_MDIO_USER_INT0	Packet Accelerator 0 subsystem MDIO interrupt
43	NETCP_MDIO_USER_INT1	Packet Accelerator 0 subsystem MDIO interrupt
44	NETCP_MISC_INT	Packet Accelerator 0 subsystem MDIO interrupt
45	TRACER_CORE_0_INT	Tracer sliding time window interrupt for DSP0 L2
46	TRACER_CORE_1_INT	Tracer sliding time window interrupt for DSP1 L2
47	TRACER_CORE_2_INT	Tracer sliding time window interrupt for DSP2 L2
48	TRACER_CORE_3_INT	Tracer sliding time window interrupt for DSP3 L2
49	TRACER_DDR_INT	Tracer sliding time window interrupt for MSMC-DDR3A
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 DMA internal bus slave port
57	TRACER_SEM_INT	Tracer sliding time window interrupt for Semaphore interrupt
58	SEM_ERR0	Semaphore error interrupt
59	SEM_ERR1	Semaphore error interrupt
60	SEM_ERR2	Semaphore error interrupt
61	SEM_ERR3	Semaphore error interrupt
62	BOOTCFG_INT	BOOTCFG error interrupt
63	NETCP_GLOBAL_STARVE_INT	Packet Accelerator Packet DMA starvation interrupt
64	MPU_0_INT	MPU0 interrupt
65	MSMC_SCRUB_CERROR	MSMC error interrupt
66	MPU_1_INT	MPU1 interrupt
67	NETCP_LOCAL_STARVE_INT	Packet Accelerator Packet DMA starvation interrupt
68	MPU_2_INT	MPU2 interrupt
69	QMSS_INTD_1_PKTDMA_0	Navigator Packet DMA interrupt
70	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	
76	MSMC_MPF_ERROR0	Memory protection fault indicators for system master PrivID = 0
77	MSMC_MPF_ERROR1	Memory protection fault indicators for system master PrivID = 1
78	MSMC_MPF_ERROR2	Memory protection fault indicators for system master PrivID = 2
79	MSMC_MPF_ERROR3	Memory protection fault indicators for system master PrivID = 3
80	MSMC_MPF_ERROR4	Memory protection fault indicators for system master PrivID = 4

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
81	MSMC_MPF_ERROR5	Memory protection fault indicators for system master PrivID = 5
82	MSMC_MPF_ERROR6	Memory protection fault indicators for system master PrivID = 6
83	MSMC_MPF_ERROR7	Memory protection fault indicators for system master PrivID = 7
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	
93	NETCP_PA_ECC_INT	Packet Accelerator ECC interrupt
94	NETCP_SA_ECC_INT	Security Accelerator ECC interrupt
95	NETCP_SWITCH_ECC_INT	Packet Accelerator Switch ECC interrupt
96	QMSS_ECC_INT	Navigator ECC interrupt
97	Reserved	
98	Reserved	
99	Reserved	
100	QMSS_QUE_PEND_637	Navigator transmit queue pending event for indicated queue
101	QMSS_QUE_PEND_638	Navigator transmit queue pending event for indicated queue
102	QMSS_QUE_PEND_639	Navigator transmit queue pending event for indicated queue
103	QMSS_QUE_PEND_640	Navigator transmit queue pending event for indicated queue
104	QMSS_QUE_PEND_641	Navigator transmit queue pending event for indicated queue
105	QMSS_QUE_PEND_642	Navigator transmit queue pending event for indicated queue
106	QMSS_QUE_PEND_643	Navigator transmit queue pending event for indicated queue
107	QMSS_QUE_PEND_644	Navigator transmit queue pending event for indicated queue
108	QMSS_QUE_PEND_645	Navigator transmit queue pending event for indicated queue
109	QMSS_QUE_PEND_646	Navigator transmit queue pending event for indicated queue
110	QMSS_QUE_PEND_647	Navigator transmit queue pending event for indicated queue
111	QMSS_QUE_PEND_648	Navigator transmit queue pending event for indicated queue
112	QMSS_QUE_PEND_649	Navigator transmit queue pending event for indicated queue
113	QMSS_QUE_PEND_650	Navigator transmit queue pending event for indicated queue
114	QMSS_QUE_PEND_651	Navigator transmit queue pending event for indicated queue
115	QMSS_QUE_PEND_605	Navigator transmit queue pending event for indicated queue
116	QMSS_QUE_PEND_606	Navigator transmit queue pending event for indicated queue
117	AEMIF_EASYNCERR	Asynchronous EMIF16 error interrupt
118	Reserved	
119	Reserved	
120	Reserved	
121	Reserved	
122	Reserved	
123	Reserved	
124	Reserved	
125	Reserved	
126	Reserved	
127	Reserved	

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
128	Reserved	
129	Reserved	
130	Reserved	
131	Reserved	
132	Reserved	
133	Reserved	
134	Reserved	
135	Reserved	
136	Reserved	
137	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
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

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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	QMSS_QUE_PEND_607	Navigator transmit queue pending event for indicated queue
190	QMSS_QUE_PEND_608	Navigator transmit queue pending event for indicated queue
191	QMSS_QUE_PEND_609	Navigator transmit queue pending event for indicated queue
192	QMSS_QUE_PEND_610	Navigator transmit queue pending event for indicated queue
193	QMSS_QUE_PEND_611	Navigator transmit queue pending event for indicated queue
194	QMSS_QUE_PEND_612	Navigator transmit queue pending event for indicated queue
195	QMSS_QUE_PEND_613	Navigator transmit queue pending event for indicated queue
196	QMSS_QUE_PEND_614	Navigator transmit queue pending event for indicated queue
197	QMSS_QUE_PEND_615	Navigator transmit queue pending event for indicated queue
198	QMSS_QUE_PEND_616	Navigator transmit queue pending event for indicated queue
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	
204	MPU_5_INT	MPU5 addressing violation interrupt and protection violation interrupt
205	MPU_6_INT	MPU6 addressing violation interrupt and protection violation interrupt
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	
209	Reserved	
210	SR_0_VPSMPSACK	SmartReflex VPVOLTUPDATE asserted and SMPS has acknowledged in a defined time interval
211	DDR3_0_ERR	DDR3A error interrupt
212	Reserved	
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

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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	QMSS_QUE_PEND_617	Navigator transmit queue pending event for indicated queue
230	QMSS_QUE_PEND_618	Navigator transmit queue pending event for indicated queue
231	QMSS_QUE_PEND_619	Navigator transmit queue pending event for indicated queue
232	QMSS_QUE_PEND_620	Navigator transmit queue pending event for indicated queue
233	QMSS_QUE_PEND_621	Navigator transmit queue pending event for indicated queue
234	QMSS_QUE_PEND_622	Navigator transmit queue pending event for indicated queue
235	QMSS_QUE_PEND_623	Navigator transmit queue pending event for indicated queue
236	QMSS_QUE_PEND_624	Navigator transmit queue pending event for indicated queue
237	QMSS_QUE_PEND_652	Navigator transmit queue pending event for indicated queue
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

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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_625	Navigator transmit queue pending event for indicated queue
274	QMSS_QUE_PEND_626	Navigator transmit queue pending event for indicated queue
275	QMSS_QUE_PEND_627	Navigator transmit queue pending event for indicated queue
276	QMSS_QUE_PEND_628	Navigator transmit queue pending event for indicated queue
277	Reserved	
278	Reserved	
279	Reserved	
280	Reserved	
281	Reserved	
282	Reserved	
283	SEM_INT0	Semaphore interrupt
284	SEM_INT1	Semaphore interrupt
285	SEM_INT2	Semaphore interrupt
286	SEM_INT3	Semaphore interrupt
287	Reserved	
288	Reserved	
289	Reserved	
290	Reserved	
291	SEM_INT8	Semaphore interrupt
292	SEM_INT9	Semaphore interrupt
293	Reserved	
294	Reserved	
295	Reserved	
296	Reserved	
297	Reserved	
298	Reserved	
299	SEM_ERR8	Semaphore error interrupt
300	SEM_ERR9	Semaphore error interrupt
301	Reserved	
302	Reserved	
303	Reserved	
304	Reserved	
305	Reserved	
306	Reserved	
307	Reserved	
308	Reserved	
309	FFTC_0_INT0	FFTC interrupt
310	FFTC_0_INT1	FFTC interrupt
311	FFTC_0_INT2	FFTC interrupt
312	FFTC_0_INT3	FFTC interrupt
313	FFTC_1_INT0	FFTC interrupt
314	FFTC_1_INT1	FFTC interrupt

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
315	FFTC_1_INT2	FFTC interrupt
316	FFTC_1_INT3	FFTC interrupt
317	QMSS_QUE_PEND_629	Navigator transmit queue pending event for indicated queue
318	QMSS_QUE_PEND_630	Navigator transmit queue pending event for indicated queue
319	QMSS_QUE_PEND_631	Navigator transmit queue pending event for indicated queue
320	QMSS_QUE_PEND_632	Navigator transmit queue pending event for indicated queue
321	QMSS_QUE_PEND_633	Navigator transmit queue pending event for indicated queue
322	QMSS_QUE_PEND_634	Navigator transmit queue pending event for indicated queue
323	QMSS_QUE_PEND_635	Navigator transmit queue pending event for indicated queue
324	QMSS_QUE_PEND_636	Navigator transmit queue pending event for indicated queue
325	QMSS_QUE_PEND_589	Navigator transmit queue pending event for indicated queue
326	IQNET_INT0	IQNET interrupt
327	IQNET_INT1	IQNET interrupt
328	IQNET_PKDMA_STARVE_INT1	IQNET interrupt
329	Reserved	
330	IQNET_ATEVT20	IQNET timer event
331	IQNET_ATEVT21	IQNET timer event
332	IQNET_ATEVT22	IQNET timer event
333	IQNET_ATEVT23	IQNET timer event
334	IQNET_ATEVT0	IQNET timer event
335	IQNET_ATEVT1	IQNET timer event
336	IQNET_ATEVT2	IQNET timer event
337	IQNET_ATEVT3	IQNET timer event
338	IQNET_ATEVT4	IQNET timer event
339	IQNET_ATEVT5	IQNET timer event
340	IQNET_ATEVT6	IQNET timer event
341	IQNET_ATEVT7	IQNET timer event
342	IQNET_ATEVT8	IQNET timer event
343	IQNET_ATEVT9	IQNET timer event
344	IQNET_ATEVT10	IQNET timer event
345	IQNET_ATEVT11	IQNET timer event
346	IQNET_ATEVT12	IQNET timer event
347	IQNET_ATEVT13	IQNET timer event
348	IQNET_ATEVT14	IQNET timer event
349	IQNET_ATEVT15	IQNET timer event
350	IQNET_ATEVT16	IQNET timer event
351	IQNET_ATEVT17	IQNET timer event
352	IQNET_ATEVT18	IQNET timer event
353	IQNET_ATEVT19	IQNET timer event
354	Reserved	
355	Reserved	
356	Reserved	
357	Reserved	
358	Reserved	
359	Reserved	
360	Reserved	
361	Reserved	

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
362	PSC_ALLINT	PSC interrupt
363	Reserved	
364	Reserved	
365	Reserved	
366	Reserved	
367	Reserved	
368	Reserved	
369	Reserved	
370	ARM_NCNTVIRQ1	ARM NCNTVIRQ1
371	ARM_NCNTVIRQ0	ARM NCNTVIRQ0
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
379	Reserved	
380	Reserved	
381	Reserved	
382	CFG_3P_U_INT	CFG 3P_U interrupt
383	Reserved	
384	TRACER_SPI_ROM_EMIF_INT	Tracer sliding time window interrupt for SPI/ROM/EMIF16 modules
385	Reserved	
386	Reserved	
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_INT00	USB interrupt
400	USB_INT04	USB interrupt
401	USB_INT05	USB interrupt
402	USB_INT06	USB interrupt
403	USB_INT07	USB interrupt
404	USB_INT08	USB interrupt
405	USB_INT09	USB interrupt
406	USB_INT10	USB interrupt
407	USB_INT11	USB interrupt
408	USB_MISCINT	USB miscellaneous interrupt

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
409	USB_OABSINT	USB OABS interrupt
410	Reserved	
411	Reserved	
412	Reserved	
413	Reserved	
414	QMSS_QUE_PEND_590	Navigator transmit queue pending event for indicated queue
415	QMSS_QUE_PEND_591	Navigator transmit queue pending event for indicated queue
416	QMSS_QUE_PEND_592	Navigator transmit queue pending event for indicated queue
417	QMSS_QUE_PEND_593	Navigator transmit queue pending event for indicated queue
418	Reserved	
419	Reserved	
420	Reserved	
421	Reserved	
422	QMSS_QUE_PEND_594	Navigator transmit queue pending event for indicated queue
423	QMSS_QUE_PEND_595	Navigator transmit queue pending event for indicated queue
424	QMSS_QUE_PEND_596	Navigator transmit queue pending event for indicated queue
425	QMSS_QUE_PEND_597	Navigator transmit queue pending event for indicated queue
426	Reserved	
427	Reserved	
428	Reserved	
429	Reserved	
430	Reserved	
431	Reserved	
432	Reserved	
433	Reserved	
434	I2C_0_REVT	I2C0 receive
435	I2C_0_XEVT	I2C0 transmit
436	I2C_1_REVT	I2C1 receive
437	I2C_1_XEVT	I2C1 transmit
438	I2C_2_REVT	I2C2 receive
439	I2C_2_XEVT	I2C2 transmit
440	QMSS_QUE_PEND_598	Navigator transmit queue pending event for indicated queue
441	QMSS_QUE_PEND_599	Navigator transmit queue pending event for indicated queue
442	TETB_OVFLINT0	ETB0 overflow (emulation trace buffer)
443	TETB_UNFLINT0	ETB0 underflow
444	TETB_OVFLINT1	ETB1 overflow (emulation trace buffer)
445	TETB_UNFLINT1	ETB1 underflow
446	TETB_OVFLINT2	ETB2 overflow (emulation trace buffer)
447	TETB_UNFLINT2	ETB2 underflow
448	TETB_OVFLINT3	ETB3 overflow (emulation trace buffer)
449	TETB_UNFLINT3	ETB3 underflow
450	QMSS_QUE_PEND_600	Navigator transmit queue pending event for indicated queue
451	QMSS_QUE_PEND_601	Navigator transmit queue pending event for indicated queue
452	QMSS_QUE_PEND_602	Navigator transmit queue pending event for indicated queue
453	QMSS_QUE_PEND_603	Navigator transmit queue pending event for indicated queue
454	QMSS_QUE_PEND_604	Navigator transmit queue pending event for indicated queue
455	USB_INT01	USB interrupt

Table 7-25. CIC2 Event Inputs (Secondary Events for EDMA3CC0, EDMACC1 and EDMA3CC2) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
456	ARM_NCNTPNSIRQ1	ARM NCNTPNSIRQ1
457	ARM_NCNTPNSIRQ0	ARM NCNTPNSIRQ0
458	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
459	Reserved	
460	Reserved	
461	USB_INT02	USB interrupt
462	USB_INT03	USB interrupt
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	IPC_GR0	IPC interrupt generation
472	IPC_GR1	IPC interrupt generation
473	IPC_GR2	IPC interrupt generation
474	IPC_GR3	IPC interrupt generation
475	Reserved	
476	Reserved	
477	Reserved	
478	Reserved	

7.3.2 CIC Registers

This section includes the CIC memory map information and registers.

7.3.2.1 CIC0 Register Map

Table 7-26. CIC0 Registers

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x0	REVISION_REG	Revision Register
0x4	CONTROL_REG	Control Register
0xC	HOST_CONTROL_REG	Host Control 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

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
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
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

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
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
0x440	CH_MAP_REG16	Interrupt Channel Map Register for 64 to 64+3
0x444	CH_MAP_REG17	Interrupt Channel Map Register for 68 to 68+3
0x448	CH_MAP_REG18	Interrupt Channel Map Register for 72 to 72+3
0x44C	CH_MAP_REG19	Interrupt Channel Map Register for 76 to 76+3
0x450	CH_MAP_REG20	Interrupt Channel Map Register for 80 to 80+3
0x454	CH_MAP_REG21	Interrupt Channel Map Register for 84 to 84+3
0x458	CH_MAP_REG22	Interrupt Channel Map Register for 88 to 88+3
0x45C	CH_MAP_REG23	Interrupt Channel Map Register for 92 to 92+3
0x460	CH_MAP_REG24	Interrupt Channel Map Register for 96 to 96+3
0x464	CH_MAP_REG25	Interrupt Channel Map Register for 100 to 100+3
0x468	CH_MAP_REG26	Interrupt Channel Map Register for 104 to 104+3
0x46C	CH_MAP_REG27	Interrupt Channel Map Register for 108 to 108+3
0x470	CH_MAP_REG28	Interrupt Channel Map Register for 112 to 112+3
0x474	CH_MAP_REG29	Interrupt Channel Map Register for 116 to 116+3
0x478	CH_MAP_REG30	Interrupt Channel Map Register for 120 to 120+3
0x47C	CH_MAP_REG31	Interrupt Channel Map Register for 124 to 124+3
0x480	CH_MAP_REG32	Interrupt Channel Map Register for 128 to 128+3
0x484	CH_MAP_REG33	Interrupt Channel Map Register for 132 to 132+3
0x488	CH_MAP_REG34	Interrupt Channel Map Register for 136 to 136+3

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x48C	CH_MAP_REG35	Interrupt Channel Map Register for 140 to 140+3
0x490	CH_MAP_REG36	Interrupt Channel Map Register for 144 to 144+3
0x494	CH_MAP_REG37	Interrupt Channel Map Register for 148 to 148+3
0x498	CH_MAP_REG38	Interrupt Channel Map Register for 152 to 152+3
0x49C	CH_MAP_REG39	Interrupt Channel Map Register for 156 to 156+3
0x4a0	CH_MAP_REG40	Interrupt Channel Map Register for 160 to 160+3
0x4a4	CH_MAP_REG41	Interrupt Channel Map Register for 164 to 164+3
0x4a8	CH_MAP_REG42	Interrupt Channel Map Register for 168 to 168+3
0x4AC	CH_MAP_REG43	Interrupt Channel Map Register for 172 to 172+3
0x4b0	CH_MAP_REG44	Interrupt Channel Map Register for 176 to 176+3
0x4b4	CH_MAP_REG45	Interrupt Channel Map Register for 180 to 180+3
0x4b8	CH_MAP_REG46	Interrupt Channel Map Register for 184 to 184+3
0x4BC	CH_MAP_REG47	Interrupt Channel Map Register for 188 to 188+3
0x4C0	CH_MAP_REG48	Interrupt Channel Map Register for 192 to 192+3
0x4C4	CH_MAP_REG49	Interrupt Channel Map Register for 196 to 196+3
0x4C8	CH_MAP_REG50	Interrupt Channel Map Register for 200 to 200+3
0x4CC	CH_MAP_REG51	Interrupt Channel Map Register for 204 to 204+3
0X4D0	CH_MAP_REG52	Interrupt Channel Map Register for 208 to 208+3
0X4D4	CH_MAP_REG53	Interrupt Channel Map Register for 212 to 212+3
0X4D8	CH_MAP_REG54	Interrupt Channel Map Register for 216 to 216+3
0X4DC	CH_MAP_REG55	Interrupt Channel Map Register for 220 to 220+3
0X4E0	CH_MAP_REG56	Interrupt Channel Map Register for 224 to 224+3
0X4E4	CH_MAP_REG57	Interrupt Channel Map Register for 228 to 228+3
0X4E8	CH_MAP_REG58	Interrupt Channel Map Register for 232 to 232+3
0X4FC	CH_MAP_REG59	Interrupt Channel Map Register for 236 to 236+3
0X4F0	CH_MAP_REG60	Interrupt Channel Map Register for 240 to 240+3
0X4F4	CH_MAP_REG61	Interrupt Channel Map Register for 244 to 244+3
0X4F8	CH_MAP_REG62	Interrupt Channel Map Register for 248 to 248+3
0X4FC	CH_MAP_REG63	Interrupt Channel Map Register for 252 to 252+3
0X500	CH_MAP_REG64	Interrupt Channel Map Register for 256 to 256+3
0X504	CH_MAP_REG65	Interrupt Channel Map Register for 260 to 260+3
0X508	CH_MAP_REG66	Interrupt Channel Map Register for 264 to 264+3
0X50C	CH_MAP_REG67	Interrupt Channel Map Register for 268 to 268+3
0X510	CH_MAP_REG68	Interrupt Channel Map Register for 272 to 272+3
0X514	CH_MAP_REG69	Interrupt Channel Map Register for 276 to 276+3
0X518	CH_MAP_REG70	Interrupt Channel Map Register for 280 to 280+3
0X51C	CH_MAP_REG71	Interrupt Channel Map Register for 284 to 284+3
0X520	CH_MAP_REG72	Interrupt Channel Map Register for 288 to 288+3
0X524	CH_MAP_REG73	Interrupt Channel Map Register for 292 to 292+3
0X528	CH_MAP_REG74	Interrupt Channel Map Register for 296 to 296+3
0X52C	CH_MAP_REG75	Interrupt Channel Map Register for 300 to 300+3
0X520	CH_MAP_REG76	Interrupt Channel Map Register for 304 to 304+3
0X524	CH_MAP_REG77	Interrupt Channel Map Register for 308 to 308+3
0X528	CH_MAP_REG78	Interrupt Channel Map Register for 312 to 312+3
0x52C	CH_MAP_REG79	Interrupt Channel Map Register for 316 to 316+3
0x530	CH_MAP_REG80	Interrupt Channel Map Register for 320 to 320+3
0x534	CH_MAP_REG81	Interrupt Channel Map Register for 324 to 324+3

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x538	CH_MAP_REG82	Interrupt Channel Map Register for 328 to 328+3
0x53C	CH_MAP_REG83	Interrupt Channel Map Register for 332 to332+3
0x540	CH_MAP_REG84	Interrupt Channel Map Register for336 to 336+3
0x544	CH_MAP_REG85	Interrupt Channel Map Register for 340 to 340+3
0x548	CH_MAP_REG86	Interrupt Channel Map Register for 344 to 344+3
0x54C	CH_MAP_REG87	Interrupt Channel Map Register for 348 to 348+3
0x550	CH_MAP_REG88	Interrupt Channel Map Register for 352 to 352+3
0x554	CH_MAP_REG89	Interrupt Channel Map Register for 356 to 356+3
0x558	CH_MAP_REG90	Interrupt Channel Map Register for 360 to 360+3
0x55C	CH_MAP_REG91	Interrupt Channel Map Register for 364 to 364+3
0x560	CH_MAP_REG92	Interrupt Channel Map Register for 368 to 368+3
0x564	CH_MAP_REG93	Interrupt Channel Map Register for372 to 372+3
0x568	CH_MAP_REG94	Interrupt Channel Map Register for 376 to 376+3
0x56C	CH_MAP_REG95	Interrupt Channel Map Register for 380 to 380+3
0x570	CH_MAP_REG96	Interrupt Channel Map Register for 384 to 384+3
0x574	CH_MAP_REG97	Interrupt Channel Map Register for 388 to 388+3
0x578	CH_MAP_REG98	Interrupt Channel Map Register for 392 to 392+3
0x57C	CH_MAP_REG99	Interrupt Channel Map Register for 396 to 396+3
0x580	CH_MAP_REG100	Interrupt Channel Map Register for 400 to 400+3
0x584	CH_MAP_REG101	Interrupt Channel Map Register for 404 to 404+3
0x588	CH_MAP_REG102	Interrupt Channel Map Register for 408 to 408+3
0x58C	CH_MAP_REG103	Interrupt Channel Map Register for 412 to412+3
0x590	CH_MAP_REG104	Interrupt Channel Map Register for 416 to 416+3
0x594	CH_MAP_REG105	Interrupt Channel Map Register for 420 to 420+3
0x598	CH_MAP_REG106	Interrupt Channel Map Register for 424 to 424+3
0x59C	CH_MAP_REG107	Interrupt Channel Map Register for 428 to 428+3
0x5A0	CH_MAP_REG108	Interrupt Channel Map Register for 432 to 432+3
0x5A4	CH_MAP_REG109	Interrupt Channel Map Register for 436 to 436+3
0x5A8	CH_MAP_REG110	Interrupt Channel Map Register for 440 to 440+3
0x5AC	CH_MAP_REG111	Interrupt Channel Map Register for 444 to 444+3
0x5B0	CH_MAP_REG112	Interrupt Channel Map Register for 448 to 448+3
0x5B4	CH_MAP_REG113	Interrupt Channel Map Register for 452 to 452+3
0x5B8	CH_MAP_REG114	Interrupt Channel Map Register for 456 to 456+3
0x5BC	CH_MAP_REG115	Interrupt Channel Map Register for 460 to 460+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
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

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
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
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

Table 7-26. CIC0 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
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 64 to 64+3
0x844	HINT_MAP_REG17	Host Interrupt Map Register for 68 to 68+3
0x848	HINT_MAP_REG18	Host Interrupt Map Register for 72 to 72+3
0x84C	HINT_MAP_REG19	Host Interrupt Map Register for 76 to 76+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

7.3.2.2 CIC2 Register Map

Table 7-27. 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 7-27. 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 7-27. 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 7-27. 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

7.3.3 Inter-Processor Register Map

Table 7-28. IPC Generation Registers (IPCGRx)

ADDRESS START	ADDRESS END	SIZE	REGISTER NAME	DESCRIPTION
0x02620200	0x02620203	4B	NMIGR0	NMI Event Generation Register for C66x CorePac0
0x02620204	0x02620207	4B	NMIGR1	NMI Event Generation Register for C66x CorePac1
0x02620208	0x0262020B	4B	NMIGR2	NMI Event Generation Register for C66x CorePac2
0x0262020C	0x0262020F	4B	NMIGR3	NMI Event Generation Register for C66x CorePac3
0x02620210	0x02620213	4B	Reserved	
0x02620214	0x02620217	4B	Reserved	
0x02620218	0x0262021B	4B	Reserved	

Table 7-28. IPC Generation Registers (IPCGRx) (continued)

ADDRESS START	ADDRESS END	SIZE	REGISTER NAME	DESCRIPTION
0x0262021C	0x0262021F	4B	Reserved	
0x02620220	0x0262023F	32B	Reserved	
0x02620240	0x02620243	4B	IPCGR0	IPC Generation Register for C66x CorePac0
0x02620244	0x02620247	4B	IPCGR1	IPC Generation Register for C66x CorePac1
0x02620248	0x0262024B	4B	IPCGR2	IPC Generation Register for C66x CorePac2
0x0262024C	0x0262024F	4B	IPCGR3	IPC Generation Register for C66x CorePac3
0x02620250	0x02620253	4B	Reserved	
0x02620254	0x02620257	4B	Reserved	
0x02620258	0x0262025B	4B	Reserved	
0x0262025C	0x0262025F	4B	Reserved	
0x02620260	0x02620263	4B	IPCGR8	IPC Generation Register for ARM CorePac0
0x02620264	0x02620267	4B	IPCGR9	IPC Generation Register for ARM CorePac1
0x02620268	0x0262026B	4B	Reserved	
0x0262026C	0x0262026F	4B	Reserved	
0x02620270	0x0262027B	12B	Reserved	
0x0262027C	0x0262027F	4B	IPCGRH	IPC Generation Register for Host
0x02620280	0x02620283	4B	IPCAR0	IPC Acknowledgment Register for C66x CorePac0
0x02620284	0x02620287	4B	IPCAR1	IPC Acknowledgment Register for C66x CorePac1
0x02620288	0x0262028B	4B	IPCAR2	IPC Acknowledgment Register for C66x CorePac2
0x0262028C	0x0262028F	4B	IPCAR3	IPC Acknowledgment Register for C66x CorePac3
0x02620290	0x02620293	4B	Reserved	
0x02620294	0x02620297	4B	Reserved	
0x02620298	0x0262029B	4B	Reserved	
0x0262029C	0x0262029F	4B	Reserved	
0x026202A0	0x026202A3	4B	IPCAR8	IPC Acknowledgment Register for ARM CorePac0
0x026202A4	0x026202A7	4B	IPCAR9	IPC Acknowledgment Register for ARM CorePac1
0x026202A8	0x026202AB	4B	Reserved	
0x026202AC	0x026202AF	4B	Reserved	
0x026202B0	0x026202BB	12B	Reserved	
0x026202A0	0x026202BB	28B	Reserved	
0x026202BC	0x026202BF	4B	IPCARH	IPC Acknowledgement Register for host

7.3.4 $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$

The Non-Maskable Interrupts ($\overline{\text{NMI}}$) can be generated by chip-level registers and the $\overline{\text{LRESET}}$ can be generated by software writing into LPSC registers. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ can also be asserted by device pins or watchdog timers. One $\overline{\text{NMI}}$ pin and one $\overline{\text{LRESET}}$ pin are shared by all eight C66x CorePacs on the device. The CORESEL[3:0] pins can be configured to select between the eight C66x CorePacs available as shown in Table 7-29.

Table 7-29. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ Decoding

CORESEL[3:0] PIN INPUT	$\overline{\text{LRESET}}$ PIN INPUT	$\overline{\text{NMI}}$ PIN INPUT	$\overline{\text{LRESETNMIEN}}$ PIN INPUT	RESET MUX BLOCK OUTPUT
XXXX	X	X	1	No local reset or $\overline{\text{NMI}}$ assertion
0000	0	X	0	Assert local reset to C66x CorePac0
0001	0	X	0	Assert local reset to C66x CorePac1
0010	0	X	0	Assert local reset to C66x CorePac2
0011	0	X	0	Assert local reset to C66x CorePac3
1XXX	0	X	0	Assert local reset to all C66x CorePacs

Table 7-29. $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ Decoding (continued)

CORESEL[3:0] PIN INPUT	$\overline{\text{LRESET}}$ PIN INPUT	$\overline{\text{NMI}}$ PIN INPUT	$\overline{\text{LRESETNMIEN}}$ PIN INPUT	RESET MUX BLOCK OUTPUT
0000	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to C66x CorePac0
0001	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to C66x CorePac1
0010	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to C66x CorePac2
0011	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to C66x CorePac3
1XXX	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to all C66x CorePacs
0000	1	0	0	Assert $\overline{\text{NMI}}$ to C66x CorePac0
0001	1	0	0	Assert $\overline{\text{NMI}}$ to C66x CorePac1
0010	1	0	0	Assert $\overline{\text{NMI}}$ to C66x CorePac2
0011	1	0	0	Assert $\overline{\text{NMI}}$ to C66x CorePac3
1XXX	1	0	0	Assert $\overline{\text{NMI}}$ to all C66x CorePacs

7.4 Enhanced Direct Memory Access (EDMA3) Controller for 66AK2L06

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 C66x DSP CorePac or the ARM CorePac.

There are 3 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

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 8.2 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 DDR3A 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

7.4.1 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 7.1. 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).

7.4.2 EDMA3 Channel Controller Configuration

Table 7-30 shows the configuration for each of the EDMA3 channel controllers present on the device.

Table 7-30. EDMA3 Channel Controller Configuration

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

7.4.3 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 7-31 shows the configuration of each of the EDMA3 transfer controllers present on the device.

Table 7-31. EDMA3 Transfer Controller Configuration

	EDMA3 CC0		EDMA3 CC1				EDMA3 CC2			
PARAMETER	**TC0**	**TC1**	**TC0**	**TC1**	**TC2**	**TC3**	**TC0**	**TC1**	**TC2**	**TC3**
FIFOSIZE	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
DSTREGDEPTH	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

7.4.4 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 66AK2L06, 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 7-32. EDMA3CC0 Events for 66AK2L06

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

Table 7-32. EDMA3CC0 Events for 66AK2L06 (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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	QMSS_QUE_PEND_560	Navigator transmit queue pending event for indicated queue
25	QMSS_QUE_PEND_561	Navigator transmit queue pending event for indicated queue
26	QMSS_QUE_PEND_562	Navigator transmit queue pending event for indicated queue
27	QMSS_QUE_PEND_563	Navigator transmit queue pending event for indicated queue
28	QMSS_QUE_PEND_564	Navigator transmit queue pending event for indicated queue
29	QMSS_QUE_PEND_565	Navigator transmit queue pending event for indicated queue
30	PCIE_0_INT8	PCIe_0 interrupt
31	PCIE_0_INT9	PCIe_0 interrupt
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
37	GPIO_INT5	GPIO interrupt
38	GPIO_INT6	GPIO interrupt
39	GPIO_INT7	GPIO interrupt
40	TIMER_0_INTL	Timer interrupt low
41	TIMER_0_INTH	Timer interrupt high
42	TIMER_1_INTL	Timer interrupt low
43	TIMER_1_INTH	Timer interrupt high
44	TIMER_2_INTL	Timer interrupt low
45	TIMER_2_INTH	Timer interrupt high
46	TIMER_3_INTL	Timer interrupt low
47	TIMER_3_INTH	Timer interrupt high
48	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
49	QMSS_QUE_PEND_605	Navigator transmit queue pending event for indicated queue
50	CIC_2_OUT52	CIC2 Interrupt Controller output
51	CIC_2_OUT53	CIC2 Interrupt Controller output
52	PCIE_1_INT8	PCIe_1 interrupt
53	PCIE_1_INT9	PCIe_1 interrupt
54	QMSS_QUE_PEND_589	Navigator transmit queue pending event for indicated queue
55	QMSS_QUE_PEND_590	Navigator transmit queue pending event for indicated queue
56	IQNET_ATEVT0	IQNET timer event
57	IQNET_ATEVT1	IQNET timer event
58	IQNET_ATEVT11	IQNET timer event
59	IQNET_ATEVT12	IQNET timer event
60	IQNET_ATEVT13	IQNET timer event
61	IQNET_ATEVT14	IQNET timer event
62	IQNET_ATEVT15	IQNET timer event
63	IQNET_ATEVT16	IQNET timer event

Table 7-33. EDMA3CC1 Events for 66AK2L06

EVENT NO.	EVENT NAME	DESCRIPTION
0	SPI_0_INT0	SPI0 interrupt
1	SPI_0_INT1	SPI0 interrupt
2	SPI_0_XEVT	SPI0 transmit event
3	SPI_0_REVT	SPI0 receive event
4	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
5	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
6	GPIO_INT0	GPIO interrupt
7	GPIO_INT1	GPIO interrupt
8	GPIO_INT2	GPIO interrupt
9	GPIO_INT3	GPIO interrupt
10	IQNET_ATEVT0	IQNET Timer event
11	IQNET_ATEVT1	IQNET Timer event
12	IQNET_ATEVT17	IQNET Timer event
13	IQNET_ATEVT18	IQNET Timer event
14	QMSS_QUE_PEND_591	Navigator transmit queue pending event for indicated queue
15	QMSS_QUE_PEND_592	Navigator transmit queue pending event for indicated queue
16	QMSS_QUE_PEND_593	Navigator transmit queue pending event for indicated queue
17	QMSS_QUE_PEND_594	Navigator transmit queue pending event for indicated queue
18	QMSS_QUE_PEND_595	Navigator transmit queue pending event for indicated queue
19	QMSS_QUE_PEND_596	Navigator transmit queue pending event for indicated queue
20	QMSS_QUE_PEND_597	Navigator transmit queue pending event for indicated queue
21	QMSS_QUE_PEND_598	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	PCIE_0_INT10	PCIe_0 interrupt
39	PCIE_0_INT11	PCIe_0 interrupt
40	PCIE_1_INT10	PCIe_1 interrupt
41	PCIE_1_INT11	PCIe_1 interrupt
42	CIC_2_OUT0	CIC2 Interrupt Controller output
43	CIC_2_OUT1	CIC2 Interrupt Controller output
44	CIC_2_OUT2	CIC2 Interrupt Controller output
45	CIC_2_OUT3	CIC2 Interrupt Controller output
46	CIC_2_OUT4	CIC2 Interrupt Controller output

Table 7-33. EDMA3CC1 Events for 66AK2L06 (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
47	CIC_2_OUT5	CIC2 Interrupt Controller output
48	CIC_2_OUT6	CIC2 Interrupt Controller output
49	CIC_2_OUT7	CIC2 Interrupt Controller output
50	CIC_2_OUT8	CIC2 Interrupt Controller output
51	IQNET_ATEVT19	IQNET Timer event
52	IQNET_ATEVT20	IQNET Timer event
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	GPIO_INT11	GPIO interrupt
58	SPI_2_XEVT	SPI DMA TX event
59	SPI_2_REVT	SPI DMA RX event
60	CIC_2_OUT18	CIC2 Interrupt Controller output
61	CIC_2_OUT19	CIC2 Interrupt Controller output
62	USIM_RREQ	USIM read DMA event
63	USIM_WREQ	USIM write DMA event

Table 7-34. EDMA3CC2 Events for 66AK2L06

EVENT NO.	EVENT NAME	DESCRIPTION
0	Reserved	
1	Reserved	
2	Reserved	
3	Reserved	
4	IQNET_ATEVT0	IQNET timer event
5	IQNET_ATEVT1	IQNET timer event
6	TETB_FULLINT0	TETB4 is full
7	TETB_HFULLINT0	TETB4 is half full
8	TETB_FULLINT1	TETB5 is full
9	TETB_HFULLINT1	TETB5 is half full
10	TETB_FULLINT2	TETB6 is full
11	TETB_HFULLINT2	TETB6 is half full
12	TETB_FULLINT3	TETB7 is full
13	TETB_HFULLINT3	TETB7 is half full
14	UART_1_URXEVT	UART1 receive event
15	UART_1_UTXEVT	UART1 transmit event
16	QMSS_QUE_PEND_603	Navigator transmit queue pending event for indicated queue
17	QMSS_QUE_PEND_604	Navigator transmit queue pending event for indicated queue
18	IQNET_ATEVT11	IQNET timer event
19	IQNET_ATEVT12	IQNET timer event
20	IQNET_ATEVT13	IQNET timer event
21	IQNET_ATEVT21	IQNET timer event
22	IQNET_ATEVT22	IQNET timer
23	IQNET_ATEVT23	IQNET timer
24	Reserved	
25	Reserved	
26	GPIO_INT0	GPIO interrupt

Table 7-34. EDMA3CC2 Events for 66AK2L06 (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
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	Reserved	
35	Reserved	
36	Reserved	
37	Reserved	
38	CIC_2_OUT48	CIC2 Interrupt Controller output
39	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
47	CIC_2_OUT27	CIC2 Interrupt Controller output
48	UART_2_URXEVT	UART2 receive event
49	SPI_0_XEVT	SPI0 receive event
50	SPI_0_REVT	SPI0 transmit event
51	Reserved	
52	Reserved	
53	Reserved	
54	Reserved	
55	Reserved	
56	Reserved	
57	Reserved	
58	Reserved	
59	Reserved	
60	UART_2_UTXEVT	UART2 transmit event
61	UART_3_URXEVT	UART3 receive event
62	UART_3_UTXEVT	UART0 transmit event
63	Reserved	

8 System Interconnect

On the KeyStone II devices, the C66x CorePac, 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 DDR3A 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

8.1 Internal Buses and Switch Fabrics

The C66x CorePacs, 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.

