TMS320TCI6612

Communications Infrastructure KeyStone SoC

Data Manual

Literature Number: SPRS784D
February 2013

Release History

Release	Date	Description/Comments
D	February 2013	Added TeraNet switch fabric figures in System Interconnect section Added pin map quadrants figures in Terminals section Added boot mode pins mapping table with CorePac as boot master Added HOUT timing diagram in Host Interrupt Output section Updated device nomenclature figure Updated switch fabric connection matrix tables Updated descriptions for DDR3/PASS PLL initialization sequence Updated MPU registers reset values Updated all SerDes clocks to discrete frequencies in the Clock Input Timing Requirements table Corrected jitter max values in Main PLL Controller Clock Input Timing table
C	May 2012	Updated peripheral timing numbers in peripheral information and electrical specifications chapter Updated bootmode pin mapping Added DEVSPEED register information Added wordswap section and table to ARM chapter Changed all references of ARM Subsystem to ARM CorePac Replaced DSP with SoC where applicable Added footnote explaining NOR and NAND supported memory sizes in memory map
SPRS784B	November 2011	Initial release

For detailed revision information, see "Revision History" on page A-259.

Contents

List of Figures

List of Tables

1 TMS320TCI6612 Features

- Two TMS320C66x™ DSP Core Subsystems, Each With
  - 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 Core
    - 32K Byte L1D Per Core
    - 1024K Byte Local L2 Per Core
- 1.2- GHz ARM Cortex-A8 Microprocessor
  - ARMv7-Compatible, Dual-Issue, In-Order Execution Engine
  - Includes Neon Media Coprocessor for Advanced SIMD Media Processing Architecture and VFP Architecture
  - Memory
    - 256K Byte L2 Cache
    - 32K Byte L1I
    - 32K Byte L1D
- Multicore Shared Memory Controller (MSMC)
  - 2048K Byte MSMC SRAM Memory Shared by Two DSP Cores
  - Memory Protection Unit for Both MSM SRAM and DDR3_EMIF
- Hardware Coprocessors
  - Two Enhanced Coprocessors for Turbo Decoding
    - Supports WCDMA/HSPA/HSPA+/TD-SCDMA, LTE, and WiMAX
    - Supports Up To 365 Mbps for LTE and Up to 233 Mbps for WCDMA
    - Low DSP Overhead – HW Interleaver Table Generation and CRC Check
  - Two Viterbi Decoders
    - Supports More Than 38 Mbps @ 40 bit Block Size
  - WCDMA Receive Acceleration Coprocessor
    - Up to 256 Users @ 8 Fingers w/o Measurement
  - WCDMA Transmit Acceleration Coprocessor
    - Up to 256 Users With Two Radio Links and Diversity
  - Two Fast Fourier Transform Coprocessors
    - 2048 pt FFT in 4.8 µs
  - Bit Rate Coprocessor
    - WCDMA/HSPA+, TD-SCDMA, LTE, and WiMAX Uplink and Downlink Bit Processing
    - Includes Encoding, Rate Matching/Dematching, Segmentation, Multiplexing, and More
    - Supports Up To 914 Mbps for LTE and 405 Mbps for WCDMA/TD-SCDMA
- Multicore Navigator
  - 8192 Multipurpose Hardware Queues with Queue Manager
  - Packet-Based DMA for Zero-Overhead Transfers
- Network Coprocessor
  - Packet Accelerator Enables Support for
    - Transport Plane IPsec, GTP-U, SCTP, PDCP
    - L2 User Plane PDCP (RoHC, Air Ciphering)
    - 1 Gbps Wire Speed Throughput at 1.5M Packets Per Second
  - Security Accelerator Engine Enables Support for
    - IPSec, SRTP, 3GPP and WiMAX Air Interface, and SSL/TLS Security
    - ECB, CBC, CTR, F8, A5/3, CCM, GCM, HMAC, CMAC, GMAC, AES, DES, 3DES, Kasumi, SNOW 3G, SHA-1, SHA-2 (256-bit Hash), MD5
    - Up to 2.8 Gbps Encryption Speed
- Four Rake/Search Accelerators (RSA) for
  - Chip Rate Processing for WCDMA Rel'99, HSDPA, and HSDPA+
  - Reed-Muller Decoding
- Peripherals
  - Four-Lane SerDes-Based Antenna Interface (AIF2)
    - Operating at Up to 6.144 Gbps Per Lane
    - Compliant with CPRI Standards for 3G / 4G (WCDMA, LTE TDD, LTE FDD, TD-SCDMA, and WiMAX)
  - Four Lanes of SRIO 2.1
    - 5 GBaud Operation Per Lane
    - Supports Direct I/O, Message Passing
  - Two Lanes PCIe Gen2
    - Supports Up To 5 GBaud Per Lane
  - Four Lanes of Hyperlink
    - Supports Connections to Other KeyStone Architecture Devices Providing Resource Scalability
    - Supports Combined Rate of Up to 50 Gbaud
  - Gigabit Ethernet (GbE) Switch Subsystem
    - Two SGMII Ports
    - IEEE1588 Support
  - 64-Bit DDR3 Interface with Speeds up to 1333 MHz
  - EMIF16 Interface
  - Two UART Interfaces
  - I^2C Interface
  - 32 GPIO pins
  - SPI Interface
  - USIM Interface
  - Semaphore Module
  - Ten 64-Bit Timers
  - Three On-Chip PLLs
  - SoC Security Support
- Commercial Temperature:
  - 0°C to 100°C
- Extended Temperature:
  - - 40°C to 100°C

1.1 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 that provides adequate internal bandwidth for nonblocking access to all processing cores, peripherals, coprocessors, and I/O. This is achieved with four main hardware elements: Multicore Navigator, TeraNet, Multicore Shared Memory Controller, and HyperLink.

Multicore Navigator is an innovative packet-based manager that controls 8192 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 two 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 TeraNet's capacity, so packet movement cannot be blocked by memory access.

HyperLink provides a 50-Gbps chip-level interconnect that allows SoCs to work in tandem. Its low-protocol overhead and high throughput make Hyperlink an ideal interface for chip-to-chip interconnections. Working with Multicore Navigator, HyperLink dispatches tasks to tandem devices transparently and executes tasks as if they are running on local resources.

1.2 Device Description

The TMS320TCI6612 Communications Infrastructure KeyStone SoC is a member of the C66xx SoC family based on TI's new KeyStone Multicore SoC Architecture designed specifically for high performance wireless infrastructure applications. The TCI6612 provides a very high performance Enterprise/Pico basestation platform for developing all wireless standards including WCDMA/HSPA/HSPA+, TD-SCDMA, GSM, TDD-LTE, FDD-LTE, and WiMAX. Even with aggregate data rates for 20-MHz LTE systems above 400 Mbps per sector, the TCI6612 can support two sectors running at full rate. The TCI6612 also sets a new standard for clock speed with operating frequencies up to 1.2 GHz.

The TCI6612 supports a dual mode of operation for simultaneous support of WCDMA and LTE. It is pin compatible with the TCI6614.

TI's SoC architecture provides a programmable platform integrating various subsystems (C66x cores, IP network, radio layers 1 and 2, and transport 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 A8 Core microprocessor in the TCI6612 enables the ability for layer 3 processing on-chip. Operations such as Traffic Control, Local O&M, NBAP, and SCTP processing can all be performed with the ARM A8 Core.

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 core incorporates 90 new instructions targeted for floating point (FPi) and vector math oriented (VPi) processing. These enhancements yield tremendous performance improvements in multi-antenna 4.8G signal processing for algorithms like MIMO and beamforming.

The TCI6612 contains many wireless basestation coprocessors to offload the bulk of the processing demands of layer 1 and layer 2 base station processing. This keeps the cores free for receiver algorithms and other differentiating functions. The SoC contains several copies of key coprocessors such as the FFTC and TCP3d. A key coprocessor for enabling high data rates is the bit rate coprocessor (BCP), which handles the entire downlink bit processing chain and much of the receive bit processing. The architectural elements of the SoC (Multicore Navigator) ensure that all the bits are processed without any CPU intervention or overhead, allowing the system to make optimal use of its resources.

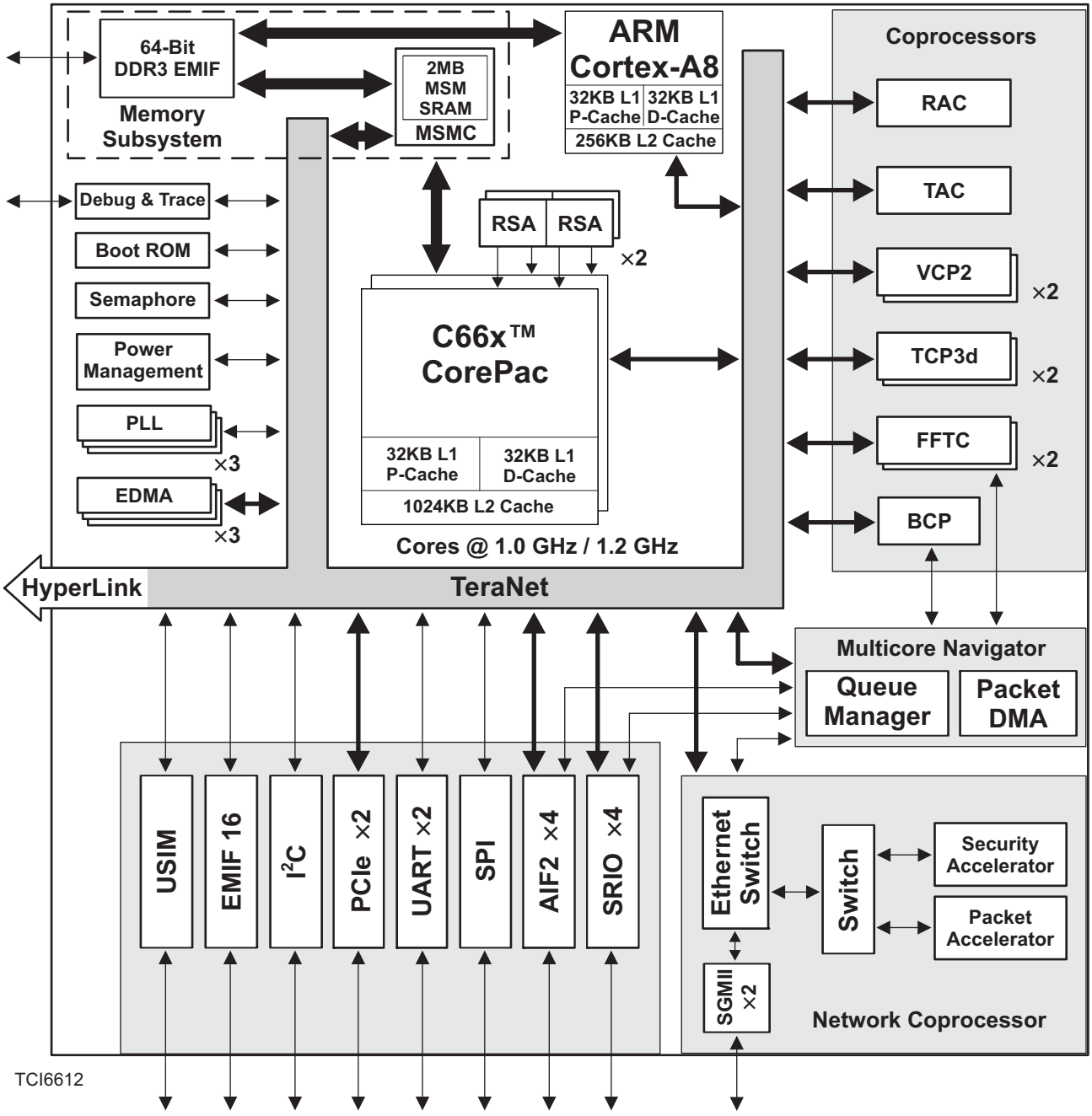
TI's scalable multicore SoC architecture solutions provide developers with a range of software- and hardware-compatible devices to minimize development time and maximize reuse across all base station platforms from Femto to Macro.

The TCI6612 device has a complete set of development tools that includes: a C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows debugger interface for visibility into source code execution.

1.3 Functional Block Diagram

Figure 1-1 shows the functional block diagram of the TMS320TCI6612 device.

Figure 1-1 Functional Block Diagram



2 Device Overview

2.1 Device Description

Table 2-1 provides an overview of the TMS320TCI6612 SoC. The table shows significant features of the TCI6612 device, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

Table 2-1 TCI6612 Processor Description (Part 1 of 2)

HARDWARE FEATURES		TMS320TCI6612
Peripherals	DDR3 Memory Controller (64-bit bus width) [1.5-V I/O] (clock source = DDRREFCLKN|P)	1
	EDMA3 (16 independent channels) [CPU/2 clock rate]	1
	EDMA3 (64 independent channels) [CPU/3 clock rate]	2
	High-speed 1×/2×/4× Serial RapidIO Port (4 lanes)	1
	Second-generation Antenna Interface (AIF2) (4 lanes)	1
	I^2C	1
	SPI	1
	PCIe (2 lanes)	1
	UART	2
	10/100/1000 Gigabit Ethernet (GbE) Switch Subsystem	2
	Management Data Input/Output (MDIO)	1
	USIM (clock source = CPU/6 clock frequency)	1
	EMIF16 (clock source = CPU/6 clock frequency)	1
	64-Bit Timers (Configurable) (internal clock source = CPU/6 clock frequency)	Ten 64-bit (each configurable as two 32-bit timers, except Timer 0/1)
	General-Purpose Input/Output Port (GPIO)	32
Encoder/Decoder Coprocessors	VCP2 (clock source = CPU/3 clock frequency)	2
	TCP3d (clock source = CPU/2 clock frequency)	2
	FFTC (clock source = CPU/3 clock frequency)	2
	BCP (clock source = CPU/3 clock frequency)	1
Accelerators	Receive Accelerator (RAC)	1
	Transmit Accelerator (TAC)	1
	Rake/Search Accelerator	4
	Packet Accelerator	1
	Security Accelerator (1)	1
On-Chip Memory	Size (Bytes)	4912KB
	Organization	64KB CorePac L1 program memory controller [SRAM/Cache] 64KB CorePac L1 data memory controller [SRAM/Cache] 2048KB CorePac L2 unified memory/cache 256KB ARM L2 cache 32KB ARM L1I 32KB ARM L1D 128KB ARM CorePac secure ROM 48KB ARM CorePac public ROM 64KB ARM CorePac OCM RAM 2048KB MSMC SRAM 128KB L3 ROM
C66x CorePac Revision ID	CorePac Revision ID Register (address location: 0181 2000h)	See Section 5.5 "CorePac Revision" on page 116.

Table 2-1 TCI6612 Processor Description (Part 2 of 2)

HARDWARE FEATURES		TMS320TCI6612
JTAG BSDL_ID	JTAGID register (address location: 0x02620018)	See Section 3.3.3 "JTAG ID (JTAGID) Register Description" on page 82
Frequency	MHz	1200 (1.2 GHz) 1000 (1.0 GHz)
Cycle Time	ns	0.83 ns (1.2 GHz) 1 ns (1.0 GHz)
Voltage	Core (V)	SmartReflex variable supply
	I/O (V)	1.0 V, 1.5 V, and 1.8 V
BGA Package	25 mm × 25 mm	900-Pin flip-chip plastic BGA
Process Technology	µm	0.040 µm
Product Status (2)	Product Preview (PP), Advance Information (AI), or Production Data (PD)	PD
End of Table 2-1		

1 The Security Accelerator function is subject to export control and will be enabled *only* for approved device shipments.
2 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.

The C66x Central Processing Unit (CPU) 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 CPUs support 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. On the C66x CPU, the vector processing capability is improved by extending the width of the SIMD instructions. C66x CPUs can execute instructions that operate on 128-bit vectors. For example the QMPY32 instruction is able to perform the element-to-element multiplication between two vectors of four 32-bit data each. 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.

The C66x CPU consists of eight functional units, two register files, and two data paths as shown in Figure 2-1. The two general-purpose register files (A and B) each contain 32 32-bit registers for a total of 64 registers. The general-purpose registers can be used for data or can be data address pointers. The data types supported include packed 8-bit data, packed 16-bit data, 32-bit data, 40-bit data, and 64-bit data. Multiplies also support 128-bit data. 40-bit-long or 64-bit-long values are stored in register pairs, with the 32 LSBs of data placed in an even register and the remaining 8 or 32 MSBs in the next upper register (which is always an odd-numbered register). 128-bit data values are stored in register quadruplets, with the 32 LSBs of data placed in a register that is a multiple of 4 and the remaining 96 MSBs in the next 3 upper registers.

The eight functional units (.M1, .L1, .D1, .S1, .M2, .L2, .D2, and .S2) are each capable of executing one instruction every clock cycle. The .M functional units perform all multiply operations. The .S and .L units perform a general set of arithmetic, logical, and branch functions. The .D units primarily load data from memory to the register file and store results from the register file into memory.

Each C66x .M unit can perform one of the following fixed-point operations each clock cycle: four 32 × 32 bit multiplies, sixteen 16 × 16 bit multiplies, four 16 × 32 bit multiplies, four 8 × 8 bit multiplies, four 8 × 8 bit multiplies with add operations, and four 16 × 16 multiplies with add/subtract capabilities. There is also support for Galois field multiplication for 8-bit and 32-bit data. Many communications algorithms such as FFTs and modems require complex multiplication. Each C66x .M unit can perform one 16 × 16 bit complex multiply with or without rounding capabilities, two 16 × 16 bit complex multiplies with rounding capability, and a 32 × 32 bit complex multiply with rounding capability. The C66x can also perform two 16 × 16 bit and one 32 × 32 bit complex multiply instructions

that multiply a complex number with a complex conjugate of another number with rounding capability. Communication signal processing also requires an extensive use of matrix operations. Each C66x .M unit is capable of multiplying a [1 × 2] complex vector by a [2 × 2] complex matrix per cycle with or without rounding capability. A version also exists allowing multiplication of the conjugate of a [1 × 2] vector with a [2 × 2] complex matrix.

Each C66x .M unit also includes IEEE floating-point multiplication operations from the C674x CPU. This includes one single-precision multiply each cycle and one double precision multiply every 4 cycles. There is also a mixed-precision multiply that allows multiplication of a single-precision value by a double-precision value and an operation allowing multiplication of two single-precision numbers resulting in a double-precision number. The C66x CPU improves the performance over the C674x double-precision multiplies by adding a instruction allowing one double-precision multiply per cycle and also reduces the number of delay slots from 10 down to 4. Each C66x .M unit can also perform one the following floating-point operations each clock cycle: one, two, or four single-precision multiplies or a complex single-precision multiply.

The .L and .S units can now support up to 64-bit operands. This allows for new versions of many of the arithmetic, logical, and data packing instructions to allow for more parallel operations per cycle. Additional instructions were added yielding performance enhancements of the floating point addition and subtraction instructions, including the ability to perform one double precision addition or subtraction per cycle. Conversion to/from integer and single-precision values can now be done on both .L and .S units on the C66x. Also, by taking advantage of the larger operands, instructions were also added to double the number of these conversions that can be done. The .L unit also has additional instructions for logical AND and OR instructions, as well as, 90 degree or 270 degree rotation of complex numbers (up to two per cycle). Instructions have also been added that allow for the computing the conjugate of a complex number.

The MFENCE instruction is a new instruction introduced on the C66x SoC. This instruction will create a CPU stall until the completion of all the CPU-triggered memory transactions, including:

- Cache line fills
- Writes from L1D to L2 or from the CorePac to MSMC and/or other system endpoints
- Victim write backs
- Block or global coherence operations
- Cache mode changes
- Outstanding XMC prefetch requests

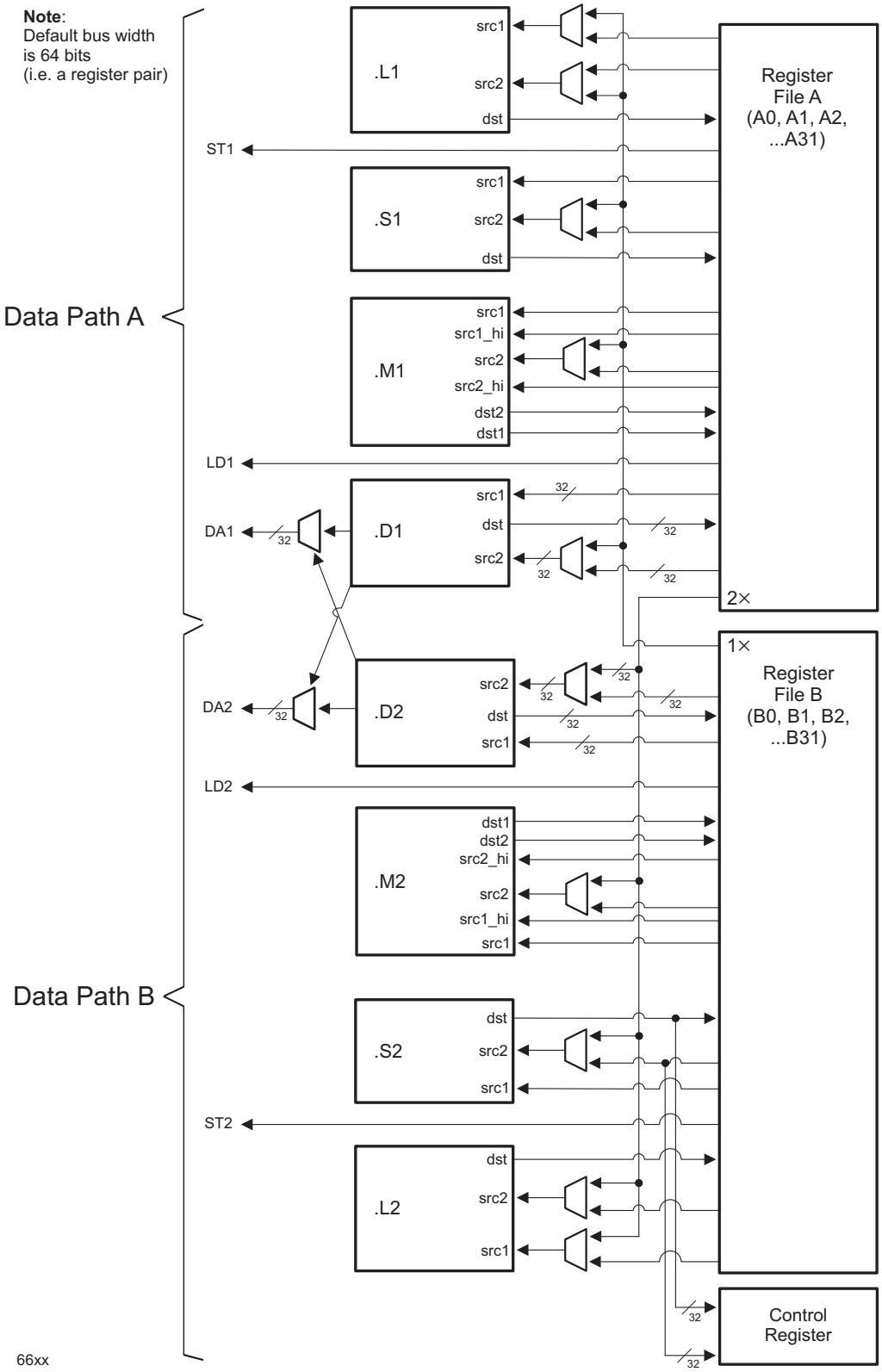
This is useful as a simple mechanism for programs to wait for these requests to reach their endpoint. It also provides ordering guarantees for writes arriving at a single endpoint via multiple paths, multiprocessor algorithms that depend on ordering, and manual coherence operations.

For more details on the C66x CPU and its enhancements over the C64x+ and C674x architectures, see the following documents (2.13 "Related Documentation from Texas Instruments" on page 75):

- *C66x CPU and Instruction Set Reference Guide*
- *C66x DSP Cache User Guide*
- *C66x CorePac User Guide*

Figure 2-1 shows the DSP core functional units and data paths.

Figure 2-1 TMS320TCI6612 CPU (DSP Core) Data Paths



2.2 Memory Map Summary

Table 2-2 shows the memory map address ranges of the TMS320TCI6612 device.

Table 2-2 Memory Map Summary (Part 1 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0000 0000	007F FFFF	0 0000 0000	0 007F FFFF	8M	Reserved
0080 0000	008F FFFF	0 0080 0000	0 008F FFFF	1M	L2 SRAM
0090 0000	00DF FFFF	0 0090 0000	0 00DF FFFF	5M	Reserved
00E00000	00E0 7FFF	0 00E00000	0 00E0 7FFF	32K	L1P SRAM
00E08000	00EF FFFF	0 00E08000	0 00EF FFFF	1M-32K	Reserved
00F00000	00F0 7FFF	0 00F00000	0 00F0 7FFF	32K	L1D SRAM
00F08000	00FF FFFF	0 00F08000	0 00FF FFFF	1M-32K	Reserved
0100 0000	01BF FFFF	0 0100 0000	0 01BF FFFF	12 M	C66x CorePac registers
01C0 0000	01CF FFFF	0 01C0 0000	0 01CF FFFF	1M	Reserved
01D0 0000	01D0 007F	0 01D0 0000	0 01D0 007F	128	Tracer_MSMC_0
01D0 0080	01D0 7FFF	0 01D0 0080	0 01D0 7FFF	32K-128	Reserved
01D0 8000	01D0 807F	0 01D0 8000	0 01D0 807F	128	Tracer_MSMC_1
01D0 8080	01D0 FFFF	0 01D0 8080	0 01D0 FFFF	32K-128	Reserved
01D1 0000	01D1 007F	0 01D1 0000	0 01D1 007F	128	Tracer_MSMC_2
01D1 0080	01D1 7FFF	0 01D1 0080	0 01D1 7FFF	32K-128	Reserved
01D1 8000	01D1 807F	0 01D1 8000	0 01D1 807F	128	Tracer_MSMC_3
01D1 8080	01D1 FFFF	0 01D1 8080	0 01D1 FFFF	32K-128	Reserved
01D2 0000	01D2 007F	0 01D2 0000	0 01D2 007F	128	Tracer_QM_DMA
01D2 0080	01D2 7FFF	0 01D2 0080	0 01D2 7FFF	32K-128	Reserved
01D2 8000	01D2 807F	0 01D2 8000	0 01D2 807F	128	Tracer_DDR
01D2 8080	01D2 FFFF	0 01D2 8080	0 01D2 FFFF	32K-128	Reserved
01D3 0000	01D3 007F	0 01D3 0000	0 01D3 007F	128	Tracer_SM
01D3 0080	01D3 7FFF	0 01D3 0080	0 01D3 7FFF	32K-128	Reserved
01D3 8000	01D3 807F	0 01D3 8000	0 01D3 807F	128	Tracer_QM_CFG
01D3 8080	01D3 FFFF	0 01D3 8080	0 01D3 FFFF	32K-128	Reserved
01D4 0000	01D4 007F	0 01D4 0000	0 01D4 007F	128	Tracer_CFG
01D4 0080	01D4 7FFF	0 01D4 0080	0 01D4 7FFF	32K-128	Reserved
01D4 8000	01D4 807F	0 01D4 8000	0 01D4 807F	128	Tracer_L2_0
01D4 8080	01D4 FFFF	0 01D4 8080	0 01D4 FFFF	32K-128	Reserved
01D5 0000	01D5 007F	0 01D5 0000	0 01D5 007F	128	Tracer_L2_1
01D5 0080	01D5 7FFF	0 01D5 0080	0 01D5 7FFF	32K-128	Reserved
01D5 8000	01D5 807F	0 01D5 8000	0 01D5 807F	128	Reserved
01D5 8080	01D5 FFFF	0 01D5 8080	0 01D5 FFFF	32K-128	Reserved
01D6 0000	01D6 007F	0 01D6 0000	0 01D6 007F	128	Reserved
01D6 0080	01D6 7FFF	0 01D6 0080	0 01D6 7FFF	32K-128	Reserved
01D6 8000	01D6 807F	0 01D6 8000	0 01D6 807F	128	Tracer_RAC_FE
01D6 8080	01D6 FFFF	0 01D6 8080	0 01D6 FFFF	32K-128	Reserved
01D7 0000	01D7 007F	0 01D7 0000	0 01D7 007F	128	Tracer_RAC_CFG
01D7 0080	01D7 7FFF	0 01D7 0080	0 01D7 7FFF	32K-128	Reserved
01D7 8000	01D7 807F	0 01D7 8000	0 01D7 807F	128	Tracer_TAC

Table 2-2 Memory Map Summary (Part 2 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
01D7 8080	01D7 FFFF	0 01D7 8080	0 01D7 FFFF	32K-128	Reserved
01D8 0000	01D8 007F	0 01D8 0000	0 01D8 007F	128	Tracer_TNet_6P_A
01D8 0080	01D8 7FFF	0 01D8 0080	0 01D8 7FFF	32K-128	Reserved
01D8 8000	01D8 807F	01D8 8000	01D8 807F	128	Tracer_DDR_2
01D8 8080	01DF FFFF	01D8 8080	0 01DF FFFF	480K-128	Reserved
01E0 0000	01E3 FFFF	0 01E0 0000	0 01E3 FFFF	256K	Reserved
01E4 0000	01E7 FFFF	0 01E4 0000	0 01E7 FFFF	256K	Reserved
01E8 0000	01EB FFFF	0 01E8 0000	0 01EB FFFF	256K	Reserved
01EC 0000	01EF FFFF	0 01EC 0000	0 01EF FFFF	256K	Reserved
01F0 0000	01F7 FFFF	0 01F0 0000	0 01F7 FFFF	512k	AIF2 control
01F8 0000	01F8 FFFF	0 01F8 0000	0 01F8 FFFF	64K	Reserved
01F9 0000	01F9 FFFF	0 01F9 0000	0 01F9 FFFF	64K	Reserved
01FA 0000	01FB FFFF	0 01FA 0000	0 01FB FFFF	128K	Reserved
01FC 0000	01FD FFFF	0 01FC 0000	0 01FD FFFF	128K	Reserved
01FE 0000	01FF FFFF	0 01FE 0000	0 01FF FFFF	128k	Reserved
0200 0000	0208 FFFF	0 0200 0000	0 0208 FFFF	576K	Packet Accelerator configuration
0209 0000	020B FFFF	0 0209 0000	0 020B FFFF	192K	Ethernet switch subsystem configuration
020C 0000	020F FFFF	0 020C 0000	0 020F FFFF	256K	Security Accelerator subsystem configuration
0210 0000	0210 FFFF	0 0210 0000	0 0210 FFFF	64K	RAC - FEI control
0211 0000	0211 FFFF	0 0211 0000	0 0211 FFFF	64K	RAC - BEI control
0212 0000	0213 FFFF	0 0212 0000	0 0213 FFFF	128K	RAC - GCCP 0 control
0214 0000	0215 FFFF	0 0214 0000	0 0215 FFFF	128K	RAC - GCCP 1 control
0216 0000	0217 FFFF	0 0216 0000	0 0217 FFFF	128K	Reserved
0218 0000	0218 7FFF	0 0218 0000	0 0218 7FFF	32k	TAC - FEI control
0218 8000	0218 FFFF	0 0218 8000	0 0218 FFFF	32k	TAC- BEI control
0219 0000	0219 FFFF	0 0219 0000	0 0219 FFFF	64k	TAC - SGCCP 0 control
021A 0000	021A FFFF	0 021A 0000	0 021A FFFF	64K	TAC - SGCCP 1 Control
021B 0000	021B FFFF	0 021B 0000	0 021B FFFF	64K	Reserved
021C 0000	021C 03FF	0 021C 0000	0 021C 03FF	1K	TCP3d-A
021C 0400	021C 7FFF	0 021C 0400	0 021C 7FFF	31K	Reserved
021C 8000	021C 83FF	0 021C 8000	0 021C 83FF	1K	TCP3d-B
021C 8400	021C FFFF	0 021C 8400	0 021C FFFF	31K	Reserved
021D 0000	021D 00FF	0 021D 0000	0 021D 00FF	256	VCP2_A
021D 0100	021D 3FFF	0 021D 0100	0 021D 3FFF	16K	Reserved
021D 4000	021D 40FF	0 021D 4000	0 021D 40FF	256	VCP2_B
021D 4100	021D 7FFF	0 021D 4100	0 021D 7FFF	16K	Reserved
021D 8000	021D 80FF	0 021D 8000	0 021D 80FF	256	Reserved
021D 8100	021D BFFF	0 021D 8100	0 021D BFFF	16K	Reserved
021D C000	021D C0FF	0 021D C000	0 021D C0FF	256	Reserved
021D C100	021D FFFF	0 021D C100	0 021D FFFF	16K	Reserved
021E 0000	021E 0FFF	0 021E 0000	0 021E 0FFF	4K	Reserved
021E 1000	021E FFFF	0 021E 1000	0 021E FFFF	60k	Reserved
021F 0000	021F 07FF	0 021F 0000	0 021F 07FF	2K	FFTC-A configuration

Table 2-2 Memory Map Summary (Part 3 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
021F 0800	021F 3FFF	0 021F 0800	0 021F 3FFF	14K	Reserved
021F 4000	021F 47FF	0 021F 4000	0 021F 47FF	2K	FFTC-B configuration
021F 4800	021F FFFF	0 021F 4800	0 021F FFFF	46K	Reserved
0220 0000	0220 007F	0 0220 0000	0 0220 007F	128	Timer0
0220 0080	0220 FFFF	0 0220 0080	0 0220 FFFF	64K-128	Reserved
0221 0000	0221 007F	0 0221 0000	0 0221 007F	128	Timer1
0221 0080	0221 FFFF	0 0221 0080	0 0221 FFFF	64K-128	Reserved
0222 0000	0222 007F	0 0222 0000	0 0222 007F	128	Reserved
0222 0080	0222 FFFF	0 0222 0080	0 0222 FFFF	64K-128	Reserved
0223 0000	0223 007F	0 0223 0000	0 0223 007F	128	Reserved
0223 0080	0223 FFFF	0 0223 0080	0 0223 FFFF	64K-128	Reserved
0224 0000	0224 007F	0 0224 0000	0 0224 007F	128	Timer4
0224 0080	0224 FFFF	0 0224 0080	0 0224 FFFF	64K-128	Reserved
0225 0000	0225 007F	0 0225 0000	0 0225 007F	128	Timer5
0225 0080	0225 FFFF	0 0225 0080	0 0225 FFFF	64K-128	Reserved
0226 0000	0226 007F	0 0226 0000	0 0226 007F	128	Timer6
0226 0080	0226 FFFF	0 0226 0080	0 0226 FFFF	64K-128	Reserved
0227 0000	0227 007F	0 0227 0000	0 0227 007F	128	Timer7
0227 0080	0227 FFFF	0 0227 0080	0 0227 FFFF	64K-128	Reserved
0228 0000	0228 007F	0 0228 0000	0 0228 007F	128	Timer8
0228 0080	0228 FFFF	0 0228 0080	0 0228 FFFF	64K-128	Reserved
0229 0000	0229 007F	0 0229 0000	0 0229 007F	128	Timer9
0229 0080	0229 FFFF	0 0229 0080	0 0229 FFFF	64K-128	Reserved
022A 0000	022A 007F	0 022A 0000	0 022A 007F	128	Timer10
022A 0080	022A FFFF	0 022A 0080	0 022A FFFF	64K-128	Reserved
022B 0000	022B 007F	0 022B 0000	0 022B 007F	128	Timer11
022B 0080	022B FFFF	0 022B 0080	0 022B FFFF	64K-128	Reserved
022C 0000	022C 007F	0 022C 0000	0 022C 007F	128	Reserved
022C 0080	022C FFFF	0 022C 0080	0 022C FFFF	64K-128	Reserved
022D 0000	022D 007F	0 022D 0000	0 022D 007F	128	Reserved
022D 0080	022D FFFF	0 022D 0080	0 022D FFFF	64K-128	Reserved
022E 0000	022E 007F	0 022E 0000	0 022E 007F	128	Reserved
022E 0080	022E FFFF	0 022E 0080	0 022E FFFF	64K-128	Reserved
022F 0000	022F 007F	0 022F 0000	0 022F 007F	128	Reserved
022F 0080	022F FFFF	0 022F 0080	0 022F FFFF	64K-128	Reserved
0230 0000	0230 FFFF	0 0230 0000	0 0230 FFFF	64K	Reserved
0231 0000	0231 01FF	0 0231 0000	0 0231 01FF	512	PLL Controller
0231 0200	0231 FFFF	0 0231 0200	0 0231 FFFF	64K-512	Reserved
0232 0000	0232 01FF	0 0232 0000	0 0232 01FF	512	GPIO
0232 0200	0232 FFFF	0 0232 0200	0 0232 FFFF	64K-512	Reserved
0233 0000	0233 03FF	0 0233 0000	0 0233 03FF	1K	SmartReflex
0233 0400	0233 FFFF	0 0233 0400	0 0233 FFFF	63K	Reserved
0234 0000	0234 FFFF	0 0234 0000	0 0234 FFFF	64K	Reserved

Table 2-2 Memory Map Summary (Part 4 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0235 0000	0235 0FFF	0 0235 0000	0 0235 0FFF	4K	Power Sleep Controller (PSC)
0235 1000	0235 FFFF	0 0235 1000	0 0235 FFFF	64K-4K	Reserved
0236 0000	0236 03FF	0 0236 0000	0 0236 03FF	1K	Memory Protection Unit (MPU) 0
0236 0400	0236 7FFF	0 0236 0400	0 0236 7FFF	31K	Reserved
0236 8000	0236 83FF	0 0236 8000	0 0236 83FF	1K	Memory Protection Unit (MPU) 1
0236 8400	0236 FFFF	0 0236 8400	0 0236 FFFF	31K	Reserved
0237 0000	0237 03FF	0 0237 0000	0 0237 03FF	1K	Memory Protection Unit (MPU) 2
0237 0400	0237 7FFF	0 0237 0400	0 0237 7FFF	31K	Reserved
0237 8000	0237 83FF	0 0237 8000	0 0237 83FF	1K	Memory Protection Unit (MPU) 3
0237 8400	0237 FFFF	0 0237 8400	0 0237 FFFF	31K	Reserved
0238 0000	0238 03FF	0 0238 0000	0 0238 03FF	1K	Memory Protection Unit (MPU) 4
0238 0400	0237 FFFF	0 0238 0400	0 0237 FFFF	31K	Reserved
0238 8000	0238 83FF	0 0238 8000	0 0238 83FF	1K	Memory Protection Unit (MPU) 5
0238 8400	0238 FFFF	0 0238 8400	0 0238 FFFF	31K	Reserved
0239 0000	0239 3FFF	0 0239 0000	0 0239 3FFF	1K	Memory Protection Unit (MPU) 6
0239 0400	0239 7FFF	0 0239 0400	0 0239 7FFF	31K	Reserved
0239 8000	0239 83FF	0 0239 8000	0 0239 83FF	1K	Memory Protection Unit (MPU) 7
0239 8400	0243 FFFF	0 0239 8400	0 0243 FFFF	687K	Reserved
0244 0000	0244 3FFF	0 0244 0000	0 0244 3FFF	16K	DSP trace formatter 0
0244 4000	0244 FFFF	0 0244 4000	0 0244 FFFF	48K	Reserved
0245 0000	0245 3FFF	0 0245 0000	0 0245 3FFF	16K	DSP trace formatter 1
0245 4000	0245 FFFF	0 0245 4000	0 0245 FFFF	48K	Reserved
0246 0000	0246 3FFF	0 0246 0000	0 0246 3FFF	16K	Reserved
0246 4000	0246 FFFF	0 0246 4000	0 0246 FFFF	48K	Reserved
0247 0000	0247 3FFF	0 0247 0000	0 0247 3FFF	16K	Reserved
0247 4000	0247 FFFF	0 0247 4000	0 0247 FFFF	48K	Reserved
0248 0000	0248 3FFF	0 0248 0000	0 0248 3FFF	16K	Reserved
0248 4000	0248 FFFF	0 0248 4000	0 0248 FFFF	48K	Reserved
0249 0000	0249 3FFF	0 0249 0000	0 0249 3FFF	16K	Reserved
0249 4000	0249 FFFF	0 0249 4000	0 0249 FFFF	48K	Reserved
024A 0000	024A 3FFF	0 024A 0000	0 024A 3FFF	16K	Reserved
024A 4000	024A FFFF	0 024A 4000	0 024A FFFF	48K	Reserved
024B 0000	024B 3FFF	0 024B 0000	0 024B 3FFF	16K	Reserved
024B 4000	024B FFFF	0 024B 4000	0 024B FFFF	48K	Reserved
024C 0000	024C 01FF	0 024C 0000	0 024C 01FF	512	Reserved
024C 0200	024C 03FF	0 024C 0200	0 024C 03FF	1K-512	Reserved
024C 0400	024C 07FF	0 024C 0400	0 024C 07FF	1K	Reserved
024C 0800	024C FFFF	0 024C 0800	0 024C FFFF	62K	Reserved
024D 0000	024F FFFF	0 024D 0000	0 024F FFFF	192K	Reserved
0250 0000	0250 007F	0 0250 0000	0 0250 007F	128	Reserved
0250 0080	0250 7FFF	0 0250 0080	0 0250 7FFF	32K-128	Reserved
0250 8000	0250 FFFF	0 0250 8000	0 0250 FFFF	32K	Reserved
0251 0000	0251 FFFF	0 0251 0000	0 0251 FFFF	64K	Reserved

Table 2-2 Memory Map Summary (Part 5 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0252 0000	0252 03FF	0 0252 0000	0 0252 03FF	1K	SEC_KEY_MGR_A
0252 0400	0252 0FFF	0 0252 0400	0 0252 0FFF	3K	Reserved
0252 1000	0252 13FF	0 0252 1000	0 0252 13FF	1K	SEC_KEY_MGR_B
0252 1400	0252 1FFF	0 0252 1400	0 0252 1FFF	3K	Reserved
0252 2000	0252 2FFF	0 0252 2000	0 0252 2FFF	4K	OTP Memory
0252 3000	0252 7FFF	0 0252 3000	0 0252 7FFF	20K	Reserved
0252 8000	0252 8FFF	0 0252 8000	0 0252 8FFF	4K	USIM
0252 9000	0252 FFFF	0 0252 9000	0 0252 FFFF	28K	Reserved
0253 0000	0253 007F	0 0253 0000	0 0253 007F	128	I^2C data & control
0253 0080	0253 FFFF	0 0253 0080	0 0253 FFFF	64K-128	Reserved
0254 0000	0254 003F	0 0254 0000	0 0254 003F	64	UART_0
0254 0400	0254 0FFF	0 0254 0400	0 0254 0FFF	64K-64	Reserved
0254 1000	0254 103F	0 0254 1000	0 0254 103F	64	UART_1
0254 1040	0254 FFFF	0 0254 1040	0 0254 FFFF	63K-64	Reserved
0255 0000	0257 FFFF	0 0255 0000	0 0257 FFFF	192K	Reserved
0258 0000	025B FFFF	0 0258 0000	0 025B FFFF	256K	Debug_SS VBUSP (includes both CFG and coresight ETB space)
025C 0000	025F FFFF	0 025C 0000	0 025F FFFF	256K	Reserved
0260 0000	0260 1FFF	0 0260 0000	0 0260 1FFF	8K	Secondary Chip Interrupt Controller CIC 0
0260 2000	0260 3FFF	0 0260 2000	0 0260 3FFF	8K	Reserved
0260 4000	0260 5FFF	0 0260 4000	0 0260 5FFF	8K	Secondary Chip Interrupt Controller CIC 1
0260 6000	0260 7FFF	0 0260 6000	0 0260 7FFF	8K	Reserved
0260 8000	0260 9FFF	0 0260 8000	0 0260 9FFF	8K	Secondary Chip Interrupt Controller CIC 2
0260 A000	0260 BFFF	0 0260 A000	0 0260 BFFF	8K	Reserved
0260 C000	0260 DFFF	0 0260 C000	0 0260 DFFF	8K	Secondary Chip Interrupt Controller CIC 3 (for ARM)
0260 E000	0260 FFFF	0 0260 E000	0 0260 FFFF	8K	Reserved
0261 0000	0261 0FFF	0 0261 0000	0 0261 0FFF	4K	INTD
0261 1000	0261 FFFF	0 0261 1000	0 0261 FFFF	60K	Reserved
0262 0000	0262 07FF	0 0262 0000	0 0262 07FF	2K	Chip-level registers
0262 0800	0262 FFFF	0 0262 0800	0 0262 FFFF	62K	Reserved
0263 0000	0263 FFFF	0 0263 0000	0 0263 FFFF	64K	Reserved
0264 0000	0264 07FF	0 0264 0000	0 0264 07FF	2K	Semaphore
0264 0800	0264 FFFF	0 0264 0800	0 0264 FFFF	64K-2K	Reserved
0265 0000	026F FFFF	0 0265 0000	0 026F FFFF	704K	Reserved
0270 0000	0270 7FFF	0 0270 0000	0 0270 7FFF	32K	EDMA channel controller (EDMA3CC) 0
0270 8000	0271 FFFF	0 0270 8000	0 0271 FFFF	96K	Reserved
0272 0000	0272 7FFF	0 0272 0000	0 0272 7FFF	32K	EDMA channel controller (EDMA3CC) 1
0272 8000	0273 FFFF	0 0272 8000	0 0273 FFFF	96K	Reserved
02740000	0274 7FFF	0 02740000	0 0274 7FFF	32K	EDMA channel controller (EDMA3CC) 2
0274 8000	0275 FFFF	0 0274 8000	0 0275 FFFF	96K	Reserved
0276 0000	0276 03FF	0 0276 0000	0 0276 03FF	1K	EDMA EDMA3CC0 transfer controller (EDMA3TC) 0
0276 0400	0276 7FFF	0 0276 0400	0 0276 7FFF	31K	Reserved
0276 8000	0276 83FF	0 0276 8000	0 0276 83FF	1K	EDMA EDMA3CC0 transfer controller (EDMA3TC) 1
0276 8400	0276 FFFF	0 0276 8400	0 0276 FFFF	31K	Reserved

Table 2-2 Memory Map Summary (Part 6 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
0277 0000	0277 03FF	0 0277 0000	0 0277 03FF	1K	EDMA EDMA3CC1 transfer controller (EDMA3TC) 0
0277 0400	0277 7FFF	0 0277 0400	0 0277 7FFF	31K	Reserved
0277 8000	0277 83FF	0 0277 8000	0 0277 83FF	1K	EDMA EDMA3CC1 transfer controller (EDMA3TC) 1
0278 0400	0277 FFFF	0 0278 0400	0 0277 FFFF	31K	Reserved
0278 0000	0278 03FF	0 0278 0000	0 0278 03FF	1K	EDMA EDMA3CC1 Transfer controller (EDMA3TC) 2
0278 0400	0278 7FFF	0 0278 0400	0 0278 7FFF	31K	Reserved
0278 8000	0278 83FF	0 0278 8000	0 0278 83FF	1K	EDMA EDMA3CC1 transfer controller (EDMA3TC) 3
0278 8400	0278 FFFF	0 0278 8400	0 0278 FFFF	31K	Reserved
0279 0000	0279 03FF	0 0279 0000	0 0279 03FF	1K	EDMA EDMA3CC2 transfer controller (EDMA3TC) 0
0279 0400	0279 7FFF	0 0279 0400	0 0279 7FFF	31K	Reserved
0279 8000	0279 83FF	0 0279 8000	0 0279 83FF	1K	EDMA EDMA3CC2 transfer controller (EDMA3TC) 1
0279 8400	0279 FFFF	0 0279 8400	0 0279 FFFF	31K	Reserved
027A 0000	027A 03FF	0 027A 0000	0 027A 03FF	1K	EDMA EDMA3CC2 Transfer controller (EDMA3TC) 2
027A 0400	027A 7FFF	0 027A 0400	0 027A 7FFF	31K	Reserved
027A 8000	027A 83FF	0 027A 8000	0 027A 83FF	1K	EDMA EDMA3CC2 transfer controller (EDMA3TC) 3
027A 8400	027A FFFF	0 027A 8400	0 027A FFFF	31K	Reserved
027B 0000	027B FFFF	0 027B 0000	0 027B FFFF	64K	Reserved
027C 0000	027C FFFF	0 027C 0000	0 027C FFFF	64k	Reserved
027D 0000	027D 1000	0 027D 0000	0 027D 1000	4k	TI embedded trace buffer (TETB) - core 0
027D 1001	027D FFFF	0 027D 1001	0 027D FFFF	60k	Reserved
027E 0000	027E 1000	0 027E 0000	0 027E 1000	4k	TI embedded trace buffer (TETB) - core 1
027E 1001	027E FFFF	0 027E 1001	0 027E FFFF	60k	Reserved
027F 0000	027F 1000	0 027F 0000	0 027F 1000	4k	Reserved
027F 1001	027F FFFF	0 027F 1001	0 027F FFFF	60k	Reserved
0280 0000	0280 1000	0 0280 0000	0 0280 1000	4	Reserved
0280 1001	0280 FFFF	0 0280 1001	0 0280 FFFF	60k	Reserved
0281 0000	0281 3FFF	0 0281 0000	0 0281 3FFF	16k	Reserved
0281 4000	0281 FFFF	0 0281 4000	0 0281 FFFF	48k	Reserved
0282 0000	0282 3FFF	0 0282 0000	0 0282 3FFF	16k	Reserved
0282 4000	0282 FFFF	0 0282 4000	0 0282 FFFF	48k	Reserved
0283 0000	0283 3FFF	0 0283 0000	0 0283 3FFF	16k	Reserved
0283 4000	0283 FFFF	0 0283 4000	0 0283 FFFF	48k	Reserved
0284 0000	0284 3FFF	0 0284 0000	0 0284 3FFF	16k	Reserved
0284 4000	0284 FFFF	0 0284 4000	0 0284 FFFF	48k	Reserved
0285 0000	0285 7FFF	0 0285 0000	0 0285 7FFF	32k	TI embedded trace buffer (TETB) - system
0285 8000	0285 FFFF	0 0285 8000	0 0285 FFFF	32k	Reserved
0286 0000	028F FFFF	0 0286 0000	0 028F FFFF	640K	Reserved
0290 0000	0292 0FFF	0 0290 0000	0 0292 0FFF	132K	Serial RapidIO configuration
0292 1000	029F FFFF	0 0292 1000	0 029F FFFF	1M-132K	Reserved
02A0 0000	02AF FFFF	0 02A0 0000	0 02AF FFFF	1M	Queue Manager subsystem configuration
02B0 0000	02BF FFFF	0 02B0 0000	0 02BF FFFF	1M	Reserved
02C0 0000	02FF FFFF	0 02C0 0000	0 02FF FFFF	4M	Reserved
03000 000	07FF FFFF	0 03000 000	0 07FF FFFF	80M	Reserved

Table 2-2 Memory Map Summary (Part 7 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	**End**	**Start**	**End**	**Bytes**	**Description**
0800 0000	0800 FFFF	0 0800 0000	0 0800 FFFF	64k	Extended Memory Controller (XMC) configuration
0801 0000	0BBF FFFF	0 0801 0000	0 0BBF FFFF	60M-64k	Reserved
0BC0 0000	0BCF FFFF	0 0BC0 0000	0 0BCF FFFF	1M	Multicore Shared Memory Controller (MSMC) configuration
0BD0 0000	0BFF FFFF	0 0BD0 0000	0 0BFF FFFF	3M	Reserved
0C00 0000	0C1F FFFF	0 0C00 0000	0 0C1F FFFF	2M	Multicore Shared Memory (MSM)
0C20 0000	0C3F FFFF	0 0C20 0000	0 0C3F FFFF	2M	Reserved
0C40 0000	0FFF FFFF	0 0C40 0000	0 0FFF FFFF	60 M	Reserved
1000 0000	107F FFFF	0 1000 0000	0 107F FFFF	8M	Reserved
1080 0000	108F FFFF	0 1080 0000	0 108F FFFF	1M	CorePac0 L2 SRAM
1090 0000	10DF FFFF	0 1090 0000	0 10DF FFFF	5M	Reserved
10E0 0000	10E0 7FFF	0 10E0 0000	0 10E0 7FFF	32k	CorePac0 L1P SRAM
10E0 8000	10EF FFFF	0 10E0 8000	0 10EF FFFF	1M-32K	Reserved
10F0 0000	10F0 7FFF	0 10F0 0000	0 10F0 7FFF	32k	CorePac0 L1D SRAM
10F0 8000	117F FFFF	0 10F0 8000	0 117F FFFF	9M-32k	Reserved
1180 0000	118F FFFF	0 1180 0000	0 118F FFFF	1M	CorePac1 L2 SRAM
1190 0000	11DF FFFF	0 1190 0000	0 11DF FFFF	5M	Reserved
11E0 0000	11E0 7FFF	0 11E0 0000	0 11E0 7FFF	32k	CorePac1 L1P SRAM
11E0 8000	11EF FFFF	0 11E0 8000	0 11EF FFFF	1M-32K	Reserved
11F0 0000	11F0 7FFF	0 11F0 0000	0 11F0 7FFF	32k	CorePac1 L1D SRAM
11F0 8000	127F FFFF	0 11F0 8000	0 127F FFFF	9M-32k	Reserved
1280 0000	128F FFFF	0 1280 0000	0 128F FFFF	1M	Reserved
1290 0000	12DF FFFF	0 1290 0000	0 12DF FFFF	5M	Reserved
12E0 0000	12E0 7FFF	0 12E0 0000	0 12E0 7FFF	32k	Reserved
12E0 8000	12EF FFFF	0 12E0 8000	0 12EF FFFF	1M-32K	Reserved
12F0 0000	12F0 7FFF	0 12F0 0000	0 12F0 7FFF	32k	Reserved
12F0 8000	137F FFFF	0 12F0 8000	0 137F FFFF	9M-32k	Reserved
1380 0000	1388 FFFF	0 1380 0000	0 1388 FFFF	1M	Reserved
1390 0000	13DF FFFF	0 1390 0000	0 13DF FFFF	5M	Reserved
13E0 0000	13E0 7FFF	0 13E0 0000	0 13E0 7FFF	32k	Reserved
13E0 8000	13EF FFFF	0 13E0 8000	0 13EF FFFF	1M-32K	Reserved
13F0 0000	13F0 7FFF	0 13F0 0000	0 13F0 7FFF	32k	Reserved
13F0 8000	147F FFFF	0 13F0 8000	0 147F FFFF	9M-32k	Reserved
1480 0000	1487 FFFF	0 1480 0000	0 1487 FFFF	512K	Reserved
1488 0000	148F FFFF	0 1488 0000	0 148F FFFF	512K	Reserved
1490 0000	14DF FFFF	0 1490 0000	0 14DF FFFF	5M	Reserved
14E0 0000	14E0 7FFF	0 14E0 0000	0 14E0 7FFF	32k	Reserved
14E0 8000	14EF FFFF	0 14E0 8000	0 14EF FFFF	1M-32K	Reserved
14F0 0000	14F0 7FFF	0 14F0 0000	0 14F0 7FFF	32k	Reserved
14F0 8000	157F FFFF	0 14F0 8000	0 157F FFFF	9M-32k	Reserved
1580 0000	1587 FFFF	0 1580 0000	0 1587 FFFF	512K	Reserved
1588 0000	158F FFFF	0 1588 0000	0 158F FFFF	512K	Reserved
1590 0000	15DF FFFF	0 1590 0000	0 15DF FFFF	5M	Reserved
15E0 0000	15E0 7FFF	0 15E0 0000	0 15E0 7FFF	32k	Reserved

Table 2-2 Memory Map Summary (Part 8 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	End	Start	End	Bytes	Description
15E0 8000	15EF FFFF	0 15E0 8000	0 15EF FFFF	1M-32K	Reserved
15F0 0000	15F0 7FFF	0 15F0 0000	0 15F0 7FFF	32k	Reserved
15F0 8000	167F FFFF	0 15F0 8000	0 167F FFFF	9M-32k	Reserved
1680 0000	1687 FFFF	0 1680 0000	0 1687 FFFF	512K	Reserved
1688 0000	168F FFFF	0 1688 0000	0 168F FFFF	512K	Reserved
1690 0000	16DF FFFF	0 1690 0000	0 16DF FFFF	5M	Reserved
16E0 0000	16E0 7FFF	0 16E0 0000	0 16E0 7FFF	32k	Reserved
16E0 8000	16EF FFFF	0 16E0 8000	0 16EF FFFF	1M-32K	Reserved
16F0 0000	16F0 7FFF	0 16F0 0000	0 16F0 7FFF	32k	Reserved
16F0 8000	177F FFFF	0 16F0 8000	0 177F FFFF	9M-32k	Reserved
1780 0000	1787 FFFF	0 1780 0000	0 1787 FFFF	512K	Reserved
1788 0000	178F FFFF	0 1788 0000	0 178F FFFF	512K	Reserved
1790 0000	17DF FFFF	0 1790 0000	0 17DF FFFF	5M	Reserved
17E0 0000	17E0 7FFF	0 17E0 0000	0 17E0 7FFF	32k	Reserved
17E0 8000	17EF FFFF	0 17E0 8000	0 17EF FFFF	1M-32K	Reserved
17F0 0000	17F0 7FFF	0 17F0 0000	0 17F0 7FFF	32k	Reserved
17F0 8000	1FFF FFFF	0 17F0 8000	0 1FFF FFFF	129M-32k	Reserved
2000 0000	200F FFFF	0 2000 0000	0 200F FFFF	1M	System trace module (STM) configuration
2010 0000	201F FFFF	0 2010 0000	0 201F FFFF	1M	Reserved
2020 0000	205F FFFF	0 2020 0000	0 205F FFFF	4M	Reserved
2060 0000	206F FFFF	0 2060 0000	0 206F FFFF	1M	TCP3d-B Data
2070 0000	207F FFFF	0 2070 0000	0 207F FFFF	1M	Reserved
2080 0000	208F FFFF	0 2080 0000	0 208F FFFF	1M	TCP3d-A data
2090 0000	2090 3FFF	0 2090 0000	0 2090 3FFF	16K	Reserved
2090 4000	209F FFFF	0 2090 4000	0 209F FFFF	1M-16K	Reserved
20A0 0000	20A3 FFFF	0 20A0 0000	0 20A3 FFFF	256K	Reserved
20A4 0000	20A4 FFFF	0 20A4 0000	0 20A4 FFFF	64K	Reserved
20A5 0000	20AF FFFF	0 20A5 0000	0 20AF FFFF	704k	Reserved
20B0 0000	20B1 FFFF	0 20B0 0000	0 20B1 FFFF	128k	Boot ROM
20B2 0000	20BE FFFF	0 20B2 0000	0 20BE FFFF	832k	Reserved
20BF 0000	20BF 01FF	0 20BF 0000	0 20BF 01FF	512	SPI
20BF 0400	20BF FFFF	0 20BF 0400	0 20BF FFFF	63k	Reserved
20C0 0000	20C0 00FF	0 20C0 0000	0 20C0 00FF	256	EMIF16 configuration
20C0 0100	20FF FFFF	0 20C0 0100	0 20FF FFFF	4M-256	Reserved
2100 0000	2100 01FF	1 0000 0000	1 0000 01FF	512	DDR3 EMIF configuration
2100 0100	213F FFFF	0 2100 0100	0 213F FFFF	4M-256	Reserved
2140 0000	2140 00FF	0 2140 0000	0 2140 00FF	256	HyperLink configuration
2140 0400	217F FFFF	0 2140 0400	0 217F FFFF	4M-1K	Reserved
2180 0000	2180 7FFF	0 2180 0000	0 2180 7FFF	32K	PCIe configuration
2180 8000	21BF FFFF	0 2180 8000	0 21BF FFFF	4M-32K	Reserved
21C0 0000	21FF FFFF	0 21C0 0000	0 21FF FFFF	4M	Reserved
2200 0000	229F FFFF	0 2200 0000	0 229F FFFF	10M	Reserved
22A0 0000	22A0 FFFF	0 22A0 0000	0 22A0 FFFF	64K	VCP2_A

Table 2-2 Memory Map Summary (Part 9 of 9)

Logical 32 bit Address		Physical 36 bit Address			
Start	**End**	**Start**	**End**	**Bytes**	**Description**
22A1 0000	22AF FFFF	0 22A1 0000	0 22AF FFFF	1M-64K	Reserved
22B0 0000	22B0 FFFF	0 22B0 0000	0 22B0 FFFF	64K	VCP2_B
22B1 0000	22BF FFFF	0 22B1 0000	0 22BF FFFF	1M-64K	Reserved
22C0 0000	22C0 FFFF	0 22C0 0000	0 22C0 FFFF	64K	Reserved
22C1 0000	22CF FFFF	0 22C1 0000	0 22CF FFFF	1M-64K	Reserved
22D0 0000	22D0 FFFF	0 22D0 0000	0 22D0 FFFF	64K	Reserved
22D1 0000	22DF FFFF	0 22D1 0000	0 22DF FFFF	1M-64K	Reserved
22E0 0000	23FF FFFF	0 22E0 0000	0 23FF FFFF	18M	Reserved
2400 0000	2FFF FFFF	0 2400 0000	0 2FFF FFFF	192M	Reserved
3000 0000	331F FFFF	0 3000 0000	0 331F FFFF	50M	Reserved
3320 0000	335F FFFF	0 3320 0000	0 335F FFFF	4M	RAC data
3360 0000	33FF FFFF	0 3360 0000	0 33FF FFFF	10M	Reserved
3400 0000	341F FFFF	0 3400 0000	0 341F FFFF	2M	Queue Manager subsystem data
3420 0000	342F FFFF	0 3420 0000	0 342F FFFF	1M	Reserved
3430 0000	3439 FFFF	0 3430 0000	0 3439 FFFF	640K	Reserved
343A 0000	343F FFFF	0 343A 0000	0 343F FFFF	384K	Reserved
3440 0000	347F FFFF	0 3440 0000	0 347F FFFF	4M	Reserved
3480 0000	34BF FFFF	0 3480 0000	0 34BF FFFF	4M	Reserved
34C0 0000	34C2 FFFF	0 34C0 0000	0 34C2 FFFF	192K	TAC data
34C3 0000	34FF FFFF	0 34C3 0000	0 34FF FFFF	4M-192K	Reserved
3500 0000	351F FFFF	0 3500 0000	0 351F FFFF	2M	Reserved
3520 0000	3521 FFFF	0 3520 0000	0 3521 FFFF	128K	BCP configuration
3522 0000	35FF FFFF	0 3522 0000	0 35FF FFFF	14M-128k	Reserved
3600 0000	3FFF FFFF	0 3600 0000	0 3FFF FFFF	160M	Reserved
4000 0000	4FFF FFFF	0 4000 0000	0 4FFF FFFF	256M	HyperLink data
5000 0000	5FFF FFFF	0 5000 0000	0 5FFF FFFF	256M	Reserved
6000 0000	6FFF FFFF	0 6000 0000	0 6FFF FFFF	256M	PCIe data
7000 0000	73FF FFFF	0 7000 0000	0 73FF FFFF	64M	EMIF16 CE0 data memory, supports NAND, NOR, or SRAM memory (1)
7400 0000	77FF FFFF	0 7400 0000	0 77FF FFFF	64M	EMIF16 CE1 data memory, supports NAND, NOR, or SRAM memory
7800 0000	7BFF FFFF	0 7800 0000	0 7BFF FFFF	64M	EMIF16 CE2 data memory, supports NAND, NOR, or SRAM memory
7C00 0000	7FFF FFFF	0 7C00 0000	0 7FFF FFFF	64M	EMIF16 CE3 data memory, supports NAND, NOR, or SRAM memory
8000 0000	FFFF FFFF	8 0000 0000	8 7FFF FFFF	2G	DDR3 EMIF data (2)
End of Table 2-2					

1 32MB per chip select for 16-bit NOR and SRAM. 16MB per chip select for 8-bit NOR and SRAM. More than 32MB allowed by NAND flash.

2 The memory map shows only the default MPAX configuration of DDR3 memory space. For the extended DDR3 memory space access (up to 8GB), please refer to the MPAX configuration details in *C66x CorePac User Guide* and *Multicore Shared Memory Controller (MSMC) for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 75.

2.3 Boot Sequence

The boot sequence is a process by which the SoC's internal memory is loaded with program and data sections. The SoC's internal registers are programmed with predetermined values. The boot sequence is started automatically after each power-on reset. A hard reset, soft reset or local reset to an individual C66x CorePac should not affect the state of the hardware boot controller on the device. For more details on the initiators of the resets, see section 8.5 "Reset Controller" on page 140.

For nonsecure devices, there are two types of booting: CorePac as the boot master and the ARM as the boot master. For secure devices, the CorePac is always the secure master and the CorePac or ARM can be the boot master. There are also two boot ROMs in the TCI6612. One is for the chip-level and one is dedicated to ARM boot. Both the CorePacs and the ARM need to read the bootmode register to determine how to proceed with the boot.

The TCI6612 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[13:0] device configuration inputs to determine the software configuration that must be completed. The BOOTMODE field in DEVSTAT register (bits [14:1]) latches the BOOTMODE[13:0] device configuration pins at power-on reset. For more details on Boot Sequence, see the *Bootloader for the C66x DSP User Guide* in "Related Documentation from Texas Instruments" on page 75.

2.4 Boot Modes Supported and PLL Settings

The device supports several boot processes, which leverage the internal boot ROM. From a hardware perspective, there are four possible boot modes:

- **Public ROM Boot when the CorePac 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 C66x CorePac. Both the CorePac and the ARM must read the BOOTMODE field in DEVSTAT register to determine which is the boot master.

  After the BootROM of the ARM reads the BOOTMODE field to determine that the CorePac is the boot master, the ARM stays idle by executing WFI instruction and waiting for the CorePac's interrupt. The chip BootROM reads the BOOTMODE field to determine that the CorePac is the boot master, then the CorePac performs the boot process and CorePac1 executes an IDLE instruction. The bootROM writes to the ARM_boot_ADDR register. After the boot process is completed, the CorePac performing boot interrupts CorePac1 and the ARM through the IPC register. Then CorePac1 and the ARM complete boot management operations and begin executing from a predefined location in memory.
- **Public ROM Boot when the ARM is the boot master** — The only difference from the public ROM Boot when the CorePac is the boot master is that the ARM performs the boot process while the CorePac executes idle instructions. When the ARM finishes the boot process, it sends interrupts to the CorePac through IPC registers, and the CorePac complete the boot management operations and begin executing from the predefined locations.
- **Secure ROM Boot when the CorePac is the boot master** — On secure devices, the C66x CorePac is always the secure master. The C66x CorePac and the ARM are released from reset simultaneously and the CorePac begins executing from secure ROM. Software in the secure ROM will free up internal RAM pages, after which the C66x CorePac initiates the boot process. The C66x CorePac performs any authentication and decryption required on the bootloaded image for the CorePac and for the ARM prior to beginning execution.
- **Secure ROM Boot when the ARM is the boot master** — On secure devices, the C66x CorePac is always the secure master. The C66x CorePac and the ARM are released from reset simultaneously and the CorePac begins executing from secure ROM. Software in the secure ROM will free up internal RAM pages, after which the C66x CorePac initiates the boot process. The C66x CorePac performs any authentication and decryption required on the bootloaded image and for the ARM prior to beginning execution.

The boot process performed by the C66x CorePac and the ARM in public ROM boot and secure ROM boot are determined by the BOOTMODE field in the DEVSTAT register. The C66x CorePac and the ARM read this value, and then execute the associated boot process in software. "Master" bit determines whether the boot is CorePac boot or ARM boot. Figure 2-2 shows the bits associated with BOOTMODE field in the DEVSTAT register when the CorePac is the boot master. Figure 2-11 shows the bits associated with BOOTMODE field in the DEVSTAT register when the ARM is the boot master. The PLL settings are shown at the end of this section, and the PLL set-up details can be found in Section 8.6 "Main PLL and the PLL Controller" on page 146.

2.5 Pin Decoding with CorePac as Boot Master

When the "Master" bit of DEVSTAT register is set to 0, the CorePac initiates boot.

Figure 2-2 DEVSTAT - Boot Mode Pin Decoding with CorePac as Boot Master

DEVSTAT - Boot Mode Pins ROM Mapping														
14	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**Mode**
Master (0)	X	X	X	Wait Enable	MEM Width	Chip Select Region		Sub-Mode	X	X	0	0	0	No Boot/EMIF16 (NOR Boot)
	X	X	X	Lane Setup	Data Rate		Ref Clock		X	X	0	0	1	Serial Rapid I/O
	X	X	X	SerDes Clock Mult		Ext Connection		Dev ID			0	1	0	Ethernet (SGMII)
											0	1	1	
	X	X	X	Ref Clk	BAR Config				X	X	1	0	0	PCIe
	Mode	Address		Speed	Parameter Index						1	0	1	I^2C Master Mode
	X	X	X	Mode	Receive I^2C Address				X	X	1	0	1	I^2C Slave Mode
	Mode		4, 5 Pin	Addr Width	Chip Select		Parameter Index		X	X	1	1	0	SPI
	X	X	X	X	Data Rate		Ref Clock		X	X	1	1	1	HyperLink

2.5.1 Boot Device Field

The Boot Device field (DEVSTAT[3:1]) defines the boot device that is chosen. Table 2-3 shows the supported boot modes.

Table 2-3 Boot Device Field

Bit	Field	Description
3-1	Boot Device	Selects boot mode for the device. 0 = No boot / EMIF16 (NOR boot) 1 = Serial Rapid I/O 2 = Ethernet (SGMII) (PASS PLL configuration assumes input rate same as SYSCLK(P|N); BOOTMODE[12:10] values drive the PASS PLL configuration during boot) 3 = Ethernet (SGMII) (PASS PLL configuration assumes input rate same as SRIOSGMIICLK(P|N); BOOTMODE[9:8] values drive the PASS PLL configuration during boot) 4 = PCIe 5 = I^2C 6 = SPI 7 = HyperLink
End of Table 2-3		

2.5.2 Device Configuration Field

The device configuration field (DEVSTAT[10:4]) is used to configure the boot peripheral and, therefore, the bit definitions depend on the boot mode.

2.5.2.1 No Boot / EMIF16 (NOR Boot) Device Configuration

Figure 2-3 No Boot / EMIF16 (NOR Boot) Configuration Fields

10	9	8	7	6	5	4
Wait Enable	MEM width	Chip Select Region		Sub-Mode	Reserved	

Table 2-4 No Boot / EMIF16 (NOR Boot) Configuration Field Descriptions

Bit	Field	Description
10	Wait Enable	Enables extended wait mode 0 = Wait enable disabled 1 = Wait enable enabled
9	MEM Width	Selects the EMIF16 data bus width for NOR flash 0 = 8-bit width 1 = 16-bit width
8-7	Chip Select Region	Sets the chip select region 0 = Chip selection region CE0 1 = Chip selection region CE1 2 = Chip selection region CE2 3 = Reserved
6	Sub-Mode	Selects either boot through EMIF16 interface or no boot 0 = No Boot 1 = EMIF16 Boot
5-4	Reserved	Reserved
End of Table 2-4		

2.5.2.2 Serial Rapid I/O Boot Device Configuration

The device ID is always set to 0xff (8-bit node IDs) or 0xffff (16 bit node IDs) at power-on reset.

Figure 2-4 Serial Rapid I/O Device Configuration Fields

10	9	8	7	6	5	4
Lane Setup	Data Rate		Ref Clock		Reserved	

Table 2-5 Serial Rapid I/O Configuration Field Descriptions

Bit	Field	Description
10	Lane Setup	Selects the port width configuration 0 = Port configured as 4 ports each 1 lane wide (4 -1× ports) 1 = Port configured as 2 ports 2 lanes wide (2 – 2× ports)
9-8	Data Rate	Selects the line rate for each lane 0 = 1.25 GBaud 1 = 2.5 GBaud 2 = 3.125 GBaud 3 = 5.0 GBaud
7-6	Ref Clock	Selects the reference clock input to the SERDES clock multiplier 0 = 156.25 MHz 1 = 250 MHz 2 = 312.5 MHz
5-4	Reserved	Reserved
End of Table 2-5		

In SRIO boot mode, both the message mode and DirectIO mode will be enabled by default. If use of the memory reserved for received messages is required and reception of messages cannot be prevented, the master can disable the message mode by writing to the boot table and generating a boot restart.

2.5.2.3 Ethernet (SGMII) Boot Device Configuration

Figure 2-5 Ethernet (SGMII) Device Configuration Fields

10	9	8	7	6	5	4
SerDes Clock Mult		Ext connection		Dev ID		

Table 2-6 Ethernet (SGMII) Configuration Field Descriptions

Bit	Field	Description
10-9	SerDes clock mult	Selects the SERDES clock multiplier for the input reference clock The output frequency of the PLL must be 1.25 GBaud 0 = ×8 for input clock of 156.25 MHz 1 = ×5 for input clock of 250 MHz 2 = ×4 for input clock of 312.5 MHz 3 = Reserved
8-7	Ext connection	Selects the external connection type 0 = Mac to Mac connection, master with auto negotiation 1 = Mac to Mac connection, slave, and Mac to Phy 2 = Mac to Mac, forced link 3 = Mac to fiber connection
6-4	Device ID	This value is used in the device ID field of the Ethernet-ready frame and can range from 0 to 7.
End of Table 2-6		

2.5.2.4 PCIe Boot Device Configuration

Extra device configuration is provided in the PCIe bits in the DEVSTAT register.

Figure 2-6 PCIe Device Configuration Fields

10	9	8	7	6	5	4
Ref Clk	BAR Config				Reserved	

Table 2-7 PCIe Device Configuration Field Descriptions

Bit	Field	Description
10	Ref Clk	Selects the reference clock input to the SERDES multiplier. 0 = 100 MHz reference clock 1 = 250 Mhz reference clock
9-6	BAR Config	BAR configuration options See Table 2-8. This value can range from 0 to 0xf
5-4	Reserved	Reserved
End of Table 2-7		

Table 2-8 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
End of Table 2-8								

2.5.2.5 I²C Boot Device Configuration

2.5.2.5.1 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 will make the initial read of the I²C EEPROM while the PLL is in bypass mode. The initial read will contain the desired clock multiplier, which will be set up prior to any subsequent reads.

Figure 2-7 I²C Master Mode Device Configuration Fields

13	12	11	10	9	8	7	6	5	4
Mode	Address		Speed	Parameter Index					

Table 2-9 I²C Master Mode Device Configuration Field Descriptions

Bit	Field	Description
13	Mode	Mode select 0 = Master mode 1 = Passive mode (see "I2C Passive Mode" on page 35)
12-11	Address	Selects which address to boot from EEPROM 0 = Boot from I²C EEPROM at I²C bus address 0x50 1 = Boot from I²C EEPROM at I²C bus address 0x51 2 = Boot from I²C EEPROM at I²C bus address 0x52 3 = Boot from I²C EEPROM at I²C bus address 0x53
10	Speed	Selects I²C bus frequency 0 = I²C slow mode. Initial data rate is SYSCLK / 5000 until PLLs and clocks are programmed 1 = I²C fast mode. Initial data rate is SYSCLK / 250 until PLLs and clocks are programmed
9-4	Parameter Index	Identifies the index of the configuration table initially read from the I²C EEPROM. This value can range from 0 to 63.
End of Table 2-9		

2.5.2.5.2 I²C Passive Mode

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

Figure 2-8 I²C Passive Mode Device Configuration Fields

10	9	8	7	6	5	4
Mode (1)	Receive I²C Address				Reserved	

Table 2-10 I²C Passive Mode Device Configuration Field Descriptions

Bit	Field	Description
10	Mode	Mode select 0 = Master mode (see "I2C Master Mode" on page 35) 1 = Passive mode
9-6	Receive I²C Address	The I²C Bus address the device will listen to for data. This value can range from 0 to 15.
5-4	Reserved	Reserved
End of Table 2-10		

2.5.2.6 SPI Boot Device Configuration

In SPI boot mode, the SPI device configuration uses ten bits of device configuration instead of seven as used in other boot modes.

Figure 2-9 SPI Device Configuration Fields

13	12	11	10	9	8	7	6	5	4
Mode		4, 5 Pin	Addr Width	Chip Select		Parameter Table Index		Reserved	

Table 2-11 SPI Device Configuration Field Descriptions

Bit	Field	Description
13-12	Mode	Clk Pol / 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. 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.
11	4, 5 Pin	Selects the operational mode 0 = 4-pin mode used 1 = 5-pin mode used
10	Addr Width	Selects the SPI address width 0 = 16-bit address values are used 1 = 24-bit address values are used
9-8	Chip Select	Chip select field value This value can range from 0 to 3.
7-6	Parameter Table Index	Specifies which parameter table is loaded This value can range from 0 to 3
5-4	Reserved	Reserved
End of Table 2-11		

2.5.2.7 HyperLink Boot Device Configuration

Figure 2-10 HyperLink Boot Device Configuration Fields

10	9	8	7	6	5	4
Reserved	Data Rate		Ref Clock		Reserved	

Table 2-12 HyperLink Boot Device Configuration Field Descriptions

Bit	Field	Description
10	Reserved	Reserved
9-8	Data Rate	Selects the line rate for each link 0 = 1.25 GBaud 1 = 3.125 GBaud 2 = 6.25 GBaud 3 = Reserved
7-6	Ref Clocks	Selects the reference clock input 0 = 156.25 MHz 1 = 250 MHz 2 = 312.5 MHz
5-4	Reserved	Reserved
End of Table 2-12		

2.6 Pin Decoding with ARM as Boot Master

When the *Master* bit of DEVSTAT is set to 1, the ARM initiates boot.

Figure 2-11 DEVSTAT - Boot Mode Pin Decoding with ARM as Boot Master

DEVSTAT - Boot Mode Pins ROM Mapping													
14	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Master (1)	PLL Configuration			Boot Mode Sequence				Boot Mode Configuration				Reserved	

2.6.1 Boot Mode Sequence Field

With ARM as the boot master, two boot modes are chosen by the boot mode sequence. If a failure on the primary boot mode is detected, then the secondary boot mode is attempted.

Table 2-13 ARM Boot Mode Sequence

DEVSTAT Bits				Boot Mode Sequence	
10	**9**	**8**	**7**	**1st**	**2nd**
0	0	0	0	No Boot	N/A
0	0	0	1	UART	EMIF16 (NOR)
0	0	1	0	UART	EMIF16 (NOR)/wait
0	0	1	1	UART	NAND (1)
0	1	0	0	Ethernet	EMIF16 (NOR)
0	1	0	1	Ethernet	EMIF16 (NOR)/wait
0	1	1	0	Ethernet	NAND
0	1	1	1	PCIe	EMIF16 (NOR)
1	0	0	0	PCIe	EMIF16 (NOR)/wait
1	0	0	1	PCIe	NAND
1	0	1	0	SPI	EMIF16 (NOR)
1	0	1	1	SPI	EMIF16 (NOR)/wait
1	1	0	0	SPI	NAND
1	1	0	1	EMIF16 (NOR)	NAND
1	1	1	0	NAND/I^2C	EMIF16 (NOR)
1	1	1	1	NAND	EMIF16 (NOR)
End of Table 2-13					

1 DEVSTAT values are used during NAND boot. NAND boot setup parameters are determined by the boot ROM. For information on NAND boot setup parameters, please refer to the Bootloader for the C66x DSP User Guide in Related Documentation from Texas Instruments.

2.6.2 Boot Mode Configuration Field

2.6.2.1 UART Boot Mode Configuration

Figure 2-12 UART Boot Mode Configuration Fields

6	5	4	3
Reserved			Port

Table 2-14 UART Boot Mode Configuration Field Descriptions

Bit	Field	Description
6-4	Reserved	Reserved
3	Port	Selects UART port number 0 = Boot from UART port 0 1 = Boot from UART port 1
End of Table 2-14		

2.6.2.2 Ethernet Boot Mode Configuration

Figure 2-13 Ethernet Boot Mode Configuration Fields

6	5	4	3
SERDES Reference Clock		SGMII Connection	

Table 2-15 Ethernet Boot Mode Configuration Field Descriptions

Bit	Field	Description
6-5	SERDES Reference Clock	Selects reference clock input to SERDES multiplier 0 = 56.25 MHz reference clock 1 = 250.0 MHz reference clock 2 = 312.5 MHz reference clock 3 = Reserved
4-3	SGMII Connection	Selects connection type 0 = Mac to Mac connection, ARM master with auto-negotiation 1 = Mac to Mac connection, ARM slave with auto-negotiation 2 = Mac to Mac forced link, 1-GHz full duplex 3 = Mac to fiber
End of Table 2-15		

2.6.2.3 PCIe Boot Mode Configuration

Figure 2-14 PCIe Boot Mode Configuration Fields

6	5	4	3
BAR Configuration			

The BAR Configuration field matches the description shown in Table 2-8.

2.6.2.4 SPI Boot Mode Configuration

Figure 2-15 SPI Boot Mode Configuration Fields

6	5	4	3
SPI Mode		Address Width	Chip Select

Table 2-16 SPI Boot Mode Configuration Field Descriptions

Bit	Field	Description
6-5	SPI Mode	Clock Pol / 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 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.
4	Address Width	Selects SPI address width 0 = 16 bit data address 1 = 24 bit data address
3	Chip Select	Selects from chip select 0 or 1 0 = Chip select 0 active 1 = Chip select 1 active
End of Table 2-16		

2.6.2.5 EMIF16 (NOR) Boot Mode Configuration

Figure 2-16 EMIF16 (NOR) Boot Mode Configuration Fields

6	5	4	3
Wait Enable	Mem Width	Chip Select Region	

Table 2-17 EMIF16 (NOR) Boot Mode Configuration Field Descriptions

Bit	Field	Description
6	Wait Enable	Enables extended wait mode 0 = No wait enable 1 = Wait enable
5	Mem Width	Selects the EMIF16 data bus width for NOR flash 0 = 8-bit width 1 = 16-bit width
4-3	Chip Select Region	Chip Select region 0 = Chip select region CE0 1 = Chip select region CE1 2 = Chip select region CE2 3 = Chip select region CE3
End of Table 2-17		

2.7 PLL Settings

The PLL default settings are determined by the BOOTMODE[12:10] bits. Table 2-18 shows settings for various input clock frequencies. This will set the PLL to the maximum clock setting for the device.

OUTPUT_DIVIDE is the value of the field of SECCTL[22:19]. This will set the PLL to the maximum clock setting for the device (with OUTPUT_DIVIDE=2, by default).

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

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 2-3 for details on configuring Ethernet boot mode. The output from the PASS PLL goes through an on-chip divider to reduce the operating frequency before reaching the NETCP. The PASS PLL generates 1050 MHz, and after the chip divider (/3), feeds 350 MHz to the NETCP.

The Main PLL is controlled using a PLL Controller and a chip-level MMR. The DDR3 PLL and PASS PLL are controlled by chip-level MMRs. For details on how to setup the PLL see Section 8.6 "Main PLL and the PLL Controller" on page 146. For details on the operation of the PLL Controller module, see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

Table 2-18 C66x CorePac System PLL Configuration

BOOTMODE [12:10]	Input Clock Freq (MHz)	800 MHz Device			1000 MHz Device			1200 MHz Device			PA = 350 MHz (1)		
		PLLD	PLLM	SoC ƒ	PLLD	PLLM	SoC ƒ	PLLD	PLLM	SoC ƒ	PLLD	PLLM	SoC ƒ (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	24	255	800	4	63	1000	24	383	1200	24	335	1050
0b101	250.00	4	31	800	0	7	1000	4	47	1200	4	41	1050
0b110	312.50	24	127	800	4	31	1000	24	191	1200	24	167	1050
0b111	122.88	47	624	800	28	471	999.989	31	624	1200	11	204	1049.6
End of Table 2-18													

1 The PASS PLL generates 1050 MHz and is internally divided by 3 to feed 350 MHz to the Packet Accelerator.
2 ƒ represents frequency in MHz.

2.8 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 as well as the definition of a completely customized boot.

2.9 SoC Security

The TMS320TCI6612 contains security architecture that allows both the CorePac and ARM CorePac to perform secure accesses within the device. This security architecture is designed to provide the following:

- Customer software authentication and protection through secure boot and runtime security
- Network interface security and network security protocol support

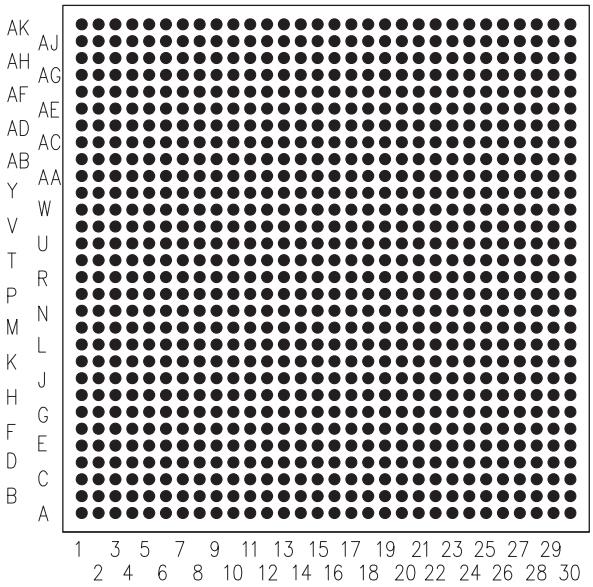
For more information, see the *Security Addendum for KeyStone I Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75

2.10 Terminals

2.10.1 Package Terminals

Figure 2-17 shows the TMS320TCI6612 CMS plastic ball grid array package (bottom view).

Figure 2-17 CMS 900-Pin Plastic BGA Package Bottom View



2.10.2 Pin Map

The following figures show the TMS320TCI6612 pin assignments in four quadrants (A, B, C, and D).

Figure 2-18 Pin Map Quadrants (Bottom View)

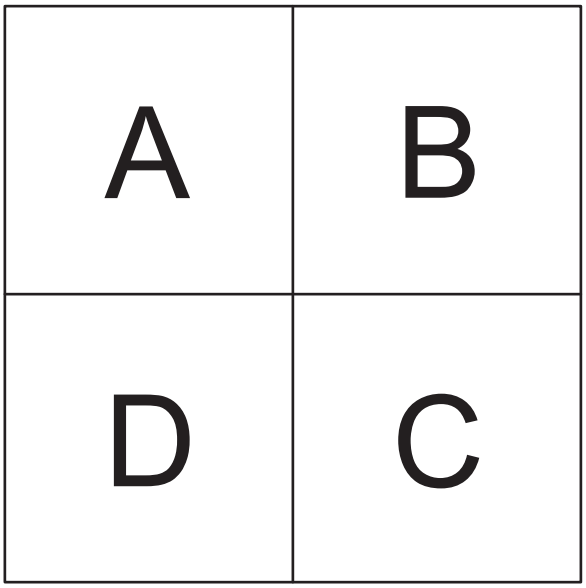


Figure 2-19 Upper Left Quadrant TMS320TCI6612 — Bottom View

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AK	VSS	RIORXN2	RIORXP2	VSS	RIORXP0	RIORXN0	VSS	PCIERXP0	PCIERXN0	VSS	SRIOSGMII CLKN	SRIOSGMII CLKP	PCIECLKP	PASSCLKN	SPISCS2
AJ	RIORXN3	RIORXP3	VSS	RIORXP1	RIORXN1	VSS	RIOTXP0	RIOTXN0	VSS	PCIERXN1	PCIERXP1	VSS	PCIECLKN	PASSCLKP	SPISCS3
AH	VSS	SGMII1RXN	SGMII1RXP	VSS	RIOTXN2	RIOTXP2	VSS	PCIETXP1	PCIETXN1	VSS	RSV25	RSV24	USIMCLK	SPICLK	SPISCS4
AG	SGMII0RXN	SGMII0RXP	VSS	RIOTXN3	RIOTXP3	VSS	RIOTXN1	RIOTXP1	VSS	PCIETXP0	PCIETXN0	VSS	USIMRST	SPIDIN	SPISCS0
AF	VSS	SGMII1TXN	SGMII1TXP	VSS	VSS	VDDT2	VSS	VDDT2	VDDR_2	VDDT2	VSS	MDCLK	USIMIO	SPIDOUT	SPISCS1
AE	RSV15	VSS	SGMII0TXN	SGMII0TXP	VDDR_4	VSS	VDDT2	VSS	VDDT2	VSS	VDDT2	VSS	EXTFRAME EVENT	VSS	DVDD18
AD	EMIFD11	EMIFD13	VSS	RSV17	RSV16	RSV22	RSV10	VDDT2	VSS	VDDT2	VSS	MDIO	VSS	AVDDA3	VSS
AC	VSS	EMIFD10	EMIFD14	EMIFD15	VDDR_3	VSS	RSV11	VSS	CVDD	VSS	VDDT2	VSS	DVDD18	VSS	DVDD18
AB	EMIFD06	EMIFD08	EMIFD09	EMIFD12	VSS	DVDD18	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
AA	EMIFD07	VSS	EMIFD04	EMIFD01	EMIFA22	VSS	DVDD18	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
Y	EMIFD05	EMIFD02	VSS	EMIFA18	EMIFA10	DVDD18	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
W	VSS	EMIFD03	EMIFA20	EMIFA09	EMIFA01	VSS	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
V	EMIFD00	EMIFA21	EMIFA17	VSS	$\overline{\text{EMIFCE3}}$	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
U	EMIFA23	VSS	EMIFA16	EMIFA06	EMIFRNW	VSS	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
T	EMIFA19	EMIFA15	VSS	EMIFA05	$\overline{\text{EMIFWE}}$	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS

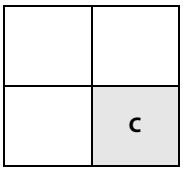
A	

Figure 2-20 Upper Right Quadrant TMS320TCI6612 — Bottom View

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
UART1CTS	UART0RTS	GPIO06	GPIO04	VSS	GPIO14	GPIO11	VSS	EMU17	EMU16	EMU15	EMU13	EMU10	EMU07	VSS	AK
UART1RTS	UART0CTS	GPIO07	GPIO05	GPIO00	GPIO12	GPIO24	GPIO23	EMU18	TIMI0	EMU14	EMU12	EMU08	EMU06	PHYSYNC	AJ
UART0TXD	VSS	GPIO08	GPIO03	GPIO15	GPIO26	GPIO22	GPIO20	GPIO17	TIMO0	EMU09	VSS	EMU05	EMU02	RP1FBN	AH
UART1TXD	UART0RXD	GPIO09	GPIO02	GPIO13	GPIO19	VSS	GPIO16	GPIO30	VSS	EMU11	EMU04	EMU01	VSS	RP1FBP	AG
UART1RXD	SCL	GPIO10	GPIO01	GPIO25	GPIO18	GPIO29	GPIO28	GPIO31	TIMO1	TDI	EMU03	EMU00	RADSYNC	RP1CLKP	AF
VSS	SDA	VSS	GPIO21	GPIO27	RSV01	$\overline{\text{POR}}$	RSV20	RSV04	TIMI1	TMS	$\overline{\text{TRST}}$	TCK	RP1CLKN	SYSCLKP	AE
DVDD18	VSS	DVDD18	VSS	DVDD18	VSS	DVDD18	VSS	RSV05	ALTCORE CLKN	TDO	VSS	SYSCLKOUT	VSS	SYSCLKN	AD
VSS	DVDD18	VSS	DVDD18	VSS	DVDD18	VSS	DVDD18	VSS	ALTCORE CLKP	CORECLK SEL	RSV38	VSS	RSV34	VSS	AC
CVDD	VSS	CVDD	VSS	CVDD	VSS	DVDD18	VDDT1	VSS	AVDDA1	VSS	RSV37	RSV35	RSV33	RSV32	AB
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VSS	VDDR_5	VSS	VDDT1	VSS	RSV36	VSS	RSV31	AA
CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VDDT1	VSS	VDDT1	VSS	AIFTXN2	VSS	AIFRXN2	VSS	Y
VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	VSS	VDDT1	VSS	VDDT1	AIFTXP2	AIFTXN3	AIFRXP2	AIFRXN3	W
CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VDDT1	VSS	RSV26	VSS	VSS	AIFTXP3	VSS	AIFRXP3	V
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VSS	VDDR_6	VSS	VDDT1	AIFTXP0	VSS	AIFRXP0	VSS	U
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VDDT1	VSS	VDDT1	VSS	AIFTXN0	AIFTXN1	AIFRXN0	AIFRXP1	T

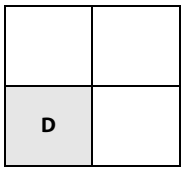
	B

Figure 2-21 Lower Right Quadrant TMS320TCI6612 — Bottom View



VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VSS	VDDT1	RSV27	VDDT1	VSS	AIFTXP1	VSS	AIFRXN1	R
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VDDT1	VSS	VSS	VSS	MCMTXN2	VSS	MCMRXP2	VSS	P
VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	VSS	VDDR_1	VDDT1	MCMTXP3	MCMTXP2	VSS	MCMRXN2	MCMRXN3	N
CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VDDT1	VSS	VSS	MCMTXN3	VSS	VSS	VSS	MCMRXP3	M
VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	VSS	VDDT1	RSV14	VSS	MCMTXP1	VSS	MCMRXP1	VSS	L
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VDDT1	VSS	VDDT1	MCMTXN0	MCMTXN1	VSS	MCMRXN1	MCMRXP0	K
VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD18	RSV13	VSS	MCMTXP0	VSS	RSV29	VSS	MCMRXN0	J
DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD18	VSS	DVDD18	RSV12	VSS	RSV30	VCNTL0	VCNTL3	VSS	H
VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD18	VSS	MCMTX FLCLK	VCNTL1	MCMTX FLDAT	MCMRX FLCLK	VCNTL2	MCMREF CLKOUTN	G
DDRA12	DDRA13	$\overline{\text{DDRCE1}}$	$\overline{\text{DDRCE0}}$	DDRD32	DDRD39	DDRD37	DDRD45	DDRD46	VSS	MCMTX PMCLK	MCMTX PMDAT	MCMRX PMDAT	MCMRX FLDAT	MCMREF CLKOUTP	F
DDRA09	DDRA04	VSS	$\overline{\text{DDRCAS}}$	VSS	DDRDQM4	DDRD38	VSS	DDRD44	DDRD47	DDRD55	VSS	MCMRX PMCLK	MCMCLKP	MCMCLKN	E
DDRA08	DDRA00	DDRODT1	$\overline{\text{DDRRAS}}$	DDRCKE0	DDRD33	DDRD36	DDRD41	DDRD42	DDRD53	DDRD54	DDRD60	DDRD63	DDRD61	DDRD62	D
VSS	DDRA02	DDRODT0	VSS	$\overline{\text{DDRRESET}}$	VSS	DDRD35	DDRD43	VSS	DDRDQS6N	DDRD52	DDRD51	DDRD58	VSS	DDRDQS7P	C
DDRA01	DDRA03	DDRBA2	$\overline{\text{DDRWE}}$	DDRCLK OUTN0	DDRDQS4N	DDRD34	DDRD40	DDRDQS5N	DDRDQS6P	DDRD48	DDRD50	DDRD59	DDRD57	DDRDQS7N	B
DDRA07	VSS	DDRBA0	DDRBA1	DDRCLK OUTP0	DDRDQS4P	VSS	DDRDQM5	DDRDQS5P	VSS	DDRD49	DDRDQM6	DDRDQM7	DDRD56	VSS	A
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	

Figure 2-22 Lower Left Quadrant TMS320TCI6612 — Bottom View



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	VSS	EMIFA14	EMIFA08	EMIFA07	EMIFCE0	VSS	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
P	EMIFA13	EMIFA11	EMIFA04	VSS	EMIFCE1	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
N	EMIFA12	VSS	EMIFA03	EMIFWAIT0	EMIFCE2	VSS	DVDD18	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
M	VSS	EMIFA02	EMIFWAIT1	VSS	VSS	DVDD18	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
L	EMIFA00	EMIFBE1	EMIFOE	VSS	DVDD18	VSS	DVDD18	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
K	EMIFBE0	VSS	PACLKSEL	PTV15	VSS	DVDD18	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
J	RSV03	NMI	VSS	CORESEL1	AVDDA2	VPP	DVDD18	VSS	DVDD18	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15
H	LRESET NMIEN	RSV08	CORESEL2	RESETSTAT	RSV21	RSV28	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS
G	RESETFULL	RESET	CORESEL0	HOUT	VSS	DDRD10	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15
F	RSV09	BOOT COMPLETE	LRESET	VSS	DDRD02	DDRD09	DDRD11	DDRD17	DDRD24	DDRDQM3	VSS	DDRCB02	DDRCB01	DDRA15	VREFHSTL
E	DDRSL RATE0	RSV07	VSS	DDRD03	DDRD08	VSS	DDRDQM2	DDRD19	VSS	DDRD26	DDRD29	VSS	DDRCB00	DDRA14	VSS
D	DDRSL RATE1	RSV06	DDRD04	DDRD07	DDRDQM1	DDRD15	DDRD16	DDRD18	DDRD27	DDRD25	DDRD31	DDRDQM8	DDRCB03	DDRCKE1	DDRA11
C	DDRCLKN	VSS	DDRD00	VSS	DDRDQS1N	DDRD14	VSS	DDRDQS2N	DDRD28	VSS	DDRD30	DDRCB04	VSS	DDRCLK OUTP1	DDRA10
B	DDRCLKP	DDRD01	DDRDQS0P	DDRD05	DDRDQS1P	DDRD13	DDRD20	DDRDQS2P	DDRD23	DDRDQS3N	DDRCB06	DDRCB05	DDRDQS8P	DDRCLK OUTN1	DDRA05
A	VSS	DDRDQM0	DDRDQS0N	DDRD06	VSS	DDRD12	DDRD22	VSS	DDRD21	DDRDQS3P	DDRCB07	VSS	DDRDQS8N	VSS	DDRA06

2.11 Terminal Functions

The terminal functions table (Table 2-20) 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 2-21) lists the various power supply pins and ground pins and gives functional pin descriptions. Table 2-22 shows all pins arranged by signal name. Table 2-23 shows all pins arranged by ball number.

There are 19 pins that have a secondary function as well as a primary function. The secondary function is indicated with a dagger (†).

For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see "Device Configuration" on page 76.

Use the symbol definitions in Table 2-19 when reading Table 2-20.

Table 2-19 I/O Functional Symbol Definitions

Functional Symbol	Definition	Table 2-20 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 Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75.	IPD/IPU
A	Analog signal	Type
GND	Ground	Type
I	Input terminal	Type
O	Output terminal	Type
S	Supply voltage	Type
Z	Three-state terminal or high impedance	Type

Table 2-20 Terminal Functions — Signals and Control by Function (Part 1 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
AIF2				
AIFRXN0	T29	I		Antenna interface receive data (4 links)
AIFRXP0	U29	I		
AIFRXN1	R30	I		
AIFRXP1	T30	I		
AIFRXN2	Y29	I		
AIFRXP2	W29	I		
AIFRXN3	W30	I		
AIFRXP3	V30	I		
AIFTXN0	T27	O		Antenna interface transmit data (4 links)
AIFTXP0	U27	O		
AIFTXN1	T28	O		
AIFTXP1	R28	O		
AIFTXN2	Y27	O		
AIFTXP2	W27	O		
AIFTXN3	W28	O		
AIFTXP3	V28	O		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 2 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
AIF2 Timer (AT) Module				
EXTFRAMEEVENT	AE13	OZ	Down	Frame sync clock output
PHYSYNC	AJ30	I	Down	Frame sync input for phy timer
RP1CLKN	AE29	I		Frame sync interface clock used to drive the frame synchronization interface (OBSAI RP1 clock)
RP1CLKP	AF30	I		
RP1FBN	AH30	I		Frame burst to drive frame indicators to the frame synchronization module (OBSAI RP1)
RP1FBP	AG30	I		
RADSYNC	AF29	I	Down	Frame sync input for radio timer
Boot Configuration Pins				
BOOTMODE00 †	AF19	IOZ	Down	User-defined boot mode pins See 2.4 "Boot Modes Supported and PLL Settings" on page 30 for more details (Pins shared with GPIO[01:13])
BOOTMODE01 †	AG19	IOZ	Down	
BOOTMODE02 †	AH19	IOZ	Down	
BOOTMODE03 †	AK19	IOZ	Down	
BOOTMODE04 †	AJ19	IOZ	Down	
BOOTMODE05 †	AK18	IOZ	Down	
BOOTMODE06 †	AJ18	IOZ	Down	
BOOTMODE07 †	AH18	IOZ	Down	
BOOTMODE08 †	AG18	IOZ	Down	
BOOTMODE09 †	AF18	IOZ	Down	
BOOTMODE10 †	AK22	IOZ	Down	
BOOTMODE11 †	AJ21	IOZ	Down	
BOOTMODE12 †	AG20	IOZ	Down	
BOOTMODE13 †	AG23	IOZ	Down	User-defined boot mode pin (pin shared with GPIO[16])
LENDIAN †	AJ20	IOZ	Up	Little endian configuration pin (pin shared with GPIO[00])
PACLKSEL	K3	O	Down	PA clock select to choose between PASSCLK and the output of Main PLL MUX (dependent on CORECLKSEL pin) to the PA Subsystem PLL
PCIESSEN †	AJ25	I	Down	PCIe_SS module enable pin (pin shared with TIMI0)
PCIESSMODE0 †	AK21	IOZ	Down	PCIe_SS mode 0 pin (pin shared with GPIO[14])
PCIESSMODE1 †	AH20	IOZ	Down	PCIe_SS mode 1 pin (pin shared with GPIO[15])
Clock / Reset				
ALTCORECLKN	AD25	I		System clock input to antenna interface and main PLL (main PLL optional vs. ALTCORECLK)
ALTCORECLKP	AC25	I		
BOOTCOMPLETE	F2	O	Down	Boot progress indication output
CORECLKSEL	AC26	I	Down	Core clock select to select between SYSCLK(N|P) and ALTCORECCLK to the main PLL
CORESEL0	G3	I	Down	Select for the target core for $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$
CORESEL1	J4	I	Down	
CORESEL2	H3	I	Down	
DDRCLKN	C1	I		DDR reference clock input to DDR PLL
DDRCLKP	B1	I		
HOUT	G4	OZ	Up	Interrupt output pulse created by IPCGRH
$\overline{\text{LRESET}}$	F3	I	Up	Warm reset
$\overline{\text{LRESETNMIEN}}$	H1	I	Up	Enable for core selects
MCMCLKN	E30	I		MCM Reference Clock to drive the MCM SERDES
MCMCLKP	E29	I		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 3 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
$\overline{\text{NMI}}$	J2	I	Up	Non-maskable interrupt
PASSCLKN	AK14	I		Packet subsystem reference clock
PASSCLKP	AJ14	I		
PCIECLKN	AJ13	I		PCIe clock input to drive PCIe SerDes
PCIECLKP	AK13	I		
$\overline{\text{POR}}$	AE22	I		Power-on reset
PTV15	K4	A		PTV Compensation NMOS Reference Input. A precision resistor placed between the PTV15 pin and ground is used to closely tune the output impedance of the DDR interface drivers to 50 Ω. Presently, the recommended value for this 1% resistor is 45.3 Ω.
$\overline{\text{RESET}}$	G2	I	Up	Warm reset of non-isolated portion on the device
$\overline{\text{RESETFULL}}$	G1	I	Up	Full reset
$\overline{\text{RESETSTAT}}$	H4	O	Up	Reset status output
SRIOSGMIICLKN	AK11	I		SGMII reference clock to drive the SGMII SerDes
SRIOSGMIICLKP	AK12	I		
SYSCLKN	AD30	I		System clock input to antenna interface and Main PLL (Main PLL optional vs. ALTCORECLK)
SYSCLKP	AE30	I		
SYSCLKOUT	AD28	O	Down	System clock output to be used as a general purpose output clock for debug purposes
DDR				
DDRA00	D17	OZ		DDR EMIF address bus
DDRA01	B16	OZ		
DDRA02	C17	OZ		
DDRA03	B17	OZ		
DDRA04	E17	OZ		
DDRA05	B15	OZ		
DDRA06	A15	OZ		
DDRA07	A16	OZ		
DDRA08	D16	OZ		
DDRA09	E16	OZ		
DDRA10	C15	OZ		
DDRA11	D15	OZ		
DDRA12	F16	OZ		
DDRA13	F17	OZ		
DDRA14	E14	OZ		
DDRA15	F14	OZ		
DDRBA0	A18	OZ		DDR EMIF bank address
DDRBA1	A19	OZ		
DDRBA2	B18	OZ		
$\overline{\text{DDRCAS}}$	E19	OZ		DDR EMIF column address strobe

Table 2-20 Terminal Functions — Signals and Control by Function (Part 4 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRCB00	E13	IOZ		DDR EMIF check bits
DDRCB01	F13	IOZ		
DDRCB02	F12	IOZ		
DDRCB03	D13	IOZ		
DDRCB04	C12	IOZ		
DDRCB05	B12	IOZ		
DDRCB06	B11	IOZ		
DDRCB07	A11	IOZ		
$\overline{\text{DDRCE0}}$	F19	OZ		DDR EMIF chip enable
$\overline{\text{DDRCE1}}$	F18	OZ		
DDRCKE0	D20	OZ		DDR EMIF clock enable0
DDRCKE1	D14	OZ		DDR EMIF clock enable1
DDRCLKOUTP0	A20	OZ		DDR EMIF output clocks to drive SDRAMs (one clock pair per SDRAM)
DDRCLKOUTN0	B20	OZ		
DDRCLKOUTP1	C14	OZ		
DDRCLKOUTN1	B14	OZ		
DDRD00	C3	IOZ		DDR EMIF data bus
DDRD01	B2	IOZ		
DDRD02	F5	IOZ		
DDRD03	E4	IOZ		
DDRD04	D3	IOZ		
DDRD05	B4	IOZ		
DDRD06	A4	IOZ		
DDRD07	D4	IOZ		
DDRD08	E5	IOZ		
DDRD09	F6	IOZ		
DDRD10	G6	IOZ		
DDRD11	F7	IOZ		
DDRD12	A6	IOZ		
DDRD13	B6	IOZ		
DDRD14	C6	IOZ		
DDRD15	D6	IOZ		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 5 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRD16	D7	IOZ		DDR EMIF data bus
DDRD17	F8	IOZ		
DDRD18	D8	IOZ		
DDRD19	E8	IOZ		
DDRD20	B7	IOZ		
DDRD21	A9	IOZ		
DDRD22	A7	IOZ		
DDRD23	B9	IOZ		
DDRD24	F9	IOZ		
DDRD25	D10	IOZ		
DDRD26	E10	IOZ		
DDRD27	D9	IOZ		
DDRD28	C9	IOZ		
DDRD29	E11	IOZ		
DDRD30	C11	IOZ		
DDRD31	D11	IOZ		
DDRD32	F20	IOZ		DDR EMIF data bus
DDRD33	D21	IOZ		
DDRD34	B22	IOZ		
DDRD35	C22	IOZ		
DDRD36	D22	IOZ		
DDRD37	F22	IOZ		
DDRD38	E22	IOZ		
DDRD39	F21	IOZ		
DDRD40	B23	IOZ		
DDRD41	D23	IOZ		
DDRD42	D24	IOZ		
DDRD43	C23	IOZ		
DDRD44	E24	IOZ		
DDRD45	F23	IOZ		
DDRD46	F24	IOZ		
DDRD47	E25	IOZ		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 6 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRD48	B26	IOZ		DDR EMIF data bus
DDRD49	A26	IOZ		
DDRD50	B27	IOZ		
DDRD51	C27	IOZ		
DDRD52	C26	IOZ		
DDRD53	D25	IOZ		
DDRD54	D26	IOZ		
DDRD55	E26	IOZ		
DDRD56	A29	IOZ		
DDRD57	B29	IOZ		
DDRD58	C28	IOZ		
DDRD59	B28	IOZ		
DDRD60	D27	IOZ		
DDRD61	D29	IOZ		
DDRD62	D30	IOZ		
DDRD63	D28	IOZ		
DDRDQM0	A2	OZ		DDR EMIF data masks
DDRDQM1	D5	OZ		
DDRDQM2	E7	OZ		
DDRDQM3	F10	OZ		
DDRDQM4	E21	OZ		
DDRDQM5	A23	OZ		
DDRDQM6	A27	OZ		
DDRDQM7	A28	OZ		
DDRDQM8	D12	OZ		
DDRDQS0P	B3	IOZ		DDR EMIF data strobe
DDRDQS0N	A3	IOZ		
DDRDQS1P	B5	IOZ		
DDRDQS1N	C5	IOZ		
DDRDQS2P	B8	IOZ		
DDRDQS2N	C8	IOZ		
DDRDQS3P	A10	IOZ		
DDRDQS3N	B10	IOZ		
DDRDQS4P	A21	IOZ		
DDRDQS4N	B21	IOZ		
DDRDQS5P	A24	IOZ		
DDRDQS5N	B24	IOZ		
DDRDQS6P	B25	IOZ		
DDRDQS6N	C25	IOZ		
DDRDQS7P	C30	IOZ		
DDRDQS7N	B30	IOZ		
DDRDQS8P	B13	IOZ		
DDRDQS8N	A13	IOZ		
DDRODT0	C18	OZ		DDR EMIF on die termination outputs used to set termination on the SDRAMs

Table 2-20 Terminal Functions — Signals and Control by Function (Part 7 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
DDRODT1	D18	OZ		DDR EMIF on die termination outputs used to set termination on the SDRAMs
$\overline{\text{DDRRAS}}$	D19	OZ		DDR EMIF row address strobe
$\overline{\text{DDRRESET}}$	C20	OZ		DDR reset signal
DDRSLRATE0	E1	I	Down	DDR slew rate control
DDRSLRATE1	D1	I	Down	
$\overline{\text{DDRWE}}$	B19	OZ		DDR EMIF write enable
VREFHSTL	F15	P		Reference voltage input for HSTL15 buffers used by DDR EMIF (VDDS15/2)
EMIF16				
EMIFA00	L1	O	Down	EMIF address
EMIFA01	W5	O	Down	
EMIFA02	M2	O	Down	
EMIFA03	N3	O	Down	
EMIFA04	P3	O	Down	
EMIFA05	T4	O	Down	
EMIFA06	U4	O	Down	
EMIFA07	R4	O	Down	
EMIFA08	R3	O	Down	
EMIFA09	W4	O	Down	
EMIFA10	Y5	O	Down	
EMIFA11	P2	O	Down	
EMIFA12	N1	O	Down	
EMIFA13	P1	O	Down	EMIF address
EMIFA14	R2	O	Down	
EMIFA15	T2	O	Down	
EMIFA16	U3	O	Down	
EMIFA17	V3	O	Down	
EMIFA18	Y4	O	Down	
EMIFA19	T1	O	Down	
EMIFA20	W3	O	Down	
EMIFA21	V2	O	Down	
EMIFA22	AA5	O	Down	
EMIFA23	U1	O	Down	
$\overline{\text{EMIFBE0}}$	K1	O	Up	EMIF control signals
$\overline{\text{EMIFBE1}}$	L2	O	Up	
$\overline{\text{EMIFCE0}}$	R5	O	Up	
$\overline{\text{EMIFCE1}}$	P5	O	Up	
$\overline{\text{EMIFCE2}}$	N5	O	Up	
$\overline{\text{EMIFCE3}}$	V5	O	Up	
$\overline{\text{EMIFOE}}$	L3	O	Up	
EMIF$\overline{\text{RW}}$	U5	O	Up	
EMIFWAIT0	N4	I	Down	
EMIFWAIT1	M3	I	Down	
$\overline{\text{EMIFWE}}$	T5	O	Up	

Table 2-20 Terminal Functions — Signals and Control by Function (Part 8 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
EMIFD00	V1	IOZ	Down	EMIF data
EMIFD01	AA4	IOZ	Down	
EMIFD02	Y2	IOZ	Down	
EMIFD03	W2	IOZ	Down	
EMIFD04	AA3	IOZ	Down	
EMIFD05	Y1	IOZ	Down	
EMIFD06	AB1	IOZ	Down	
EMIFD07	AA1	IOZ	Down	
EMIFD08	AB2	IOZ	Down	
EMIFD09	AB3	IOZ	Down	
EMIFD10	AC2	IOZ	Down	
EMIFD11	AD1	IOZ	Down	
EMIFD12	AB4	IOZ	Down	
EMIFD13	AD2	IOZ	Down	
EMIFD14	AC3	IOZ	Down	
EMIFD15	AC4	IOZ	Down	
EMU				
EMU00	AF28	IOZ	Up	Emulation and trace port
EMU01	AG28	IOZ	Up	
EMU02	AH29	IOZ	Up	
EMU03	AF27	IOZ	Up	
EMU04	AG27	IOZ	Up	
EMU05	AH28	IOZ	Up	
EMU06	AJ29	IOZ	Up	
EMU07	AK29	IOZ	Up	
EMU08	AJ28	IOZ	Up	
EMU09	AH26	IOZ	Up	
EMU10	AK28	IOZ	Up	
EMU11	AG26	IOZ	Up	
EMU12	AJ27	IOZ	Up	
EMU13	AK27	IOZ	Up	
EMU14	AJ26	IOZ	Up	
EMU15	AK26	IOZ	Up	
EMU16	AK25	IOZ	Up	
EMU17	AK24	IOZ	Up	
EMU18	AJ24	IOZ	Up	

Table 2-20 Terminal Functions — Signals and Control by Function (Part 9 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
EMU19 †	AH24	IOZ	Up	Emulation and trace ports (Pins shared with GPIO[17:31])
EMU20 †	AF21	IOZ	Up	
EMU21 †	AG21	IOZ	Up	
EMU22 †	AH23	IOZ	Up	
EMU23 †	AE19	IOZ	Up	
EMU24 †	AH22	IOZ	Up	
EMU25 †	AJ23	IOZ	Up	
EMU26 †	AJ22	IOZ	Up	
EMU27 †	AF20	IOZ	Up	
EMU28 †	AH21	IOZ	Up	
EMU29 †	AE20	IOZ	Up	
EMU30 †	AF23	IOZ	Up	
EMU31 †	AF22	IOZ	Up	
EMU32 †	AG24	IOZ	Up	
EMU33 †	AF24	IOZ	Up	
General Purpose Input/Output (GPIO)				
GPIO00	AJ20	IOZ	Up	General purpose input/output These GPIO pins have secondary functions assigned to them as mentioned in the ***Boot Configuration Pins*** section of this table, above.
GPIO01	AF19	IOZ	Down	
GPIO02	AG19	IOZ	Down	
GPIO03	AH19	IOZ	Down	
GPIO04	AK19	IOZ	Down	
GPIO05	AJ19	IOZ	Down	
GPIO06	AK18	IOZ	Down	
GPIO07	AJ18	IOZ	Down	
GPIO08	AH18	IOZ	Down	
GPIO09	AG18	IOZ	Down	
GPIO10	AF18	IOZ	Down	
GPIO11	AK22	IOZ	Down	
GPIO12	AJ21	IOZ	Down	
GPIO13	AG20	IOZ	Down	
GPIO14	AK21	IOZ	Down	
GPIO15	AH20	IOZ	Down	
GPIO16	AG23	IOZ	Down	

Table 2-20 Terminal Functions — Signals and Control by Function (Part 10 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
GPIO17	AH24	IOZ	Up	General purpose input/output These GPIO pins have secondary functions assigned to them as mentioned in the ***EMU*** section of this table, above.
GPIO18	AF21	IOZ	Up	
GPIO19	AG21	IOZ	Up	
GPIO20	AH23	IOZ	Up	
GPIO21	AE19	IOZ	Up	
GPIO22	AH22	IOZ	Up	
GPIO23	AJ23	IOZ	Up	
GPIO24	AJ22	IOZ	Up	
GPIO25	AF20	IOZ	Up	
GPIO26	AH21	IOZ	Up	
GPIO27	AE20	IOZ	Up	
GPIO28	AF23	IOZ	Up	
GPIO29	AF22	IOZ	Up	
GPIO30	AG24	IOZ	Up	
GPIO31	AF24	IOZ	Up	
HyperLink				
MCMRXN0	J30	I		Serial HyperLink receive data (4 links)
MCMRXP0	K30	I		
MCMRXN1	K29	I		
MCMRXP1	L29	I		
MCMRXN2	N29	I		
MCMRXP2	P29	I		
MCMRXN3	N30	I		
MCMRXP3	M30	I		
MCMTXN0	K26	O		Serial HyperLink transmit data (4 links)
MCMTXP0	J26	O		
MCMTXN1	K27	O		
MCMTXP1	L27	O		
MCMTXN2	P27	O		
MCMTXP2	N27	O		
MCMTXN3	M26	O		
MCMTXP3	N26	O		
MCMRXFLCLK	G28	O	Down	Serial HyperLink sideband signals
MCMRXFLDAT	F29	O	Down	
MCMTXFLCLK	G25	I	Down	
MCMTXFLDAT	G27	I	Down	
MCMRXPMCLK	E28	I	Down	
MCMRXPMDAT	F28	I	Down	
MCMTXPMCLK	F26	O	Down	
MCMTXPMDAT	F27	O	Down	
MCMREFCLKOUTN	G30	O		Reference clock output for daisy chain connection
MCMREFCLKOUTP	F30	O		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 11 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
I^2C				
SCL	AF17	IOZ		I^2C clock
SDA	AE17	IOZ		I^2C data
JTAG				
TCK	AE28	I	Up	JTAG clock input
TDI	AF26	I	Up	JTAG data input
TDO	AD26	OZ	Up	JTAG data output
TMS	AE26	I	Up	JTAG test mode input
$\overline{TRST}$	AE27	I	Down	JTAG reset
MDIO				
MDCLK	AF12	O	Down	MDIO clock
MDIO	AD12	IOZ	Up	MDIO data
PCIe				
PCIERXN0	AK9	I		PCIexpress receive data (2 links)
PCIERXP0	AK8	I		
PCIERXN1	AJ10	I		
PCIERXP1	AJ11	I		
PCIETXN0	AG11	O		PCIexpress transmit data (2 links)
PCIETXP0	AG10	O		
PCIETXN1	AH9	O		
PCIETXP1	AH8	O		
Serial RapidIO				
RIORXN0	AK6	I		Serial RapidIO receive data (2 links)
RIORXP0	AK5	I		
RIORXN1	AJ5	I		
RIORXP1	AJ4	I		
RIORXN2	AK2	I		Serial RapidIO receive data (2 links)
RIORXP2	AK3	I		
RIORXN3	AJ1	I		
RIORXP3	AJ2	I		
RIOTXN0	AJ8	O		Serial RapidIO transmit data (2 links)
RIOTXP0	AJ7	O		
RIOTXN1	AG7	O		
RIOTXP1	AG8	O		
RIOTXN2	AH5	O		Serial RapidIO transmit data (2 links)
RIOTXP2	AH6	O		
RIOTXN3	AG4	O		
RIOTXP3	AG5	O		
SGMII				
SGMII0RXN	AG1	I		Ethernet MAC SGMII port 0 receive data
SGMII0RXP	AG2	I		
SGMII0TXN	AE3	O		Ethernet MAC SGMII port 0 transmit data
SGMII0TXP	AE4	O		

Table 2-20 Terminal Functions — Signals and Control by Function (Part 12 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
SGMII1RXN	AH2	I		Ethernet MAC SGMII port 1 receive data
SGMII1RXP	AH3	I		
SGMII1TXN	AF2	O		Ethernet MAC SGMII port 1 transmit data
SGMII1TXP	AF3	O		
SmartReflex				
VCNTL0	H28	OZ		Voltage control outputs to variable core power supply
VCNTL1	G26	OZ		
VCNTL2	G29	OZ		
VCNTL3	H29	OZ		
SPI				
SPICLK	AH14	OZ	Down	SPI clock
SPIDIN	AG14	I	Down	SPI data in
SPIDOUT	AF14	OZ	Down	SPI data out
SPISCS0	AG15	OZ	Up	SPI interface enable 0
SPISCS1	AF15	OZ	Up	SPI interface enable 1
SPISCS2	AK15	OZ	Up	SPI interface enable 2
SPISCS3	AJ15	OZ	Up	SPI interface enable 3
SPISCS4	AH15	OZ	Up	SPI interface enable 4
Timer				
TIMI0	AJ25	I	Down	Timer inputs
TIMI1	AE25	I	Down	
TIMO0	AH25	OZ	Down	Timer outputs
TIMO1	AF25	OZ	Down	
UART				
UART0CTS	AJ17	I	Down	UART0 clear to send
UART0RTS	AK17	OZ	Down	UART0 request to send
UART0RXD	AG17	I	Down	UART0 serial data in
UART0TXD	AH16	OZ	Down	UART0 serial data out
UART1CTS	AK16	I	Down	UART1 clear to send
UART1RTS	AJ16	OZ	Down	UART1 request to send
UART1RXD	AF16	I	Down	UART1 serial data in
UART1TXD	AG16	OZ	Down	UART1 serial data out
USIM				
USIMCLK	AH13	OZ	Down	USIM clock
USIMIO	AF13	IOZ	Down	USIM data
USIMRST	AG13	OZ	Down	USIM reset

Table 2-20 Terminal Functions — Signals and Control by Function (Part 13 of 13)

Signal Name	Ball No.	Type	IPD/IPU	Description
Reserved				
RSV01	AE21			Pull up to 1.8 V
RSV03	J1			Leave unconnected
RSV04	AE24			Leave unconnected
RSV05	AD24			Leave unconnected
RSV06	D2			Leave unconnected
RSV07	E2			Leave unconnected
RSV08	H2			Connect to ground
RSV09	F1			Leave unconnected
RSV10	AD7			Leave unconnected
RSV11	AC7			Leave unconnected
RSV12	H25			Leave unconnected
RSV13	J24			Leave unconnected
RSV14	L25			Leave unconnected
RSV15	AE1			Leave unconnected
RSV16	AD5			Leave unconnected
RSV17	AD4			Leave unconnected
RSV20	AE23			Leave unconnected
RSV21	H5			Leave unconnected
RSV22	AD6			Leave unconnected
RSV24	AH12			Leave unconnected
RSV25	AH11			Leave unconnected
RSV26	V25			Leave unconnected
RSV27	R25			Leave unconnected
RSV29	J28			Leave unconnected
RSV30	H27			Leave unconnected
RSV31	AA30			Leave unconnected
RSV32	AB30			Leave unconnected
RSV33	AB29			Leave unconnected
RSV34	AC29			Leave unconnected
RSV35	AB28			Leave unconnected
RSV36	AA28			Leave unconnected
RSV37	AB27			Leave unconnected
RSV38	AC27			Leave unconnected
End of Table 2-20				

Table 2-21 Terminal Functions — Power and Ground

Supply	Ball No.	Volts	Description
AVDDA1	AB25	1.8	**Main PLL Power Supply Pin place holder** PLL Supply: CORE_PLL
AVDDA2	J5	1.8	**DDR PLL Power Supply Pin place holder** PLL Supply: DDR3_PLL
AVDDA3	AD14	1.8	**PA PLL Power Supply Pin place holder** PLL Supply: PASS_PLL
CVDD	K8, K10, K12, K14, K16, K18, K20, K22, L11, L13, L15, L17, L19, M8, M12, M14, M16, M18, M22, N11, N13, N15, N17, N19, P8, P10, P12, P14, P16, P18, P20, P22, R9, R11, R13, R15, R17, R19, R21, T8, T10, T12, T14, T16, T18, T20, T22, U9, U11, U13, U15, U17, U19, U21, V8, V10, V12, V14, V16, V18, V22, W9, W11, W13, W15, W17, W19, Y8, Y12, Y14, Y16, Y18, Y22, AA11, AA13, AA15, AA17, AA19, AA21, AB8, AB12, AB14, AB16, AB18, AB20, AC9	0.9 to 1.1	SmartReflex core supply voltage
CVDD1	L9, L21, M10, M20, N9, N21, V20, W21, Y10, Y20, AA9, AB10	1.0	Fixed core supply voltage
DVDD15	G7, G9, G11, G13, G15, G17, G19, G21, H8, H10, H12, H14, H16, H18, H20, J11, J13, J15, J17, J19, J21	1.5	DDR3 IO supply
DVDD18	G23, H22, H24, J7, J9, J23, K6, L5, L7, M6, N7, P6, R7, T6, U7, V6, W7, Y6, AA7, AB6, AB22, AC13, AC15, AC17, AC19, AC21, AC23, AD16, AD18, AD20, AD22, AE15	1.8	IO supply
VDDR_1	N24	1.5	HyperLink SerDes regulator supply
VDDR_2	AF9	1.5	PCIe SerDes regulator supply
VDDR_3	AC5	1.5	SGMII SerDes regulator supply
VDDR_4	AE5	1.5	SRIO SerDes regulator supply
VDDR_5	AA24	1.5	AIF SerDes regulator supply
VDDR_6	U24	1.5	
VDDT1	K23, K25, L24, M23, N25, P23, R24, R26, T23, T25, U26, V23, W24, W26, Y23, Y25, AA26, AB23	1.0	HyperLink/AIF SerDes termination supply
VDDT2	AC11, AD8, AD10, AE7, AE9, AE11, AF6, AF8, AF10	1.0	SGMII/SRIO/PCIe SerDes termination supply
VPP	J6	1.8	Supply voltage for OTP memory on secure devices [1]. See the *Security Addendum for KeyStone I Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75 for more information. Leave unconnected on unsecure devices.
VREFHSTL	F15	0.75	DDR3 reference voltage
VSS	A1, A5, A8, A12, A14, A17, A22, A25, A30, C2, C4, C7, C10, C13, C16, C19, C21, C24, C29, E3, E6, E9, E12, E15, E18, E20, E23, E27, F4, F11, F25, G5, G8, G10, G12, G14, G16, G18, G20, G22, G24, H7, H9, H11, H13, H15, H17, H19, H21, H23, H26, H30, J3, J8, J10, J12, J14, J16, J18, J20, J22, J25, J27, J29, K2, K5, K7, K9, K11, K13, K15, K17, K19, K21, K24, K28, L4, L6, L8, L10, L12, L14, L16, L18, L20, L22, L23, L26, L28, L30, M1, M4, M5, M7, M9, M11, M13, M15, M17, M19, M21, K15, K17, K19, K21, K24, K28, L4, L6, L8, L10, L12, L14, L16, L18, L20, L22, L23, L26, L28, L30, M1, M4, M5, M7, M9, M11, M13, M15, M17, M19, M21, M21, M24, M25, M27, M28, M29, N2, N6, N8, N10, N12, N14, N16, N18, N20, N22, N23, N28, P4, P7, P9, P11, P13, P15, P17, P19, P21, P24, P25, P26, P28, P30, R1, R6, R8, R10, R12, R14, R16, R18, R20, R22, R23, R27, R29, T3, T7, T9, T11, T13, T15, T17, T19, T21, T24, T26, U2, U6, U8, U10, U12, U14, U16, U18, U20, U22, U23, U25, U28, U30, V4, V7, V9, V11, V13, V15, V17, V19, V21, V24, V26, V27, V29, W1, W6, W8, W10, W12, W14, W16, W18, W20, W22, W23, W25, Y3, Y7, Y9, Y11, Y13, Y15, Y17, Y19, Y21, Y24, Y26, Y28, Y30, AA2, AA6, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AA23, AA25, AA27, AA29, AB5, AB7, AB9, AB11, AB13, AB15, AB17, AB19, AB21, AB24, AB26, AC1, AC6, AC8, AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AC28, AC30, AD3, AD9, AD11, AD13, AD15, AD17, AD19, AD21, AD23, AD27, AD29, AE2, AE6, AE8, AE10, AE12, AE14, AE16, AE18, AF1, AF4, AF5, AF7, AF11, AG3, AG6, AG9, AG12, AG22, AG25, AG29, AH1, AH4, AH7, AE18, AF1, AF4, AF5, AF7, AF11, AG3, AG6, AG9, AG12, AG22, AG25, AG29, AH1, AH4, AH7, AH10, AH17, AH27, AJ3, AJ6, AJ9, AJ12, AK1, AK4, AK7, AK10, AK20, AK23, AK30	Gnd	Ground
End of Table 2-21			

1 The secure version of the TCI6612 device contains hardware features to support security within the device. See Figure 2-23 for the *SoC SECURITY* symbol in the device nomenclature.