8.2 Switch Fabric Connections Matrix - Data Space

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

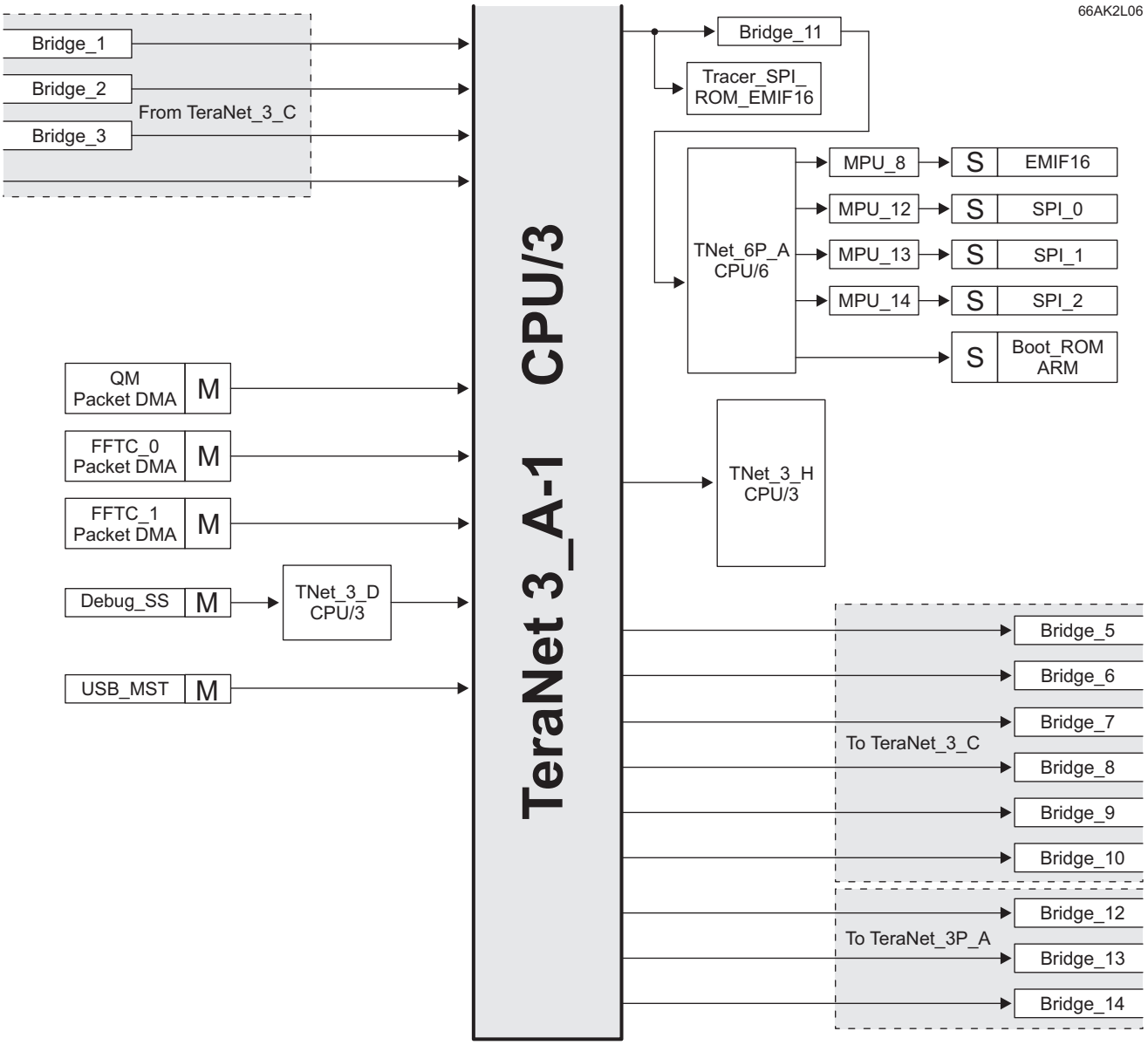


Figure 8-1. TeraNet 3_A-1

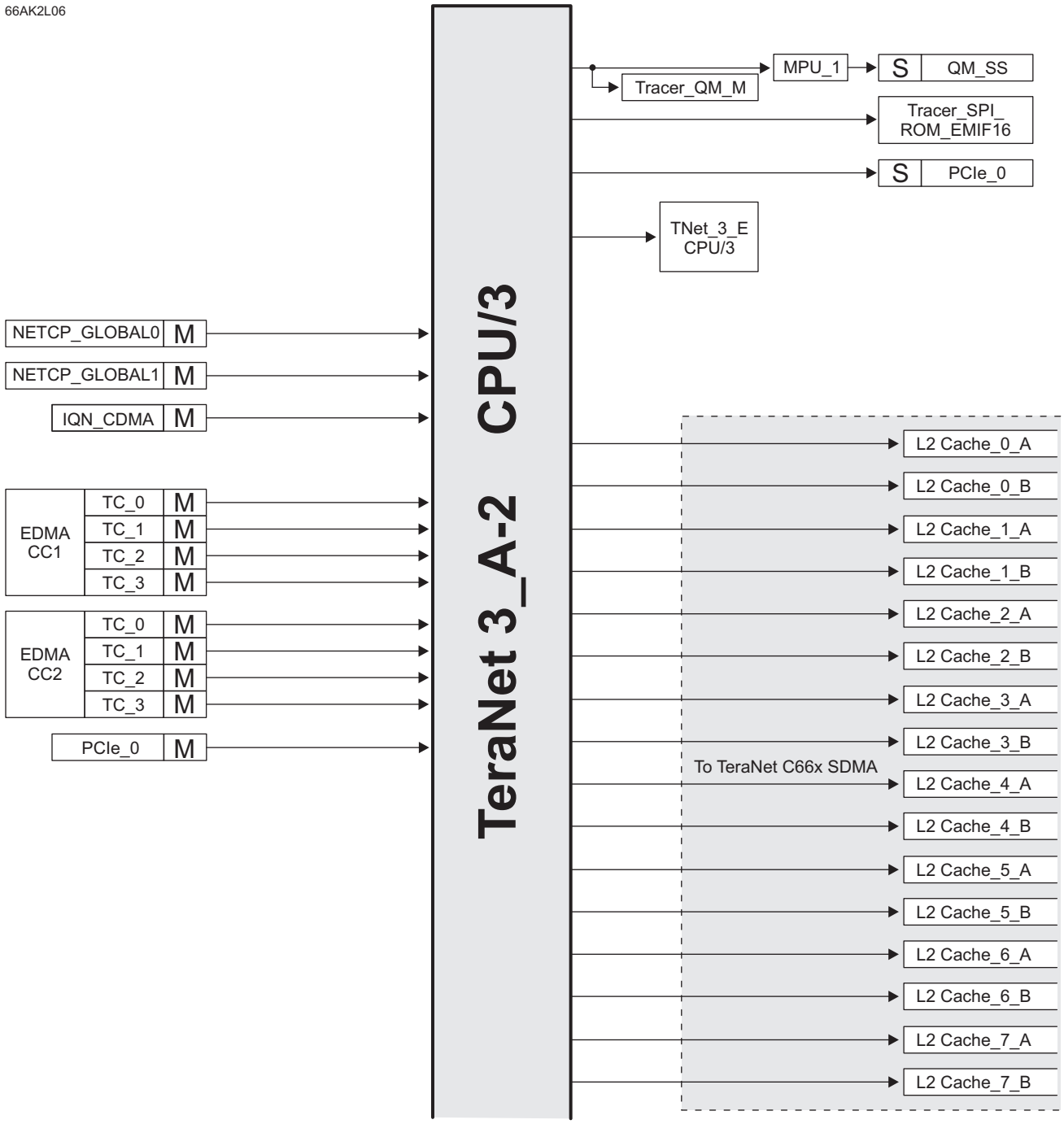


Figure 8-2. TeraNet 3_A-2

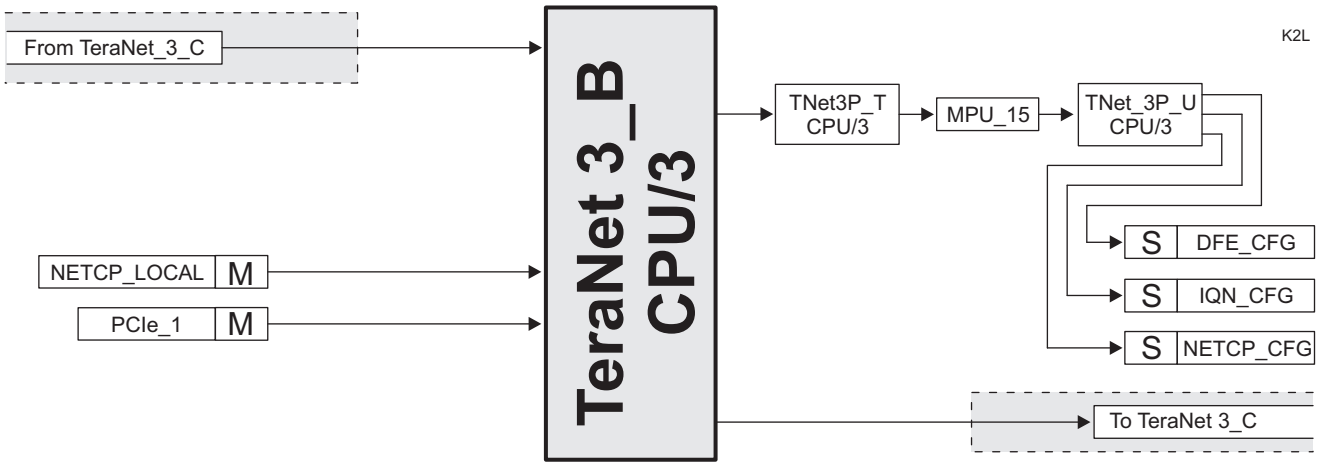


Figure 8-3. TeraNet 3_B

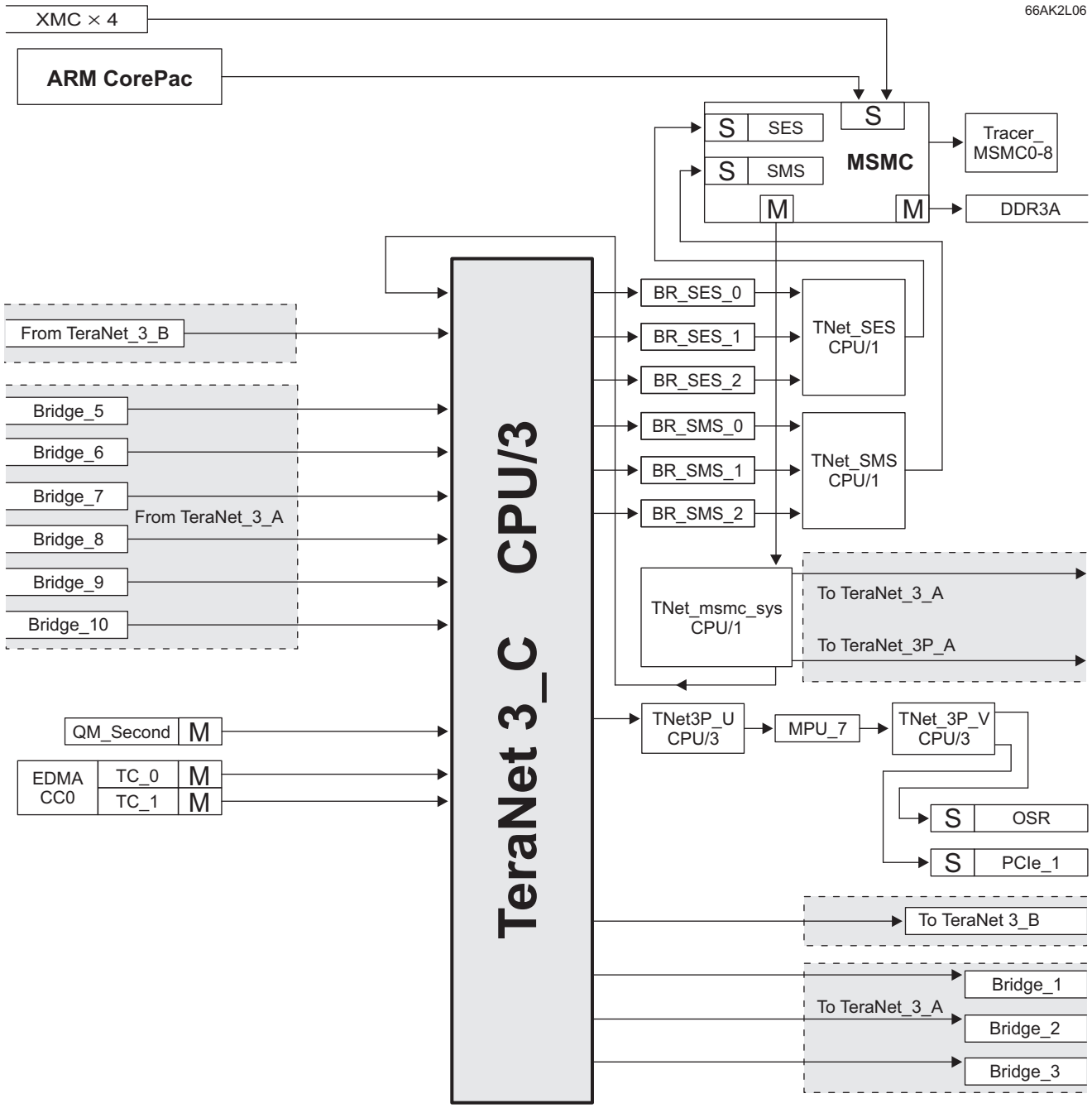


Figure 8-4. TeraNet 3_C

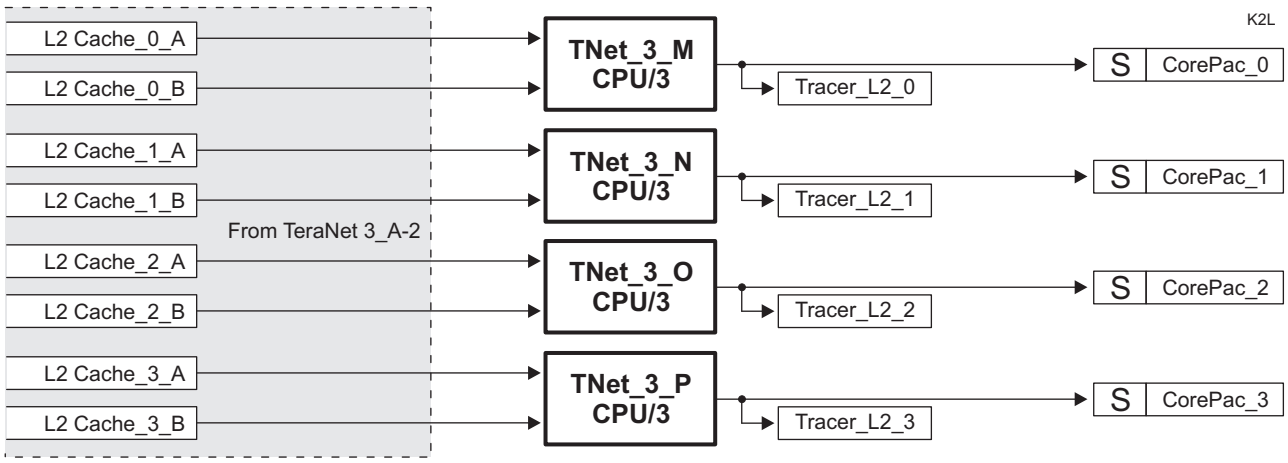


Figure 8-5. TeraNet C66x to SDMA

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 8-1. Data Space Interconnect - Section 1

	SLAVES									
MASTERS	**AEMIF16**	**Reserved**	**Reserved**	**BootROM_ARM**	**BootROM_C66x**	**CorePac0_SDMA**	**CorePac1_SDMA**	**CorePac2_SDMA**	**CorePac3_SDMA**	**DBG_STM**
IQN	-			-	-	Y	Y	Y	Y	-
Reserved										
Reserved										
Reserved										
CorePac0_CFG	-			-	-	-	-	-	-	-
CorePac1_CFG	-			-	-	-	-	-	-	-
CorePac2_CFG	-			-	-	-	-	-	-	-
CorePac3_CFG	-			-	-	-	-	-	-	-
CPT_BCR_CFG	-			-	-	-	-	-	-	Y
CPT_CFG	-			-	-	-	-	-	-	Y
CPT_DDR3A	-			-	-	-	-	-	-	Y
CPT_INTC	-			-	-	-	-	-	-	Y
CPT_L2_(0-3)	-			-	-	-	-	-	-	Y
CPT_MSMC(0-3)	-			-	-	-	-	-	-	Y
CPT_QM_CFG1	-			-	-	-	-	-	-	Y
CPT_QM_CFG2	-			-	-	-	-	-	-	Y
CPT_QM_M	-			-	-	-	-	-	-	Y
Reserved										
Reserved										
Reserved										
CPT_SM	-			-	-	-	-	-	-	Y
CPT_SPI_ROM_EMIF16	-			-	-	-	-	-	-	Y
Reserved										
CPT_TPCC(0_4)T	-			-	-	-	-	-	-	Y
CPT_TPCC(1_2_3)T	-			-	-	-	-	-	-	Y
DBG_DAP	Y			Y	Y	Y	Y	Y	Y	Y
EDMA0_CC_TR	-			-	-	-	-	-	-	-
EDMA0_TC0_RD	2,11			2,11	2,11	Y	Y	Y	Y	-
EDMA0_TC0_WR	2,11			-	-	Y	Y	Y	Y	-
EDMA0_TC1_RD	3,11			3,11	3,11	Y	Y	Y	Y	-
EDMA0_TC1_WR	3,11			-	-	Y	Y	Y	Y	-
EDMA1_CC_TR	-			-	-	-	-	-	-	-
EDMA1_TC0_RD	11			11	11	Y	Y	Y	Y	-
EDMA1_TC0_WR	11			-	-	Y	Y	Y	Y	Y
EDMA1_TC1_RD	11			Y	Y	Y	Y	Y	Y	-

Table 8-1. Data Space Interconnect - Section 1 (continued)

MASTERS	SLAVES									
	AEMIF16	Reserved	Reserved	BootROM_ARM	BootROM_C66x	CorePac0_SDMA	CorePac1_SDMA	CorePac2_SDMA	CorePac3_SDMA	DBG_STM
EDMA1_TC1_WR	11			-	-	Y	Y	Y	Y	-
EDMA1_TC2_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA1_TC2_WR	11			-	-	Y	Y	Y	Y	-
EDMA1_TC3_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA1_TC3_WR	11			-	-	Y	Y	Y	Y	Y
EDMA2_CC_TR	-			-	-	-	-	-	-	-
EDMA2_TC0_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA2_TC0_WR	11			-	-	Y	Y	Y	Y	Y
EDMA2_TC1_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA2_TC1_WR	11			-	-	Y	Y	Y	Y	-
EDMA2_TC2_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA2_TC2_WR	11			-	-	Y	Y	Y	Y	Y
EDMA2_TC3_RD	11			Y	Y	Y	Y	Y	Y	-
EDMA2_TC3_WR	11			-	-	Y	Y	Y	Y	-
FFTC_0	11			-	-	Y	Y	Y	Y	Y
FFTC_1	11			-	-	Y	Y	Y	Y	Y
MSMC_SYS	11			11	11	Y	Y	Y	Y	Y
NETCP	-			-	-	Y	Y	Y	Y	-
PCIE_0_1	11			-	-	Y	Y	Y	Y	Y
QM_Master1	-			-	-	Y	Y	Y	Y	-
QM_Master2	-			-	-	Y	Y	Y	Y	-
QM_SEC	-			-	-	Y	Y	Y	Y	Y
Reserved										
Reserved										
Reserved										
USB	-			-	-	Y	Y	Y	Y	Y

Table 8-2. Data Space Interconnect - Section 2

MASTERS	SLAVES								
	MSMC_SES	MSMC_SMS	OSR	PCIe_0_1	QM	SPI(0-2)	Reserved	Reserved	Reserved
IQN	SES_2	SMS_2	Y	-	Y	-			
Reserved									
Reserved									
Reserved									
CorePac0_CFG	-	-	-	-	-	-			
CorePac1_CFG	-	-	-	-	-	-			
CorePac2_CFG	-	-	-	-	-	-			
CorePac3_CFG	-	-	-	-	-	-			
CPT_BCR_CFG	-	-	-	-	-	-			

Table 8-2. Data Space Interconnect - Section 2 (continued)

MASTERS	SLAVES								
	MSMC_SES	MSMC_SMS	OSR	PCIe_0_1	QM	SPI(0-2)	Reserved	Reserved	Reserved
CPT_CFG	-	-	-	-	-	-			
CPT_DDR3A	-	-	-	-	-	-			
CPT_INTC	-	-	-	-	-	-			
CPT_L2_(0-3)	-	-	-	-	-	-			
CPT_MSMC(0-3)	-	-	-	-	-	-			
CPT_QM_CFG1	-	-	-	-	-	-			
CPT_QM_CFG2	-	-	-	-	-	-			
CPT_QM_M	-	-	-	-	-	-			
Reserved									
Reserved									
Reserved									
CPT_SM	-	-	-	-	-	-			
CPT_SPI_ROM_EMIF16	-	-	-	-	-	-			
Reserved									
CPT_TPCC(0_4)T	-	-	-	-	-	-			
CPT_TPCC(1_2_3)T	-	-	-	-	-	-			
DBG_DAP	Y	Y	Y	Y	Y	Y			
EDMA0_CC_TR	-	-	-	-	-	-			
EDMA0_TC0_RD	SES_0	SMS_0	Y	Y	Y	2,11			
EDMA0_TC0_WR	SES_0	SMS_0	Y	Y	Y	2,11			
EDMA0_TC1_RD	SES_1	SMS_1	Y	Y	-	3,11			
EDMA0_TC1_WR	SES_1	SMS_1	Y	Y	-	3,11			
EDMA1_CC_TR	-	-	-	-	-	-			
EDMA1_TC0_RD	SES_0	SMS_0	Y	Y	Y	11			
EDMA1_TC0_WR	SES_0	SMS_0	Y	Y	Y	11			
EDMA1_TC1_RD	SES_1	SMS_1	Y	Y	Y	11			
EDMA1_TC1_WR	SES_1	SMS_1	Y	Y	Y	11			
EDMA1_TC2_RD	SES_1	SMS_1	Y	Y	-	11			
EDMA1_TC2_WR	SES_1	SMS_1	Y	Y	-	11			
EDMA1_TC3_RD	SES_1	SMS_1	Y	Y	-	11			
EDMA1_TC3_WR	SES_1	SMS_1	Y	Y	-	11			
EDMA2_CC_TR	-	-	-	-	-	-			
EDMA2_TC0_RD	SES_2	SMS_2	Y	Y	Y	11			
EDMA2_TC0_WR	SES_2	SMS_2	Y	Y	Y	11			
EDMA2_TC1_RD	SES_2	SMS_2	Y	Y	Y	11			
EDMA2_TC1_WR	SES_2	SMS_2	Y	Y	Y	11			
EDMA2_TC2_RD	SES_0	SMS_0	Y	Y	-	11			
EDMA2_TC2_WR	SES_0	SMS_0	Y	Y	-	11			
EDMA2_TC3_RD	SES_0	SMS_0	Y	Y	-	11			
EDMA2_TC3_WR	SES_0	SMS_0	Y	Y	-	11			
FFTC_0	SES_1	SMS_1	Y	-	Y	11			
FFTC_1	SES_1	SMS_1	Y	-	Y	11			
MSMC_SYS	-	-	Y	Y	Y	11			
NETCP	SES_1	SMS_1	Y	Y	Y	-			

Table 8-2. Data Space Interconnect - Section 2 (continued)

MASTERS	SLAVES								
	MSMC_SES	MSMC_SMS	OSR	PCIe_0_1	QM	SPI(0-2)	Reserved	Reserved	Reserved
PCIe_0_1	SES_2	SMS_2	Y	-	Y	11			
QM_Master1	SES_0	SMS_0	Y	-	Y	-			
QM_Master2	SES_1	SMS_1	Y	-	Y	-			
QM_SEC	SES_2	SMS_2	Y	-	-	-			
Reserved									
Reserved									
Reserved									
USB	SES_0	SMS_0	Y	-	Y	-			

Table 8-3. Data Space Interconnect - Section 3

TeraNet_3_A MASTERS	SLAVES					
	BR_5 (to TeraNet_3_C)	BR_6 (to TeraNet_3_C)	BR_7 (to TeraNet_3_C)	BR_8 (to TeraNet_3_C)	BR_9 (to TeraNet_3_C)	BR_10 (to TeraNet_3_C)
EDMA1_TC0_RD	Y	-	-	-	-	-
EDMA1_TC0_WR	Y	-	-	-	-	-
EDMA1_TC1_RD	-	Y	-	-	-	-
EDMA1_TC1_WR	-	Y	-	-	-	-
EDMA1_TC2_RD	-	-	Y	-	-	-
EDMA1_TC2_WR	-	-	Y	-	-	-
EDMA1_TC3_RD	-	-	-	Y	-	-
EDMA1_TC3_WR	-	-	-	Y	-	-
EDMA2_TC0_RD	-	-	-	-	Y	-
EDMA2_TC0_WR	-	-	-	-	Y	-
EDMA2_TC1_RD	-	-	-	-	-	Y
EDMA2_TC1_WR	-	-	-	-	-	Y
EDMA2_TC2_RD	Y	-	-	-	-	-
EDMA2_TC2_WR	Y	-	-	-	-	-
EDMA2_TC3_RD	-	Y	-	-	-	-
EDMA2_TC3_WR	-	Y	-	-	-	-
Debug_SS	-	-	-	-	Y	-
Reserved						
Reserved						
Reserved						
PCIE_0	-	-	-	-	-	Y
NETCP	-	-	-	-	Y	-
BR_50	Y	-	-	-	-	-
IQN_CDMA	-	-	-	-	Y	-
USB_MST	-	-	Y	-	-	-

Table 8-3. Data Space Interconnect - Section 3 (continued)

TeraNet_3_A MASTERS	SLAVES					
	BR_5 (to TeraNet_3_C)	BR_6 (to TeraNet_3_C)	BR_7 (to TeraNet_3_C)	BR_8 (to TeraNet_3_C)	BR_9 (to TeraNet_3_C)	BR_10 (to TeraNet_3_C)
FFTC_B_CDMA	-	-	-	Y	-	-
FFTC_A_CDMA	-	Y	-	-	-	-
NETCP_GLOBAL0	-	-	Y	-	-	-
QM_MST1	Y	-	-	-	-	-

Table 8-4. Data Space Interconnect - Section 4

TeraNet_3_C MASTERS	SLAVES								
	BR_SES_0	BR_SMS_0	BR_SES_1	BR_SMS_1	BR_SMS_2	BR_SES_2	Bridge_1	Bridge_2	Bridge_3
EDMA0_TC0_RD	Y	Y	-	-	-	-	-	Y	-
EDMA0_TC0_WR	Y	Y	-	-	-	-	-	Y	-
EDMA0_TC1_RD	-	-	Y	Y	-	-	-	-	Y
EDMA0_TC1_WR	-	-	Y	Y	-	-	-	-	Y
MSMC_SYS_MST (via Tnet_msmc_sys)	-	-	-	-	-	-	-	-	-
BR_5	Y	Y	-	-	-	-	-	-	-
BR_6	Y	Y	-	-	-	-	-	-	-
BR_7	-	-	Y	Y	-	-	-	-	-
BR_8	-	-	Y	Y	-	-	-	-	-
BR_9	-	-	-	-	Y	Y	-	-	-
BR_10	-	-	-	-	Y	Y	-	-	-
NETCP_LOCAL/PCIE_1 - From TeraNet_3_B	-	-	-	-	Y	Y	Y	-	-
QM_Second	-	-	-	-	Y	Y	-	Y	-
RAC0_BE1_LP	-	-	-	-	Y	Y	-	-	Y

8.3 TeraNet Switch Fabric Connections Matrix - Configuration Space

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

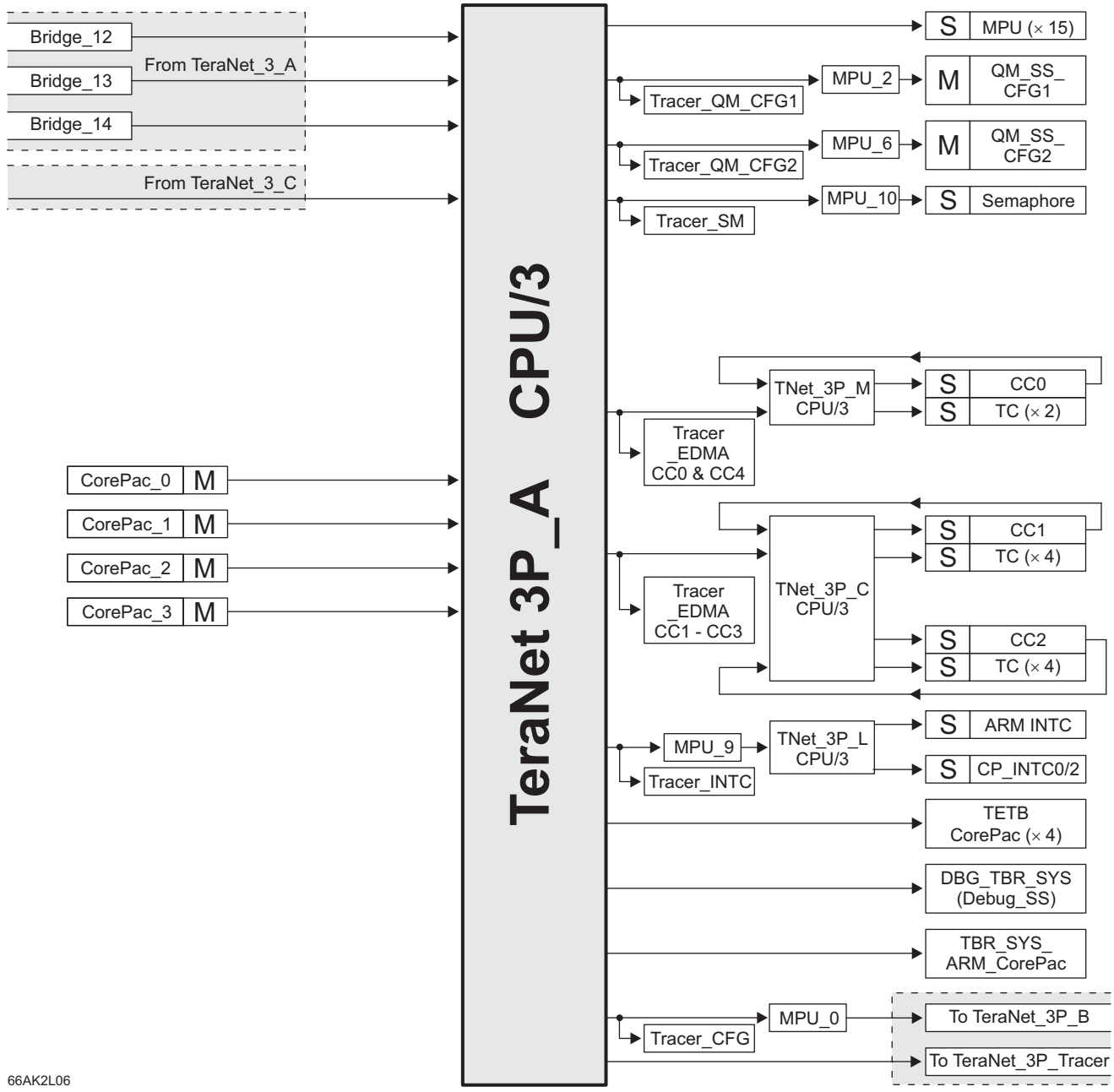


Figure 8-6. TeraNet 3P_A

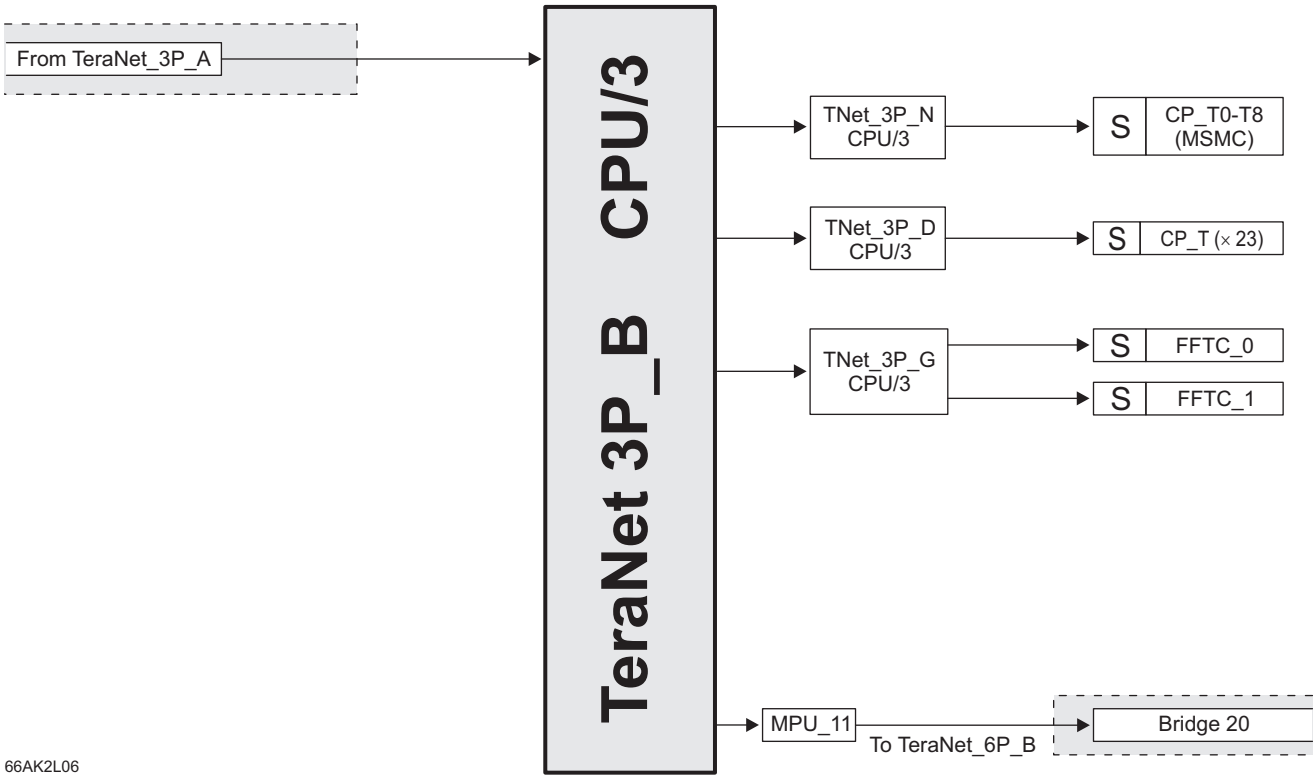


Figure 8-7. TeraNet 3P_B

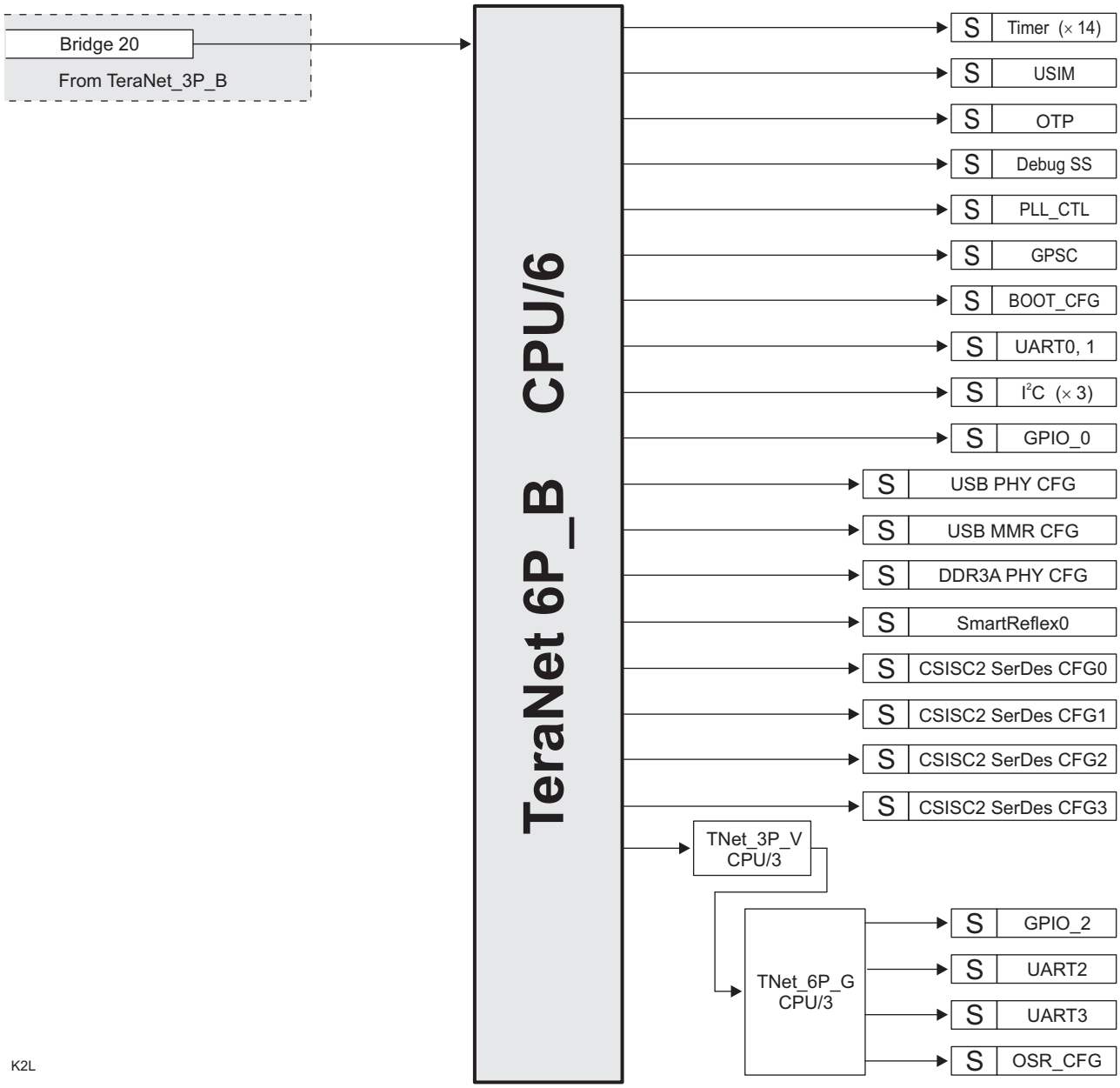


Figure 8-8. TeraNet 6P_B

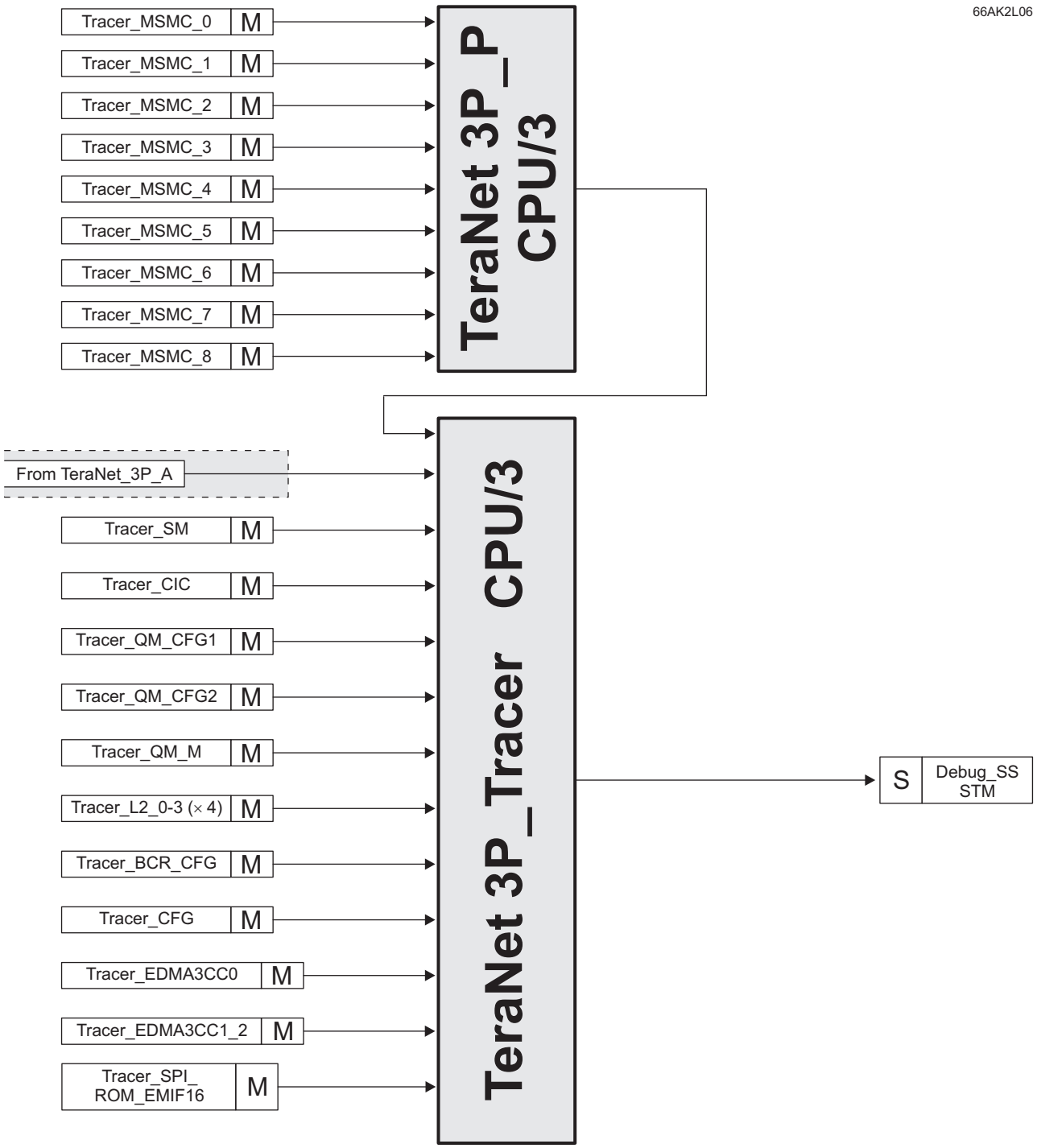


Figure 8-9. TeraNet 3P_Tracer

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 8-5. Configuration Space Interconnect - Section 1