Table 2-22 Terminal Functions — By Signal Name (Part 1 of 13)

Signal Name	Ball Number
AIFRXN0	T29
AIFRXN1	R30
AIFRXN2	Y29
AIFRXN3	W30
AIFRXP0	U29
AIFRXP1	T30
AIFRXP2	W29
AIFRXP3	V30
AIFTXN0	T27
AIFTXN1	T28
AIFTXN2	Y27
AIFTXN3	W28
AIFTXP0	U27
AIFTXP1	R28
AIFTXP2	W27
AIFTXP3	V28
ALTCORECLKN	AD25
ALTCORECLKP	AC25
AVDDA1	AB25
AVDDA2	J5
AVDDA3	AD14
BOOTCOMPLETE	F2
BOOTMODE00 †	AF19
BOOTMODE01 †	AG19
BOOTMODE02 †	AH19
BOOTMODE03 †	AK19
BOOTMODE04 †	AJ19
BOOTMODE05	AK18
BOOTMODE06 †	AJ18
BOOTMODE07 †	AH18
BOOTMODE08 †	AG18
BOOTMODE09 †	AF18
BOOTMODE10 †	AK22
BOOTMODE11 †	AJ21
BOOTMODE12 †	AG20
BOOTMODE13 †	AG23
LENDIAN †	AJ20
PCIESSEN †	AJ25
PCIESSMODE0 †	AK21
PCIESSMODE1 †	AH20
CORECLKSEL	AC26
CORESEL0	G3

Table 2-22 Terminal Functions — By Signal Name (Part 2 of 13)

Signal Name	Ball Number
CORESEL1	J4
CORESEL2	H3
CVDD	K8, K10, K12, K14, K16, K18, K20, K22, L11, L13, L15, L17, L19, M8, M12, M14, M16, M18, M22, N11, N13, N15, N17, N19, P8, P10, P12, P14, P16, P18, P20
CVDD	P22, R9, R11, R13, R15, R17, R19, R21, T8, T10, T12, T14, T16, T18, T20, T22, U9, U11, U13, U15, U17, U19, U21, V8, V10, V12, V14, V16, V18, V22, W9, W11
CVDD	W13, W15, W17, W19, Y8, Y12, Y14, Y16, Y18, Y22, AA11, AA13, AA15, AA17, AA19, AA21, AB8, AB12, AB14, AB16, AB18, AB20, AC9
CVDD1	L9, L21, M10, M20, N9, N21, V20, W21, Y10, Y20, AA9, AB10
DDRA00	D17
DDRA01	B16
DDRA02	C17
DDRA03	B17
DDRA04	E17
DDRA05	B15
DDRA06	A15
DDRA07	A16
DDRA08	D16
DDRA09	E16
DDRA10	C15
DDRA11	D15
DDRA12	F16
DDRA13	F17
DDRA14	E14
DDRA15	F14
DDRBA0	A18
DDRBA1	A19
DDRBA2	B18
$\overline{\text{DDRCAS}}$	E19
DDRCB00	E13
DDRCB01	F13

Table 2-22 Terminal Functions — By Signal Name (Part 3 of 13)

Signal Name	Ball Number
DDRCB02	F12
DDRCB03	D13
DDRCB04	C12
DDRCB05	B12
DDRCB06	B11
DDRCB07	A11
$\overline{\text{DDRCE0}}$	F19
$\overline{\text{DDRCE1}}$	F18
DDRCKE0	D20
DDRCKE1	D14
DDRCLKN	C1
DDRCLKOUTN0	B20
DDRCLKOUTN1	B14
DDRCLKOUTP0	A20
DDRCLKOUTP1	C14
DDRCLKP	B1
DDRD00	C3
DDRD01	B2
DDRD02	F5
DDRD03	E4
DDRD04	D3
DDRD05	B4
DDRD06	A4
DDRD07	D4
DDRD08	E5
DDRD09	F6
DDRD10	G6
DDRD11	F7
DDRD12	A6
DDRD13	B6
DDRD14	C6
DDRD15	D6
DDRD16	D7
DDRD17	F8
DDRD18	D8
DDRD19	E8
DDRD20	B7
DDRD21	A9
DDRD22	A7
DDRD23	B9
DDRD24	F9
DDRD25	D10

Table 2-22 Terminal Functions — By Signal Name (Part 4 of 13)

Signal Name	Ball Number
DDRD26	E10
DDRD27	D9
DDRD28	C9
DDRD29	E11
DDRD30	C11
DDRD31	D11
DDRD32	F20
DDRD33	D21
DDRD34	B22
DDRD35	C22
DDRD36	D22
DDRD37	F22
DDRD38	E22
DDRD39	F21
DDRD40	B23
DDRD41	D23
DDRD42	D24
DDRD43	C23
DDRD44	E24
DDRD45	F23
DDRD46	F24
DDRD47	E25
DDRD48	B26
DDRD49	A26
DDRD50	B27
DDRD51	C27
DDRD52	C26
DDRD53	D25
DDRD54	D26
DDRD55	E26
DDRD56	A29
DDRD57	B29
DDRD58	C28
DDRD59	B28
DDRD60	D27
DDRD61	D29
DDRD62	D30
DDRD63	D28
DDRDQM0	A2
DDRDQM1	D5
DDRDQM2	E7
DDRDQM3	F10

Table 2-22 Terminal Functions — By Signal Name (Part 5 of 13)

Signal Name	Ball Number
DDRDQM4	E21
DDRDQM5	A23
DDRDQM6	A27
DDRDQM7	A28
DDRDQM8	D12
DDRDQS0N	A3
DDRDQS0P	B3
DDRDQS1N	C5
DDRDQS1P	B5
DDRDQS2N	C8
DDRDQS2P	B8
DDRDQS3N	B10
DDRDQS3P	A10
DDRDQS4N	B21
DDRDQS4P	A21
DDRDQS5N	B24
DDRDQS5P	A24
DDRDQS6N	C25
DDRDQS6P	B25
DDRDQS7N	B30
DDRDQS7P	C30
DDRDQS8N	A13
DDRDQS8P	B13
DDRODT0	C18
DDRODT1	D18
$\overline{\text{DDRRAS}}$	D19
$\overline{\text{DDRRESET}}$	C20
DDRSLRATE0	E1
DDRSLRATE1	D1
$\overline{\text{DDRWE}}$	B19
DVDD15	G7, G9, G11, G13, G15, G17, G19, G21, H8, H10, H12, H14, H16, H18, H20, J11, J13, J15, J17, J19, J21
DVDD18	G23, H22, H24, J7, J9, J23, K6, L5, L7, M6, N7, P6, R7, T6, U7, V6, W7, Y6, AA7, AB6, AB22, AC13, AC15, AC17, AC19, AC21, AC23, AD16, AD18, AD20, AD22, AE15
EMIFA00	L1
EMIFA01	W5

Table 2-22 Terminal Functions — By Signal Name (Part 6 of 13)

Signal Name	Ball Number
EMIFA02	M2
EMIFA03	N3
EMIFA04	P3
EMIFA05	T4
EMIFA06	U4
EMIFA07	R4
EMIFA08	R3
EMIFA09	W4
EMIFA10	Y5
EMIFA11	P2
EMIFA12	N1
EMIFA13	P1
EMIFA14	R2
EMIFA15	T2
EMIFA16	U3
EMIFA17	V3
EMIFA18	Y4
EMIFA19	T1
EMIFA20	W3
EMIFA21	V2
EMIFA22	AA5
EMIFA23	U1
$\overline{\text{EMIFBE0}}$	K1
$\overline{\text{EMIFBE1}}$	L2
$\overline{\text{EMIFCE0}}$	R5
$\overline{\text{EMIFCE1}}$	P5
$\overline{\text{EMIFCE2}}$	N5
$\overline{\text{EMIFCE3}}$	V5
EMIFD00	V1
EMIFD01	AA4
EMIFD02	Y2
EMIFD03	W2
EMIFD04	AA3
EMIFD05	Y1
EMIFD06	AB1
EMIFD07	AA1
EMIFD08	AB2
EMIFD09	AB3
EMIFD10	AC2
EMIFD11	AD1
EMIFD12	AB4
EMIFD13	AD2

Table 2-22 Terminal Functions — By Signal Name (Part 7 of 13)

Signal Name	Ball Number
EMIFD14	AC3
EMIFD15	AC4
$\overline{\text{EMIFOE}}$	L3
EMIFRNW	U5
EMIFWAIT0	N4
EMIFWAIT1	M3
$\overline{\text{EMIFWE}}$	T5
EMU00	AF28
EMU01	AG28
EMU02	AH29
EMU03	AF27
EMU04	AG27
EMU05	AH28
EMU06	AJ29
EMU07	AK29
EMU08	AJ28
EMU09	AH26
EMU10	AK28
EMU11	AG26
EMU12	AJ27
EMU13	AK27
EMU14	AJ26
EMU15	AK26
EMU16	AK25
EMU17	AK24
EMU18	AJ24
EXTFRAMEEVENT	AE13
GPIO00	AJ20
GPIO01	AF19
GPIO02	AG19
GPIO03	AH19
GPIO04	AK19
GPIO05	AJ19
GPIO06	AK18
GPIO07	AJ18
GPIO08	AH18
GPIO09	AG18
GPIO10	AF18
GPIO11	AK22
GPIO12	AJ21
GPIO13	AG20
GPIO14	AK21

Table 2-22 Terminal Functions — By Signal Name (Part 8 of 13)

Signal Name	Ball Number
GPIO15	AH20
GPIO16	AG23
GPIO17	AH24
GPIO18	AF21
GPIO19	AG21
GPIO20	AH23
GPIO21	AE19
GPIO22	AH22
GPIO23	AJ23
GPIO24	AJ22
GPIO25	AF20
GPIO26	AH21
GPIO27	AE20
GPIO28	AF23
GPIO29	AF22
GPIO30	AG24
GPIO31	AF24
HOUT	G4
$\overline{\text{LRESETNMIEN}}$	H1
$\overline{\text{LRESET}}$	F3
MCMCLKN	E30
MCMCLKP	E29
MCMREFCLKOUTN	G30
MCMREFCLKOUTP	F30
MCMRXFLCLK	G28
MCMRXFLDAT	F29
MCMRXN0	J30
MCMRXN1	K29
MCMRXN2	N29
MCMRXN3	N30
MCMRXP0	K30
MCMRXP1	L29
MCMRXP2	P29
MCMRXP3	M30
MCMRXPMCLK	E28
MCMRXPMDAT	F28
MCMTXFLCLK	G25
MCMTXFLDAT	G27
MCMTXN0	K26
MCMTXN1	K27
MCMTXN2	P27
MCMTXN3	M26

Table 2-22 Terminal Functions — By Signal Name (Part 9 of 13)

Signal Name	Ball Number
MCMTXP0	J26
MCMTXP1	L27
MCMTXP2	N27
MCMTXP3	N26
MCMTXPMCLK	F26
MCMTXPMDAT	F27
MDCLK	AF12
MDIO	AD12
$\overline{\text{NMI}}$	J2
PACLKSEL	K3
PASSCLKN	AK14
PASSCLKP	AJ14
PCIECLKN	AJ13
PCIECLKP	AK13
PCIERXN0	AK9
PCIERXN1	AJ10
PCIERXP0	AK8
PCIERXP1	AJ11
PCIETXN0	AG11
PCIETXN1	AH9
PCIETXP0	AG10
PCIETXP1	AH8
PHYSYNC	AJ30
$\overline{\text{POR}}$	AE22
PTV15	K4
RADSYNC	AF29
$\overline{\text{RESETFULL}}$	G1
$\overline{\text{RESETSTAT}}$	H4
$\overline{\text{RESET}}$	G2
RIORXN0	AK6
RIORXN1	AJ5
RIORXN2	AK2
RIORXN3	AJ1
RIORXP0	AK5
RIORXP1	AJ4
RIORXP2	AK3
RIORXP3	AJ2
RIOTXN0	AJ8
RIOTXN1	AG7
RIOTXN2	AH5
RIOTXN3	AG4
RIOTXP0	AJ7

Table 2-22 Terminal Functions — By Signal Name (Part 10 of 13)

Signal Name	Ball Number
RIOTXP1	AG8
RIOTXP2	AH6
RIOTXP3	AG5
RP1CLKN	AE29
RP1CLKP	AF30
RP1FBN	AH30
RP1FBP	AG30
RSV01	AE21
RSV03	J1
RSV04	AE24
RSV05	AD24
RSV06	D2
RSV07	E2
RSV08	H2
RSV09	F1
RSV10	AD7
RSV11	AC7
RSV12	H25
RSV13	J24
RSV14	L25
RSV15	AE1
RSV16	AD5
RSV17	AD4
RSV20	AE23
RSV21	H5
RSV22	AD6
RSV24	AH12
RSV25	AH11
RSV26	V25
RSV27	R25
RSV28	H6
RSV29	J28
RSV30	H27
RSV31	AA30
RSV32	AB30
RSV33	AB29
RSV34	AC29
RSV35	AB28
RSV36	AA28
RSV37	AB27
RSV38	AC27
SCL	AF17

Table 2-22 Terminal Functions — By Signal Name (Part 11 of 13)

Signal Name	Ball Number
SDA	AE17
SGMII0RXN	AG1
SGMII0RXP	AG2
SGMII0TXN	AE3
SGMII0TXP	AE4
SGMII1RXN	AH2
SGMII1RXP	AH3
SGMII1TXN	AF2
SGMII1TXP	AF3
SPICLK	AH14
SPIDIN	AG14
SPIDOUT	AF14
SPISCS0	AG15
SPISCS1	AF15
SPISCS2	AK15
SPISCS3	AJ15
SPISCS4	AH15
SRIOSGMIICLKN	AK11
SRIOSGMIICLKP	AK12
SYSCLKN	AD30
SYSCLKOUT	AD28
SYSCLKP	AE30
TCK	AE28
TDI	AF26
TDO	AD26
TIMI0	AJ25
TIMI1	AE25
TIMO0	AH25
TIMO1	AF25
TMS	AE26
$\overline{\text{TRST}}$	AE27
UART0CTS	AJ17
UART0RTS	AK17
UART0RXD	AG17
UART0TXD	AH16
UART1CTS	AK16
UART1RTS	AJ16
UART1RXD	AF16
UART1TXD	AG16
USIMCLK	AH13
USIMIO	AF13
USIMRST	AG13

Table 2-22 Terminal Functions — By Signal Name (Part 12 of 13)

Signal Name	Ball Number
VCNTL0	H28
VCNTL1	G26
VCNTL2	G29
VCNTL3	H29
VDDR_1	N24
VDDR_2	AF9
VDDR_3	AC5
VDDR_4	AE5
VDDR_5	AA24
VDDR_6	U24
VDDT1	K23, K25, L24, M23, N25, P23, R24, R26, T23, T25, U26, V23, W24, W26, Y23, Y25, AA26, AB23
VDDT2	AC11, AD8, AD10, AE7, AE9, AE11, AF6, AF8, AF10
VPP	J6
VREFHSTL	F15
VSS	A1, A5, A8, A12, A14, A17, A22, A25, A30, C2, C4, C7, C10, C13, C16, C19, C21, C24, C29, E3, E6, E9, E12, E15, E18, E20, E23, E27, F4, F11, F25, G5, G8, G10, G12
VSS	G14, G16, G18, G20, G22, G24, H7, H9, H11, H13, H15, H17, H19, H21, H23, H26, H30, J3, J8, J10, J12, J14, J16, J18, J20, J22, J25, J27, J29, K2, K5, K7, K9, K11, K13
VSS	K15, K17, K19, K21, K24, K28, L4, L6, L8, L10, L12, L14, L16, L18, L20, L22, L23, L26, L28, L30, M1, M4, M5, M7, M9, M11, M13, M15, M17, M19, M21
VSS	M21, M24, M25, M27, M28, M29, N2, N6, N8, N10, N12, N14, N16, N18, N20, N22, N23, N28, P4, P7, P9, P11, P13, P15, P17, P19, P21, P24, P25, P26, P28

Table 2-22 Terminal Functions — By Signal Name (Part 13 of 13)

Signal Name	Ball Number
VSS	P30, R1, R6, R8, R10, R12, R14, R16, R18, R20, R22, R23, R27, R29, T3, T7, T9, T11, T13, T15, T17, T19, T21, T24, T26, U2, U6, U8, U10, U12, U14, U16, U18
VSS	U20, U22, U23, U25, U28, U30, V4, V7, V9, V11, V13, V15, V17, V19, V21, V24, V26, V27, V29, W1, W6, W8, W10, W12, W14, W16, W18, W20, W22, W23, W25
VSS	Y3, Y7, Y9, Y11, Y13, Y15, Y17, Y19, Y21, Y24, Y26, Y28, Y30, AA2, AA6, AA8, AA10, AA12, AA14, AA16, AA18, AA20, AA22, AA23, AA25, AA27, AA29, AB5
VSS	AB7, AB9, AB11, AB13, AB15, AB17, AB19, AB21, AB24, AB26, AC1, AC6, AC8, AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AC28, AC30, AD3
VSS	AD9, AD11, AD13, AD15, AD17, AD19, AD21, AD23, AD27, AD29, AE2, AE6, AE8, AE10, AE12, AE14, AE16, AE18, AF1, AF4, AF5, AF7, AF11, AG3, AG6
VSS	AG9, AG12, AG22, AG25, AG29, AH1, AH4, AH7, AE18, AF1, AF4, AF5, AF7, AF11, AG3, AG6, AG9, AG12, AG22, AG25, AG29, AH1, AH4, AH7, AH10
VSS	AH17, AH27, AJ3, AJ6, AJ9, AJ12, AK1, AK4, AK7, AK10, AK20, AK23, AK30
End of Table 2-22	

Table 2-23 Terminal Functions — By Ball Number (Part 1 of 22)

Ball Number	Signal Name
A1	VSS
A2	DDRDQM0
A3	DDRDQS0N
A4	DDRD06
A5	VSS
A6	DDRD12
A7	DDRD22
A8	VSS
A9	DDRD21
A10	DDRDQS3P
A11	DDRCB07
A12	VSS
A13	DDRDQS8N
A14	VSS
A15	DDRA06
A16	DDRA07
A17	VSS
A18	DDRBA0
A19	DDRBA1
A20	DDRCLKOUTP0
A21	DDRDQS4P
A22	VSS
A23	DDRDQM5
A24	DDRDQS5P
A25	VSS
A26	DDRD49
A27	DDRDQM6
A28	DDRDQM7
A29	DDRD56
A30	VSS
B1	DDRCLKP
B2	DDRD01
B3	DDRDQS0P
B4	DDRD05
B5	DDRDQS1P
B6	DDRD13
B7	DDRD20
B8	DDRDQS2P
B9	DDRD23
B10	DDRDQS3N
B11	DDRCB06
B12	DDRCB05

Table 2-23 Terminal Functions — By Ball Number (Part 2 of 22)

Ball Number	Signal Name
B13	DDRDQS8P
B14	DDRCLKOUTN1
B15	DDRA05
B16	DDRA01
B17	DDRA03
B18	DDRBA2
B19	$\overline{\text{DDRWE}}$
B20	DDRCLKOUTN0
B21	DDRDQS4N
B22	DDRD34
B23	DDRD40
B24	DDRDQS5N
B25	DDRDQS6P
B26	DDRD48
B27	DDRD50
B28	DDRD59
B29	DDRD57
B30	DDRDQS7N
C1	DDRCLKN
C2	VSS
C3	DDRD00
C4	VSS
C5	DDRDQS1N
C6	DDRD14
C7	VSS
C8	DDRDQS2N
C9	DDRD28
C10	VSS
C11	DDRD30
C12	DDRCB04
C13	VSS
C14	DDRCLKOUTP1
C15	DDRA10
C16	VSS
C17	DDRA02
C18	DDRODT0
C19	VSS
C20	$\overline{\text{DDRRESET}}$
C21	VSS
C22	DDRD35
C23	DDRD43
C24	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 3 of 22)

Ball Number	Signal Name
C25	DDRDQS6N
C26	DDRD52
C27	DDRD51
C28	DDRD58
C29	VSS
C30	DDRDQS7P
D1	DDRSLRATE1
D2	RSV06
D3	DDRD04
D4	DDRD07
D5	DDRDQM1
D6	DDRD15
D7	DDRD16
D8	DDRD18
D9	DDRD27
D10	DDRD25
D11	DDRD31
D12	DDRDQM8
D13	DDRCB03
D14	DDRCKE1
D15	DDRA11
D16	DDRA08
D17	DDRA00
D18	DDRODT1
D19	$\overline{\text{DDRRAS}}$
D20	DDRCKE0
D21	DDRD33
D22	DDRD36
D23	DDRD41
D24	DDRD42
D25	DDRD53
D26	DDRD54
D27	DDRD60
D28	DDRD63
D29	DDRD61
D30	DDRD62
E1	DDRSLRATE0
E2	RSV07
E3	VSS
E4	DDRD03
E5	DDRD08
E6	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 4 of 22)

Ball Number	Signal Name
E7	DDRDQM2
E8	DDRD19
E9	VSS
E10	DDRD26
E11	DDRD29
E12	VSS
E13	DDRCB00
E14	DDRA14
E15	VSS
E16	DDRA09
E17	DDRA04
E18	VSS
E19	$\overline{\text{DDRCAS}}$
E20	VSS
E21	DDRDQM4
E22	DDRD38
E23	VSS
E24	DDRD44
E25	DDRD47
E26	DDRD55
E27	VSS
E28	MCMRXPMCLK
E29	MCMCLKP
E30	MCMCLKN
F1	RSV09
F2	BOOTCOMPLETE
F3	$\overline{\text{LRESET}}$
F4	VSS
F5	DDRD02
F6	DDRD09
F7	DDRD11
F8	DDRD17
F9	DDRD24
F10	DDRDQM3
F11	VSS
F12	DDRCB02
F13	DDRCB01
F14	DDRA15
F15	VREFHSTL
F16	DDRA12
F17	DDRA13
F18	$\overline{\text{DDRCE1}}$

Table 2-23 Terminal Functions — By Ball Number (Part 5 of 22)

Ball Number	Signal Name
F19	$\overline{\text{DDRCE0}}$
F20	DDRD32
F21	DDRD39
F22	DDRD37
F23	DDRD45
F24	DDRD46
F25	VSS
F26	MCMTXPMCLK
F27	MCMTXPMDAT
F28	MCMRXPMDAT
F29	MCMRXFLDAT
F30	MCMREFCLKOUTP
G1	$\overline{\text{RESETFULL}}$
G2	$\overline{\text{RESET}}$
G3	CORESEL0
G4	HOUT
G5	VSS
G6	DDRD10
G7	DVDD15
G8	VSS
G9	DVDD15
G10	VSS
G11	DVDD15
G12	VSS
G13	DVDD15
G14	VSS
G15	DVDD15
G16	VSS
G17	DVDD15
G18	VSS
G19	DVDD15
G20	VSS
G21	DVDD15
G22	VSS
G23	DVDD18
G24	VSS
G25	MCMTXFLCLK
G26	VCNTL1
G27	MCMTXFLDAT
G28	MCMRXFLCLK
G29	VCNTL2
G30	MCMREFCLKOUTN

Table 2-23 Terminal Functions — By Ball Number (Part 6 of 22)

Ball Number	Signal Name
H1	$\overline{\text{LRESETNMIEN}}$
H2	RSV08
H3	CORESEL2
H4	$\overline{\text{RESETSTAT}}$
H5	RSV21
H6	RSV28
H7	VSS
H8	DVDD15
H9	VSS
H10	DVDD15
H11	VSS
H12	DVDD15
H13	VSS
H14	DVDD15
H15	VSS
H16	DVDD15
H17	VSS
H18	DVDD15
H19	VSS
H20	DVDD15
H21	VSS
H22	DVDD18
H23	VSS
H24	DVDD18
H25	RSV12
H26	VSS
H27	RSV30
H28	VCNTL0
H29	VCNTL3
H30	VSS
J1	RSV03
J2	$\overline{\text{NMI}}$
J3	VSS
J4	CORESEL1
J5	AVDDA2
J6	VPP
J7	DVDD18
J8	VSS
J9	DVDD18
J10	VSS
J11	DVDD15
J12	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 7 of 22)

Ball Number	Signal Name
J13	DVDD15
J14	VSS
J15	DVDD15
J16	VSS
J17	DVDD15
J18	VSS
J19	DVDD15
J20	VSS
J21	DVDD15
J22	VSS
J23	DVDD18
J24	RSV13
J25	VSS
J26	MCMTXP0
J27	VSS
J28	RSV29
J29	VSS
J30	MCMRXN0
K1	$\overline{\text{EMIFBE0}}$
K2	VSS
K3	PACLKSEL
K4	PTV15
K5	VSS
K6	DVDD18
K7	VSS
K8	CVDD
K9	VSS
K10	CVDD
K11	VSS
K12	CVDD
K13	VSS
K14	CVDD
K15	VSS
K16	CVDD
K17	VSS
K18	CVDD
K19	VSS
K20	CVDD
K21	VSS
K22	CVDD
K23	VDDT1
K24	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 8 of 22)

Ball Number	Signal Name
K25	VDDT1
K26	MCMTXN0
K27	MCMTXN1
K28	VSS
K29	MCMRXN1
K30	MCMRXP0
L1	EMIFA00
L2	$\overline{\text{EMIFBE1}}$
L3	$\overline{\text{EMIFOE}}$
L4	VSS
L5	DVDD18
L6	VSS
L7	DVDD18
L8	VSS
L9	CVDD1
L10	VSS
L11	CVDD
L12	VSS
L13	CVDD
L14	VSS
L15	CVDD
L16	VSS
L17	CVDD
L18	VSS
L19	CVDD
L20	VSS
L21	CVDD1
L22	VSS
L23	VSS
L24	VDDT1
L25	RSV14
L26	VSS
L27	MCMTXP1
L28	VSS
L29	MCMRXP1
L30	VSS
M1	VSS
M2	EMIFA02
M3	EMIFWAIT1
M4	VSS
M5	VSS
M6	DVDD18

Table 2-23 Terminal Functions — By Ball Number (Part 9 of 22)

Ball Number	Signal Name
M7	VSS
M8	CVDD
M9	VSS
M10	CVDD1
M11	VSS
M12	CVDD
M13	VSS
M14	CVDD
M15	VSS
M16	CVDD
M17	VSS
M18	CVDD
M19	VSS
M20	CVDD1
M21	VSS
M22	CVDD
M23	VDDT1
M24	VSS
M25	VSS
M26	MCMTXN3
M27	VSS
M28	VSS
M29	VSS
M30	MCMRXP3
N1	EMIFA12
N2	VSS
N3	EMIFA03
N4	EMIFWAIT0
N5	$\overline{\text{EMIFCE2}}$
N6	VSS
N7	DVDD18
N8	VSS
N9	CVDD1
N10	VSS
N11	CVDD
N12	VSS
N13	CVDD
N14	VSS
N15	CVDD
N16	VSS
N17	CVDD
N18	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 10 of 22)

Ball Number	Signal Name
N19	CVDD
N20	VSS
N21	CVDD1
N22	VSS
N23	VSS
N24	VDDR_1
N25	VDDT1
N26	MCMTXP3
N27	MCMTXP2
N28	VSS
N29	MCMRXN2
N30	MCMRXN3
P1	EMIFA13
P2	EMIFA11
P3	EMIFA04
P4	VSS
P5	$\overline{\text{EMIFCE1}}$
P6	DVDD18
P7	VSS
P8	CVDD
P9	VSS
P10	CVDD
P11	VSS
P12	CVDD
P13	VSS
P14	CVDD
P15	VSS
P16	CVDD
P17	VSS
P18	CVDD
P19	VSS
P20	CVDD
P21	VSS
P22	CVDD
P23	VDDT1
P24	VSS
P25	VSS
P26	VSS
P27	MCMTXN2
P28	VSS
P29	MCMRXP2
P30	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 11 of 22)

Ball Number	Signal Name
R1	VSS
R2	EMIFA14
R3	EMIFA08
R4	EMIFA07
R5	$\overline{\text{EMIFCE0}}$
R6	VSS
R7	DVDD18
R8	VSS
R9	CVDD
R10	VSS
R11	CVDD
R12	VSS
R13	CVDD
R14	VSS
R15	CVDD
R16	VSS
R17	CVDD
R18	VSS
R19	CVDD
R20	VSS
R21	CVDD
R22	VSS
R23	VSS
R24	VDDT1
R25	RSV27
R26	VDDT1
R27	VSS
R28	AIFTXP1
R29	VSS
R30	AIFRXN1
T1	EMIFA19
T2	EMIFA15
T3	VSS
T4	EMIFA05
T5	$\overline{\text{EMIFWE}}$
T6	DVDD18
T7	VSS
T8	CVDD
T9	VSS
T10	CVDD
T11	VSS
T12	CVDD

Table 2-23 Terminal Functions — By Ball Number (Part 12 of 22)

Ball Number	Signal Name
T13	VSS
T14	CVDD
T15	VSS
T16	CVDD
T17	VSS
T18	CVDD
T19	VSS
T20	CVDD
T21	VSS
T22	CVDD
T23	VDDT1
T24	VSS
T25	VDDT1
T26	VSS
T27	AIFTXN0
T28	AIFTXN1
T29	AIFRXN0
T30	AIFRXP1
U1	EMIFA23
U2	VSS
U3	EMIFA16
U4	EMIFA06
U5	EMIFRNW
U6	VSS
U7	DVDD18
U8	VSS
U9	CVDD
U10	VSS
U11	CVDD
U12	VSS
U13	CVDD
U14	VSS
U15	CVDD
U16	VSS
U17	CVDD
U18	VSS
U19	CVDD
U20	VSS
U21	CVDD
U22	VSS
U23	VSS
U24	VDDR_6

Table 2-23 Terminal Functions — By Ball Number (Part 13 of 22)

Ball Number	Signal Name
U25	VSS
U26	VDDT1
U27	AIFTXP0
U28	VSS
U29	AIFRXP0
U30	VSS
V1	EMIFD00
V2	EMIFA21
V3	EMIFA17
V4	VSS
V5	$\overline{\text{EMIFCE3}}$
V6	DVDD18
V7	VSS
V8	CVDD
V9	VSS
V10	CVDD
V11	VSS
V12	CVDD
V13	VSS
V14	CVDD
V15	VSS
V16	CVDD
V17	VSS
V18	CVDD
V19	VSS
V20	CVDD1
V21	VSS
V22	CVDD
V23	VDDT1
V24	VSS
V25	RSV26
V26	VSS
V27	VSS
V28	AIFTXP3
V29	VSS
V30	AIFRXP3
W1	VSS
W2	EMIFD03
W3	EMIFA20
W4	EMIFA09
W5	EMIFA01
W6	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 14 of 22)

Ball Number	Signal Name
W7	DVDD18
W8	VSS
W9	CVDD
W10	VSS
W11	CVDD
W12	VSS
W13	CVDD
W14	VSS
W15	CVDD
W16	VSS
W17	CVDD
W18	VSS
W19	CVDD
W20	VSS
W21	CVDD1
W22	VSS
W23	VSS
W24	VDDT1
W25	VSS
W26	VDDT1
W27	AIFTXP2
W28	AIFTXN3
W29	AIFRXP2
W30	AIFRXN3
Y1	EMIFD05
Y2	EMIFD02
Y3	VSS
Y4	EMIFA18
Y5	EMIFA10
Y6	DVDD18
Y7	VSS
Y8	CVDD
Y9	VSS
Y10	CVDD1
Y11	VSS
Y12	CVDD
Y13	VSS
Y14	CVDD
Y15	VSS
Y16	CVDD
Y17	VSS
Y18	CVDD

Table 2-23 Terminal Functions — By Ball Number (Part 15 of 22)

Ball Number	Signal Name
Y19	VSS
Y20	CVDD1
Y21	VSS
Y22	CVDD
Y23	VDDT1
Y24	VSS
Y25	VDDT1
Y26	VSS
Y27	AIFTXN2
Y28	VSS
Y29	AIFRXN2
Y30	VSS
AA1	EMIFD07
AA2	VSS
AA3	EMIFD04
AA4	EMIFD01
AA5	EMIFA22
AA6	VSS
AA7	DVDD18
AA8	VSS
AA9	CVDD1
AA10	VSS
AA11	CVDD
AA12	VSS
AA13	CVDD
AA14	VSS
AA15	CVDD
AA16	VSS
AA17	CVDD
AA18	VSS
AA19	CVDD
AA20	VSS
AA21	CVDD
AA22	VSS
AA23	VSS
AA24	VDDR_5
AA25	VSS
AA26	VDDT1
AA27	VSS
AA28	RSV36
AA29	VSS
AA30	RSV31

Table 2-23 Terminal Functions — By Ball Number (Part 16 of 22)

Ball Number	Signal Name
AB1	EMIFD06
AB2	EMIFD08
AB3	EMIFD09
AB4	EMIFD12
AB5	VSS
AB6	DVDD18
AB7	VSS
AB8	CVDD
AB9	VSS
AB10	CVDD1
AB11	VSS
AB12	CVDD
AB13	VSS
AB14	CVDD
AB15	VSS
AB16	CVDD
AB17	VSS
AB18	CVDD
AB19	VSS
AB20	CVDD
AB21	VSS
AB22	DVDD18
AB23	VDDT1
AB24	VSS
AB25	AVDDA1
AB26	VSS
AB27	RSV37
AB28	RSV35
AB29	RSV33
AB30	RSV32
AC1	VSS
AC2	EMIFD10
AC3	EMIFD14
AC4	EMIFD15
AC5	VDDR_3
AC6	VSS
AC7	RSV11
AC8	VSS
AC9	CVDD
AC10	VSS
AC11	VDDT2
AC12	VSS

Table 2-23 Terminal Functions — By Ball Number (Part 17 of 22)

Ball Number	Signal Name
AC13	DVDD18
AC14	VSS
AC15	DVDD18
AC16	VSS
AC17	DVDD18
AC18	VSS
AC19	DVDD18
AC20	VSS
AC21	DVDD18
AC22	VSS
AC23	DVDD18
AC24	VSS
AC25	ALTCORECLKP
AC26	CORECLKSEL
AC27	RSV38
AC28	VSS
AC29	RSV34
AC30	VSS
AD1	EMIFD11
AD2	EMIFD13
AD3	VSS
AD4	RSV17
AD5	RSV16
AD6	RSV22
AD7	RSV10
AD8	VDDT2
AD9	VSS
AD10	VDDT2
AD11	VSS
AD12	MDIO
AD13	VSS
AD14	AVDDA3
AD15	VSS
AD16	DVDD18
AD17	VSS
AD18	DVDD18
AD19	VSS
AD20	DVDD18
AD21	VSS
AD22	DVDD18
AD23	VSS
AD24	RSV05

Table 2-23 Terminal Functions — By Ball Number (Part 18 of 22)

Ball Number	Signal Name
AD25	ALTCORECLKN
AD26	TDO
AD27	VSS
AD28	SYSCLKOUT
AD29	VSS
AD30	SYSCLKN
AE1	RSV15
AE2	VSS
AE3	SGMII0TXN
AE4	SGMII0TXP
AE5	VDDR_4
AE6	VSS
AE7	VDDT2
AE8	VSS
AE9	VDDT2
AE10	VSS
AE11	VDDT2
AE12	VSS
AE13	EXTFRAMEEVENT
AE14	VSS
AE15	DVDD18
AE16	VSS
AE17	SDA
AE18	VSS
AE19	GPIO21
AE20	GPIO27
AE21	RSV01
AE22	$\overline{\text{POR}}$
AE23	RSV20
AE24	RSV04
AE25	TIMI1
AE26	TMS
AE27	$\overline{\text{TRST}}$
AE28	TCK
AE29	RP1CLKN
AE30	SYSCLKP
AF1	VSS
AF2	SGMII1TXN
AF3	SGMII1TXP
AF4	VSS
AF5	VSS
AF6	VDDT2

Table 2-23 Terminal Functions — By Ball Number (Part 19 of 22)

Ball Number	Signal Name
AF7	VSS
AF8	VDDT2
AF9	VDDR_2
AF10	VDDT2
AF11	VSS
AF12	MDCLK
AF13	USIMIO
AF14	SPIDOUT
AF15	SPISCS1
AF16	UART1RXD
AF17	SCL
AF18	GPIO10
AF18	BOOTMODE09 †
AF19	GPIO01
AF19	BOOTMODE00 †
AF20	GPIO25
AF21	GPIO18
AF22	GPIO29
AF23	GPIO28
AF24	GPIO31
AF25	TIMO1
AF26	TDI
AF27	EMU03
AF28	EMU00
AF29	RADSYNC
AF30	RP1CLKP
AG1	SGMII0RXN
AG2	SGMII0RXP
AG3	VSS
AG4	RIOTXN3
AG5	RIOTXP3
AG6	VSS
AG7	RIOTXN1
AG8	RIOTXP1
AG9	VSS
AG10	PCIETXP0
AG11	PCIETXN0
AG12	VSS
AG13	USIMRST
AG14	SPIDIN
AG15	SPISCS0
AG16	UART1TXD

Table 2-23 Terminal Functions — By Ball Number (Part 20 of 22)

Ball Number	Signal Name
AG17	UART0RXD
AG18	GPIO09
AG18	BOOTMODE08 †
AG19	GPIO02
AG19	BOOTMODE01 †
AG20	GPIO13
AG20	BOOTMODE12 †
AG21	GPIO19
AG22	VSS
AG23	GPIO16
AG23	BOOTMODE13 †
AG24	GPIO30
AG25	VSS
AG26	EMU11
AG27	EMU04
AG28	EMU01
AG29	VSS
AG30	RP1FBP
AH1	VSS
AH2	SGMII1RXN
AH3	SGMII1RXP
AH4	VSS
AH5	RIOTXN2
AH6	RIOTXP2
AH7	VSS
AH8	PCIETXP1
AH9	PCIETXN1
AH10	VSS
AH11	RSV25
AH12	RSV24
AH13	USIMCLK
AH14	SPICLK
AH15	SPISCS4
AH16	UART0TXD
AH17	VSS
AH18	GPIO08
AH18	BOOTMODE07 †
AH19	GPIO03
AH19	BOOTMODE02 †
AH20	GPIO15
AH20	PCIESSMODE1 †
AH21	GPIO26

Table 2-23 Terminal Functions — By Ball Number (Part 21 of 22)

Ball Number	Signal Name
AH22	GPIO22
AH23	GPIO20
AH24	GPIO17
AH25	TIMO0
AH26	EMU09
AH27	VSS
AH28	EMU05
AH29	EMU02
AH30	RP1FBN
AJ1	RIORXN3
AJ2	RIORXP3
AJ3	VSS
AJ4	RIORXP1
AJ5	RIORXN1
AJ6	VSS
AJ7	RIOTXP0
AJ8	RIOTXN0
AJ9	VSS
AJ10	PCIERXN1
AJ11	PCIERXP1
AJ12	VSS
AJ13	PCIECLKN
AJ14	PASSCLKP
AJ15	SPISCS3
AJ16	UART1RTS
AJ17	UART0CTS
AJ18	GPIO07
AJ18	BOOTMODE06 †
AJ19	GPIO05
AJ19	BOOTMODE04 †
AJ20	GPIO00
AJ20	LENDIAN †
AJ21	GPIO12
AJ21	BOOTMODE11 †
AJ22	GPIO24
AJ23	GPIO23
AJ24	EMU18
AJ25	TIMI0
AJ25	PCIESSEN †
AJ26	EMU14
AJ27	EMU12
AJ28	EMU08

Table 2-23 Terminal Functions — By Ball Number (Part 22 of 22)

Ball Number	Signal Name
AJ29	EMU06
AJ30	PHYSYNC
AK1	VSS
AK2	RIORXN2
AK3	RIORXP2
AK4	VSS
AK5	RIORXP0
AK6	RIORXN0
AK7	VSS
AK8	PCIERXP0
AK9	PCIERXN0
AK10	VSS
AK11	SRIOSGMIICLKN
AK12	SRIOSGMIICLKP
AK13	PCIECLKP
AK14	PASSCLKN
AK15	SPISCS2
AK16	UART1CTS
AK17	UART0RTS
AK18	GPIO06
AK18	BOOTMODE05
AK19	GPIO04
AK19	BOOTMODE03 †
AK20	VSS
AK21	GPIO14
AK21	PCIESSMODE0 †
AK22	GPIO11
AK22	BOOTMODE10 †
AK23	VSS
AK24	EMU17
AK25	EMU16
AK26	EMU15
AK27	EMU13
AK28	EMU10
AK29	EMU07
AK30	VSS
End of Table 2-23	

2.12 Development

2.12.1 Development Support

In case the customer would like to develop their own features and software on the TCI6612 device, TI offers an extensive line of development tools for the TMS320C6000™ DSP 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 C6000™ DSP-based applications:

- **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 C6000™ DSP multiprocessor system debug)

2.12.2 Device Support

2.12.2.1 Device and Development-Support Tool Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all DSP devices and support tools. Each DSP commercial family member has one of three prefixes: TMX, TMP, or TMS (e.g., TMX320CMH). Texas Instruments recommends two of three possible prefix designators for its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

- **TMX:** Experimental device that is not necessarily representative of the final device's electrical specifications
- **TMP:** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification
- **TMS:** Fully qualified production device

Support tool development evolutionary flow:

- **TMDX:** Development-support product that has not yet completed Texas Instruments internal qualification testing.
- **TMDS:** Fully qualified development-support product

TMX and TMP devices and TMDX development-support tools are shipped with the following disclaimer:

Developmental product is intended for internal evaluation purposes.

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

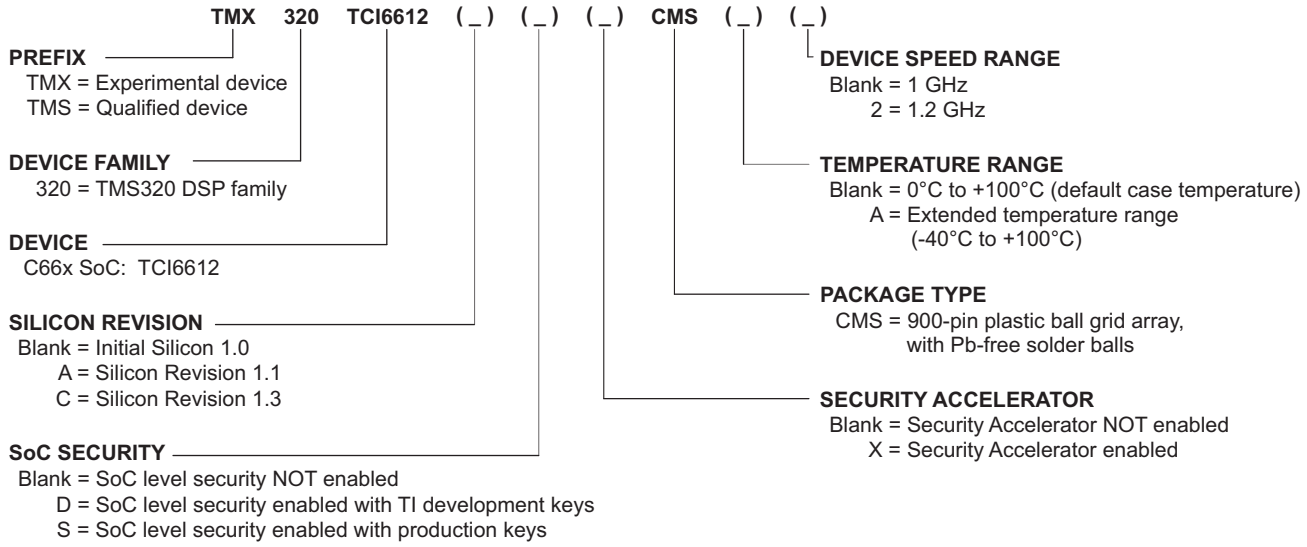
Predictions show that prototype devices (TMX or TMP) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices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

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, CYP), 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 TMS320TCI6612 in the CMS package type, see the TI website www.ti.com or contact your TI sales representative.

Figure 2-23 provides a legend for reading the complete device name for any C66x+™ DSP generation member.

Figure 2-23 C66x™ SoC Device Nomenclature (including the TMS320TCI6612 SoC)



2.13 Related Documentation from Texas Instruments

These documents describe the TMS320TCI6612 Communications Infrastructure KeyStone SoC. Copies of these documents are available on the Internet at www.ti.com

64-bit Timer (Timer 64) for KeyStone Devices User Guide	SPRUGV5
Antenna Interface 2 (AIF2) for KeyStone I Devices User Guide	SPRUGV7
Bit Coprocessor (BCP) for KeyStone Devices User Guide	SPRUGZ1
Bootloader for the C66x DSP User Guide	SPRUGY5
Chip Interrupt Controller (CIC) for KeyStone Devices User Guide	SPRUGW4
C66x CorePac User Guide	SPRUGW0
C66x CPU and Instruction Set Reference Guide	SPRUGH7
C66x DSP Cache User Guide	SPRUGY8
DDR3 Design Guide for KeyStone Devices	SPRABI1
Emulation and Trace Headers Technical Reference	SPRU655
Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide	SPRUGS5
External Memory Interface (EMIF16) for KeyStone Devices User Guide	SPRUGZ3
Fast Fourier Transform Coprocessor (FFTC) for KeyStone Devices User Guide	SPRUGS2
General Purpose Input/Output (GPIO) for KeyStone Devices User Guide	SPRUGV1
Gigabit Ethernet (GbE) Switch Subsystem (1 GB) for KeyStone Devices User Guide	SPRUGV9
Hardware Design Guide for KeyStone Devices	SPRABI2
HyperLink for KeyStone Devices User Guide	SPRUGW8
Inter Integrated Circuit (I^2C) for KeyStone Devices User Guide	SPRUGV3
Memory Protection Unit (MPU) for KeyStone Devices User Guide	SPRUGW5
Multicore Navigator for KeyStone Devices User Guide	SPRUGR9
Multicore Shared Memory Controller (MSMC) for KeyStone I Devices User Guide	SPRUGW7
Packet Accelerator (PA) for KeyStone Devices User Guide	SPRUGS4
Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User Guide	SPRUGS6
Phase Locked Loop (PLL) for KeyStone Devices User Guide	SPRUGV2
Power Management for KeyStone Devices	SPRABH0
Power Sleep Controller (PSC) for KeyStone Devices User Guide	SPRUGV4
Receive Accelerator (RAC) for KeyStone Devices User Guide	SPRUGY9
Security Addendum for KeyStone I Devices (1)	SPRABS3
Serial Peripheral Interface (SPI) for KeyStone Devices User Guide	SPRUGP2
Serial RapidIO (SRIO) for KeyStone Devices User Guide	SPRUGW1
Transmit Accelerator (TAC) for KeyStone Devices User Guide	SPRUGZ4
Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User Guide	SPRUGS0
Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User Guide	SPRUGP1
Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems	SPRA387
Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs	SPRA753
Using IBIS Models for Timing Analysis	SPRA839
Viterbi Coprocessor (VCP2) for KeyStone Devices User Guide	SPRUGV6

1 Contact a TI sales office to obtain this document.

3 Device Configuration

On the TMS320TCI6612 device, 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.

3.1 Device Configuration at Device Reset

Table 3-1 describes the device configuration pins. The logic level 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.

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 3.4 "Pullup/Pulldown Resistors" on page 95.

Table 3-1 TMS320TCI6612 Device Configuration Pins

Configuration Pin	Pin No.	IPD/IPU (1)	Functional Description
LENDIAN(1) (2)	AJ20	IPU	Device endian mode (LENDIAN) 0 = Device operates in big endian mode 1 = Device operates in little endian mode
BOOTMODE[13:0](1) (2)	AF19,AG19,AH19, AK19,AJ19,AK18, AJ18,AH18,AG18, AF18,AK22,AJ21, AG20, AG23	IPD	Method of boot See "Boot Modes Supported and PLL Settings" on page 30 for more details. See the Bootloader for the C66x DSP User Guide in "Related Documentation from Texas Instruments" on page 75 for detailed information on boot configuration
PCIESSMODE[1:0] (1) (2)	AK21, AH20	IPD	PCIe subsystem mode selection 00 = PCIe in end point mode 01 = PCIe legacy end point (no support for MSI) 10 = PCIe in root complex mode 11 = Reserved
PCIESSEN (1) (2)	AJ23	IPD	PCIe subsystem enable/disable 0 = PCIE Subsystem is disabled 1 = PCIE Subsystem is enabled
CORECLKSEL(1)	AC26	IPD	Core clock select 0 = SYSCLK is used as the input to Main PLL 1 = ALTCORECLK is used as the input to Main PLL
PACLKSEL(1)	AD23	IPD	Packet accelerator subsystem clock select 0 = SYSCLK / ALTCORECLK (controlled by CORECLKSEL pin) is used as the input to PA_SS PLL 1 = PASSCLK is used as the input to PASS PLL
End of Table 3-1			

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 3.4 "Pullup/Pulldown Resistors" on page 95.

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

3.2 Peripheral Selection After Device Reset

Several of the peripherals on the TMS320TCI6612 are controlled by the Power Sleep Controller (PSC). By default, the PCIe, SRIO, HyperLink, RAC, TAC, FFTC, AIF2, TCP3d, and VCP 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 will automatically enable 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 *Power Sleep Controller (PSC) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

3.3 Device State Control Registers

The TMS320TCI6612 device has a set of registers that are used to control the status of its peripherals. These registers are shown in Table 3-2.

Table 3-2 Device State Control Registers (Part 1 of 4)

Address Start	Address End	Size	Field	Description
0x02620000	0x02620007	8B	Reserved	
0x02620008	0x02620017	16B	Reserved	
0x02620018	0x0262001B	4B	JTAGID	See section 3.3.3
0x0262001C	0x0262001F	4B	Reserved	
0x02620020	0x02620023	4B	DEVSTAT	See section 3.3.1
0x02620024	0x02620037	20B	Reserved	
0x02620038	0x0262003B	4B	KICK0	See section 3.3.4
0x0262003C	0x0262003F	4B	KICK1	
0x02620040	0x02620043	4B	SoC_BOOT_ADDR0	The boot address for C66x SoC CorePac0
0x02620044	0x02620047	4B	SoC_BOOT_ADDR1	The boot address for C66x SoC CorePac1
0x02620048	0x0262004B	4B	Reserved	Reserved
0x0262004C	0x0262004F	4B	Reserved	Reserved
0x02620050	0x02620053	4B	Reserved	
0x02620054	0x02620057	4B	Reserved	
0x02620058	0x0262005B	4B	Reserved	
0x0262005C	0x0262005F	4B	Reserved	
0x02620060	0x02620063	4B	ARM_BOOT_ADDR	The boot address for ARM
0x02620064	0x0262010F	172B	Reserved	
0x02620110	0x02620117	8B	MACID	See section 8.20
0x02620118	0x0262012F	24B	Reserved	
0x02620130	0x02620133	4B	LRSTNMIPINSTAT_CLR	See section 3.3.6
0x02620134	0x02620137	4B	RESET_STAT_CLR	See section 3.3.8
0x02620138	0x0262013B	4B	Reserved	
0x0262013C	0x0262013F	4B	BOOTCOMPLETE	See section 3.3.9
0x02620140	0x02620143	4B	Reserved	
0x02620144	0x02620147	4B	RESET_STAT	See section 3.3.7
0x02620148	0x0262014B	4B	LRSTNMIPINSTAT	See section 3.3.5
0x0262014C	0x0262014F	4B	DEVCFG	See section 3.3.2
0x02620150	0x02620153	4B	PWRSTATECTL	See section 3.3.10

Table 3-2 Device State Control Registers (Part 2 of 4)

Address Start	Address End	Size	Field	Description
0x02620154	0x02620157	4B	SRIO_SERDES_STS	See 2.13 "Related Documentation from Texas Instruments" on page 75
0x02620158	0x0262015B	4B	SGMII_SERDES_STS	
0x0262015C	0x0262015F	4B	PCIE_SERDES_STS	
0x02620160	0x02620160	4B	HYPERLINK_SERDES_STS	
0x02620164	0x02620167	4B	AIF2_A_SERDES_STS	
0x02620168	0x0262016B	4B	AIF2_B_SERDES_STS	
0x0262016C	0x0262017F	20B	Reserved	
0x02620180	0x02620183	4B	Reserved	
0x02620184	0x0262018F	12B	Reserved	
0x02620190	0x02620193	4B	Reserved	
0x02620194	0x02620197	4B	Reserved	
0x02620198	0x0262019B	4B	Reserved	
0x0262019C	0x0262019F	4B	Reserved	
0x026201A0	0x026201A3	4B	Reserved	
0x026201A4	0x026201A7	4B	Reserved	
0x026201A8	0x026201AB	4B	Reserved	
0x026201AC	0x026201AF	4B	Reserved	
0x026201B0	0x026201B3	4B	Reserved	
0x026201B4	0x026201B7	4B	Reserved	
0x026201B8	0x026201BB	4B	Reserved	
0x026201BC	0x026201BF	4B	Reserved	
0x026201C0	0x026201C3	4B	Reserved	
0x026201C4	0x026201C7	4B	Reserved	
0x026201C8	0x026201CB	4B	Reserved	
0x026201CC	0x026201CF	4B	Reserved	
0x026201D0	0x026201FF	48B	Reserved	
0x02620200	0x02620203	4B	NMIGR0	See section 3.3.11
0x02620204	0x02620207	4B	NMIGR1	
0x02620208	0x0262020B	4B	Reserved	
0x0262020C	0x0262020F	4B	Reserved	
0x02620210	0x02620213	4B	Reserved	
0x02620214	0x02620217	4B	Reserved	
0x02620218	0x0262021B	4B	Reserved	
0x0262021C	0x0262021F	4B	Reserved	
0x02620220	0x0262023F	32B	Reserved	
0x02620240	0x02620243	4B	IPCGR0	See section 3.3.12 (Note: These registers are NOT protected by the kicker mechanism)
0x02620244	0x02620247	4B	IPCGR1	
0x02620248	0x0262024B	4B	Reserved	
0x0262024C	0x0262024F	4B	Reserved	
0x02620250	0x02620253	4B	Reserved	
0x02620254	0x02620257	4B	Reserved	
0x02620258	0x0262025B	4B	Reserved	
0x0262025C	0x0262025F	4B	Reserved	
0x02620260	0x0262027B	28B	Reserved	

Table 3-2 Device State Control Registers (Part 3 of 4)

Address Start	Address End	Size	Field	Description
0x0262027C	0x0262027F	4B	IPCGRH	See section 3.3.14 (Note: This register is NOT protected by the kicker mechanism)
0x02620280	0x02620283	4B	IPCAR0	See section 3.3.13 (Note: These registers are NOT protected by the kicker mechanism)
0x02620284	0x02620287	4B	IPCAR1	
0x02620288	0x0262028B	4B	Reserved	
0x0262028C	0x0262028F	4B	Reserved	
0x02620290	0x02620293	4B	Reserved	
0x02620294	0x02620297	4B	Reserved	
0x02620298	0x0262029B	4B	Reserved	
0x0262029C	0x0262029F	4B	Reserved	
0x026202A0	0x026202BB	28B	Reserved	
0x026202BC	0x026202BF	4B	IPCARH	See section 3.3.15 (Note: This register is NOT protected by the kicker mechanism)
0x026202C0	0x026202FF	64B	Reserved	
0x02620300	0x02620303	4B	TINPSEL	See section 3.3.16
0x02620304	0x02620307	4B	TOUTPSEL	See section 3.3.17
0x02620308	0x0262030B	4B	RSTMUX0	See section 3.3.18
0x0262030C	0x0262030F	4B	RSTMUX1	
0x02620310	0x02620313	4B	Reserved	
0x02620314	0x02620317	4B	Reserved	
0x02620318	0x0262031B	4B	RSTMUX8	See section 3.3.18
0x0262031C	0x0262031F	4B	Reserved	
0x02620320	0x02620323	4B	Reserved	
0x02620324	0x02620327	4B	Reserved	
0x02620328	0x0262032B	4B	MAINPLLCTL0	See section 8.6 "Main PLL and the PLL Controller" on page 146
0x0262032C	0x0262032F	4B	MAINPLLCTL1	
0x02620330	0x02620333	4B	DDR3PLLCTL0	See section 8.7 "DDR3 PLL" on page 160
0x02620334	0x02620337	4B	DDR3PLLCTL1	
0x02620338	0x0262033B	4B	PASSPLLCTL0	See section 8.8 "PASS PLL" on page 163
0x0262033C	0x0262033F	4B	PASSPLLCTL1	
0x02620340	0x02620343	4B	SGMII_SERDES_CFGPLL	See 2.13 "Related Documentation from Texas Instruments" on page 75
0x02620344	0x02620347	4B	SGMII_SERDES_CFGRX0	
0x02620348	0x0262034B	4B	SGMII_SERDES_CFGTX0	
0x0262034C	0x0262034F	4B	SGMII_SERDES_CFGRX1	
0x02620350	0x02620353	4B	SGMII_SERDES_CFGTX1	
0x02620354	0x02620357	4B	Reserved	
0x02620358	0x0262035B	4B	PCIE_SERDES_CFGPLL	See 2.13 "Related Documentation from Texas Instruments" on page 75
0x0262035C	0x0262035F	4B	Reserved	
0x02620360	0x02620363	4B	SRIO_SERDES_CFGPLL	See 2.13 "Related Documentation from Texas Instruments" on page 75
0x02620364	0x02620367	4B	SRIO_SERDES_CFGRX0	
0x02620368	0x0262036B	4B	SRIO_SERDES_CFGTX0	
0x0262036C	0x0262036F	4B	SRIO_SERDES_CFGRX1	
0x02620370	0x02620373	4B	SRIO_SERDES_CFGTX1	
0x02620374	0x02620377	4B	SRIO_SERDES_CFGRX2	

Table 3-2 Device State Control Registers (Part 4 of 4)

Address Start	Address End	Size	Field	Description
0x02620378	0x0262037B	4B	SRIO_SERDES_CFGTX2	
0x0262037C	0x0262037F	4B	SRIO_SERDES_CFGRX3	
0x02620380	0x02620383	4B	SRIO_SERDES_CFGTX3	
0x02620384	0x02620387	4B	Reserved	
0x02620388	0x026203AF	28B	Reserved	
0x026203B0	0x026203B3	4B	Reserved	
0x026203B4	0x026203B7	4B	HYPERLINK_SERDES_CFGPLL	
0x026203B8	0x026203BB	4B	HYPERLINK_SERDES_CFGRX0	
0x026203BC	0x026203BF	4B	HYPERLINK_SERDES_CFGTX0	
0x026203C0	0x026203C3	4B	HYPERLINK_SERDES_CFGRX1	
0x026203C4	0x026203C7	4B	HYPERLINK_SERDES_CFGTX1	
0x026203C8	0x026203CB	4B	HYPERLINK_SERDES_CFGRX2	
0x026203CC	0x026203CF	4B	HYPERLINK_SERDES_CFGTX2	
0x026203D0	0x026203D3	4B	HYPERLINK_SERDES_CFGRX3	
0x026203D4	0x026203D7	4B	HYPERLINK_SERDES_CFGTX3	
0x026203D8	0x026203DB	4B	Reserved	
0x026203DC	0x026203F7	28B	Reserved	
0x026203F8	0x026203FB	4B	DEVSPEED	
0x026203FC	0x026203FF	4B	Reserved	
0x02620400	0x02620403	4B	PKTDMA_PRI_ALLOC	See section 4.4 "Bus Priorities" on page 107
0x02620404	0x02620467	100B	Reserved	
0x02620580	0x02620584	4B	ARM_CPRIORITY	See section 4.4.1 "ARM Priority" on page 107
End of Table 3-2				

3.3.1 Device Status (DEVSTAT) Register

The Device Status Register depicts the device configuration selected upon a power-on reset by either the $\overline{\text{POR}}$ or $\overline{\text{RESETFULL}}$ pin. Once set, these bits will remain set until a power-on reset. The Device Status Register is shown in Figure 3-1 and described in Table 3-3.

Figure 3-1 Device Status Register

31 19	18	17	16 15	14 1	0
Reserved	PACLKSEL	PCIESSEN	PCIESSMODE	BOOTMODE	LENDIAN
R-0		R-x	R/W-xx	R/W-xxxxxxxxxxxx	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 3-3 Device Status Register Field Descriptions

Bit	Field	Description
31-19	Reserved	Reserved. Read only, writes have no effect.
18	PACLKSEL	PA clock select to select the reference clock for PA Sub-System PLL 0 = Selects output of Main PLL MUX (SYSCLK vs. ALTCORECLK - depending on CORECLKSEL pin) 1 = Selects PASSCLKP/N
17	PCIESSEN	PCIe module enable 0 = PCIe module disabled 1 = PCIe module enabled
16-15	PCIESSMODE	PCIe mode selection pins 00b = PCIe in end-point mode 01b = PCIe in legacy end-point mode (no support for MSI) 10b = PCIe in root complex mode 11b = Reserved
14-1	BOOTMODE	Determines the bootmode configured for the device. For more information on bootmode, see Section 2.4 "Boot Modes Supported and PLL Settings" on page 30 and see the *Bootloader for the C66x DSP User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.
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)
End of Table 3-3		

3.3.2 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 3-2 and described in Table 3-4.

Figure 3-2 Device Configuration Register (DEVCFG)

31 1	0
Reserved	SYSCLKOUTEN
R-0	R/W-1

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

Table 3-4 Device Configuration Register Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved. Read only, writes have no effect.
0	SYSCLKOUTEN	SYSCLKOUT enable 0 = No clock output 1 = Clock output enabled (default)
End of Table 3-4		

3.3.3 JTAG ID (JTAGID) Register Description

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the device, the JTAG ID register resides at address location 0x02620018. The JTAG ID Register is shown in Figure 3-3 and described in Table 3-5.

Figure 3-3 JTAG ID (JTAGID) Register

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

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

Table 3-5 JTAG ID Register Field Descriptions

Bit	Field	Value	Description
31-28	VARIANT	xxxxb	Variant value. The value of this field depends on the silicon revision being used.
27-12	PART NUMBER	1011 1001 0110 0010b	Part Number for boundary scan
11-1	MANUFACTURER	0000 0010 111b	Manufacturer
0	LSB	1b	This bit is read as a 1 for TMS320TCI6612
End of Table 3-5			

3.3.4 Kicker Mechanism (KICK0 and KICK1) Register

The Bootcfg module contains a kicker mechanism to prevent any spurious writes from changing any of the Bootcfg MMR 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). On the TCI6612, the exception to this are the IPC registers such as IPCGRx and IPCARx. These registers are not protected by the kicker mechanism. This mechanism requires two MMR writes to the KICK0 and KICK1 registers with exact data values before the kicker lock mechanism is un-locked. See Table 3-2 "Device State Control Registers" on page 77 for the address location. Once released then all the Bootcfg MMRs having write permissions are writable (the read-only MMRs are still read only). The first KICK0 data is 0x83e70b13. The second KICK1 data is 0x95a4f1e0. Writing any other data value to either of these kick MMRs will lock the kicker mechanism and block any writes to Bootcfg MMRs. In order to ensure protection to all Bootcfg MMRs, software must always re-lock the kicker mechanism after completing the MMR writes.

3.3.5 LRESETNMI PIN Status (LRSTNMIPINSTAT) Register

The LRSTNMIPINSTAT Register is created in Boot Configuration to latch the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL. The LRESETNMI PIN Status Register is shown in Figure 3-4 and described in Table 3-6.

Figure 3-4 LRESETNMI PIN Status Register (LRSTNMIPINSTAT)

31 20	19 18	17	16	15 4	3 2	1	0
Reserved	Reserved	NMI1	NMI0	Reserved	Reserved	LR1	LR0
R, +000000000000	R-0	R-0	R-0	R, +000000000000	R-0	R-0	R-0

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

Table 3-6 LRESETNMI PIN Status Register Field Descriptions

Bit	Field	Description
31-20	Reserved	Reserved
19	Reserved	Reserved
18	Reserved	Reserved
17	NMI1	CorePac1 in NMI
16	NMI0	CorePac0 in NMI
15-4	Reserved	Reserved
3	Reserved	Reserved
2	Reserved	Reserved
1	LR1	CorePac1 in local reset
0	LR0	CorePac0 in local reset
End of Table 3-6		

3.3.6 LRESETNMI PIN Status Clear (LRSTNMIPINSTAT_CLR) Register

The LRSTNMIPINSTAT_CLR Register is used to clear the status of $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ based on CORESEL[2:0]. The LRESETNMI PIN Status Clear Register is shown in Figure 3-5 and described in Table 3-7.

Figure 3-5 LRESETNMI PIN Status Clear Register (LRSTNMIPINSTAT_CLR)

31 20	19 18	17	16	15 4	3 2	1	0
Reserved	Reserved	NMI1	NMI0	Reserved	Reserved	LR1	LR0
R,+000000000000	R-0	WC,+0	WC,+0	R,+000000000000	R-0	WC,+0	WC,+0

Legend: R = Read only; -*n* = value after reset; WC = Write 1 to Clear

Table 3-7 LRESETNMI PIN Status Clear Register Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-20	Reserved	Reserved
19	Reserved	Reserved
18	Reserved	Reserved
17	NMI1	CorePac1 in NMI clear
16	NMI0	CorePac0 in NMI clear
15-4	Reserved	Reserved
3	Reserved	Reserved
2	Reserved	Reserved

Table 3-7 LRESETNMI PIN Status Clear Register Field Descriptions (Part 2 of 2)

Bit	Field	Description
1	LR1	CorePac1 in local reset clear
0	LR0	CorePac0 in local reset clear
End of Table 3-7		

3.3.7 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, if desired.

- **In case of local reset:** The LRx bits are written as 1 and GR bit is written as 0 only when the CorePac receives an local reset without receiving a global reset.
- **In case of global reset:** The LRx bits are written as 0 and GR bit is written as 1 only when a global reset is asserted.

The Reset Status Register is shown in Figure 3-6 and described in Table 3-8.

Figure 3-6 Reset Status Register (RESET_STAT)

31	30 ... 4	3 ... 2	1	0
GR	Reserved	Reserved	LR1	LR0
R, +1	R, + 000 0000 0000 0000 0000 0000 0000	R-0	R,+0	R,+0

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

Table 3-8 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	Reserved	Reserved.
2	Reserved	Reserved.
1	LR1	CorePac1 reset status 0 = CorePac1 has not received a local reset. 1 = CorePac1 received a local reset.
0	LR0	CorePac0 reset status 0 = CorePac0 has not received a local reset. 1 = CorePac0 received a local reset.
End of Table 3-8		

3.3.8 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 Figure 3-7 and described in Table 3-9.

Figure 3-7 Reset Status Clear Register (RESET_STAT_CLR)

31	30 4	3 2	1	0
GR	Reserved	Reserved	LR1	LR0
RW, +0	R, + 000 0000 0000 0000 0000 0000 0000	R-0	RW,+0	RW,+0

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

Table 3-9 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	Reserved	Reserved.
2	Reserved	Reserved.
1	LR1	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	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.
End of Table 3-9		

3.3.9 Boot Complete (BOOTCOMPLETE) Register

The BOOTCOMPLETE register controls the BOOTCOMPLETE pin status. The purpose is to indicate the completion of the ROM booting process. The Boot Complete Register is shown in Figure 3-8 and described in Table 3-10.

Figure 3-8 Boot Complete Register (BOOTCOMPLETE)

31 9	8	7 4	3 2	1	0
Reserved	BC_ARM	Reserved		BC1	BC0
R, + 0000 0000 0000 0000 0000 000	RW, +0	R, +0000 00		RW,+0	RW,+0

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

Table 3-10 Boot Complete Register Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-9	Reserved	Reserved
8	BC_ARM	ARM CorePac boot status 0 = ARM boot NOT complete 1 = ARM boot complete
7-2	Reserved	Reserved

Table 3-10 Boot Complete Register Field Descriptions (Part 2 of 2)

Bit	Field	Description
1	BC1	CorePac1 boot status 0 = CorePac1 boot NOT complete 1 = CorePac1 boot complete
0	BC0	CorePac0 boot status 0 = CorePac0 boot NOT complete 1 = CorePac0 boot complete
End of Table 3-10		

The BCx bit indicates the boot complete status of the corresponding core. All BCx bits will be 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.

Boot ROM code will be implemented such that each core will set its corresponding BCx bit immediately before branching to the predefined location in memory.

3.3.10 Power State Control (PWRSTATECTL) Register

The PWRSTATECTL Register is controlled by the software to indicate the power-saving mode. ROM code reads this register to differentiate between the various power saving modes. This register is cleared only by POR and will survive all other device resets. See the *Hardware Design Guide for KeyStone Devices* in"Related Documentation from Texas Instruments" on page 75 for more information. The Power State Control Register is shown in Figure 3-9 and described in Table 3-11.

Figure 3-9 Power State Control Register (PWRSTATECTL)

31	3	2	1	0
GENERAL_PURPOSE		HIBERNATION_MODE	HIBERNATION	STANDBY
RW, +0000 0000 0000 0000 0000 0000 0000 0		RW,+0	RW,+0	RW,+0

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

Table 3-11 Power State Control Register Field Descriptions

Bit	Field	Description
31-3	GENERAL_PURPOSE	Used to provide a start address for execution out of the hibernation modes. See the *Bootloader for the C66x DSP User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.
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
End of Table 3-11		

3.3.11 NMI Even Generation to CorePac (NMIGRx) Register

NMIGRx registers are used for generating NMI events to the corresponding CorePac. The TCI6612 has two NMIGRx registers (NMIGR0 and NMIGR1). The NMIGR0 register generates an NMI event to CorePac0 and the NMIGR1 register generates an NMI event to CorePac1. Writing a 1 to the NMIG field generates a NMI pulse. Writing a 0 has no effect and Reads return 0 and have no other effect. The NMI Even Generation to CorePac Register is shown in Figure 3-10 and described in Table 3-12.

Figure 3-10 NMI Generation Register (NMIGRx)

31	1	0
GENERAL_PURPOSE		NMIG
R, +0000 0000 0000 0000 0000 0000 0000 000		RW,+0

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

Table 3-12 NMI Generation Register Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	NMIG	NMI generation Reads return 0 Writes: 0 = No effect 1 = Creates NMI pulse to the corresponding CorePac — CorePac0 for NMIGR0, etc.
End of Table 3-12		

3.3.12 IPC Generation (IPCGRx) Registers

IPCGRx are the IPC interrupt generation registers to facilitate inter CorePac interrupts.

The TCI6612 has two IPCGRx registers (IPCGR0 and IPCGR1). These registers can be used by external hosts or CorePacs to generate interrupts to other CorePacs. A write of 1 to IPCG field of IPCGRx register will generate an interrupt pulse to CorePacx (0 <= x <= 1).

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 Generation Register is shown in Figure 3-11 and described in Table 3-13.

Figure 3-11 IPC Generation Registers (IPCGRx)

31	30	29	28	27 – 8	7	6	5	4	3 – 1	0
SRCS27	SRCS26	SRCS25	SRCS24	SRCS23 – SRCS4	SRCS3	SRCS2	SRCS1	SRCS0	Reserved	IPCG
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +000	RW +0

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

Table 3-13 IPC Generation Registers Field Descriptions

Bit	Field	Description
31-4	SRCSx	Source ID select 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	IPC generation Reads return 0. Writes: 0 = No effect 1 = Creates an Inter-SoC interrupt
End of Table 3-13		

3.3.13 IPC Acknowledgement (IPCARx) Registers

IPCARx are the IPC interrupt-acknowledgement registers to facilitate inter-CorePac core interrupts.

The TCI6612 has two IPCARx registers (IPCAR0 and IPCAR1). 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 Acknowledgement Register is shown in Figure 3-12 and described in Table 3-14.

Figure 3-12 IPC Acknowledgement Registers (IPCARx)

31	30	29	28	27 – 8	7	6	5	4	3 – 0
SRCC27	SRCC26	SRCC25	SRCC24	SRCC23 – SRCC4	SRCC3	SRCC2	SRCC1	SRCC0	Reserved
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +0000

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

Table 3-14 IPC Acknowledgement Registers Field Descriptions

Bit	Field	Description
31-4	SRCCx	Source ID control Reads return current value of internal register bit. Writes: 0 = No effect 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved
End of Table 3-14		

3.3.14 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 (CPU/6) cycles followed by a deassertion of four bootcfg clock cycles. Generating the pulse results in a pulse-blocking window that is eight CPU/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 CPU/6 cycle window -- the pulse blocking window. In order to generate back-to-back pulses, the back-to-back writes to the IPCGRH register must be greater than eight CPU/6 cycle window. The IPC Generation Host Register is shown in Figure 3-13 and described in Table 3-15.

Figure 3-13 IPC Generation Host Register (IPCGRH)

31	30	29	28	27 – 8	7	6	5	4	3 – 1	0
SRCS27	SRCS26	SRCS25	SRCS24	SRCS23 – SRCS4	SRCS3	SRCS2	SRCS1	SRCS0	Reserved	IPCG
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +000	RW +0

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

Table 3-15 IPC Generation Host Register Field Descriptions

Bit	Field	Description
31-4	SRCSx	Source ID select 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	IPC generation Reads return 0. Writes: 0 = No effect 1 = Creates an interrupt pulse on ARM (host interrupt/event output in HOUT pin)
End of Table 3-15		

3.3.15 IPC Acknowledgement Host (IPCARH) Register

IPCARH registers are provided to facilitate an ARM interrupt from one of the CorePacs. Use of IPCARH is the same as other IPCAR registers. The IPC Acknowledgement Host Register is shown in Figure 3-14 and described in Table 3-16.

Figure 3-14 IPC Acknowledgement Host Register (IPCARH)

31	30	29	28	27 – 8	7	6	5	4	3 – 0
SRCC27	SRCC26	SRCC25	SRCC24	SRCC23 – SRCC4	SRCC3	SRCC2	SRCC1	SRCC0	Reserved
RW +0	RW +0	RW +0	RW +0	RW +0 (per bit field)	RW +0	RW +0	RW +0	RW +0	R, +0000

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

Table 3-16 IPC Acknowledgement Host Register Field Descriptions

Bit	Field	Description
31-4	SRCCx	Source ID control Reads return current value of internal register bit. Writes: 0 = No effect 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved
End of Table 3-16		

3.3.16 Timer Input Selection Register (TINPSEL)

Timer input selection is handled within the control register TINPSEL. The Timer Input Selection Register is shown below.

Figure 3-15 Timer Input Selection Register (TINPSEL)

31							24
Reserved							
R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
23	22	21	20	19	18	17	16
TINPHSEL11	TINPLSEL11	TINPHSEL10	TINPLSEL10	TINPHSEL9	TINPLSEL9	TINPHSEL8	TINPLSEL8
R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
15	14	13	12	11	10	9	8
TINPHSEL7	TINPLSEL7	TINPHSEL6	TINPLSEL6	TINPHSEL5	TINPLSEL5	TINPHSEL4	TINPLSEL4
R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
7			4	3	2	1	0
Reserved				TINPHSEL1	TINPLSEL1	TINPHSEL0	TINPLSEL0
R/W-0				R/W-1	R/W-0	R/W-1	R/W-0

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

Table 3-17 Timer Input Selection Field Description (Part 1 of 2)

Bit	Field	Description
31-24	Reserved	Reserved
23	TINPHSEL11	Input select for TIMER11 high 0 = TIMI0 1 = TIMI1
22	TINPLSEL11	Input select for TIMER11 low 0 = TIMI0 1 = TIMI1
21	TINPHSEL10	Input select for TIMER10 high 0 = TIMI0 1 = TIMI1
20	TINPLSEL10	Input select for TIMER10 low 0 = TIMI0 1 = TIMI1

Table 3-17 Timer Input Selection Field Description (Part 2 of 2)

Bit	Field	Description
19	TINPHSEL9	Input select for TIMER9 high 0 = TIMI0 1 = TIMI1
18	TINPLSEL9	Input select for TIMER9 low 0 = TIMI0 1 = TIMI1
17	TINPHSEL8	Input select for TIMER8 high 0 = TIMI0 1 = TIMI1
16	TINPLSEL8	Input select for TIMER8 low 0 = TIMI0 1 = TIMI1
15	TINPHSEL7	Input select for TIMER7 high 0 = TIMI0 1 = TIMI1
14	TINPLSEL7	Input select for TIMER7 low 0 = TIMI0 1 = TIMI1
13	TINPHSEL6	Input select for TIMER6 high 0 = TIMI0 1 = TIMI1
12	TINPLSEL6	Input select for TIMER6 low 0 = TIMI0 1 = TIMI1
11	TINPHSEL5	Input select for TIMER5 high 0 = TIMI0 1 = TIMI1
10	TINPLSEL5	Input select for TIMER5 low 0 = TIMI0 1 = TIMI1
9	TINPHSEL4	Input select for TIMER4 high 0 = TIMI0 1 = TIMI1
8	TINPLSEL4	Input select for TIMER4 low 0 = TIMI0 1 = TIMI1
7-4	Reserved	Reserved
3	TINPHSEL1	Input select for TIMER1 high 0 = TIMI0 1 = TIMI1
2	TINPLSEL1	Input select for TIMER1 low 0 = TIMI0 1 = TIMI1
1	TINPHSEL0	Input select for TIMER0 high 0 = TIMI0 1 = TIMI1
0	TINPLSEL0	Input select for TIMER0 low 0 = TIMI0 1 = TIMI1
End of Table 3-17		

3.3.17 Timer Output Selection Register (TOUTPSEL)

The timer output selection is handled within the control register TOUTSEL. The Timer Output Selection Register is shown in Figure 3-16 and described in Table 3-18.

Figure 3-16 Timer Output Selection Register (TOUTPSEL)

31	10	9	5	4	0
Reserved		TOUTPSEL1		TOUTPSEL0	
R,+00000000000000000000000000		RW,+0001		RW,+0000	

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

Table 3-18 Timer Output Selection Field Description

Bit	Field	Description			
31-9	Reserved	Reserved			
9-5	TOUTPSEL1	Output select for TIMO1			
		00000 = TOUTL0	00110 = Reserved	01100 = TOUTL6	10010 = TOUTL9
		00001 = Reserved	00111 = Reserved	01101 = TOUTH6	10011 = TOUTH9
		00010 = TOUTL1	01000 = TOUTL4	01110 = TOUTL7	10100 = TOUTL10
		00011 = Reserved	01001 = TOUTH4	01111 = TOUTH7	10101 = TOUTH10
		00100 = Reserved	01010 = TOUTL5	10000 = TOUTL8	10110 = TOUTL11
		00101 = Reserved	01011 = TOUTH5	10001 = TOUTH8	10111 = TOUTH11
					Others = Reserved
4-0	TOUTPSEL0	Output select for TIMO0			
		00000: TOUTL0	00110: Reserved	01100: TOUTL6	10010: TOUTL9
		00001: Reserved	00111: Reserved	01101: TOUTH6	10011: TOUTH9
		00010: TOUTL1	01000: TOUTL4	01110: TOUTL7	10100: TOUTL10
		00011: Reserved	01001: TOUTH4	01111: TOUTH7	10101: TOUTH10
		00100: Reserved	01010: TOUTL5	10000: TOUTL8	10110: TOUTL11
		00101: Reserved	01011: TOUTH5	10001: TOUTH8	10111: TOUTH11
					Others: Reserved
End of Table 3-18					

3.3.18 Reset Mux (RSTMUXx) Register

The software controls the Reset Mux block through the reset multiplex registers using RSTMUX0 and RSTMUX1 for both of the CorePacs on the TCI6612. These registers are located in Bootcfg memory space. The Reset Mux Register is shown in Figure 3-17 and described in Table 3-19. RSTMUX8 controls the ARM's watchdog timer and has the same register definitions as the other RSTMUX registers. Because the ARM does not support local reset, the local reset in RSTMUX8 triggers a device reset.

Figure 3-17 Reset Mux Register (RSTMUX0 and RSTMUX1)

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 3-19 Reset Mux Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved
9	EVTSTATCLR	Clear for EVTSTAT field 0 = No effect 1 = Clears the EVTSTAT bit
8	Reserved	Reserved
7-5	DELAY	Delay between NMI and local reset 000b = 256 CPU/6 cycles delay between NMI & local reset, when OMODE = 100b 001b = 512 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 010b = 1024 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 011b = 2048 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 100b = 4096 CPU/6 cycles delay between NMI & local reset, when OMODE=100b (Default) 101b = 8192 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 110b = 16384 CPU/6 cycles delay between NMI & local reset, when OMODE=100b 111b = 32768 CPU/6 cycles delay between NMI & local reset, when OMODE=100b
4	EVTSTAT	Event Status for WD timer event 0 = No event received (Default) 1 = WD timer event received by Reset Mux block
3-1	OMODE	Output mode for reset mux block 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 CorePac 011b = WD timer event input to the reset mux block causes NMI input to CorePac 100b = WD timer event input to the reset mux block causes NMI input followed by local reset input to CorePac. The delay between NMI and local reset is set in DELAY bit field. 101b = WD timer event input to the reset mux block causes device reset 110b = Reserved 111b = Reserved
0	LOCK	Locks register fields 0 = Register fields are not locked (default) 1 = Register fields are locked until the next timer reset
End of Table 3-19		

3.3.19 Device Speed (DEVSPEED) Register

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

Figure 3-18 Device Speed Register (DEVSPEED)

31	23	22	0
DEVSPEED		Reserved	
R-n		R-n	

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

Table 3-20 Device Speed Register Field Descriptions

Bit	Field	Description
31-23	DEVSPEED	Indicates the speed of the device (read only) 0b0000 0000 0 = 800 MHz 0b0000 0000 1 = 1000 MHz 0b0000 0001 x = 1200 MHz 0b001x xxxx x = 1200 MHz 0b01xx xxxx x = 1000 MHz 0b1xxx xxxx x = 800 MHz
22-0	Reserved	Reserved. Read only
End of Table 3-20		

3.4 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 is 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 3-1), 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. Make 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 can still ensure 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 TMS320TCI6612 device, see Section 7.3 "Electrical Characteristics" on page 125.

To determine which pins on the device include internal pullup/pulldown resistors, see Table 2-16 "Terminal Functions — Power and Ground" on page 48.

4 System Interconnect

On the TMS320TCI6612 device, the C66x CorePacs, the ARM CorePac, the EDMA3 transfer controllers, and the system peripherals are interconnected through the TeraNet, which is a non-blocking switch fabric enabling fast and contention-free internal data movement. The TeraNet allows for low-latency, concurrent data transfers between master peripherals and slave peripherals. The TeraNet also allows for seamless arbitration between the system masters when accessing system slaves.

4.1 Internal Buses, Bridges, and Switch Fabrics

Two types of buses exist in the device: data buses and configuration buses. Some peripherals have both a data bus interface and a configuration bus interface, while others have only one type of interface. Furthermore, the bus interface width and speed varies from peripheral to peripheral.

Configuration buses are mainly used to access the register space of a peripheral and the data buses are used mainly for data transfers. However, in some cases, the configuration bus is also used to transfer data. For example, data is transferred to the VCP2 via its configuration bus. Similarly, the data bus can also be used to access the register space of a peripheral. For example, the DDR3 memory controller registers are accessed through their data bus interface.

The C66x CorePac, the ARM CorePac, 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 EDMA3 to perform transfers to and from them. Examples of masters include the EDMA3 traffic controllers, SRIO, and Gigabit Ethernet (GbE) Switch Subsystem. Examples of slaves include the SPI, UART, and I^2C.

The data switch fabric, known as the data TeraNet, is a high-throughput interconnect mainly used to move data across the system. The data TeraNet is further divided into two smaller TeraNets. One connects very high speed masters to slaves via 256-bit data buses running at a CPU/2 frequency. The other connects masters to slaves via 128-bit data buses running at a CPU/3 frequency. Peripherals that match the native bus width of the TeraNet they are connected to can connect directly to the data TeraNet. Other peripherals require a bridge.

The configuration switch fabric, also known as the configuration TeraNet, is mainly used to access peripheral registers. The configuration TeraNet connects the C66x CorePac and masters on the data switch fabric to slaves via 32-bit configuration buses running at a CPU/3 frequency. As with the data TeraNet, some peripherals require the use of a bridge to interface to the configuration TeraNet. For more information, see Section 4.2 "Switch Fabric Connection Matrices").

4.2 Switch Fabric Connection Matrices

The following tables list the various master to slave endpoint connections on the device.

Intersecting cells may contain one of the following:

- **Y** — There is a direct 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 4-1 Switch Fabric Connection Matrix Section 1 (Part 1 of 2)

Masters	Slaves																								
	CorePacx_SDMA (x=core number)	Boot_ROM,	SPI	PCIe_Slave	QM_Slave	HyperLink_Slave	MSMC_SES	MSMC_SMS	STM	TETB_System	TETBx (x=core number)	EMIF16	Coresight ETB	DDR_EMIF	VCP2	TCP3d	TAC_BE	RAC_FE	BCP_CFG	EDMA3CC0	EDMA3CC1	EDMA3CC2	EDMA3CC0_TC (0-1)	EDMA3CC1_TC (0-3)	EDMA3CC2_TC (0-3)	Semaphore
ARM_port0	-	-	-	-	-	-	-	-	-	-	-	-	-	75	-	-	-	-	-	-	-	-	-	-	-	-
ARM_port1	77	77	77	77	77	77, 8	77, 8	77, 8	77, 12	77, 12	77, 12	77	77	-	-	77	-	-	77	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12
BCP_Packet DMA	Y	-	-	-	Y	7	7	7	-	-	-	-	-	7	-	Y	-	-	-	-	-	-	-	-	-	-
BCP_DIO0	Y	-	-	Y	-	7	7	7	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-	-
BCP_DIO1	Y	-	-	Y	-	10	10	10	-	-	-	-	-	10	-	-	-	-	-	-	-	-	-	-	-	-
HyperLink_Master	1	1	1	1	1	-	Y	Y	-	-	-	1	-	Y	1	1	1	1	1	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12
EDMA3CC0_TC0_RD	3	3	3	3	-	Y	Y	Y	-	3, 12	-	3	-	Y	-	-	-	-	-	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-
EDMA3CC0_TC0_WR	3	-	3	3	-	Y	Y	Y	-	-	-	3	-	Y	-	-	-	-	-	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-
EDMA3CC0_TC1_RD	3	3	3	3	-	Y	Y	Y	-	3, 12	-	3	-	Y	-	-	-	-	-	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-
EDMA3CC0_TC1_WR	3	-	3	3	-	Y	Y	Y	-	-	-	3	-	Y	-	-	-	-	-	3, 12	3, 12	3, 12	3, 12	3, 12	3, 12	-
EDMA3CC1_TC0_RD	Y	Y	Y	Y	-	5	5	5	-	12	-	Y	-	5	-	-	-	Y	Y	12	12	12	12	12	12	12
EDMA3CC1_TC0_WR	Y	-	Y	Y	-	5	5	5	12	-	-	Y	-	5	-	-	-	Y	Y	12	12	12	12	12	12	12
EDMA3CC1_TC1_RD	Y	Y	Y	Y	Y	6	6	6	-	-	13	Y	-	6		-	-	Y	Y	13	13	13	13	13	13	-
EDMA3CC1_TC1_WR	Y	-	Y	Y	Y	6	6	6	-	-	-	Y	-	6	-	-	-	Y	Y	13	13	13	13	13	13	-
EDMA3CC1_TC2_RD	Y	Y	Y	Y	-	7	7	7	-	-	-	Y	Y	7	-	-	Y	-	-	14	14	14	14	14	14	-
EDMA3CC1_TC2_WR	Y	-	Y	Y	-	7	7	7	-	-	-	Y	-	7	-	-	Y	-	-	14	14	14	14	14	14	-
EDMA3CC1_TC3_RD	Y	Y	Y	Y	-	8	8	8	-	12	-	Y	-	8	-	-	Y	-	-	12	12	12	12	12	12	12
EDMA3CC1_TC3_WR	Y	-	Y	Y	-	8	8	8	12	-	-	Y	-	8	-	-	Y	-	-	12	12	12	12	12	12	12
EDMA3CC2_TC0_RD	Y	Y	Y	Y	-	9	9	9	-	12	-	Y	-	9	Y	Y	-	-	-	12	12	12	12	12	12	12
EDMA3CC2_TC0_WR	Y	-	Y	Y	-	9	9	9	12	-	-	Y	-	9	Y	Y	-	-	-	12	12	12	12	12	12	12
EDMA3CC2_TC1_RD	Y	Y	Y	Y	Y	10	10	10	-	-	13	Y	-	10	Y	Y	-	-	-	13	13	13	13	13	13	-
EDMA3CC2_TC1_WR	Y	-	Y	Y	Y	10	10	10	-	-	-	Y	-	10	Y	Y	-	-	-	13	13	13	13	13	13	
EDMA3CC2_TC2_RD	Y	Y	Y	Y	-	5	5	5	-	12	-	Y	-	5	Y	Y	-	-	-	12	12	12	12	12	12	12

Table 4-1 Switch Fabric Connection Matrix Section 1 (Part 2 of 2)

Masters	Slaves																								
	CorePacx_SDMA (x=core number)	Boot_ROM,	SPI	PCIe_Slave	QM_Slave	HyperLink_Slave	MSMC_SES	MSMC_SMS	STM	TETB_System	TETBx (x=core number)	EMIF16	Coresight ETB	DDR_EMIF	VCP2	TCP3d	TAC_BE	RAC_FE	BCP_CFG	EDMA3CC0	EDMA3CC1	EDMA3CC2	EDMA3CC0_TC (0-1)	EDMA3CC1_TC (0-3)	EDMA3CC2_TC (0-3)	Semaphore
EDMA3CC2_TC2_WR	Y	-	Y	Y	-	5	5	5	12	-	-	Y	-	5	Y	Y	-	-	-	12	12	12	12	12	12	12
EDMA3CC2_TC3_RD	Y	Y	Y	Y	-	6	6	6	-	-	-	Y	Y	6	Y	-	-	-	-	14	14	14	14	14	14	-
EDMA3CC2_TC3_WR	Y	-	Y	Y	-	6	6	6	-	-	-	Y	-	6	Y	-	-	-	-	14	14	14	14	14	14	-
SRIO_Packet DMA	Y	-	-	-	Y	5	5	5	-	-	-	Y	-	5	-	-	-	-	-	-	-	-	-	-	-	-
SRIO_Master	Y	-	Y	-	Y	10	10	10	12	12	12	Y	Y	10	Y	Y	Y	Y	Y	12	12	12	12	12	12	12
PCIe_Master	Y	-	Y	-	Y	10	10	10	12	12	12	Y	Y	10	Y	Y	Y	Y	Y	12	12	12	12	12	12	12
NETCP	Y	-	-	-	Y	10	10	10	-	-	-	-	-	10	-	-	-	-	-	-	-	-	-	-	-	-
MSMC_Data_Master	4	Y	Y	4	4	Y	-	-	4, 12	-	-	4	-	-	4	4	4	4	4	-	-	-	-	-	-	-
MSMC_EMIF	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
QM_Packet DMA	Y	-	-	-	Y	8	8	8	-	-	-	-	-	8	-	Y	-	-	-	-	-	-	-	-	-	-
QM_Second	Y	-	-	-	-	10	10	10	-	-	-	-	-	10	-	-	-	-	-	12	12	12	12	12	12	12
DAP_Master	Y	Y	Y	Y	Y	10	10	10	12	12	12	Y	Y	10	Y	Y	Y	Y	Y	12	12	12	12	12	12	12
FFTC_A	Y	-	-	-	Y	6	6	6	-	-	-	-	-	6	-	-	-	-	-	-	-	-	-	-	-	-
FFTC_B	Y	-	-	-	Y	9	9	9	-	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE0	Y	-	-	-	-	7	7	7	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE1	4	-	-	-	-	Y	Y	Y	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-
AIF	Y	-	-	-	Y	7	7	7	-	-	-	-	-	7	-	-	Y	Y	-	-	-	-	-	-	-	-
TAC_FE	Y	-	-	-	-	9	9	9	-	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y		-	-
EDMA3CC1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-
EDMA3CC2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-						Y	-
CorePacx_CFG (x = core number)	-	-	-	-	-	-	-	-	-	Y	Y	-	Y	-	-	-	-	-	-	Y	Y	Y	Y	Y	Y	Y
Tracer Master Port	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

End of Table 4-1

Table 4-2 Switch Fabric Connection Matrix Section 2 (Part 1 of 2)

Masters	Slaves																									
	QM_SS_CFG	Tracers	NETCP_CFG	SRIO_CFG	Timer	GPIO	I2C	SEC_CTL	SEC_Key_MGR(A/B)	Boot_CFG/Chip Level Registers	PSC	PLL_CTL	CIC(0 to 3)	MPUs	Debug_SS_CFG	SmartReflex_MMR	RAC_CFG	FFTC_CFG	TAC_CFG	TCP3d_CFG	VCP2_CFG	AIF_CFG	UART_CFG	USIM	OTP Memory	INTD
ARM_port0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ARM_port1	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12	77, 12
BCP_Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCP_DIO0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BCP_DIO1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HyperLink_Master	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12	1, 12
EDMA3CC0_TC0_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC0_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC1_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC1_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC2_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1_TC3_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC1_TC3_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC2_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC2_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA3CC2_TC3_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2_TC3_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIO_Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIO_Master	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
PCIe_Master	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
NETCP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MSMC_Data_Master	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MSMC_EMIF	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Packet DMA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Second	-	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4-2 Switch Fabric Connection Matrix Section 2 (Part 2 of 2)

Masters	Slaves																									
	QM_SS_CFG	Tracers	NETCP_CFG	SRIO_CFG	Timer	GPIO	I2C	SEC_CTL	SEC_Key_MGR(A/B)	Boot_CFG/Chip Level Registers	PSC	PLL_CTL	CIC(0 to 3)	MPUs	Debug_SS_CFG	SmartReflex_MMR	RAC_CFG	FFTC_CFG	TAC_CFG	TCP3d_CFG	VCP2_CFG	AIF_CFG	UART_CFG	USIM	OTP Memory	INTD
DAP_Master	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
FFTC_A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FFTC_B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RAC_BE1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AIF	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TAC_FE	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3CC2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CorePacx_CFG (x = core number)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Tracer Master Port	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

4.3 TeraNet Switch Fabric Connections

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

Figure 4-1 TeraNet 2_A

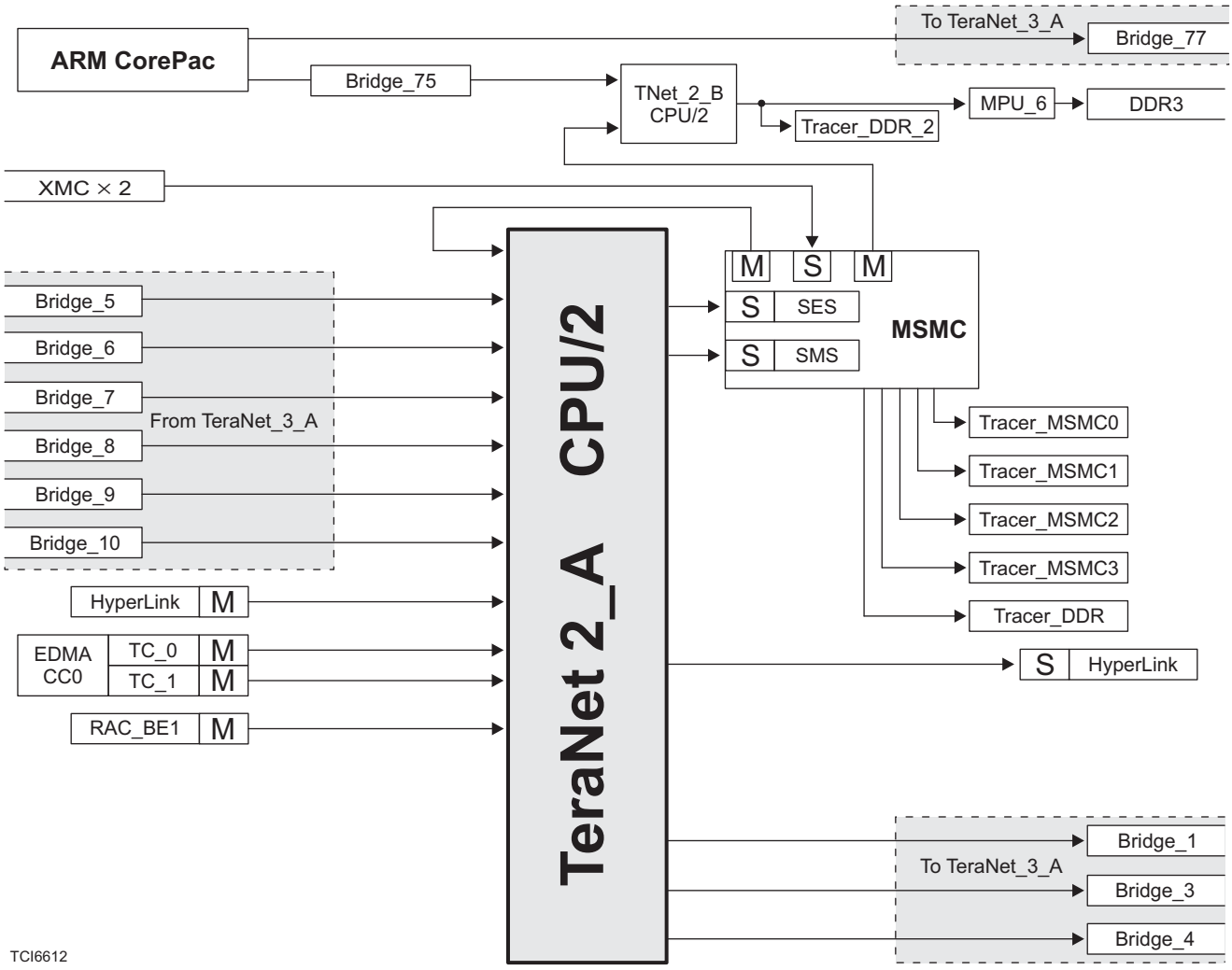


Figure 4-2 TeraNet 3_A

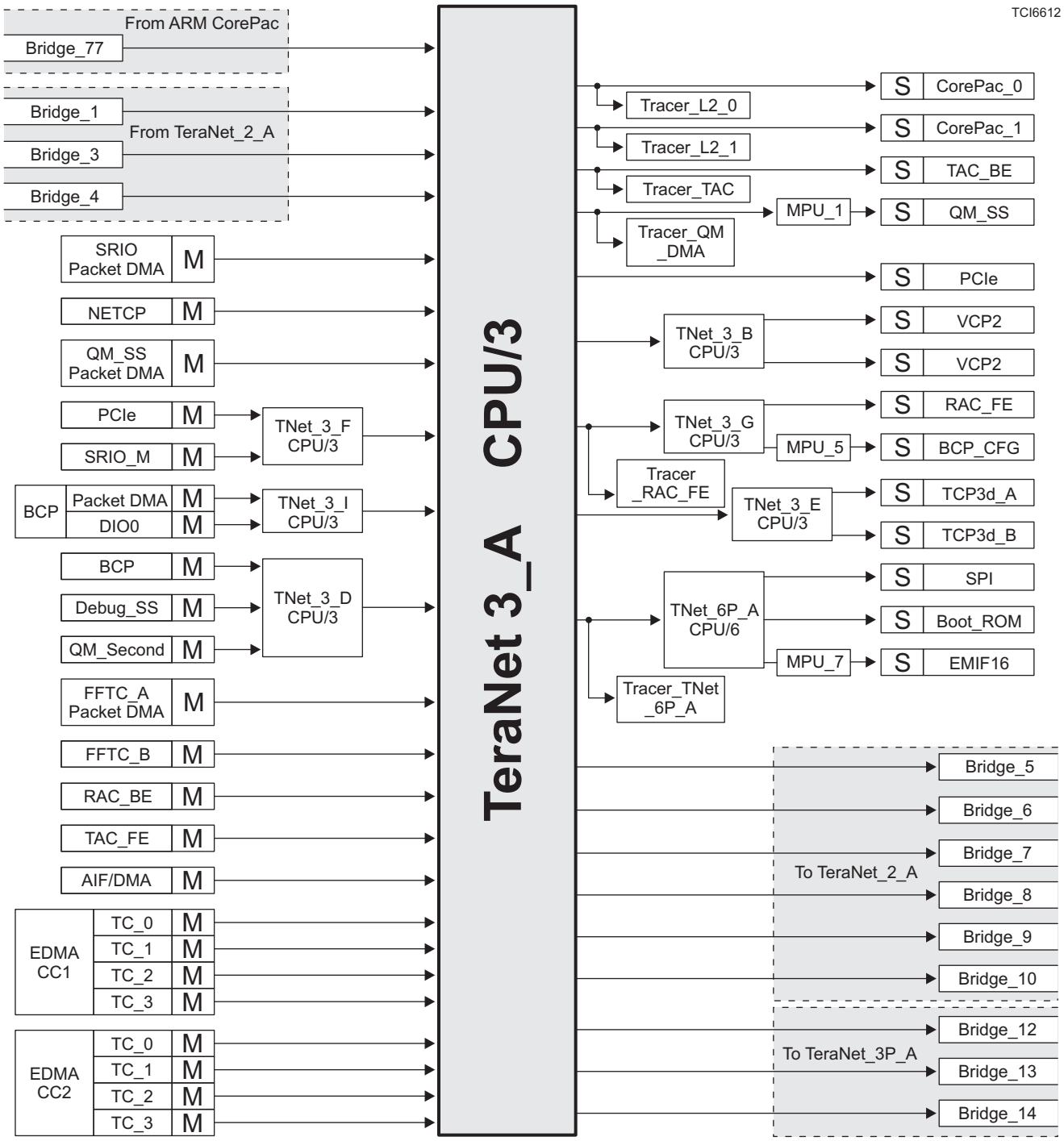


Figure 4-3 TeraNet 3P_A

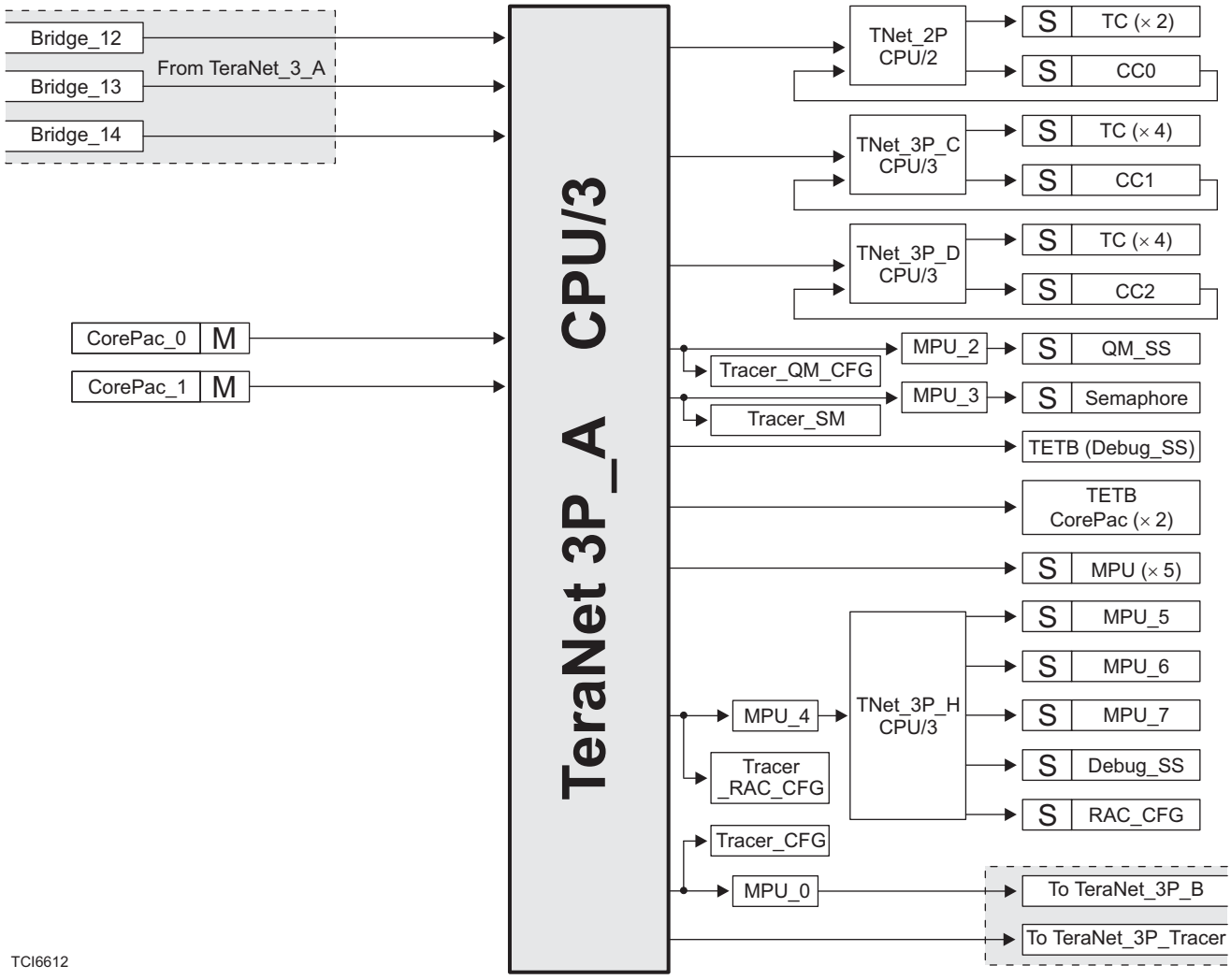


Figure 4-4 TeraNet 3P_B

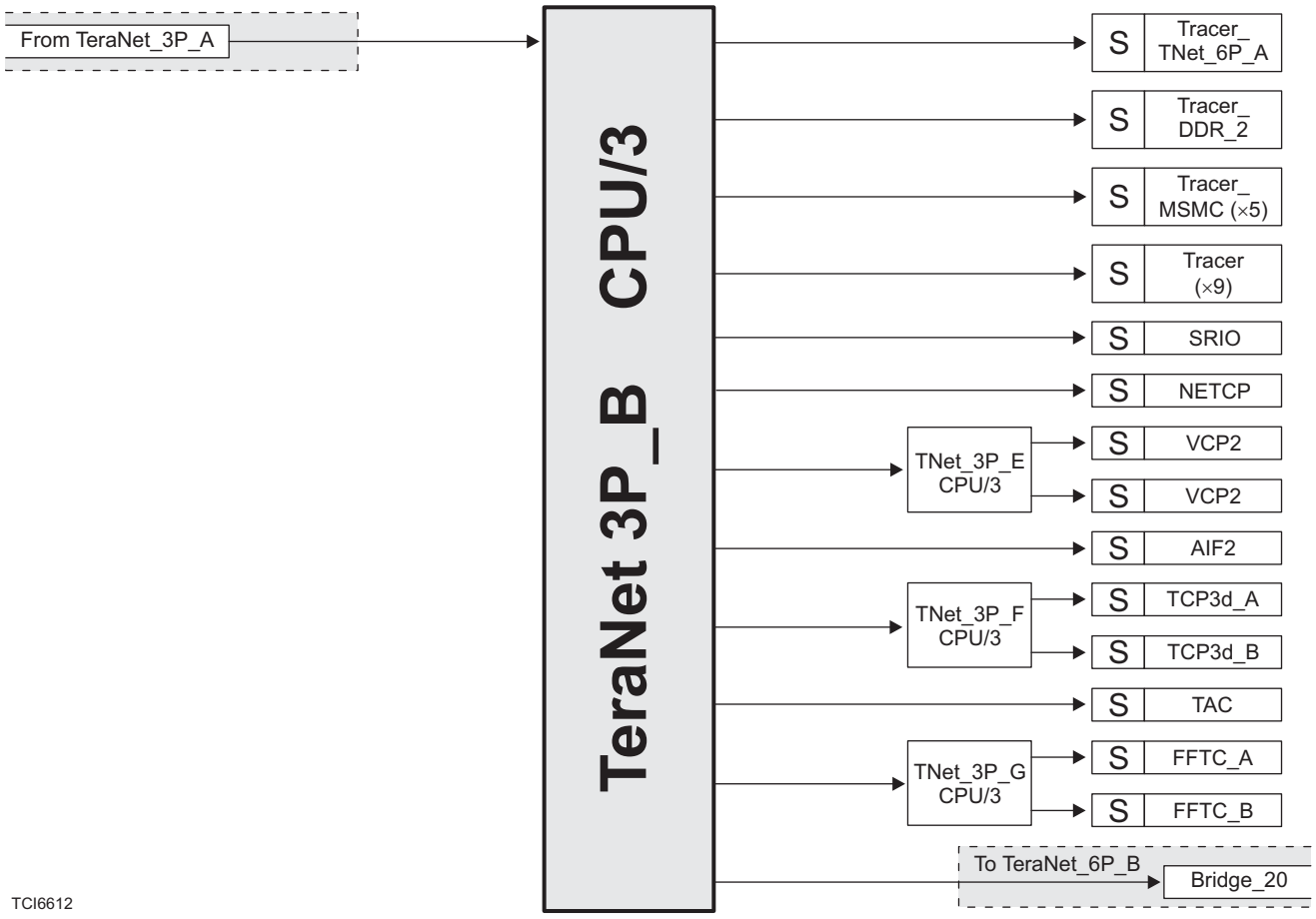


Figure 4-5 TeraNet 3P_Tracer

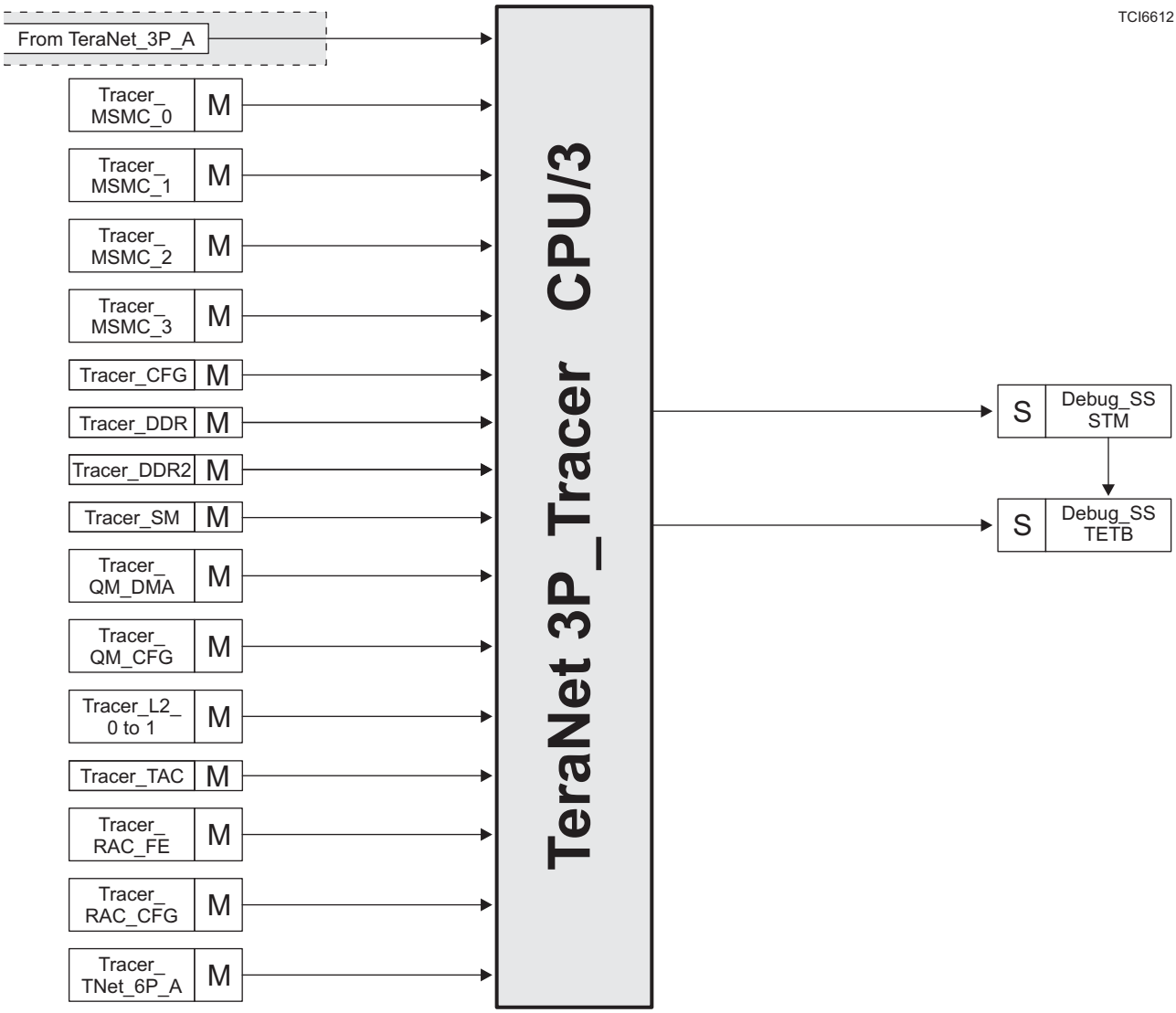
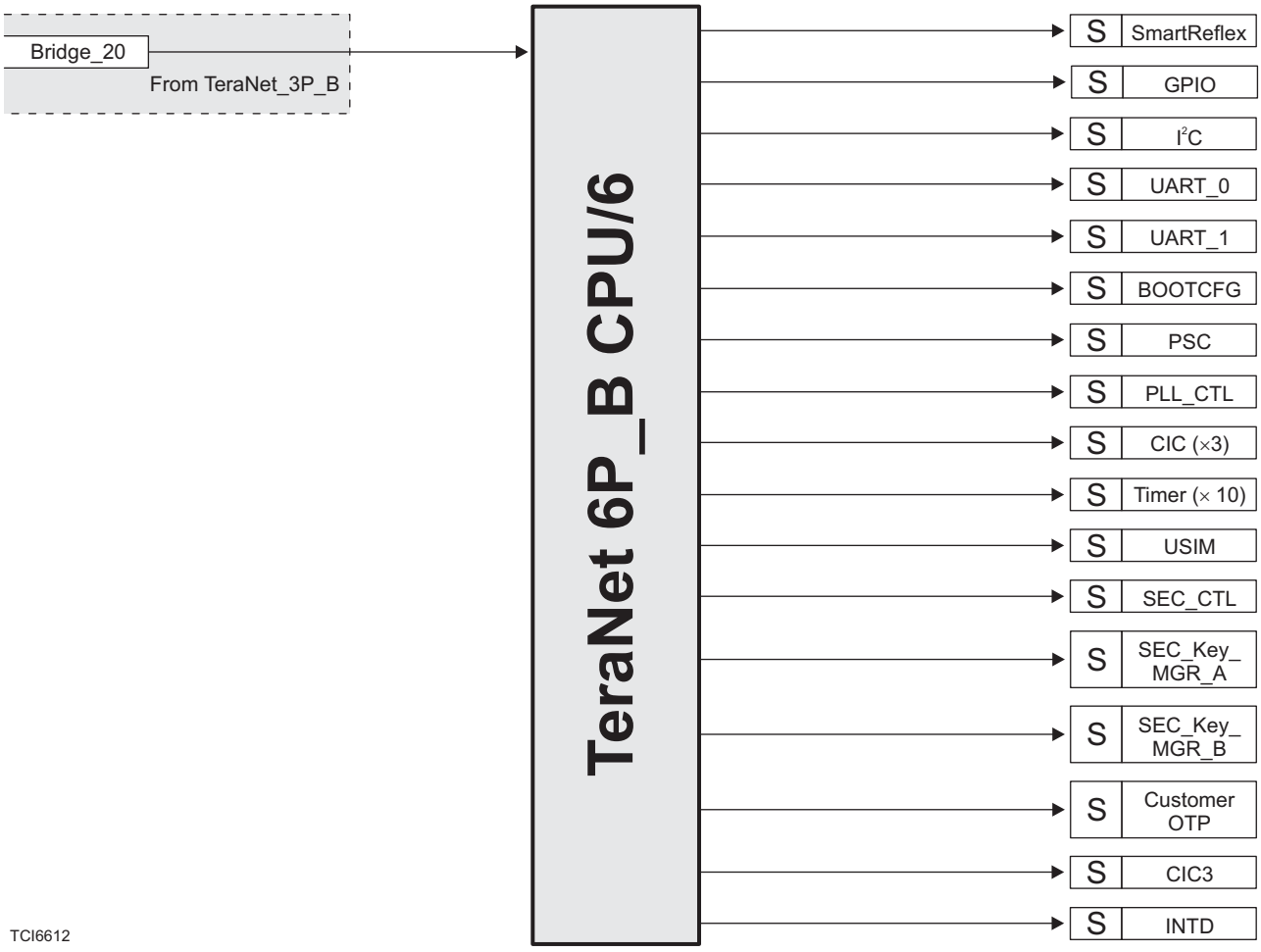


Figure 4-6 TeraNet 6P_B



4.4 Bus Priorities

The priority level of all master peripheral traffic is defined at the TeraNet boundary. User programmable priority registers are present to 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 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 a priority allocation register inside the IP. The priority level for transaction from this master port is described by PKTDMA_PRI_ALLOC register in Figure 4-7 and Table 4-3.

Figure 4-7 Packet DMA Priority Allocation Register (PKTDMA_PRI_ALLOC)

31	3	2	0
Reserved		PKTDMA_PRI	
Y/W-00000000000000000000000001000011		RW-000	

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

Table 4-3 Packet DMA Priority Allocation Register Field Descriptions

Bit	Field	Description
31-3	Reserved	Reserved.
2-0	PKDTDMA_PRI	Control the priority level for the transactions from Packet DMA master port, which access the external linking RAM.
End of Table 4-3		

4.4.1 ARM Priority

ARM has two master ports: master port 0 and master port 1. The priority for each port at the system level can be controlled independently by the fields in the ARM Priority Register shown in Figure 4-8 and Table 4-4.

Figure 4-8 ARM Priority Register

31	27	26	24	23	19	18	16
Reserved		ARM_PORT0_EPRI		Reserved		ARM_PORT0_PRI	
R, +0000 0		R/W, +110		R, +0000 0		R/W, +111	

15	11	10	8	7	3	2	0
Reserved		ARM_PORT1_EPRI		Reserved		ARM_PORT1_PRI	
R, +0000 0		R/W, +110		R, +0000 0		R/W, +111	

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

Table 4-4 ARM Priority Register Field Descriptions

Bit	Field	Description
31-27	Reserved	Reserved
26-24	ARM_PORT0_EPRI	Escalated priority control for the transactions from ARM 128-bit master port. The escalated priority should be set to equal or higher than the priority level specified in the normal priority field (ARM_PORT0_PRI). If the escalated priority level is lower than the priority level, the master port behavior is undetermined. So the escalated priority field should be programmed along with the priority field. 000 = Priority 0 (highest priority) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111= Priority 7 (lowest priority)
23-19	Reserved	Reserved
18-16	ARM_PORT0_PRI	Priority control for the transactions from ARM 128-bit master port 000 = Priority 0 (highest priority) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111= Priority 7 (lowest priority)
15-11	Reserved	Reserved
10-8	ARM_PORT1_EPRI	Escalated priority control for the transactions from ARM 64-bit master port. The escalated priority should be set to equal or higher than the priority level specified in the normal priority field (ARM_PORT1_PRI). If the escalated priority level is lower than the priority level, the master port behavior is undetermined. So the escalated priority field should be programmed along with the priority field. 000 = Priority 0 (highest priority) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111= Priority 7 (lowest priority)
7-6	Reserved	Reserved
2-0	ARM_PORT1_PRI	Priority control for the transactions from ARM 64-bit master port 000 = Priority 0 (highest priority) 001 = Priority 1 010 = Priority 2 011 = Priority 3 100 = Priority 4 101 = Priority 5 110 = Priority 6 111= Priority 7 (lowest priority)
End of Table 4-4		

For all other modules, see the respective User Guides in 2.13 "Related Documentation from Texas Instruments" on page 75 for programmable priority registers.