	SLAVES																									
MASTERS	**ADTF(0-7)_CFG**	**IQN_CFG**	**CSISC2 SERDES CFG0-1**	**CSISC2 SERDES CFG2-3**	**ARM_CFG**	**Reserved**	**BCR_CFG**	**BOOTCFG_CFG**	**CP_INTC_CFG**	**CPT_BCR_CFG_CFG**	**CPT_CFG_CFG**	**CPT_DDR3A_CFG**	**CPT_INTC(0-2)_CFG**	**CPT_L2_(0-3)_CFG**	**CPT_MSMC(0-3)_CFG**	**CPT_QM_CFG1_CFG**	**CPT_QM_CFG2_CFG**	**CPT_QM_M_CFG**	**Reserved**	**Reserved**	**Reserved**	**CPT_SM_CFG**	**CPT_SPI_ROM_EMIF16_CFG**	**Reserved**	**CPT_TPCC0_CFG**	**CPT_TPCC1_2_CFG**
IQN	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
Reserved																										
Reserved																										
CorePac0_CFG	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y		Y	Y
CorePac1_CFG	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y		Y	Y
CorePac2_CFG	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y		Y	Y
CorePac3_CFG	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				Y	Y		Y	Y
DBG_DAP	Y	Y	Y	Y	Y		Y	Y	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	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
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	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
EDMA2_CC_TR	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
EDMA2_TC0_RD	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
EDMA2_TC1_RD	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
EDMA2_TC1_WR	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
EDMA2_TC2_RD	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
EDMA2_TC3_RD	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
EDMA2_TC3_WR	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
FFTC_0	12	12	12	12	12		12	12	12	12	12	12	12	12	12	12	12	12				12	12		12	12

Table 8-5. Configuration Space Interconnect - Section 1 (continued)

	SLAVES																									
MASTERS	**ADTF(0-7)_CFG**	**IQN_CFG**	**CSISC2 SERDES CFG0-1**	**CSISC2 SERDES CFG2-3**	**ARM_CFG**	**Reserved**	**BCR_CFG**	**BOOTCFG_CFG**	**CP_INTC_CFG**	**CPT_BCR_CFG_CFG**	**CPT_CFG_CFG**	**CPT_DDR3A_CFG**	**CPT_INTC(0-2)_CFG**	**CPT_L2_(0-3)_CFG**	**CPT_MSMC(0-3)_CFG**	**CPT_QM_CFG1_CFG**	**CPT_QM_CFG2_CFG**	**CPT_QM_M_CFG**	**Reserved**	**Reserved**	**Reserved**	**CPT_SM_CFG**	**CPT_SPI_ROM_EMIF16_CFG**	**Reserved**	**CPT_TPCC0_CFG**	**CPT_TPCC1_2_CFG**
FFTC_1	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
NETCP	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
PCIe_0_1	12	12	12	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		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-
Reserved																										
Reserved																										
Reserved																										
USB	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	-	-				-	-		-	-

Table 8-6. Configuration Space Interconnect - Section 2

	SLAVES																				
MASTER	**DBG_CFG**	**DBG_TBR_SYS**	**DDR3A_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**	**FFTC_(0-1)_CFG**	**GIC_CFG**	**GPIO_0_1_CFG**	**I2C(0-2)_CFG**	**MPU(0-14)_CFG**	**NETCP_CFG**	**NETCP_SERDES_CFG**	**OTP_CFG**	**PCIe_SERDES_CFG**	**PLL_CTL_CFG**	**PSC_CFG**	**QM_CFG1**
IQN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Reserved																					
Reserved																					
CorePac0_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac1_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac2_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
CorePac3_CFG	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
DBG_DAP	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	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_WR	-	-	-	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_RD	-	12	-	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_WR	-	-	-	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
EDMA1_TC0_WR	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	-	-	-	-	-	-	-	-	-	-	-	-

Table 8-6. Configuration Space Interconnect - Section 2 (continued)

	SLAVES																				
MASTER	**DBG_CFG**	**DBG_TBR_SYS**	**DDR3A_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**	**FFTC_(0-1)_CFG**	**GIC_CFG**	**GPIO_0_1_CFG**	**I2C(0-2)_CFG**	**MPU(0-14)_CFG**	**NETCP_CFG**	**NETCP_SERDES_CFG**	**OTP_CFG**	**PCIe_SERDES_CFG**	**PLL_CTL_CFG**	**PSC_CFG**	**QM_CFG1**
EDMA1_TC1_WR	-	-	-	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_RD	-	-	-	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_WR	-	-	-	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
EDMA1_TC3_WR	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
EDMA2_TC0_WR	12	-	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC1_RD	-	-	-	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC1_WR	-	-	-	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
EDMA2_TC2_WR	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	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC3_WR	-	-	-	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-
FFTC_0	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
FFTC_1	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
NETCP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCIe_0_1	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	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Master2	-	-	-	12	-	12	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SEC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-
Reserved																					
Reserved																					
USB	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 8-7. Configuration Space Interconnect - Section 3

	SLAVES																	
MASTERS	**QM_CFG1**	**QM_CFG2**	**Reserved**	**SEC_MGR_CFG**	**SM_CFG**	**SR_CFG(0-1)**	**Reserved**	**TBR_SYS_ARM**	**Reserved**	**TETB0_CFG**	**TETB1_CFG**	**TETB2_CFG**	**TIMER(0-19)_CFG**	**UART(0-1)_CFG**	**USB_MMR_CFG**	**USB_PHY_CFG**	**USIM_CFG**	**Reserved**
IQN	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
Reserved																		
Reserved																		
CorePac0_CFG	Y	Y		Y	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	
CorePac1_CFG	Y	Y		Y	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	
CorePac2_CFG	Y	Y		Y	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	

Table 8-7. Configuration Space Interconnect - Section 3 (continued)

MASTERS	SLAVES																	
	QM_CFG1	QM_CFG2	Reserved	SEC_MGR_CFG	SM_CFG	SR_CFG(0-1)	Reserved	TBR_SYS_ARM	Reserved	TETB0_CFG	TETB1_CFG	TETB2_CFG	TIMER(0-19)_CFG	UART(0-1)_CFG	USB_MMR_CFG	USB_PHY_CFG	USIM_CFG	Reserved
CorePac3_CFG	Y	Y		Y	Y	Y		Y		Y	Y	Y	Y	Y	Y	Y	Y	
DBG_DAP	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	
EDMA1_TC0_WR	12	12		12	-	12		12		-	-	-	12	12	12	12	12	
EDMA1_TC1_RD	-	-		-	-	-		-		13	13	-	-	-	-	-	-	
EDMA1_TC1_WR	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
EDMA1_TC2_RD	-	-		-	-	-		-		-	-	14	-	-	-	-	-	
EDMA1_TC2_WR	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
EDMA1_TC3_RD	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	-	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	-	-	-	-	-	-	
EDMA2_TC1_WR	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
EDMA2_TC2_RD	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	-	-	-	-	-	
EDMA2_TC3_WR	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
FFTC_0	12	12		12	12	12		12		12	12	12	12	12	12	12	12	
FFTC_1	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	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
PCIe_0_1	12	12		12	12	12		12		12	12	12	12	12	12	12	12	
QM_Master1	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
QM_Master2	-	-		-	-	-		-		-	-	-	-	-	-	-	-	
QM_SEC	-	-		-	-	-		12		-	-	-	-	-	12	-	-	
Reserved																		
Reserved																		
Reserved																		
USB	-	-		-	-	-		12		12	12	12	-	-	-	-	-	

8.4 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. Examples include the C66x CorePacs, whose priorities are set through software in the UMC control registers. 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 Figure 9-45 and Table 9-60.

For all other modules, see the respective User's Guides listed in Section 3.5 for programmable priority registers.

9 Device Boot and Configuration

9.1 Device Boot

9.1.1 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 66AK2L06 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 66AK2L06 devices, there are two types of booting: the C66x CorePac as the boot master and the ARM CorePac as the boot master. The ARM CorePac does not support no-boot mode. Both the C66x CorePacs and the ARM CorePac need to read the bootmode register to determine how to proceed with the boot.

Table 9-1 shows memory space reserved for boot by the C66x CorePac.

Table 9-1. C66x DSP Boot RAM Memory Map

START ADDRESS	SIZE	DESCRIPTION
0x80_0000	0x1_0000	Reserved
0x8e_7f80	0x80	C66x CorePac ROM version string
0x8e_8000	0x7f00	Ethernet Package memory
0x8e_fe80	0x7e80	PCIe config block
0x8e_fff0	4	Host Data Address (boot magic address for secure boot through master peripherals)
0x8f_7800	0x410	Secure host Data structure
0x8f_a290	0x4000	Boot Stack
0x8f_e290	0x90	Boot Log Data
0x8f_e320	0x20	Boot Status Stack
0x8f_e410	0xf0	Boot Stats
0x8f_e520	0x13fc	Boot Data
0x8f_f91c	0x404	Boot Trace Info
0x8f_fd20	0x180	DDR Config
0x8f_fea0	0x60	Boot RAM call table
0x8f_ff00	0x80	Boot Parameter table
0x8f_fff8	0x4	Secure Signal Magic address
0x8f_fffc	0x4	Boot Magic address

Table 9-2 shows addresses reserved for boot by the ARM CorePac.

Table 9-2. ARM Boot RAM Memory Map

START ADDRESS	SIZE	DESCRIPTION
0x0c1d_8000	0x180	ARM0 Version info
0x0c1d_0180	0x80	ARM0 Boot progress stack
0x0c1d_0200	0x100	ARM0 Boot stats
0x0c1d_0300	0x100	ARM0 Boot Log data
0x0c1d_0400	0x100	ARM0 RAM Call tables
0x0c1d_0500	0x100	RAM0 Boot Parameter tables
0x0c1d_0600	0x99e0	ARM0 Local core Boot data

Table 9-2. ARM Boot RAM Memory Map (continued)

START ADDRESS	SIZE	DESCRIPTION
0x0c1d_9fe0	0x2020	ARM0 Boot Trace data
0x0c1d_c000	0x180	ARM1(1) Version info
0x0c1d_c180	0x80	ARM1 Boot progress stack
0x0c1d_c200	0x100	ARM1 Boot stats
0x0c1d_c300	0x100	ARM1 Boot Log data
0x0c1d_c400	0x100	ARM1 RAM Call tables
0x0c1d_c500	0x100	RAM1 Boot Parameter tables
0x0c1d_c600	0x99e0	ARM1 Local core Boot data
0x0c1d_cfe0	0x2020	ARM1 Boot Trace data
0xc0c1e_0000	0x4000	ARM0 Secure Load data
0xc0c1e_4000	0x2ab0	ARM0 Secure Boot data
0xc0c1e_6ab0	0x1550	ARM0 Secure Stack
0xc0c1e_8000	0x4000	ARM1 Secure Load data
0xc0c1e_c000	0x2ab0	ARM1 Secure Boot data
0xc0c1e_eab0	0x1550	ARM1 Secure Stack

(1) The addresses shown for core 1 are the physical addresses. Boot ROM enables the non-secure MMU during the boot process, and the physical memory shown for ARM core 1(non-secure area) is mapped to the same virtual addresses used by core 0. Core 0 has a flat map. Likewise for the secure MMU in the secure memory region. When the non-secure boot ROM exits normally the non-secure MMU is disabled, and for non-secure devices the secure MMU is disabled as well.

9.1.2 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 C6xx CorePac0 is the boot master** — The C66x CorePac is released from reset and begins executing from the L3 ROM base address. The ARM CorePac is also released from reset at the same time as the C66xCorePac. Both the C66x CorePac and the ARM CorePac read the bootmode register inside the bootCFG module to determine which is the boot master.

  After the Boot ROM for the Cortex-A15 processor reads the bootmode to determine that the C66x CorePac is the boot master, all Cortex-A15 processors stay idle by executing WFI instruction and waiting for the C66x CorePac's interrupt. The chip Boot ROM reads the bootmode register to determine that the C66x CorePac0 is the boot master, then the C66x CorePac0 performs the boot process and the other C66x CorePacs execute an IDLE instruction. After the boot process is completed, the C66x CorePac0 begins to execute the code downloaded during the boot process. If the downloaded code included code for the other C66x cores and/or the Cortex-A15 processor cores, the downloaded code may contain logic to write the code execution addresses to the boot address register for the core that is to execute it. The C66x CorePac0 can then generate an interrupt to the core causing it to execute the code. When they receive the IPC interrupt, the rest of the C66x CorePacs and the ARM CorePac complete boot management operations and begin executing from the predefined location in memory.
- **Public ROM Boot when the ARM CorePac Core0 is the boot master** — The only difference between this boot mode and when the C66x CorePac is the boot master, is that the ARM CorePac performs the boot process while the C66x CorePacs execute idle instructions. When the ARM CorePac Core0 finishes the boot process, it may send interrupts to the C66x CorePacs and Cortex-A15 processor cores through IPC registers. The C66x CorePacs complete the boot management operations and begin executing from the predefined locations.

The boot process performed by the C66x CorePac0 and the ARM CorePac Core0 in public ROM boot is determined by the BOOTMODE[15:0] value in the DEVSTAT register. The C66x CorePac0 and the ARM CorePac Core0 read this value, and then execute the associated boot process in software. Bit 8 determines whether the boot is C66x CorePac boot or ARM CorePac boot. The figure below shows the bits associated with BOOTMODE[15:0] (DEVSTAT[16:1]) when the C66x CorePac or ARM CorePac is the boot master. Note that Figure 9-1 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 11.5.

NOTE

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

Figure 9-1. 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>ARMEN</td><td>SYSEN</td><td colspan="3" rowspan="2">ARM PLL CONFIG</td><td rowspan="8">Boot Master</td><td colspan="3" rowspan="2">SYS PLL CONFIG</td><td rowspan="2">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>Ref clk</td><td colspan="5">Bar Config</td><td colspan="3">SYS PLL CONFIG</td><td>Port</td><td>0</td><td>0</td><td>1</td><td>PCIe</td></tr>
<tr><td>X</td><td>X</td><td>X</td><td colspan="3" rowspan="2">Param Index</td><td>Bus Addr</td><td>X</td><td rowspan="2">Port</td><td rowspan="2">Min</td><td>0</td><td>1</td><td>0</td><td>I2C MASTER</td></tr>
<tr><td>Width1</td><td colspan="2">Mode</td><td>Csel</td><td>Width0</td><td>0</td><td>1</td><td>1</td><td>SPI</td></tr>
<tr><td>0</td><td>Wait</td><td>Width</td><td colspan="2" rowspan="2">Chip Sel</td><td colspan="3" rowspan="4">ARM PLL CONFIG</td><td colspan="3" rowspan="4">SYS PLL CONFIG</td><td>X</td><td>1</td><td>0</td><td>0</td><td>EMIF</td></tr>
<tr><td>Clear</td><td colspan="2">First Block</td><td rowspan="3">Min</td><td>1</td><td>0</td><td>1</td><td>NAND</td></tr>
<tr><td>Lane Setup</td><td>NETCP clk</td><td>Ref clk</td><td colspan="2">Ext Con</td><td>1</td><td>1</td><td>0</td><td>Ethernet</td></tr>
<tr><td>X</td><td>X</td><td>X</td><td colspan="2">Port</td><td>1</td><td>1</td><td>1</td><td>UART</td></tr>
</table>

9.1.2.1 Boot Device Field

The Boot Device field BOOTMODE[16-14-4-3-2-1] and the Boot Device field BOOTMODE[8] define the boot device and the boot master that is chosen. Table 9-3 shows the supported boot modes.

Table 9-3. 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 • C66x is a boot master when BOOTMODE[8]=1 – Sleep = XX[Min]000b – I2C Slave = [Slave Addr1]1[Min]000b – PCI = [Ref clk][Bar Config2]Port001b – I2C Master = XX[Min]010b – SPI = [Width1][Mode0][Min]011b – EMIF = 0[Width][Min]100b – NAND = Clear[FirstBlock0][Min]101b – Ethernet (SGMII) = Lane Setup [Ref Clk][Min]110b – UART = XX[Min]111b

9.1.2.2 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.

9.1.2.2.1 Sleep Boot Mode Configuration

Figure 9-2. 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	X	ARMen	SYSEN	ARM PLL Cfg			Boot Master	Sys PLL Config			Min	000			Lendian

Table 9-4. Sleep Boot Configuration Field Descriptions

Bit	Field	Description
16-14	Reserved	Reserved
13	ARMen	Enable the ARM PLL • 0 = PLL disabled • 1 = PLL enabled
12	SYSEN	Enable the System PLL • 0 = PLL disabled (default) • 1 = PLL enabled
11-9	ARM PLL Setting	The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies.
8	Boot Master	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
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 9-23 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

9.1.2.2.2 I^2C Boot Device Configuration

9.1.2.2.2.1 I^2C Passive Mode

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

Figure 9-3. I^2C 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		ARM PLL Cfg			Boot Master	Sys PLL Config			Min	000			Lendian

Table 9-5. 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
14	Boot Devices	Boot Device[14] used in conjunction with Boot Devices [Used in 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	ARM PLL Setting	The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies.
8	Boot Master	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
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 9-23 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

9.1.2.2.2.2 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 9-4. 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			Param Idx/Offset			Bus Addr		Boot Master	Reserved	Port		Min	010			Lendian

Table 9-6. I²C Master Mode Device Configuration Field Descriptions

Bit	Field	Description
16-14	Reserved	Reserved
13-11	Param Idx/Offset	Parameter Table Index: 0-7 This value specifies the parameter table index when the C66x is the boot master This value specifies the start read address at 8K times this value when the ARM is the boot master

Table 9-6. I²C Master Mode Device Configuration Field Descriptions (continued)

Bit	Field	Description
10-9	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
8	Boot Master	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
7	Reserved	Reserved
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] • 010 = I²C Master • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

9.1.2.2.3 SPI Boot Device Configuration

Figure 9-5. 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
Width1	Mode		Param Idx/Offset			Csel		Boot Master	Width0	Port		Min	011			Lendian

Table 9-7. SPI Device Configuration Field Descriptions

Bit	Field	Description
16-7	Width1:Width0	SPI address width configuration • 00 = 16-bit address values are used • 01 = 24-bit address values are used (default) • 01 = 32-bit address values are used • 11 = 32-bit address values are used
15-14	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.
13-11	Param Idx/Offset	Parameter Table Index: 0-7 This value specifies the parameter table index when the C66x is the boot master This value specifies the start read address at 8K times this value when the ARM is the boot master

Table 9-7. SPI Device Configuration Field Descriptions (continued)

Bit	Field	Description
10-9	Csel	The chip select field value 0-3 (default = 0)
8	Boot Master	Boot Master select • 0 = ARM is boot master (default) • 1 = C66x is boot master
7	Width0	Width0
6-5	Port	Specify SPI port • 0 = SPI0 used (default) • 1 = SPI1 used • 2 = SPI2 used • 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] • 011 = SPI boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

9.1.2.2.4 EMIF Boot Device Configuration

Figure 9-6. 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**
Base Addr	Wait	Width	Chip Sel		ARM PLL Cfg			Boot Master=0	Sys PLL Cfg			Min	100			Lendian

Table 9-8. EMIF Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Base Addr	Base address (0-3) used to calculate the branch address. Branch address is the chip select plus Base Address *16MB
15	Wait	Extended Wait • 0 = Extended Wait disabled • 1 = Extended Wait enabled
14	Width	EMIF Width • 0 = 8-bit EMIF Width • 1 = 16-bit EMIF Width
13-12	Chip Sel	Chip Sel 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	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
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 9-23 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 [4] • 100 = EMIF boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

9.1.2.2.5 NAND Boot Device Configuration

Figure 9-7. 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
Clear	First Block		Chip Sel		ARM PLL Cfg			Boot Master	Sys PLL Cfg			Min	101			Lendian

Table 9-9. NAND Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Clear	ClearNAND • 0 = Device is not a ClearNAND (default) • 1 = Device is a ClearNAND
15-14	First Block	First Block. This value is used to calculate the first block read. The first block read is the first block value *16.
13-12	Chip Sel	Chip Sel 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 ($\overline{\text{EMIFCE3}}$)

Table 9-9. NAND Boot Device Configuration Field Descriptions (continued)

Bit	Field	Description
11-9	ARM PLL Setting	ARM PLL Setting. The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies.
8	Boot Master	Boot Master select • 0 = ARM is boot master (default) • 1 = C66x is boot master
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 9-23 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

9.1.2.3 Ethernet (SGMII) Boot Device Configuration

Figure 9-8. 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**
Lane Setup	NetCP clk	Ref Clock	Ext Con		ARM PLL Cfg			Boot Master	Sys PLL Cfg			Min	110			Lendian

Table 9-10. Ethernet (SGMII) Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Lane Setup	Lane Setup. • 0 = All SGMII ports enabled (default) • 1 = Only SGMII port 0 enabled
15	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 125 Mhz SerDes reference clock (default)
14	Ref Clock	NETCP SerDes reference clock frequency • 0 = 125MHz (default) • 1 = 156.25MHz
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/ARM PLL Setting	When Boot Master =0 (ARM is Boot Master), pin[11:9] used as ARM PLL Setting. The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies. When Boot Master =1 (C66x is Boot Master), pin [10:9] are used as Lane Set up. • 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 • 4-7 = Reserved
8	Boot Master	Boot Master select • 0 = ARM is boot master (default) • 1 = C66x is boot master
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 9-23 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 • 110 = Ethernet boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

9.1.2.3.1 PCIe Boot Device Configuration

Figure 9-9. 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				ARM PLL Cfg			Boot Master=0	Sys PLL Cfg			Port	001			Lendian

Table 9-11. 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 9-12.
11-9	ARM PLL Setting	ARM PLL Setting. The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies.
8	Boot Master	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
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 9-23 shows settings for various input clock frequencies.
4	Port	Boot configured for boot. • 0 = Port 0 configured for boot • 1 = Port 1 configured for boot.
3-1	Boot Devices	Boot Devices[4:1] • 001 = PCIe boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

Table 9-12. BAR Config / PCIe Window Sizes

		32-BIT ADDRESS TRANSLATION					64-BIT ADDRESS TRANSLATION	
BAR CFG	BAR0	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

9.1.2.3.2 UART Boot Device Configuration

Figure 9-10. 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	ARM PLL Cfg			Boot Master	Sys PLL Config			Min	111			Lendian

Table 9-13. UART Boot Configuration Field Descriptions

Bit	Field	Description
16-13	Reserved	Not Used
12	Port	UART Port number • 0 = UART0 (default) • 1 = UART1
11-9	ARM PLL Setting	The PLL default settings are determined by the [11:9] bits. This will set the PLL to the maximum clock setting for the device. Table 9-23 shows settings for various input clock frequencies.
8	Boot Master	Boot Master select • 0 = ARM is boot master • 1 = C66x is boot master
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 9-23 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

9.1.2.4 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 9-14.

Table 9-14. 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	Sec PLL Config, MSW	ARM PLL configuration, MSW
14	Sec PLL Config, LSW	ARM PLL configuration, LSW
16	System Freq	The Frequency of the system clock in MHz
18	Core Freq	The frequency of the core clock in MHz

Table 9-14. Boot Parameter Table Common Parameters (continued)

BYTE OFFSET	NAME	DESCRIPTION
20	Boot Master	Set to TRUE if C66x is the master core.

9.1.2.4.1 EMIF16 Boot Parameter Table

Table 9-15. 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

9.1.2.4.2 Ethernet Boot Parameter Table

Table 9-16. 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 PASS 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

Table 9-16. Ethernet Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
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
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	Reserved		
64	Eth Ref, High	SGMII reference clock frequency, MHz. Only 12500 and 15625 are supported.	NO
66	Eth Ref, Low		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

9.1.2.4.3 PCIe Boot Parameter Table

Table 9-17. 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 9-17. 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	Window 5 Size	Window 5 Size. Valid only if the address width is 32.	NO
40	Vendor ID	Vendor ID	NO
42	Device ID	Device ID	NO
44	Class code Rev ID MSW	Class code revision ID MSW	NO
46	Class code Rev ID LSW	Class code revision ID LSW	NO
60	Timeout period (Secs)	The timeout period. Values 0 disables the time out	

9.1.2.4.4 I^2C Boot Parameter Table

Table 9-18. 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
32	Bus Frequency	The desired I^2C data rate (kHz)	NO
34	Next Dev Addr	The next device address to boot (Used only if boot config option is selected)	NO
36	Next Dev Addr Ext	The extended next device address to boot (Used only if boot config option is selected)	NO
38	Address Delay	The number of CPU cycles to delay between writing the address to an I^2C EEPROM and reading data.	NO

9.1.2.4.5 SPI Boot Parameter Table

Table 9-19. 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

9.1.2.4.6 UART Boot Parameter Table

Table 9-20. 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

Table 9-20. UART Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
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
50	Blob Base, MSW	For blob format, base address, MSW	NO
52	Blob Base, LSW	For blob format, base address, LSW	NO

9.1.2.4.7 NAND Boot Parameter Table

Table 9-21. 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
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

9.1.2.4.8 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 9-22

Table 9-22. 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

Table 9-22. DDR3 Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
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	lpDfrNvmTiming	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
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

9.1.2.5 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

9.1.3 System PLL Settings

The PLL default settings are determined by the BOOTMODE[7:5] bits. Table 9-23 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 PASS PLL is also shown. The PASS 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 9-10 for details on configuring Ethernet boot mode. The output from the PASS PLL goes through an on-chip divider to reduce the frequency before reaching the NETCP. The PASS 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. The DDR3A PLL and NETCP PLL are controlled by chip level MMRs. For details on how to set up the PLL see Section 11.5. For details on the operation of the PLL controller module, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

Table 9-23. System PLL Configuration

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

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

9.1.3.1 ARM CorePac System PLL Settings

The PLL default settings are determined by the BOOTMODE[11:9] bits. Table 9-24 shows 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]

The ARM CorePac PLL is controlled using a PLL controller and a chip-level MMR. For details on how to set up the PLL see Section 11.5. For details on the operation of the PLL controller module, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

Table 9-24. ARM PLL Configuration

BOOTMODE [11:9]	INPUT CLOCK FREQ (MHz)	800 MHz DEVICE			1000 MHz DEVICE			1200 MHz DEVICE		
		PLLD	PLLM	ARM *f*	PLLD	PLLM	ARM *f*	PLLD	PLLM	DSP *f*
0b000	50.00	0	31	800	0	39	1000	0	47	1200
0b001	66.67	0	23	800.04	0	29	1000.05	0	35	1200.06
0b010	80.00	0	19	800	0	24	1000	0	29	1200
0b011	100.00	0	15	800	0	19	1000	0	23	1200
0b100	156.25	0	40	800.78	4	63	1000	24	45	1197.92
0b101	250.00	4	31	800	0	7	1000	4	47	1200
0b110	312.50	7	40	800.78	4	31	1000	2	22	1197.92
0b111	122.88	0	12	798.72	3	64	999.40	0	19	1228.80

9.2 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.

9.2.1 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 9-25 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 6.4.

Table 9-25. Device Configuration Pins

CONFIGURATION PIN	PIN NO.	IPD/IPU(1)	DESCRIPTION
LENDIAN(1)(2)	F29	IPU	Device endian mode (LENDIAN) • 0 = Device operates in big endian mode • 1 = Device operates in little endian mode
BOOTMODE[15:0](1)(2)	B31, E32, A31, F30, E30, F31, G30, A30, C30, D30, E29, B29, A35, D29, B30, F29	IPD	Method of boot • See Section 9.1.2 for more details. • See the *KeyStone II Architecture ARM Bootloader User's Guide* (SPRUHJ3) for detailed information on boot configuration.
AVSIFSEL[1:0](1)(2)	M1, M2	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

(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 6.4.
(2) These signal names are the secondary functions of these pins.

9.2.2 Peripheral Selection After Device Reset

Several of the peripherals on the 66AK2L06 are controlled by the Power Sleep Controller (PSC). By default, the PCIe, FFTC, and AIF2 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).

9.2.3 Device State Control Registers

The 66AK2L06 device has a set of registers that are used to control the status of its peripherals. These registers are shown in Table 9-26.