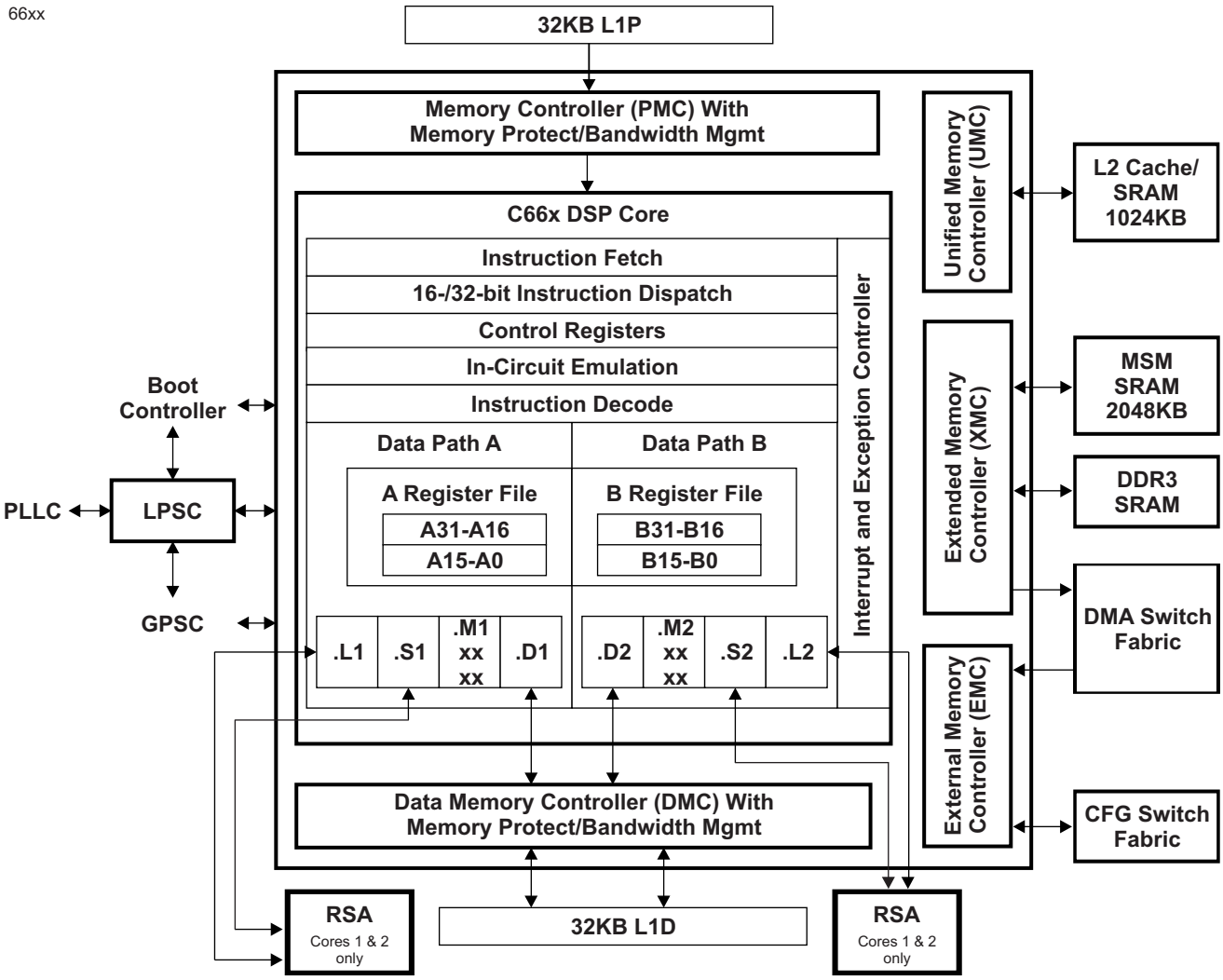
5 C66x CorePac

The C66x CorePac consists of several components:

- The C66x DSP core
- Level-one and level-two memories (L1P, L1D, L2)
- RSA accelerator (on cores 1 and 2 only)
- Data Trace Formatter (DTF)
- Embedded Trace Buffer (ETB)
- Interrupt controller
- Power-down controller
- External memory controller
- Extended memory controller
- A dedicated power/sleep controller (LPSC)

The C66x CorePac also provides support for memory protection and bandwidth management (for resources local to the CorePac). Figure 5-1 shows a block diagram of the C66x CorePac.

Figure 5-1 C66x CorePac Block Diagram



For more detailed information on the C66x CorePac in the TCI6612 device, see the *C66x CorePac User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

5.1 Memory Architecture

Each core in the TMS320TCI6612 device contains a 1024-KB level-2 memory (L2), a 32-KB level-1 program memory (L1P), and a 32-KB level-1 data memory (L1D). The device also contain a 2048-KB multicore shared memory (MSM). All memory on the TCI6612 has a unique location in the memory map (see Table 2-2 "Memory Map Summary" on page 21.

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 *Bootloader for the C66x DSP User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.

For more information on the operation L1 and L2 caches, see the *C66x DSP Cache User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.

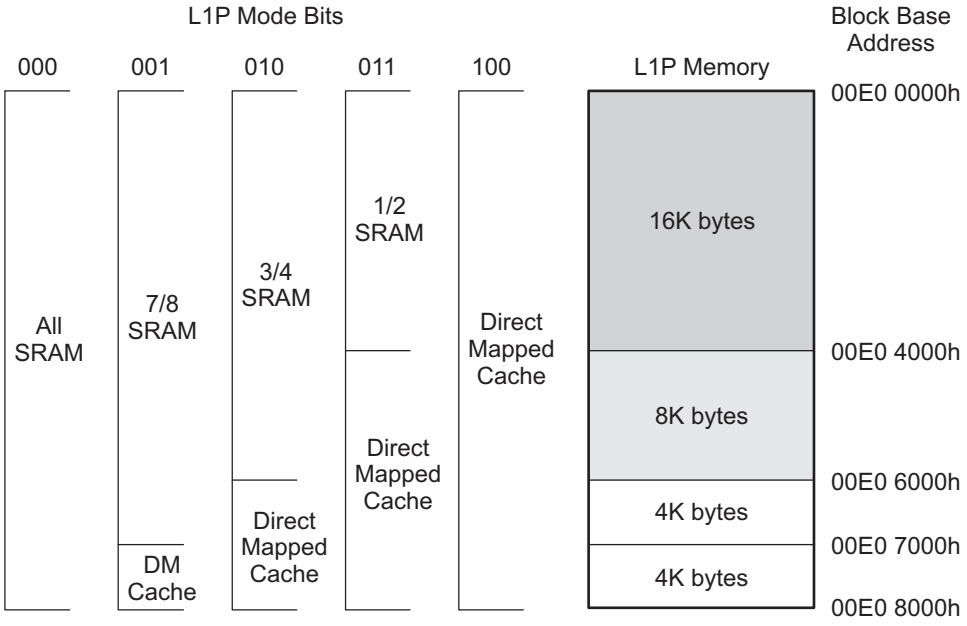
5.1.1 L1P Memory

The L1P memory configuration for the TCI6612 device is as follows:

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

Figure 5-2 shows the available SRAM/cache configurations for L1P.

Figure 5-2 TMS320TCI6612 L1P Memory Configurations



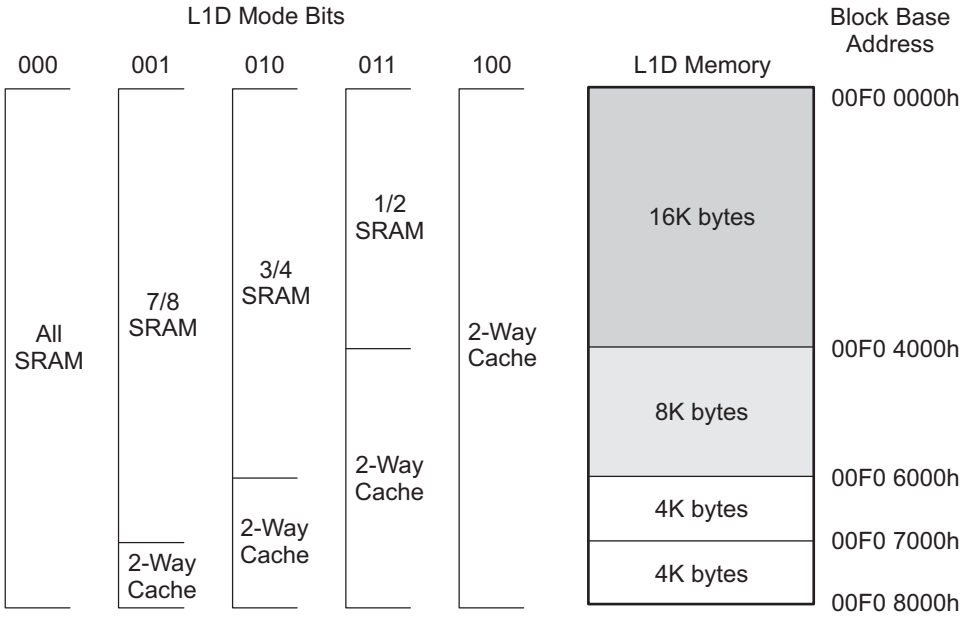
5.1.2 L1D Memory

The L1D memory configuration for the TCI6612 device is as follows:

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

Figure 5-3 shows the available SRAM/cache configurations for L1D.

Figure 5-3 TMS320TCI6612 L1D Memory Configurations



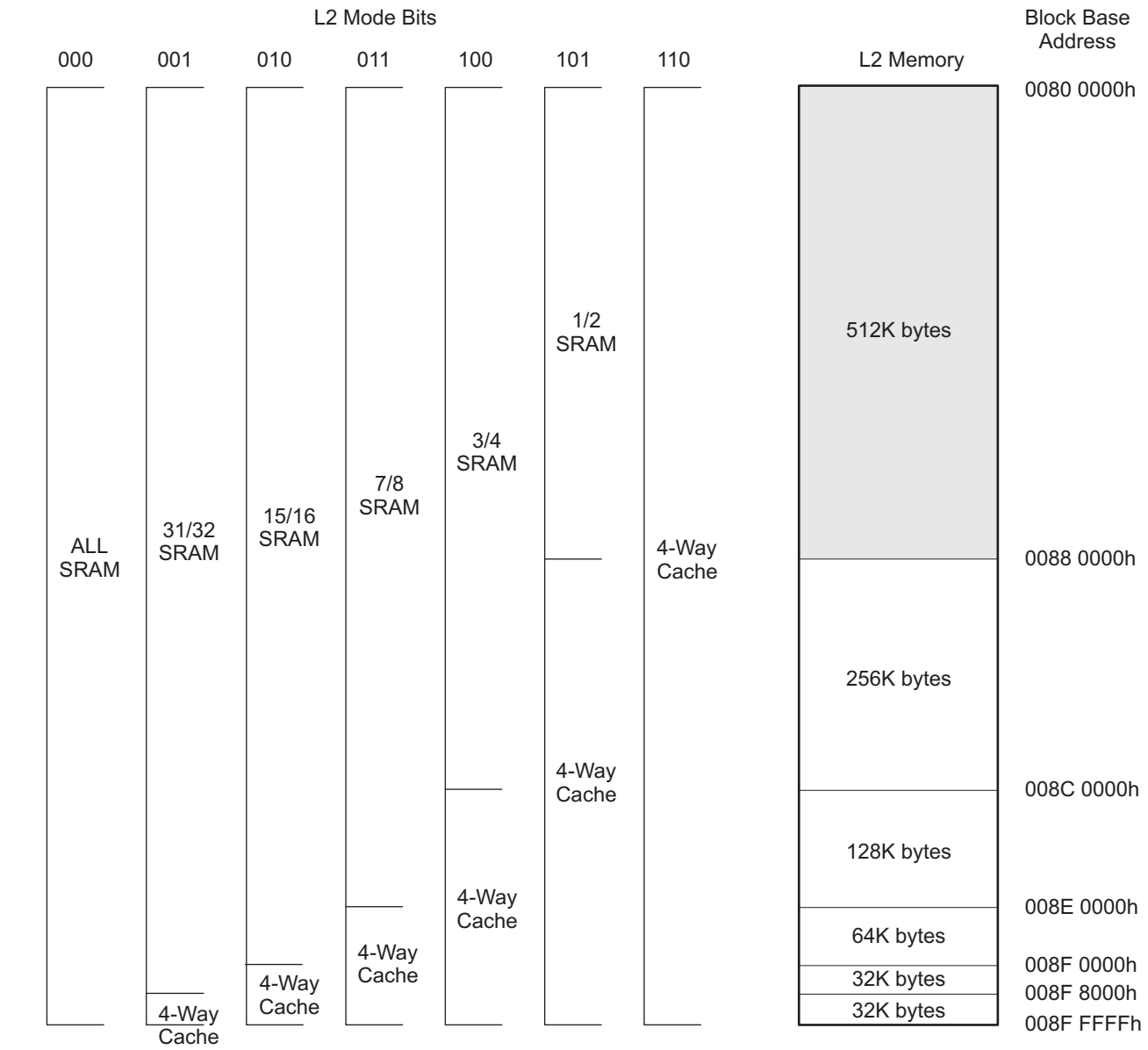
5.1.3 L2 Memory

The L2 memory configuration for the TCI6612 device is as follows:

- Total memory size is 4096KB
- Each core contains 1024KB of memory
- Local starting address for each core 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 5-4 shows the available SRAM/cache configurations for L2. By default, L2 is configured as all SRAM after device reset.

Figure 5-4 TMS320TCI6612 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 either of the two CorePacs as their own L2 base addresses. For CorePac0, as mentioned, this is equivalent to 0x10800000. And for CorePac1, this is equivalent to 0x11800000. 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.

5.1.4 MSM SRAM

The MSM SRAM configuration for the TCI6612 device is as follows:

- Memory size is 2048KB
- The MSM can be configured as shared L2 or shared L3 memory
- Allows extension of external addresses from 2GB to 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 *Multicore Shared Memory Controller (MSMC) for KeyStone Devices User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.

5.1.5 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.

5.2 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 SoC 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 SoC 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 SoC 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 5-1.

Table 5-1 Available Memory Page Protection Schemes

AIDx (1) Bit	Local Bit	Description
0	0	No access to memory page is permitted.
0	1	Only direct access by SoC is permitted.
1	0	Only accesses by system masters and IDMA are permitted (includes EDMA and IDMA accesses initiated by the SoC).
1	1	All accesses permitted.
End of Table 5-1		

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. An SoC 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 event to SoC 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 *C66x CorePac User Guide* in2.13 "Related Documentation from Texas Instruments" on page 75.

5.3 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:

- SoC-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), see Section 4.4 "Bus Priorities" on page 107. 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 *C66x CorePac Reference Guide* (literature number SPRUGW0.)

5.4 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.

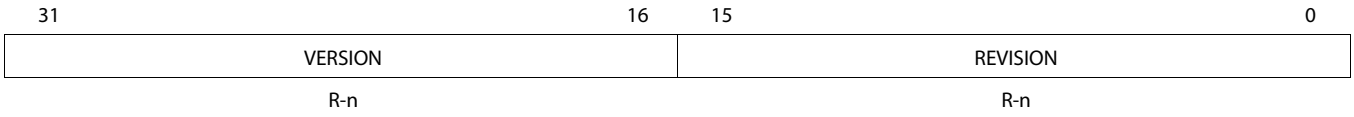
Note—The TCI6612 does not support power-down modes for the L2 memory at this time.

More information on the power-down features of the C66x CorePac can be found in the *C66x CorePac Reference Guide* (literature number SPRUGW0).

5.5 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 Table 5-2 and described in Table 5-2. The C66x CorePac revision is dependant on the silicon revision being used.

Figure 5-5 CorePac Revision ID Register (MM_REVID)



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

Table 5-2 CorePac Revision ID Register Field Descriptions

Bit	Field	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.
End of Table 5-2			

5.6 C66x CorePac Register Descriptions

See the *C66x CorePac User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75 for register offsets and definitions.

6 ARM CorePac

6.1 Introduction

The ARM CorePac of the TMS320TCI6612 handles transactions between the ARM core (ARM® Cortex™-A8 processor), the L3 interconnect, and the interrupt controller (INTC). The ARM CorePac integrates the A8 Core processor with additional logic for protocol conversion, emulation, interrupt handling, and debug enhancements. The A8 Core is an ARMv7-compatible, dual-issue, in-order execution engine, with integrated L1 and L2 caches and a NEON™ SIMD media processing unit.

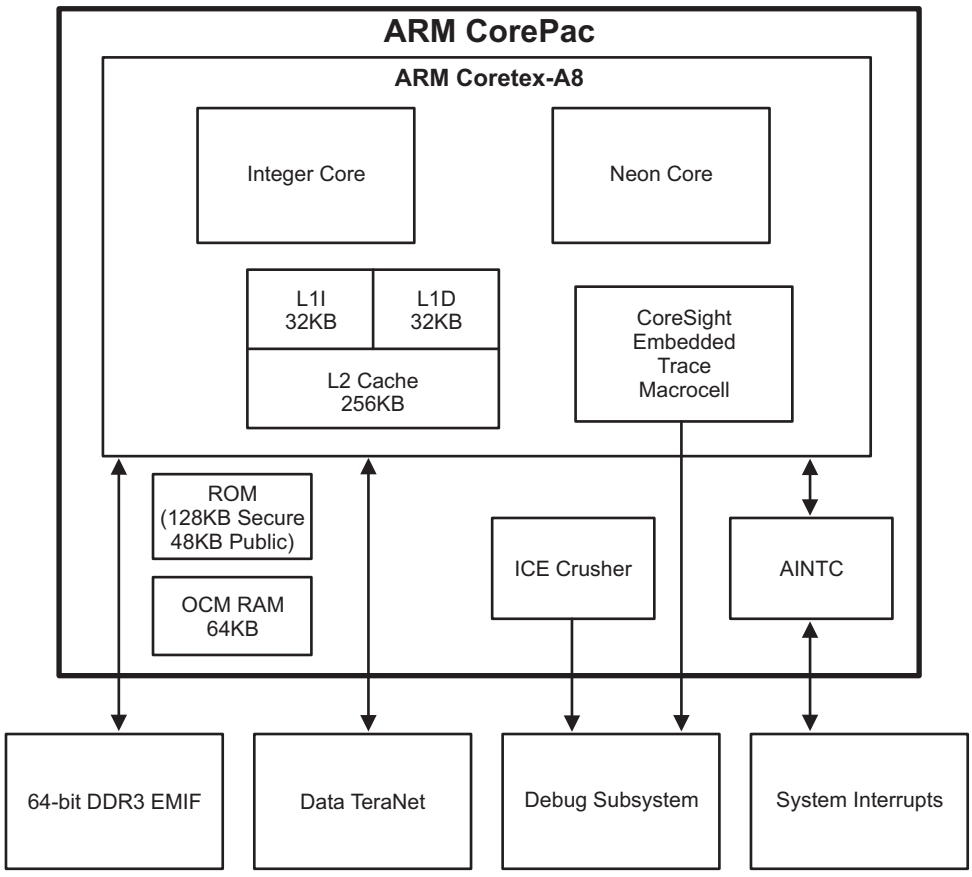
An interrupt controller is included in the ARM CorePac to handle host interrupt requests in the system.

The ARM CorePac includes CoreSight-compliant logic to allow the debug subsystem access to the A8 Core debug and emulation resources, including the embedded trace macrocell.

The ARM CorePac has three functional clock domains, including a high-frequency clock domain used by the A8 Core. The high-frequency domain is isolated from the rest of the device by asynchronous bridges.

Figure 6-1 shows an overall view of the ARM CorePac.

Figure 6-1 ARM CorePac



6.2 Features

The key features of the ARM CorePac are as follows:

- ARM microprocessor
  - A8 Core revision R1P1.
  - ARM architecture version 7 ISA.
  - Two-issue, in-order execution pipeline.
  - L1 and L2 instruction and data cache of 32 KB, 4-way, 16 word line with 128 bit interface.
  - Integrated L2 cache of 256KB, 8-way, 16 word line, 128-bit interface to L1 along with ECC/parity.
  - Includes the Neon media coprocessor (NEON™), which implements the advanced SIMD media processing architecture and the VFPv3 architecture.
  - The external interface uses the AXI protocol configured to 128-bit data width.
  - Includes the embedded trace macrocell (ETM) 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.
- Security
  - SECMON interface to A8 Core
  - Security state machine
  - Firewall
  - Secure RAM and ROM
- Interrupt controller
  - Supports up to 128 interrupt requests
- Emulation/debug
  - Compatible with CoreSight architecture.
- Clock generation
  - Through SYSCLK1 and SYSCLK2

6.3 System Integration

The ARM CorePac integrates the following group of submodules.

- **ARM A8 Core Processor:** Provides a high processing capability, including the NEON technology for mobile multimedia acceleration. The ARM A8 Core communicates with the rest of the ARM system through an AXI bus with an AXI2OCP 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 the Interrupt Controller section).
- **Clock Divider:** Provides the required divided clocks to the internal modules of the ARM CorePac and has a clock input from SYSCLK2.
- **In-Circuit Emulator:** Fully compatible with CoreSight architecture and enables debugging capabilities.

6.4 ARM A8 Core

6.4.1 Overview

The ARM A8 Core 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 VFPv3 floating point architecture. For details, see the ARM Cortex-A8 Technical Reference Manual.

6.4.2 Features

Table 6-1 shows the features supported by the ARM core.

Table 6-1 ARM Core Supported Features

Features	Description
ARM version 7 ISA	Standard ARM instruction set + Thumb2™, JazelleX™ Java accelerator, and media extensions
	Backward compatible with previous ARM ISA versions
A8 Core version	R1P1
L1 Lcache and Dcache	32KB, 4-way, 16-word line, 128-bit interface
L2 cache	256KB, 8-way, 16-word line, 128-bit interface to L1, ECC/Parity is supported.
	L2 valid bits cleared by software loop or by hardware
Flat memories	176K bytes of ROM
	64K bytes of RAM
TLB	Fully associative and separate ITLB with 32 entries and DTLB with 32 entries
CoreSight ETM	The CoreSight ETM is embedded within the ARM CorePac. The 32KB buffer (ETB) exists at the chip level debugSS
Branch target address cache	512 entries
Enhanced memory management unit	Mapping sizes are 4KB, 64KB, 1MB, and 16MB
Integer core	Main core for processing integer instructions
Neon core	Gives greatly enhanced throughput for media workloads and VFP-Lite support
Buses	128-bit AXI internal bus from A8 Core routed by an AXI2OCP bridge to the interrupt controller, ROM, RAM, and 3 asynchronous OCP bridges (128 bits, and 64 bits)
Low interrupt latency	Closely coupled INTC to the ARM core with 128 interrupt lines
Vectored interrupt controller port	Present
JTAG-based debug	Supported via debug access port
Trace support	CoreSight trace supported

6.4.3 ARM Interrupt Controller

The host ARM interrupt controller (AINTC) is responsible for prioritizing all service requests from the system peripherals and generating either nIRQ or nFIQ to the host. The type of the interrupt (nIRQ or nFIQ) and the priority of the interrupt inputs are programmable. The AINTC interfaces to the ARM processor via the AXI port through an AXI2OCP bridge and runs at half the processor speed. It has the capability to handle up to 128 requests, which can be steered/prioritized as A8 nFIQ or nIRQ interrupt requests.

The general features of the AINTC are:

- Up to 128 level sensitive interrupts 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 the Interrupt section for more details.

6.4.4 Endianess

The ARM core operates only in little endian mode. When the TCI6612 runs in big endian mode the bridges in the ARM CorePac are responsible for performing the endian conversion.

6.5 Word Swap

Word swap is used only when big endian mode is enabled; word swap has no impact when big endian is disabled (little endian). This word swap will be done automatically by ARM CorePac hardware. Table 3-3 summarizes the transactions to/from the following address ranges with word swap in KeyStone devices.

Table 3-3 Word Swapping Region

Region Index	Start Address	End Address	Region Size	Notes
1	0x0000_0000	0x07FF_FFFF	128MB	Generic CFG space
2	0x0BC0_0000	0x0BCF_FFFF	1MB	MSMC CFG space
3	0x20BF_0000	0x20BF_FFFF	64KB	SPI
4	0x20C0_0000	0x20FF_FFFF	4MB	EMIF16 CFG
5	0x2100_0000	0x21FF_FFFF	16MB	HyperLink/PCIe/DDR CFG space
6	0x3360_0000	0x337F_FFFF	2MB	Currently reserved for CFG purpose
7	0x3400_0000	0x341F_FFFF	2MB	QM
8	0x3500_0000	0x35FF_FFFF	16MB	BCP CFG

6.6 CFG Connection

The ARM CorePac does not have a slave port. The TCI6612 masters cannot access the ARM CorePac's internal memory space.

6.7 Main TeraNet Connection

There are two master ports coming out of the ARM CorePac:

1. Master port 0 is a 128-bit wide port for the transactions going to the DDR_EMIF data space.
2. Master port 1 is a 64-bit wide port used to access the rest of the system.

6.8 Clocking and Reset

6.8.1 Clocking

The ARM CorePac does not include an embedded DPLL. The clock is sourced from the Main PLL Controller. A clock divider within the subsystem is used for deriving the clocks for other internal modules. The main ARM core clock has a maximum frequency of 1.2 Ghz, and uses the same clock source as the CorePacs. All major modules inside the ARM CorePac are clocked at half the frequency of the ARM core, such as the ICECrusher and AINTC modules. The emulation clock within the ARM core runs at one third the frequency of the ARM core. The divider of the output clock is programmable, with the frequency relative to the ARM core.

6.8.2 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 be reset only during $\overline{\text{POR}}$ and $\overline{\text{RESETFULL}}$.

For the complete programming model, see the ARM Cortex-A8 Technical Reference Manual:
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0344c/index.html

6.9 ARM CorePac Memory Map

Table 6-2 shows the ARM core memory map.

Table 6-2 ARM CorePac Memory Map

Region	Address Range	Size	Additional note
Internal Memory (Access Not Routed To External OCP Ports)			
Boot ROM (128 KB) secure	0x4000_0000 – 0x4001_FFFF	1MB	
ROM public (48 KB)	0x4002_0000 – 0x4002_BFFF		
Reserved	0x4002_C000 – 0x400F_FFFF		
Reserved	0x4020_0000 – 0x402E_FFFF	1MB	
SRAM (64KB) secure/public	0x402F_0000 – 0x402F_FFFF		
Internal Reserved			
Reserved	0x4010_0000 – 0x401F_FFFF	1MB	
Private Peripheral Map (Access Not Routed To External OCP Ports)			
Arm interrupt controller (AINTC)	0x4820_0000 – 0x4820_0FFF	4KB	
Reserved	0x4820_1000 – 0x4827_FFFF	508KB	
Secure state machine (SSM)	0x4828_0000 – 0x4828_0FFF	4KB	
Reserved	0x4828_1000 – 0x482F_FFFF	508KB	
128-bit OCP Master Port 0 (to DDR3_EMIF Data Space)			
DDR3_EMIF	0x8000_0000 – 0xFFFF_FFFF	2GB	Connects to the DDR3_EMIF through the TeraNet
64-bit OCP Master Port 1 (To The Rest Of The System Except The DDR3_EMIF Data Space)			
Boot space [1]	0x0000_0000 – 0x000F_FFFF	1MB	It is redirected to 0x4000 0000 – 0x 400F FFFF for boot.
L3	The rest of address range not listed from 0x0000_0000 – 0x7FFF_FFFF	(2GB – 5MB)	The ARM has a different memory map view for the address range between 0x3000_0000 to 0x4FFF_FFFF compared to the rest of the SoC masters. When the ARM issues a transaction in the address range between 0x3000_ 0000 to 0x3FFF_FFFF, the transaction is swapped with address 0x4000_000 to 0x4FFFF_FFFF before the transaction is sent to the rest of device. On the other hand, the transactions from ARM to address space 0x4000_0000 to 0x4FFF_FFFF is swapped with address 0x3000_ 0000 to 0x3FFF_FFFF before it is sent to the rest of the device. This address swapping is done by the OCP2VBUS bridge.
End of Table 6-2			

Table 6-3 shows how the ARM views portions of the memory map differently from other masters as a result of address swapping.

Table 6-3 Address Comparison between ARM and non-ARM Masters

	Virtual Address From Non_ARM Masters	Virtual Address From ARM
RAC_Data_A	0x3320_0000 to 0x335F_FFFF	0x4320_0000 to 0x435F_FFFF
QM_SS_VBUSM	0x3400_0000 to 0x341F_FFFF	0x4400_0000 to 0x441F_FFFF
TAC_BE1	0x34C0_0000 to 0x34C2_0000	0x44C0_0000 to 0x44C2_0000
BCP_CFG	0x3520_0000 to 0x3521_FFFF	0x4520_0000 to 0x4521_FFFF
Hyperlink	0x4000_0000 to 0x4FFF_FFFF	0x3000_0000 to 0x3FFF_FFFF
End of Table 6-3		

7 Device Operating Conditions

7.1 Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings [1]
Over Operating Case Temperature Range (Unless Otherwise Noted)

Supply voltage range [2]:	CVDD		-0.3 V to 1.3 V
	CVDD1		-0.3 V to 1.3 V
	DVDD15		-0.3 V to 2.45 V
	DVDD18		-0.3 V to 2.45 V
	VREFHSTL		0.49 × DVDD15 to 0.51 × DVDD15
	VDDT1, VDDT2		-0.3 V to 1.3 V
	VDDR1, VDDR2, VDDR3, VDDR4, VDDR5, VDDR6		-0.3 V to 2.45 V
	AVDDA1, AVDDA2, AVDDA3		-0.3 V to 2.45 V
	VSS Ground		0 V
Input voltage (V_I) range:	LVCMOS (1.8 V)		-0.3 V to DVDD18+0.3 V
	DDR3		-0.3 V to 2.45 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 CVDD1+0.3 V
Output voltage (V_O) range:	LVCMOS (1.8 V)		-0.3 V to DVDD18+0.3 V
	DDR3		-0.3 V to 2.45 V
	I^2C		-0.3 V to 2.45 V
	SerDes		-0.3 V to CVDD1+0.3 V
Operating case temperature range, T_C:	Commercial	1-GHz CPU	0°C to 100°C
		1.2-GHz CPU	0°C to 100°C
	Extended	1-GHz CPU	-40°C to 100°C
		1.2-GHz CPU	-40°C to 100°C
Overshoot/undershoot [3]	LVCMOS (1.8 V)		20% Overshoot/Undershoot for 20% of Signal Duty Cycle
	DDR3		
	I^2C		
Storage temperature range, T_{stg}:			-65°C to 150°C
End of Table 7-1			

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 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

7.2 Recommended Operating Conditions

Table 7-2 Recommended Operating Conditions (1) (2)

			Min	Nom	Max	Unit
CVDD	SR Core supply	Initial Startup	1.045	1.1 (3)	1.155	V
		1000MHz - Device	SRVnom*0.95 (4)	0.85-1.1	SRVnom*1.05	
		1200MHz - Device	SRVnom*0.95	0.85-1.1	SRVnom*1.05	
CVDD1	Core supply		0.95	1	1.05	V
DVDD18	1.8-V supply I/O voltage		1.71	1.8	1.89	V
DVDD15	1.5-V supply I/O voltage		1.425	1.5	1.575	V
VREFHSTL	DDR3 reference voltage		0.49 × DVDD15	0.5 × DVDD15	0.51 × DVDD15	V
V_{DDRx} (5)	SerDes regulator supply		1.425	1.5	1.575	V
V_{DDAx}	PLL analog supply		1.71	1.8	1.89	V
V_{DDTx}	SerDes termination supply		0.95	1	1.05	V
V_{SS}	Ground		0	0	0	V
V_{IH}	High-level input voltage	LVCMOS (1.8 V)	0.65 × DVDD18			V
		I^2C	0.7 × DVDD18			V
		DDR3 EMIF	VREFHSTL + 0.1			V
V_{IL}	Low-level input voltage	LVCMOS (1.8 V)			0.35 × DVDD18	V
		DDR3 EMIF	-0.3		VREFHSTL - 0.1	V
		I^2C			0.3 × DVDD18	V
T_C	Operating case temperature	Commercial	0		100	°C
		Extended	-40		100	°C

End of Table 7-2

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 The initial CVDD voltage at power on will be 1.1 V nominal and it must transition to VID set value immediately after being presented on 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 set from the factory for each individual device.

5 Where x = 1, 2, 3, 4... to indicate all supplies of the same kind.

7.3 Electrical Characteristics

Table 7-3 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} High-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OH}$	DVDD18 - 0.45			V
	DDR3		DVDD15 - 0.4			
	I^2C (2)					
V_{OL} Low-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OL}$			0.45	V
	DDR3				0.4	
	I^2C	I_O = 3 mA, pulled up to 1.8 V			0.4	
I_I (3) Input current [DC]	LVCMOS (1.8 V)	No IPD/IPU	-5		5	µA
		Internal pullup	50	100	170 (4)	
		Internal pulldown	-170	-100	-50	
	I^2C	0.1 × DVDD18 V < V_I < 0.9 × DVDD18 V	-10		10	
I_{OH} High-level output current [DC]	LVCMOS (1.8 V)				-6	mA
	DDR3				-8	
	I^2C					
I_{OL} Low-level output current [DC]	LVCMOS (1.8 V)				6	mA
	DDR3				8	
	I^2C				3	
I_{OZ} (5) Off-state output current [DC]	LVCMOS (1.8 V)		-2		2	µA
	DDR3		-2		2	
	I^2C		-2		2	

End of Table 7-3

1 For test conditions shown as MIN, MAX, or TYP, use the appropriate value specified in the recommended operating conditions table.
2 I^2C uses open collector IOs and does not have a V_{OH} Minimum.
3 I_I applies to input-only pins and bi-directional pins. For input-only pins, I_I indicates the input leakage current. For bi-directional pins, I_I includes input leakage current and off-state (Hi-Z) output leakage current.
4 For $\overline{\text{RESETSTAT}}$, max DC input current is 300 µA.
5 I_{OZ} applies to output-only pins, indicating off-state (Hi-Z) output leakage current.

Table 7-4 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
CV_{DD}	Supply core voltage	LJCB	CORECLK(P|N) PLL input buffer
			ALTCORECLK(P|N) PLL input buffer
			SRIOSGMIICLK(P|N) SerDes PLL input buffer
			DDRCLK(P|N) PLL input buffer
			PCIECLK(P|N) SerDes PLL input buffer
			MCMCLK(P|N) SerDes PLL input buffer
			PASSCLK(P|N) PLL input buffer
DV_{DD15}	1.5-V supply I/O voltage	DDR3 (1.5 V)	All DDR3 memory controller peripheral I/O buffer
DV_{DD18}	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 AIF peripheral I/O buffer
			All RESETs, NMI, Control peripheral I/O buffer
			All Smart Reflex peripheral I/O buffer
			All HyperLink sideband peripheral I/O buffer
			All MDIO peripheral I/O buffer
			All UART peripheral I/O buffer
			All EMIF16 peripheral I/O buffer
		Open-drain (1.8 V)	All I^2C peripheral I/O buffer
V_{DDT1}	Hyperlink/AIF SERDES Termination and analogue front-end supply	SERDES/CML	Hyperlink/AIF SerDes CML IO buffer
V_{DDT2}	SRIO/SGMII/PCIE SERDES Termination and analogue front-end supply	SERDES/CML	SRIO/SGMII/PCIE SerDes CML IO buffer
End of Table 7-4			

1 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 See the Hardware Design Guide for KeyStone Devices (literature number SPRABI2) for more information about individual peripheral I/O.

8 TMS320TCI6612 Peripheral Information and Electrical Specifications

This chapter covers the various peripherals on the TMS320TCI6612 device. Peripheral-specific information, timing diagrams, electrical specifications, and register memory maps are described in this chapter.

8.1 Parameter Information

This section is left for future revisions.

8.2 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.

8.3 Power Supplies

The following sections describe the proper power-supply sequencing and timing needed to properly power on the TCI6612. The various power supply rails and their primary function is listed in Table 8-1 below.

Table 8-1 Power Supply Rails on TMS320TCI6612

Name	Primary Function	Voltage	Notes
CVDD	SmartReflex core supply voltage	0.85 V - 1.1 V	Variable core supply
CVDD1	Core supply voltage for memory array	1.0 V	Fixed supply at 1.0 V
VDDT1	HyperLink/AIF SerDes termination supply	1.0 V	Filtered version of CVDD1. Special considerations for noise. Filter is not needed if HyperLink and AIF are both not in use.
VDDT2	SGMII/SRIO/PCIE SerDes termination supply	1.0 V	Filtered version of CVDD1. Special considerations for noise. Filter is not needed if SGMII/SRIO/PCIE are all not in use.
DVDD15	1.5-V DDR3 IO supply	1.5 V	Fixed supply at 1.5 V
VDDR1	HyperLink SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if HyperLink is not in use.
VDDR2	PCIE SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if PCIE is not in use.
VDDR3	SGMII SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if SGMII is not in use.
VDDR4	SRIO SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if SRIO is not in use.
VDDR5	AIF SerDes regulator supply	1.5 V	Filtered version of DVDD15. Special considerations for noise. Filter is not needed if AIF is not in use.
VDDR6			
DVDD18	1.8-V IO supply	1.8 V	Fixed supply at 1.8 V
AVDDA1	Main PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
AVDDA2	DDR3 PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
AVDDA3	PASS PLL supply	1.8 V	Filtered version of DVDD18. Special considerations for noise.
VPP	OPT Memory supply	1.8 V	Supply for 4Kbits OTP memory on secure devices [1]. See the *Security Addendum for KeyStone I Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75 for more information. Leave unconnected on unsecure devices.
VREFHSTL	0.75-V DDR3 reference voltage	0.75 V	Should track the 1.5-V supply. Use 1.5 V as source.
VSS	Ground	GND	Ground
End of Table 8-1			

1 The secure version of the TCI6612 device contains hardware features to support security within the device. See Figure 2-23 for the *SoC SECURITY* symbol in the device nomenclature.

8.3.1 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, VDDT1-2
3. DVDD18, AVDD1, AVDD2
4. DVDD15, VDDR1-6

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

1. DVDD18, AVDD1, AVDD2
2. CVDD
3. CVDD1, VDDT1-2
4. DVDD15, VDDR1-6

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

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

The device initialization is broken 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. $\overline{\text{POR}}$ must be held low for the entire power stabilization phase.

This is followed by the device initialization phase. The rising edge of $\overline{\text{POR}}$ followed by the rising edge of $\overline{\text{RESETFULL}}$ will trigger the end of the initialization phase but both must be inactive for the initialization to complete. $\overline{\text{POR}}$ must always go inactive before $\overline{\text{RESETFULL}}$ goes inactive as described below. The following section has a mention of SYSCLK1 in many places. SYSCLK1 here refers to the clock input that has been selected as the source for the Main PLL. See Figure 8-7 for more details.

8.3.1.1 Core-Before-IO Power Sequencing

Figure 8-1 shows the power sequencing and reset control of TMS320TCI6612 for device initialization. $\overline{\text{POR}}$ may be removed after the power has been stable for the required 100 µsec. $\overline{\text{RESETFULL}}$ must be held low for a period after the rising edge of $\overline{\text{POR}}$ 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 $\overline{\text{RESETFULL}}$ and must meet the setup and hold times specified. SYSCLK1 must always be active before $\overline{\text{POR}}$ can be removed. Core-before-IO power sequencing is defined in Table 8-2.

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.

Figure 8-1 Core Before IO Power Sequencing

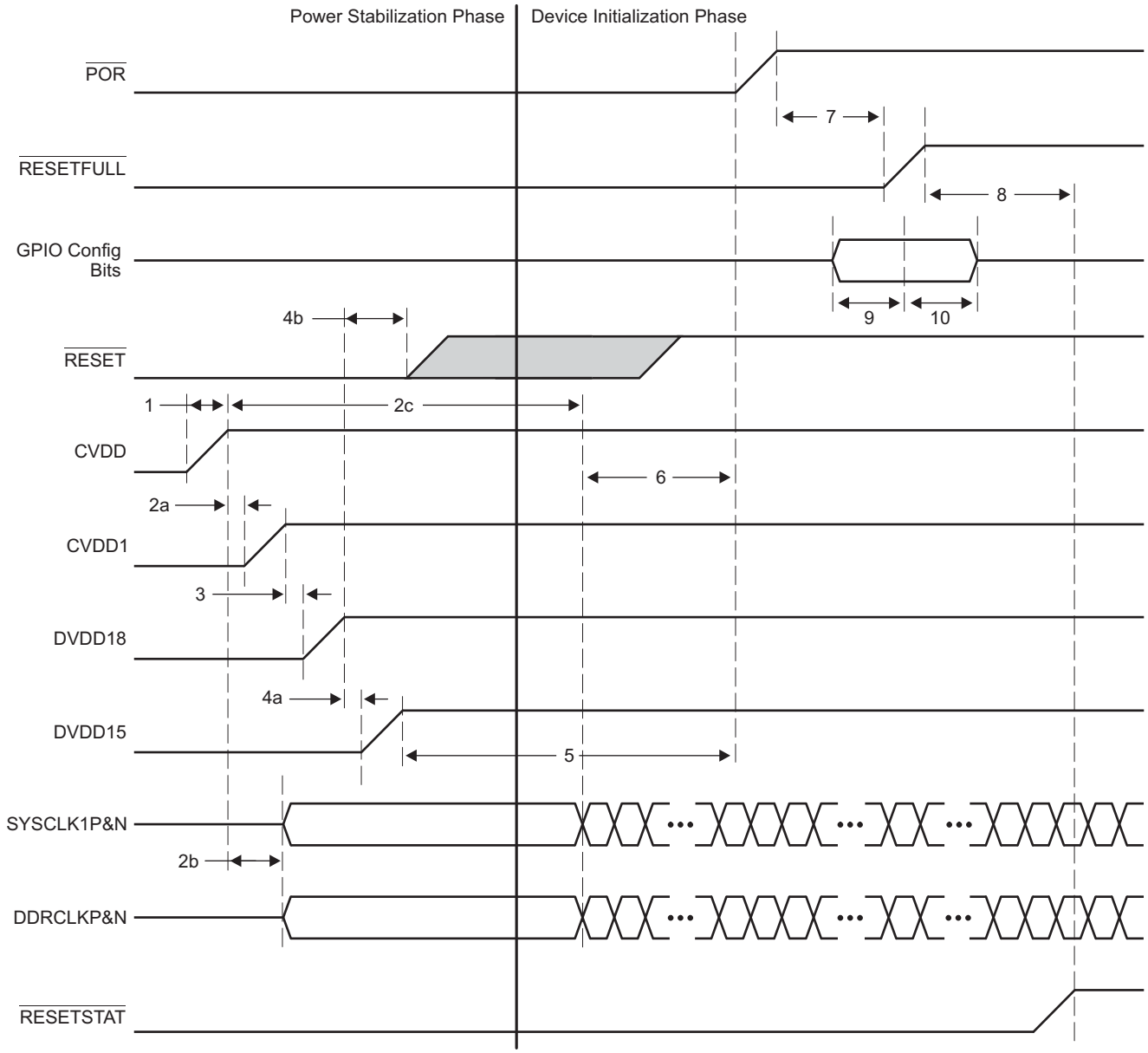


Table 8-2 Core Before IO Power Sequencing

Time	System State
1	**Begin Power Stabilization Phase** • CVDD (core AVS) ramps 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 async reset (created from $\overline{\text{POR}}$) is put into the reset state.
2a	• CVDD1 (core constant) ramps at the same time or shortly following CVDD. Although ramping CVDD1 and CVDD simultaneously 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 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.
2b	• Once CVDD is valid, the clock drivers should 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.
2c	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6.
3	• Filtered versions of 1.8 V can ramp simultaneously with DVDD18. • 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.
4a	• DVDD15 (1.5 V) supply is ramped up following DVDD18. Although ramping DVDD18 and DVDD15 simultaneously is permitted, the voltage for DVDD15 ***must never exceed*** DVDD18.
4b	• $\overline{\text{RESET}}$ may be driven high any time after DVDD18 is at a valid level. In a $\overline{\text{POR}}$-controlled boot, $\overline{\text{RESET}}$ must be high before $\overline{\text{POR}}$ is driven high.
5	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after power has stabilized. **End Power Stabilization Phase**
6	**Begin Device Initialization Phase** • 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.
7	• $\overline{\text{RESETFULL}}$ must be held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level.
8	• 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**
9	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$
10	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$
End of Table 8-2	

8.3.1.2 IO-Before-Core Power Sequencing

The timing diagram for IO-before-core power sequencing is shown in Figure 8-2 and defined in Table 8-3.

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.

Figure 8-2 IO Before Core Power Sequencing

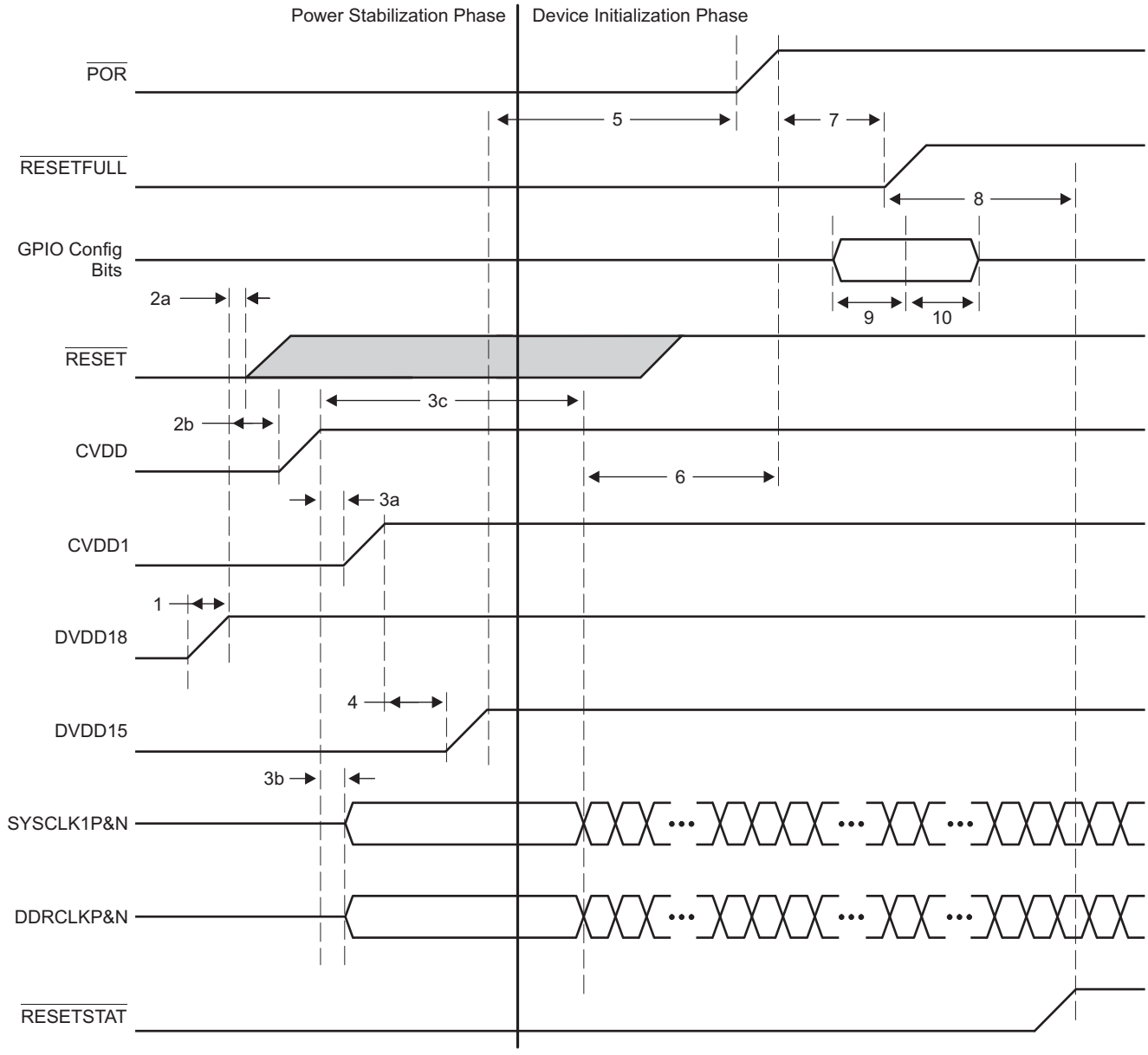


Table 8-3 IO Before Core Power Sequencing

Time	System State
1	**Begin Power Stabilization Phase** • Because $\overline{\text{POR}}$ is low, all the core logic having async reset (created from POR) are put into reset state once the core supply ramps. $\overline{\text{POR}}$ must remain low through Power Stabilization Phase. • Filtered versions of 1.8 V can ramp simultaneously with DVDD18. • $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. • All input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 could cause damage to the device.
2a	• $\overline{\text{RESET}}$ may be driven high anytime after DVDD18 is at a valid level.
2b	• CVDD (core AVS) ramps up.
3a	• CVDD1 (core constant) ramps at the same time or following CVDD. Although ramping CVDD1 and CVDD simultaneously 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 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.
3b	• Once CVDD is valid, the clock drivers should 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 with one leg high and one leg low.
3c	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by t6.
4	• DVDD15 (1.5 V) supply is ramped up following CVDD1.
5	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after power has stabilized. **End Power Stabilization Phase**
6	**Begin Device Initialization Phase** • 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. • $\overline{\text{POR}}$ must remain low.
7	• $\overline{\text{RESETFULL}}$ is held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level. • The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin.
8	• 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**
9	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$
10	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$
End of Table 8-3	

8.3.1.3 Prolonged Resets

Holding the device in $\overline{\text{POR}}$, $\overline{\text{RESETFULL}}$, or $\overline{\text{RESET}}$ for long periods of time will affect the long-term reliability of the part. The device should not be held in a reset for times exceeding one hour and should not be held in reset for more the 5% of the time during which power is applied. Exceeding these limits will cause a gradual reduction in the reliability of the part. This can be avoided by allowing the SoC 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.

8.3.2 Power-Down Sequence

The power down sequence is the exact reverse of the power-up sequence described above. The goal is to prevent a large 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 also affect long term reliability.

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 8-4 describes the clock sequencing and the conditions that affect the 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 either be active or in a static state with one leg pulled low and the other connected to CVDD.

Table 8-4 Clock Sequencing

Clock	Condition	Sequencing
DDRCLK	None	Must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
SYSCLK	CORECLKSEL = 0	SYSCLK used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	CORECLKSEL = 1	SYSCLK used only for AIF. Clock most be present before the reset to the AIF is removed.
ALTCORECLK	CORECLKSEL = 0	ALTCORECLK is not used and should be tied to a static state.
	CORECLKSEL = 1	ALTCORECLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
PASSCLK	PASSCLKSEL = 0	PASSCLK is not used and should be tied to a static state.
	PASSCLKSEL = 1	PASSCLK is used as a source for the PA_SS PLL. It must be present before the PA_SS PLL is removed from reset and programmed.
SRIOSGMIICLK	An SGMII port will be used.	SRIOSGMIICLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	SGMII will not be used. SRIO will be used as a boot device.	SRIOSGMIICLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	SGMII will not be used. SRIO will be used after boot.	SRIOSGMIICLK is used as a source to the SRIO SERDES PLL. It must be present before the SRIO is removed from reset and programmed.
	SGMII will not be used. SRIO will not be used.	SRIOSGMIICLK is not used and should be tied to a static state.
PCIECLK	PCIE will be used as a boot device.	PCIECLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	PCIE will be used after boot.	PCIECLK is used as a source to the PCIE SERDES PLL. It must be present before the PCIE is removed from reset and programmed.
	PCIE will not be used.	PCIECLK is not used and should be tied to a static state.
MCMCLK	HyperLink will be used as a boot device.	MCMCLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	HyperLink will be used after boot.	MCMCLK is used as a source to the HyperLink SERDES PLL. It must be present before the HyperLink is removed from reset and programmed.
	HyperLink will not be used.	MCMCLK is not used and should be tied to a static state.
End of Table 8-4		

8.3.3 Power Supply Decoupling and Bulk Capacitors

In order 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 Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.3.4 SmartReflex

Increasing the device complexity increases its power consumption and with the smaller transistor structures responsible for higher achievable clock rates and increased performance, comes an inevitable penalty, increasing the leakage currents. Leakage currents are present in any active circuit, independently 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, 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 TMS320TCI6612 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 TCI6612 device.

To guarantee maximizing performance and minimizing power consumption of the device, SmartReflex is required to be implemented whenever the TCI6612 device is used. The voltage selection is done using four VCNTL pins that are used to select the output voltage of the core voltage regulator.

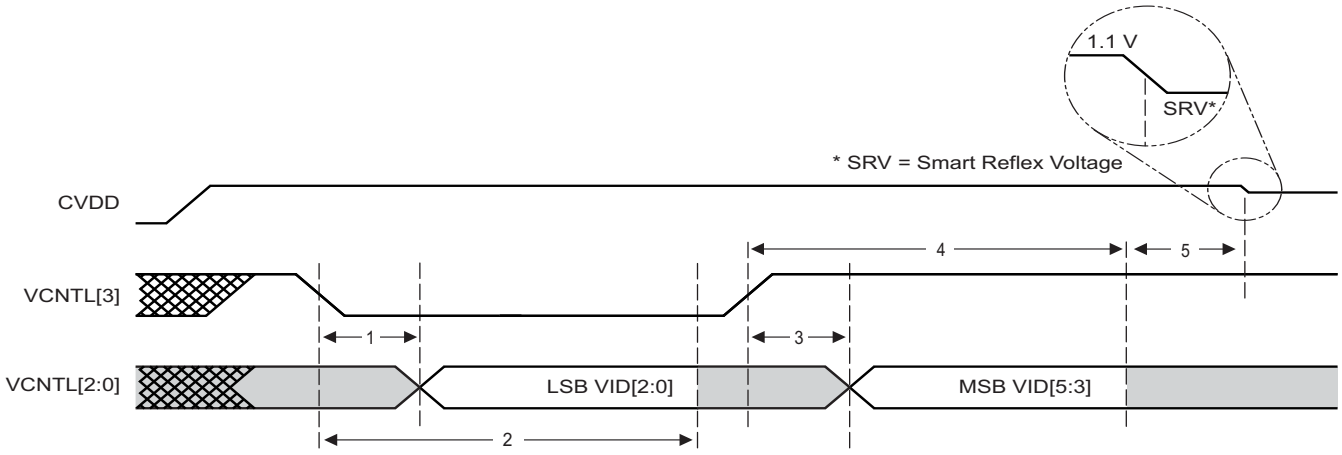
For information on implementation of SmartReflex see the *Power Management for KeyStone Devices* application report and the *Hardware Design Guide for KeyStone Devices* in2.13 "Related Documentation from Texas Instruments" on page 75.

Table 8-5 SmartReflex 4-Pin VID Interface Switching Characteristics
(see Figure 8-3)

No.	Parameter		Min	Max	Unit
1	td(Bn-SELECTL)	Delay time - VCNTL[2:0] (B[2:0]]) valid after VCNTL[3] (select) low		300.00	ns
2	toh(SELECTL-Bn)	Output hold time - VCNTL[2:0] (B[2:0]]) valid after VCNTL[3] (select) low	0.07	172020C (1)	ms
3	td(Bn-SELECTH)	Delay time - VCNTL[2:0] (B[2:0]]) valid after VCNTL[3] (select) high		300.00	ns
4	toh(SELECTH-Bn)	Output hold time - VCNTL[2:0] (B[2:0]]) valid after VCNTL[3] (select) high	0.07	172020C	ms
5		VCNTL being valid to CVDD being switched to SmartReflex voltage (2)		10	ms
End of Table 8-5					

1 C = 1/SYSCLK1 frequency (See Figure 8-9) in ms
2 SmartReflex voltage needs to be set before execution of application code

Figure 8-3 SmartReflex 4-Pin VID Interface Timing



8.4 Power Sleep Controller (PSC)

The Power Sleep Controller (PSC) controls 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 *Power Sleep Controller (PSC) for KeyStone Devices* User Guide in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.4.1 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.

Table 8-6 shows the TMS320TCI6612 power domains.

Table 8-6 Power Domains

Domain	Block(s)	Note	Power Connection
0	Most peripheral logic	Cannot be disabled (includes HyperLink, VCP2_A, GPIO 16-31, ARM interrupt controller)	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	PCIe	Logic can be powered down	Software control
4	SRIO	Logic can be powered down	Software control
5	BCP	Logic can be powered down	Software control
6	ARM CorePac	ARM CorePac reset control	Always on
7	MSMC RAM	MSMC RAM can be powered down	Software control
8	RAC and TAC	Logic can be powered down	Software control
9	FFTC_A and FFTC_B	Logic can be powered down	Software control
10	AIF2	Logic can be powered down	Software control
11	TCP3d_A	RAMs can be powered down	Software control
12	VCP2_B	RAMs can be powered down	Software control
13	C66x CorePac 0, L1/L2 RAMs	Always on	Software control via C66x core. For details, see the C66x CorePac Reference Guide.
14	C66x CorePac 1, L1/L2 RAMs	Logic and L1/L2 RAMs can be powered down	
15	Reserved	Reserved	Reserved
16	Reserved	Reserved	Reserved
17	TCP3d_B	RAMs can be powered down	Software control
18	Reserved	Reserved	Reserved
End of Table 8-6			

8.4.2 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.

Table 8-7 shows the TMS320TCI6612 clock domains.

Table 8-7 Clock Domains

LPSC Number	Module(s)	Notes
0	Shared LPSC for all peripherals other than those listed in this table	Always on
1	SmartReflex	Always on
2	DDR3 EMIF	Always on
3	HyperLink	Software control
4	VCP2_A	Software control
5	Debug subsystem and tracers	Software control
6	Per-core TETB and System TETB	Software control
7	Packet Accelerator	Software control
8	Ethernet SGMIIs	Software control
9	Security Accelerator	Software control
10	PCIe	Software control
11	SRIO	Software control
12	BCP	Software control
13	ARM CorePac reset control	Software control
14	MSMC RAM	Software control
15	RAC	Software control
16	TAC	Software control
17	FFTC_A and FFTC_B	Software control
18	AIF2	Software control
19	TCP3d_A	Software control
20	VCP2_B	Software control
21	Reserved	Reserved
22	Reserved	Reserved
23	C66x CorePac0 and Timer 0	Always on
24	C66x CorePac1 and Timer 1	Always on
25	C66x CorePac0 RSAs	Software control
26	Reserved	Reserved
27	C66x CorePac1 RSAs	Software control
28	Reserved	Reserved
29	TCP3d_B	Software control
30	Reserved	Reserved
No LPSC	Bootcfg, PSC, and PLL controller	Reserved
End of Table 8-7		

8.4.3 PSC Register Memory Map

Table 8-8 shows the PSC Register memory map.

Table 8-8 PSC Register Memory Map (Part 1 of 3)

Offset	Register	Description
0x000	PID	Peripheral Identification Register
0x004 - 0x010	Reserved	Reserved
0x014	VCNTLID	Voltage Control Identification Register (1)
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 (AlwaysOn)
0x204	PDSTAT1	Power Domain Status Register 1 (per-core TETB and System TETB)
0x208	PDSTAT2	Power Domain Status Register 2 (Network Coprocessor)
0x20C	PDSTAT3	Power Domain Status Register 3 (PCIe)
0x210	PDSTAT4	Power Domain Status Register 4 (SRIO)
0x214	PDSTAT5	Power Domain Status Register 5 (BCP)
0x218	PDSTAT6	Power Domain Status Register 6 (Reserved)
0x21C	PDSTAT7	Power Domain Status Register 7 (MSMC RAM)
0x220	PDSTAT8	Power Domain Status Register 8 (RAC and TAC)
0x224	PDSTAT9	Power Domain Status Register 9 (FFTC_A and FFTC_B)
0x228	PDSTAT10	Power Domain Status Register 10 (AIF2)
0x22C	PDSTAT11	Power Domain Status Register 11 (TCP3d_A)
0x230	PDSTAT12	Power Domain Status Register 12 (VCP2_B)
0x234	PDSTAT13	Power Domain Status Register 13 (C66x CorePac0)
0x238	PDSTAT14	Power Domain Status Register 14 (C66x CorePac1)
0x23C	Reserved	Reserved
0x240	Reserved	Reserved
0x244	PDSTAT17	Power Domain Status Register 17 (TCP3d_B)
0x248	Reserved	Reserved
0x24C - 0x2FC	Reserved	Reserved
0x300	PDCTL0	Power Domain Control Register 0 (AlwaysOn)
0x304	PDCTL1	Power Domain Control Register 1 (Per-core TETB and System TETB)
0x308	PDCTL2	Power Domain Control Register 2 (Network Coprocessor)
0x30C	PDCTL3	Power Domain Control Register 3 (PCIe)
0x310	PDCTL4	Power Domain Control Register 4 (SRIO)
0x314	PDCTL5	Power Domain Control Register 5 (BCP)
0x318	PDCTL6	Power Domain Control Register 6 (Reserved)
0x31C	PDCTL7	Power Domain Control Register 7 (MSMC RAM)
0x320	PDCTL8	Power Domain Control Register 8 (RAC and TAC)
0x324	PDCTL9	Power Domain Control Register 9 (FFTC_A and FFTC_B)
0x328	PDCTL10	Power Domain Control Register 10 (AIF2)
0x32C	PDCTL11	Power Domain Control Register 11 (TCP3d_A)
0x330	PDCTL12	Power Domain Control Register 12 (VCP2_B)

Table 8-8 PSC Register Memory Map (Part 2 of 3)

Offset	Register	Description
0x334	PDCTL13	Power Domain Control Register 13 (C66x CorePac0)
0x338	PDCTL14	Power Domain Control Register 14 (C66x CorePac1)
0x33C	Reserved	Reserved
0x340	Reserved	Reserved
0x344	PDCTL17	Power Domain Control Register 17 (TCP3d_B)
0x348	Reserved	Reserved
0x34C - 0x7FC	Reserved	Reserved
0x800	MDSTAT0	Module Status Register 0 (Never Gated)
0x804	MDSTAT1	Module Status Register 1 (SmartReflex)
0x808	MDSTAT2	Module Status Register 2 (DDR3 EMIF)
0x80C	MDSTAT3	Module Status Register 3(Hyperlink)
0x810	MDSTAT4	Module Status Register 4 (VCP2_A)
0x814	MDSTAT5	Module Status Register 5 (debug subsystem and tracers)
0x818	MDSTAT6	Module Status Register 6 (per-core TETB and system TETB)
0x81C	MDSTAT7	Module Status Register 7 (Packet Accelerator)
0x820	MDSTAT8	Module Status Register 8 (Ethernet SGMIIs)
0x824	MDSTAT9	Module Status Register 9 (Security Accelerator)
0x828	MDSTAT10	Module Status Register 10 (PCIe)
0x82C	MDSTAT11	Module Status Register 11 (SRIO)
0x830	MDSTAT12	Module Status Register 12 (BCP)
0x834	MDSTAT13	Module Status Register 13 (Reserved)
0x838	MDSTAT14	Module Status Register 14 (MSMC RAM)
0x83C	MDSTAT15	Module Status Register 15 (RAC)
0x840	MDSTAT16	Module Status Register 16 (TAC)
0x844	Reserved	Reserved
0x848	MDSTAT18	Module Status Register 18 (AIF2)
0x84C	MDSTAT19	Module Status Register 19 (TCP3d_A)
0x850	MDSTAT20	Module Status Register 20 (VCP2_B)
0x854	Reserved	Reserved
0x858	Reserved	Reserved
0x85C	MDSTAT23	Module Status Register 23 (C66x CorePac0 and Timer 0)
0x860	MDSTAT24	Module Status Register 24 (C66x CorePac1and Timer 1)
0x864	MDSTAT25	Module Status Register 25 (C66x CorePac0 RSAs)
0x868	Reserved	Reserved
0x86C	MDSTAT27	Module Status Register 27 (C66x CorePac1 RSAs)
0x870	Reserved	Reserved
0x874	MDSTAT29	Module Status Register 29 (TCP3d_B)
0x878	Reserved	Reserved
0x87C - 0x9FC	Reserved	Reserved
0xA00	MDCTL0	Module Control Register 0 (Never Gated)
0xA04	MDCTL1	Module Control Register 1 (SmartReflex)
0xA08	MDCTL2	Module Control Register 2 (DDR3 EMIF)
0xA0C	MDCTL3	Module Control Register 3(Hyperlink)
0xA10	MDCTL4	Module Control Register 4 (VCP2_A)

Table 8-8 PSC Register Memory Map (Part 3 of 3)

Offset	Register	Description
0xA14	MDCTL5	Module Control Register 5 (debug subsystem and tracers)
0xA18	MDCTL6	Module Control Register 6 (Per-core TETB and System TETB)
0xA1C	MDCTL7	Module Control Register 7 (Packet Accelerator)
0xA20	MDCTL8	Module Control Register 8 (Ethernet SGMIIs)
0xA24	MDCTL9	Module Control Register 9 (Security Accelerator)
0xA28	MDCTL10	Module Control Register 10 (PCIe)
0xA2C	MDCTL11	Module Control Register 11 (SRIO)
0xA30	MDCTL12	Module Control Register 12 (BCP)
0xA34	MDCTL13	Module Control Register 13 (Reserved)
0xA38	MDCTL14	Module Control Register 14 (MSMC RAM)
0xA3C	MDCTL15	Module Control Register 15 (RAC)
0xA40	MDCTL16	Module Control Register 16 (TAC)
0xA44	MDCTL17	Module Control Register 17 (FFTC_A and FFTC_B)
0xA48	MDCTL18	Module Control Register 18 (AIF2)
0xA4C	MDCTL19	Module Control Register 19 (TCP3d_A)
0xA50	MDCTL20	Module Control Register 20 (VCP2_B)
0xA54	Reserved	Reserved
0xA58	Reserved	Reserved
0xA5C	MDCTL23	Module Control Register 23 (C66x CorePac0 and Timer 0)
0xA60	MDCTL24	Module Control Register 24 (C66x CorePac1and Timer 1)
0xA64	MDCTL25	Module Control Register 25 (C66x CorePac0 RSAs)
0xA68	Reserved	Reserved
0xA6C	MDCTL27	Module Control Register 27 (C66x CorePac1 RSAs)
0xA70	Reserved	Reserved
0xA74	MDCTL29	Module Control Register 29 (TCP3d_B)
0xA78	Reserved	Reserved
0xA7C - 0xFFC	Reserved	Reserved
End of Table 8-8		

1 VCNTLID register is available for debug purpose only.

8.5 Reset Controller

The reset controller detects the different type of resets supported on the TMS320TCI6612 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 8-9 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 8.5.7 "Reset Electrical Data/Timing" on page 144.