Table 9-26. 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 9.2.3.4
0x0262001C	0x0262001F	4B	Reserved	
0x02620020	0x02620023	4B	DEVSTAT	See Section 9.2.3.1
0x02620024	0x02620037	20B	Reserved	
0x02620038	0x0262003B	4B	KICK0	See Section 9.2.3.5
0x0262003C	0x0262003F	4B	KICK1	
0x02620040	0x02620043	4B	DSP_BOOT_ADDR0	The boot address for C66x CorePac in secure mode.
0x02620044	0x02620047	4B	DSP_BOOT_ADDR1	
0x02620048	0x0262004B	4B	DSP_BOOT_ADDR2	
0x0262004C	0x0262004F	4B	DSP_BOOT_ADDR3	
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 11.18
0x02620118	0x0262012F	24B	Reserved	
0x02620130	0x02620133	4B	LRSTNMIPINSTAT_CLR	See Section 9.2.3.7
0x02620134	0x02620137	4B	RESET_STAT_CLR	See Section 9.2.3.9
0x02620138	0x0262013B	4B	Reserved	
0x0262013C	0x0262013F	4B	BOOTCOMPLETE	See Section 9.2.3.10
0x02620140	0x02620143	4B	Reserved	
0x02620144	0x02620147	4B	RESET_STAT	See Section 9.2.3.8
0x02620148	0x0262014B	4B	LRSTNMIPINSTAT	See Section 9.2.3.6
0x0262014C	0x0262014F	4B	DEVCFG	See Section 9.2.3.2
0x02620150	0x02620153	4B	PWRSTATECTL	See Section 9.2.3.13
0x02620154	0x02620157	4B	UART0_DISABLE	See Section 9.2.3.12.2
0x02620158	0x0262015B	4B	UART1_DISABLE	
0x0262015C	0x0262015F	4B	UART2_DISABLE	
0x02620160	0x02620163	4B	UART3_DISABLE	
0x02620164	0x02620167	4B	USB_DISABLE	See Section 9.2.3.12.2
0x02620168	0x0262016B	4B	Reserved	
0x0262016C	0x0262017F	20B	Reserved	
0x02620180	0x02620183	4B	SmartReflex Class0	See Section 11.2.4
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	

Table 9-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
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	NMIGR0	See Section 9.2.3.14
0x02620204	0x02620207	4B	NMIGR1	
0x02620208	0x0262020B	4B	NMIGR2	
0x0262020C	0x0262020F	4B	NMIGR3	
0x02620210	0x02620213	4B	Reserved	
0x02620214	0x02620217	4B	Reserved	
0x02620218	0x0262021B	4B	Reserved	
0x0262021C	0x0262021F	4B	Reserved	
0x02620220	0x0262023F	32B	Reserved	
0x02620240	0x02620243	4B	IPCGR0	See Section 9.2.3.15
0x02620244	0x02620247	4B	IPCGR1	
0x02620248	0x0262024B	4B	IPCGR2	
0x0262024C	0x0262024F	4B	IPCGR3	
0x02620250	0x02620253	4B	IPCGR4	
0x02620254	0x02620257	4B	IPCGR5	
0x02620258	0x0262025B	4B	IPCGR6	
0x0262025C	0x0262025F	4B	IPCGR7	
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 9.2.3.17
0x02620280	0x02620283	4B	IPCAR0	See Section 9.2.3.16
0x02620284	0x02620287	4B	IPCAR1	
0x02620288	0x0262028B	4B	IPCAR2	
0x0262028C	0x0262028F	4B	IPCAR3	
0x02620290	0x02620293	4B	IPCAR4	
0x02620294	0x02620297	4B	IPCAR5	
0x02620298	0x0262029B	4B	IPCAR6	
0x0262029C	0x0262029F	4B	IPCAR7	
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 9.2.3.18
0x026202C0	0x026202D7	24B	Reserved	

Table 9-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x026202D8	0x026202DB	4B	TINPSEL0	See Section 9.2.3.19
0x026202DC	0x026202DF	4B	Reserved	
0x026202E0	0x026202E3	4B	TINPSEL2	See Figure 9-33
0x026202E4	0x026202E7	4B	TINPSEL3	See Figure 9-34
0x026202E8	0x026202EB	4B	TINPSEL4	See Figure 9-35
0x026202EC	0x026202EF	4B	Reserved	
0x026202F0	0x026202F3	4B	Reserved	
0x026202F4	0x026202F7	4B	Reserved	
0x026202F8	0x026202FB	4B	TOUTPSEL0	See Section 9.2.3.20
0x026202FC	0x026202FF	4B	TOUTPSEL1	
0x02620300	0x02620303	4B	Reserved	
0x02620304	0x02620307	4B	Reserved	
0x02620308	0x0262030B	4B	RSTMUX0	See Section 9.2.3.21
0x0262030C	0x0262030F	4B	RSTMUX1	
0x02620310	0x02620313	4B	RSTMUX2	
0x02620314	0x02620317	4B	RSTMUX3	
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	Reserved	
0x02620334	0x02620337	4B	Reserved	
0x02620338	0x0262034F	4B	Reserved	
0x02620350	0x02620353	4B	MAINPLLCTL0	See Section 11.5
0x02620354	0x02620357	4B	MAINPLLCTL1	
0x02620358	0x0262035B	4B	PASSPLLCTL0	See Section 9.1.3
0x0262035C	0x0262035F	4B	PASSPLLCTL1	
0x02620360	0x02620363	4B	DDR3APLLCTL0	See Section 11.6
0x02620364	0x02620367	4B	DDR3APLLCTL1	
0x02620368	0x0262036B	4B	Reserved	
0x0262036C	0x0262036F	4B	Reserved	
0x02620370	0x02620373	4B	ARMPLLCTL0	See Section 9.1.3.1
0x02620374	0x02620377	4B	ARMPLLCTL1	
0x02620378	0x0262037B	4B	DFEPLLCTL0	See Section 11.8.1
0x0262037C	0x0262037F	4B	DFEPLLCTL1	
0x02620380	0x0262039B	124B	Reserved	
0x0262039C	0x0262039F	4B	Reserved	
0x02620400	0x02620403	4B	ARMENDIAN_CFG0_0	See Section 9.2.3.22
0x02620404	0x02620407	4B	ARMENDIAN_CFG0_1	
0x02620408	0x0262040B	4B	ARMENDIAN_CFG0_2	
0x0262040C	0x026205FF	62B	Reserved	
0x02620600	0x0262068F	256B	Reserved	
0x02620690	0x02620693	4B	PIN_MUXCTL0	See Section 9.2.3.3
0x02620694	0x02620697	4B	PIN_MUXCTL1	
0x02620698	0x0262069B	4B	PIN_MUXCTL2	

Table 9-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x0262069C	0x026206E7	76B	Reserved	
0x026206E8	0x026206EB	4B	DFE_CLKDIV_CTL	See Section 11.8.2
0x026206EC	0x026206EF	4B	DFE_CLKSYNC_CTL	See Section 11.8.3
0x026206F0	0x026206F7	8B	Reserved	
0x026206F8	0x026206FB	4B	RSTISOCTL	See Section 9.2.3.24
0x026206FC	0x026206FF	4B	IQN_RSTREQ_CTL	See Section 9.2.3.23
0x02620700	0x02620703	4B	CHIP_MISC_CTL0	See Section 9.2.3.28
0x02620704	0x0262070F	12B	Reserved	
0x02620710	0x02620713	4B	SYSENDSTAT	See Section 9.2.3.30
0x02620714	0x0262071F	12B	Reserved	
0x02620720	0x0262072F	16B	PLLCLKSEL_STAT	See Section 9.2.3.31
0x02620730	0x02620733	4B	SYNECLK_PINCTL	See Section 9.2.3.32
0x02620734	0x02620737	4B	Reserved	
0x02620738	0x0262074F	24B	USB_PHY_CTL	See Section 9.2.3.33
0x02620750	0x02620843	244B	Reserved	
0x02620844	0x02620847	4B	DSP_BOOT_ADDR0_NS	DSP Boot Address register (in non secure mode). See Section 9.2.3.11
0x02620848	0x0262084B	4B	DSP_BOOT_ADDR1_NS	
0x0262084C	0x0262084F	4B	DSP_BOOT_ADDR2_NS	
0x02620850	0x02620853	4B	DSP_BOOT_ADDR3_NS	
0x02620854	0x02620C7B	1064B	Reserved	
0x02620C7C	0x02620C7F	4B	CHIP_MISC_CTL1	See Section 9.2.3.29
0x02620C80	0x02620C8F	16B	Reserved	
0x02620C90	0x02620C93	4B	DEVSPEED	See Section 9.2.3.22
0x02620C94	0x02620FFF	876B	Reserved	

9.2.3.1 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 9-11. Device Status Register

31 29	28	27	26	25
Reserved	CSISC2_3_MUXSEL	CSISC2_0_CLKCTL	CSISC2_0_MUXSEL	DDR3A_MAP_EN
R-0	R-x	R-x	R-1	R-1

24 20	19	18 17	16 1	0
Reserved	MAINPLLODSEL	AVSIFSEL	BOOTMODE	LENDIAN
R-x	R/W-x	R/W-x	R/W-x xxxx xxxx xxxx xxx	R-x (1)

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

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

Table 9-27. Device Status Register Field Descriptions

Bit	Field	Description
31-29	Reserved	Reserved. Read only, writes have no effect.
28	CSISC2_3_MUXSEL	SerDes Mux selection between SGMII and PCIe0/1. • 0 = CSIS2_3 is assigned to SGMII (lane 2 and 3). • 1 = CSIS2_3 is assigned to PCIe_0 lane 0 and PCIe_1 lane 0.
27	CSISC2_0_CLKCTL	SerDes reference clock selection • 0 = CSIS2_0 and CSIS2_1 use separate SerDes reference clocks. CSIS2_0_REFCLK drives CSIS2_0 and CSIS2_1_REFCLK drives CSIS2_1. • 1 = CSIS2_0 and CSIS2_1 share a single common SerDes reference clock (default) CSIS2_0_REFCLK drives both CSIS2_0 and CSIS2_1.
26	CSISC2_0_MUXSEL	SerDes Mux selection between JESD and AIL. • 0 = CSIS2_0 is assigned to JESD (lane 0 and 1). • 1 = CSIS2_0 is assigned to AIL (lane 0 and 1).
25	DDR3A_MAP_EN	DDR3A mapping enable • 0 = Reserved • 1 = DDR3A memory is accessible in 32b space from ARM, i.e., at 0x0:8000_0000 - 0x0:FFFF_FFFF. DDR3A is also accessible at 0x8:0000_0000 - 0x9:FFFF_FFFF, with the space 0x0:8000_0000 - 0x0:FFFF_FFFF address aliased at 0x8:0000_0000 - 0x8:7FFF_FFFF.
24-20	Reserved	Reserved
19	MAINPLLODSEL	Main PLL Output divider select • 0 = Main PLL output divider needs to be set to 2 by BOOTROM (default) • 1 = Reserved
18-17	AVSIFSEL	AVS interface selection • 00 - AVS 4pin 6bit Dual-Phase VCNTL[5:2] (Default) • 01 - AVS 4pin 4bit Single-Phase VCNTL[5:2] • 10 - AVS 6pin 6bit Single-Phase VCNTL[5:0] • 11 - I^2C
16-1	BOOTMODE	Determines the bootmode configured for the device. For more information on bootmode, see Section 9.1.2 and 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)

9.2.3.2 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 9-12 and described in Table 9-28.

Figure 9-12. Device Configuration Register (DEVCFG)

31	5	4	3	2	1	0
Reserved		PCIESS_1_MODE		PCIESS_0_MODE		SYSCLKOUTEN
R-0		R/W-00		R/W-00		R/W-1

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

Table 9-28. Device Configuration Register Field Descriptions

Bit	Field	Description
31-5	Reserved	Reserved. Read only, writes have no effect.
4-3	PCIESS_1_MODE	Device Type Input of PCIeSS_1 • 00 = Endpoint • 01 = Legacy Endpoint • 10 = Rootcomplex • 11 = Reserved
2-1	PCIESS_0_MODE	Device Type Input of PCIeSS_0 • 00 = Endpoint • 01 = Legacy Endpoint • 10 = Rootcomplex • 11 = Reserved
0	SYSCLKOUTEN	SYSCLKOUT enable • 0 = No clock output • 1 = Clock output enabled (default)

9.2.3.3 Pin Mux Control Register

Pin mux control registers are used to control the various pins that are muxed at the chip level.

9.2.3.3.1 Pin Mux Control 0 Register (PIN_MUXCTL0)

The Pin Mux Control 0 Register is shown in Figure 9-13 and described in Table 9-29.

Figure 9-13. Pin Mux Control 0 Register (PIN_MUXCTL0)

31	8	7	6	5	4	3	2	1	0
Reserved		UART3_EMIFA_SEL		UART2_EMIFA_SEL		Reserved	UART01_SPI2_SEL	DFESYNC_RP1_SEL	AVSIF_SEL
R-0		R/W-0		R/W-00		R, 0	R/W-0	R/W-0	R/W-0

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

Table 9-29. Pin Mux Control 0 Register Field Descriptions

Bit	Field	Description
31-8	Reserved	Reserved. Read only, writes have no effect.
7-6	UART3_EMIFA_SEL	• 00 - Select EMIF A22- A23, CE2-CE3 (Default) • 01 - Select UART3 without flow control (RXD, TXD only) and EMIF CE2 - CE3 • 10 - Reserved • 11 - Select UART3 with flow control

Table 9-29. Pin Mux Control 0 Register Field Descriptions (continued)

Bit	Field	Description
5-4	UART2_EMIFA_SEL	• 00 - Select EMIF A18-A21 (Default) • 01 - Select UART2 without flow control (RXD, TXD only) and EMIF A18-19 • 10 - Reserved • 11 - Select UART2 with flow control
3	Reserved	Reserved. Read only, writes have no effect.
2	UART01_SPI2_SEL	• 0 - Select UART0 and UART1 flow control signals (UART0CTS, UART0RTS, UART1CTS and UART1RTS) (Default) • 1 - Select SPI2 with CS0 (SPI2CLK, SPI2CS0, SPI2SOMI, SPI2SIMO)
1	DFESYNC_RP1_SEL	• 0 - Select DFE_SYNC_IN0 and DFE_SYNC_IN_1(Default) • 1 - Select SRP1CLK and RP1FB
0	AVSIF_SEL	• 0 - Select CVNTL4-5 (Default) • 1 - Select SmartReflex I2C port (VCL and VD pins)

9.2.3.3.2 Pin Mux Control 1 Register (PIN_MUXCTL1)

The Pin Mux Control 1 Register is shown in Figure 9-14 and described in Table 9-30.

Figure 9-14. Pin Mux Control 1 Register (PIN_MUXCTL1)

31	17	16	15	4	3	0
GPIO_EMU_SEL		Rsvd	GPIO_TIMIO_SEL		GPIO_SPI2CS_SEL	
R/W-0		R-0	R/W-0		R/W-0	

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

Table 9-30. Pin Mux Control 1 Register Field Descriptions

Bit	Field	Description
31	GPIO_EMU_SEL[31]	• 0 - Select GPIO31 (Default) • 1 - Select EMU33
30	GPIO_EMU_SEL[30]	• 0 - Select GPIO30(Default) • 1 - Select EMU32
29	GPIO_EMU_SEL[29]	• 0 - Select GPIO29 (Default) • 1 - Select EMU31
28	GPIO_EMU_SEL[28]	• 0 - Select GPIO28(Default) • 1 - Select EMU30
27	GPIO_EMU_SEL[27]	• 0 - Select GPIO27 (Default) • 1 - Select EMU29
26	GPIO_EMU_SEL[26]	• 0 - Select GPIO26 (Default) • 1 - Select EMU28
25	GPIO_EMU_SEL[25]	• 0 - Select GPIO25 (Default) • 1 - Select EMU27
24	GPIO_EMU_SEL[24]	• 0 - Select GPIO24 (Default) • 1 - Select EMU26
23	GPIO_EMU_SEL[23]	• 0 - Select GPIO23 (Default) • 1 - Select EMU25
22	GPIO_EMU_SEL[22]	• 0 - Select GPIO22 (Default) • 1 - Select EMU24
21	GPIO_EMU_SEL[21]	• 0 - Select GPIO21 (Default) • 1 - Select EMU23

Table 9-30. Pin Mux Control 1 Register Field Descriptions (continued)

Bit	Field	Description
20	GPIO_EMU_SEL[20]	• 0 - Select GPIO20 (Default) • 1 - Select EMU22
19	GPIO_EMU_SEL[19]	• 0 - Select GPIO19 (Default) • 1 - Select EMU21
18	GPIO_EMU_SEL[18]	• 0 - Select GPIO18 (Default) • 1 - Select EMU20
17	GPIO_EMU_SEL[17]	• 0 - Select GPIO17 (Default) • 1 - Select EMU19
16	Reserved	Reserved
15	GPIO_TIMIO_SEL[15]	• 0 - Select GPIO15(Default) • 1 - Select TIMO7
14	GPIO_TIMIO_SEL[14]	• 0 - Select GPIO14(Default) • 1 - Select TIMO6
13	GPIO_TIMIO_SEL[13]	• 0 - Select GPIO13(Default) • 1 - Select TIMO5
12	GPIO_TIMIO_SEL[12]	• 0 - Select GPIO12(Default) • 1 - Select TIMO4
11	GPIO_TIMIO_SEL[11]	• 0 - Select GPIO11(Default) • 1 - Select TIMO3
10	GPIO_TIMIO_SEL[10]	• 0 - Select GPIO10(Default) • 1 - Select TIMO2
9	GPIO_TIMIO_SEL[9]	• 0 - Select GPIO9(Default) • 1 - Select TIMI7
8	GPIO_TIMIO_SEL[8]	• 0 - Select GPIO8(Default) • 1 - Select TIM6
7	GPIO_TIMIO_SEL[7]	• 0 - Select GPIO7(Default) • 1 - Select TIMI5
6	GPIO_TIMIO_SEL[6]	• 0 - Select GPIO6(Default) • 1 - Select TIMI4
5	GPIO_TIMIO_SEL[5]	• 0 - Select GPIO5(Default) • 1 - Select TIMI3
4	GPIO_TIMIO_SEL[4]	• 0 - Select GPIO4(Default) • 1 - Select TIMI2
3	GPIO_SPI2CS_SEL[3]	• 0 - Select GPIO3(Default) • 1 - Select SPI2CS4
2	GPIO_SPI2CS_SEL[2]	• 0 - Select GPIO2 (Default) • 1 - Select SPI2CS3
1	GPIO_SPI2CS_SEL[2]	• 0 - Select GPIO1 (Default) • 1 - Select SPI2CS2
0	GPIO_SPI2CS_SEL[0]	• 0 - Select GPIO0 (Default) • 1 - Select SPI2CS1

9.2.3.3.3 Pin Mux Control 2 Register (PIN_MUXCTL2)

The Pin Mux Control 1 Register is shown in Figure 9-14 and described in Table 9-30.

Figure 9-15. Pin Mux Control 2 Register (PIN_MUXCTL2)

31	16	15	0
GPIO_DFEIO_SEL		GPIO_EMIFA_SEL	
R/W-0		R/W-0	

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

Table 9-31. Pin Mux Control 2 Register Field Descriptions

Bit	Field	Description
31	GPIO_DFEIO_SEL[31]	• 0 - Select DFEIO17 (Default) • 1 - Select GPIO63
30	GPIO_DFEIO_SEL[30]	• 0 - Select DFEIO16 (Default) • 1 - Select GPIO62
29	GPIO_DFEIO_SEL[29]	• 0 - Select DFEIO15 (Default) • 1 - Select GPIO61
28	GPIO_DFEIO_SEL[28]	• 0 - Select DFEIO14(Default) • 1 - Select GPIO60
27	GPIO_DFEIO_SEL[27]	• 0 - Select DFEIO13 (Default) • 1 - Select GPIO59
26	GPIO_DFEIO_SEL[26]	• 0 - Select DFEIO12 (Default) • 1 - Select GPIO58
25	GPIO_DFEIO_SEL[25]	• 0 - Select DFEIO11 (Default) • 1 - Select GPIO57
24	GPIO_DFEIO_SEL[24]	• 0 - Select DFEIO10 (Default) • 1 - Select GPIO56
23	GPIO_DFEIO_SEL[23]	• 0 - Select DFEIO9 (Default) • 1 - Select GPIO55
22	GPIO_DFEIO_SEL[22]	• 0 - Select DFEIO8 (Default) • 1 - Select GPIO54
21	GPIO_DFEIO_SEL[21]	• 0 - Select DFEIO7 (Default) • 1 - Select GPIO53
20	GPIO_DFEIO_SEL[20]	• 0 - Select DFEIO6 (Default) • 1 - Select GPIO52
19	GPIO_DFEIO_SEL[19]	• 0 - Select DFEIO5 (Default) • 1 - Select GPIO51
18	GPIO_DFEIO_SEL[18]	• 0 - Select DFEIO4 (Default) • 1 - Select GPIO50
17	GPIO_DFEIO_SEL[17]	• 0 - Select DFEIO3 (Default) • 1 - Select GPIO49
16	GPIO_DFEIO_SEL[16]	• 0 - Select DFEIO2 (Default) • 1 - Select GPIO48
15	GPIO_EMIFA_SEL[15]	• 0 - Select EMIFA17 (Default) • 1 - Select GPIO47
14	GPIO_EMIFA_SEL[14]	• 0 - Select EMIFA16 (Default) • 1 - Select GPIO46

Table 9-31. Pin Mux Control 2 Register Field Descriptions (continued)

Bit	Field	Description
13	GPIO_EMIFA_SEL[13]	• 0 - Select EMIFA15(Default) • 1 - Select GPIO45
12	GPIO_EMIFA_SEL[12]	• 0 - Select EMIFA14 (Default) • 1 - Select GPIO44
11	GPIO_EMIFA_SEL[11]	• 0 - Select EMIFA13 (Default) • 1 - Select GPIO43
10	GPIO_EMIFA_SEL[10]	• 0 - Select EMIFA10 (Default) • 1 - Select GPIO42
9	GPIO_EMIFA_SEL[9]	• 0 - Select EMIFA09 (Default) • 1 - Select GPIO41
8	GPIO_EMIFA_SEL[8]	• 0 - Select EMIFA08 (Default) • 1 - Select GPIO40
7	GPIO_EMIFA_SEL[7]	• 0 - Select EMIFA07 (Default) • 1 - Select GPIO39
6	GPIO_EMIFA_SEL[6]	• 0 - Select EMIFA06 (Default) • 1 - Select GPIO38
5	GPIO_EMIFA_SEL[5]	• 0 - Select EMIFA05 (Default) • 1 - Select GPIO37
4	GPIO_TIMIO_SEL[4]	• 0 - Select EMIFA04 (Default) • 1 - Select GPIO36
3	GPIO_TIMIO_SEL[3]	• 0 - Select EMIFA03 (Default) • 1 - Select GPIO35
2	GPIO_TIMIO_SEL[2]	• 0 - Select EMIFA02 (Default) • 1 - Select GPIO34
1	GPIO_TIMIO_SEL[2]	• 0 - Select EMIFA01 (Default) • 1 - Select GPIO33
0	GPIO_TIMIO_SEL[0]	• 0 - Select EMIFA00 (Default) • 1 - Select GPIO32

9.2.3.4 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 9-16. JTAG ID (JTAGID) Register

31 28	27 12	11 1	0
VARIANT	PART NUMBER	MANUFACTURER	LSB
R-xxxx	R-1011 1001 1010 0111	R-0000 0010 111	R-1

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

Table 9-32. JTAG ID Register Field Descriptions

Bit	Field	Value	Description
31-28	VARIANT	xxxx	Variant value
27-12	PART NUMBER	1011 1001 1010 0111	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.

9.2.3.5 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 9-26 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.

9.2.3.6 LRESETNMI PIN Status (LRSTNMIPINSTAT) Register

The LRSTNMIPINSTAT Register latches the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$. The LRESETNMI PIN Status Register is shown in the figure and table below.

Figure 9-17. LRESETNMI PIN Status Register (LRSTNMIPINSTAT)

31 12	11	10	9	8	7 4	3	2	1	0
Reserved	NMI3	NMI2	NMI1	NMI0	Reserved	LR3	LR2	LR1	LR0
R-0	R-0	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 9-33. LRESETNMI PIN Status Register Field Descriptions

Bit	Field	Description
31-12	Reserved	Reserved
11	NMI3	C66x CorePac3 in NMI
10	NMI2	C66x CorePac2 in NMI
9	NMI1	C66x CorePac1 in NMI

Table 9-33. LRESETNMI PIN Status Register Field Descriptions (continued)

Bit	Field	Description
8	NMI0	C66x CorePac0 in NMI
7-4	Reserved	Reserved
3	LR3	C66x CorePac3 in Local Reset
2	LR2	C66x CorePac2 in Local Reset
1	LR1	C66x CorePac1 in Local Reset
0	LR0	C66x CorePac0 in Local Reset

9.2.3.7 LRESETNMI PIN Status Clear (LRSTNMIPINSTAT_CLR) Register

The LRSTNMIPINSTAT_CLR Register clears the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL[2:0]. The LRESETNMI PIN Status Clear Register is shown in the figure and table below.

Figure 9-18. LRESETNMI PIN Status Clear Register (LRSTNMIPINSTAT_CLR)

31	12	11	10	9	8	7	4	3	2	1	0
Reserved		NMI3	NMI2	NMI1	NMI0	Reserved		LR3	LR2	LR1	LR0
R-0		R-0	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 9-34. LRESETNMI PIN Status Clear Register Field Descriptions

Bit	Field	Description
31-12	Reserved	Reserved
11	NMI3	C66x CorePac3 in NMI Clear
10	NMI2	C66x CorePac2 in NMI Clear
9	NMI1	C66x CorePac1 in NMI Clear
8	NMI0	C66x CorePac0 in NMI Clear
7-4	Reserved	Reserved
3	LR3	C66x CorePac3 in Local Reset Clear
2	LR2	C66x CorePac2 in Local Reset Clear
1	LR1	C66x CorePac1 in Local Reset Clear
0	LR0	C66x CorePac0 in Local Reset Clear

9.2.3.8 Reset Status (RESET_STAT) Register

The Reset Status Register (RESET_STAT) captures the status of local reset (LRx) for each of the cores and also the global device reset (GR). Software can use this information to take different device initialization steps.

- **In case of local reset:** The LRx bits are written as 1 and the GR bit is written as 0 only when the C66x CorePac receives a local reset without receiving a global reset.
- **In case of global reset:** The LRx bits are written as 0 and 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 9-19. Reset Status Register (RESET_STAT)

31	30	4	3	2	1	0
GR	Reserved		LR3	LR2	LR1	LR0
R-1	R- 0		R-0	R-0	R-0	R-0

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

Table 9-35. 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.
30-4	Reserved	Reserved.
3	LR3	C66x CorePac3 reset status • 0 = C66x CorePac3 has not received a local reset. • 1 = C66x CorePac3 received a local reset.
2	LR2	C66x CorePac2 reset status • 0 = C66x CorePac2 has not received a local reset. • 1 = C66x CorePac2 received a local reset.
1	LR1	C66x CorePac1 reset status • 0 = C66x CorePac1 has not received a local reset. • 1 = C66x CorePac1 received a local reset.
0	LR0	C66x CorePac0 reset status • 0 = C66x CorePac0 has not received a local reset. • 1 = C66x CorePac0 received a local reset.

9.2.3.9 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 9-20. Reset Status Clear Register (RESET_STAT_CLR)

31	30 4	3	2	1	0
GR	Reserved	LR3	LR2	LR1	LR0
R-1	R- 0	R-0	R-0	R-0	R-0

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

Table 9-36. 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.
30-4	Reserved	Reserved.
3	LR3	C66x CorePac3 reset clear bit • 0 = Writing a 0 has no effect. • 1 = Writing a 1 to the LR3 bit clears the corresponding bit in the RESET_STAT register.
2	LR2	C66x CorePac2 reset clear bit • 0 = Writing a 0 has no effect. • 1 = Writing a 1 to the LR2 bit clears the corresponding bit in the RESET_STAT register.
1	LR1	C66x CorePac1 reset clear bit • 0 = Writing a 0 has no effect. • 1 = Writing a 1 to the LR1 bit clears the corresponding bit in the RESET_STAT register.
0	LR0	C66x CorePac0 reset clear bit • 0 = Writing a 0 has no effect. • 1 = Writing a 1 to the LR0 bit clears the corresponding bit in the RESET_STAT register.

9.2.3.10 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 9-21. Boot Complete Register (BOOTCOMPLETE)

31	10	9	8	7	4	3	2	1	0
Reserved		BC9	BC8	Reserved		BC3	BC	BC1	BC0
R,-0		RW-0	RW-0	R,-0		RW-0	RW-0	RW-0	RW-0

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

Table 9-37. Boot Complete Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved.
9	BC9	ARM CorePac 1 boot status • 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
7-4	Reserved	Reserved.
3	BC3	C66x CorePac 3 boot status • 0 = C66x CorePac 3 boot NOT complete • 1 = C66x CorePac 3 boot complete
2	BC2	C66x CorePac2 boot status • 0 = C66x CorePac 2 boot NOT complete • 1 = C66x CorePac 2 boot complete
1	BC1	C66x CorePac1 boot status • 0 = C66x CorePac 1 boot NOT complete • 1 = C66x CorePac 1 boot complete
0	BC0	C66x CorePac0 boot status • 0 = C66x CorePac 0 boot NOT complete • 1 = C66x CorePac 0 boot complete

The BCx bit indicates the boot complete status of the corresponding C66x 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 C66x CorePac sets its corresponding BCx bit immediately before branching to the predefined location in memory.

9.2.3.11 DSP BOOTADDR NS Registers (DSP_BOOT_ADDR0_NS to DSP_BOOT_ADDR3_NS)

DSP_BOOT_ADDR0_NS through DSP_BOOT_ADDR3_NS registers control the boot address for C66x CorePac 0 through C66x CorePac 3 in non-secure mode in either secure device or non-secure device respectively. The DSP_BOOT_ADDRx_NS Register is shown in Figure 9-22 and described in Table 9-38.

Figure 9-22. DSP Non-Secure BOOTADDRx Register (DSP_BOOT_ADDRx_NS)

31	10	9	0
BA_RST_VAL[21:0]		Reserved	
R/W-0		R-0	

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

Table 9-38. DSP Non-Secure BOOTADDRx Register Field Descriptions

Bit	Field	Description
31-10	BA_RST_VAL[21:0]	C66x CorePacx Boot Address Value [21:0]. The value can be read/written by any master or the emulator.
9-0	Reserved	

9.2.3.12 Post Boot Disable Registers

9.2.3.12.1 UARTx Post Boot Disable Registers (UARTx_DISABLE)

UART[3:0] peripherals can be disabled post-boot by setting the corresponding UARTx_DISABLE.

These register bits are “sticky” bits, that is they can be set to “1” only once by the software after POR and they will be cleared to “0” only on POR. In other words, the software is only allowed to switch from peripheral enabled to disabled state once and not from disabled to enabled state. This feature will be used in the secondary bootloader to disable the UARTx peripherals for security precaution.

Figure 9-23. UARTx Post Boot Disable Registers (UARTx_DISABLE)

31	1	0
Reserved		DISABLE
R-0		RW-0

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

Table 9-39. UARTx Post Boot Disable Registers Field Descriptions

Bit	Field	Description
31-1	Reserved	
0	DISABLE	UART port disable • 0 = UART port is enabled (default) • 1 = UART port is disabled

9.2.3.12.2 USB Post Boot Disable Registers (USB_DISABLE)

USB peripheral can be disabled post-boot by setting the corresponding USB_DISABLE.

This register bit is “sticky” bit, that is it can be set to “1” only once by the software after POR and it will be cleared to “0” only on POR. In other words, the software is only allowed to switch from peripheral enabled to disabled state once and not from disabled to enabled state. This feature will be used in the secondary bootloader to disable the USB peripherals for security precaution.

Figure 9-24. USB Post Boot Disable Registers (USB_DISABLE)

31	1	0
Reserved		DISABLE
R-0		RW-0

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

Table 9-40. USB Post Boot Disable Registers Field Descriptions

Bit	Field	Description
31-1	Reserved	
0	DISABLE	USB port disable • 0 = USB port is enabled (default) • 1 = USB port is disabled

9.2.3.13 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 9-25 and described in Table 9-41.

Figure 9-25. 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 9-41. 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. See the *KeyStone Architecture DSP Bootloader User's Guide* (SPRUGY5).
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

9.2.3.14 NMI Event Generation to C66x CorePac (NMIGRx) Register

NMIGRx registers generate NMI events to the corresponding C66x CorePac. The 66AK2L06 has four NMIGRx registers (NMIGR0 through NMIGR3). The NMIGR0 register generates an NMI event to C66x CorePac0. Writing a 1 to the NMIG field generates an NMI pulse. Writing a 0 has no effect and Reads return 0 and have no other effect. The NMI event generation to the C66x CorePac is shown in Figure 9-26 and described in Table 9-42.

Figure 9-26. NMI Generation Register (NMIGRx)

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

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

Table 9-42. NMI Generation Register Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	NMIG	Reads return 0 Writes: • 0 = No effect • 1 = Creates NMI pulse to the corresponding C66x CorePac — C66x CorePac0 for NMIGR0, etc.

9.2.3.15 IPC Generation (IPCGRx) Registers

The IPCGRx Registers facilitate inter-C66x CorePac interrupts.

The 66AK2L06 device has six IPCGRx registers (IPCGR0 through IPCGR3 and 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:

- C66x CorePacx (0 <= x <= 3)
- ARM CorePac core (8<=x<=9)

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 9-27 and described in Table 9-43.

Figure 9-27. 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 9-43. 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-DSP interrupt.

9.2.3.16 IPC Acknowledgment (IPCARx) Registers

The IPCARx registers facilitate inter-CorePac interrupt acknowledgment.

The 66AK2L06 device has six 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 9-28. 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 9-44. 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

9.2.3.17 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 9-29 and described in Table 9-45.

Figure 9-29. 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 9-45. 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)

9.2.3.18 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 9-30 and described in Table 9-46.

Figure 9-30. 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 9-46. 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

9.2.3.19 Timer Input Selection Register (TINPSELx)

The Timer Input Selection Register selects timer inputs and is shown in the figures and table below.

Figure 9-31. Timer Input Selection Register (TINPSEL0) for Timer 0-Timer3

31	30	28	27	26	24
Reserved	TINPHSEL3		Reserved	TINPLSEL3	
R-0	RW-0		R-0	RW-0	
23	22	20	19	18	16
Reserved	TINPHSEL2		Reserved	TINPLSEL2	
R-0	RW-0		R-0	RW-0	
15	14	12	11	10	8
Reserved	TINPHSEL1		Reserved	TINPLSEL1	
R-0	RW-0		R-0	RW-0	
7	6	4	3	2	0
Reserved	TINPHSEL0		Reserved	TINPLSEL0	
R-0	RW-0		R-0	RW-0	

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

Figure 9-32. Timer Input Selection Register (TINPSEL1) for Timer 4-Timer7

31	30	28	27	26	24
Reserved	TINPHSEL7		Reserved	TINPLSEL7	
R-0	RW-0		R-0	RW-0	
23	22	20	19	18	16
Reserved	TINPHSEL6		Reserved	TINPLSEL6	
R-0	RW-0		R-0	RW-0	
15	14	12	11	10	8
Reserved	TINPHSEL5		Reserved	TINPLSEL5	
R-0	RW-0		R-0	RW-0	
7	6	4	3	2	0
Reserved	TINPHSEL4		Reserved	TINPLSEL4	
R-0	RW-0		R-0	RW-0	

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

Figure 9-33. Timer Input Selection Register (TINPSEL2) for Timer 8-Timer11

31	30	28	27	26	24
Reserved	TINPHSEL11		Reserved	TINPLSEL11	
R-0	RW-0		R-0	RW-0	
23	22	20	19	18	16
Reserved	TINPHSEL10		Reserved	TINPLSEL10	
R-0	RW-0		R-0	RW-0	
15	14	12	11	10	8
Reserved	TINPHSEL9		Reserved	TINPLSEL9	
R-0	RW-0		R-0	RW-0	
7	6	4	3	2	0
Reserved	TINPHSEL8		Reserved	TINPLSEL8	
R-0	RW-0		R-0	RW-0	

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

Figure 9-34. Timer Input Selection Register (TINPSEL3) for Timer 12-Timer15

31	30 28	27	26 24
Reserved	TINPHSEL15	Reserved	TINPLSEL15
R-0	RW-0	R-0	RW-0
23	22 20	19	18 16
Reserved	TINPHSEL14	Reserved	TINPLSEL14
R-0	RW-0	R-0	RW-0
15	14 12	11	10 8
Reserved	TINPHSEL13	Reserved	TINPLSEL13
R-0	RW-0	R-0	RW-0
7	6 4	3	2 0
Reserved	TINPHSEL12	Reserved	TINPLSEL12
R-0	RW-0	R-0	RW-0

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

Figure 9-35. Timer Input Selection Register (TINPSEL4) for Timer 16-Timer17

31 16
Reserved
R-0

15	14 12	11	10 8
Reserved	TINPHSEL17	Reserved	TINPLSEL17
R-0	RW-0	R-0	RW-0
7	6 4	3	2 0
Reserved	TINPHSEL16	Reserved	TINPLSEL16
R-0	RW-0	R-0	RW-0

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

Table 9-47. Timer Input Selection Field Description

Bit	Field	Description
31-15 11 7 3	Reserved	
14-12 6-4	TINPHSELx5	Input select for TIMER16 high and TIMER17 high.
10-8 2-0	TINPLSELx5	Input select for TIMER16 low and TIMER17 low.

9.2.3.20 Timer Output Selection Register (TOUTPSELx)

The control register TOUTSELx handles the timer output selection and is shown in Figure 9-36 and Figure 9-37 and described in Table 9-48.

Figure 9-36. Timer Output Selection 0 Register (TOUTPSEL0)

31 30	29 24	23 22	21 16	15 14	13 8	7 6	5 0
Reserved	TOUTPSEL3	Reserved	TOUTPSEL2	Reserved	TOUTPSEL1	Reserved	TOUTPSEL0
R-0	RW-010001	R-0	RW-010000	R-0	RW-010101	R-0	RW-010100

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

Figure 9-37. Timer Output Selection1 Register (TOUTPSEL1)

31 30	29 24	23 22	21 16	15 14	13 8	7 6	5 0
Reserved	TOUTPSEL7	Reserved	TOUTPSEL5	Reserved	TOUTPSEL4	Reserved	TOUTPSEL3
R-0	RW-010001	R-0	RW-010000	R-0	RW-010101	R-0	RW-010100

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

Table 9-48. Timer Output Selection Field Description

Bit	Field	Description
31-30 23-22 15-14 7-6	Reserved	Reserved
29-24 21-16 13-8 5-0	TOUTPSELx	Output select for TIMO[x] • 000000: TOUTL0 • 010000: TOUTL8 • 110000: TOUTL8 • 000001: TOUTH0 • 010001: TOUTH8 • 110001: TOUTH8 • 000010: TOUTL1 • 010010: TOUTL9 • 110010: TOUTL9 • 000011: TOUTH1 • 010011: TOUTH9 • 110011: TOUTH9 • 000100: TOUTL2 • 010100: TOUTL10 • 110100: reserved • 000101: TOUTH2 • 010101: TOUTH10 • 110101: reserved • 000110: TOUTL3 • 010110: TOUTL11 • 110110: reserved • 000111: TOUTH3 • 010111: TOUTH11 • 110111: reserved • 001000: reserved • 011000: TOUTL12 • 111000: reserved • 001001: reserved • 011001: TOUTH12 • 111001: reserved • 001010: reserved • 011010: TOUTL13 • 111010: reserved • 001011: reserved • 011011: TOUTH13 • 111011: reserved • 001100: reserved • 011100: TOUTL14 • 111100: reserved • 001101: reserved • 011101: TOUTH14 • 111101: reserved • 001110: reserved • 011110: TOUTL15 • 111110: reserved • 001111: reserved • 011111: TOUTH15 • 111111: reserved

9.2.3.21 Reset Mux (RSTMUXx) Register

Software controls the Reset Mux block through the reset multiplex registers using RSTMUX0 through RSTMUX3 for each of the C66x CorePacs and RSTMUX8 and RSTMUX9 for the ARM CorePac on the device. These registers are located in Bootcfg memory space. The Reset Mux Register is shown in the figure and table below.

Figure 9-38. 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 9-49. Reset Mux Register 0..3 (RSTMUX0-RSTMUX3) 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

Table 9-49. Reset Mux Register 0..3 (RSTMUX0-RSTMUX3) Field Descriptions (continued)

Bit	Field	Description
7-5	DELAY	Delay cycles between NMI & local reset • 000b = 256 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 001b = 512 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 010b = 1024 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 011b = 2048 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 100b = 4096 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b (default) • 101b = 8192 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 110b = 16384 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b • 111b = 32768 SYSCLK1/6 cycles delay between NMI & local reset, when OMODE = 100b
4	EVTSTAT	Event status • 0 = No event received (Default) • 1 = WD timer event received by Reset Mux block
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 = WD Timer Event input to the Reset Mux block causes local reset input to C66x CorePac. • 011b = WD Timer Event input to the Reset Mux block causes NMI input to C66x CorePac. • 100b = WD Timer Event input to the Reset Mux block causes NMI input followed by local reset input to C66x CorePac. Delay between NMI and local reset is set in DELAY bit field. • 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

Table 9-50. Reset Mux Register 8 and 9 (RSTMUX8-RSTMUX9) 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 NMI & local 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 9-50. Reset Mux Register 8 and 9 (RSTMUX8-RSTMUX9) 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 66AK2L06. 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

9.2.3.22 Device Speed (DEVSPEED) Register

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

Figure 9-39. 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 9-51. 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 = Reserved • 0b0000 0000 1xxx = Reserved • 0b0000 0001 xxxx = Reserved • 0b0000 001x xxxx = Reserved • 0b0000 01xx xxxx = Reserved • 0b0000 1xxx xxxx = 1200 MHz • 0b0001 xxxx xxxx= 1000 MHz • 0b001x xxxx xxxx = 800 MHz
15-12	Reserved	Reserved. Read only

Table 9-51. 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 = Reserved • 0b0000 0000 1xxx = Reserved • 0b0000 0001 xxxx = Reserved • 0b0000 001x xxxx = Reserved • 0b0000 01xx xxxx = Reserved • 0b0000 1xxx xxxx = 1200 MHz • 0b0001 xxxx xxxx= 1000 MHz • 0b001x xxxx xxxx = 800 MHz

9.2.3.23 IQN Reset Request Control (IQN_RSTREQ_CTL) Register

Incoming CPRI packet via IQN-AIL lanes can request for a device reset. This reset request is controlled and the status is latched by the IQN_RSTREQ_CTL register.The register is shown below.

Figure 9-40. IQN Reset Request Control Register (IQN_RSTREQ_CTL)

31 10	9	8 5	4	3 2	1	0
Reserved	EVTSTATCLR	Reserved	EVTSTAT	Reserved	EN	Reserved
R-0	RC-0	R-0	R-0	R-0	RW-0	R-0

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

Table 9-52. IQN Reset Request Control Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved. Read only
9	EVTSTATCLR	Event status clear • 0 = writing 0 to this bit has not effect • 1= writing 1 to this bit clears EVTSTAT
8-5	Reserved	Reserved. Read only
4	EVTSTAT	Event status. This bit is cleared only on $\overline{\text{POR}}$ and $\overline{\text{RESETFULL}}$. • 0 = No IQN reset event occurred. • 1= IQN reset received by SoC. This bit is set only when the request is enabled by setting the EN bit to 1.
3-2	Reserved	Reserved. Read only
1	EN	IQN reset request enable. • 0 = IQN reset request is masked (disabled). • 1= IQN reset request is enabled. The reset signal from IQN causes device reset signal to the PLL controller.
0	Reserved	Reserved. Read only

9.2.3.24 SerDes Reset Isolation (RSTISOCTL) Register

This register is used to control the SerDes reset isolation for AIL and SGMII lanes. When AILRSTISOEN bit is set to ‘1’, it indicates that the AIL serdes lanes must not be reset during Non-POR chip resets. When SGMIIRSTISOEN bit is set to ‘1’, it indicates that the SGMII serdes lanes must not be reset during Non-POR chip resets.