Table 8-9 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 and ARM CorePac. The device configuration pins are latched only during power-on reset.
Hard reset	$\overline{\text{RESET}}$ pin PLLCTL [1] register (RSCTRL) Watchdog timers Emulation	Hard reset resets everything except for the ARM interrupt controller, test, emulation logic and reset isolation modules. This reset is also different from power-on reset in that the PLLCTL assumes power and clocks are stable when hard reset is asserted. The device configurations pins are not re-latched. Emulation initiated reset is always a hard reset. By default these initiators are configured as hard reset, but can be configured (except emulation) as soft reset in the RSCFG register of PLLCTL. Contents of 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 will behave like hard reset except that EMIF16 MMRs, DDR3 EMIF MMRs, the sticky bits in PCIe MMRs, and external memory contents are retained. By default these initiators are configured as hard reset, but can be configured as soft reset in the RSCFG register of PLLCTL. Contents of 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 CorePac, without disturbing clock alignment or memory contents. The device configuration pins are not re-latched.
End of Table 8-9		

1 All masters in the device have access to the PLLCTL registers.

8.5.1 Power-on Reset

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

Power-on reset is initiated by the following

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

During power-up, the $\overline{\text{POR}}$ pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. A $\overline{\text{RESETFULL}}$ pin is also provided to allow the on-board host to reset the entire device including the reset isolated logic. The assumption is that, device is already powered up and hence unlike $\overline{\text{POR}}$, $\overline{\text{RESETFULL}}$ pin will be driven by the on-board host control other than the power good circuitry. For power-on reset, the Main 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 pin asserted (driven low). While POR is asserted, all pins except RESETSTAT will be set to high-impedance. After the POR pin is de-asserted (driven high), all Z group pins, low group pins, and high group pins are set to their reset state and will remain at 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 Table 3-2 "Device State Control Registers" on page 77).
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 will be driven low indicating that the device is in reset.
3. POR must be held active until all supplies on the board are stable then for at least an additional time 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. The chip-level PLLs is taken out of reset and begins its locking sequence, and all power-on device initialization also begins.
5. After device initialization is complete, the RESETSTAT pin is de-asserted (driven high). By this time, DDR3 PLL has already completed its locking sequence and is outputting a valid clock. The system clocks of both PLL controllers are allowed to finish their current cycles and then 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 device 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 together with the POR pin.

8.5.2 Hard Reset

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

Hard reset is initiated by the following:

- RESET pin
- RSCTRL register in PLLCTL
- Watchdog timer
- Emulation

All the above initiators by default are configured to act as hard reset. Except emulation, all the other 3 initiators can be configured as soft resets in the RSCFG register in PLLCTL.

The following sequence must be followed during a Hard reset:

1. The RESET pin is pulled active low for a minimum of 24 CLKIN1 cycles. During this time, the RESET signal is able to propagate to all modules (except those specifically mentioned above). All I/O are Hi-Z for modules affected by RESET, to prevent off-chip contention during the warm reset.
2. Once all logic is reset, RESETSTAT is driven active 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 together with the POR pin.

8.5.3 Soft Reset

A soft reset will behave like a hard reset except that EMIF16 MMRs, DDR3 EMIF MMRs, PCIe MMRs sticky bits, and external memory contents are retained. POR should also remain de-asserted during this time.

Soft reset is initiated by the following

- RESET pin
- RSCTRL register in PLLCTL
- Watchdog timer
- Emulation

All the above initiators by default are configured to act as hard reset. Except emulation, all the other 3 initiators can be configured as soft resets in the RSCFG register in PLLCTL.

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

During a soft reset, the following happens:

1. The RESETSTAT pin goes low to indicate an internal reset is being generated. The reset is allowed to propagate through the system. Internal system clocks are not affected. PLLs also remain locked.
2. After device initialization is complete, the RESETSTAT pin is deasserted (driven high). In addition, the PLL controllers pause their system clocks for about 8 cycles.

   At this point:

   - The state of the peripherals before the soft reset is not changed.
   - The I/O pins are controlled as dictated by the DEVSTAT register.
   - The DDR3 MMRs and the PCIe MMRs sticky bits retain their previous values. Only the DDR3 Memory Controller and PCIe state machines are reset by the soft reset.
   - The PLL controllers are operating in the mode prior to soft reset. System clocks are unaffected.

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

8.5.4 Local Reset

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

Local reset is initiated by the following (for more details see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75):

- $\overline{\text{LRESET}}$ 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 "Reset Configuration Register (RSTCFG)" on page 154 and "CIC Registers" on page 196.
  - Local reset
  - NMI
  - NMI followed by a time delay and then a local reset for the core selected
  - Hard reset by requesting reset via PLLCTL
- LPSC MMRs

8.5.5 Reset Priority

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

- Power-on reset
- Hard/soft reset

8.5.6 Reset Controller Register

The Reset Controller Registers are part of the PLLCTL MMRs. All TCI6612 device-specific MMRs are covered in Section 8.6.2 "PLL Controller Memory Map" on page 149. For more details on these registers and how to program them, see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.5.7 Reset Electrical Data/Timing

Table 8-10 Reset Timing Requirements [1] [2]
(see Figure 8-4 and Figure 8-5)

No.			Min	Max	Unit
		RESETFULL Pin Reset			
1	tw(RESETFULL)	Pulse width - pulse width RESETFULL low	500C		ns
		Soft/Hard-Reset			
2	tw(RESET)	Pulse width - pulse width RESET low	500C		ns
End of Table 8-10					

1 If CORECLKSEL = 0, C = 1 ÷ CORECLK(N|P) frequency in ns.
2 If CORECLKSEL = 1, C = 1 ÷ ALTCORECLK frequency in ns.

Table 8-11 Reset Switching Characteristics Over Recommended Operating Conditions [1] [2]
(see Figure 8-4 and Figure 8-5)

No.	Parameter		Min	Max	Unit
		RESETFULL Pin Reset			
3	td(RESETFULLH-RESETSTATH)	Delay time - RESETSTAT high after RESETFULL high		50000C	ns
		Soft/Hard Reset			
4	td(RESETH-RESETSTATH)	Delay time - RESETSTAT high after RESET high		50000C	ns
End of Table 8-11					

1 If CORECLKSEL = 0, C = 1 ÷ CORECLK(N|P) frequency in ns.
2 If CORECLKSEL = 1, C = 1 ÷ ALTCORECLK frequency in ns.

Figure 8-4 RESETFULL Reset Timing

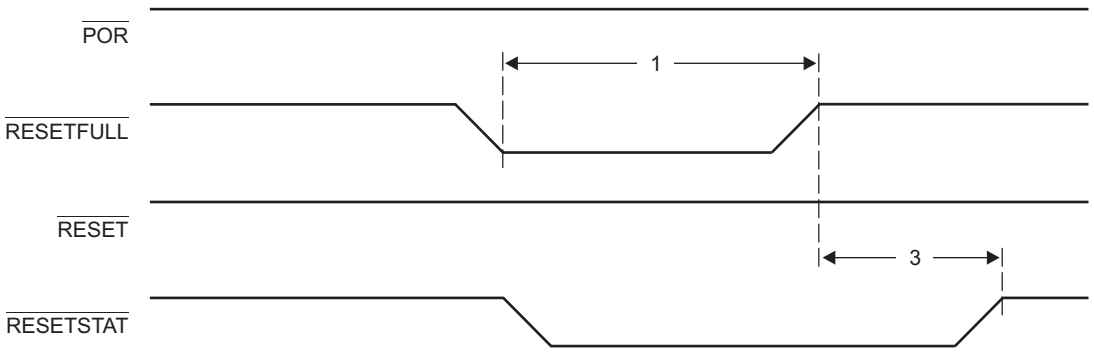


Figure 8-5 Soft/Hard Reset Timing

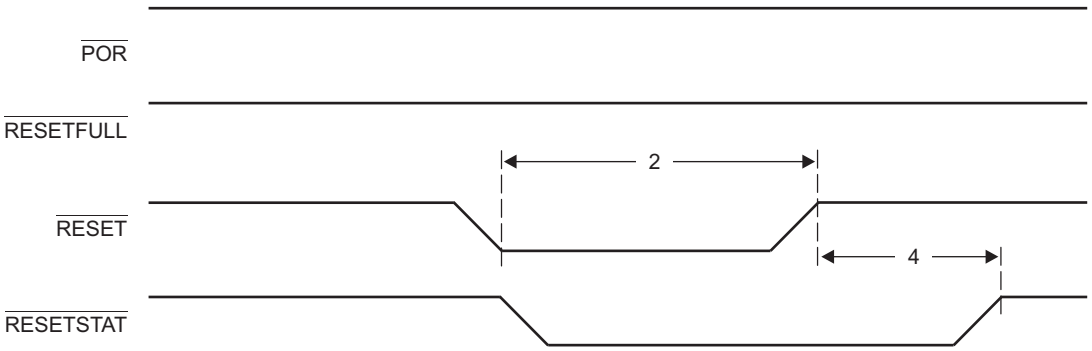
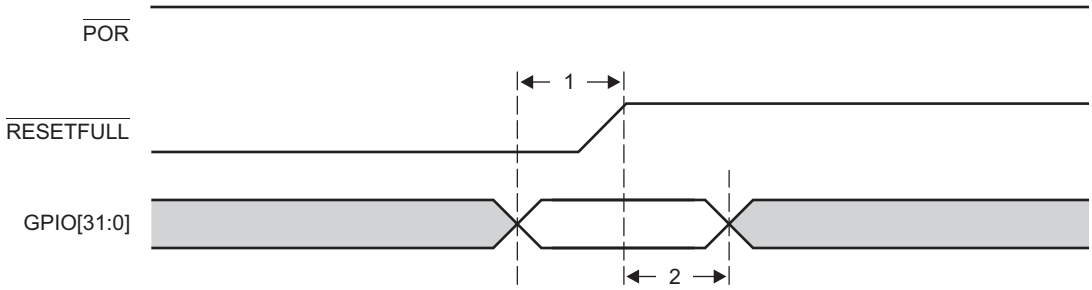


Table 8-12 Boot Configuration Timing Requirements (1) (2)
See Figure 8-6)

No.			Min	Max	Unit
1	tsu(GPIOn-$\overline{\text{RESETFULL}}$)	Setup time - GPIO valid before $\overline{\text{RESETFULL}}$ asserted	12C		ns
2	th($\overline{\text{RESETFULL}}$-GPIOn)	Hold time - GPIO valid after $\overline{\text{RESETFULL}}$ asserted	12C		ns
End of Table 8-12					

1 If CORECLKSEL = 0, C = 1 ÷ CORECLK(N|P) frequency in ns.
2 If CORECLKSEL = 1, C = 1 ÷ ALTCORECLK frequency in ns.

Figure 8-6 Boot Configuration Timing

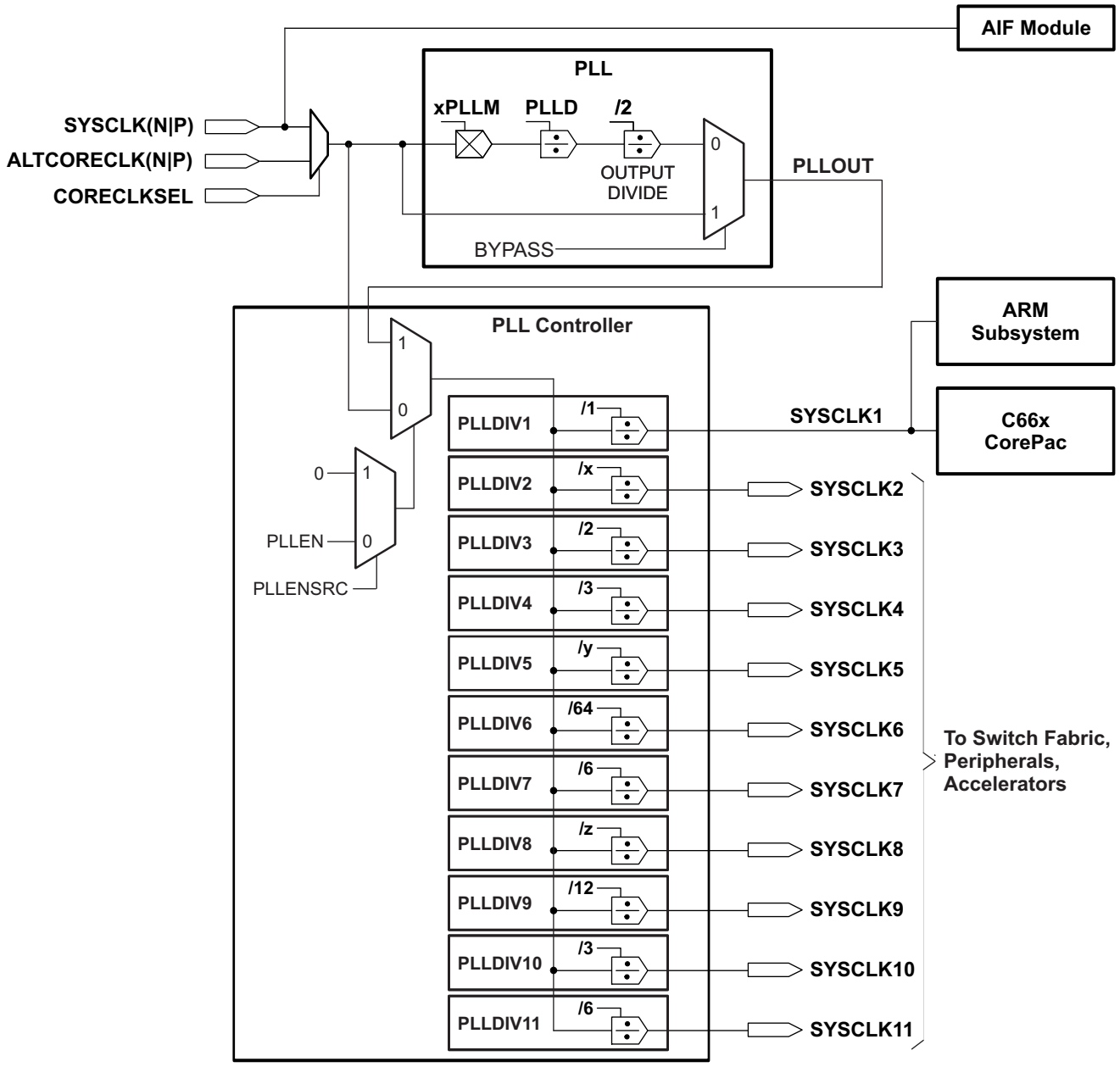


8.6 Main PLL and the PLL Controller

This section provides a description of the Main PLL and the PLL Controller. For details on the operation of the PLL Controller module, see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

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. Figure 8-7 shows a block diagram of the main PLL and the PLL Controller.

Figure 8-7 Main PLL and PLL Controller



Note—The Main PLL controller registers can be accessed by any master in the device. PLLM[5:0] bits of the multiplier are controlled by the PLLM register inside the PLL controller and PLLM[12:6] bits are controlled by the chip level MAINPLLCTL0 register. The Output Divide and Bypass logic of the PLL are controlled by bit-fields in SECCTL register in the PLL Controller. Only PLLDIV2, PLLDIV5, and PLLDIV8 are programmable on the TCI6612 device. See the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in section 2.13 "Related Documentation from Texas Instruments" on page 75 for more details on how to program the PLL controller.

The inputs, multiply and division factor within the PLL, post-division for each of the chip-level clocks is achieved using the combination of this PLL and the PLL Controller. The PLL Controller also controls reset propagation through the chip, clock alignment, and test points. The 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 Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75 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 8.6.5 "Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Electrical Data/Timing".

CAUTION—The PLL controller module as described in the see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75 includes a superset of features, some of which are not supported on the TMS320TCI6612 device. The following sections describe the registers that are supported; it should be assumed that any registers not included in these sections is 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.

8.6.1 Main PLL Controller Device-Specific Information

8.6.1.1 Internal Clocks and Maximum Operating Frequencies

The Main PLL, used to drive the CorePacs, the switch fabric, and a majority of the peripheral clocks (all but the DDR3 and the PASS modules) requires a PLL controller to manage the various clock divisions, gating, and synchronization. The Main PLL's PLL Controller has several SYSCLK outputs that are listed below, along with the clock description. 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 CorePac0~CorePac1, ARM, RAC, and RSA.
- **SYSCLK2:** 1/x-rate clock for CorePac (emulation). Also used for the ARM CorePac. Default rate for this is 1/3. This is programmable from /1 to /32, where this clock does not violate the max of 350 MHz. The SYSCLK2 can be turned off by software.
- **SYSCLK3:** 1/2-rate clock used to clock the MSMC, TCP3d, HyperLink, CPU/2 TeraNet, DDR EMIF and CPU/2 EDMA.
- **SYSCLK4:** 1/3-rate clock for the switch fabrics and fast peripherals. The Debug_SS and ETBs use this as well.
- **SYSCLK5:** 1/y-rate clock for system trace module only. Default rate for this is 1/5. It is configurable and the max configurable clock is 210 MHz and min configuration clock is 32 MHz. The SYSCLK5 can be turned off by software.
- **SYSCLK6:** 1/64-rate clock. 1/64 rate clock (emif_ptv) used to clock the PVT compensated buffers for DDR3 EMIF.

- **SYSCLK7:** 1/6-rate clock for slow peripherals like UART and sources the SYSCLKOUT output pin.
- **SYSCLK8:** 1/z-rate clock. This clock is used as slow_sysclck in the system. Default for this is 1/64. This is programmable from /24 to /80.
- **SYSCLK9:** 1/12-rate clock for SmartReflex.
- **SYSCLK10:** 1/3-rate clock for SRIO only.
- **SYSCLK11:** 1/6-rate clock for PSC only.

Only SYSCLK2, SYSCLK5, and SYSCLK8 are programmable on TMS320TCI6612 device.

Note—In case any of the other programmable SYSCLKs are set slower than 1/64 rate, then SYSCLK8 (SLOW_SYSCLK) needs to be programmed to either match, or be slower than, the slowest SYSCLK in the system.

8.6.1.2 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 PLL mode, SYSCLK1 is generated from the PLL output using the values set in PLLM and PLLD fields in the MAINPLLCTL0 register. In bypass mode, PLL input is fed directly out as SYSCLK1.

All hosts must hold off accesses to the SoC while the frequency of its internal clocks is changing. A mechanism must be in place such that the SoC notifies the host when the PLL configuration has completed.

8.6.1.3 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 powerup. The PLL should not be operated 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 8-13.

The PLL lock time is the amount of time needed from when the PLL is taken out of reset (PLLRST = 1 with PLLEN = 0) to when to when the PLL Controller can be switched to PLL mode (PLLEN = 1). The Main PLL lock time is given in Table 8-13.

Table 8-13 Main PLL Stabilization, Lock, and Reset Times

	Min	Typ	Max	Unit
PLL stabilization time	100			µs
PLL lock time			500 × C (1)	
PLL reset time	1000			ns
End of Table 8-13				

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

8.6.2 PLL Controller Memory Map

The memory map of the PLL Controller is shown in Table 8-14. TMS320TCI6612-specific PLL Controller register definitions can be found in the sections following Table 8-14, for other registers in the table, see the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

CAUTION—Note that only the registers documented here are accessible on the TMS320TCI6612. Other addresses in the PLL controller memory map including the reserved registers should not be modified. 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. It is recommended to use read-modify-write sequence to make any changes to the valid bits in the register.

Table 8-14 PLL Controller Registers (Including Reset Controller) (Part 1 of 2)

Hex Address Range	Field	Register Name
0231 0000 - 0231 00E3	-	Reserved
0231 00E4	RSTYPE	Reset Type Status Register (Reset Controller)
0231 00E8	RSTCTRL	Software Reset Control Register (Reset Controller)
0231 00EC	RSTCFG	Reset Configuration Register (Reset Controller)
0231 00F0	RSISO	Reset isolation register (Reset Controller)
0231 00F0 - 0231 00FF	-	Reserved
0231 0100	PLLCTL	PLL Control Register
0231 0104	-	Reserved
0231 0108	SECCTL	PLL Secondary Control Register
0231 010C	-	Reserved
0231 0110	PLLM	PLL Multiplier Control Register
0231 0114	-	Reserved
0231 0118	PLLDIV1	Reserved
0231 011C	PLLDIV2	PLL controller divider 2 register
0231 0120	PLLDIV3	Reserved
0231 0124	-	Reserved
0231 0128	-	Reserved
0231 012C - 0231 0134	-	Reserved
0231 0138	PLLCMD	PLL Controller Command Register
0231 013C	PLLSTAT	PLL Controller Status Register
0231 0140	ALNCTL	PLL Controller Clock Align Control Register
0231 0144	DCHANGE	PLLDIV Ratio Change Status Register
0231 0148	CKEN	Reserved
0231 014C	CKSTAT	Reserved
0231 0150	SYSTAT	SYSCLK Status Register
0231 0154 - 0231 015C	-	Reserved
0231 0160	PLLDIV4	Reserved
0231 0164	PLLDIV5	PLL Controller Divider 5 Register
0231 0168	PLLDIV6	Reserved
0231 016C	PLLDIV7	Reserved
0231 0170	PLLDIV8	PLL Controller Divider 8 Register

Table 8-14 PLL Controller Registers (Including Reset Controller) (Part 2 of 2)

Hex Address Range	Field	Register Name
0231 0174 - 0231 0193	PLLDIV9 - PLLDIV16	Reserved
0231 0194 - 0231 01FF	-	Reserved
End of Table 8-14		

8.6.2.1 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register contains extra fields to control the Main PLL and is shown in Figure 8-8 and described in Table 8-15.

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

31 24	23	22 19	18 0
Reserved	BYPASS	OUTPUT_DIVIDE	Reserved
R-0000 0000	RW-0	RW-0001	RW-001 0000 0000 0000 0000

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

Table 8-15 PLL Secondary Control Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Reserved
23	BYPASS	Main PLL bypass enable 0 = Main PLL bypass disabled 1 = Main PLL bypass enabled
22-19	OUTPUT_DIVIDE	Output divider ratio bits 0h = ÷1. Divide frequency by 1. 1h = ÷2. Divide frequency by 2. 2h - Fh = Reserved.
18-0	Reserved	Reserved
End of Table 8-15		

8.6.2.2 PLL Controller Divider Register (PLLDIV2, PLLDIV5, and PLLDIV8)

The PLL Controller divider registers (PLLDIV2, PLLDIV5, and PLLDIV8) are shown in Figure 8-9 and described in Table 8-16. The default values of the RATIO field on a reset for PLLDIV2, PLLDIV5, and PLLDIV8 are different and mentioned in the footnote of Figure 8-9.

Figure 8-9 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 D2EN for PLLDIV2; D5EN for PLLDIV5; D8EN for PLLDIV8
2 n=02h for PLLDIV2; n=04h for PLLDIV5; n=3Fh for PLLDIV8

Table 8-16 PLL Controller Divider Register (PLLDIVn) Field Descriptions

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

8.6.2.3 PLL Controller Clock Align Control Register (ALNCTL)

The PLL Controller Clock Align Control Register (ALNCTL) is shown in Figure 8-10 and described in Table 8-17.

Figure 8-10 PLL Controller Clock Align Control Register (ALNCTL)

31	8	7	6	5	4	3	2	1	0
Reserved		ALN8	Reserved		ALN5	Reserved		ALN2	Reserved
R-0		R/W-1	R-0		R/W-1	R-0		R/W-1	R-0

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

Table 8-17 PLL Controller Clock Align Control Register Field Descriptions

Bit	Field	Description
31-8 6-5 3-2 0	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7 4 1	ALN8 ALN5 ALN2	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*.
End of Table 8-17		

8.6.2.4 PLLDIV Divider Ratio Change Status Register (DCHANGE)

Whenever a different ratio is written to the PLLDIV*n* registers, the PLLCTL flags the change in the DCHANGE status register. During the GO operation, the PLL controller will change 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 8-11 and described in Table 8-18.

Figure 8-11 PLLDIV Divider Ratio Change Status Register (DCHANGE)

31	8	7	6	5	4	3	2	1	0
Reserved		SYS8	Reserved		SYS5	Reserved		SYS2	Reserved
R-0		R/W-0	R-0		R/W-0	R-0		R/W-0	R-0

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

Table 8-18 PLLDIV Divider Ratio Change Status Register Field Descriptions

Bit	Field	Description
31-8 6-5 3-2 0	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7 4 1	SYS8 SYS5 SYS2	Indicates 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.
End of Table 8-18		

8.6.2.5 SYSCLK Status Register (SYSTAT)

The SYSCLK Status Register (SYSTAT) shows the status of SYSCLK[11:1]. SYSTAT is shown in Figure 8-12 and described in Table 8-19.

Figure 8-12 SYSCLK Status Register (SYSTAT)

31	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SYS11ON	SYS10ON	SYS9ON	SYS8ON	SYS7ON	SYS6ON	SYS5ON	SYS4ON	SYS3ON	SYS2ON	SYS1ON
R-n		R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1

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

Table 8-19 SYSCLK Status Register Field Descriptions

Bit	Field	Description
31-11	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
10-0	SYS[N [(1)]]ON	SYSCLK[N] on status 0 = SYSCLK[N] is gated 1 = SYSCLK[N] is on
End of Table 8-19		

1 Where N = 1, 2, 3,....N (Not all these output clocks may be used on a specific device. For more information, see the device-specific data manual)

8.6.2.6 Reset Type Status Register (RSTYPE)

The Reset Type Status (RSTYPE) Register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The Reset Type Status Register is shown in Figure 8-13 and described in Table 8-20.

Figure 8-13 Reset Type Status Register (RSTYPE)

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

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

Table 8-20 Reset Type Status Register Field Descriptions

Bit	Field	Description
31-29	Reserved	Reserved. Read only. 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. Read only. 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. Read only. 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
End of Table 8-20		

8.6.2.7 Reset Control Register (RSTCTRL)

This register contains a key that enables writes to the MSB of this register and the RSTCFG register. The key value is 0x5A69. A valid key will be stored as 0x000C, any other key value is invalid. When the RSTCTRL or the RSTCFG is written, the key is invalidated. Every write must be set up with a valid key. The Software Reset Control Register (RSTCTRL) is shown in Figure 8-14 and described in Table 8-21.

Figure 8-14 Reset Control Register (RSTCTRL)

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

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

1 Writes are conditional based on valid key.

Table 8-21 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.
End of Table 8-21		

8.6.2.8 Reset Configuration Register (RSTCFG)

This register is used to configure the type of reset initiated by $\overline{\text{RESET}}$, watchdog timer and the PLL Controller's RSTCTRL register; i.e., a hard reset or a soft reset. By default, these resets will be hard resets. The Reset Configuration Register (RSTCFG) is shown in Figure 8-15 and described in Table 8-22.

Figure 8-15 Reset Configuration Register (RSTCFG)

31 16	15 14	13	12	11 4	3 0
Reserved	Reserved	PLLCTLRSTTYPE	$\overline{\text{RESET}}$TYPE	Reserved	WDTYPE[N (1)
R-0x0000	R-00	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 output may be used on a specific device. For more information, see the device-specific data manual)
2 Writes are conditional based on valid key. For details, see Section 8.6.2.7 "Reset Control Register (RSTCTRL)".

Table 8-22 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
End of Table 8-22		

8.6.2.9 Reset Isolation Register (RSISO)

This register is used to select the module clocks that must maintain their clocking without pausing through non power-on reset. Setting any of these bits effectively blocks reset to all PLLCTL registers in order to maintain current values of PLL multiplier, divide ratios, and other settings. Along with setting 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 MDCTLx register see the *Power Sleep Controller (PSC) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75. The Reset Isolation Register (RSTCTRL) is shown in Figure 8-16 and described in Table 8-23.

Figure 8-16 Reset Isolation Register (RSISO)

31 16	15 10	9	8	7 4	3	2 0
Reserved	Reserved	SRIOISO	SRISO	Reserved	AIF2ISO	Reserved
R-0x0000	R-0x00	R/W-0	R/W-0	R-0x0	R/W-0	R-000

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

Table 8-23 Reset Isolation Register Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved
9	SRIOISO	Isolate SRIO module 0 = Not reset isolated 1 = Reset isolated
8	SRISO	Isolate SmartReflex 0 = Not reset isolated 1 = Reset isolated
7-4	Reserved	Reserved
3	AIF2ISO	Isolate AIF2 module 0 = Not reset isolated 1 = Reset isolated
2-0	Reserved	Reserved
End of Table 8-23		

8.6.3 Main PLL Control Registers

The Main PLL uses two chip-level registers (MAINPLLCTL0 and MAINPLLCTL1) along with the 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 un-locking sequence using KICK0/KICK1 registers. For valid configurable values into the MAINPLLCTL register see Section 2.7 "PLL Settings" on page 40. See 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 82 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers reset only on a $\overline{\text{POR}}$ reset. See Figure 8-17 and Table 8-24 for MAINPLLCTL0 details and Figure 8-18 and Table 8-25 for MAINPLLCTL1 details.

Figure 8-17 Main PLL Control Register (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 8-24 Main PLL Control Register Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ should be programmed to a value equal to half of PLLM[12:0]. Example: PLLM = 15, then BWADJ = 7
23-19	Reserved	Reserved
18-12	PLLM[12:6]	A 13-bit field that selects the values for the multiplication factor (see note below)
11-6	Reserved	Reserved
5-0	PLLD	A 6-bit field that selects the values for the reference divider
End of Table 8-24		

Figure 8-18 Main PLL Control Register (MAINPLLCTL1)

31	7	6	5	4	3	0
Reserved		ENSAT	Reserved		BWADJ[11:8]	
RW - 00000000000000000000000000		RW - 0	RW- 00		RW- 0000	

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

Table 8-25 Main PLL Control Register (MAINPLLCTL1) Field Descriptions

Bit	Field	Description
31-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper operation of the Main PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in the MAINPLLCTL0 and MAINPLLCTL1 registers. The combination (BWADJ[11:0]) should be programmed to a value equal to half of PLLM[12:0] if PLLM has even values or to be rounded half down of PLLM[12:0] if PLLM has odd values. Example: PLLM=15, then BWADJ=7
End of Table 8-25		

Note—PLLM[5:0] bits of the multiplier is controlled by the PLLM register inside the PLL controller and 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 *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75 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 the 8.6.2.1 "PLL Secondary Control Register (SECCTL)" on page 150 for more details.

8.6.4 Main PLL and PLL Controller Initialization Sequence

See the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 75 for details on the initialization sequence for Main PLL and PLL Controller.

8.6.5 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Electrical Data/Timing

Table 8-26 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (1) (Part 1 of 2)
(see Figure 8-19 and Figure 8-20)

No.			Min	Max	Unit
		SYSCLK[P:N]			
1	tc(SYSCLKN)	Cycle time _ SYSCLKN cycle time	3.25 or 6.51 or 8.138 (2)		ns
1	tc(SYSCLKP)	Cycle time _ SYSCLKP cycle time	3.25 or 6.51 or 8.138 (2)		ns
3	tw(SYSCLKN)	Pulse width _ SYSCLKN high	0.45*tc(SYSCLKN)	0.55*tc(SYSCLKN)	ns
2	tw(SYSCLKN)	Pulse width _ SYSCLKN low	0.45*tc(SYSCLKN)	0.55*tc(SYSCLKN)	ns
2	tw(SYSCLKP)	Pulse width _ SYSCLKP high	0.45*tc(SYSCLKP)	0.55*tc(SYSCLKP)	ns
3	tw(SYSCLKP)	Pulse width _ SYSCLKP low	0.45*tc(SYSCLKP)	0.55*tc(SYSCLKP)	ns
4	tr(SYSCLKN_250 mv)	Transition time _ SYSCLKN Rise time (250 mV)	50	350	ps
4	tf(SYSCLKN_250 mv)	Transition time _ SYSCLKN Fall time (250 mV)	50	350	ps
4	tr(SYSCLKP_250 mv)	Transition time _ SYSCLKP Rise time (250 mV)	50	350	ps
4	tf(SYSCLKP_250 mv)	Transition time _ SYSCLKP Fall time (250 mV)	50	350	ps
5	tj(SYSCLKN)	Jitter, peak_to_peak _ periodic SYSCLKN		0.02*tc(SYSCLKN) (3)	ps
5	tj(SYSCLKP)	Jitter, peak_to_peak _ periodic SYSCLKP		0.02*tc(SYSCLKN) (3)	ps
		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(ALTCORECLKN_250 mv)	Transition time _ ALTCORECLKN rise time (250 mV)	50	350	ps
4	tf(ALTCORECLKN_250 mv)	Transition time _ ALTCORECLKN Fall time (250 mV)	50	350	ps
4	tr(ALTCORECLKP_250 mv)	Transition time _ ALTCORECLKP rise time (250 mV)	50	350	ps
4	tf(ALTCORECLKP_250 mv)	Transition time _ ALTCORECLKP fall time (250 mV)	50	350	ps
5	tj(ALTCORECLKN)	Jitter, peak_to_peak _ periodic ALTCORECLKN		0.02*tc(ALTCORECLKN)	ps
5	tj(ALTCORECLKP)	Jitter, peak_to_peak _ periodic ALTCORECLKP		0.02*tc(ALTCORECLKP)	ps
		SRIOSGMIICLK[P:N]			
1	tc(SRIOSGMIICLKN)	Cycle time _ SRIOSGMIICLKN cycle time	3.2 or 4 or 6.4		ns
1	tc(SRIOSGMIICLKP)	Cycle time _ SRIOSGMIICLKP cycle time	3.2 or 4 or 6.4		ns
3	tw(SRIOSGMIICLKN)	Pulse width _ SRIOSGMIICLKN high	0.45*tc(SRIOSGMIICLKN)	0.55*tc(SRIOSGMIICLKN)	ns
2	tw(SRIOSGMIICLKN)	Pulse width _ SRIOSGMIICLKN low	0.45*tc(SRIOSGMIICLKN)	0.55*tc(SRIOSGMIICLKN)	ns
2	tw(SRIOSGMIICLKP)	Pulse width _ SRIOSGMIICLKP high	0.45*tc(SRIOSGMIICLKP)	0.55*tc(SRIOSGMIICLKP)	ns
3	tw(SRIOSGMIICLKP)	Pulse width _ SRIOSGMIICLKP low	0.45*tc(SRIOSGMIICLKP)	0.55*tc(SRIOSGMIICLKP)	ns
4	tr(SRIOSGMIICLKN_250 mv)	Transition time _ SRIOSGMIICLKN rise time(250 mV)	50	350	ps
4	tf(SRIOSGMIICLKN_250 mv)	Transition time _ SRIOSGMIICLKN fall time (250 mV)	50	350	ps
4	tr(SRIOSGMIICLKP_250 mv)	Transition time _ SRIOSGMIICLKP rise time(250 mV)	50	350	ps
4	tf(SRIOSGMIICLKP_250 mv)	Transition time _ SRIOSGMIICLKP fall time (250 mV)	50	350	ps
5	tj(SRIOSGMIICLKN)	Jitter, peak_to_peak _ periodic SRIOSGMIICLKN		4 (3)	ps,RMS
5	tj(SRIOSGMIICLKP)	Jitter, peak_to_peak _ periodic SRIOSGMIICLKP		4 (3)	ps,RMS
5	tj(SRIOSGMIICLKN)	Jitter, peak_to_peak _ periodic SRIOSGMIICLKN (SRIO not used)		8 (3)	ps,RMS

Table 8-26 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing Requirements (1) (Part 2 of 2)
(see Figure 8-19 and Figure 8-20)

No.			Min	Max	Unit
5	tj(SRIOSGMIICLKP)	Jitter, peak_to_peak _ periodic SRIOSGMIICLKP (SRIO not used)		8 (3)	ps,RMS
		HyperLink CLK[P:N]			
1	tc(MCMCLKN)	Cycle Time _ MCMCLKN cycle time	3.2 or 4 or 6.4		ns
1	tc(MCMCLKP)	Cycle Time _ MCMCLKP cycle time	3.2 or 4 or 6.4		ns
3	tw(MCMCLKN)	Pulse Width _ MCMCLKN high	0.45*tc(MCMCLKN)	0.55*tc(MCMCLKN)	ns
2	tw(MCMCLKN)	Pulse Width _ MCMCLKN low	0.45*tc(MCMCLKN)	0.55*tc(MCMCLKN)	ns
2	tw(MCMCLKP)	Pulse width _ MCMCLKP high	0.45*tc(MCMCLKP)	0.55*tc(MCMCLKP)	ns
3	tw(MCMCLKP)	Pulse width _ MCMCLKP low	0.45*tc(MCMCLKP)	0.55*tc(MCMCLKP)	ns
4	tr(MCMCLKN_250 mv)	Transition time _ MCMCLKN rise time (250 mV)	50	350	ps
4	tf(MCMCLKN_250 mv)	Transition Time _ MCMCLKN fall time (250 mV)	50	350	ps
4	tr(MCMCLKP_250 mv)	Transition Time _ MCMCLKP rise time (250 mV)	50	350	ps
4	tf(MCMCLKP_250 mv)	Transition Time _ MCMCLKP fall time (250 mV)	50	350	ps
5	tj(MCMCLKN)	Jitter, peak_to_peak _ periodic MCMCLKN		4 (3)	ps,RMS
5	tj(MCMCLKP)	Jitter, peak_to_peak _ periodic MCMCLKP		4 (3)	ps,RMS
		PCIECLK[P:N]			
1	tc(PCIECLKN)	Cycle time _ PCIECLKN cycle time	3.2 or 4 or 6.4 or 10		ns
1	tc(PCIECLKP)	Cycle time _ PCIECLKP cycle time	3.2 or 4 or 6.4 or 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(PCIECLKN_250 mv)	Transition time _ PCIECLKN rise time (250 mV)	50	350	ps
4	tf(PCIECLKN_250 mv)	Transition time _ PCIECLKN fall time (250 mV)	50	350	ps
4	tr(PCIECLKP_250 mv)	Transition time _ PCIECLKP rise time (250 mV)	50	350	ps
4	tf(PCIECLKP_250 mv)	Transition time _ PCIECLKP fall time (250 mV)	50	350	ps
5	tj(PCIECLKN)	Jitter, peak_to_peak _ periodic PCIECLKN		4 (3)	ps,RMS
5	tj(PCIECLKP)	Jitter, peak_to_peak _ periodic PCIECLKP		4 (3)	ps,RMS
End of Table 8-26					

1 See the Hardware Design Guide for KeyStone Devices in "Related Documentation from Texas Instruments" on page 75 for detailed recommendations.
2 If AIF2 is being used then SYSCLK(N|P) can be programmed only to fixed values, if AIF2 is not being used then the SYSCLK(N|P) range is the same as the ALTCORECLK(N|P) range.
3 The jitter frequency mask shown in the Hardware Design Guide for KeyStone Devices in "Related Documentation from Texas Instruments" on page 75 must also be met for the specific operating mode chosen.

Figure 8-19 Main PLL Controller/SRIO/HyperLink/PCIe Clock Input Timing

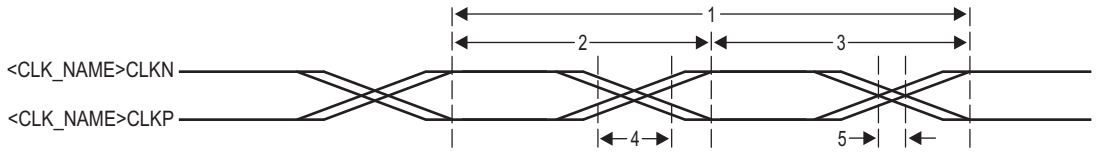
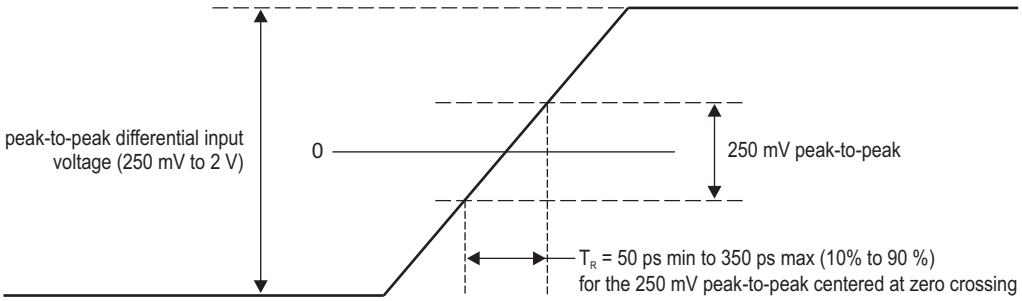


Figure 8-20 Main PLL Transition Time

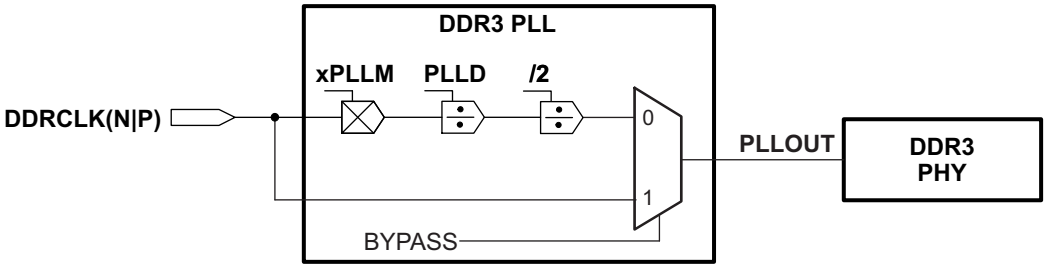


8.7 DDR3 PLL

The DDR3 PLL generates interface clocks for the DDR3 memory controller. When coming out of power-on reset, DDR3 PLL is programmed to a valid frequency during the boot config before being enabled and used.

DDR3 PLL power is supplied externally via the Main PLL power-supply pin (AVDDA2). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone Devices* in 2.13 "Related Documentation from Texas Instruments" on page 75 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).

Figure 8-21 DDR3 PLL Block Diagram



8.7.1 DDR3 PLL Control Registers

The DDR3 PLL, which is used to drive the DDR PHY for the EMIF, does not use a PLL Controller. DDR3 PLL can be controlled using the DDR3PLLCTL0 and DDR3PLLCTL1 registers located in the Bootcfg module. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using KICK0/KICK1 registers. For suggested configurable values see 2.7 "PLL Settings" on page 40. See 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 82 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers are reset on $\overline{\text{POR}}$ only.

Figure 8-22 DDR3 PLL Control Register (DDR3PLLCTL0) [1]

31 — 24	23	22 — 19	18 — 6	5 — 0
BWADJ[7:0]	BYPASS	Reserved	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

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

1 This register is Reset on POR only. The regreset, reset and bgreset from PLL are all tied to a common pll0_ctrl_rst_n The pwrdn, regpwrdn, bgpwrdn are all tied to common pll0_ctrl_to_pll_pwrdn.

Table 8-27 DDR3 PLL Control Register 0 Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. The combination (BWADJ[11:0]) should be programmed to a value equal to half of PLLM[12:0] if PLLM has even values or to be rounded half down of PLLM[12:0] if PLLM has odd values. Example: PLLM=15, then BWADJ=7
23	BYPASS	Enable bypass mode 0 = Bypass disabled 1 = Bypass enabled
22-19	Reserved	Reserved
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
End of Table 8-27		

Figure 8-23 DDR3 PLL Control Register 1 (DDR3PLLCTL1)

31 14	13	12 7	6	5 4	3 0
Reserved	PLLRST	Reserved	ENSAT	Reserved	BWADJ[11:8]
R-0000 0000 0000 0000 00	RW-0	R-0	RW-0	R-0	RW-0000

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

Table 8-28 DDR3 PLL Control Register 1 Field Descriptions

Bit	Field	Description
31-14	Reserved	Reserved
13	PLLRST	PLL reset bit 0 = PLL reset is released 1 = PLL reset is asserted
12-7	Reserved	Reserved
6	ENSAT	Enables saturation behavior Needs to be set to 1 for proper operation of PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. The combination (BWADJ[11:0]) should be programmed to a value equal to half of PLLM[12:0] if PLLM has even values or to be rounded half down of PLLM[12:0] if PLLM has odd values. Example: PLLM=15, then BWADJ=7
End of Table 8-28		

8.7.2 DDR3 PLL Device-Specific Information

As shown in Figure 8-21, the output of DDR3 PLL (PLLOUT) is divided by 2 and directly fed to the DDR3 memory controller. The DDR3 PLL is affected by power-on reset. During power-on resets, the internal clocks of the DDR3 PLL are affected as described in Section 8.5 "Reset Controller" on page 140. DDR3 PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{\text{RESETSTAT}}$ pin goes high. It does not lose lock during any of the other resets.

8.7.3 DDR3 PLL Initialization Sequence

See the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 75 for details on the initialization sequence for DDR3 PLL.

Note—The DDR3 interface needs to reset every time the DDR3 PLL is re-programmed.

8.7.4 DDR3 PLL Input Clock Electrical Data/Timing

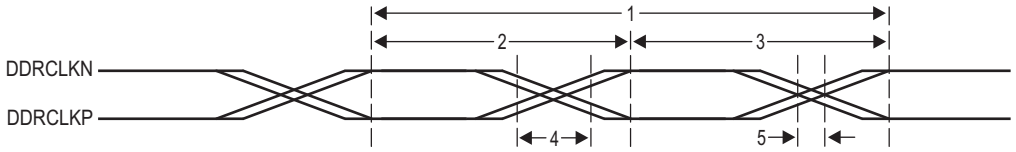
Table 8-29 DDR3 PLL DDRREFCLK(N|P) Timing Requirements (Part 1 of 2)
(see Figure 8-24 and Figure 8-20)

No.			Min	Max	Unit
		DDRCLK[P:N]			
1	tc(DDRCLKN)	Cycle time _ DDRCLKN cycle time	3.2	25	ns
1	tc(DDRCLKP)	Cycle time _ DDRCLKP cycle time	3.2	25	ns
3	tw(DDRCLKN)	Pulse width _ DDRCLKN high	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKN)	Pulse width _ DDRCLKN low	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKP)	Pulse width _ DDRCLKP high	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
3	tw(DDRCLKP)	Pulse width _ DDRCLKP low	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
4	tr(DDRCLKN_250 mv)	Transition time _ DDRCLKN rise time (250 mV)	50	350	ps

Table 8-29 DDR3 PLL DDRREFCLK(N|P) Timing Requirements (Part 2 of 2)
(see Figure 8-24 and Figure 8-20)

No.			Min	Max	Unit
4	tf(DDRCLKN_250 mv)	Transition time _ DDRCLKN fall time (250 mV)	50	350	ps
4	tr(DDRCLKP_250 mv)	Transition time _ DDRCLKP rise time (250 mV)	50	350	ps
4	tf(DDRCLKP_250 mv)	Transition time _ DDRCLKP fall time (250 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(DDRCLKN)	ps
End of Table 8-29					

Figure 8-24 DDR3 PLL DDRCLK Timing

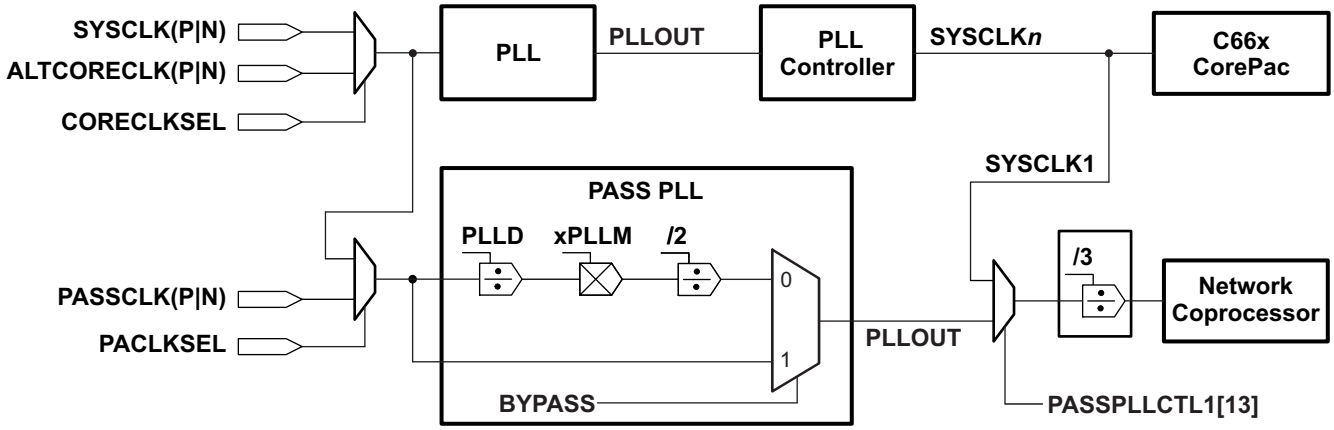


8.8 PASS PLL

The PASS PLL generates interface clocks for the Network Coprocessor. Using the PACLKSEL pin the user can select the input source of PASS PLL as either the output of Main PLL mux or the PASSCLK clock reference sources. When coming out of power-on reset, PASS PLL comes out in a bypass mode and needs to be programmed to a valid frequency before being enabled and used.

PASS PLL power is supplied externally via the PASS PLL power-supply pin (AVDDA3). An external EMI filter circuit must be added to all PLL supplies. Please see the Hardware Design Guide for KeyStone Devices in 2.13 "Related Documentation from Texas Instruments" on page 75 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).

Figure 8-25 PASS PLL Block Diagram



8.8.1 PASS PLL Control Register

The PASS PLL, which is used to drive the Network Coprocessor, does not use a PLL controller. PASS PLL can be controlled using the PAPLLCTL0 and PAPLLCTL1 registers located in Bootcfg module. These MMRs (memory-mapped registers) exists inside the Bootcfg space. To write to these registers, software should go through an un-locking sequence using KICK0/KICK1 registers. For suggested configurable values see 2.7 "PLL Settings" on page 40. See 3.3.4 "Kicker Mechanism (KICK0 and KICK1) Register" on page 82 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers are reset on $\overline{\text{POR}}$ only.

Figure 8-26 PASS PLL Control Register (PASSPLLCTL0) [1]

31 — 24	23	22 — 19	18 — 6	5 — 0
BWADJ[7:0]	BYPASS	Reserved	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

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

1 This register is Reset on POR only. The regreset, reset, and bgreset from PLL are all tied to a common pll0_ctrl_rst_n. The pwrdn, regpwrdn, and bgpwrdn are all tied to common pll0_ctrl_to_pll_pwrdn.

Table 8-30 PASS PLL Control Register Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ should be programmed to a value equal to half of PLLM[12:0]. Example: PLLM = 15, then BWADJ = 7
23	BYPASS	Enable bypass mode 0 = Bypass disabled 1 = Bypass enabled
22-19	Reserved	Reserved
18-6	PLLM	A 13-bit field that selects the values for the multiplication factor (see note below)
5-0	PLLD	A 6-bit field that selects the values for the reference divider
End of Table 8-30		

Figure 8-27 PASS PLL Control Register 1 (PASSPLLCTL1)

31 15	14	13	12 7	6	5 4	3 0
Reserved	PLLRST	PLLSELECT	Reserved	ENSAT	Reserved	BWADJ[11:8]
R-0000 0000 0000 0000 0	RW-0		R-0000 000	RW-0	R-0	RW-0000

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

Table 8-31 PASS 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	PLLSELECT	PASS PLL select bit 0 - Reserved 1 - PASS PLL output clock is used as the input to PASS
12-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper operation of the PLL
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in PASSPLLCTL0 and PASSPLLCTL1 registers. The combination (BWADJ[11:0]) should be programmed to a value equal to half of PLLM[12:0] if PLLM has even values or to be rounded half down of PLLM[12:0] if PLLM has odd values. Example: PLLM=15, then BWADJ=7
End of Table 8-31		

8.8.2 PASS PLL Device-Specific Information

As shown in Figure 8-25, the output of PASS PLL (PLLOUT) is divided by 3 and directly fed to the Network Coprocessor. The PASS PLL is affected by power-on reset. During power-on resets, the internal clocks of the PASS PLL are affected as described in Section 8.5 "Reset Controller" on page 140. PASS PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{\text{RESETSTAT}}$ pin goes high. It does not lose lock during any of the other resets.

8.8.3 PASS PLL Initialization Sequence

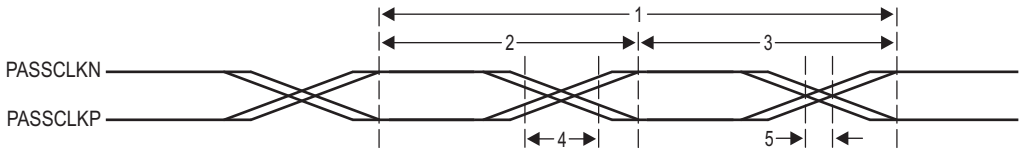
See the *Phase Locked Loop (PLL) for KeyStone Devices User Guide* in "Related Documentation from Texas Instruments" on page 75 for details on the initialization sequence for PASS PLL.

8.8.4 PASS PLL Input Clock Electrical Data/Timing

Table 8-32 PASS PLL Timing Requirements
(See Figure 8-28 and Figure 8-20)

No.	Parameter		Min	Max	Unit
	PASSCLK[P:N]				
1	tc(PASSCLKN)	Cycle time _ PASSCLKN cycle time	3.2	25	ns
1	tc(PASSCLKP)	Cycle time _ PASSCLKP cycle time	3.2	25	ns
3	tw(PASSCLKN)	Pulse width _ PASSCLKN high	0.45*tc(PASSCLKN)	0.55*tc(PASSCLKN)	ns
2	tw(PASSCLKN)	Pulse width _ PASSCLKN low	0.45*tc(PASSCLKN)	0.55*tc(PASSCLKN)	ns
2	tw(PASSCLKP)	Pulse width _ PASSCLKP high	0.45*tc(PASSCLKP)	0.55*tc(PASSCLKP)	ns
3	tw(PASSCLKP)	Pulse width _ PASSCLKP low	0.45*tc(PASSCLKP)	0.55*tc(PASSCLKP)	ns
4	tr(PASSCLKN_250 mv)	Transition time _ PASSCLKN rise time (250 mV)	50	350	ps
4	tf(PASSCLKN_250 mv)	Transition time _ PASSCLKN fall time (250 mV)	50	350	ps
4	tr(PASSCLKP_250 mv)	Transition time _ PASSCLKP Rise time (250 mV)	50	350	ps
4	tf(PASSCLKP_250 mv)	Transition time _ PASSCLKP Fall time (250 mV)	50	350	ps
5	tj(PASSCLKN)	Jitter, peak_to_peak _ periodic PASSCLKN		0.02*tc(PASSCLKN)	ps, pk-pk
5	tj(PASSCLKP)	Jitter, peak_to_peak _ periodic PASSCLKP		0.02*tc(PASSCLKP)	ps, pk-pk

Figure 8-28 PASS PLL Timing



8.9 Enhanced Direct Memory Access (EDMA3) Controller

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

There are 3 EDMA Channel Controllers on the device, EDMA3CC0, EDMA3CC1, and EDMA3CC2. EDMA3CC0 is optimized to be used for transfers to/from/within the MSMC and DDR-3 subsytems. The others are to be 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
- 128 PaRAM entries for EDMA3CC0, 512 each for EDMA3CC1 and EDMA3CC2
  - Used to define transfer context for channels
  - Each PaRAM entry can be used as a DMA entry, QDMA entry, or link entry
- 16 DMA channels for EDMA3CC0, 64 each for EDMA3CC1 and EDMA3CC2
  - Manually triggered (CPU writes to channel controller register), external event triggered, and 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, 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

In the context of this document, EDMA3TCs associated with EDMA3CC0 are referred to as EDMA3CC0 EDMA3TC0 and 1. EDMA3TCs associated with EDMA3CC1 and 2 are each referred to as EDMA3CCx EDMA3TC0 - 3, where x is 1 or 2. Each of the transfer controllers has a direct connection to the TeraNet. Table 4-3 "Packet DMA Priority Allocation Register Field Descriptions" lists the peripherals that can be accessed by the transfer controllers.

8.9.1 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. On the TCI6612, the EDMA can use constant addressing mode only with the enhanced Viterbi decoder coprocessor (VCP) and the enhanced turbo decoder coprocessor (TCP). Constant addressing mode is not supported by any other peripheral or internal memory in the SoC. Note that increment mode is supported by all peripherals, including VCP and TCP. For more information on these two addressing modes, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

For the range of memory addresses that include EDMA3 Channel Controller (EDMA3CC) Control Registers and EDMA3 Transfer Controller (EDMA3TC) Control Register see Section 2.2 "Memory Map Summary" on page 21. For memory offsets and other details on EDMA3CC and EDMA3TC Control Registers entries, see the *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.9.2 EDMA3 Channel Controller Configuration

Table 8-33 provides the configuration for each of the EDMA3 channel controllers present on the device.

Table 8-33 EDMA3 Channel Controller Configuration

Description	EDMA3 CC0	EDMA3 CC1	EDMA3 CC2
Number of DMA channels in channel controller	16	64	64
Number of QDMA channels	8	8	8
Number of interrupt channels	16	64	64
Number of PaRAM set entries	128	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
End of Table 8-33			

8.9.3 EDMA3 Transfer Controller Configuration

Each transfer controller on a 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 set. The number of destination FIFO register set 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 8-34 provides the configuration for each of the EDMA3 transfer controllers present on the device.