Figure 9-41. SerDes Reset Isolation (RSTISOCTL)

31	2	1	0
Reserved		SGMIIRSTISOEN	AILRSTISOEN
R-n		RW-0	RW-0

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

Table 9-53. SerDes Reset Isolation Register Field Descriptions

Bit	Field	Description
31-2	Reserved	Reserved. Read only
1	SGMIIRSTISOEN	SGMII SerDes lane reset isolation. • 0 = SGMII SerDes lane reset isolation disabled for all lanes. • 1 = SGMII SerDes lane reset isolation enabled for all lanes, i.e. serdes lanes will not be reset on Non-POR chip resets.
0	AILRSTISOEN	AIL SerDes lane reset isolation. • 0 = AIL SerDes lane reset isolation disabled for all lanes. • 1 = AIL SerDes lane reset isolation enabled for all lanes, i.e. serdes lanes will not be reset on Non-POR chip resets.

9.2.3.25 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 9-42. 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 9-54. ARM Endian Configuration Register 0 Default Values

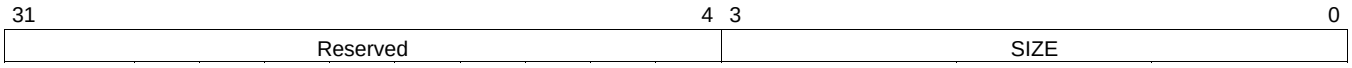
ARM ENDIAN CONFIGURATION REGISTER 0	DEFAULT VALUES
ARMENDIAN_CFG0_0	0x0001C000
ARMENDIAN_CFG1_0	0x00020000
ARMENDIAN_CFG2_0	0x000BC000
ARMENDIAN_CFG3_0	0x00210000
ARMENDIAN_CFG4_0	0x0023A000
ARMENDIAN_CFG5_0	0x00240000
ARMENDIAN_CFG6_0	0x01000000
ARMENDIAN_CFG7_0	0xFFFFFF00

Table 9-55. 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

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

Figure 9-43. ARM Endian Configuration Register 1 (ARMENDIAN_CFGr_1), r=0..7



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

Table 9-56. 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	0x0000000A
ARMENDIAN_CFG6_1	0x00000000
ARMENDIAN_CFG7_1	0x00000000

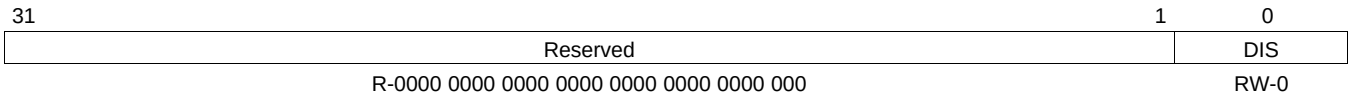
Table 9-57. ARM Endian Configuration Register 1 Field Descriptions

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

9.2.3.27 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 9-44. ARM Endian Configuration Register 2 (ARMENDIAN_CFGr_2), r=0..7



Legend: RW = Read/Write

Table 9-58. 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 9-59. 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

9.2.3.28 Chip Miscellaneous Control (CHIP_MISC_CTL0) Register

Figure 9-45. Chip Miscellaneous Control Register (CHIP_MISC_CTL0)

31	19	18	17
Reserved		USB_PME_EN	Reserved
R-0		RW-0	R-0

16	13	12	11	3	2	0
Reserved		MSMC_BLOCK_PARITY_RST	Reserved		QM_PRIORITY	
R-0		RW-0	RW-0		RW-0	

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

Table 9-60. 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.

9.2.3.29 Chip Miscellaneous Control (CHIP_MISC_CTL1) Register

Figure 9-46. Chip Miscellaneous Control Register (CHIP_MISC_CTL1)

31	12	11	10	0
Reserved		DDR3A_PSC_LOCK_n	Reserved	
R- 0		RW-0	RW-0000000000000	

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

Table 9-61. Chip Miscellaneous Control Register (CHIP_MISC_CTL1) Field Descriptions

Bit	Field	Description
31-12	Reserved	Reserved.
11	DDR3A_PSC_LOCK_n	By default this bit is set to 0. In this case it doesn't allow DDR3A to be clock gated and it will always be in the enabled state. It avoids DDR3A PSC transitioning through its various state-machines. When this bit is set to 1, the DDR3A PSC is un-locked, thereby allowing DDR3A to be reset independently of the rest of the device.
10-0	Reserved	

9.2.3.30 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 $\overline{\text{POR}}$ or $\overline{\text{RESETFULL}}$.

Figure 9-47. 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 9-62. 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 - C66x/System is in Big Endian • 1 - C66x/System is in Little Endian

9.2.3.31 PLL Input Clock Selection Status Register (PLLCLKSEL_STAT)

This register provides a way for reading the status of the PLL input clock selection pins. Any time this register is read, it reflects the status of the clock pin selection.

Figure 9-48. PLL Input Clock Selection Status Register

31	2	1	0
Reserved			CORECLKSELSTAT
R-0			R-00

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

Table 9-63. PLL Input Clock Selection Status Register Descriptions

Bit	Field	Description
31-2	Reserved	Reserved
1-0	CORECLKSELSTAT [1:0]	Reflects the status of the clock pin selection: • 00 - SYSCLK drives the MAIN/ARM/NETCP PLLs (default) • 01 - ALTCORECLK drives the MAIN/ARM/NETCP PLLs • 1x - DDR3ACLK drives the MAIN/ARM/NETCP PLLs

9.2.3.32 SYNECLK_PINCTL Register

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

Figure 9-49. SYNECLK_PINCTL Register

31	3	2	0
Reserved		TSRXCLKOUT0SEL	
R-0		RW-0	

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

Table 9-64. SYNECLK_PINCTL Register Descriptions

Bit	Field	Description
31-3	Reserved	Reserved
2-0	TSRXCLKOUT0SEL	• 000 - SGMII Lane 0 recovered clock (default). • 001 - SGMII Lane 1 recovered clock. • 010 - SGMII Lane 2 recovered clock. • 011 - SGMII Lane 3 recovered clock. • 100 - AIL Lane 0 recovered clock. • 101 - AIL Lane 1 recovered clock. • 110 - AIL Lane 0 divided by 4 recovered clock. • 111 - AIL Lane 1 divided by 4 recovered clock.

9.2.3.33 USB PHY Control (USB_PHY_CTLx) Registers

The following registers control the USB PHY.

Figure 9-50. 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 9-65. 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.

Table 9-65. USB_PHY_CTL0 Register Field Descriptions (continued)

Bit	Field	Description
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.
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 9-51. 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 9-66. 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).

Table 9-66. USB_PHY_CTL1 Register Field Descriptions (continued)

Bit	Field	Description
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.
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 9-52. 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 9-67. 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.

Table 9-67. USB_PHY_CTL2 Register Field Descriptions (continued)

Bit	Field	Description
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.
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 9-53. 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 9-68. 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.
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 9-54. 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 9-69. 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.

Table 9-69. USB_PHY_CTL4 Register Field Descriptions (continued)

Bit	Field	Description
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.
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 9-55. 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 9-70. 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.

10 Device Operating Conditions

10.1 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
	DVDDR	-0.3 V to 1.98 V
	DVDD18	-0.3 V to 2.45 V
	DVDD33	-0.3 V to 3.63 V
	DDR3AVREFSSTL	0.49 × DVDDR to 0.51 × DVDDR
	VDDAHV	-0.3 V to 1.98 V
	VDDALV	-0.3 V to 0.935 V
	VDDUSB	-0.3V to 0.935 V
	AVDDA1, AVDDA2, AVDDA3,AVDDA4, AVDDA5	-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
	DDR3A	-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
	DDR3A	-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
	DDR3A	
	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

10.2 Recommended Operating Conditions[1][2]

			MIN	NOM	MAX	UNIT
CVDD	SR DSP core supply	Initial[3]	0.95	1.05	1.1	V
		1000MHz - Device	SRVnom*0.95[4]	SRVnom	SRVnom*1.05	V
		1200MHz - Device	SRVnom*0.95[4]	SRVnom	SRVnom*1.05	V
CVDD1	DSP Core supply		0.952	1.0	1.047	V
DVDD18	1.8-V supply I/O voltage		1.71	1.8	1.89	V
DVDDR	DDR supply voltage		1.28	1.35/1.5	1.57	V
DDR3VREFSSTL	DDR3A reference voltage		0.49 × DVDDR	0.5 × DVDDR	0.51 × DVDDR	V
VDDAHV	SerDes regulator supply		1.71	1.8	1.89	V
AVDDx[5]	PLL analog, DDR DLL supply		1.71	1.8	1.89	V
VDDALH	SerDes termination supply		0.807	0.85	0.892	V
DVDD33	USB		3.135	3.3	3.465	V
VDDUSB	USB		0.807	0.85	0.892	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
		DDR3A EMIF	VREFSSTL + 0.1			V
V_{IL}[6]	Low-level input voltage	LVCMOS (1.8 V)			0.35 × DVDD18	V
		DDR3A EMIF	-0.3		VREFSSTL - 0.1	V
		I^2C			0.3 × DVDD18	V
T_C	Operating case temperature	Commercial	0		100	°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 DSP 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.

10.3 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
		DDR3A		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
		DDR3A				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
		DDR3A				-8	
		I^2C(5)			(5)		
I_{OL}	Low-level output current [DC]	LVCMOS (1.8 V)				6	mA
		DDR3A				8	
		I^2C				3	
I_{OZ}(6)	Off-state output current [DC]	LVCMOS (1.8 V)		-10		10	µA
		DDR3A		-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.

10.4 Power Supply to Peripheral I/O Mapping

Table 10-1. 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	SYSCLK(P|N) PLL input buffer
			ALTCORECLK(P|N) PLL input buffer
			DDR3ACLK(P|N) PLL input buffer
		LVDS	TSREFCLK(P|N) Input buffer
			RP1CLK(P|N) input buffer
VDDALV		LJCB	SERDES low voltage
VDDAHV	SerDes IO voltage	SerDes/CML	PCIECLK(P|N) SerDes Clock Reference
			SGMIICLK(P|N) SerDes PLL input buffer
			USBCLK(P|M) SerDes Clock Reference
DVDDR	DDR3 supply I/O voltage	DDR3A	All DDR3Amemory 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 RESETs, NMI, control peripheral I/O buffer
			All SmartReflex 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

(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.

11 66AK2L06 Peripheral Information and Electrical Specifications

This chapter covers the various peripherals on the 66AK2L06 device. Peripheral-specific information, timing diagrams, electrical specifications, and register memory maps are described in this chapter.

11.1 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.

11.2 Power Supplies

The following sections describe the proper power-supply sequencing and timing needed to properly power on the 66AK2L06. The various power supply rails and their primary functions are listed in Table 11-1.

Table 11-1. Power Supply Rails on the 66AK2L06

NAME	PRIMARY FUNCTION	VOLTAGE	NOTES
AVDDAx	Core PLL, DDR3 DLL supply voltage	1.8 V	Core PLL, DDR3 DLL supply
CVDD	SmartReflex DSP core supply voltage	0.8 - 1.1 V	DSP variable core supply
CVDD1	DSP core fixed supply voltage	0.95 V	DSP Core fixed supply
DVDDR	DDR3A I/O power supply voltage	DDR3 I/O supply voltage	DDR3A I/O power supply
DVDD18	1.8-V I/O power supply voltage	1.8 V	1.8-V I/O power supply
DVDD33	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
VDDUSB	USB LV PHY power supply voltage	0.85 V	USB LV PHY supply
VPH	Filtered 3.3-V supply voltage	3.3 V	Filtered 3.3-V USB supply
VP, VPTX	Filtered 0.85-V supply voltage	0.85 V	Filtered 0.85-V USB supply
VSS	Ground	GND	Ground

11.2.1 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. DVDDR
4. VDDALV, VDDUSB, VP, VPTX
5. DVDD33, VPH

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. DVDDR
5. VDDALV, VDDUSB, VP, VPTX
6. DVDD33, VPH

The clock input buffers for SYSCLK, ARMCLK, ALTCORECLK, DDR3ACLK, 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 DVDDR are not failsafe and should not be driven high before these voltages are active. Driving these IO cells high before DVDD18 or DVDDR 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 11-8 for more details.

11.2.1.1 Core-Before-IO Power Sequencing

The details of the Core-before-IO power sequencing are defined in Table 11-2. Figure 11-1 shows power sequencing and reset control of the 66AK2L06. 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 11-1) 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 11-2. 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 11-2. 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 DDR3ACLK 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	• DVDDR 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, VDDUSB, VP and VPTX ramp up within 80ms of when DVDDR is valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
5	• DVDD33 supply is ramped up within 80 ms of when VDDALV, VDDUSB, VP and VPTX 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 $\overline{\text{RESETFULL}}$.

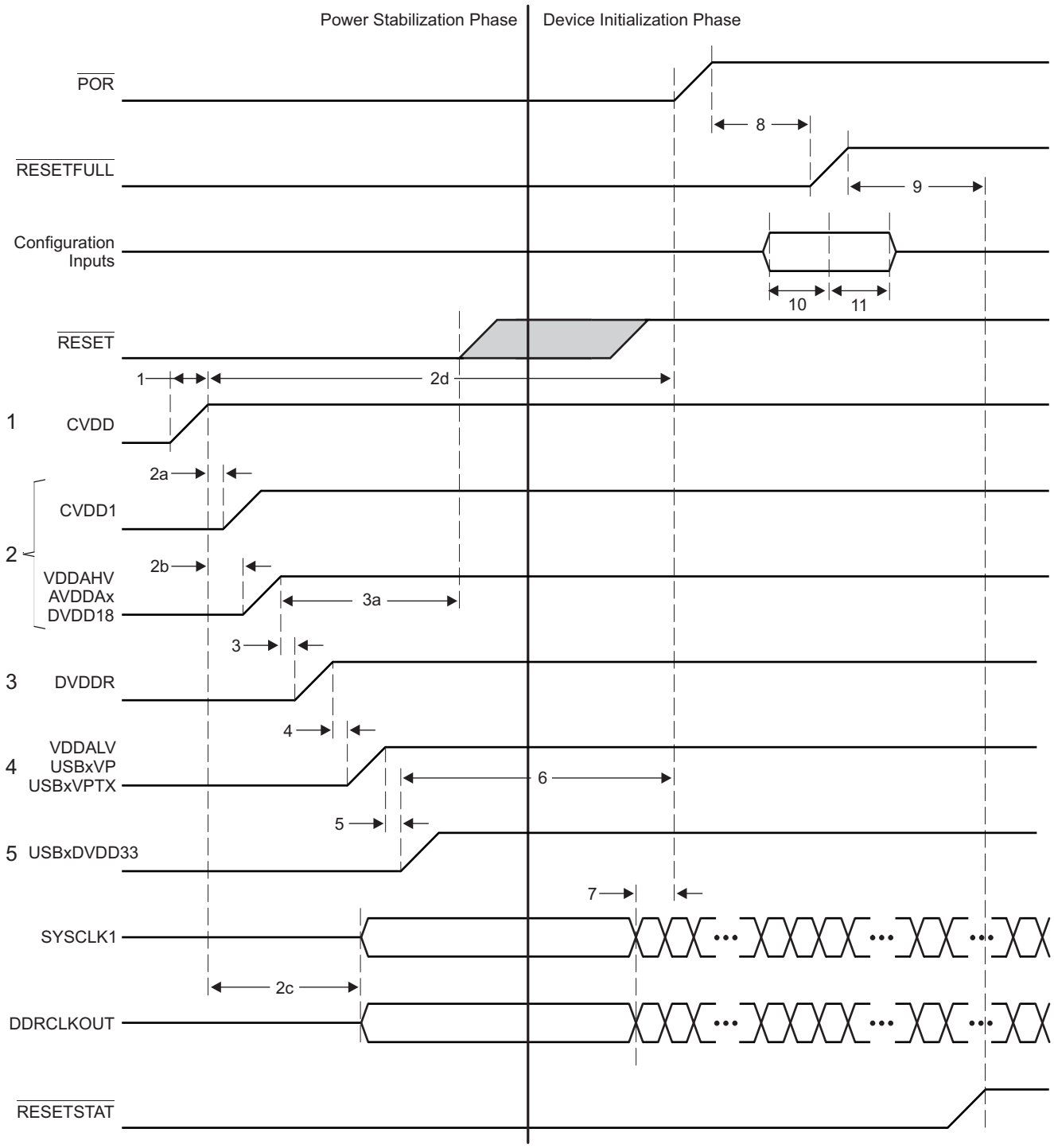


Figure 11-1. Core-Before-IO Power Sequencing

11.2.1.2 IO-Before-Core Power Sequencing

The timing diagram for IO-before-core power sequencing is shown in Figure 11-2 and defined in Table 11-3.

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 11-3. 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 DDR3ACLK 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	• DVDDR 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, VDDUSB, VP and VPTX should ramp up within 80 ms of when DVDDR is valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
6	• DVDD33 supply is ramped up within 80 ms of when VDDALV, VDDUSB, VP and VPTX 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 $\overline{\text{RESETFULL}}$.

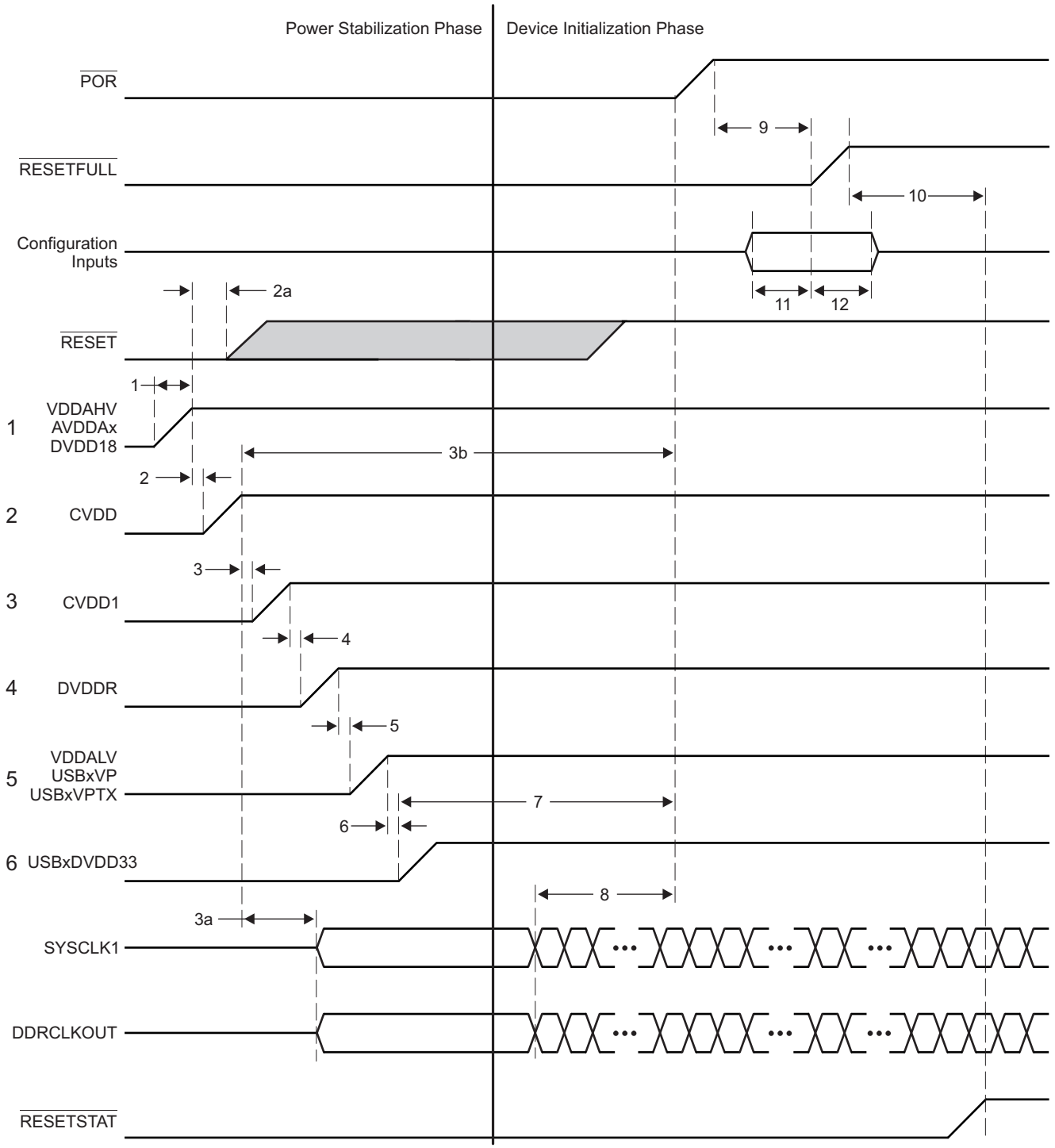


Figure 11-2. IO-Before-Core Power Sequencing

11.2.1.3 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.

11.2.1.4 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 11-4 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 11-4. Clock Sequencing

CLOCK	CONDITION	SEQUENCING
SYSCLK	CORECLKSEL[1:0] = 01 or 10	SYSCLK is not used and should be tied to a static state.
	CORECLKSEL[1:0] = 00	SYSCLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
ALTCORECLK	CORECLKSEL[1:0] = 00 or 10	ALTCORECLK is not used and should be tied to a static state.
	CORECLKSEL[1:0] = 01	ALTCORECLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
DDR3ACLK	CORECLKSEL[1:0] = 00 or 01	DDR3ACLK is not used.
	CORECLKSEL[1:0] = 10	DDR3ACLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.

11.2.2 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.

11.2.3 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).

11.2.4 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 66AK2L06 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 66AK2L06 device.

The 66AK2L06 device supports SmartReflex Class0 and 'Class0 with Temperature Compensation'. To help maximize performance and minimize power consumption of the device, SmartReflex 'Class0 with Temperature Compensation' needs to be implemented. Power consumption is expected to be higher with only Class0 as compared to 'Class0 with Temperature Compensation'. 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 11-5. SmartReflex 4-Pin 6-bit VID Interface Switching Characteristics

(see Figure 11-3)

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 11-10)

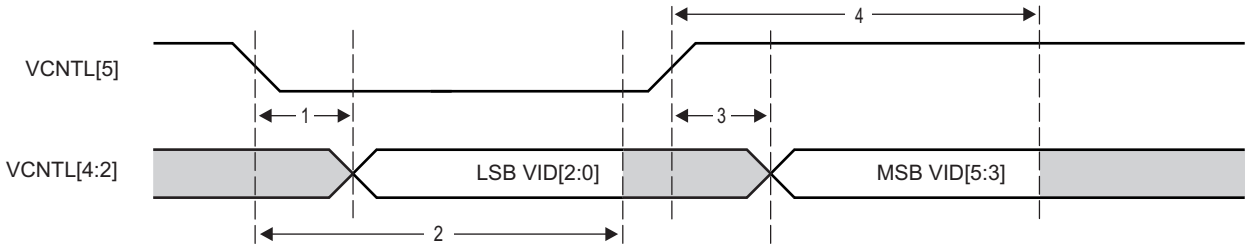


Figure 11-3. SmartReflex 4-Pin 6-Bit VID Interface Timing

11.3 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).

11.3.1 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 66AK2L06 power domains.

Table 11-6. Power Domains

DOMAIN	BLOCK(S)	NOTE	POWER CONNECTION
0	Most peripheral logic (BOOTCFG, EMIF16, I^2C, INTC, GPIO, USB, USIM)	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	PCIe1	Logic can be powered down	Software control
5	DFE__PD2	Logic can be powered down	Software control
6	Smart Reflex		
7	MSMC RAM	MSMC RAM can be powered down	Software control
8	C66x Core 0, L1/L2 RAMs	L2 RAMs can sleep	Software control via C66x CorePac. For details, see the *TMS320C66x DSP CorePac User's Guide* (SPRUGW0).
9	C66x Core 1, L1/L2 RAMs	L2 RAMs can sleep	
10	C66x Core 2, L1/L2 RAMs	L2 RAMs can sleep	
11	C66x Core 3, L1/L2 RAMs	L2 RAMs can sleep	
12	Reserved		
13	Reserved		
14	Reserved		
15	Reserved		
16	EMIF(DDR3A)	Logic can be powered down	Software control
17	Reserved		
18	DFE_PD0	Logic can be powered down	Software control
19	FFTC_0	Logic can be powered down	Software control
20	Reserved		
21	OSR (On Chip Standalone RAM)	RAMs can be powered down	Software control
22	Reserved		
23	Reserved		
24	Reserved		
25	Reserved		
26	Reserved		
27	DFE_PD1	Logic can be powered down	Software control
28	FFTC_1	Logic can be powered down	Software control
29	IQN_AIL	Logic can be powered down	Software control
30	Reserved		
31	ARM CorePac	Logic can be powered down	Software control

11.3.2 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 11-7 shows the 66AK2L06 clock domains.

Table 11-7. Clock Domains

LPSC NUMBER	MODULE(S)	NOTES
0	Shared LPSC for all peripherals other than those listed in this table	Always on
1	DFE_IQN_SYS	Software control
2	USB	Software control
3	EMIF16	Software control
4	Reserved	
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	PCIe0	Software control
11	PCIe1	Software control
12	DFE_PD2	Software control
13	SmartReflex	Always on
14	MSMC RAM	Software control
15	C66x CorePac0	Software control
16	C66x CorePac1	Software control
17	C66x CorePac2	Software control
18	C66x CorePac3	Software control
19	Reserved	
20	Reserved	
21	Reserved	
22	Reserved	
23	DDR3A EMIF	Software control
24	Reserved	
25	Reserved	
26	Reserved	
27	DFE__PD0	Software control
28	FFTC_0	Software control
29	Reserved	
30	Reserved	
31	Reserved	
32	Reserved	
33	Reserved	
34	OSR	Software control
35	Reserved	
36	Reserved	
37	Reserved	
38	Reserved	
39	Reserved	
40	Reserved	
41	Reserved	
42	Reserved	
43	Reserved	
44	Reserved	
45	Reserved	
46	Reserved	

Table 11-7. Clock Domains (continued)

LPSC NUMBER	MODULE(S)	NOTES
47	Reserved	
48	Reserved	
49	Reserved	
50	Reserved	
50	IQN_AIL	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

11.3.3 PSC Register Memory Map

Table 11-8 shows the PSC Register memory map.

Table 11-8. 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
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

Table 11-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
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
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

Table 11-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
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
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

Table 11-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
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
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

Table 11-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
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

11.4 Reset Controller

The reset controller detects the different type of resets supported on the 66AK2L06 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 11-9 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 11.4.8.

Table 11-9. 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.
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.

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

11.4.1 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 POR pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. Also a 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 RESETFULL pin, unlike 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 POR and RESETFULL pins asserted (driven low). While POR is asserted, all pins except RESETSTAT will be set to high-impedance. After the 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 9.2.3).
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 RESETSTAT is driven low, indicating that the device is in reset.
3. POR 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 11.2.1) for the chip-level PLLs to lock.
4. The POR 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.
5. After device initialization is complete, the RESETSTAT pin is de-asserted (driven high). By this time, the DDR3A 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.
6. 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 POR and RESET pins are both de-asserted (driven high). Therefore, in the sequence described above, if the RESET pin is held low past the low period of the POR pin, most of the device will remain in reset. The RESET pin should not be tied to the POR pin.

11.4.2 Hard Reset

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

Hard reset is initiated by the following:

- 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 RESET pin is asserted (driven low) for a minimum of 24 CLKIN1 cycles. During this time, the 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 RESET.

2. Once all logic is reset, RESETSTAT is asserted (driven low) to denote that the device is in reset.
3. The 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 RESETSTAT pin is de-asserted (driven high).

NOTE

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

11.4.3 Soft Reset

A soft reset behaves like a hard reset except that the EMIF16 MMRs, DDR3A EMIF MMRs, PCIe MMRs sticky bits, and external memory content are retained. POR should also remain de-asserted during this time.

Soft reset is initiated by the following:

- 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 DDR3A memory controller registers are ***not*** reset. If the user places the DDR3A SDRAM in self-refresh mode before invoking the soft reset, the DDR3A SDRAM memory content is retained.

During a soft reset, the following occurs:

1. The 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 DDR3A MMRs and PCIe MMRs retain their previous values. Only the DDR3A 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.

11.4.4 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 based on the setting of the CORESEL[2:0] and RSTCFG registers in the PLL Controller. (See Section 11.5.3.8 and Section 7.3.2.)
  - 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).

11.4.5 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).

11.4.6 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

11.4.7 Reset Controller Register

The reset controller registers are part of the PLL Controller MMRs. All 66AK2L06 device-specific MMRs are covered in Section 11.5.3. For more details on these registers and how to program them, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

11.4.8 Reset Electrical Data/Timing

Table 11-10. Reset Timing Requirements(1)

(see Figure 11-4 and Figure 11-5)

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 11-11. Reset Switching Characteristics(1)

(see Figure 11-4 and Figure 11-5)

NO.	PARAMETER		MIN	MAX	UNIT
	$\overline{\text{RESETFULL}}$ Pin Reset				
3	td($\overline{\text{RESETFULL}}$H-$\overline{\text{RESETSTAT}}$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

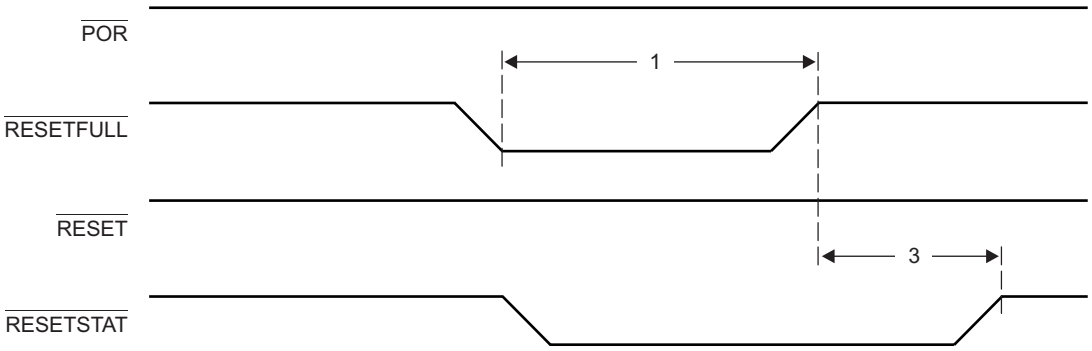


Figure 11-4. RESETFULL Reset Timing

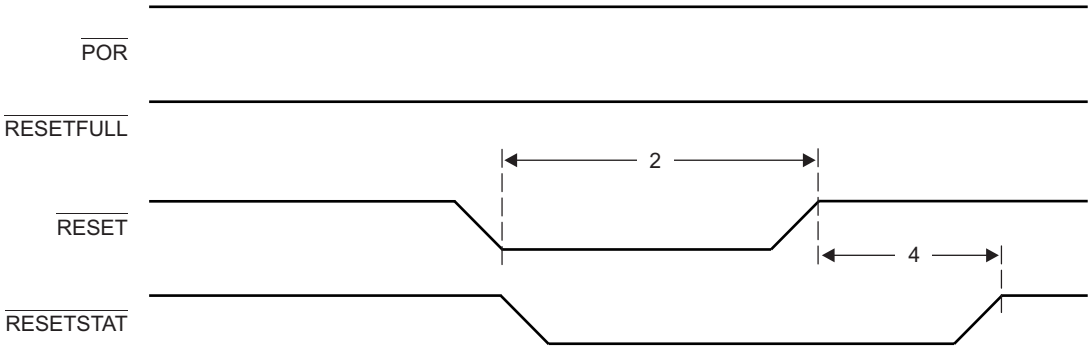


Figure 11-5. Soft/Hard Reset Timing

Table 11-12. Boot Configuration Timing Requirements(1)

(see Figure 11-6)

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

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

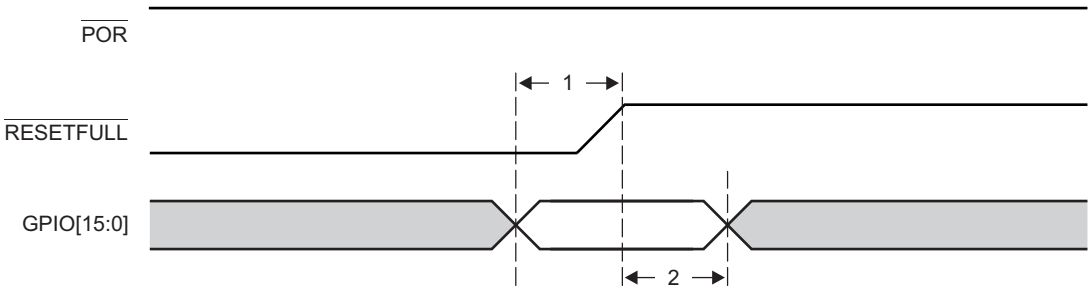


Figure 11-6. Boot Configuration Timing

11.5 PLLs

This section provides a description of the Main PLL, ARM PLL, DDR3A PLL, NETCP PLL, DFE PLL and the PLL Controller (Figure 11-7).

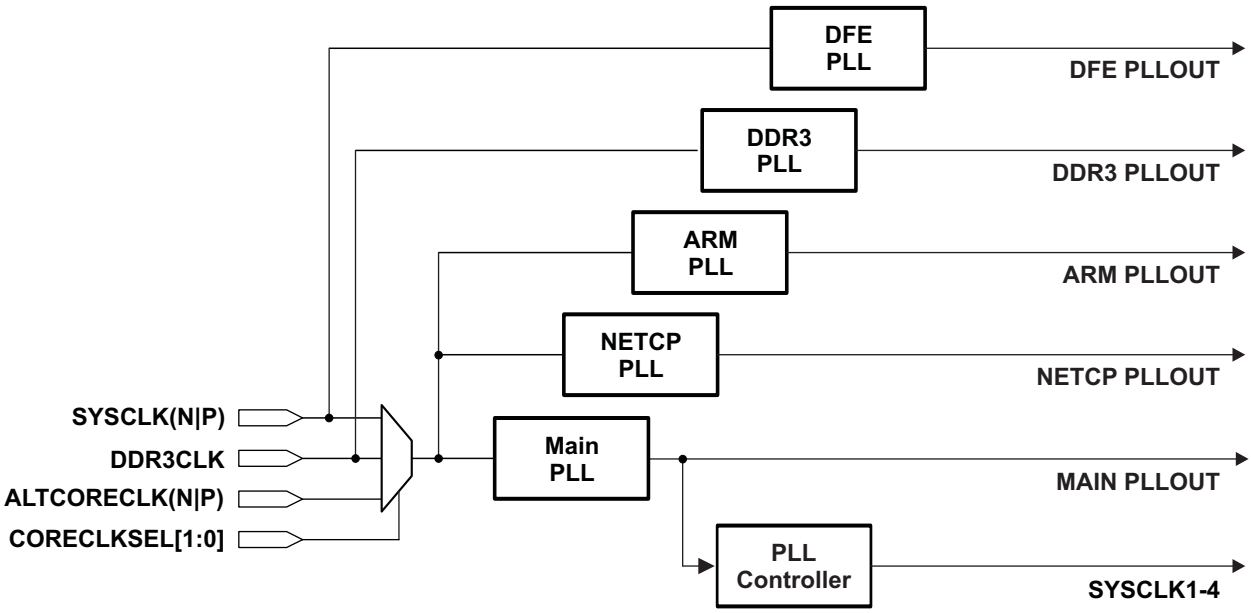


Figure 11-7. K2L PLLs

The ARM PLL, DDR3A PLL NETCP PLL, and DFE PLL are used to provide dedicated clock to the ARM CorePac, DDR3A, NETCP and DFE 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 main PLL, ARM PLL, NETCP PLL and DFE PLL based on boot mode. (See Section 9 for more details.)

11.5.1 Main PLL and PLL Controller

The Main 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 main PLL bypassed. Figure 11-8 shows a block diagram of the Main 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 DDR3A PLL is used to supply clock to DDR3A EMIF logic. This PLL can also be used without programming the PLL Controller. Instead, they can be controlled using the chip-level registers (DDR3APLLCTL0, DDR3APLLCTL1) 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.

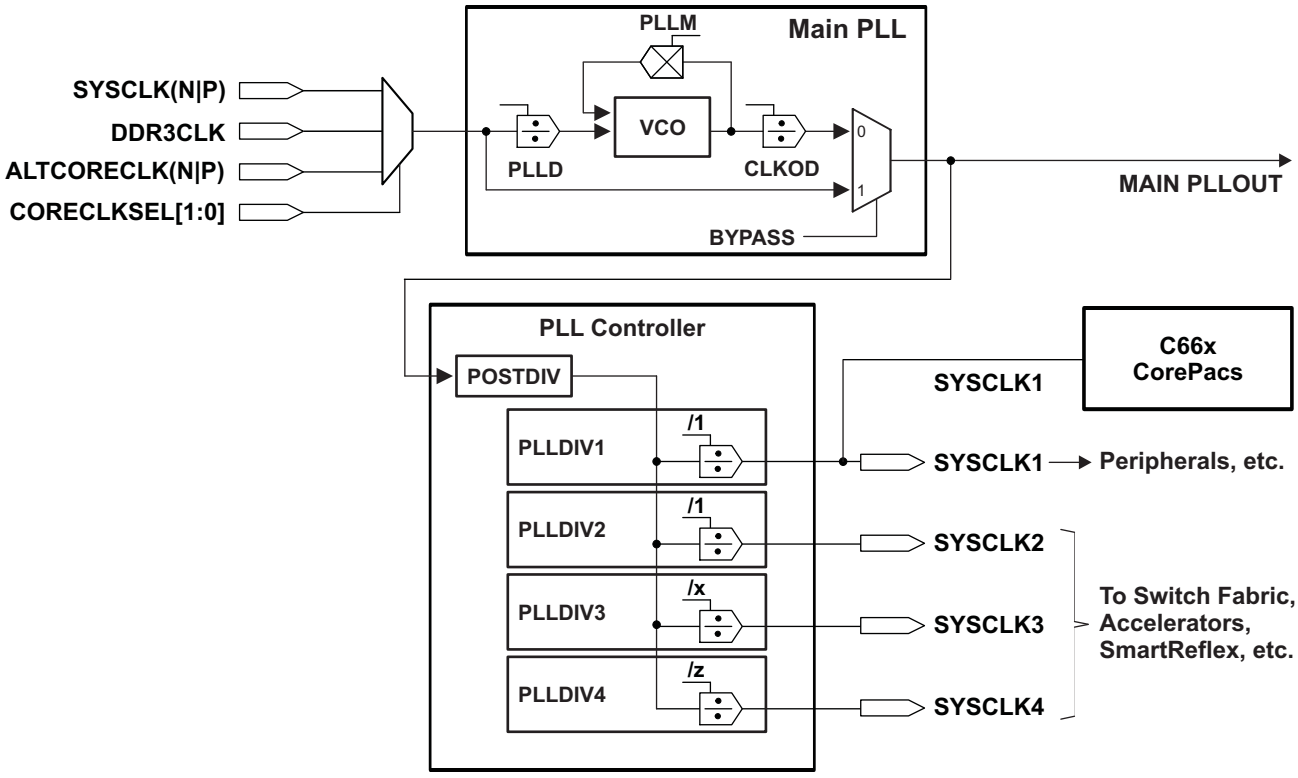


Figure 11-8. Main PLL and PLL Controller

Note that the Main 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 MAINPLLCTL0 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 Main PLL Controller. The Main PLL Controller also controls reset propagation through the chip, clock alignment, and test points. The Main PLL Controller monitors the PLL status and provides an output signal indicating when the PLL is locked.