Table 8-34 EDMA3 Transfer Controller Configuration

Parameter	EDMA3 CC0		EDMA3 CC1				EDMA3 CC2			
	TC0	TC1	TC0	TC1	TC2	TC3	TC0	TC1	TC2	TC3
FIFOSIZE	1024 bytes	1024 bytes	1024 bytes	512 bytes	1024 bytes	512 bytes	1024 bytes	512 bytes	512 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	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes	64 bytes
End of Table 8-34										

8.9.4 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 16 DMA channels for EDMA3CC0, 64 each for EDMA3CC1 and EDMA3CC2 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 lists the source of the synchronization event associated with each of the EDMA EDMA3CC DMA channels. On the TCI6612, 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 *Enhanced Direct Memory Access 3 (EDMA3) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

Table 8-35 EDMA3CC0 Events for TCI6612

Event Number	Event	Event Description
0	TINT9L	Timer interrupt low
1	TINT9H	Timer interrupt high
2	TINT10L	Timer interrupt low
3	TINT10H	Timer interrupt high
4	TINT11L	Timer interrupt low
5	TINT11H	Timer interrupt high
6	AIF_SEVT0	AIF radio timing sync event 0
7	AIF_SEVT1	AIF radio timing sync event 1
8	CIC2_OUT0	CIC2_OUT0
9	CIC2_OUT1	CIC2_OUT1
10	CIC2_OUT2	CIC2_OUT2
11	CIC2_OUT3	CIC2_OUT3
12	CIC2_OUT4	CIC2_OUT4
13	CIC2_OUT5	CIC2_OUT5
14	GPIO4	GPIO interrupt
15	GPIO5	GPIO interrupt
End of Table 8-35		

Table 8-36 EDMA3CC1 Events for TCI6612 (Part 1 of 3)

Event Number	Event	Event Description
0	SPIINT0	SPI interrupt
1	SPIINT1	SPI interrupt
2	SPIXEVT	Transmit event
3	SPIREVT	Receive event

Table 8-36 EDMA3CC1 Events for TCI6612 (Part 2 of 3)

Event Number	Event	Event Description
4	I2CREVT	I^2C receive event
5	I2CXEVT	I^2C transmit event
6	GPINT0	GPIO interrupt
7	GPINT1	GPIO interrupt
8	GPINT2	GPIO interrupt
9	GPINT3	GPIO interrupt
10	INTDST0	INTD interrupt 0
11	INTDST1	INTD interrupt 1
12	AIF_SEVT2	AIF radio timing sync event 2
13	AIF_SEVT3	AIF radio timing sync event 3
14	AIF_SEVT4	AIF radio timing sync event 4
15	AIF_SEVT5	AIF radio timing sync event 5
16	AIF_SEVT6	AIF radio timing sync event 6
17	AIF_SEVT7	AIF radio timing sync event 7
18	SEMINT0	Semaphore interrupt
19	SEMINT1	Semaphore interrupt
20	SEMINT2	Semaphore interrupt
21	SEMINT3	Semaphore interrupt
22	TINT4L	Timer interrupt low
23	TINT4H	Timer interrupt high
24	TINT5L	Timer interrupt low
25	TINT5H	Timer interrupt high
26	TINT6L	Timer interrupt low
27	TINT6H	Timer interrupt high
28	TINT7L	Timer interrupt low
29	TINT7H	Timer interrupt high
30	RAC_INT0	RAC_ interrupt 0
31	RAC_INT1	RAC_ interrupt 1
32	RAC_INT2	RAC_interrupt 2
33	RAC_INT3	RAC_interrupt 3
34	RAC_DEVENT0	RAC_debug Event
35	RAC_DEVENT1	RAC_debug Event
36	TAC_INTD	TAC error interrupt
37	TACDEVENT0	TAC debug event
38	TACDEVENT1	TAC debug event
39 - 44	Reserved	
45	CIC1_OUT2	Interrupt Controller output
46	CIC1_OUT3	Interrupt Controller output
47	CIC1_OUT4	Interrupt Controller output
48	CIC1_OUT5	Interrupt Controller output
49	CIC1_OUT6	Interrupt Controller output
50	CIC1_OUT7	Interrupt Controller output
51	CIC1_OUT8	Interrupt Controller output
52	CIC1_OUT9	Interrupt Controller output

Table 8-36 EDMA3CC1 Events for TCI6612 (Part 3 of 3)

Event Number	Event	Event Description
53	CIC1_OUT10	Interrupt Controller output
54	CIC1_OUT11	Interrupt Controller output
55	CIC1_OUT12	Interrupt Controller output
56	CIC1_OUT13	Interrupt Controller output
57	CIC1_OUT14	Interrupt Controller output
58	CIC1_OUT15	Interrupt Controller output
59	CIC1_OUT16	Interrupt Controller output
60	CIC1_OUT17	Interrupt Controller output
61	CIC1_OUT18	Interrupt Controller output
62	CIC1_OUT19	Interrupt Controller output
63	CIC1_OUT20	Interrupt Controller output
End of Table 8-36		

Table 8-37 EDMA3CC2 Events for TCI6612 (Part 1 of 2)

Event Number	Event	Event Description
0	TCP3D_A_REVT0	TCP3D_A receive event 0
1	TCP3D_A_REVT1	TCP3D_A receive event 1
2	URXEVT1	UART receive event 1
3	UTXEVT1	UART transmit event 1
4	URXEVT0	UART receive event 0
5	UTXEVT0	UART transmit event 0
6	GPINT0	GPIO interrupt
7	GPINT1	GPIO interrupt
8	GPINT2	GPIO interrupt
9	GPINT3	GPIO interrupt
10	VCP_A_REVT	Receive event
11	VCP_A_XEVT	Transmit event
12	VCP_B_REVT	Receive event
13	VCP_B_XEVT	Transmit event
14-17	Reserved	
18	SEMINT0	Semaphore interrupt
19	SEMINT1	Semaphore interrupt
20	SEMINT2	Semaphore interrupt
21	SEMINT3	Semaphore interrupt
22	TINT9L	Timer interrupt low
23	TINT9H	Timer interrupt high
24	TINT10L	Timer interrupt low
25	TINT10H	Timer interrupt high
26	TINT11L	Timer interrupt low
27	TINT11H	Timer interrupt high
28	TINT7L	Timer interrupt low
29	TINT7H	Timer interrupt high
30	SPIXEVT	SPI transmit event
31	SPIREVT	SPI receive event

Table 8-37 EDMA3CC2 Events for TCI6612 (Part 2 of 2)

Event Number	Event	Event Description
32	I2CREVT	I^2C receive event
33	I2CXEVT	I^2C transmit event
34	TCP3D_B_REVT0	TCP3D_B receive event0
35	TCP3D_B_REVT1	TCP3D_B receive event1
36	CIC1_OUT23	Interrupt Controller output
37	CIC1_OUT24	Interrupt Controller output
38	CIC1_OUT25	Interrupt Controller output
39	CIC1_OUT26	Interrupt Controller output
40	CIC1_OUT27	Interrupt Controller output
41	CIC1_OUT28	Interrupt Controller output
42	CIC1_OUT29	Interrupt Controller output
43	CIC1_OUT30	Interrupt Controller output
44	CIC1_OUT31	Interrupt Controller output
45	CIC1_OUT32	Interrupt Controller output
46	CIC1_OUT33	Interrupt Controller output
47	CIC1_OUT34	Interrupt Controller output
48	CIC1_OUT35	Interrupt Controller output
49	CIC1_OUT36	Interrupt Controller output
50	CIC1_OUT37	Interrupt Controller output
51	CIC1_OUT38	Interrupt Controller output
52	CIC1_OUT39	Interrupt Controller output
53	CIC1_OUT40	Interrupt Controller output
54	CIC1_OUT41	Interrupt Controller output
55	CIC1_OUT42	Interrupt Controller output
56	CIC1_OUT43	Interrupt Controller output
57	CIC1_OUT44	Interrupt Controller output
58	CIC1_OUT45	Interrupt Controller output
59	CIC1_OUT46	Interrupt Controller output
60	CIC1_OUT47	Interrupt Controller output
61	SEMINT7	Semaphore Interrupt 7
62	POSDMARREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
63	POSDMAWREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
End of Table 8-37		

8.10 Interrupts

8.10.1 Interrupt Sources and Interrupt Controller

The CPU interrupts on the TCI6612 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 twelve 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 CIC blocks, CIC[3:0], with one controller per C66x CorePac. This is clocked using CPU/6.

The event controllers consist of simple combination logic to provide additional events to each C66x CorePac, the ARM, and the EDMA3CC. CIC0 provides 26 additional events to each of the C66x CorePacs (18 core specific and 8 broadcast), CIC1 provides 19 and 25 additional events to EDMA3CC1 and EDMA3CC2 respectively, and CIC2 provides 6 and 32 additional events to EDMA3CC0 and HyperLink respectively. CIC3 provides 33 additional events to the ARM. Because the ARM does not have NMI input, the NMI event from the watch dog timer is connected to ARM's input event 127 through an INTD for pulse to level conversion.

- A primary event indicates that the event needs to be connected to either CorePac, ARM (through INTD) or EDMA3CC directly.
- A secondary event indicates that the event can be selected or combined with other events before it is routed to EDMA3CC/CorePacs/ARM through CICs.
- A broadcast event indicates that the event is connected to all C66x CorePacs directly.
- Some events from a few modules are level-based interrupts and need to be routed to CorePacs that require pulse based interrupts. They must be aggregated and converted to one pulse interrupt by the INTD before reaching the CIC. A *through INTD* comment is added after these interrupts. These CP_INTDs are not shown in the diagram.
- Nearly all events from modules are pulse-based interrupts and need to be routed to ARM as primary inputs. Because the ARM requires level-based interrupts, they must be converted to level interrupts by INTD before reaching the ARM. A chip-level INTD is used for the purpose and is shown in the diagram.

The events that are routed to the C66x CorePacs for AET purposes, from those EDMA3CC and FSYNC events that are not otherwise provided to each C66x CorePac. For more details on the CIC features, see the *Interrupt Controller (CIC) for KeyStone Devices User Guid*e in 2.13 "Related Documentation from Texas Instruments" on page 75.

Note—Modules such as FFTC, TCP3d, TAC, AIF, MPU, BOOT_CFG, and the tracers have level interrupts and EOI handshaking interface. The EOI value is 0 for TCP3d, TAC, AIF, MPU, BOOT_CFG, and the tracers. For FFTC, the EOI values are 0 for FFTC_x_INTD0, 1 for FFTC_x_INTD1, 2 for FFTC_x_INTD2, and 3 for FFTC_x_INTD3 (where FFTC_x can be either FFTC_A or FFTC_B).

Figure 8-29 shows the TCI6612 interrupt topology.

Figure 8-29 Interrupt Topology

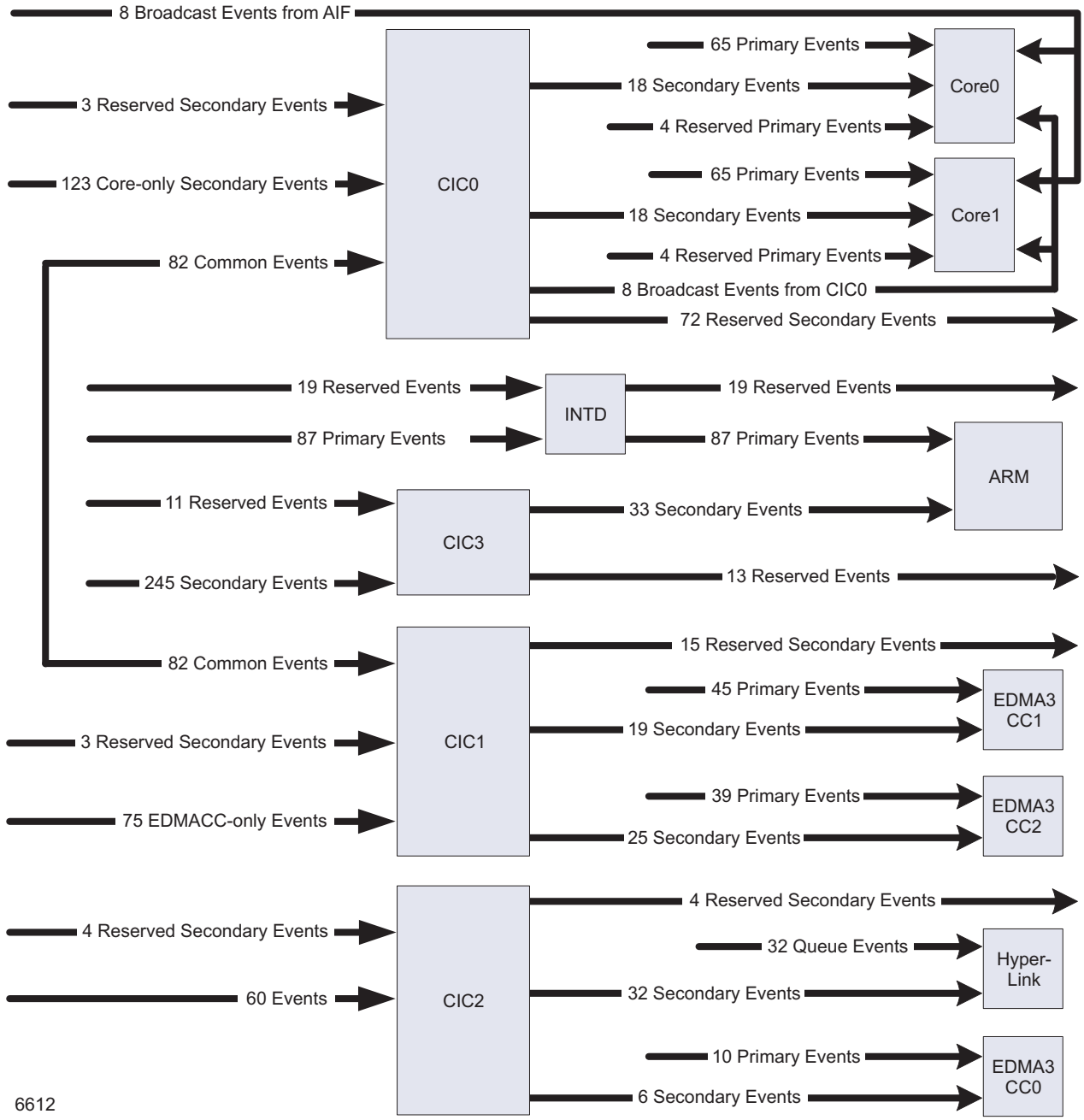


Table 8-38 shows the mapping of system events. For more information on the Interrupt Controller, see the *C66x CorePac User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

Table 8-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 1 of 4)

Event Number	Interrupt Event	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	TETBHFULLINTn (1)	TETB is half full
5	TETBFULLINTn (1)	TETB is full
6	TETBACQINTn (1)	Acquisition has been completed
7	TETBOVFLINTn (1)	Overflow condition interrupt
8	TETBUNFLINTn (1)	Underflow condition interrupt
9	EMU_DTDMA	Emulation interrupt for: 1. Host scan access 2. DTDMA transfer complete 3. AET interrupt
10	MSMC_mpf_errorn (2)	Memory protection fault indicators for local CorePac
11	Reserved	
12	Reserved	
13	IDMA0	IDMA channel 0 interrupt
14	IDMA1	IDMA channel 1 interrupt
15	SEMERRn (3)	Semaphore error interrupt
16	SEMINTn (3)	Semaphore interrupt
17	PCIEXpress_MSI_INTn (4)	Message signaled interrupt mode
18	PCIEXpress_MSI_INTn+1 (4)	Message signaled interrupt mode
19	RAC_INTn (5)	RAC interrupt
20	INTDST(n+16) (6)	SRIO interrupt
21	INTDST(n+20) (6)	SRIO interrupt
22	CIC0_OUT(64+0+10*n) (7)	Interrupt Controller output
23	CIC0_OUT(64+1+10*n) (7)	Interrupt Controller output
24	CIC0_OUT(64+2+10*n) (7)	Interrupt Controller output
25	CIC0_OUT(64+3+10*n) (7)	Interrupt Controller output
26	CIC0_OUT(64+4+10*n) (7)	Interrupt Controller output
27	CIC0_OUT(64+5+10*n) (7)	Interrupt Controller output
28	CIC0_OUT(64+6+10*n) (7)	Interrupt Controller output
29	CIC0_OUT(64+7+10*n) (7)	Interrupt Controller output
30	CIC0_OUT(64+8+10*n) (7)	Interrupt Controller output
31	CIC0_OUT(64+9+10*n) (7)	Interrupt Controller output
32	QM_INT_LOW_0	QM interrupt for 0~31 queues
33	QM_INT_LOW_1	QM interrupt for 32~63 queues
34	QM_INT_LOW_2	QM interrupt for 64~95 queues
35	QM_INT_LOW_3	QM interrupt for 96~127 queues
36	QM_INT_LOW_4	QM interrupt for 128~159 queues
37	QM_INT_LOW_5	QM interrupt for 160~191 queues
38	QM_INT_LOW_6	QM interrupt for 192~223 queues

Table 8-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 2 of 4)

Event Number	Interrupt Event	Description
39	QM_INT_LOW_7	QM interrupt for 224~255 queues
40	QM_INT_LOW_8	QM interrupt for 256~287 queues
41	QM_INT_LOW_9	QM interrupt for 288~319 queues
42	QM_INT_LOW_10	QM interrupt for 320~351 queues
43	QM_INT_LOW_11	QM interrupt for 352~383 queues
44	QM_INT_LOW_12	QM interrupt for 384~415 queues
45	QM_INT_LOW_13	QM interrupt for 416~447 queues
46	QM_INT_LOW_14	QM interrupt for 448~479 queues
47	QM_INT_LOW_15	QM interrupt for 480~511 queues
48	QM_INT_HIGH_n (8)	QM interrupt for queue 704+n
49	QM_INT_HIGH_(n+4) (8)	QM interrupt for queue 708+n
50	QM_INT_HIGH_(n+8) (8)	QM interrupt for queue 712+n
51	QM_INT_HIGH_(n+12) (8)	QM interrupt for queue 716+n
52	QM_INT_HIGH_(n+16) (8)	QM interrupt for queue 720+n
53	QM_INT_HIGH_(n+20) (8)	QM interrupt for queue 724+n
54	QM_INT_HIGH_(n+24) (8)	QM interrupt for queue 728+n
55	QM_INT_HIGH_(n+28) (8)	QM interrupt for queue 732+n
56	CIC0_OUT0	Interrupt Controller output
57	CIC0_OUT1	Interrupt Controller output
58	CIC0_OUT2	Interrupt Controller output
59	CIC0_OUT3	Interrupt Controller output
60	CIC0_OUT4	Interrupt Controller output
61	CIC0_OUT5	Interrupt Controller output
62	CIC0_OUT6	Interrupt Controller output
63	CIC0_OUT7	Interrupt Controller output
64	TINTLn (9)	Local timer interrupt low
65	BCP_ERRORn (AT)	BCP error
66	TINT4L	Timer 4 interrupt low
67	TINT4H	Timer 4 interrupt high
68	TINT5L	Timer 5 interrupt low
69	TINT5H	Timer 5 interrupt high
70	TINT6L	Timer 6 interrupt low
71	TINT6H	Timer 6 interrupt high
72	TINT7L	Timer 7 interrupt low
73	TINT7H	Timer 7 interrupt high
74	CIC0_OUT(8+16*n) (7)	Interrupt Controller output
75	CIC0_OUT(9+16*n) (7)	Interrupt Controller output
76	CIC0_OUT(10+16*n) (7)	Interrupt Controller output
77	CIC0_OUT(11+16*n) (7)	Interrupt Controller output
78	GPINT4	Local GPIO interrupt
79	GPINT5	Local GPIO interrupt
80	GPINT6	Local GPIO interrupt
81	GPINT7	Local GPIO interrupt
82	GPINT8	Local GPIO interrupt

Table 8-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 3 of 4)

Event Number	Interrupt Event	Description
83	GPINT9	Local GPIO interrupt
84	GPINT10	Local GPIO interrupt
85	GPINT11	Local GPIO interrupt
86	GPINT12	Local GPIO interrupt
87	GPINT13	Local GPIO interrupt
88	GPINT14	Local GPIO interrupt
89	GPINT15	Local GPIO interrupt
90	IPC_LOCAL	Inter SoC interrupt from IPCGRn
91	GPINTn (10)	Local GPIO interrupt
92	CIC0_OUT(12+16*n) (7)	Interrupt Controller output
93	CIC0_OUT(13+16*n) (7)	Interrupt Controller output
94	CIC0_OUT(14+16*n) (7)	Interrupt Controller output
95	CIC0_OUT(15+16*n) (7)	Interrupt Controller output
96	INTERR	Dropped CPU interrupt event
97	EMC_IDMAERR	Invalid IDMA parameters
98	Reserved	
99	Reserved	
100	EFIINTA	EFI interrupt from side A
101	EFIINTB	EFI interrupt from side B
102	AIF_SEVT0	AIF radio timing sync event 0
103	AIF_SEVT1	AIF radio timing sync event 1
104	AIF_SEVT2	AIF radio timing sync event 2
105	AIF_SEVT3	AIF radio timing sync event 3
106	AIF_SEVT4	AIF radio timing sync event 4
107	AIF_SEVT5	AIF radio timing sync event 5
108	AIF_SEVT6	AIF radio timing sync event 6
109	AIF_SEVT7	AIF radio timing sync event 7
110	MDMAERREVT	VbusM error event
111	Reserved	
112	EDMA3CC0_EDMACC_AETEVT	EDMA3CC0 AET event
113	PMC_ED	Single bit error detected during DMA read
114	EDMA3CC1_EDMACC_AETEVT	EDMA3CC1 AET event
115	EDMA3CC2_EDMACC_AETEVT	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

Table 8-38 System Event Mapping — C66x CorePac Primary Interrupts (Part 4 of 4)

Event Number	Interrupt Event	Description
126	EMC_CMPA	CPU memory protection fault
127	EMC_BUSERR	Bus Error Interrupt
End of Table 8-38		

1 Core [n] will receive TETBHFULLINTn, TETBFULLINTn, TETBACQINTn, TETBOVFLINTn and TETBUNFLINTn.
2 Core [n] will receive MSMC_mpf_errorn.
3 Core [n] will receive SEMINTn and SEMERRn.
4 Core [n] will receive PCIEXpress_MSI_INTn and PCIEXpress_MSI_INTn+1.
5 Core [n] will receive RACINTn.
6 Core [n] will receive INTDST(n+16) and INTDST(n+20).
7 n is core number.
8 n is core number.
9 Core [n] will receive TINTLn and TINTHn.
10 Core [n] will receive GPINTn.

Table 8-39 Events for ARM CorePac (Part 1 of 4)

Event Number	Interrupt Event	Description
0	EMUINT (1)	Emulation interrupt
1	EVT1COMMTX (1)	COMMTX interrupt
2	COMMRX (1)	COMMRX interrupt
3	PMUIRQ (1)	IRQ
4	Reserved	
5	SSM_WFI_IRQ (1)	Secure FIQ_WFI for process scheduling
6	SSM_IRQ (1)	Secure IRQ
7	QM_INT_HIGH_1	QM
8	IPC_H	IPC register inside Boot_CFG
9	QM_INT_HIGH_0	QM
10	Interconnect errors for application program	ARM CorePac event
11	QM_INT_HIGH_2	QM
12	QM_INT_HIGH_3	QM
13	QM_INT_HIGH_4	QM
14	QM_INT_HIGH_5	QM
15	QM_INT_HIGH_6	QM
16	QM_INT_HIGH_7	QM
17	QM_INT_HIGH_8	QM
18	QM_INT_HIGH_9	QM
19	QM_INT_HIGH_10	QM
20	QM_INT_HIGH_11	QM
21	QM_INT_HIGH_12	QM
22	QM_INT_HIGH_13	QM
23	QM_INT_HIGH_14	QM
24	QM_INT_HIGH_15	QM
25	QM_INT_HIGH_16	QM
26	QM_INT_HIGH_17	QM
27	QM_INT_HIGH_18	QM
28	QM_INT_HIGH_19	QM
29	QM_INT_HIGH_20	QM
30	QM_INT_HIGH_21	QM

Table 8-39 Events for ARM CorePac (Part 2 of 4)

Event Number	Interrupt Event	Description
31	QM_INT_HIGH_22	QM
32	QM_INT_HIGH_23	QM
33	QM_INT_HIGH_24	QM
34	QM_INT_HIGH_25	QM
35	QM_INT_HIGH_26	QM
36	QM_INT_HIGH_27	QM
37	QM_INT_HIGH_28	QM
38	QM_INT_HIGH_29	QM
39	QM_INT_HIGH_30	QM
40	QM_INT_HIGH_31	QM
41	QM_INT_TXQ_PEND_650	QM
42	QM_INT_TXQ_PEND_651	QM
43	QM_INT_TXQ_PEND_652	QM
44	QM_INT_TXQ_PEND_653	QM
45	QM_INT_TXQ_PEND_654	QM
46	QM_INT_TXQ_PEND_655	QM
47	QM_INT_TXQ_PEND_656	QM
48	QM_INT_TXQ_PEND_657	QM
49	QM_INT_PASS_TXQ_PEND_670	QM
50	QM_INT_PASS_TXQ_PEND_671	QM
51	HyperLink_int_o	HyperLink
52	SEMERR7	Semaphore_Local
53	SEMINT7	Semaphore_Local
54	INTDST20	SRIO
55	INTDST21	SRIO
56	INTDST22	SRIO
57	INTDST23	SRIO
58	TINT4L	Timer64_4
59	TINT4H	Timer64_4
60	TINT5L	Timer64_5
61	TINT5H	Timer64_5
62	TINT6L	Timer64_6
63	TINT6H	Timer64_6
64	TINT7L	Timer64_7
65	TINT7H	Timer64_7
66	PCIEXpress_ERR_INT	PCIEXpress
67	PCIEXpress_PM_INT	PCIEXpress
68	PCIEXpress_Legacy_INTA	PCIEXpress
69	PCIEXpress_Legacy_INTB	PCIEXpress
70	PCIEXpress_Legacy_INTC	PCIEXpress
71	PCIEXpress_Legacy_INTD	PCIEXpress
72	PCIEXpress_MSI_INT4	PCIEXpress
73	PCIEXpress_MSI_INT5	PCIEXpress
74	PCIEXpress_MSI_INT6	PCIEXpress

Table 8-39 Events for ARM CorePac (Part 3 of 4)

Event Number	Interrupt Event	Description
75	PCIEXpress_MSI_INT7	PCIEXpress
76	TINT8L	Timer64_8
77	TINT8H	Timer64_8
78	TINT9L	Timer64_9
79	TINT9H	Timer64_9
80	TINT10L	Timer64_10
81	TINT10H	Timer64_10
82	TINT11L	Timer64_11
83	TINT11H	Timer64_11
84	TCP3D_A_REVT0	TCP3D_A
85	TCP3D_A_REVT1	TCP3D_A
86	TCP3D_B_REVT0	TCP3D_B
87	TCP3D_B_REVT1	TCP3D_B
88	CPU/3_1_EDMA3CCINT2	CPU/3 EDMA
89	CPU/3_1_EDMA3CCINT6	CPU/3 EDMA
90	CPU/3_2_EDMA3CCINT2	CPU/3 EDMA
91	CPU/3_2_EDMA3CCINT6	CPU/3 EDMA
92	CPU/2_EDMA3CCINT2	CPU/2 EDMA
93	CPU/2_EDMA3CCINT6	CPU/2 EDMA
94	CIC3_OUT0	CIC Controller3
95	CIC3_OUT1	CIC Controller3
96	CIC3_OUT2	CIC Controller3
97	CIC3_OUT3	CIC Controller3
98	CIC3_OUT4	CIC Controller3
99	CIC3_OUT5	CIC Controller3
100	CIC3_OUT6	CIC Controller3
101	CIC3_OUT7	CIC Controller3
102	CIC3_OUT8	CIC Controller3
103	CIC3_OUT9	CIC Controller3
104	CIC3_OUT10	CIC Controller3
105	CIC3_OUT11	CIC Controller3
106	CIC3_OUT12	CIC Controller3
107	CIC3_OUT13	CIC Controller3
108	CIC3_OUT14	CIC Controller3
109	CIC3_OUT15	CIC Controller3
110	CIC3_OUT16	CIC Controller3
111	CIC3_OUT17	CIC Controller3
112	CIC3_OUT18	CIC Controller3
113	CIC3_OUT19	CIC Controller3
114	CIC3_OUT20	CIC Controller3
115	CIC3_OUT21	CIC Controller3
116	CIC3_OUT22	CIC Controller3
117	CIC3_OUT23	CIC Controller3
118	CIC3_OUT24	CIC Controller3

Table 8-39 Events for ARM CorePac (Part 4 of 4)

Event Number	Interrupt Event	Description
119	CIC3_OUT25	CIC Controller3
120	CIC3_OUT26	CIC Controller3
121	CIC3_OUT27	CIC Controller3
122	CIC3_OUT28	CIC Controller3
123	CIC3_OUT29	CIC Controller3
124	CIC3_OUT30	CIC Controller3
125	CIC3_OUT31	CIC Controller3
126	CIC3_OUT32	CIC Controller3
127	Watch_dog_NMI	From RESETMUX for timer 8
End of Table 8-39		

1 Internal to A8 Core.

Table 8-40 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 1 of 5)

Input Event# on CIC	System Interrupt	Description
0	EDMA3CC1 EDMACC_ERRINT	EDMA3CC1 error interrupt
1	EDMA3CC1 EDMACC_MPINT	EDMA3CC1 memory protection interrupt
2	EDMA3CC1 EDMATC_ERRINT0	EDMA3CC1 EDMA3TC0 error interrupt
3	EDMA3CC1 EDMATC_ERRINT1	EDMA3CC1 EDMA3TC1 error interrupt
4	EDMA3CC1 EDMATC_ERRINT2	EDMA3CC1 EDMA3TC2 error interrupt
5	EDMA3CC1 EDMATC_ERRINT3	EDMA3CC1 EDMA3TC3 error interrupt
6	EDMA3CC1 EDMACC_GINT	EDMA3CC1 GINT
7	Reserved	
8	EDMA3CC1 EDMA3CCINT0	EDMA3CC1 individual completion interrupt
9	EDMA3CC1 EDMA3CCINT1	EDMA3CC1 individual completion interrupt
10	EDMA3CC1 EDMA3CCINT2	EDMA3CC1 individual completion interrupt
11	EDMA3CC1 EDMA3CCINT3	EDMA3CC1 individual completion interrupt
12	EDMA3CC1 EDMA3CCINT4	EDMA3CC1 individual completion interrupt
13	EDMA3CC1 EDMA3CCINT5	EDMA3CC1 individual completion interrupt
14	EDMA3CC1 EDMA3CCINT6	EDMA3CC1 individual completion interrupt
15	EDMA3CC1 EDMA3CCINT7	EDMA3CC1 individual completion interrupt
16	EDMA3CC2 EDMACC_ERRINT	EDMA3CC2 error interrupt
17	EDMA3CC2 EDMACC_MPINT	EDMA3CC2 memory protection interrupt
18	EDMA3CC2 EDMATC_ERRINT0	EDMA3CC2 EDMA3TC0 error interrupt
19	EDMA3CC2 EDMATC_ERRINT1	EDMA3CC2 EDMA3TC1 error interrupt
20	EDMA3CC2 EDMATC_ERRINT2	EDMA3CC2 EDMA3TC2 error interrupt
21	EDMA3CC2 EDMATC_ERRINT3	EDMA3CC2 EDMA3TC3 error interrupt
22	EDMA3CC2 EDMACC_GINT	EDMA3CC2 GINT
23	MPU_Combined_Address_Error	MPU0-7_ADDR_ERR_INT combined
24	EDMA3CC2 EDMA3CCINT0	EDMA3CC2 individual completion interrupt
25	EDMA3CC2 EDMA3CCINT1	EDMA3CC2 individual completion interrupt
26	EDMA3CC2 EDMA3CCINT2	EDMA3CC2 individual completion interrupt
27	EDMA3CC2 EDMA3CCINT3	EDMA3CC2 individual completion interrupt
28	EDMA3CC2 EDMA3CCINT4	EDMA3CC2 individual completion interrupt

Table 8-40 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 2 of 5)

Input Event# on CIC	System Interrupt	Description
29	EDMA3CC2 EDMA3CCINT5	EDMA3CC2 individual completion interrupt
30	EDMA3CC2 EDMA3CCINT6	EDMA3CC2 individual completion interrupt
31	EDMA3CC2 EDMA3CCINT7	EDMA3CC2 individual completion interrupt
32	EDMA3CC0 EDMACC_ERRINT	EDMA3CC0 error interrupt
33	EDMA3CC0 EDMACC_MPINT	EDMA3CC0 memory protection interrupt
34	EDMA3CC0 EDMATC_ERRINT0	EDMA3CC0 EDMA3TC0 error interrupt
35	EDMA3CC0 EDMATC_ERRINT1	EDMA3CC0 EDMA3TC1 error interrupt
36	EDMA3CC0 EDMACC_GINT	EDMA3CC0 GINT
37	MPU_Combined_PROT_Error	MPU0-7_PROT_ERR_INT combined
38	EDMA3CC0INT0	EDMA3CC0 individual completion interrupt
39	EDMA3CC0INT1	EDMA3CC0 individual completion interrupt
40	EDMA3CC0INT2	EDMA3CC0 individual completion interrupt
41	EDMA3CC0INT3	EDMA3CC0 individual completion interrupt
42	EDMA3CC0INT4	EDMA3CC0 individual completion interrupt
43	EDMA3CC0INT5	EDMA3CC0 individual completion interrupt
44	EDMA3CC0INT6	EDMA3CC0 individual completion Interrupt
45	EDMA3CC0INT7	EDMA3CC0 individual completion interrupt
46	Reserved	
47	Tracer_DDR_2_INTD	Tracer sliding time window interrupt for DDR3 EMIF
48	PCIEXpress_ERR_INT	Protocol error interrupt
49	PCIEXpress_PM_INT	Power management interrupt
50	PCIEXpress_Legacy_INTA	Legacy interrupt mode
51	PCIEXpress_Legacy_INTB	Legacy interrupt mode
52	PCIEXpress_Legacy_INTC	Legacy interrupt mode
53	PCIEXpress_Legacy_INTD	Legacy interrupt mode
54	SPIINT0	SPI interrupt0
55	SPIINT1	SPI interrupt1
56	SPIXEVT	SPI transmit event
57	SPIREVT	SPI receive event
58	I2CINT	I^2C interrupt
59	I2CREVT	I^2C receive event
60	I2CXEVT	I^2C transmit event
61	Reserved	
62	Reserved	
63	TETBHFULLINT	TETB is half full
64	TETBFULLINT	TETB is full
65	TETBACQINT	Acquisition has been completed
66	TETBOVFLINT	Overflow condition occurred
67	TETBUNFLINT	Underflow condition occurred
68	mdio_link_intr0	Packet Accelerator subsystem MDIO interrupt
69	mdio_link_intr1	Packet Accelerator subsystem MDIO interrupt
70	mdio_user_intr0	Packet Accelerator subsystem MDIO interrupt
71	mdio_user_intr1	Packet Accelerator subsystem MDIO interrupt
72	misc_intr	Packet Accelerator subsystem misc Interrupt

Table 8-40 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 3 of 5)

Input Event# on CIC	System Interrupt	Description
73	Tracer_core_0_INTD	Tracer sliding time window interrupt for individual core
74	Tracer_core_1_INTD	Tracer sliding time window interrupt for individual core
75	Reserved	
76	Reserved	
77	Tracer_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF
78	Tracer_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM bank0
79	Tracer_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM bank1
80	Tracer_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM bank2
81	Tracer_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM bank3
82	Tracer_CFG_INTD	Tracer sliding time window interrupt for CFG0 TeraNet
83	Tracer_QM_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG
84	Tracer_QM_DMA_INTD	Tracer sliding time window interrupt for QM_SS slave
85	Tracer_SM_INTD	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_INTD	Chip-level MMR Error Register
89	Reserved	
90	POSDMARREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
91	TINT0L	Timer0 interrupt low
92	POSDMAWREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
93	TINT1L	Timer1 interrupt low
94	Reserved	
95	Reserved	Reserved
96	Reserved	
97	Reserved	Reserved
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	KEYMGRINT1	Key MGR interrupt
102	MSMC_mpf_error8	Memory protection fault indicators for each system master PrivID
103	MSMC_mpf_error9	Memory protection fault indicators for each system master PrivID
104	MSMC_mpf_error10	Memory protection fault indicators for each system master PrivID
105	MSMC_mpf_error11	Memory protection fault indicators for each system master PrivID
106	MSMC_mpf_error12	Memory protection fault indicators for each system master PrivID
107	MSMC_mpf_error13	Memory protection fault indicators for each system master PrivID
108	MSMC_mpf_error14	Memory protection fault indicators for each system master PrivID
109	MSMC_mpf_error15	Memory protection fault indicators for each system master PrivID
110	DDR3_ERR	DDR3_EMIF Error Interrupt
111	HyperLink_int_o	HyperLink Interrupt
112	INTDST0	RapidIO Interrupt
113	INTDST1	RapidIO Interrupt
114	INTDST2	RapidIO Interrupt
115	INTDST3	RapidIO Interrupt
116	INTDST4	RapidIO Interrupt

Table 8-40 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 4 of 5)

Input Event# on CIC	System Interrupt	Description
117	INTDST5	RapidIO Interrupt
118	INTDST6	RapidIO Interrupt
119	INTDST7	RapidIO Interrupt
120	INTDST8	RapidIO Interrupt
121	INTDST9	RapidIO Interrupt
122	INTDST10	RapidIO Interrupt
123	INTDST11	RapidIO Interrupt
124	INTDST12	RapidIO Interrupt
125	INTDST13	RapidIO Interrupt
126	INTDST14	RapidIO interrupt
127	INTDST15	RapidIO interrupt
128	RACDEVENT0	RAC_debug event
129	RACDEVENT1	RAC_debug event
130	TAC_INTD	Error interrupt TACINT
131	TACDEVENT0	TAC debug event
132	TACDEVENT1	TAC debug event
133	AIF_INTD	AIF CPU error interrupt and AIF CPU alarm interrupt and starvation interrupt
134	QM_INT_PASS_TXQ_PEND_22	Queue Manager pend event
135	QM_INT_PASS_TXQ_PEND_23	Queue Manager pend event
136	QM_INT_PASS_TXQ_PEND_24	Queue Manager pend event
137	QM_INT_PASS_TXQ_PEND_25	Queue Manager pend event
138	QM_INT_PASS_TXQ_PEND_26	Queue Manager pend event
139	QM_INT_PASS_TXQ_PEND_27	Queue Manager pend event
140	QM_INT_PASS_TXQ_PEND_28	Queue Manager pend event
141	QM_INT_PASS_TXQ_PEND_29	Queue Manager pend event
142	QM_INT_PASS_TXQ_PEND_30	Queue Manager pend event
143	VCP_A_INT	Error interrupt
144	VCP_B_INT	Error interrupt
145	Reserved	
146	Reserved	
147	VCP_A_REVT	Receive event
148	VCP_A_XEVT	Transmit event
149	VCP_B_REVT	Receive event
150	VCP_B_XEVT	Transmit event
151	Reserved	
152	Reserved	
153	Reserved	
154	Reserved	
155	TCP3D_A_INTD	TCP3d_A error interrupt TCP3DINT0 and TCP3DINT1
156	TCP3D_B_INTD	TCP3d_B error interrupt TCP3DINT0 and TCP3DINT1
157	TCP3D_A_REVT0	TCP3d_A receive event0
158	TCP3D_A_REVT1	TCP3d_A receive event1
159	UARTINT1	UART 1 interrupt
160	URXEVT1	UART 1 receive event

Table 8-40 CIC0 Event Inputs — C66x CorePac Secondary Interrupts (Part 5 of 5)

Input Event# on CIC	System Interrupt	Description
161	UTXEVT1	UART 1 transmit event
162	TCP3D_B_REVT0	TCP3d_B receive event0
163	TCP3D_B_REVT1	TCP3d_B receive event1
164	UARTINT0	UART 0 interrupt
165	URXEVT0	UART 0 receive event
166	UTXEVT0	UART 0 transmit event
167	Tracer_RAC_CFG_INTD	Tracer sliding time window interrupt for RAC
168	Tracer_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
169	Tracer_TAC_INTD	Tracer sliding time window interrupt for TAC
170	MSMC_mpf_error4	Memory protection fault indicators for each system master PrivID
171	MSMC_mpf_error5	Memory protection fault indicators for each system master PrivID
172	MSMC_mpf_error6	Memory protection fault indicators for each system master PrivID
173	MSMC_mpf_error7	Memory protection fault indicators for each system master PrivID
174	Tracer_TNet_6P_A_INTD	Tracer interrupt for monitoring transactions sent to TNet_6P_A
175	QM_INT_PASS_TXQ_PEND_31	Queue Manager pend event
176	QM_INT_CDMA_0	QM interrupt for CDMA starvation
177	QM_INT_CDMA_1	QM interrupt for CDMA starvation
178	RapidIO_INT_CDMA_0	RapidIO interrupt for CDMA starvation
179	PASS_INT_CDMA_0	PASS interrupt for CDMA starvation
180	PONIRQ	USIM interrupt through INTD, level to rising edge sensitivity event
181	SmartReflex_intrreq0	SmartReflex sensor interrupt
182	SmartReflex_intrreq1	SmartReflex sensor interrupt
183	SmartReflex_intrreq2	SmartReflex sensor interrupt
184	SmartReflex_intrreq3	SmartReflex sensor interrupt
185	VPNoSMPSAck	VPVOLTUPDATE has been asserted but SMPS has not been responded to in a defined time interval
186	VPEqValue	$\overline{\text{SRSINTERUPT}}$ is asserted, but the new voltage is not different from the current SMPS voltage
187	VPMaxVdd	The new voltage required is equal to or greater than MaxVdd.
188	VPMinVdd	The new voltage required is equal to or less than MinVdd.
189	VPINIDLE	Indicating that the FSM of voltage processor is in idle.
190	VPOPPChangeDone	Indicating that the average frequency error is within the desired limit.
191	Reserved	
192	FFTC_A_INTD0	FFTC_A error event and FFTC_A debug event
193	FFTC_A_INTD1	FFTC_A error event and FFTC_A debug event
194	FFTC_A_INTD2	FFTC_A error event and FFTC_A debug event
195	FFTC_A_INTD3	FFTC_A error event and FFTC_A debug event
196	FFTC_B_INTD0	FFTC_B error event and FFTC_B debug event
197	FFTC_B_INTD1	FFTC_B error event and FFTC_B debug event
198	FFTC_B_INTD2	FFTC_B error event and FFTC_B debug event
199	FFTC_B_INTD3	FFTC_B error event and FFTC_B debug event
200	Reserved	
201	Reserved	
End of Table 8-40		

Table 8-41 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 1 of 4)

Input Event # on CIC	System Interrupt	Description
0	GPINT8	GPIO Interrupt
1	GPINT9	GPIO Interrupt
2	GPINT10	GPIO Interrupt
3	GPINT11	GPIO Interrupt
4	GPINT12	GPIO Interrupt
5	GPINT13	GPIO Interrupt
6	GPINT14	GPIO Interrupt
7	GPINT15	GPIO Interrupt
8	TETBHFULLINT	TETB is half full
9	TETBFULLINT	TETB is full
10	TETBACQINT	Acquisition has been completed
11	TETBHFULLINT0	TETB is half full
12	TETBFULLINT0	TETB is full
13	TETBACQINT0	Acquisition has been completed
14	TETBHFULLINT1	TETB is half full
15	TETBFULLINT1	TETB is full
16	TETBACQINT1	Acquisition has been completed
17	TETBHFULLINT2	TETB is half full
18	TETBFULLINT2	TETB is full
19	TETBACQINT2	Acquisition has been completed
20	TETBHFULLINT3	TETB is half full
21	TETBFULLINT3	TETB is full
22	TETBACQINT3	Acquisition has been completed
23	Reserved	
24	QM_INT_HIGH_16	QM Interrupt for IPC_core_0
25	QM_INT_HIGH_17	QM Interrupt for IPC_core_1
26	Reserved	
27	Reserved	
28	QM_INT_HIGH_20	QM Interrupt for IPC_core_0
29	QM_INT_HIGH_21	QM Interrupt for IPC_core_1
30	QM_INT_HIGH_22	Reserved
31	QM_INT_HIGH_23	Reserved
32	QM_INT_HIGH_24	QM Interrupt for IPC_core_0
33	QM_INT_HIGH_25	QM Interrupt for IPC_core_1
34	QM_INT_HIGH_26	Reserved
35	QM_INT_HIGH_27	Reserved
36	QM_INT_HIGH_28	QM Interrupt for IPC_core_0
37	QM_INT_HIGH_29	QM Interrupt for IPC_core_1
38	QM_INT_HIGH_30	Reserved
39	QM_INT_HIGH_31	Reserved
40	mdio_link_intr0	PASS_mdio Interrupt
41	mdio_link_intr1	PASS_mdio Interrupt
42	mdio_user_intr0	PASS_mdio Interrupt
43	mdio_user_intr1	PASS_mdio Interrupt

Table 8-41 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 2 of 4)

Input Event # on CIC	System Interrupt	Description
44	misc_intr	PASS_misc Interrupt
45	Tracer_core_0_INTD	Tracer sliding time window interrupt for individual core
46	Tracer_core_1_INTD	Tracer sliding time window interrupt for individual core
47	Reserved	
48	Reserved	
49	Tracer_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF
50	Tracer_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank0
51	Tracer_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank1
52	Tracer_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank2
53	Tracer_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank3
54	Tracer_CFG_INTD	Tracer sliding time window interrupt for CFG0 TeraNet
55	Tracer_QM_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG
56	Tracer_QM_DMA_INTD	Tracer sliding time window interrupt for QM_SS Slave port
57	Tracer_SM_INTD	Tracer sliding time window interrupt for Semaphore
58	SEMERR0	Semaphore interrupt
59	SEMERR1	Semaphore interrupt
60	Reserved	
61	Reserved	
62	BOOTCFG_INTD	Chip-level MMR Interrupt
63	PASS_INT_CDMA_0	PASS Interrupt for CDMA Starvation
64	MPU_Combined_Address_Error	MPU0~7_ADDR_ERR_INT combined
65	MSMC_scrub_cerror	Correctable (1-bit) soft error detected during scrub cycle
66	MPU_Combined_PROT_Error	MPU0~7_PROT_ERR_INT combined
67	RapidIO_INT_CDMA_0	RapidIO Interrupt for CDMA Starvation
68	SEMERR7	Semaphore interrupt
69	QM_INT_CDMA_0	QM Interrupt for CDMA Starvation
70	EASYNCERR	EMIF16 error interrupt
71	QM_INT_CDMA_1	QM Interrupt for CDMA Starvation
72	MSMC_dedc_cerror	Correctable (1-bit) soft error detected on SRAM read
73	MSMC_dedc_nc_error	Non-correctable (2-bit) soft error detected on SRAM read
74	MSMC_scrub_nc_error	Non-correctable (2-bit) soft error detected during scrub cycle
75	Reserved	
76	MSMC_mpf_error0	Memory protection fault indicators for each system master PrivID
77	MSMC_mpf_error1	Memory protection fault indicators for each system master PrivID
78	MSMC_mpf_error2	Memory protection fault indicators for each system master PrivID
79	MSMC_mpf_error3	Memory protection fault indicators for each system master PrivID
80	MSMC_mpf_error4	Memory protection fault indicators for each system master PrivID
81	MSMC_mpf_error5	Memory protection fault indicators for each system master PrivID
82	MSMC_mpf_error6	Memory protection fault indicators for each system master PrivID
83	MSMC_mpf_error7	Memory protection fault indicators for each system master PrivID
84	MSMC_mpf_error8	Memory protection fault indicators for each system master PrivID
85	MSMC_mpf_error9	Memory protection fault indicators for each system master PrivID
86	MSMC_mpf_error10	Memory protection fault indicators for each system master PrivID
87	MSMC_mpf_error11	Memory protection fault indicators for each system master PrivID

Table 8-41 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 3 of 4)

Input Event # on CIC	System Interrupt	Description
88	MSMC_mpf_error12	Memory protection fault indicators for each system master PrivID
89	MSMC_mpf_error13	Memory protection fault indicators for each system master PrivID
90	MSMC_mpf_error14	Memory protection fault indicators for each system master PrivID
91	MSMC_mpf_error15	Memory protection fault indicators for each system master PrivID
92	Tracer_TNet_6P_A_INTD	Tracer interrupt for monitoring transactions sent to TNet_6P_A
93	INTDST0	RapidIO Interrupt
94	INTDST1	RapidIO Interrupt
95	INTDST2	RapidIO Interrupt
96	INTDST3	RapidIO Interrupt
97	INTDST4	RapidIO Interrupt
98	INTDST5	RapidIO Interrupt
99	INTDST6	RapidIO Interrupt
100	INTDST7	RapidIO Interrupt
101	INTDST8	RapidIO Interrupt
102	INTDST9	RapidIO Interrupt
103	INTDST10	RapidIO Interrupt
104	INTDST11	RapidIO Interrupt
105	INTDST12	RapidIO Interrupt
106	INTDST13	RapidIO Interrupt
107	INTDST14	RapidIO Interrupt
108	INTDST15	RapidIO Interrupt
109	INTDST16	RapidIO Interrupt
110	INTDST17	RapidIO Interrupt
111	INTDST18	RapidIO Interrupt
112	INTDST19	RapidIO Interrupt
113	INTDST20	RapidIO Interrupt
114	INTDST21	RapidIO Interrupt
115	INTDST22	RapidIO Interrupt
116	INTDST23	RapidIO Interrupt
117	AIF_INTD	AIF CPU error interrupt and AIF CPU alarm interrupt and Starvation interrupt
118	Reserved	
119	VCP_A_INT	Error interrupt
120	VCP_B_INT	Error interrupt
121	Reserved	
122	Reserved	
123	TCP3D_A_INTD	Error interrupt TCP3DINT0 and TCP3DINT1
124	TCP3D_B_INTD	Error interrupt TCP3DINT0 and TCP3DINT1
125	Reserved	
126	FFTC_B_INTD0	FFTC_B error event and FFTC_B debug event
127	FFTC_B_INTD1	FFTC_B error event and FFTC_B debug event
128	GPINT4	GPIO Interrupt
129	GPINT5	GPIO Interrupt
130	GPINT6	GPIO Interrupt
131	GPINT7	GPIO Interrupt

Table 8-41 CIC1 Event Inputs (Secondary Events for EDMA3CC1 and EDMA3CC2) (Part 4 of 4)

Input Event # on CIC	System Interrupt	Description
132	Tracer_RAC_CFG_INTD	Tracer sliding time window interrupt for RAC
133	Tracer_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
134	Tracer_TAC_INTD	Tracer sliding time window interrupt for TAC
135	ARM_ETBFULLINT	ETB Full Interrupt for ARM
136	ARM_ETBACQINT	ETB ACQ Interrupt for ARM
137	QM_INT_HIGH_0	QM Interrupt
138	QM_INT_HIGH_1	QM Interrupt
139	QM_INT_HIGH_2	Reserved
140	QM_INT_HIGH_3	QM Interrupt
141	QM_INT_HIGH_4	QM Interrupt
142	QM_INT_HIGH_5	QM Interrupt
143	QM_INT_HIGH_6	QM Interrupt
144	QM_INT_HIGH_7	QM Interrupt
145	QM_INT_HIGH_8	QM Interrupt
146	QM_INT_HIGH_9	QM Interrupt
147	QM_INT_HIGH_10	Reserved
148	QM_INT_HIGH_11	QM Interrupt
149	QM_INT_HIGH_12	QM Interrupt
150	QM_INT_HIGH_13	QM Interrupt
151	QM_INT_HIGH_14	QM Interrupt
152	QM_INT_HIGH_15	Reserved
153	FFTC_A_INTD0	FFTC_A error event and FFTC_A debug event
154	FFTC_A_INTD1	FFTC_A error event and FFTC_A debug event
155	FFTC_A_INTD2	FFTC_A error event and FFTC_A debug event
156	FFTC_A_INTD3	FFTC_A error event and FFTC_A debug event
157	FFTC_B_INTD2	FFTC_B error event and FFTC_B debug event
158	FFTC_B_INTD3	FFTC_B error event and FFTC_B debug event
159	Tracer_DDR_2_INTD	Tracer interrupt for monitoring transactions sent to DDR3 EMIF
End of Table 8-41		

Table 8-42 CIC2 Event Inputs (Secondary Events for EDMA3CC0 and HyperLink) (Part 1 of 3)

Input Event # on CIC	System Interrupt	Description
0	GPINT0	GPIO Interrupt
1	GPINT1	GPIO Interrupt
2	GPINT2	GPIO Interrupt
3	GPINT3	GPIO Interrupt
4	Reserved	
5	Reserved	
6	GPINT6	GPIO Interrupt
7	GPINT7	GPIO Interrupt
8	GPINT8	GPIO Interrupt
9	GPINT9	GPIO Interrupt
10	GPINT10	GPIO Interrupt

Table 8-42 CIC2 Event Inputs (Secondary Events for EDMA3CC0 and HyperLink) (Part 2 of 3)

Input Event # on CIC	System Interrupt	Description
11	GPINT11	GPIO Interrupt
12	GPINT12	GPIO Interrupt
13	GPINT13	GPIO Interrupt
14	GPINT14	GPIO Interrupt
15	GPINT15	GPIO Interrupt
16	TETBHFULLINT	System TETB is half full
17	TETBFULLINT	System TETB is full
18	TETBACQINT	System Acquisition has been completed
19	TETBHFULLINT0	TETB0 is half full
20	TETBFULLINT0	TETB0 is full
21	TETBACQINT0	TETB0 Acquisition has been completed
22	TETBHFULLINT1	TETB1 is half full
23	TETBFULLINT1	TETB1 is full
24	TETBACQINT1	TETB1 Acquisition has been completed
25	TETBHFULLINT2	TETB2 is half full
26	TETBFULLINT2	TETB2 is full
27	TETBACQINT2	TETB2 Acquisition has been completed
28	TETBHFULLINT3	TETB3 is half full
29	TETBFULLINT3	TETB3 is full
30	TETBACQINT3	TETB3 Acquisition has been completed
31	Tracer_core_0_INTD	Tracer sliding time window interrupt for individual core
32	Tracer_core_1_INTD	Tracer sliding time window interrupt for individual core
33	Reserved	
34	Reserved	
35	Tracer_DDR_INTD	Tracer sliding time window interrupt for DDR3 EMIF
36	Tracer_MSMC_0_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank0
37	Tracer_MSMC_1_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank1
38	Tracer_MSMC_2_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank2
39	Tracer_MSMC_3_INTD	Tracer sliding time window interrupt for MSMC SRAM Bank3
40	Tracer_CFG_INTD	Tracer sliding time window interrupt for CFG0 TeraNet
41	Tracer_QM_CFG_INTD	Tracer sliding time window interrupt for QM_SS CFG
42	Tracer_QM_DMA_INTD	Tracer sliding time window interrupt for QM_SS Slave port
43	Tracer_SM_INTD	Tracer sliding time window interrupt for Semaphore
44	HyperLink_int_o	HyperLink Interrupt
45	Tracer_RAC_CFG_INTD	Tracer sliding time window interrupt for RAC
46	Tracer_RAC_FE_INTD	Tracer sliding time window interrupt for RAC_FE
47	Tracer_TAC_INTD	Tracer sliding time window interrupt for TAC
48	Tracer_DDR_2_INTD	Tracer interrupt for monitoring transactions sent to DDR3 EMIF
49	TINT4L	Timer64_4 Interrupt Low
50	TINT4H	Timer64_4 Interrupt High
51	TINT5L	Timer64_5 Interrupt Low
52	TINT5H	Timer64_5 Interrupt High
53	TINT6L	Timer64_6 Interrupt Low
54	TINT6H	Timer64_6 Interrupt High

Table 8-42 CIC2 Event Inputs (Secondary Events for EDMA3CC0 and HyperLink) (Part 3 of 3)

Input Event # on CIC	System Interrupt	Description
55	TINT7L	Timer64_7 Interrupt Low
56	TINT7H	Timer64_7 Interrupt High
57	Tracer_TNet_6P_A_INTD	Tracer interrupt for monitoring transactions sent to EMIF16
58	ARM_ETBFULLINT	ARM ETB FULL status interrupt
59	ARM_ETBACQINT	ARM ETB acquisition complete interrupt
60	Reserved	
61	DDR3_ERR	DDR3 EMIF Error Interrupt
62	Reserved	
63	Reserved	
End of Table 8-42		

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 1 of 7)

Input Event # on CP_CIC	System Interrupt	Description
0	SEMERR4	Semaphore interrupt
1	SEMERR5	Semaphore interrupt
2	SEMERR6	Semaphore interrupt
3	SEMINT4	Semaphore interrupt
4	SEMINT5	Semaphore interrupt
5	SEMINT6	Semaphore interrupt
6	Reserved	
7	Reserved	
8	EDMA3CC1_EDMACC_ERRINT	EDMA3CC1 error interrupt
9	EDMA3CC1_EDMACC_MPINT	EDMA3CC1 memory protection interrupt
10	EDMA3CC1_EDMATC_ERRINT0	EDMA3CC1 EDMA3TC0 error interrupt
11	EDMA3CC1_EDMATC_ERRINT1	EDMA3CC1 EDMA3TC1 error interrupt
12	EDMA3CC1_EDMATC_ERRINT2	EDMA3CC1 EDMA3TC2 error interrupt
13	EDMA3CC1_EDMATC_ERRINT3	EDMA3CC1 EDMA3TC3 error interrupt
14	EDMA3CC1_EDMACC__GINT	EDMA3CC1 GINT
15	EDMA3CC1_EDMA3CCINT3	EDMA3CC individual completion interrupt
16	EDMA3CC1_EDMA3CCINT7	EDMA3CC individual completion interrupt
17	EDMA3CC2_EDMACC_ERRINT	EDMA3CC2 error interrupt
18	EDMA3CC2_EDMACC_MPINT	EDMA3CC2 memory protection interrupt
19	EDMA3CC2_EDMATC_ERRINT0	EDMA3CC2 EDMA3TC0 error interrupt
20	EDMA3CC2_EDMATC_ERRINT1	EDMA3CC2 EDMA3TC1 error interrupt
21	EDMA3CC2_EDMATC_ERRINT2	EDMA3CC2 EDMA3TC2 error interrupt
22	EDMA3CC2_EDMATC_ERRINT3	EDMA3CC2 EDMA3TC3 error interrupt
23	EDMA3CC2_EDMACC__GINT	EDMA3CC2 GINT
24	EDMA3CC2_EDMA3CCINT3	EDMA3CC individual completion interrupt
25	EDMA3CC2_EDMA3CCINT7	EDMA3CC individual completion interrupt
26	EDMA3CC0_EDMACC_ERRINT	EDMA3CC0 error interrupt
27	EDMA3CC0_EDMACC_MPINT	EDMA3CC0 memory protection interrupt

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 2 of 7)

Input Event # on CP_CIC	System Interrupt	Description
28	EDMA3CC0_EDMATC_ERRINT0	EDMA3CC0 EDMA3TC0 error interrupt
29	EDMA3CC0_EDMATC_ERRINT1	EDMA3CC0 EDMA3TC1 error interrupt
30	EDMA3CC0_EDMACC_GINT	EDMA3CC0 GINT
31	EDMA3CC0_EDMA3CCINT3	EDMA3CC0 individual completion interrupt
32	EDMA3CC0_EDMA3CCINT7	EDMA3CC0 individual completion interrupt
33	GPINT0	GPIO
34	GPINT1	GPIO
35	GPINT2	GPIO
36	GPINT3	GPIO
37	GPINT4	GPIO
38	GPINT5	GPIO
39	GPINT6	GPIO
40	GPINT7	GPIO
41	GPINT8	GPIO
42	GPINT9	GPIO
43	GPINT10	GPIO
44	GPINT11	GPIO
45	GPINT12	GPIO
46	GPINT13	GPIO
47	GPINT14	GPIO
48	GPINT15	GPIO
49	GPINT16	GPIO
50	GPINT17	GPIO
51	GPINT18	GPIO
52	GPINT19	GPIO
53	GPINT20	GPIO
54	GPINT21	GPIO
55	GPINT22	GPIO
56	GPINT23	GPIO
57	GPINT24	GPIO
58	GPINT25	GPIO
59	GPINT26	GPIO
60	GPINT27	GPIO
61	GPINT28	GPIO
62	GPINT29	GPIO
63	GPINT30	GPIO
64	GPINT31	GPIO
65 - 74	Reserved	
75	INTDST0	SRIO
76	INTDST1	SRIO
77	INTDST2	SRIO

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 3 of 7)

Input Event # on CP_CIC	System Interrupt	Description
78	INTDST3	SRIO
79	INTDST4	SRIO
80	INTDST5	SRIO
81	INTDST6	SRIO
82	SPIINT0	SPI interrupt 0
83	SPIINT1	SPI interrupt 1
84	SPIXEVT	SPI Transmit Event
85	SPIREVT	SPI Receive Event
86	I2CINT	I2C Interrupt
87	I2CREVT	I2C receive event
88	I2CXEVT	I2C transmit event
89	KEYMGRINT_A	KEY_MGR A Interrupt
90	SECCTLINT	SEC_CTL Interrupt
92	TETBFULLINT	TETB Full Interrupt for Debug Subsystem
93	TETBACQINT	TETB Acquire Interrupt for Debug Subsystem
94	TETBOVFLINT	TETB Overflow Interrupt for Debug Subsystem
95	TETBUNFLINT	TETB Underflow Interrupt for Debug Subsystem
96	TETBHFULLINT0	TETB is half full for CorePac0
97	TETBFULLINT0	TETB is full for CorePac0
98	TETBACQINT0	TETB Acquisition has been completed for CorePac0
99	TETBOVFLINT0	TETB Overflow Interrupt for CorePac0
100	TETBUNFLINT0	TETB Underflow Interrupt for CorePac0
101	TETBHFULLINT1	TETB is half full for CorePac1
102	TETBFULLINT1	TETB is full for CorePac1
103	TETBACQINT1	TETB Acquisition has been completed for CorePac1
104	TETBOVFLINT1	TETB Overflow Interrupt for CorePac1
105	TETBUNFLINT1	TETB Underflow Interrupt for CorePac1
106	Reserved	
107	QM_INT_LOW_0	QM Interrupt
108	QM_INT_LOW_1	QM Interrupt
109	QM_INT_LOW_2	QM Interrupt
110	QM_INT_LOW_3	QM Interrupt
111	QM_INT_LOW_4	QM Interrupt
112	QM_INT_LOW_5	QM Interrupt
113	QM_INT_LOW_6	QM Interrupt
114	QM_INT_LOW_7	QM Interrupt
115	QM_INT_LOW_8	QM Interrupt
116	QM_INT_LOW_9	QM Interrupt
117	QM_INT_LOW_10	QM Interrupt
118	QM_INT_LOW_11	QM Interrupt
119	QM_INT_LOW_12	QM Interrupt

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 4 of 7)

Input Event # on CP_CIC	System Interrupt	Description
120	QM_INT_LOW_13	QM Interrupt
121	QM_INT_LOW_14	QM Interrupt
122	QM_INT_LOW_15	QM Interrupt
123	QM_INT_CDMA_0	QM Interrupt
124	QM_INT_CDMA_1	QM Interrupt
125	mdio_link_intr0	Packet Accelerator subsystem MDIO link interrupt
126	mdio_link_intr1	Packet Accelerator subsystem MDIO link interrupt
127	mdio_user_intr0	Packet Accelerator subsystem MDIO user interrupt
128	mdio_user_intr1	Packet Accelerator subsystem MDIO user interrupt
129	misc_intr	Packet Accelerator subsystem MDIO miscellaneous interrupt
130	PASS_INT_CDMA_0	PASS interrupt for Packet DMA starvation
131	Tracer_core_0	Tracer sliding time window interrupt for individual core
132	Tracer_core_1	Tracer sliding time window interrupt for individual core
133	Tracer_DDR	Tracer sliding time window interrupt for DDR3 EMIF
134	Tracer_MSMC_0	Tracer sliding time window interrupt for MSMC SRAM Bank0
135	Tracer_MSMC_1	Tracer sliding time window interrupt for MSMC SRAM Bank1
136	Tracer_MSMC_2	Tracer sliding time window interrupt for MSMC SRAM Bank2
137	Tracer_MSMC_3	Tracer sliding time window interrupt for MSMC SRAM Bank3
138	Tracer_CFG	Tracer sliding time window interrupt for CFG0 TeraNet
139	Tracer_QM_CFG	Tracer sliding time window interrupt for QM_SS CFG
140	Tracer_QM_DMA	Tracer sliding time window interrupt for QM_SS Slave port
141	Tracer_SM	Tracer sliding time window interrupt for Semaphore
142	Tracer_DDR_2	Tracer sliding time window interrupt for DDR3 EMIF
143	Tracer_RAC_CFG	Tracer sliding time window interrupt for RAC
144	Tracer_RAC_FE	Tracer sliding time window interrupt for RAC_FE
145	Tracer_TAC	Tracer sliding time window interrupt for TAC
146	PSC_ALLINT	PSC interrupt
147	BOOTCFG_ERR	BOOTCFG error interrupt
148	BOOTCFG_PROT	BOOTCFG protection interrupt
149	MPU7_ADDR_ERR_INT	MPU 7 addressing violation interrupt
150	MPU7_PROT_ERR_INT	MPU 7 protection violation interrupt
151	MPU0_ADDR_ERR_INT	MPU 0 addressing violation interrupt
152	MPU0_PROT_ERR_INT	MPU 0 protection violation interrupt
153	MPU1_ADDR_ERR_INT	MPU 1 addressing violation interrupt
154	MPU1_PROT_ERR_INT	MPU 1 protection violation interrupt
155	MPU2_ADDR_ERR_INT	MPU 2 addressing violation interrupt
156	MPU2_PROT_ERR_INT	MPU 2 protection violation interrupt
157	MPU3_ADDR_ERR_INT	MPU 3 addressing violation interrupt
158	MPU3_PROT_ERR_INT	MPU 3 protection violation interrupt
159	MPU4_ADDR_ERR_INT	MPU 4 addressing violation interrupt
160	MPU4_PROT_ERR_INT	MPU 4 protection violation interrupt