Main PLL power is supplied externally via the Main 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 SYSCLK rise and fall times should also be observed. For the input clock timing requirements, see Section 11.5.6.

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 66AK2L06 device. The following sections describe the registers that are supported.

11.5.2 Main PLL Controller Device-Specific Information

11.5.2.1 Internal Clocks and Maximum Operating Frequencies

The Main PLL, used to drive the C66x CorePacs, 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. Unlike other PLL, CLKOD functionality of Main PLL is replaced by PLL controller Post-Divider register (POSTDIV). The POSTDIV.RATIO[3:0] and POSTDIV.POSTDEN bits control the post divider ratio and divider enable respectively. PLLM[5:0] input of the Main PLL is controlled by the PLL controller PLLM register.

The Main 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:** Full-rate clock for all C66x CorePacs. 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 Section 1 for the complete list of peripherals supported in your part:

  DFE, FFTC, IQN, EMIF16, USB 3.0, USIM, PCIe, SGMII, GPIO, Timer64, I^2C, SPI, 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:** 1/x-rate clock used to clock the C66x CorePac emulation. 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.

11.5.2.2 Local Clock Dividers

The clock signals from the Main 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 11-13).

Table 11-13. Main 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, /6,	--
	Reserved		
	C66x DSP CorePacs	/1, /2, /3, /4	--
	Chip Interrupt Controllers (CICx)	/6	--
	DFE	/3	
	IQN	/3, /6	
	DDR3 Memory Controller A (also receives clocks from the DDR3A_PLL)	/2	--
	EMIF16	/6	--
	Reserved		
	Fast Fourier Transform Coprocessor (FFTC)	/3	--
	Multicore Navigator Queue Manager	/3	--
	MultiCore Shared Memory Controller (MSMC)	/1	--
	PCI express (PCIe)	/2, /3, /4, /6	--
	Reserved		
	ROM	/6	--
	Serial Gigabit Media Independent Interface (SGMII)	/2, /3, /6, /8	--
	Reserved		
	Reserved		
	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	DFE	--	/6
	CSISC2_0		
	CSISC2_1		
	Boot Config		
	General-Purpose Input/Output (GPIO)		
	I^2C		
	Security Manager		
	Serial Peripheral Interconnect (SPI)		
	TeraNet (CPU /6 domain)		
	Timers		
SYSCLK2 Internal Clock Dividers			
SYSCLK2	SmartReflex	/12, /128	--

11.5.2.3 Module Clock Input

Table 11-7 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.

11.5.2.4 Main PLL Controller Operating Modes

The Main 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 MAINPLLCTL0 Register.

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

11.5.2.5 Main 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 Main PLL reset time value, see Table 11-14.

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 Main PLL lock time is given in Table 11-14.

Table 11-14. Main 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.

11.5.3 PLL Controller Memory Map

The memory map of the Main PLL Controller is shown in Table 11-15. 66AK2L06-specific Main PLL Controller Register definitions can be found in the sections following Table 11-15. 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 Main PLL Controller registers.

Note that only registers documented here are accessible on the 66AK2L06. Other addresses in the Main PLL Controller memory map including the Reserved registers must not be modified. Furthermore, only the bits within the registers described here are supported.

Table 11-15. 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 Main PLL Controller)
00 0231 00E8	RSTCTRL	Software Reset Control Register (Reset Main PLL Controller)
00 0231 00EC	RSTCFG	Reset Configuration Register (Reset Main PLL Controller)
00 0231 00F0	RSISO	Reset Isolation Register (Reset Main 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

Table 11-15. PLL Controller Registers (Including Reset Controller) (continued)

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
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	POSTDIV	PLL Controller Post-Divide Register
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
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

11.5.3.1 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register contains extra fields to control the Main PLL and is shown in Figure 11-9 and described in Table 11-16.

Figure 11-9. 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 11-16. PLL Secondary Control Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Reserved
23	BYPASS	PLL bypass mode: • 0 = PLL is not in BYPASS mode • 1 = PLL is in BYPASS mode
22-19	OUTPUT DIVIDE	Output divider ratio bits • 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 - Fh = ÷5 to ÷16. Divide frequency range: divide frequency by 5 to divide frequency by 80.
18-0	Reserved	Reserved

11.5.3.2 PLL Controller Divider Register (PLLDIV3, and PLLDIV4)

The PLL Controller Divider Registers (PLLDIV3 and PLLDIV4) are shown in Figure 11-10 and described in Table 11-17. The default values of the RATIO field on a reset for PLLDIV3, and PLLDIV4 are different as mentioned in the footnote of Figure 11-10.

Figure 11-10. PLL Controller Divider Register (PLLDIVn)

31 16	15	14 8	7 0
Reserved	Dn(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 11-17. 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 11-10) • 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 11-10) • 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.

11.5.3.3 PLL Controller Clock Align Control Register (ALNCTL)

The PLL Controller Clock Align Control Register (ALNCTL) is shown in Figure 11-11 and described in Table 11-18.

Figure 11-11. 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 11-18. 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*.

11.5.3.4 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 11-12 and described in Table 11-19.

Figure 11-12. 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 11-19. 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.

11.5.3.5 SYSCLK Status Register (SYSTAT)

The SYSCLK Status Register (SYSTAT) shows the status of SYSCLK[4:1]. SYSTAT is shown in Figure 11-13 and described in Table 11-20.

Figure 11-13. 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 11-20. 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

11.5.3.6 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 11-14 and described in Table 11-21.

Figure 11-14. 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 11-21. 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.
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

11.5.3.7 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 11-15 and described in Table 11-22.

Figure 11-15. 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 11-22. 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.

11.5.3.8 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 Main PLL Controller's RSTCTRL Register. By default, these resets are hard resets. The Reset Configuration Register (RSTCFG) is shown in Figure 11-16 and described in Table 11-23.

Figure 11-16. 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 11.5.3.7.

Table 11-23. 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

11.5.3.9 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 Main 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 11-17 and described in Table 11-24.

Figure 11-17. Reset Isolation Register (RSISO)

31	9	8	7	0
Reserved		SRISO	Reserved	
R-0		R/W-0	R-0x0	

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

Table 11-24. 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

11.5.3.10 SerDes Reset Isolation Register (RSTISOCTL)

This register is used to control the SerDes reset isolation for AIL and SGMII lanes.

Figure 11-18. SerDes Reset Isolation Register (RSTISOCTL)

31	2	1	0
Reserved		SGMIISTISOEN	AILRSTISOEN
R-0		RW-0	RW-0

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

Table 11-25. Reset Isolation Register Field Descriptions

Bit	Field	Description
31-2	Reserved	Reserved.
1	SGMIISTISOEN	Isolate SGMII control • 0 = SGMII SerDes lane reset isolation disabled for all lanes. • 1 = SGMII SerDes lane reset isolation enabled for all lanes i.e. SerDes lanes will not be reset on Non-POR chip resets.
0	AILRSTISOEN	Isolate AIL control • 0 = AIL SerDes lane reset isolation disabled for all lanes. • 1 = AIL SerDes lane reset isolation enabled for all lanes i.e. SerDes lanes will not be reset on Non-POR chip resets.

11.5.4 Main PLL Control Registers

The Main PLL uses two chip-level registers (MAINPLLCTL0 and MAINPLLCTL1) along with the Main 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 MAINPLLCTL registers, see Section 9.1.3. See Section 9.2.3.5 for the address location of the KICK registers and their locking and unlocking sequences.

See Figure 11-19 and Table 11-26 for MAINPLLCTL0 details and Figure 11-20 and Table 11-27 for MAINPLLCTL1 details.

Figure 11-19. Main PLL Control Register 0 (MAINPLLCTL0)

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 11-26. Main PLL Control Register 0 (MAINPLLCTL0) Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in MAINPLLCTL0 and MAINPLLCTL1 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. MAINPLLCTL0 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 Main PLL is also controlled by the SECCTL register in the PLL Controller. See Section 11.5.3.1 for 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 11-20. Main PLL Control Register 1 (MAINPLLCTL1)

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 11-27. Main PLL Control Register 1 (MAINPLLCTL1) 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 MAINPLLCTL0 and MAINPLLCTL1 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.

11.5.5 ARM PLL

The ARM PLL generates interface clock for the ARM CorePac. When coming out of power-on reset, ARM PLL is programmed to a valid frequency during the boot configuration process before being enabled and used. ARM PLL power is supplied via the ARM PLL power-supply pin (AVDDA1-AVDDA5). 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.

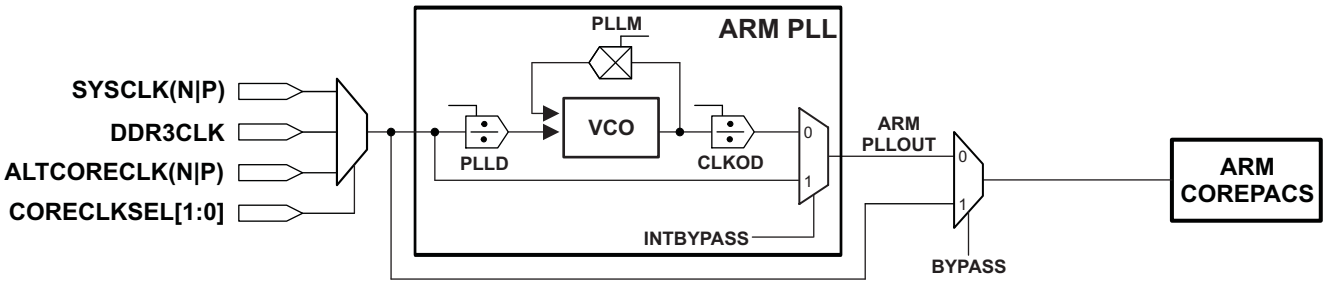


Figure 11-21. ARM PLL Block Diagram

11.5.5.1 ARM PLL Control Registers

The ARM PLL uses two chip-level registers (ARMPLLCTL0 and ARMPLLCTL1) without using the Main PLL Controller like other PLLs for its configuration. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software must go through an un-locking sequence using the KICK0 and KICK1 registers. These registers reset only on a $\overline{\text{POR}}$ reset.

For valid configurable values of the ARMPLLCTL registers, see Section 9.1.3.1. See Section 9.2.3.5 for the address location of the KICK registers and their locking and unlocking sequences.

See Figure 11-22 and Table 11-28 for ARMPLLCTL0 details and Figure 11-23 and Table 11-29 for ARMPLLCTL1 details.

Figure 11-22. ARM PLL Control Register 0 (ARMPLLCTL0)(1)

31	24	23	22	19	18	6	5	0
BWADJ[7:0]		BYPASS	CLKOD		PLLM		PLLD	
RW-0000 1001		RW-1	RW-0001		RW-0000000010011		RW-000000	

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

(1) This register is Reset on $\overline{\text{POR}}$ only. See the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

Table 11-28. ARM PLL Control Register 0 Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in ARMPLLCTL0 and ARMPLLCTL1 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	PLL bypass mode: • 0 = PLL is not in BYPASS mode • 1 = PLL is in BYPASS mode
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 multiplication factor
5-0	PLLD	A 6-bit field that selects the values for the reference divider

Figure 11-23. ARM PLL Control Register 1 (ARMPLLCTL1)

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 11-29. ARM 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 ARMPLLCTL0 and ARMPLLCTL1 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.

See the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2) for the recommended programming sequence.

11.5.6 Main PLL Controller/ARM/DFE/PCIe/USB Clock Input Electrical Data/Timing

Table 11-30. Main PLL Controller/ARM/DFE/PCIe/Shared SerDes/USB/TSREF Clock Input Timing Requirements(1)

(see Figure 11-24 through Figure 11-27)

NO.			MIN	MAX	UNIT
		SYSCLK[P:N]			
1	tc(SYSCLKN)(2)	Cycle time SYSCLKN cycle time	3.25 or 6.51 or 8.138		ns
1	tc(SYSCLKP)(2)	Cycle time SYSCLKP cycle time	3.25 or 6.51 or 8.138		ns
3	tw(SYSCLKN)	Pulse width SYSCLKN high	0.45*tc	0.55*tc	ns
2	tw(SYSCLKN)	Pulse width SYSCLKN low	0.45*tc	0.55*tc	ns
2	tw(SYSCLKP)	Pulse width SYSCLKP high	0.45*tc	0.55*tc	ns
3	tw(SYSCLKP)	Pulse width SYSCLKP low	0.45*tc	0.55*tc	ns
4	tr(SYSCLK_200 mV)	Transition time SYSCLK differential rise time (200 mV)	50	350	ps
4	tf(SYSCLK_200 mV)	Transition time SYSCLK differential fall time (200 mV)	50	350	ps
5	tj(SYSCLKN)	Jitter, peak_to_peak _ periodic SYSCLKN		0.2*tc(SYSCLKN)	ps
5	tj(SYSCLKP)	Jitter, peak_to_peak _ periodic SYSCLKP		0.2*tc(SYSCLKP)	ps

(1) See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for detailed recommendations.
(2) When DFE is used, Cycle time SYSCLKP|N min is 8.138ns

Table 11-30. Main PLL Controller/ARM/DFE/PCIe/Shared SerDes/USB/TSREF Clock Input Timing Requirements[1] (continued)

(see Figure 11-24 through Figure 11-27)

NO.			MIN	MAX	UNIT
ALTCORECLK[P:N]					
1	tc(ALTCORCLKN)	Cycle time ALTCORECLKN cycle time	3.2	25	ns
1	tc(ALTCORECLKP)	Cycle time ALTCORECLKP cycle time	3.2	25	ns
3	tw(ALTCORECLKN)	Pulse width ALTCORECLKN high	0.45*tc(ALTCORECLKN)	0.55*tc(ALTCORECLKN)	ns
2	tw(ALTCORECLKN)	Pulse width ALTCORECLKN low	0.45*tc(ALTCORECLKN)	0.55*tc(ALTCORECLKN)	ns
2	tw(ALTCORECLKP)	Pulse width ALTCORECLKP high	0.45*tc(ALTCORECLKP)	0.55*tc(ALTCORECLKP)	ns
3	tw(ALTCORECLKP)	Pulse width ALTCORECLKP low	0.45*tc(ALTCORECLKP)	0.55*tc(ALTCORECLKP)	ns
4	tr(ALTCORECLK_200 mV)	Transition time ALTCORECLK differential rise time (200 mV)	50	350	ps
4	tf(ALTCORECLK_200 mV)	Transition time ALTCORECLK differential fall time (200 mV)	50	350	ps
5	tj(ALTCORECLKN)	Jitter, peak_to_peak _ periodic ALTCORECLKN		100	ps
5	tj(ALTCORECLKP)	Jitter, peak_to_peak _ periodic ALTCORECLKP		100	ps
SGMIICLK[P:N]					
1	tc(SGMIICLKN)	Cycle time SGMIICLKN cycle time	6.4		ns
1	tc(SGMIICLKP)	Cycle time SGMIICLKP cycle time	6.4		ns
3	tw(SGMIICLKN)	Pulse width SGMIICLKN high	0.45*tc(SGMIICLKN)	0.55*tc(SGMIICLKN)	ns
2	tw(SGMIICLKN)	Pulse width SGMIICLKN low	0.45*tc(SGMIICLKN)	0.55*tc(SGMIICLKN)	ns
2	tw(SGMIICLKP)	Pulse width SGMIICLKP high	0.45*tc(SGMIICLKP)	0.55*tc(SGMIICLKP)	ns
3	tw(SGMIICLKP)	Pulse width SGMIICLKP low	0.45*tc(SGMIICLKP)	0.55*tc(SGMIICLKP)	ns
4	tr(SGMIICLK_250mV)	Transition time SGMIICLK differential rise time (250 mV)	50	350	ps
4	tf(SGMIICLK_250mV)	Transition time SGMIICLK differential fall time (250 mV)	50	350	ps
5	tj(SGMIICLKN)	Jitter, RMS SGMIICLKN		8	ps, RMS
5	tj(SGMIICLKP)	Jitter, RMS SGMIICLKP		8	ps, RMS
PCIECLK[P:N]					
1	tc(PCIECLKN)	Cycle time PCIECLKN cycle time	10		ns
1	tc(PCIECLKP)	Cycle time PCIECLKP cycle time	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[P:N])	Rise time PCIECLK[P:N] differential rise time (10% to 90%)		0.2*tc(PCIECLK[P:N])	ps
4	tf(PCIECLK[P:N])	Fall time PCIECLK[P:N] differential fall time (10% to 90%)		0.2*tc(PCIECLK[P:N])	ps
5	tj(PCIECLKN)	Jitter, RMS PCIECLKN		4	ps, RMS
5	tj(PCIECLKP)	Jitter, RMS PCIECLKP		4	ps, RMS
SHARED_SERDES_0_REFCLK[P:N]					
1	tc(SHARED_SERDES_0_REFCLKN)	Cycle time SHARED_SERDES_0_REFCLKN cycle time	8.138		ns
1	tc(SHARED_SERDES_0_REFCLKP)	Cycle time SHARED_SERDES_0_REFCLKP cycle time	8.138		ns
3	tw(SHARED_SERDES_0_REFCLKN)	Pulse width SHARED_SERDES_0_REFCLKN high	0.45*tc(SHARED_SERDES_0_REFCLKN)	0.55*tc(SHARED_SERDES_0_REFCLKN)	ns
2	tw(SHARED_SERDES_0_REFCLKN)	Pulse width SHARED_SERDES_0_REFCLKN low	0.45*tc(SHARED_SERDES_0_REFCLKN)	0.55*tc(SHARED_SERDES_0_REFCLKN)	ns
2	tw(SHARED_SERDES_0_REFCLKP)	Pulse width SHARED_SERDES_0_REFCLKP high	0.45*tc(SHARED_SERDES_0_REFCLKP)	0.55*tc(SHARED_SERDES_0_REFCLKP)	ns
3	tw(SHARED_SERDES_0_REFCLKP)	Pulse width SHARED_SERDES_0_REFCLKP low	0.45*tc(SHARED_SERDES_0_REFCLKP)	0.55*tc(SHARED_SERDES_0_REFCLKP)	ns
4	tr(SHARED_SERDES_0_REFCLK[P:N])	Rise time SHARED_SERDES_0_REFCLK[P:N] differential rise time (10% to 90%)		0.2*tc(SHARED_SERDES_0_REFCLK[P:N])	ps

Table 11-30. Main PLL Controller/ARM/DFE/PCIe/Shared SerDes/USB/TSREF Clock Input Timing Requirements[1] (continued)

(see Figure 11-24 through Figure 11-27)

NO.			MIN	MAX	UNIT
4	tf(SHARED_SERDES_0_REFCLK[P:N])	Fall time SHARED_SERDES_0_REFCLK[P:N] differential fall time (10% to 90%)		0.2*tc(SHARED_SERDES_0_REFCLK[P:N])	ps
5	tj(SHARED_SERDES_0_REFCLKN)	Jitter, RMS SHARED_SERDES_0_REFCLKN		4	ps, RMS
5	tj(SHARED_SERDES_0_REFCLKP)	Jitter, RMS SHARED_SERDES_0_REFCLKP		4	ps, RMS
SHARED_SERDES_1_REFCLK[P:N]					
1	tc(SHARED_SERDES_1_REFCLKN)	Cycle time SHARED_SERDES_1_REFCLKN cycle time	8.138		ns
1	tc(SHARED_SERDES_1_REFCLKP)	Cycle time SHARED_SERDES_1_REFCLKP cycle time	8.138		ns
3	tw(SHARED_SERDES_1_REFCLKN)	Pulse width SHARED_SERDES_1_REFCLKN high	0.45*tc(SHARED_SERDES_1_REFCLKN)	0.55*tc(SHARED_SERDES_1_REFCLKN)	ns
2	tw(SHARED_SERDES_1_REFCLKN)	Pulse width SHARED_SERDES_1_REFCLKN low	0.45*tc(SHARED_SERDES_1_REFCLKN)	0.55*tc(SHARED_SERDES_1_REFCLKN)	ns
2	tw(SHARED_SERDES_1_REFCLKP)	Pulse width SHARED_SERDES_1_REFCLKP high	0.45*tc(SHARED_SERDES_1_REFCLKP)	0.55*tc(SHARED_SERDES_1_REFCLKP)	ns
3	tw(SHARED_SERDES_1_REFCLKP)	Pulse width SHARED_SERDES_1_REFCLKP low	0.45*tc(SHARED_SERDES_1_REFCLKP)	0.55*tc(SHARED_SERDES_1_REFCLKP)	ns
4	tr(SHARED_SERDES_1_REFCLK[P:N])	Rise time SHARED_SERDES_1_REFCLK differential rise time (10% to 90%)		0.2*tc(SHARED_SERDES_1_REFCLK[P:N])	ps
4	tf(SHARED_SERDES_1_REFCLK[P:N])	Fall time CSISC2_0REFCLK differential fall time (10% to 90%)		0.2*tc(SHARED_SERDES_1_REFCLK[P:N])	ps
5	tj(CSISC2_0REFCLKN)	Jitter, RMS CSISC2_0REFCLKN		4	ps, RMS
5	tj(CSISC2_0REFCLKP)	Jitter, RMS CSISC2_0REFCLKP		4	ps, RMS
USBCLK[P:M]					
1	tc(USBCLKN)	Cycle time USBCLKN cycle time	10	10	ns
1	tc(USBCLKP)	Cycle time USBCLKP cycle time	10	10	ns
3	tw(USBCLKN)	Pulse width USBCLKN high	0.45*tc(USBCLKN)	0.55*tc(USBCLKN)	ns
2	tw(USBCLKN)	Pulse width USBCLKN 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[P:M])	Rise time USBCLK[P:M] differential rise time (10% to 90%)	75	500	ps
4	tf(USBCLK[P:M])	Fall time USBCLK[P:M] differential fall time (10% to 90%)	75	500	ps
5	tj(USBCLKN)	Jitter, RMS USBCLKN		3	ps, RMS
5	tj(USBCLKP)	Jitter, RMS USBCLKP		3	ps, RMS
TSREFCLK[P:N][3]					
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	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[P:N])	Rise time TSREFCLK differential rise time (10% to 90%)	50	350	ps
4	tf(TSREFCLK[P:N])	Fall time TSREFCLK differential fall time (10% to 90%)	50	350	ps
5	tj(TSREFCLKN)	Jitter, RMS TSREFCLKN		5.8	ps, RMS
5	tj(TSREFCLKP)	Jitter, RMS TSREFCLKP		5.8	ps, RMS

(3) TSREFCLK clock input is LVDS compliant

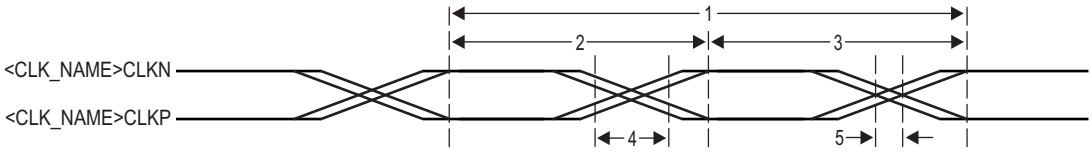


Figure 11-24. Clock Input Timing

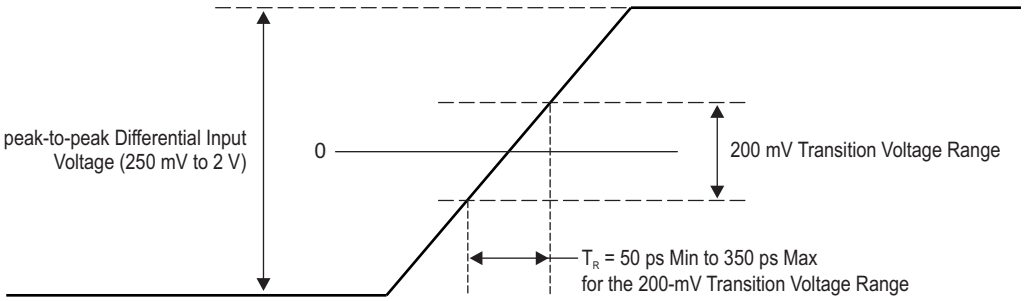


Figure 11-25. Main PLL Transition Time

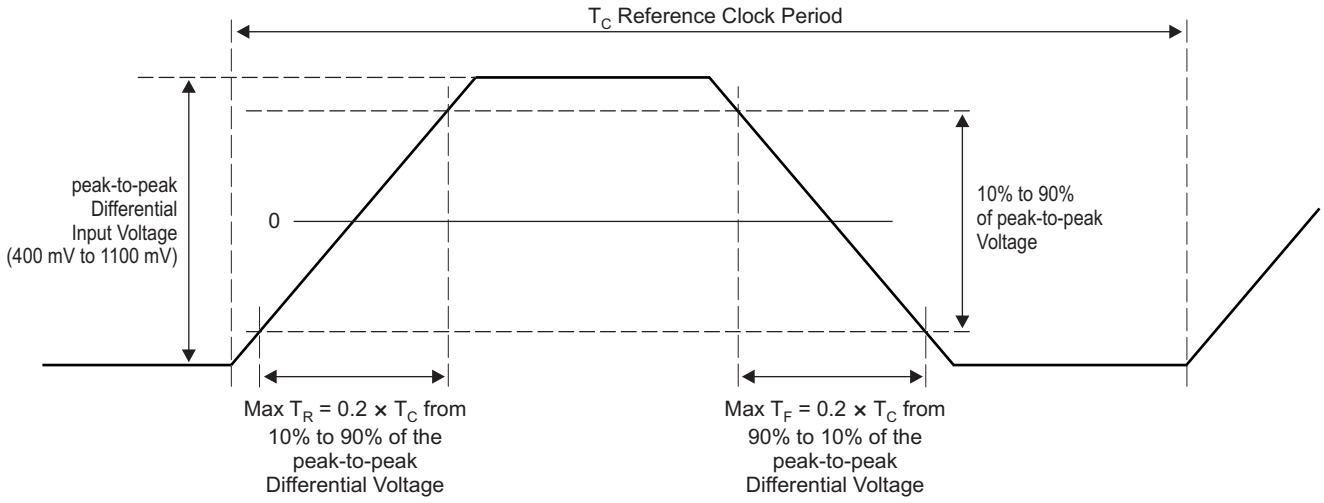


Figure 11-26. Rise and Fall Times (except for USB clock)

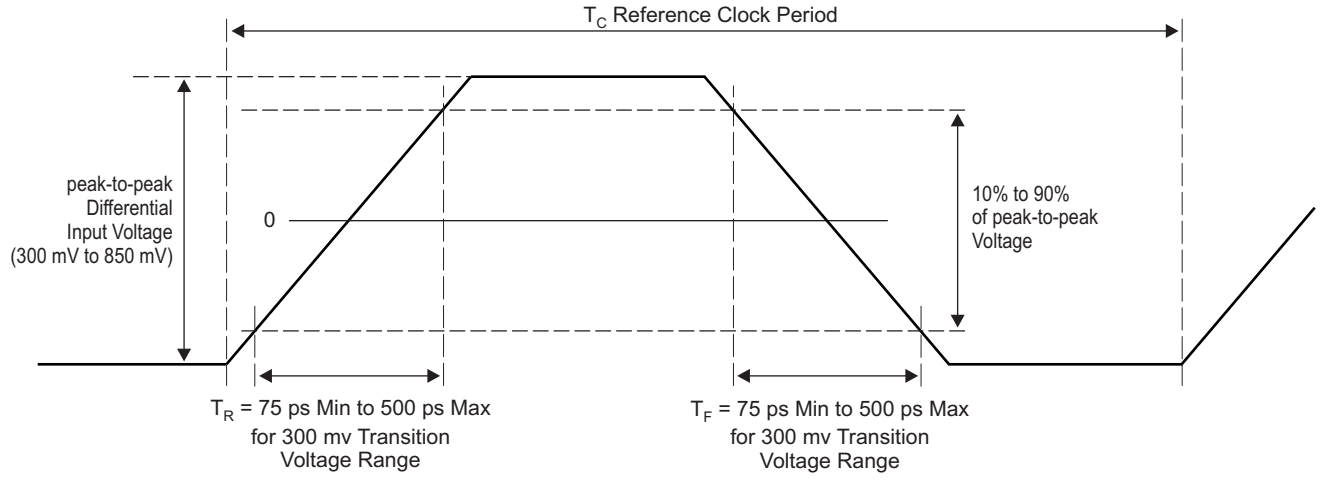


Figure 11-27. USBCLK Rise and Fall Times

11.6 DDR3A PLL

The DDR3A PLL generates interface clocks for the DDR3A memory controller. When coming out of power-on reset, DDR3A PLL is programmed to a valid frequency during the boot configuration process before being enabled and used.

DDR3A PLL power is supplied via the DDR3 PLL power-supply pin (AVDDA6-AVDDA10). 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.

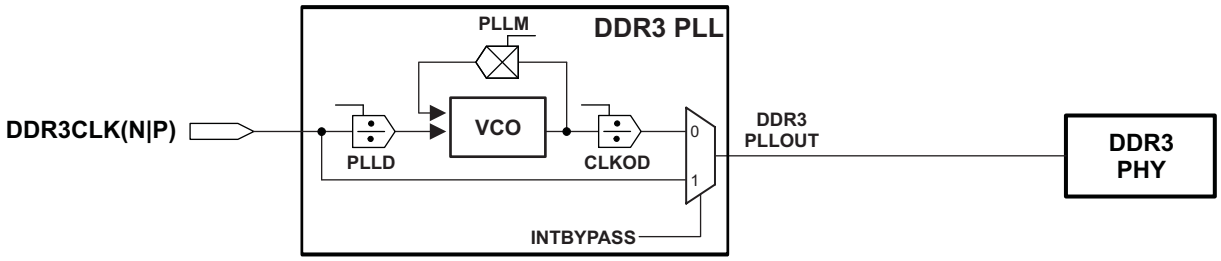


Figure 11-28. DDR3A PLL Block Diagram

11.6.1 DDR3A PLL Control Registers

The DDR3A PLL, which is used to drive the DDR3A PHY for the EMIF, does not use a PLL controller. DDR3A PLL can be controlled using the DDR3APLLCTL0 and DDR3APLLCTL1 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 9.1.3. See Section 9.2.3.5 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers are reset on POR only.

Figure 11-29. DDR3A PLL Control Register 0 (DDR3APLLCTL0)

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 11-31. DDR3A PLL Control Register 0 Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3APLLCTL0 and DDR3APLLCTL1 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 11-30. DDR3A PLL Control Register 1 (DDR3APLLCTL1)

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 11-32. DDR3A 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 inDDR3APLLCTL0 and DDR3APLLCTL1 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.

11.6.2 DDR3A PLL Device-Specific Information

As shown in Figure 11-28, the output of DDR3A PLL (PLLOUT) is divided by 2 and directly fed to the DDR3A memory controller. During power-on resets, the internal clocks of the DDR3 PLL are affected as described in Section Section 11.4. The DDR3 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 of the other resets.

11.6.3 DDR3 PLL Input Clock Electrical Data/Timing

Table 11-33 applies to DDR3A memory interface.

Table 11-33. DDR3 PLL DDRCLK(N|P) Timing Requirements

(see Figure 11-31 and Figure 11-25)

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

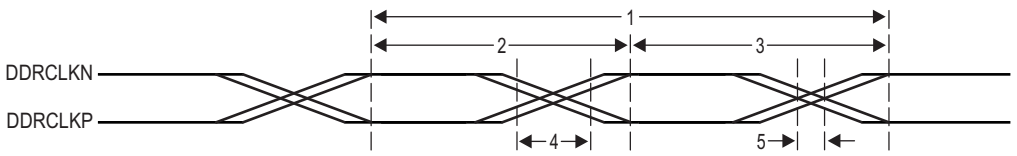


Figure 11-31. DDR3 PLL DDRCLK Timing

11.7 NETCP PLL

The NETCP PLL generates interface clocks for the Network Coprocessor. Like Main PLL and ARM PLL, using the CORECLKSEL[1:0] pins the user can select the input source of the NETCP PLL. 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.

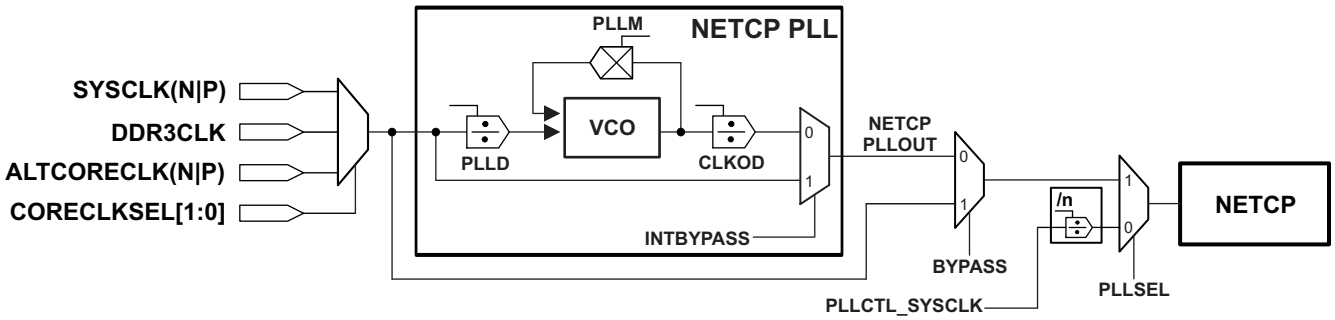


Figure 11-32. NETCP PLL Block Diagram

11.7.1 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 11-34.

11.7.2 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 9.1.3. See Section 9.2.3.5 for the address location of the registers and locking and unlocking sequences for accessing these registers. These registers are reset on POR only.

Figure 11-33. 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 11-34. 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 11-34. 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.

Table 11-35. 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	NETCPPLL	• 0 = Not supported • 1 = NETCPPLL
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.

11.7.3 NETCP PLL Device-Specific Information

As shown in Figure 11-32, 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 11.4. 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.

11.8 DFE PLL

The DFE PLL generates interface clocks for the DFE and IQNet peripherals. When coming out of power-on reset, DFE PLL comes out in a bypass mode and needs to be programmed to a valid frequency before being enabled and used.

DFE PLL power is supplied via the DFE PLL power-supply pins (AVDDA1-AVDDA5). 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.

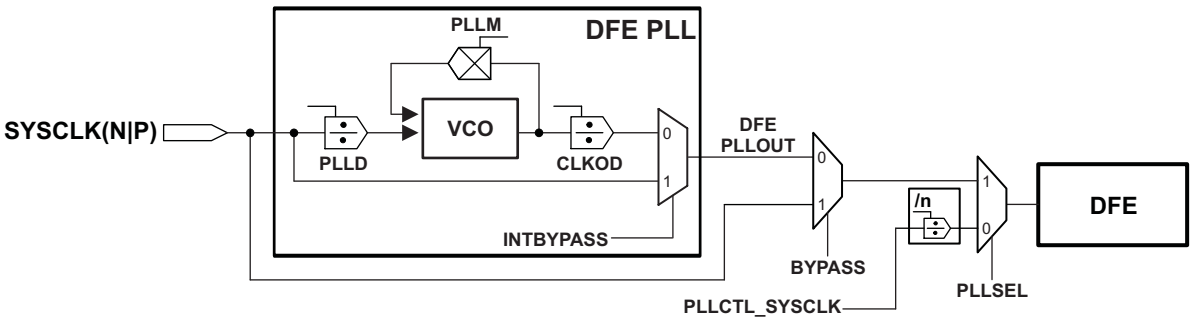


Figure 11-34. DFE PLL Block Diagram

11.8.1 DFE PLL Control Registers

The DFE PLL, which is used to drive the DFE and IQN, does not use a PLL controller. DFE PLL can be controlled using the DFEPLLCTL0 and DFEPLLCTL1 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 9.1.3.1. See Section 9.2.3.5 for the address location of the registers and locking and unlocking sequences for accessing these registers. These registers are reset on $\overline{\text{POR}}$ only.

Figure 11-35. DFE PLL Control Register 0 (DFEPLLCTL0)

31 24	23	22 19	18 6	5 0
BWADJ[7:0]	BYPASS	CLKOD	PLLM	PLLD
RW-0000 1001	RW-1	RW-0001	RW-0000000010011	RW-000000

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

Table 11-36. DFE PLL Control Register 0 Field Descriptions (DFEPLLCTL0)

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in DFEPLLCTL0 and DFEPLLCTL1 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	PLL bypass mode: • 0 = PLL is not in BYPASS mode • 1 = PLL is in BYPASS mode
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 multiplication factor (see note below). 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 11-36. DFE PLL Control Register 1 (DFEPLLCTL1)

31 15	14	13	12 7	6	5 4	3 0
Reserved	PLLRST	DFEPLL	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 11-37. DFE PLL Control Register 1 Field Descriptions (DFEPLLCTL1)

Bit	Field	Description
31-15	Reserved	Reserved
14	PLLRST	PLL Reset bit • 0 = PLL Reset is released • 1 = PLL Reset is asserted
13	DFEPLL	• 0 = Not supported • 1 = DFEPLL
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 DFEPLLCTL0 and DFEPLLCTL1 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.

11.8.2 DFE Clock Divider Control Register (DFE_CLKDIV_CTL)

The DFE_CLKDIV_CTL register is used to program the clock divider that exists at the chip level, it divides down the output of the clock signal from the DFE PLL Controller and is routed to the DFE subsytem core logic.

Figure 11-37. DFE Clock Divider Control Register (DFE_CLKDIV_CTL)

31	2	1	0
Reserved		DIV_MODE	
R-0		RW-00	

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

Table 11-38. DFE Clock Divider Control Register Field Descriptions

Bit	Field	Description
31-2	Reserved	
1-0	DIV_MODE	A 2-bit field that selects the values for the reference divider • 00 = DFE PLL output clock divided by 4 (default) • 01 = DFE PLL output clock divided by 2 • 10 = DFE PLL output clock divided by2 • 11 = Reserved

11.8.3 DFE Clock Sync Control Register (DFE_CLKSYNC_CTL)

The DFE_CLKSYNC_CTL register is used to enable the SYSCLK and SYSREF synchronization logic.

Synchronous Ethernet (SyncE) allows distribution of traceable frequency synchronization to packet nodes the need to communication with TDM network elements. It is also used to distribute timing to applications that rely on precise frequency synchronization such as wireless backhaul.

In 66AK2L06, SyncE is achieved by deriving a TSRXCLKOUTn clock signal based on recovered RX clock from the SGMII SerDes interface. The TSRXCLKOUTn is fed into an DPLL which will supply, along with a clock generator, TSREFCLK, SGMII, SYSCLK, and SYSREF clocks. SyncE may also be achieved by software PLL, via reading registers from CPTS, then drive a clock adjusting signal via SPI (similar to IEEE 1588 clock adjusting method).

Figure 11-38. DFE Clock Sync Control Register (DFE_CLKSYNC_CTL)

31	1	0
Reserved		SYNC_EN
R-0		RW-0

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

Table 11-39. DFE Clock Sync Control Register Field Descriptions

Bit	Field	Description
31-1	Reserved	
0	SYNC_EN	Sync logic enable • 0 = Sync logic not enabled (default) • 1 = Sync logic enabled

11.8.4 DFE Electrical Data/Timing

Table 11-40 provides a cross reference between the JESD204B signal names and the 66AK2L06 name.

Table 11-40. 66AK2L06 to JESD204B Signal Name Cross Reference

66AK2L06	JESD204B
DFESYNCIN 0 and 1	SYNCIN
DFESYNCOUT 0 and 1	SYNCOUT
DFESYSREF	SYSREF
SYSCLK	SYSCLK

Table 11-41. DFEIO (0-17) GPIO Input Pulse Timing Requirements

(see Figure 11-39)

NO.	PARAMETER		MIN	MAX	UNIT
2	tw(DFEGPIL)	Pulse Duration, DFEGPI Low	2P(1)		ns
1	tw(DFEGPIH)	Pulse Duration, DFEGPI High	2P		ns

(1) P = 1/SYSCLK clock frequency in ns.

Table 11-42. DFEIO (0-17) GPIO Output Timing Characteristics

(see Figure 11-39)

NO.	PARAMETER		MIN	MAX	UNIT
4	tw(DFEGPOL)	Pulse Duration, DFEGPO Low	2P(1)		ns
3	tw(DFEGPOH)	Pulse Duration, DFEGPO High	2P		ns

(1) P = 1/SYSCLK clock frequency in ns.

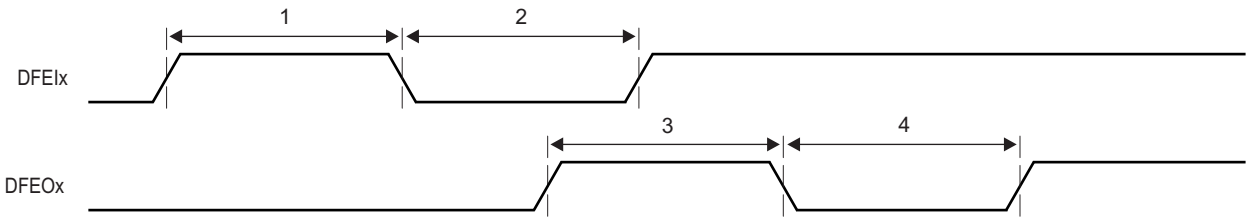


Figure 11-39. DFEIO (0-17) GPIO Input/Output

Table 11-43. DFESYNCIN Sync Input Pulse Timing Requirements

(see Figure 11-40)

NO.	PARAMETER		MIN	MAX	UNIT
2	tw(DFESYNCINN0L)	Pulse Duration, DFESYNCIN(N)0 Low	2P(1)		ns
1	tw(DFESYNCINN0H)	Pulse Duration, DFESYNCIN(N)0 High	2P		ns
2	tw(DFESYNCINP0L)	Pulse Duration, DFESYNCIN(P)0 Low	2P		ns
1	tw(DFESYNCINP0H)	Pulse Duration, DFESYNCIN(P)0 High	2P		ns
2	tw(DFESYNCINN1L)	Pulse Duration, DFESYNCIN(N)1 Low	2P		ns
1	tw(DFESYNCINN1H)	Pulse Duration, DFESYNCIN(N)1 High	2P		ns
2	tw(DFESYNCINP1L)	Pulse Duration, DFESYNCIN(P)1 Low	2P		ns
1	tw(DFESYNCINP1H)	Pulse Duration, DFESYNCIN(P)1 High	2P		ns

(1) P = 1/SYSCLK clock frequency in ns.

Table 11-44. DFESYNCOUTSync Output Pulse Switching Characteristics

(see Figure 11-40)

NO.	PARAMETER		MIN	MAX	UNIT
2	tw(DFESYNCOUTN0L)	Pulse Duration, DFESYNCOUT(N)0 Low	2P(1)		ns
1	tw(DFESYNCOUTN0H)	Pulse Duration, DFESYNCOUT(N)0 High	2P		ns

(1) P = 1/SYSCLK clock frequency in ns.

Table 11-44. DFESYNCOUTSync Output Pulse Switching Characteristics (continued)

(see Figure 11-40)

NO.	PARAMETER		MIN	MAX	UNIT
2	tw(DFESYNCOUTP0L)	Pulse Duration, DFESYNCOUT(P)0 Low	2P		ns
1	tw(DFESYNCOUTP0H)	Pulse Duration, DFESYNCOUT(P)0 High	2P		ns
2	tw(DFESYNCOUTN1L)	Pulse Duration, DFESYNCOUT(N)1 Low	2P		ns
1	tw(DFESYNCOUTN1H)	Pulse Duration, DFESYNCOUT(N)1 High	2P		ns
2	tw(DFESYNCOUTP1L)	Pulse Duration, DFESYNCOUT(P)1 Low	2P		ns
1	tw(DFESYNCOUTP1H)	Pulse Duration, DFESYNCOUT(P)1 High	2P		ns

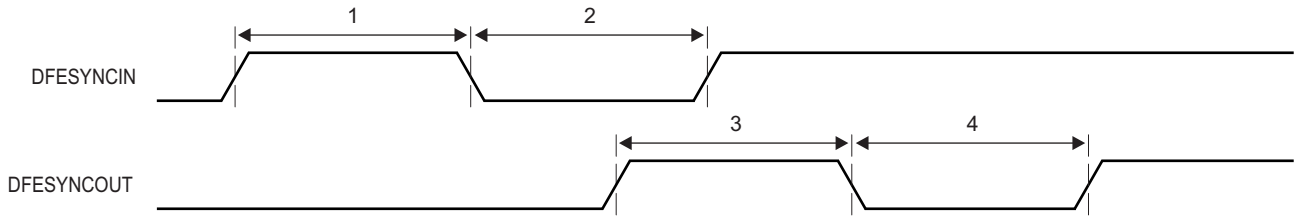


Figure 11-40. DFESYNCIN Sync Input Pulse Timing

Table 11-45. DFESYSREF Input Pulse Timing Requirements

(see Figure 11-41)

NO.	PARAMETER		MIN	MAX	UNIT
9	th(DFESYSREFN-SYSCLKP)	Hold Time - DFESYSREFN valid after SYSCLKP high	1		ns
9	th(DFESYSREFN-SYSCLKN)	Hold Time - DFESYSREFN valid after SYSCLKN low	1		ns
9	th(DFESYSREFP-SYSCLKP)	Hold Time - DFESYSREFP valid after SYSCLKP high	1		ns
9	th(DFESYSREFP-SYSCLKN)	Hold Time - DFESYSREFP valid after SYSCLKN low	1		ns
7	tr(DFESYSREFN)	Rise Time - DFESYSREFN 10% to 90%		350	ns
7	tf(DFESYSREFN)	Fall Time - DFESYSREFN 10% to 90%		350	ns
7	tr(DFESYSREFP)	Rise Time - DFESYSREFP 10% to 90%		350	ns
7	tf(DFESYSREFP)	Fall Time - DFESYSREFP 10% to 90%		350	ns

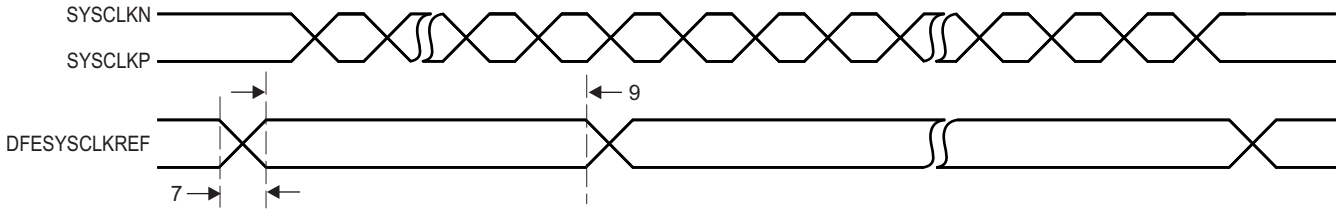


Figure 11-41. DFESYSREF Input Pulse Timing

11.9 External Interrupts

11.9.1 External Interrupts Electrical Data/Timing

Table 11-46. $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$ Timing Requirements(1)

(see Figure 11-42)

NO.			MIN	MAX	UNIT
1	tsu($\overline{\text{LRESET}}$ - $\overline{\text{LRESETNMIEN}}$)	Setup time - $\overline{\text{LRESET}}$ valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns
1	tsu($\overline{\text{NMI}}$ - $\overline{\text{LRESETNMIEN}}$)	Setup time - $\overline{\text{NMI}}$ valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns
1	tsu(CORESELn- $\overline{\text{LRESETNMIEN}}$)	Setup time - CORESEL[2:0] valid before $\overline{\text{LRESETNMIEN}}$ low	12*P		ns

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

Table 11-46. $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$ Timing Requirements[(1)] (continued)

(see Figure 11-42)

NO.			MIN	MAX	UNIT
2	th($\overline{\text{LRESETNMIEN}}$ - $\overline{\text{LRESET}}$)	Hold time - $\overline{\text{LRESET}}$ valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
2	th($\overline{\text{LRESETNMIEN}}$ - $\overline{\text{NMI}}$)	Hold time - $\overline{\text{NMI}}$ valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
2	th($\overline{\text{LRESETNMIEN}}$ -CORESELn)	Hold time - CORESEL[2:0] valid after $\overline{\text{LRESETNMIEN}}$ high	12*P		ns
3	tw($\overline{\text{LRESETNMIEN}}$)	Pulsewidth - $\overline{\text{LRESETNMIEN}}$ low width	12*P		ns

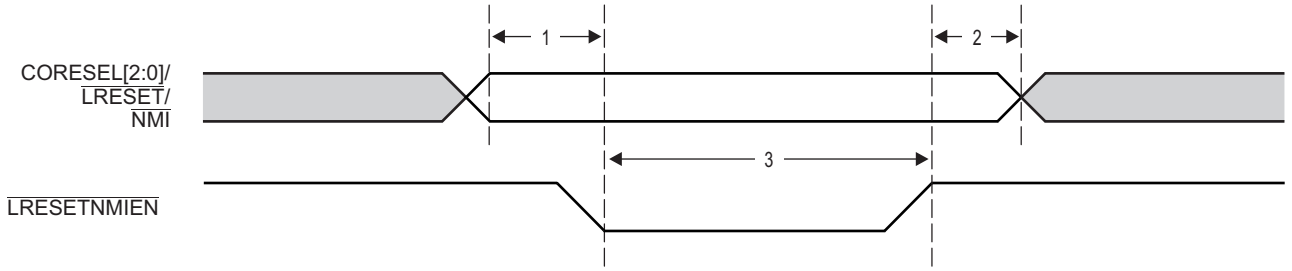


Figure 11-42. $\overline{\text{NMI}}$ and $\overline{\text{LRESET}}$ Timing

11.10 On-Chip Standalone RAM (OSR)

The 1MB OSR is added to the device for:

- QM External Linking RAM
- NetCP1.5 intermediate data buffer
- Intermediate buffering of other data storage

The OSR supported features include:

- SRAM supports ECC with Read-Modify-Write logic
- RTA memory
- Support interrupt for ECC error event
- Support Little and Big-endian modes of operation

OSR does not support any type of cache access, hence this memory space must always be marked as non-cacheable region for both DSP and ARM cores.

11.11 DDR3A Memory Controller

The 72-bit DDR3 Memory Controller bus of the 66AK2L06 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.

11.11.1 DDR3 Memory Controller Device-Specific Information

The 66AK2L06 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.

11.11.2 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.

11.11.3 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.

11.12 I^2C 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.

11.12.1 I^2C 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 66AK2L06 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 11-43 shows a block diagram of the I^2C module.

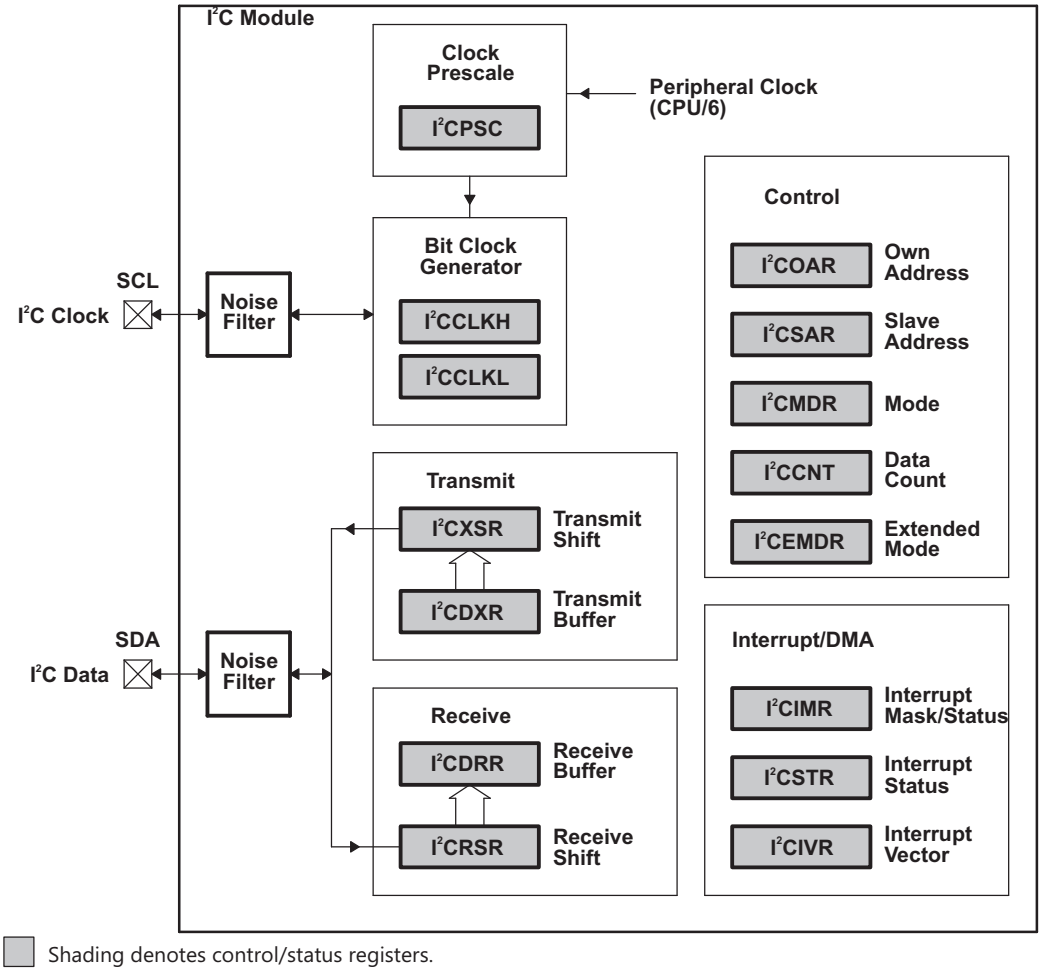


Figure 11-43. I²C Module Block Diagram

11.12.2 I²C Peripheral Register Description

Table 11-47. 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 11-47. I²C Registers (continued)

HEX ADDRESS OFFSETS	ACRONYM	REGISTER NAME
0x003C -0x007F	-	Reserved

11.12.3 I²C Electrical Data/Timing

11.12.3.1 Inter-Integrated Circuits (I²C) Timing

Table 11-48. I²C Timing Requirements (1)

(see Figure 11-44)

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.

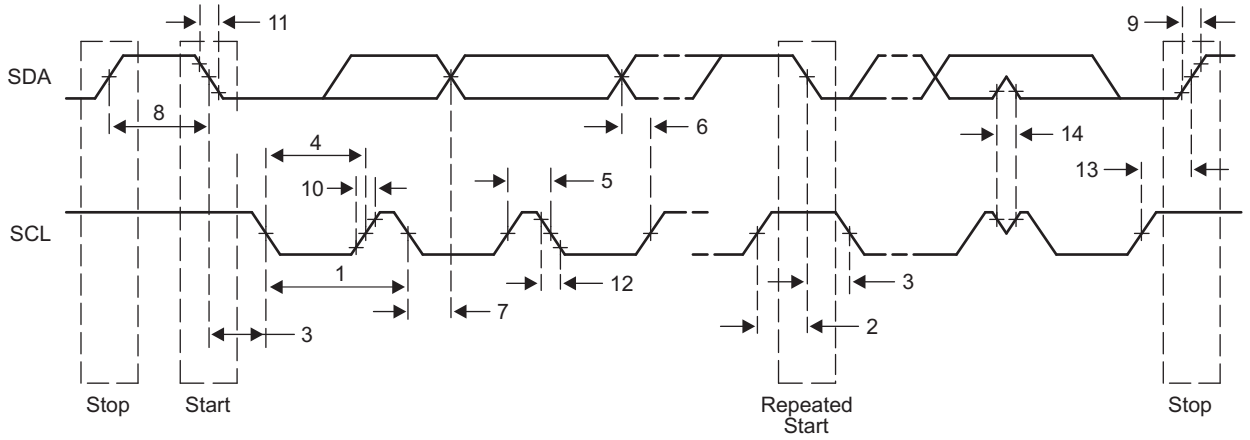


Figure 11-44. I²C Receive Timings

Table 11-49. I²C Switching Characteristics(1)

(see Figure 11-45)

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.

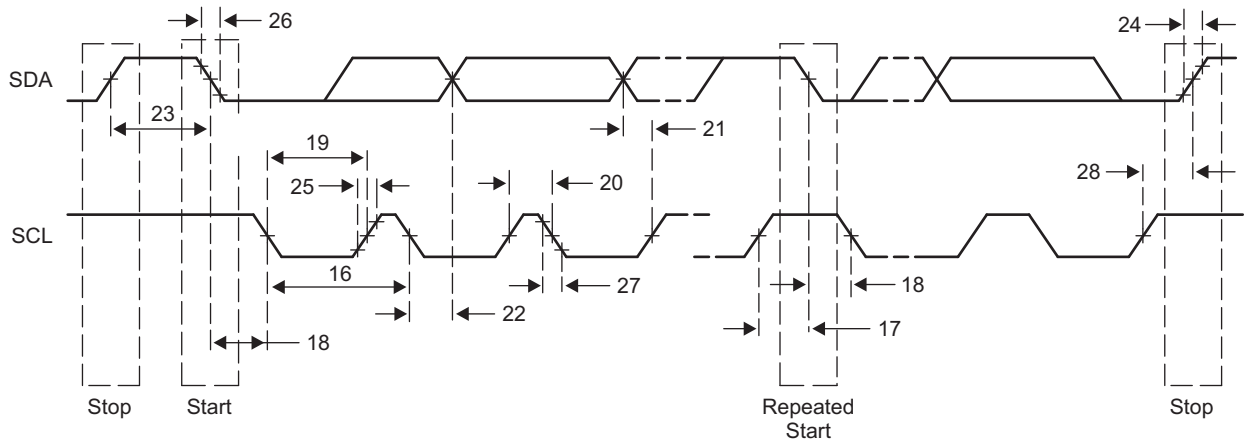


Figure 11-45. I²C Transmit Timings

11.13 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 66AK2L06 is supported only in master mode. Additional chip-level components can also be included, such as temperature sensors or an I/O expander.

11.13.1 SPI Electrical Data/Timing

Table 11-50. SPI Timing Requirements

(see Figure 11-46)

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 11-51. SPI Switching Characteristics

(see Figure 11-46 and Figure 11-47)

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 11-51. SPI Switching Characteristics (continued)

(see Figure 11-46 and Figure 11-47)

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

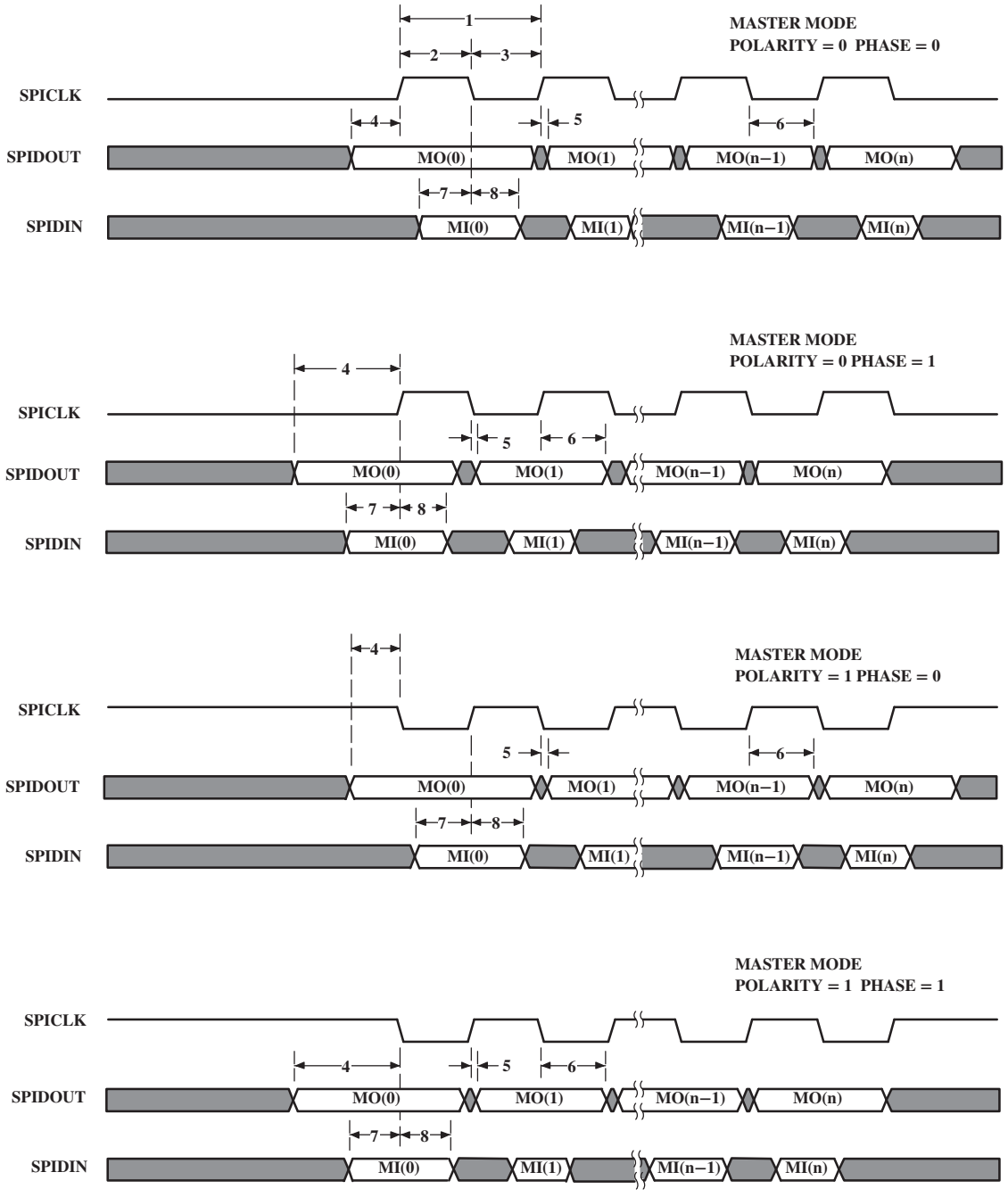


Figure 11-46. SPI Master Mode Timing Diagrams — Base Timings for 3-Pin Mode

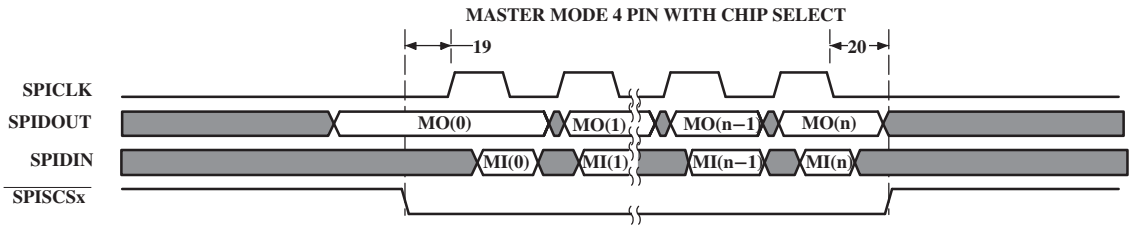


Figure 11-47. SPI Additional Timings for 4-Pin Master Mode with Chip Select Option

11.14 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 CorePacs to be sent to the peripheral device. The SoC CorePacs 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 11-52. UART Timing Requirements

(see Figure 11-48 and Figure 11-49)

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)

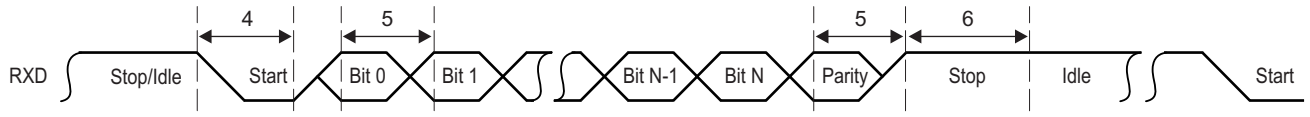


Figure 11-48. UART Receive Timing Waveform

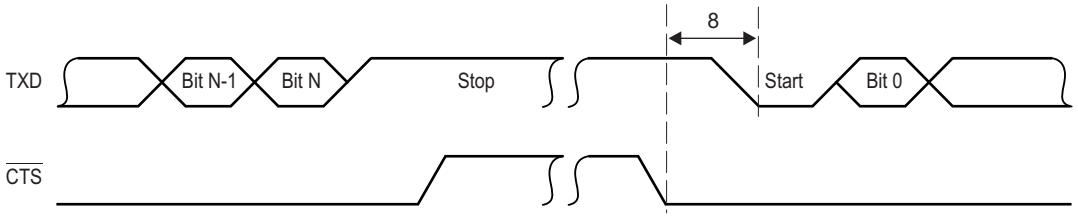


Figure 11-49. UART CTS (Clear-to-Send Input) — Autoflow Timing Waveform

Table 11-53. UART Switching Characteristics

(see Figure 11-50 and Figure 11-51)

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)

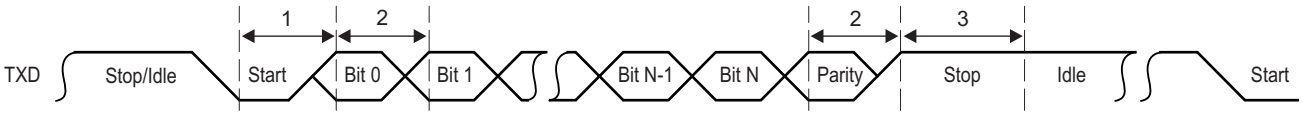


Figure 11-50. UART Transmit Timing Waveform

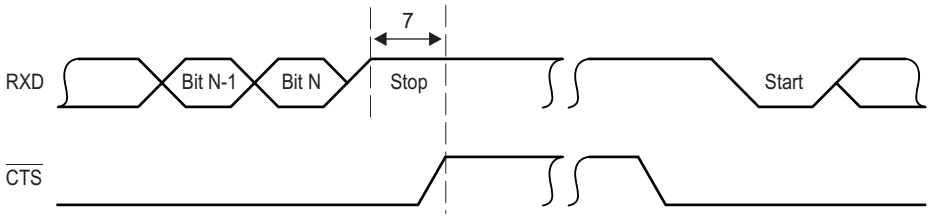


Figure 11-51. UART RTS (Request-to-Send Output) – Autoflow Timing Waveform

11.15 PCIe Peripheral

The two-lane PCI express (PCIe) module on 66AK2L06 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).

11.16 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).

11.17 Security Accelerator

The Security Accelerator (SA) provides wire-speed processing on 1 Gbps Ethernet traffic on IPSec and SRTP security protocols. It functions on the packet level with the packet and the associated security context being one of the above two 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).

11.18 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 66AK2L06. 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 11-52. 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 11-54. MACID1 Register Field Descriptions

Bit	Field	Description
31-0	MAC ID	MAC ID. Lower 32 bits.

Figure 11-53. 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 11-55. 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 11-54.

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 11-56 for interconnectivity. Regarding the SyncE signal, see Section 9.2.3.32 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 11-56. 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 11-54. 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 11-57. 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

11.19 SGMII 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) switch subsystem for correct operation. The module allows almost transparent operation of the MDIO interface, with very little attention from the C66x CorePac. For more information, see the *Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide* (SPRUGV9).

Table 11-58. MDIO Timing Requirements

(see Figure 11-55)

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

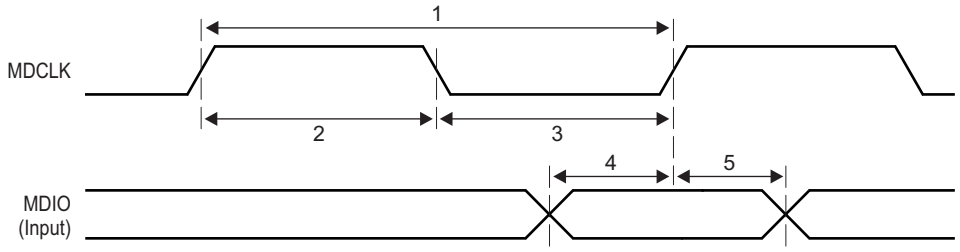


Figure 11-55. MDIO Input Timing

Table 11-59. MDIO Switching Characteristics

(see Figure 11-56)

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

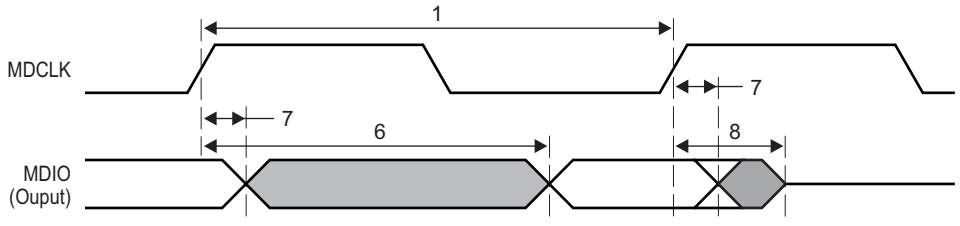


Figure 11-56. MDIO Output Timing

11.20 Timers

The timers can be used to time events, count events, generate pulses, interrupt the CorePacs, and send synchronization events to the EDMA3 channel controller.

11.20.1 Timers Device-Specific Information

The 66AK2L06 device has up to twenty 64-bit timers in total, of which Timer0 through Timer3 are dedicated to each of the four C66x CorePacs as watchdog timers and can also be used as general-purpose timers. Timer16 and Timer17 are dedicated to each of the Cortex-A15 processor cores as a watchdog timer and can also be used as general-purpose timers. The remaining timers 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 11.5.3.6) and the type of reset initiated can set by programming the Reset Configuration Register (RSTCFG) (see Section 11.5.3.8). For more information, see the *KeyStone Architecture Timer 64P User's Guide* SPRUGV5.

11.20.2 Timers Electrical Timing

The tables and figures below describe the timing requirements and switching characteristics of the timers.

Table 11-60. Timer Input Timing Requirements (1)

(see Figure 11-57)

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 11-61. Timer Output Switching Characteristics (1)

(see Figure 11-57)

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.

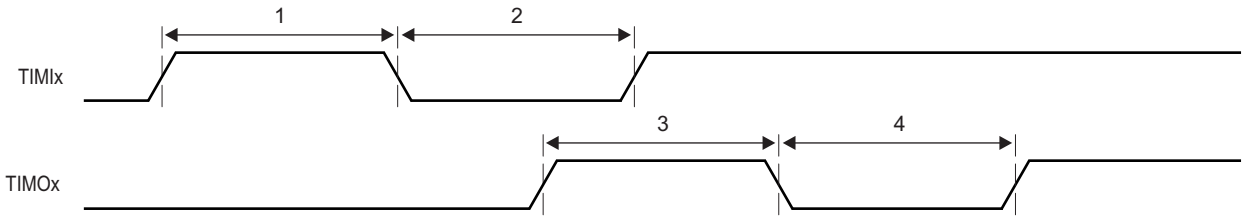


Figure 11-57. Timer Timing

11.21 Rake Search Accelerator (RSA)

There are eight Rake Search Accelerators (RSAs) on the device. Each C66x CorePac has one set of directly-connected RSA pairs. The RSA is an extension of the C66x CorePac. The C66x CorePac performs send/receive to the RSAs via the .L and .S functional units.

11.22 General-Purpose Input/Output (GPIO)

11.22.1 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 66AK2L06 device pin muxing, see Section 9.2.

These GPIO pins can also be used to generate individual core interrupts (no support of bank interrupt) and EDMA events.

11.22.2 GPIO Peripheral Register Description

Table 11-62. 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
0x0090 - 0x03FF	-	Reserved

11.22.3 GPIO Electrical Data/Timing

Table 11-63. GPIO Input Timing Requirements(1)

(see Figure 11-58)

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 11-64. GPIO Output Switching Characteristics(1)

(see Figure 11-58)

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

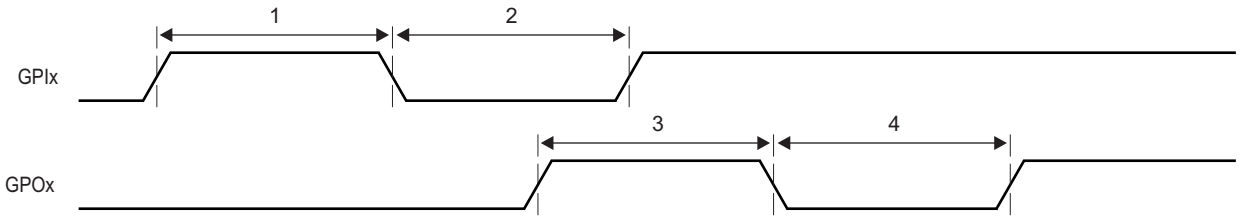


Figure 11-58. GPIO Timing

11.23 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.

11.24 IQNet (IQN)

The 66AK2L06 has the new IQNet IP. The IQN subsystem interfaces external I/O into TI DMA systems.The IQN subsystem consists of the AID module (interface for DFE), AIF timer (AT) module, one PKTDMA interface, DIO engine and IQ streaming switch (IQS).

- Transport of data streams to an integrated Digital Front-End (DFE) module via AID block
- AIL module is not supported
- Integrated AIF2 Timer (AT)
  - 24 System Events, 1 BCN counter, 8 complex RADT (radio timers)
  - Supports various timing sync sources - RP1, generic input pins, CPTS or software

For more information, see the *KeyStone II Architecture IQN2 User's Guide* (SPRUH06).

Table 11-65. AIF Timer Module Timing Requirements

(see Figure 11-59, Figure 11-60, Figure 11-61, and Figure 11-62)

NO.			MIN	MAX	UNIT
RP1 Clock and Frameburst					
1	tc(RP1CLKN)	Cycle time, RP1CLK(N)	32.55	32.55	ns
1	tc(RP1CLKP)	Cycle time, RP1CLK(P)	32.55	32.55	ns
2	tw(RP1CLKNL)	Pulse duration, RP1CLK(N) low	0.4 * C1 (1)	0.6 * C1	ns
3	tw(RP1CLKNH)	Pulse duration, RP1CLK(N) high	0.4 * C1	0.6 * C1	ns
3	tw(RP1CLKPL)	Pulse duration, RP1CLK(P) low	0.4 * C1	0.6 * C1	ns
2	tw(RP1CLKPH)	Pulse duration, RP1CLK(P) high	0.4 * C1	0.6 * C1	ns
4	tr(RP1CLKN)	Rise time - RP1CLKN 10% to 90%		350.00	ps
4	tf(RP1CLKN)	Fall time - RP1CLKN 90% to 10%		350.00	ps

(1) C1 = tc(RP1CLKN/P)

Table 11-65. AIF Timer Module Timing Requirements (continued)

(see Figure 11-59, Figure 11-60, Figure 11-61, and Figure 11-62)

NO.			MIN	MAX	UNIT
4	tr(RP1CLKP)	Rise time - RP1CLKP 10% to 90%		350.00	ps
4	tf(RP1CLKP)	Fall time - RP1CLKP 90% to 10%		350.00	ps
5	tj(RP1CLKN)	Period jitter (peak-to-peak), RP1CLK(N)		600	ps
5	tj(RP1CLKP)	Period jitter (peak-to-peak), RP1CLK(P)		600	ps
6	tw(RP1FBN)	Bit period, RP1FB(N)	8 * C1	8 * C1	ns
6	tw(RP1FBP)	Bit period, RP1FB(P)	8 * C1	8 * C1	ns
7	tr(RP1CLKN)	Rise time - RP1FBN 10% to 90%		350.00	ps
7	tf(RP1CLKN)	Fall time - RP1FBN 90% to 10%		350.00	ps
7	tr(RP1CLKP)	Rise time - RP1FBP 10% to 90%		350.00	ps
7	tf(RP1CLKP)	Fall time - RP1FBP 90% to 10%		350.00	ps
8	tsu(RP1FBN-RP1CLKP)	Setup time - RP1FBN valid before RP1CLKP high	2		ns
8	tsu(RP1FBN-RP1CLKN)	Setup time - RP1FBN valid before RP1CLKN low	2		ns
8	tsu(RP1FBN-RP1CLKP)	Setup time - RP1FBP valid before RP1CLKP high	2		ns
8	tsu(RP1FBN-RP1CLKN)	Setup time - RP1FBP valid before RP1CLKN low	2		ns
9	th(RP1FBN-RP1CLKP)	Hold time - RP1FBN valid after RP1CLKP high	2		ns
9	th(RP1FBN-RP1CLKN)	Hold time - RP1FBN valid after RP1CLKN low	2		ns
9	th(RP1FBN-RP1CLKP)	Hold time - RP1FBP valid after RP1CLKP high	2		ns
9	th(RP1FBN-RP1CLKN)	Hold time - RP1FBP valid after RP1CLKN low	2		ns
PHY Sync and Radio Sync Pulses					
10	tw(PHYSYNCH)	Pulse duration, PHYSYNC high	6.50		ns
11	tc(PHYSYNC)	Cycle time, PHYSYNC pulse to PHYSYNC pulse	10.00		ms
12	tw(RADSYNCH)	Pulse duration, RADSYNC high	6.50		ns
13	tc(RADSYNC)	Cycle time, RADSYNC pulse to RADSYNC pulse	1.00		ms

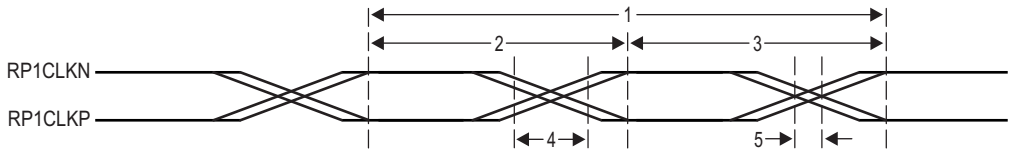


Figure 11-59. AIF RP1 Frame Synchronization Clock Timing

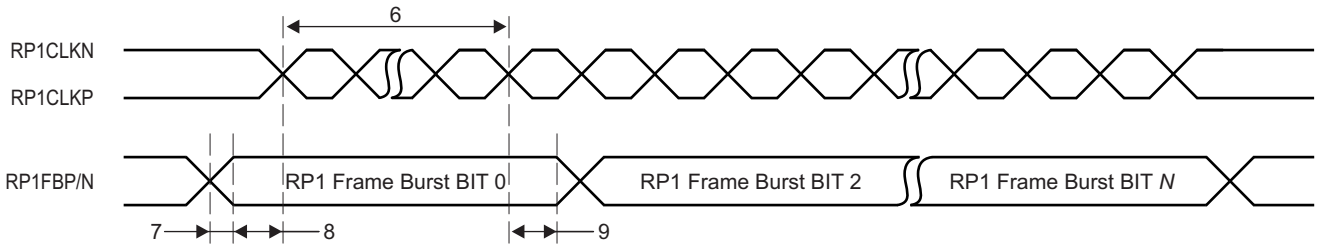


Figure 11-60. AIF RP1 Frame Synchronization Burst Timing

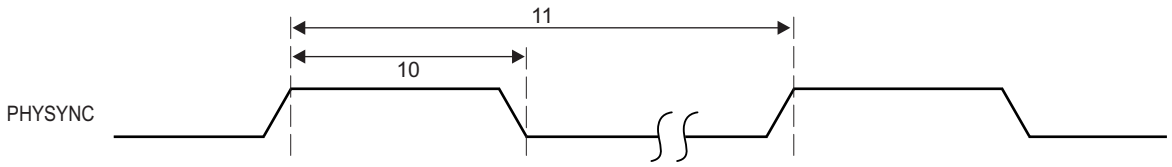


Figure 11-61. AIF Physical Layer Synchronization Pulse Timing

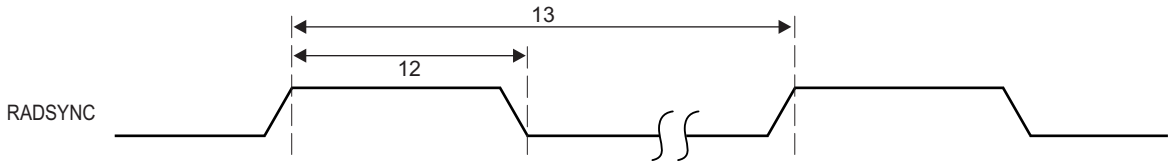


Figure 11-62. AIF Radio Synchronization Pulse Timing

Table 11-66. AIF Timer Module Switching Characteristics

(see Figure 11-63)

NO.	PARAMETER		MIN	MAX	UNIT
External Frame Event					
14	tw(EXTFRAMEEVENTH)	Pulse width, EXTFRAMEEVENT output high	8 * C1 (1)		ns
15	tw(EXTFRAMEEVENTL)	Pulse width, EXTFRAMEEVENT output low	8 * C1		ns

(1) C1 = 245.76MHz clock for CPRI and 307.2MHz clock for OBSAI.

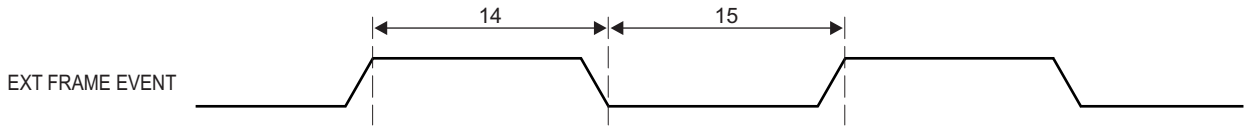


Figure 11-63. AIF Timer External Frame Event Timing

11.25 Digital Front End (DFE)

The 66AK2L06 integrates the Digital Front-End (DFE) subsystem with Digital Down/Up-Conversion (DDC/DUC) functionality. The DFE subsystem provides a direct interface to high-speed analog-to-digital and digital-to-analog data converters and analog front end.

- Support for JESD204A/B
  - Support for Four Tx and Four Rx 7.37Gbps lanes
  - Alignment across multiple lanes within a single converter or multiple converters in the same device
  - Support for Subclass 0 and 1 (Subclass 2 is not supported)
  - Deterministic latency using SYSREF signaling
  - Backward compatibility with JESD204A
- DFE clock frequency of 245.76 MHz and 368.64 MHz
- 8-bit DVGA interface (pin muxed with DFE GPIO)
- 16 DFE GPIOs to interface to Digital Variable Gain Amplifiers (DVGA), RF muxes, Power amplifier (PA) Time Division Duplex (TDD) controls. These GPIOs are different from the chip-level GPIOs. The DFE GPIOs are controlled directly by the DFE MMRs.
- DUC/DDC, RX integrated

The following features are specified in the DFE user's guide, but are not supported in this device:

- Back-end Automatic Gain Control (AGC) Support

- Digital Pre-Distortion (DPD) Support
- Crest Factor Reduction (CFR) Support

11.26 Fast Fourier Transform Coprocessor (FFTC)

There are two Fast Fourier Transform Coprocessors (FFTC) used to accelerate FFT, IFFT, DFT, and IDFT operations. For more information, see the *KeyStone Architecture Fast Fourier Transform Coprocessor (FFTC) User's Guide* (SPRUGS2).

11.27 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).

11.28 Universal Subscriber Identity Module (USIM)

The 66AK2L06 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.

11.29 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).

11.29.1 EMIF16 Electrical Data/Timing

Table 11-67. EMIF16 Asynchronous Memory Timing Requirements(1)

(see Figure 11-64 through Figure 11-67)

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

(1) E = 1/(SYSCLK1/6)

Table 11-67. EMIF16 Asynchronous Memory Timing Requirements[1] (continued)

(see Figure 11-64 through Figure 11-67)

NO.			MIN	MAX	UNIT
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
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+TA+4)*E-3	(WS+WST+WH+TA+4)*E+3	ns
15	t_{c}(CEL)	EMIF write cycle time when ew =1., meaning extended wait mode is enabled	(WS+WST+WH+TA+4)*E-3	(WS+WST+WH+TA+4)*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

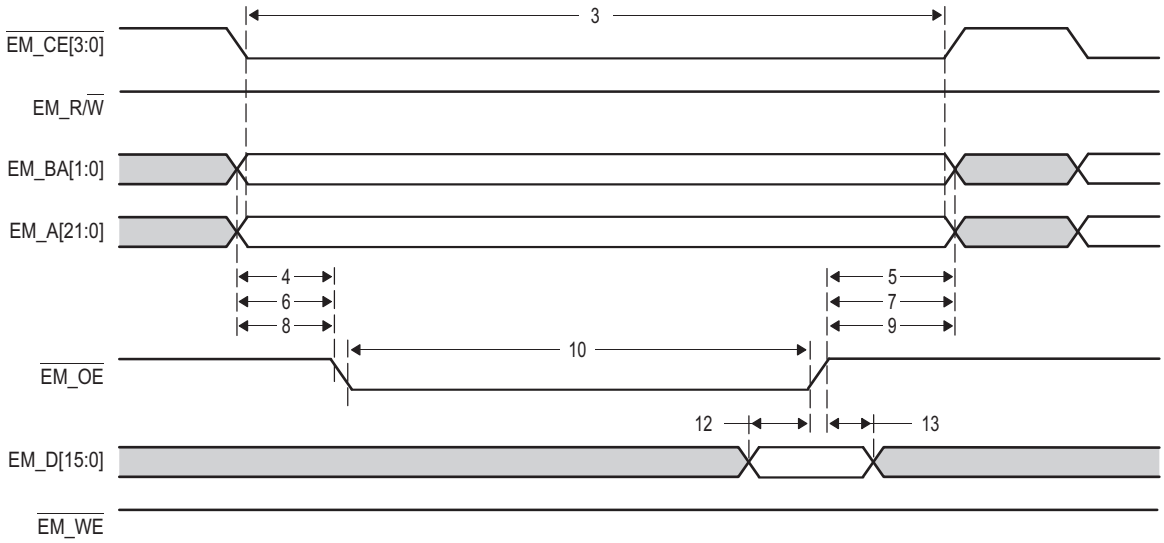


Figure 11-64. EMIF16 Asynchronous Memory Read Timing Diagram

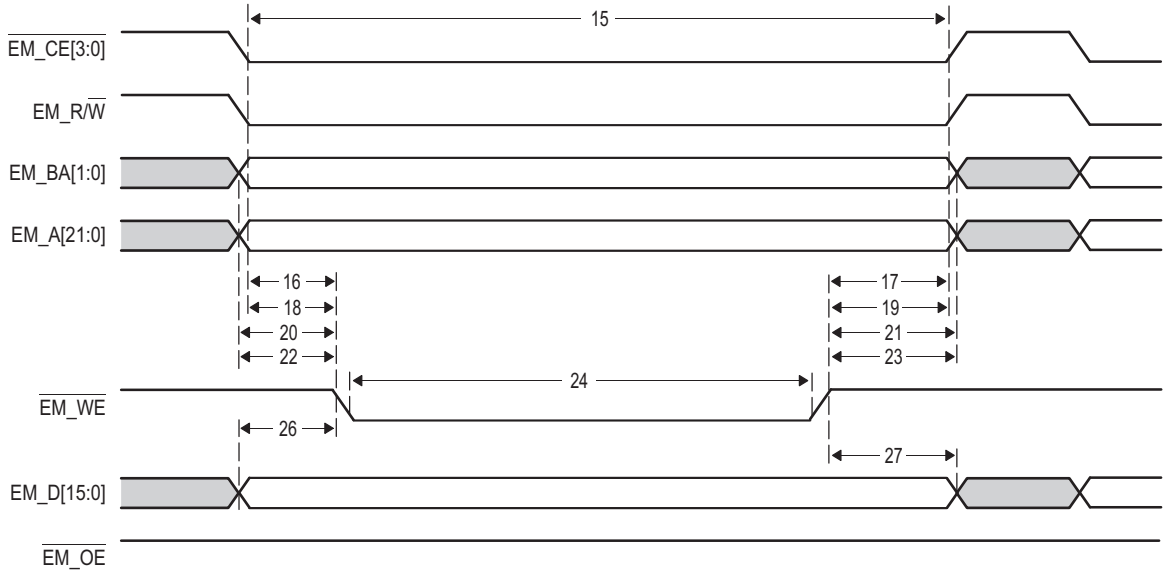


Figure 11-65. EMIF16 Asynchronous Memory Write Timing Diagram

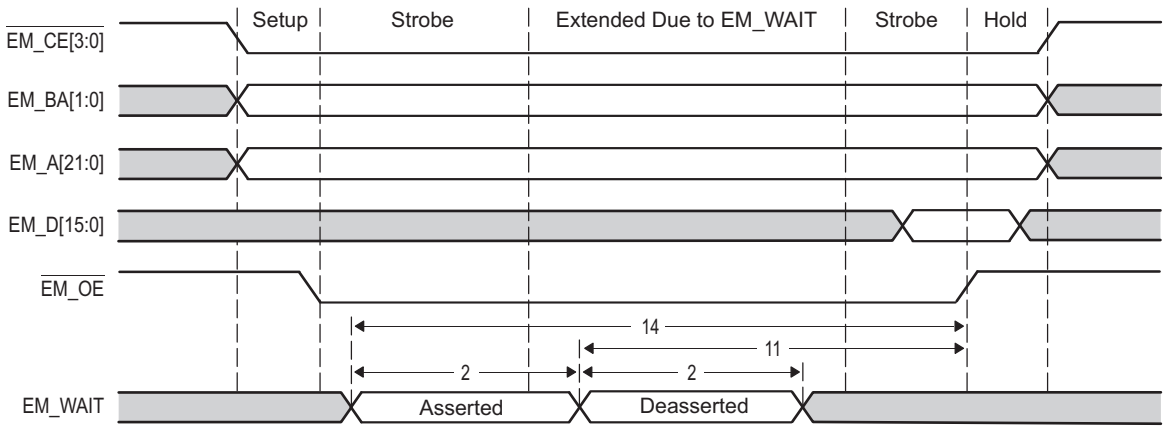


Figure 11-66. EMIF16 EM_WAIT Read Timing Diagram

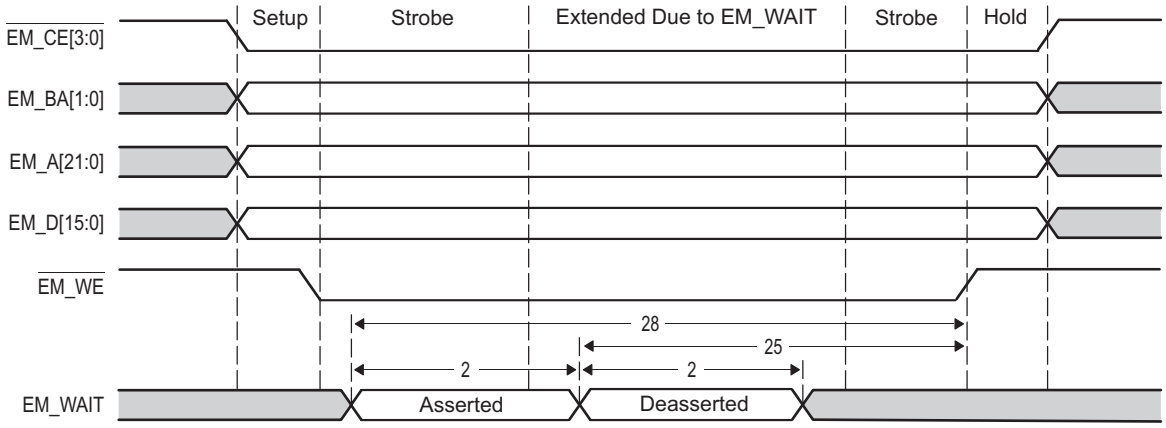


Figure 11-67. EMIF16 EM_WAIT Write Timing Diagram

11.30 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 (ARM and DSP subsystems) to provide a comprehensive hardware platform for a rich debug and development experience.

11.30.1 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 DSP and STM traces or ARM and STM traces via EMU pins is possible. Concurrent trace export of DSP and ARM is not possible via EMU pins.
  - 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/DSP 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 start of all DSP cores
  - Global stopping of all ARM and DSP 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.

11.30.1.1 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

11.30.1.2 DSP Features

- Support for Halt-mode debug
- Support for Real-time debug
- Support for Monitor mode debug
- Advanced Event Triggering (AET) for data/PC watch-points, event monitoring and visibility into external events
- Support for PC/Timing/Data/Event trace.
- TETB (TI Embedded Trace Buffer) of 4KB to store PC/Timing/Data/Event trace. The trace data is copied by EDMA to external memory for draining by device high speed serial interfaces or it can be drained through EMUx pins
- Support for Cross triggering source/sink to other C66x CorePacs and device subsystems.
- Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs application report
- Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems application report

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)

11.30.2 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.

The device has multiple processors, some with secondary JTAG TAPs (C66x CorePacs) and others with an APB memory mapped interface (ARM CorePac and Coresight components).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.

11.30.2.1 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.

There are two types of components connected through ICEPick to the external debug interface:

- Legacy JTAG Components — C66x implements a JTAG-compatible port and are directly interfaced with ICEPick and individually attached to an ICEPick secondary TAP.
- CoreSight Components — 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 11-68 shows the ICEPick secondary taps in the system. For more details on the test related P1500 TAPs, see the DFTSS specification.

Table 11-68. 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	C66x CorePac0	38	No	C66x CorePac0
2	JTAG	C66x CorePac1	38	No	C66x CorePac1
3	JTAG	C66x CorePac2	38	No	C66x CorePac2
4	JTAG	C66x CorePac3	38	No	C66x CorePac3
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).

11.31 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 DSP/STM trace, cross triggering, and debug boot modes as shown in Table 11-72. 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.

Table 11-69. Emulation Interface with Different Debug Port Configurations

EMU PINS	CROSS TRIGGERING	ARM TRACE		DSP TRACE		STM	DEBUG BOOT MODE
EMU33		TRCDT**a**[29]	TRCDT**b**[31]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU32		TRCDT**a**[28]	TRCDT**b**[30]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU31		TRCDT **a**[27]	TRCDT **b**[29]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU30		TRCDT**a**[26]	TRCDT**b**[28]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU29		TRCDT **a**[25]	TRCDT **b**[27]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU28		TRCDT**a**[24]	TRCDT**b**[26]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU27		TRCDT **a**[23]	TRCDT **b**[25]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU26		TRCDT**a**[22]	TRCDT**b**[24]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU25		TRCDT**a**[21]	TRCDT**b**[23]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU24		TRCDT**a**[20]	TRCDT**b**[22]			TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU23		TRCDT**a**[19]	TRCDT**b**[21]	TRCDT**a**[19]		TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU22		TRCDT**a**[18]	TRCDT**b**[20]	TRCDT**a**[18]		TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU21		TRCDT**a**[17]	TRCDT**b**[19]	TRCDT**a**[17]	TRCDT**b**[19]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU20		TRCDT**a**[16]	TRCDT**b**[18]	TRCDT**a**[16]	TRCDT**b**[18]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU19		TRCDT**a**[15]	TRCDT**b**[17]	TRCDT**a**[15]	TRCDT**b**[17]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU18		TRCDT**a**[14]	TRCDT**b**[16]	TRCDT**a**[14]	TRCDT**b**[16]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU17		TRCDT**a**[13]	TRCDT**b**[15]	TRCDT**a**[13]	TRCDT**b**[15]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU16		TRCDT**a**[12]	TRCDT**b**[14]	TRCDT**a**[12]	TRCDT**b**[14]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU15		TRCDT**a**[11]	TRCDT**b**[13]	TRCDT**a**[11]	TRCDT**b**[13]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	

Table 11-69. Emulation Interface with Different Debug Port Configurations (continued)

EMU PINS	CROSS TRIGGERING	ARM TRACE		DSP TRACE		STM	DEBUG BOOT MODE
EMU14		TRCDT**a**[10]	TRCDT**b**[12]	TRCDT**a**[10]	TRCDT**b**[12]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU13		TRCDT**a**[9]	TRCDT**b**[11]	TRCDT**a**[9]	TRCDT**b**[11]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU12		TRCDT**a**[8]	TRCDT**b**[10]	TRCDT**a**[8]	TRCDT**b**[10]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU11		TRCDT**a**[7]	TRCDT**b**[9]	TRCDT**a**[7]	TRCDT**b**[9]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU10		TRCDT**a**[6]	TRCDT**b**[8]	TRCDT**a**[6]	TRCDT**b**[8]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU9		TRCDT**a**[5]	TRCDT**b**[7]	TRCDT**a**[5]	TRCDT**b**[7]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU8		TRCDT**a**[4]	TRCDT**b**[6]	TRCDT**a**[4]	TRCDT**b**[6]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU7		TRCDT**a**[3]	TRCDT**b**[5]	TRCDT**a**[3]	TRCDT**b**[5]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU6		TRCDT**a**[2]	TRCDT**b**[4]	TRCDT**a**[2]	TRCDT**b**[4]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU5		TRCDT**a**[1]	TRCDT**b**[3]	TRCDT**a**[1]	TRCDT**b**[3]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU4		TRCDT**a**[0]	TRCDT**b**[2]	TRCDT**a**[0]	TRCDT**b**[2]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU3		TRCCTRL	TRCCTRL	TRCCLKB	TRCCLKB	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU2		TRCCLK	TRCCLK	TRCCLKA	TRCCLKA	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	
EMU1	Trigger1		TRCDT**b**[1]		TRCDT**b**[1]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	dbgbootmode[1]
EMU0	Trigger0		TRCDT**b**[0]		TRCDT**b**[0]	TRCDT3, or TRCDT2, or TRCDT1, or TRCDT0, or TRCCLK, or Tri-state	dbgbootmode[0]

11.31.1 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) and C66x Trace (up to 20 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)

ARM and DSP simultaneous trace is not supported.

11.31.2 Master ID for HW and SW Messages

Table 11-70 describes the master ID for the various hardware and software masters of the STM.

Table 11-70. MSTID Mapping for Hardware Instrumentation (CPTRACERS)

MSTID [7:0]	CPTRACER NAME	CLOCK DOMAIN	SID[4:0]	DESCRIPTION
0x94-0x97	CPT_MSMCx_MST, where x = 0..3	SYSCLK1/1	0x0..3	MSMC SRAM Bank 0 to MSMC SRAM Bank 3 monitors
0xB1	CPT_MSMC4_MST	SYSCLK1/1	0x4	MSMC SRAM Bank 4
0xAE - 0xB0	CPT_MSMCx_MST, where x = 5..7	SYSCLK1/1	0x5..7	MSMC SRAM Bank 5to MSMC SRAM Bank 7 monitors
0x98	CPT_DDR3A_MST	SYSCLK1/1	0x8	MSMC DDR3A port monitor
0x8C - 0x93	CPT_L2_x_MST, where x = 0..3	SYSCLK1/3	0x9..0x10	DSP 0 to 3 SDMA port monitors
0xA4	CPT_TPCC0_4_MST	SYSCLK1/3	0x11	EDMA 0 CFG port monitor
0xA5	CPT_TPCC1_2_3_MST	SYSCLK1/3	0x12	EDMA 1and EDMA2 CFG port monitor
0xA6	CPT_INTC_MST	SYSCLK1/3	0x13	INTC port monitor (for INTC 0/1 and GIC400)
0x99	CPT_SM_MST	SYSCLK1/3	0x14	Semaphore CFG port monitors
0x9A	CPT_QM_CFG1_MST	SYSCLK1/3	0x15	QMSS CFG1 port monitor
0xA0	CPT_QM_CFG2_MST	SYSCLK1/3	0x16	QMSS CFG2 port monitor
0x9B	CPT_QM_M_MST	SYSCLK1/3	0x17	QM_M CFG/DMA port monitor
0xA7	CPT_SPI_ROM_EMIF16_MST	SYSCLK1/3	0x18	SPI ROM EMIF16 CFG port monitor
0x9C	CPT_CFG_MST	SYSCLK1/3	0x19	SCR_3P_B and SCR_6P_B CFG peripheral port monitors
0x9D	Reserved			
0x9E	Reserved			
0x9F	Reserved			

Table 11-71. MSTID Mapping for Software Messages

MSTID [7:0]	CORE NAME	DESCRIPTION
0x0	C66x CorePac0	C66x CorePac MDMA Master ID
0x1	C66x CorePac1	C66x CorePac MDMA Master ID
0x2	C66x CorePac2	C66x CorePac MDMA Master ID
0x3	C66x CorePac3	C66x CorePac MDMA Master ID
0x4	Reserved	
0x5	Reserved	
0x6	Reserved	
0x7	Reserved	
0x8	A15 Core0	ARM Master IDs

Table 11-71. MSTID Mapping for Software Messages (continued)

MSTID [7:0]	CORE NAME	DESCRIPTION
0x9	A15 Core1	ARM Master ID
0xA	Reserved	
0xB	Reserved	
0x46	QMSS PDSPs	All QMSS PDSPs share the same master ID. Differentiating between the 8 PDSPs is done through the channel number used

11.31.3 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 11-72. 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			
DSPSS	YES	YES	
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 11-73. TI XTRIG Assignment

NAME	ASSIGNED XTRIG CHANNEL NUMBER
C66x CorePac0-3	XTRIG 0-3
CPTracer 0..31 (The CPTracer number refers to the SID[4:0] as shown in Table 11-70	XTRIG 8 .. 39

11.31.4 Peripherals-Related Debug Requirement

Table 11-74 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 11-74. 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 11-74. 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						
DDR3A	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						
PCIeSS 0..1	N	N	N	N	N	
NetCP (ethernet switch)	Y	Y	Y	Y	N	27
USBSS	N	N	N	N	N	NA
Accelarators						
Reserved						
Reserved						

Table 11-74. 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		
Reserved						
Reserved						
Reserved						
FFTC_0/1	Y	Y	Y	Y	N	47/48
IQN	Y	Y	Y	N	N	53

Based on the above table the number of suspend interfaces in Keystone II devices is listed below.

Table 11-75. EMUSUSP Peripheral Summary (for EMUSUSP handshake from DEBUGSS)

INTERFACES	NUM_SUSPEND_PERIPHERALS
EMUSUSP Interfaces	54
EMUSUSP Realtime Interfaces	15

Table 11-76 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 11-76. EMUSUSP Core Summary (for EMUSUSP handshake to DEBUGSS)

CORE #	ASSIGNMENT
0..3	C66x CorePac0..3
8,9	ARM CorePac 0,1
12..29	Reserved
30	Logical OR of Core #0..11
31	Logical AND of Core #0..11

11.31.5 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)

11.31.6 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).

11.31.6.1 Trace Electrical Data/Timing

Table 11-77. Trace Switching Characteristics

(see Figure 11-68)

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

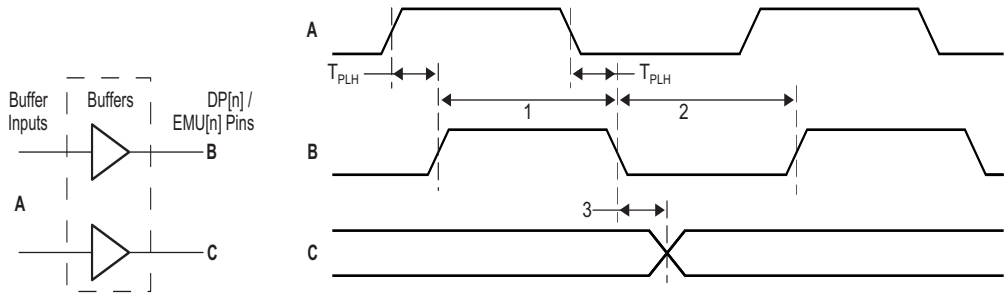


Figure 11-68. Trace Timing

11.31.7 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).

11.31.7.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the 66AK2L06 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.

11.31.7.2 JTAG Electrical Data/Timing

Table 11-78. JTAG Test Port Timing Requirements

(see Figure 11-69)

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 11-79. JTAG Test Port Switching Characteristics

(see Figure 11-69)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid		8.24	ns

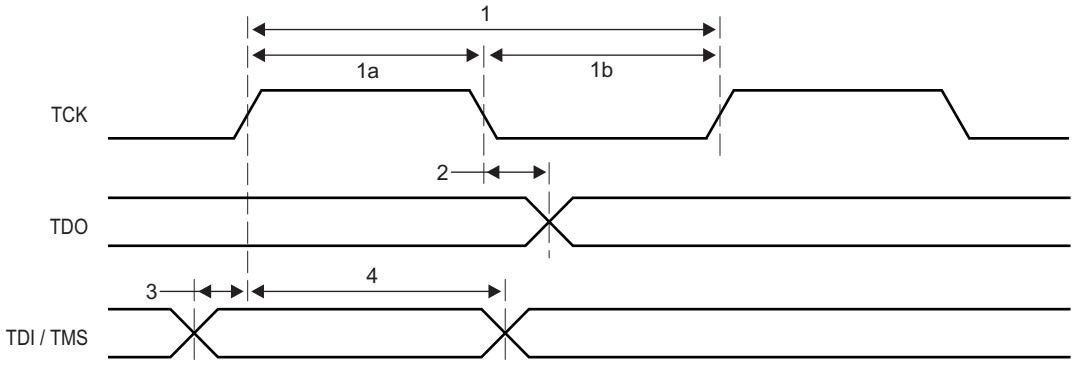


Figure 11-69. JTAG Test-Port Timing

12 Mechanical Data

12.1 Thermal Data

Table 12-1 shows the thermal resistance characteristics for the PBGA - CMS 900-pin mechanical package.

Table 12-1. Thermal Resistance Characteristics (PBGA Package) CMS

NO.			°C/W
1	$R\theta_{JC}$	Junction-to-case	.405
2	$R\theta_{JB}$	Junction-to-board	3.44

12.2 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
66AK2L06XCMS	ACTIVE	FCBGA	CMS	900	44	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 0	66AK2L06XCMS @2013 1GHZ	Samples
66AK2L06XCMS2	ACTIVE	FCBGA	CMS	900	44	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 0	66AK2L06XCMS @2013	Samples
66AK2L06XCMSA	ACTIVE	FCBGA	CMS	900	44	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 0	66AK2L06XCMS @2013 A1GHZ	Samples
66AK2L06XCMSA2	ACTIVE	FCBGA	CMS	900	44	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 0	66AK2L06XCMS @2013 A1.2GHZ	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.

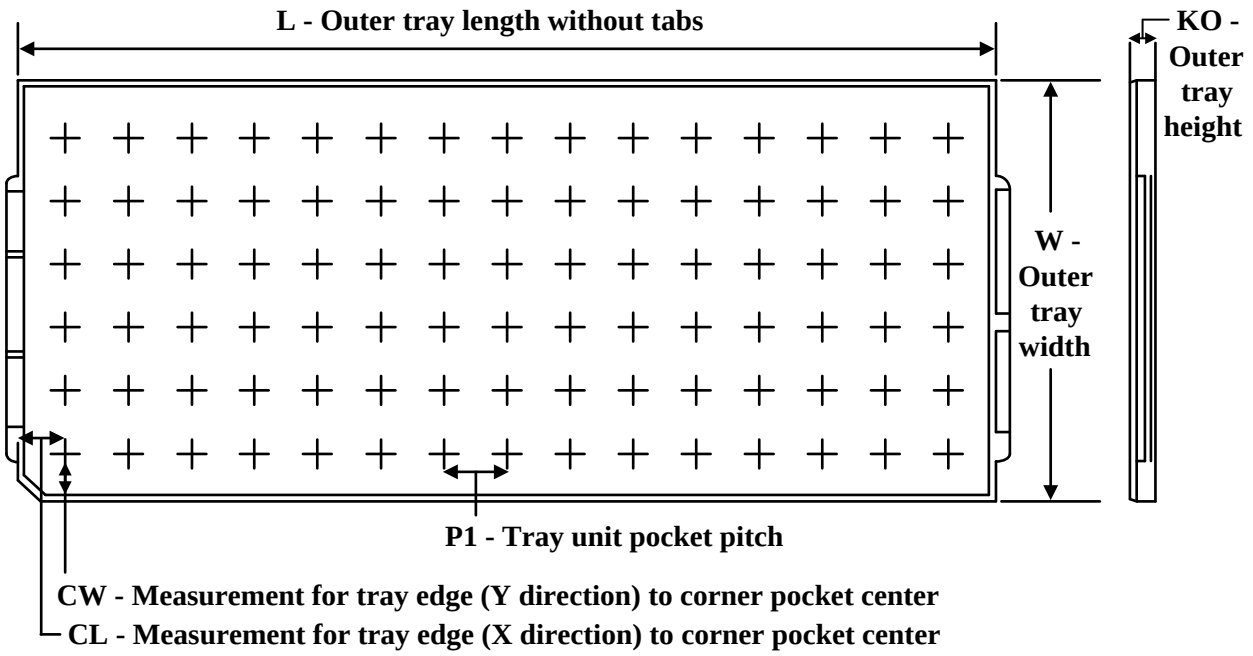
(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(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.

TRAY

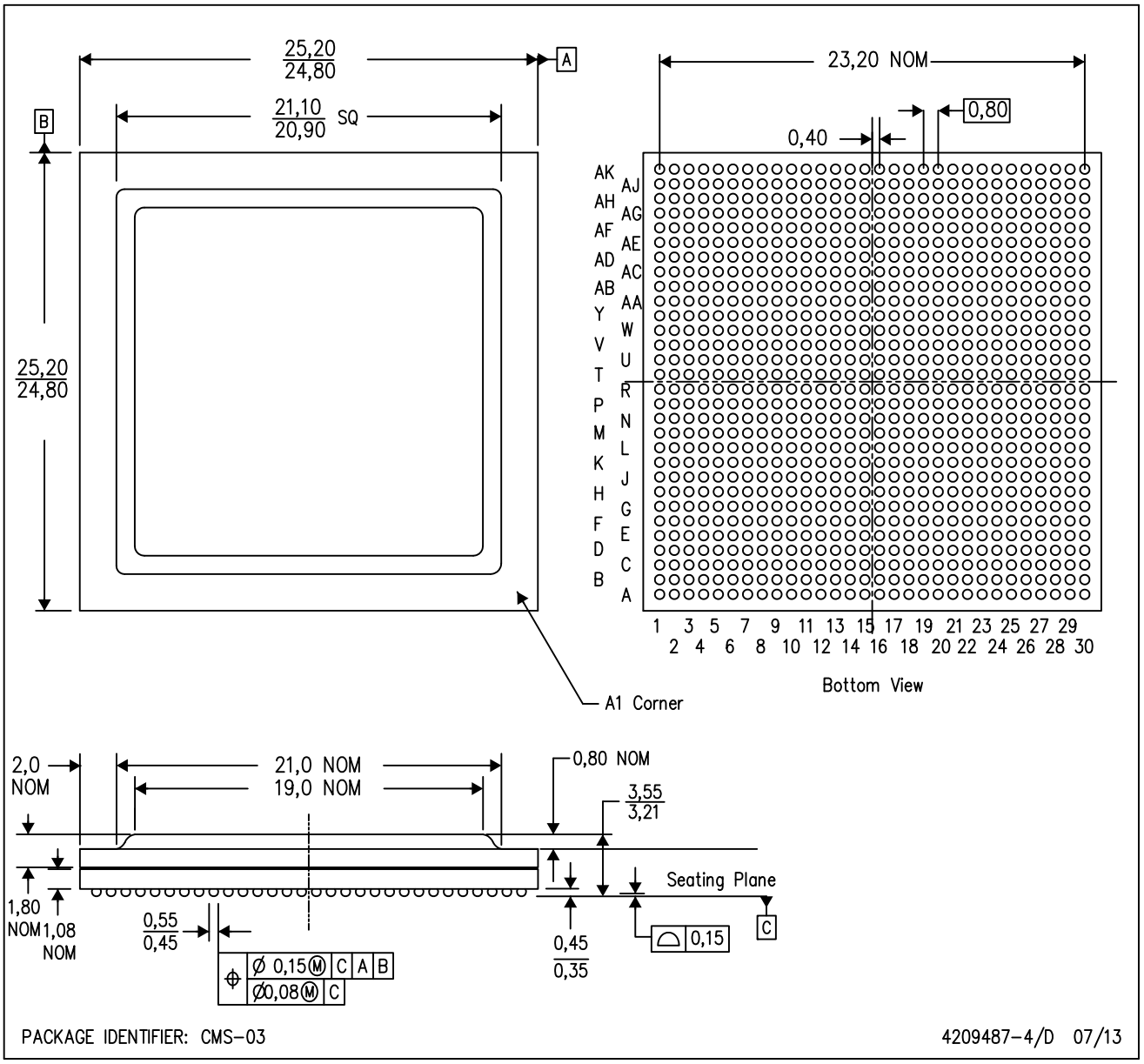


Chamfer on Tray corner indicates Pin 1 orientation of packed units.

*All dimensions are nominal

Device	Package Name	Package Type	Pins	SPQ	Unit array matrix	Max temperature (°C)	L (mm)	W (mm)	K0 (µm)	P1 (mm)	CL (mm)	CW (mm)
66AK2L06XCMS	CMS	FCBGA	900	44	4 X 11	150	315	135.9	12190	26.7	27.5	20
66AK2L06XCMS2	CMS	FCBGA	900	44	4 X 11	150	315	135.9	12190	26.7	27.5	20
66AK2L06XCMSA	CMS	FCBGA	900	44	4 X 11	150	315	135.9	12190	26.7	27.5	20
66AK2L06XCMSA2	CMS	FCBGA	900	44	4 X 11	150	315	135.9	12190	26.7	27.5	20

CMS (S–PBGA–N900) PLASTIC BALL GRID ARRAY



NOTES:
A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M–1994.
B. This drawing is subject to change without notice.
C. Thermally enhanced molded plastic package with heat slug (HSL).
D. Flip chip application only.
E. Pb–free die bump and solder ball.

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