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 5 of 7)

Input Event # on CP_CIC	System Interrupt	Description
161	MPU5_ADDR_ERR_INT	MPU 5 addressing violation interrupt
162	MPU5_PROT_ERR_INT	MPU 5 protection violation interrupt
163	MPU6_ADDR_ERR_INT	MPU 6 addressing violation interrupt
164	MPU6_PROT_ERR_INT	MPU 6 protection violation interrupt
165	MSMC_dedc_cerror	Correctable (1-bit) soft error detected on SRAM read
166	MSMC_dedc_nc_error	Non-correctable (2-bit) soft error detected on SRAM read
167	MSMC_scrub_nc_error	Non-correctable (2-bit) soft error detected during scrub cycle
168	MSMC_scrub_cerror	Correctable (2-bit) soft error detected during scrub cycle
169	MSMC_mpf_error0	Memory protection fault indicators for each system master PrivID
170	MSMC_mpf_error1	Memory protection fault indicators for each system master PrivID
171	MSMC_mpf_error2	Memory protection fault indicators for each system master PrivID
172	MSMC_mpf_error3	Memory protection fault indicators for each system master PrivID
173	MSMC_mpf_error4	Memory protection fault indicators for each system master PrivID
174	MSMC_mpf_error5	Memory protection fault indicators for each system master PrivID
175	MSMC_mpf_error6	Memory protection fault indicators for each system master PrivID
176	MSMC_mpf_error7	Memory protection fault indicators for each system master PrivID
177	MSMC_mpf_error8	Memory protection fault indicators for each system master PrivID
178	MSMC_mpf_error9	Memory protection fault indicators for each system master PrivID
179	MSMC_mpf_error10	Memory protection fault indicators for each system master PrivID
180	MSMC_mpf_error11	Memory protection fault indicators for each system master PrivID
181	MSMC_mpf_error12	Memory protection fault indicators for each system master PrivID
182	MSMC_mpf_error13	Memory protection fault indicators for each system master PrivID
183	MSMC_mpf_error14	Memory protection fault indicators for each system master PrivID
184	MSMC_mpf_error15	Memory protection fault indicators for each system master PrivID
185	DDR3_ERR	DDR3_EMIF Error Interrupt
186	Reserved	
187	Reserved	
188	Reserved	
189	TCP3D_A_INTD	TCP3D A error interrupt for TCP3DINT0 and TCP3DINT1
190	Reserved	
191	Reserved	
192	UARTINT0	UART0 Interrupt
193	URXEVT0	UART0 Receive Event
194	UTXEVT0	UART0 Transmit Event
195	UARTINT1	UART1 Interrupt
196	URXEVT1	UART1 Receive Event
197	UTXEVT1	UART1 Transmit Event
198	SmartReflex_intrreq0	SmartReflex sensor interrupt
199	SmartReflex_intrreq1	SmartReflex sensor interrupt
200	SmartReflex_intrreq2	SmartReflex sensor interrupt
201	SmartReflex_intrreq3	SmartReflex sensor interrupt

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 6 of 7)

Input Event # on CP_CIC	System Interrupt	Description
202	VPNoSMPSAck	VPVOLTUPDATE has been asserted but SMPS has not been responded to in a defined time interval
203	VPEqValue	SRSINTERUPT is asserted, but the new voltage is not different from the current SMPS voltage
204	VPMaxVdd	The new voltage required is equal to or greater than MaxVdd
205	VPMinVdd	The new voltage required is equal to or less than MinVdd
206	VPINIDLE	Indicating that the FSM of voltage processor is in idle
207	VPOPPChangeDone	Indicating that the average frequency error is within the desired limit
208	po_vcon_smpserr_intr	SMPS error interrupt
209	po_vp_smpsack_intr	SMPS acknowledgment interrupt
210	FFTC_A_INTD0	FFTC_A error event and FFTC_A debug event
211	FFTC_A_INTD1	FFTC_A error event and FFTC_A debug event
212	FFTC_A_INTD2	FFTC_A error event and FFTC_A debug event
213	FFTC_A_INTD3	FFTC_A error event and FFTC_A debug event
214	Reserved	
215	Reserved	
216	INTDST7	SRIO
217	INTDST8	SRIO
218	INTDST9	SRIO
219	INTDST10	SRIO
220	INTDST11	SRIO
221	INTDST12	SRIO
222	INTDST13	SRIO
223	INTDST14	SRIO
224	AIF_INTD	combined AIF_EVT0,AIF_EVT1,AIF_EVT_Starvation
225	Reserved	
226	Reserved	
227	AIF_SEVT0	AIF radio timing sync event 0
228	AIF_SEVT1	AIF radio timing sync event 1
229	AIF_SEVT2	AIF radio timing sync event 2
230	AIF_SEVT3	AIF radio timing sync event 3
231	AIF_SEVT4	AIF radio timing sync event 4
232	AIF_SEVT5	AIF radio timing sync event 5
233	AIF_SEVT6	AIF radio timing sync event 6
234	AIF_SEVT7	AIF radio timing sync event 7
235	Reserved	
236	PONIRQ	USIM interrupt through INTD, level to rising edge sensitivity event
237	INTDST15	SRIO
238	Reserved	
239	ARM_ETBFULLINT	ARM ETB full
240	ARM_ETBACQINT	ARM ETB acquisition has been completed
241	KEYMGRINT_B	KEY_MGR B interrupt

Table 8-43 CIC3 Event Inputs (Events for ARM) (Part 7 of 7)

Input Event # on CP_CIC	System Interrupt	Description
242	BCP_ERROR0	BCP
243	BCP_ERROR1	BCP
244	BCP_ERROR2	BCP
245	BCP_ERROR3	BCP
246	Rapid_INT_CDMA_0	SRIO Packet DMA starvation interrupt
247	Tracer_TNet_6P_A	Tracer interrupt for monitoring transactions sent to TNet_6P_A
248	FFTC_B_INTD0	FFTC_B error event and FFTC_B debug event
249	FFTC_B_INTD1	FFTC_B error event and FFTC_B debug event
250	FFTC_B_INTD2	FFTC_B error event and FFTC_B debug event
251	FFTC_B_INTD3	FFTC_B error event and FFTC_B debug event
252	POSDMARREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
253	POSDMAWREQ_INTD	From USIM, through a dedicated INTD, level to rising edge sensitivity event
254	TCP3D_B_INTD	TCP3D B error interrupt for TCP3DINT0 and TCP3DINT1
255	Reserved	
End of Table 8-43		

8.10.2 CIC Registers

This section includes the CIC memory map information and registers. The CIC prioritization feature is implemented in CIC3 only (events for ARM). See the *Interrupt Controller (CIC) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.10.2.1 CIC0 Register Map

Table 8-44 CIC0 Registers (Part 1 of 4)

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
0x214	RAW_STATUS_REG5	Raw Status Register 5
0x218	RAW_STATUS_REG6	Raw Status Register 6
0x280	ENA_STATUS_REG0	Enabled Status Register 0

Table 8-44 CIC0 Registers (Part 2 of 4)

Address Offset	Register Mnemonic	Register Name
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
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
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
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

Table 8-44 CIC0 Registers (Part 3 of 4)

Address Offset	Register Mnemonic	Register Name
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
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
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

Table 8-44 CIC0 Registers (Part 4 of 4)

Address Offset	Register Mnemonic	Register Name
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
End of Table 8-44		

8.10.2.2 CIC1 Register Map

Table 8-45 CIC1 Registers (Part 1 of 3)

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
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
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
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
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

Table 8-45 CIC1 Registers (Part 2 of 3)

Address Offset	Register Mnemonic	Register Name
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
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
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

Table 8-45 CIC1 Registers (Part 3 of 3)

Address Offset	Register Mnemonic	Register Name
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
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
End of Table 8-45		

8.10.2.3 CIC2 Register Map

Table 8-46 CIC2 Registers (Part 1 of 2)

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
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1
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

Table 8-46 CIC2 Registers (Part 2 of 2)

Address Offset	Register Mnemonic	Register Name
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
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
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
End of Table 8-46		

Table 8-47 CIC3 Registers (Part 1 of 4)

Address Offset	Register Mnemonic	Register Name
0x0	REVISION_REG	Revision Register
0x4	CONTROL_REG	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
0x80	GLB_PRI_INTR_REG	Global Prioritized index Register
0x200	RAW_STATUS_REG0	Raw Status Register 0
0x204	RAW_STATUS_REG1	Raw Status Register 1
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1
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

Table 8-47 CIC3 Registers (Part 2 of 4)

Address Offset	Register Mnemonic	Register Name
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
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

Table 8-47 CIC3 Registers (Part 3 of 4)

Address Offset	Register Mnemonic	Register Name
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
0x4ec	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
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
0x900	PRI_HINT_REG0	Host Interrupt Prioritized Index Register 0
0x904	PRI_HINT_REG1	Host Interrupt Prioritized Index Register 1
0x908	PRI_HINT_REG2	Host Interrupt Prioritized Index Register 2
0x90c	PRI_HINT_REG3	Host Interrupt Prioritized Index Register 3
0x910	PRI_HINT_REG4	Host Interrupt Prioritized Index Register 4
0x914	PRI_HINT_REG5	Host Interrupt Prioritized Index Register 5
0x918	PRI_HINT_REG6	Host Interrupt Prioritized Index Register 6
0x91c	PRI_HINT_REG7	Host Interrupt Prioritized Index Register 7
0x920	PRI_HINT_REG8	Host Interrupt Prioritized Index Register 8
0x924	PRI_HINT_REG9	Host Interrupt Prioritized Index Register 9
0x928	PRI_HINT_REG10	Host Interrupt Prioritized Index Register 10
0x92c	PRI_HINT_REG11	Host Interrupt Prioritized Index Register 11
0x930	PRI_HINT_REG12	Host Interrupt Prioritized Index Register 12
0x934	PRI_HINT_REG13	Host Interrupt Prioritized Index Register 13
0x938	PRI_HINT_REG14	Host Interrupt Prioritized Index Register 14
0x93c	PRI_HINT_REG15	Host Interrupt Prioritized Index Register15
0x940	PRI_HINT_REG16	Host Interrupt Prioritized Index Register 16

Table 8-47 CIC3 Registers (Part 4 of 4)

Address Offset	Register Mnemonic	Register Name
0x944	PRI_HINT_REG17	Host Interrupt Prioritized Index Register 17
0x948	PRI_HINT_REG18	Host Interrupt Prioritized Index Register 18
0x94c	PRI_HINT_REG19	Host Interrupt Prioritized Index Register 19
0x950	PRI_HINT_REG20	Host Interrupt Prioritized Index Register 20
0x954	PRI_HINT_REG21	Host Interrupt Prioritized Index Register 21
0x958	PRI_HINT_REG22	Host Interrupt Prioritized Index Register 22
0x95c	PRI_HINT_REG23	Host Interrupt Prioritized Index Register 23
0x960	PRI_HINT_REG24	Host Interrupt Prioritized Index Register 24
0x964	PRI_HINT_REG25	Host Interrupt Prioritized Index Register 25
0x968	PRI_HINT_REG26	Host Interrupt Prioritized Index Register 26
0x96c	PRI_HINT_REG27	Host Interrupt Prioritized Index Register 27
0x970	PRI_HINT_REG28	Host Interrupt Prioritized Index Register 28
0x974	PRI_HINT_REG29	Host Interrupt Prioritized Index Register 29
0x978	PRI_HINT_REG30	Host Interrupt Prioritized Index Register 30
0x97c	PRI_HINT_REG31	Host Interrupt Prioritized Index Register 31
0x980	PRI_HINT_REG32	Host Interrupt Prioritized Index Register 32
0x984	PRI_HINT_REG33	Host Interrupt Prioritized Index Register 33
0x988	PRI_HINT_REG34	Host Interrupt Prioritized Index Register 34
0x98c	PRI_HINT_REG35	Host Interrupt Prioritized Index Register 35
0x990	PRI_HINT_REG36	Host Interrupt Prioritized Index Register 36
0x994	PRI_HINT_REG37	Host Interrupt Prioritized Index Register 37
0x998	PRI_HINT_REG38	Host Interrupt Prioritized Index Register 38
0x99c	PRI_HINT_REG39	Host Interrupt Prioritized Index Register 39
0x9a0	PRI_HINT_REG40	Host Interrupt Prioritized Index Register 40
0x9a4	PRI_HINT_REG41	Host Interrupt Prioritized Index Register 41
0x9a8	PRI_HINT_REG42	Host Interrupt Prioritized Index Register 42
0x9ac	PRI_HINT_REG43	Host Interrupt Prioritized Index Register 43
0x9b0	PRI_HINT_REG44	Host Interrupt Prioritized Index Register 44
0x9b4	PRI_HINT_REG45	Host Interrupt Prioritized Index Register 45
0x9b8	PRI_HINT_REG46	Host Interrupt Prioritized Index Register 46
0x9bc	PRI_HINT_REG47	Host Interrupt Prioritized Index Register 47
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
End of Table 8-47		

8.10.3 Inter-Processor Register Map

Table 8-48 IPC Generation Registers (IPCGRx) (Part 1 of 2)

Address Start	Address End	Size	Register Name	Description
0x02620200	0x02620203	4B	NMIGR0	NMI Event Generation Register for CorePac0
0x02620204	0x02620207	4B	NMIGR1	NMI Event Generation Register for CorePac1
0x02620208	0x0262020B	4B	Reserved	Reserved
0x0262020C	0x0262020F	4B	Reserved	Reserved
0x02620210	0x02620213	4B	Reserved	Reserved

Table 8-48 IPC Generation Registers (IPCGRx) (Part 2 of 2)

Address Start	Address End	Size	Register Name	Description
0x02620214	0x02620217	4B	Reserved	Reserved
0x02620218	0x0262021B	4B	Reserved	Reserved
0x0262021C	0x0262021F	4B	Reserved	Reserved
0x02620220	0x0262023F	32B	Reserved	Reserved
0x02620240	0x02620243	4B	IPCGR0	IPC Generation Register for CorePac0
0x02620244	0x02620247	4B	IPCGR1	IPC Generation Register for CorePac1
0x02620248	0x0262024B	4B	Reserved	Reserved
0x0262024C	0x0262024F	4B	Reserved	Reserved
0x02620250	0x02620253	4B	Reserved	Reserved
0x02620254	0x02620257	4B	Reserved	Reserved
0x02620258	0x0262025B	4B	Reserved	Reserved
0x0262025C	0x0262025F	4B	Reserved	Reserved
0x02620260	0x0262027B	28B	Reserved	Reserved
0x0262027C	0x0262027F	4B	IPCGRH	IPC Generation Register for ARM
0x02620280	0x02620283	4B	IPCAR0	IPC Acknowledgement Register for CorePac0
0x02620284	0x02620287	4B	IPCAR1	IPC Acknowledgement Register for CorePac1
0x02620288	0x0262028B	4B	Reserved	Reserved
0x0262028C	0x0262028F	4B	Reserved	Reserved
0x02620290	0x02620293	4B	Reserved	Reserved
0x02620294	0x02620297	4B	Reserved	Reserved
0x02620298	0x0262029B	4B	Reserved	Reserved
0x0262029C	0x0262029F	4B	Reserved	Reserved
0x026202A0	0x026202BB	28B	Reserved	Reserved
0x026202BC	0x026202BF	4B	IPCARH	IPC Acknowledgement Register for ARM
End of Table 8-48				

8.10.4 $\overline{\text{NMI}}$ 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 watch dog timers. One $\overline{\text{NMI}}$ pin and one $\overline{\text{LRESET}}$ pin are shared by both CorePacs on the device. The CORESEL[2:0] pins can be configured to select between the two CorePacs available as shown in Table 8-49.

Table 8-49 $\overline{\text{LRESET}}$ and $\overline{\text{NMI}}$ Decoding (Part 1 of 2)

CORESEL[2:0] Pin Input	$\overline{\text{LRESET}}$ Pin Input	$\overline{\text{NMI}}$ Pin Input	$\overline{\text{LRESETNMIEN}}$ Pin Input	Reset Mux Block Output
XXX	X	X	1	No local reset or $\overline{\text{NMI}}$ assertion
000	0	X	0	Assert local reset to CorePac0
001	0	X	0	Assert local reset to CorePac1
010	X	X	X	Reserved
011	X	X	X	Reserved
1xx	0	X	0	Assert local reset to all CorePacs
000	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac0
001	1	1	0	De-assert local reset & $\overline{\text{NMI}}$ to CorePac1
010	X	X	X	Reserved
011	X	X	X	Reserved

Table 8-49 LRESET and NMI Decoding (Part 2 of 2)

CORESEL[2:0] Pin Input	LRESET Pin Input	NMI Pin Input	LRESETNMIEN Pin Input	Reset Mux Block Output
1xx	1	1	0	De-assert local reset & NMI to all CorePacs
000	1	0	0	Assert NMI to CorePac0
001	1	0	0	Assert NMI to CorePac1
010	X	X	X	Reserved
011	X	X	X	Reserved
1xx	1	0	0	Assert NMI to all CorePacs
End of Table 8-49				

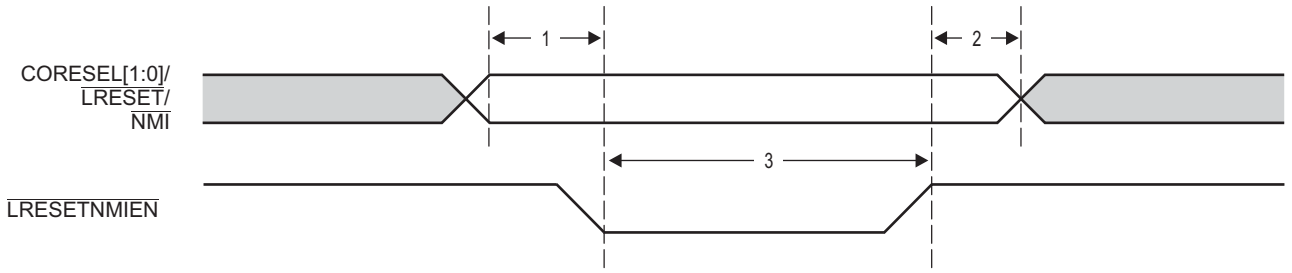
8.10.5 External Interrupts Electrical Data/Timing

Table 8-50 NMI and LRESET Timing Requirements (1)
(see Figure 8-30)

No.			Min	Max	Unit
1	tsu(LRESET-LRESETNMIENL)	Setup Time - LRESET valid before LRESETNMIEN low	12*P		ns
1	tsu(NMI-LRESETNMIENL)	Setup Time - NMI valid before LRESETNMIEN low	12*P		ns
1	tsu(CORESELn-LRESETNMIENL)	Setup Time - CORESEL[2:0] valid before LRESETNMIEN low	12*P		ns
2	th(LRESETNMIENL-LRESET)	Hold Time - LRESET valid after LRESETNMIEN low	12*P		ns
2	th(LRESETNMIENL-NMI)	Hold Time - NMI valid after LRESETNMIEN low	12*P		ns
2	th(LRESETNMIENL-CORESELn)	Hold Time - CORESEL[2:0] valid after LRESETNMIEN low	12*P		ns
3	tw(LRESETNMIEN)	Pulse Width - LRESETNMIEN low width	12*P		ns
End of Table 8-50					

1 P = 1/SYSCLK1 clock frequency in ns.

Figure 8-30 NMI and LRESET Timing



The ARM does not support local reset in the TCI6612. The local reset event generated by the watchdog timer for the ARM is used to trigger a device reset instead. The NMI event generated by the ARM's watchdog timer is routed to the ARM's interrupt input event, because the ARM does not have a specific NMI input event.

8.10.6 Host Interrupt Output

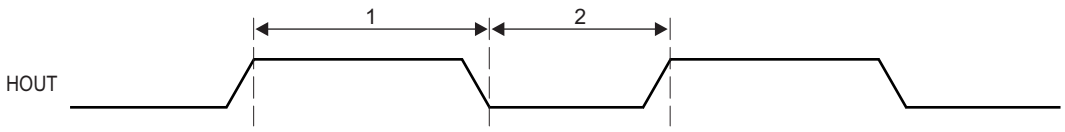
The C66x CorePac can assert an event to the external host processor using HOUT. Table 8-51 shows the timing for the HOUT pulse. See section 3.3.14 for more details.

Table 8-51 HOUT Switching Characteristics
(see Figure 8-31)

No.			Min	Max	Unit
1	$t_{w(HOUTH)}$	HOUT pulse duration high	24*P (1)		ns
2	$t_{w(HOUTL)}$	HOUT pulse duration low	24*P		ns
End of Table 8-51					

1 P = 1/SYSCLK1 clock frequency in ns.

Figure 8-31 HOUT Timing



8.11 Memory Protection Unit (MPU)

- The TCI6612 supports 8 MPUs:
- One MPU is used to protect the main CPU/3 CFG TeraNet (CFG space of all slave devices on the TeraNet is protected by the MPU).
- Two MPUs are used for the packet DMA (one for DATA PORT and another is for CFG PORT).
- One MPU is used for the Semaphore.
- One MPU is used for the RAC.
- One MPU is used for monitoring the traffic to the BCP_CFG port
- One MPU is used for monitoring the traffic to the DDR3_EMIF
- One MPU is used for monitoring the traffic to the EMIF16
- 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 *Memory Protection Unit (MPU) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.
- Table 8-52 and Table 8-53 show the configuration of each MPU and the memory regions protected by each MPU.

Table 8-52 MPU Default Configuration

Setting	MPU0 (Main CFG TeraNet)	MPU1 (QM_SS DATA PORT)	MPU2 (QM_SS CFG PORT)	MPU3 (Semaphore)	MPU4 (RAC)	MPU5 (BCP_CFG PORT)	MPU6 (DDR3_EMIF)	MPU7 (EMIF16)
Default permission	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed
Number of allowed IDs supported	16	16	16	16	16	16	16	16
Number of programmable ranges supported	16	5	16	1	2	1	16	16
Compare width	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity
End of Table 8-52								

Table 8-53 MPU Memory Regions

	Memory Protection	Start Address	End Address
MPU0	Main CFG TeraNet	0x01D00000	0x026207FF
MPU1	QM_SS DATA PORT	0x34000000	0x340BFFFF
MPU2	QM_SS CFG PORT	0x02A00000	0x02ABFFFF
MPU3	Semaphore	0x02640000	0x026407FF
MPU4	RAC	0x01F80000	0x0215FFFF
MPU5	BCP_CFG PORT	0x33600000	0x337FFFFF
MPU6	DDR3_EMIF	0x80000000	0xFFFFFFFF
MPU7	EMIF16	0x70000000	0x7FFFFFFF
End of Table 8-53			

Table 8-54 shows the unique Master ID assigned to each CorePac and peripherals on the device.

Table 8-54 Master ID Settings (Part 1 of 3)

Master ID	Target
0	CorePac0
1	CorePac1
2	Reserved
3	Reserved
4	ARM_Port1
5	Reserved
6	Reserved
7	Reserved
8	CorePac0 CFG
9	CorePac1 CFG
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	EDMA0_TC0 read
17	EDMA0_TC0 write
18	EDMA0_TC1 read
19	EDMA0_TC1 write
20	EDMA1_TC0 read
21	EDMA1_TC0 write
22	EDMA1_TC1 read
23	EDMA1_TC1write
24	EDMA1_TC2 read
25	EDMA1_TC2 write
26	EDMA1_TC3 read
27	EDMA1_TC3 write
28	EDMA2_TC0 read
29	EDMA2_TC0 write
30	EDMA2_TC1 read
31	EDMA2_TC1 write
32	EDMA2_TC2 read
33	EDMA2_TC2 write
34	EDMA2_TC3 read
35	EDMA2_TC3 write
36 to 37	Reserved
38 to 39	SRIO PKTDMA
40	FFTC_A
41	Reserved
42	FFTC_B
43	Reserved
44	Reserved

Table 8-54 Master ID Settings (Part 2 of 3)

Master ID	Target
45	Reserved
46	RAC_BE0
47	RAC_BE1
48	DAP
49	EDMA3CC0
50	EDMA3CC1
51	EDMA3CC2
52	MSMC (1)
53	PCIe
54	SRIO_M
55	HyperLink
56 to 59	Queue Manager
60 to 63	Reserved
64 to 71	AIF2
72 to 85	Reserved
86	Reserved
87	Reserved
88 to 91	Queue Manager Packet DMA
92 to 93	Network Coprocessor
94	TAC
95	Reserved
96	BCP_DIO1
97	BCP_DIO0
98	BCP_CDMA
99-127	Reserved
128	Tracer for CorePac0 L2 bank (2)
129	Tracer for CorePac1 L2 bank
130	Reserved
131	Reserved
132	Reserved
133	Reserved
134	Reserved
135	Reserved
136	Tracer_MSMC0
137	Tracer_MSMC1
138	Tracer_MSMC2
139	Tracer_MSMC3
140	Tracer_DDR
141	Tracer_SM
142	Tracer_QM_CFG
143	Tracer_QM_DMA
144	Tracer_CFG
145	Tracer_RAC_FE
146	Tracer_RAC_CFG

Table 8-54 Master ID Settings (Part 3 of 3)

Master ID	Target
147	Tracer_TAC
148	Tracer_TNet_6P_A
149	Tracer_DDR_2
150-223	Reserved
224-255	ARM_port0
End of Table 8-54	

1 The master ID for MSMC is for the transactions initiated by MSMC internally and sent to the DDR.
2 All traces are set to the same master ID and bit 7 of the master ID needs to be 1.

Table 8-55 shows the privilege ID of each CorePac and every mastering peripheral. Table 8-55 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 8-55 Device Master Settings

Privilege ID	Master	Privilege Level	Security Level	Access Type
0	CorePac0	SW dependant, driven by MSMC	SW dependant	DMA
1	CorePac1	SW dependant, driven by MSMC	SW dependant	DMA
2	Reserved	N/A	N/A	N/A
3	Reserved	N/A	N/A	N/A
4	AIF	User	Non-secure	DMA
5	TAC	User	Non-secure	DMA
6	RAC	User	Non-secure	DMA
7	ARM	User	Non-secure	DMA
8	PA_SS/FFTC/BCP/SRIO_CPPI/QM_CDMA	User	Non-secure	DMA
9	SRIO_M	User/Driven by SRIO block, 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	User	Non-secure	DMA
11	PCIe	Supervisor	Non-secure	DMA
12	DAP	Driven by debug_SS	Driven by debug_SS	DMA
13	HyperLink	Supervisor	Non-secure	DMA
14	HyperLink	Supervisor	Non-secure	DMA
15	BCP	User	Non-secure	DMA
End of Table 8-55				

8.11.1 MPU Registers

This section includes the offsets for MPU registers and definitions for device-specific MPU registers.

8.11.1.1 MPU Register Map

Table 8-56 MPU0 Registers (Part 1 of 2)

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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 10, memory page protection attributes
2B0h	PROG11_MPSAR	Programmable range 11, start address
2B4h	PROG11_MPEAR	Programmable range 11, end address

Table 8-56 MPU0 Registers (Part 2 of 2)

Offset	Name	Description
2B8h	PROG11_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-56		

Table 8-57 MPU1 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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 3, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-57		

Table 8-58 MPU2 Registers (Part 1 of 2)

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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 11, memory page protection attributes
2C0h	PROG12_MPSAR	Programmable range 12, start address

Table 8-58 MPU2 Registers (Part 2 of 2)

Offset	Name	Description
2C4h	PROG12_MPEAR	Programmable range 12, end address
2C8h	PROG12_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-58		

Table 8-59 MPU3 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_MPPA	Programmable range 0, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-59		

Table 8-60 MPU4 Registers (Part 1 of 2)

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

Table 8-60 MPU4 Registers (Part 2 of 2)

Offset	Name	Description
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPA	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_MPPA	Programmable range 1, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-60		

Table 8-61 MPU5 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_MPPA	Programmable range 0, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-61		

Table 8-62 MPU6 Registers (Part 1 of 3)

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_MPPA	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_MPPA	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address

Table 8-62 MPU6 Registers (Part 2 of 3)

Offset	Name	Description
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address

Table 8-62 MPU6 Registers (Part 3 of 3)

Offset	Name	Description
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-62		

Table 8-63 MPU7 Registers (Part 1 of 2)

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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 9, memory page protection attributes

Table 8-63 MPU7 Registers (Part 2 of 2)

Offset	Name	Description
2A0h	PROG10_MPSAR	Programmable range 10, start address
2A4h	PROG10_MPEAR	Programmable range 10, end address
2A8h	PROG10_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	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_MPPA	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear
End of Table 8-63		

8.11.1.2 Device-Specific MPU Registers

8.11.1.2.1 Configuration Register (CONFIG)

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

Figure 8-32 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-5	R-16	R-0	R-1
	MPU2	R-0	R-0	R-16	R-16	R-0	R-1
	MPU3	R-0	R-0	R-1	R-16	R-0	R-1
	MPU4	R-0	R-0	R-2	R-16	R-0	R-1
	MPU5	R-0	R-0	R-1	R-16	R-0	R-1
	MPU6	R-0	R-0	R-16	R-16	R-0	R-1
	MPU7	R-0	R-0	R-16	R-16	R-0	R-1

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

Table 8-64 Configuration Register Field Descriptions

Bit	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 reads 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
End of Table 8-64		

8.11.2 MPU Programmable Range Registers

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

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

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

Figure 8-33 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 8-65 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.
End of Table 8-65		

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

The programmable address end register holds the end address for the range. This register is writeable by a supervisor entity only. If NS = 0 (non-secure mode) in the associated MPPA 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 8-34 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 8-66 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 0.
End of Table 8-66		

8.11.2.3 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPA)

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 8-35 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPA)

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 8-67 Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (Part 1 of 3)

Bit	Field	Description
31 – 26	Reserved	Reserved. Always reads as 0.
25	AID15	Controls permission check of ID = 15 0 = AID is not checked for permissions 1 = AID is checked for permissions
24	AID14	Controls permission check of ID = 14 0 = AID is not checked for permissions 1 = AID is checked for permissions
23	AID13	Controls permission check of ID = 13 0 = AID is not checked for permissions 1 = AID is checked for permissions
22	AID12	Controls permission check of ID = 12 0 = AID is not checked for permissions 1 = AID is checked for permissions

Table 8-67 Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (Part 2 of 3)

Bit	Field	Description
21	AID11	Controls permission check of ID = 11 0 = AID is not checked for permissions 1 = AID is checked for permissions
20	AID10	Controls permission check of ID = 10 0 = AID is not checked for permissions 1 = AID is checked for permissions
19	AID9	Controls permission check of ID = 9 0 = AID is not checked for permissions 1 = AID is checked for permissions
18	AID8	Controls permission check of ID = 8 0 = AID is not checked for permissions 1 = AID is checked for permissions
17	AID7	Controls permission check of ID = 7 0 = AID is not checked for permissions 1 = AID is checked for permissions
16	AID6	Controls permission check of ID = 6 0 = AID is not checked for permissions 1 = AID is checked for permissions
15	AID5	Controls permission check of ID = 5 0 = AID is not checked for permissions 1 = AID is checked for permissions
14	AID4	Controls permission check of ID = 4 0 = AID is not checked for permissions 1 = AID is checked for permissions
13	AID3	Controls permission check of ID = 3 0 = AID is not checked for permissions 1 = AID is checked for permissions
12	AID2	Controls permission check of ID = 2 0 = AID is not checked for permissions 1 = AID is checked for permissions
11	AID1	Controls permission check of ID = 1 0 = AID is not checked for permissions 1 = AID is checked for permissions
10	AID0	Controls permission check of ID = 0 0 = AID is not checked for permissions 1 = AID is checked for permissions
9	AIDX	Controls permission check of ID > 15 0 = AID is not checked for permissions 1 = AID 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
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

Table 8-67 Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (Part 3 of 3)

Bit	Field	Description
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
End of Table 8-67		

8.11.2.4 MPU Registers Reset Values

Table 8-68 Programmable Range *n* Registers Reset Values for MPU0

	MPU0 (Main CFG TeraNet)			
Programmable Range	**Start Address (PROGn_MPSAR)**	**End Address (PROGn_MPEAR)**	**Memory Page Protection Attribute (PROGn_MPPA)**	**Memory Protection**
PROG0	0x01D0_0000	0x01D8_83FF	0x03FF_FCB6	Tracers
PROG1	0x01F0_0000	0x01F7_FFFF	0x03FF_FCB6	AIF2
PROG2	0x0200_0000	0x020F_FFFF	0x03FF_FCB6	NETCP
PROG3	0x0218_0000	0x021A_FFFF	0x03FF_FCB6	TAC
PROG4	0x021C_0000	0x021E_0FFF	0x03FF_FCB6	TCP/VCP
PROG5	0x021F_0000	0x021F_7FFF	0x03FF_FCB6	FFTC
PROG6	0x0220_0000	0x022B_FFFF	0x03FF_FCB6	Timers
PROG7	0x0231_0000	0x0231_03FF	0x03FF_FCB4	PLL
PROG8	0x0232_0000	0x0232_03FF	0x03FF_FCB6	GPIO
PROG9	0x0233_0000	0x0233_03FF	0x03FF_FCB4	SmartReflex
PROG10	0x0235_0000	0x0235_0FFF	0x03FF_FCB4	PSC
PROG11	0x0240_0000	0x0247_3FFF	0x03FF_FCB6	Tracer Formatters
PROG12	0x0250_0000	0x0252_FFFF	0x03FF_FCB4	SEC, USIM, OTP memory
PROG13	0x0253_0000	0x025A_5FFF	0x03FF_FCB6	I^2C, UART, DEBUG_SS
PROG14	0x025A_7000	0x0260_BFFF	0x03FF_FCB6	CICs
PROG15	0x0262_0000	0x0262_07FF	0x03FF_FCB4	Chip-level Registers
End of Table 8-68				

Table 8-69 Programmable Range *n* Registers Reset Values for MPU1

Programmable Range	MPU1 (QM_SS DATA PORT)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x3400_0000	0x3401_FFFF	0x03FF_FC80	Queue Manager subsystem data
PROG1	0x3402_0000	0x3406_1FFF	0x0061_8080	
PROG2	0x3406_0000	0x3406_7FFF	0x03FF_FCF4	
PROG3	0x3406_2000	0x3406_7FFF	0x03FF_FCA4	
PROG4	0x340B_8000	0x340B_FFFF	0x03FF_FCB6	
End of Table 8-69				

Table 8-70 Programmable Range *n* Registers Reset Values for MPU2

Programmable Range	MPU2 (QM_SS CFG PORT)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x02A0_0000	0x02A1_FFFF	0x03FF_FCE4	Queue Manager subsystem configuration
PROG1	0x02A2_0000	0x02A3_FFFF	0x0061_8080	
PROG2	0x02A4_0000	0x02A5_FFFF	0x0061_8080	
PROG3	0x02A6_0000	0x02A6_1FFF	0x03FF_FCF4	
PROG4	0x02A6_8000	0x02A6_8FFF	0x03FF_FCF4	
PROG5	0x02A6_2000	0x02A6_7FFF	0x03FF_FCA4	
PROG6	0x02A6_A000	0x02A6_AFFF	0x03FF_FCF4	
PROG7	0x02A6_B000	0x02A6_BFFF	0x03FF_FCF4	
PROG8	0x02A6_C000	0x02A6_DFFF	0x03FF_FCF4	
PROG9	0x02A6_E000	0x02A6_FFFF	0x03FF_FCF4	
PROG10	0x02A8_0000	0x02A8_FFFF	0x03FF_FCE4	
PROG11	0x02A9_0000	0x02A9_FFFF	0x03FF_FCF4	
PROG12	0x02AA_0000	0x02AA_7FFF	0x03FF_FCF6	
PROG13	0x02AA_8000	0x02AA_FFFF	0x03FF_FCF6	
PROG14	0x02AB_0000	0x02AB_7FFF	0x03FF_FCF6	
PROG15	0x02AB_8000	0x02AB_FFFF	0x03FF_FCF6	
End of Table 8-70				

Table 8-71 Programmable Range *n* Registers Reset Values for MPU3

Programmable Range	MPU3 (Semaphore)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attributes (PROGn_MPPA)	Memory Protection
PROG0	0x0264_0000	0x0264_07FF	0x03FD_C080	Semaphore
End of Table 8-71				

Table 8-72 Programmable Range *n* Registers Reset Values for MPU4

Programmable Range	MPU4 (RAC)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x0210_0000	0x0215_FFFF	0x03FF_C080	RAC_A_CFG
PROG1	0x01F8_0000	0x01FD_FFFF	0x0003_FCB6	RAC_B_CFG
PROG2	0x0258_0000	0x025B_FFFF	0x03FF_FCB6	Debug_SS
End of Table 8-72				

Table 8-73 Programmable Range *n* Registers Reset Values for MPU5

Programmable Range	MPU5 (BCP_CFG PORT)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x3520_0000	0x3521_FFFF	0x03FF_FCB6	BCP_CFG
End of Table 8-73				

Table 8-74 Programmable Range *n* Registers Reset Values for MPU6

Programmable Range	MPU6 (DDR3_EMIF)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x0000_0000	0x07FF_FFFF	0x03FF_FCFF	DDR3 EMIF data
PROG1	0x0800_0000	0x0FFF_FFFF	0x03FF_FCFF	
PROG2	0x1000_0000	0x17FF_FFFF	0x03FF_FCFF	
PROG3	0x1800_0000	0x1FFF_FFFF	0x03FF_FCFF	
PROG4	0x2000_0000	0x27FF_FFFF	0x03FF_FCFF	
PROG5	0x2800_0000	0x2FFF_FFFF	0x03FF_FCFF	
PROG6	0x3000_0000	0x37FF_FFFF	0x03FF_FCFF	
PROG7	0x3800_0000	0x3FFF_FFFF	0x03FF_FCFF	
PROG8	0x4000_0000	0x47FF_FFFF	0x03FF_FCFF	
PROG9	0x4800_0000	0x4FFF_FFFF	0x03FF_FCFF	
PROG10	0x5000_0000	0x57FF_FFFF	0x03FF_FCFF	
PROG11	0x5800_0000	0x5FFF_FFFF	0x03FF_FCFF	
PROG12	0x6000_0000	0x67FF_FFFF	0x03FF_FCFF	
PROG13	0x6800_0000	0x6FFF_FFFF	0x03FF_FCFF	
PROG14	0x7000_0000	0x77FF_FFFF	0x03FF_FCFF	
PROG15	0x7800_0000	0x7FFF_FFFF	0x03FF_FCFF	
End of Table 8-74				

Table 8-75 Programmable Range *n* Registers Reset Values for MPU7

Programmable Range	MPU7 (EMIF16)			
	Start Address (PROGn_MPSAR)	End Address (PROGn_MPEAR)	Memory Page Protection Attribute (PROGn_MPPA)	Memory Protection
PROG0	0x7000_0000	0x70FF_FFFF	0x03FF_FCFF	EMIF16 data
PROG1	0x7100_0000	0x71FF_FFFF	0x03FF_FCFF	
PROG2	0x7200_0000	0x72FF_FFFF	0x03FF_FCFF	
PROG3	0x7300_0000	0x73FF_FFFF	0x03FF_FCFF	
PROG4	0x7400_0000	0x74FF_FFFF	0x03FF_FCFF	
PROG5	0x7500_0000	0x75FF_FFFF	0x03FF_FCFF	
PROG6	0x7600_0000	0x76FF_FFFF	0x03FF_FCFF	
PROG7	0x7700_0000	0x77FF_FFFF	0x03FF_FCFF	
PROG8	0x7800_0000	0x78FF_FFFF	0x03FF_FCFF	
PROG9	0x7900_0000	0x79FF_FFFF	0x03FF_FCFF	
PROG10	0x7A00_0000	0x7AFF_FFFF	0x03FF_FCB6	
PROG11	0x7B00_0000	0x7BFF_FFFF	0x03FF_FCB6	
PROG12	0x7C00_0000	0x7CFF_FFFF	0x03FF_FCB6	
PROG13	0x7D00_0000	0x7DFF_FFFF	0x03FF_FCB6	
PROG14	0x7E00_0000	0x7EFF_FFFF	0x03FF_FCB6	
PROG15	0x7F00_0000	0x7FFF_FFFF	0x03FF_FCB6	
End of Table 8-75				

8.12 DDR3 Memory Controller

The 64-bit DDR3 Memory Controller bus of the TMS320TCI6612 is used to interface to JEDEC standard-compliant DDR3 SDRAM devices. The DDR3 external bus interfaces only to DDR3 SDRAM devices; it does not share the bus with any other types of peripherals.

8.12.1 DDR3 Memory Controller Device-Specific Information

The TMS320TCI6612 includes one 64-bit wide 1.5-V DDR3 SDRAM EMIF interface. The DDR3 interface can operate at 800 mega transfers per second (MTS), 1066 MTS, and 1333 MTS.

Due to the complicated nature of the interface, a limited number of topologies will be 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 SDRAM

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, then the master B read may bypass the master A write and, thus, master B may read stale data and, therefore, receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers with TCCMOD=0) will 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 via software.

If master A does not wait for indication that a write is complete, it must perform the following workaround:

1. Perform the required write.
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.

8.12.2 DDR3 Memory Controller Electrical Data/Timing

The *DDR3 Implementation Guidelines* application report in 2.13 "Related Documentation from Texas Instruments" on page 75 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.

8.13 I^2C Peripheral

The inter-integrated circuit (I^2C) module provides an interface between the SoC and other devices compliant with Philips Semiconductors Inter-IC bus (I^2C bus) specification version 2.1 and connected by way of an I^2C bus. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the SoC through the I^2C module.

8.13.1 I^2C Device-Specific Information

The TMS320TCI6612 device includes an I^2C peripheral module. NOTE: when using the I^2C module, ensure there are external pullup resistors on the SDA and SCL pins.

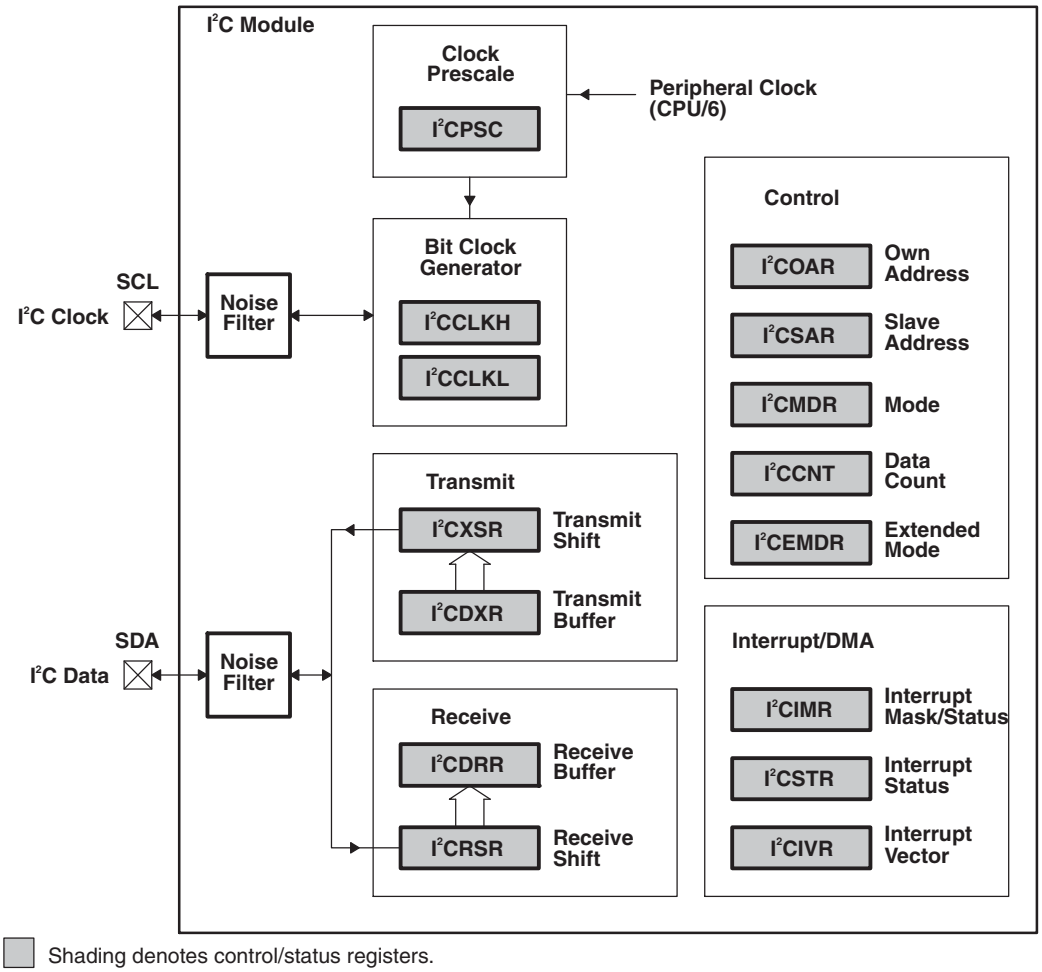
The I^2C modules on the TCI6612 may be used by the SoC to control local peripheral ICs (DACs, ADCs, etc.) or may be used to communicate with other controllers in a system or to implement a user interface.

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 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 8-36 shows a block diagram of the I2C module.

Figure 8-36 I2C Module Block Diagram



8.13.2 I2C Peripheral Register Description(s)

Table 8-76 I2C Registers (Part 1 of 2)

Hex Address Range	Field	Register Name
0253 0000	ICOAR	I2C own address register
0253 0004	ICIMR	I2C interrupt mask/status register
0253 0008	ICSTR	I2C interrupt status register
0253 000C	ICCLKL	I2C clock low-time divider register
0253 0010	ICCLKH	I2C clock high-time divider register
0253 0014	ICCNT	I2C data count register
0253 0018	ICDRR	I2C data receive register
0253 001C	ICSAR	I2C slave address register
0253 0020	ICDXR	I2C data transmit register
0253 0024	ICMDR	I2C mode register
0253 0028	ICIVR	I2C interrupt vector register
0253 002C	ICEMDR	I2C extended mode register
0253 0030	ICPSC	I2C prescaler register

Table 8-76 I²C Registers (Part 2 of 2)

Hex Address Range	Field	Register Name
0253 0034	ICPID1	I²C peripheral identification register 1 [Value: 0x0000 0105]
0253 0038	ICPID2	I²C peripheral identification register 2 [Value: 0x0000 0005]
0253 003C -0253 007F	-	Reserved
End of Table 8-76		

8.13.3 I²C Electrical Data/Timing

8.13.3.1 Inter-Integrated Circuits (I²C) Timing

Table 8-77 I²C Timing Requirements (1)
(see Figure 8-37)

No.			Standard Mode		Fast Mode		Units
			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
End of Table 8-77							

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 8-37 I²C Receive Timings

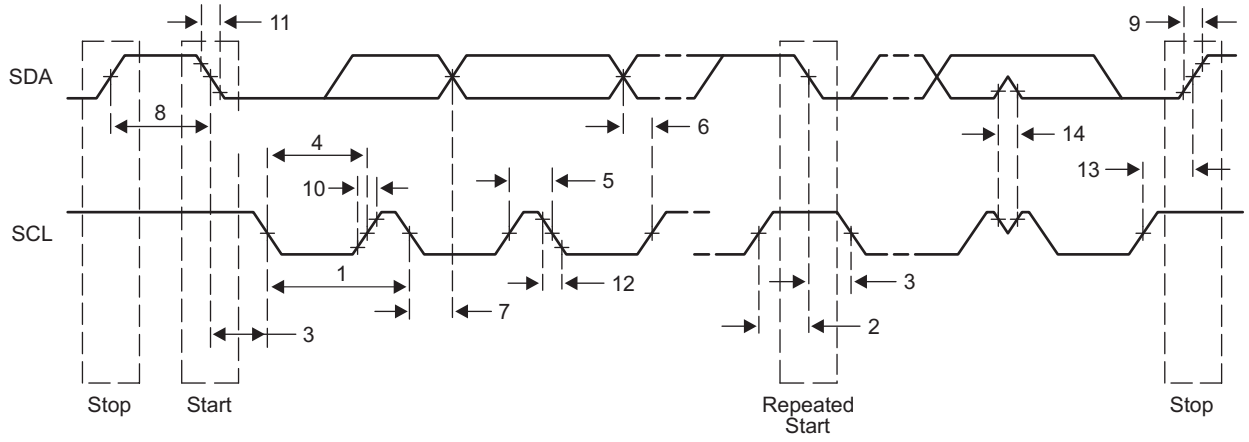


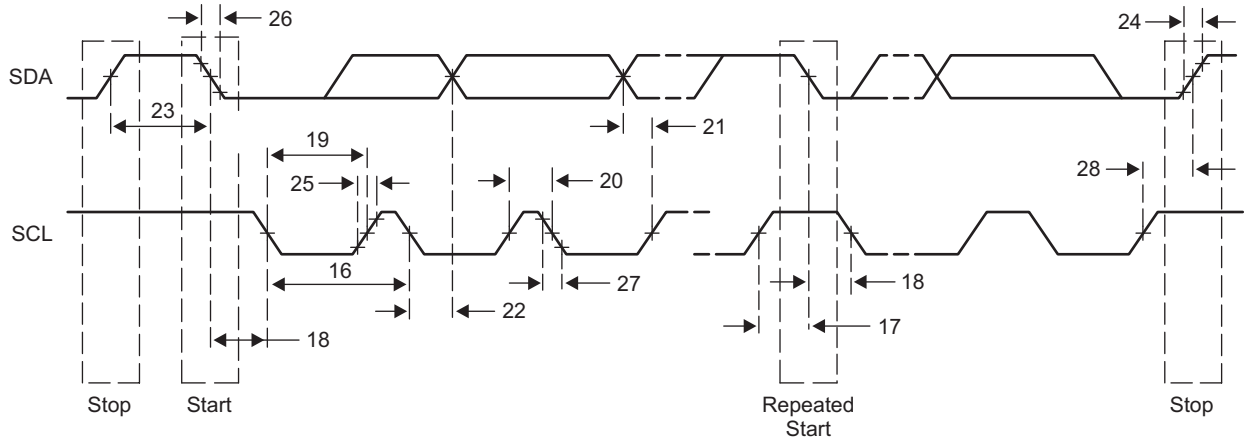
Table 8-78 I²C Switching Characteristics (1)
(see Figure 8-38)

No.	Parameter		Standard Mode Min	Standard Mode Max	Fast Mode Min	Fast Mode Max	Unit
16	$t_{c(SCL)}$	Cycle time, SCL	10		2.5		ms
17	$t_{su(SCLH-SDAL)}$	Setup time, SCL high to SDA low (for a repeated START condition)	4.7		0.6		ms
18	$t_{h(SDAL-SCLL)}$	Hold time, SDA low after SCL low (for a START and a repeated START condition)	4		0.6		ms
19	$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		ms
20	$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		ms
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	ms
23	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		ms
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		ms
	C_p	Capacitance for each I²C pin		10		10	pF

End of Table 8-78

1 C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

Figure 8-38 I²C Transmit Timings



8.14 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 a SPI ROM for boot. The SPI module on TCI6612 is supported only in Master mode. Additional chip-level components can also be included, such as temperature sensors or an I/O expander.

The TCI6612 SPI supports two modes, 3-pin and 4-pin. For the 4-pin chip-select mode the TCI6612 supports up to five chip selects.

8.14.1 SPI Electrical Data/Timing

8.14.1.1 SPI Timing

Table 8-79 SPI Timing Requirements
See Figure 8-39)

No.			Min	Max	Unit
Master Mode Timing Diagrams — Base Timings for 3-Pin Mode					
7	tsu(SDI-SPC)	Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 0	2		ns
7	tsu(SDI-SPC)	Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 1	2		ns
7	tsu(SDI-SPC)	Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 0	2		ns
7	tsu(SDI-SPC)	Input Setup Time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 1	2		ns
8	th(SPC-SDI)	Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 0	5		ns
8	th(SPC-SDI)	Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 1	5		ns
8	th(SPC-SDI)	Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 0	5		ns
8	th(SPC-SDI)	Input Hold Time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 1	5		ns
End of Table 8-79					

Table 8-80 SPI Switching Characteristics (Part 1 of 2)
(See Figure 8-39 and Figure 8-40)

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(SDO-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 0.		5	ns
4	td(SDO-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 1.		5	ns
4	td(SDO-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK Polarity = 1, Phase = 0		5	ns
4	td(SDO-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK Polarity = 1, Phase = 1		5	ns
5	td(SPC-SDO)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 0 Phase = 0		2	ns
5	td(SPC-SDO)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 0 Phase = 1		2	ns
5	td(SPC-SDO)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 1 Phase = 0		2	ns
5	td(SPC-SDO)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 1 Phase = 1		2	ns

Table 8-80 SPI Switching Characteristics (Part 2 of 2)
(See Figure 8-39 and Figure 8-40)

No.	Parameter		Min	Max	Unit
6	toh(SPC-SDO)	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-SDO)	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-SDO)	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-SDO)	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 $\overline{\text{SPISCS[n]}}$ active to first SPICLK. Polarity = 0 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from $\overline{\text{SPISCS[n]}}$ 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 $\overline{\text{SPISCS[n]}}$ active to first SPICLK. Polarity = 1 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from $\overline{\text{SPISCS[n]}}$ 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 $\overline{\text{SPISCS[n]}}$. Polarity = 0 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS[n]}}$. 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 $\overline{\text{SPISCS[n]}}$. Polarity = 1 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting $\overline{\text{SPISCS[n]}}$. Polarity = 1 Phase = 1	0.5*tc + (1*P2) - 5	0.5*tc + (1*P2) + 5	ns
	tw(SCSH)	Minimum inactive time on $\overline{\text{SPISCS[n]}}$ pin between two transfers when $\overline{\text{SPISCS[n]}}$ is not held using the CSHOLD feature.	2*P2 - 5		ns
End of Table 8-80					

1 P2=1/SYSCLK7

Figure 8-39 SPI Master Mode Timing Diagrams — Base Timings for 3-Pin Mode

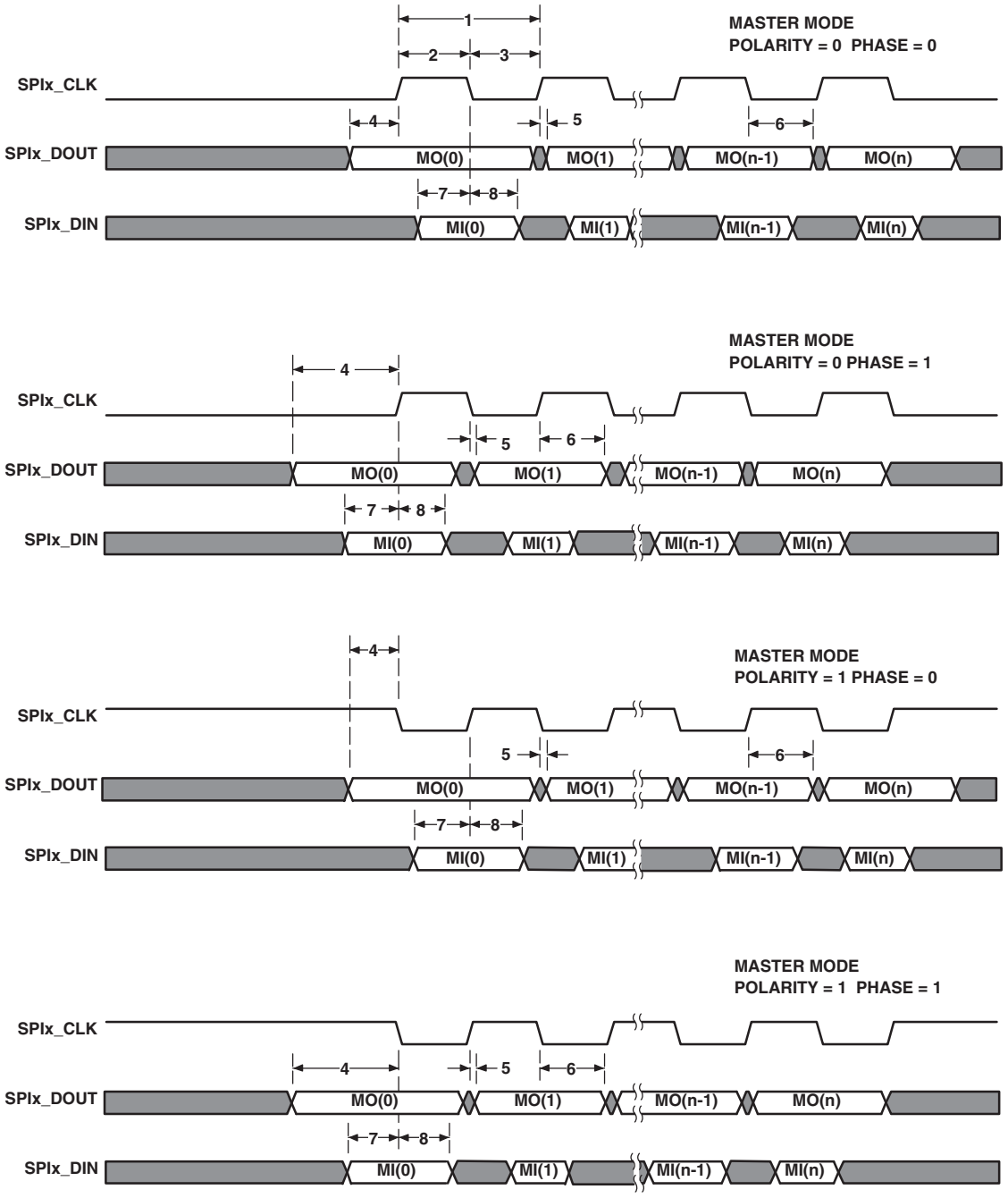
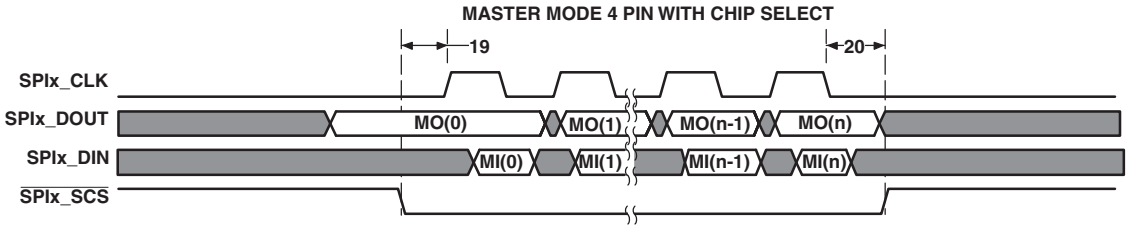


Figure 8-40 SPI Additional Timings for 4-Pin Master Mode with Chip Select Option



8.15 HyperLink Peripheral

The TMS320TCI6612 includes the HyperLink for companion chip/die interfaces. This is a four-lane SerDes interface designed to operate at up to 12.5 Gbps per lane from pin-to-pin. The interface is used to connect with external accelerators that are manufactured using TI libraries. The Hyperbridge links must be connected with DC coupling.

The interface includes the serial station management interfaces used to send power management and flow messages between devices. This consists of four LVCMOS inputs and four LVCMOS outputs configured as two 2-wire output buses and two 2-wire input buses. Each 2-wire bus includes a data signal and a clock signal.

Table 8-81 HyperLink Peripheral Timing Requirements
(see Figure 8-41, Figure 8-42 and Figure 8-43)

No.			Min	Max	Unit
FL Interface					
1	tc(MCMTXFLCLK)	Clock period - MCMTXFLCLK (C1)	5.75		ns
2	tw(MCMTXFLCLKH)	High pulse width - MCMTXFLCLK	0.4*C1	0.6*C1	ns
3	tw(MCMTXFLCLKL)	Low pulse width - MCMTXFLCLK	0.4*C1	0.6*C1	ns
6	tsu(MCMTXFLDAT-MCMTXFLCLKH)	Setup time - MCMTXFLDAT valid before MCMTXFLCLK high	1		ns
7	th(MCMTXFLCLKH-MCMTXFLDAT)	Hold time - MCMTXFLDAT valid after MCMTXFLCLK high	1		ns
6	tsu(MCMTXFLDAT-MCMTXFLCLKL)	Setup time - MCMTXFLDAT valid before MCMTXFLCLK low	1		ns
7	th(MCMTXFLCLKL-MCMTXFLDAT)	Hold time - MCMTXFLDAT valid after MCMTXFLCLK low	1		ns
PM Interface					
1	tc(MCMRXPMCLK)	Clock period - MCMRXPMCLK (C3)	5.75		ns
2	tw(MCMRXPMCLK)	High pulse width - MCMRXPMCLK	0.4*C3	0.6*C3	ns
3	tw(MCMRXPMCLK)	Low pulse width - MCMRXPMCLK	0.4*C3	0.6*C3	ns
6	tsu(MCMRXPMDAT-MCMRXPMCLKH)	Setup time - MCMRXPMDAT valid before MCMRXPMCLK high	1		ns
7	th(MCMRXPMCLKH-MCMRXPMDAT)	Hold time - MCMRXPMDAT valid after MCMRXPMCLK high	1		ns
6	tsu(MCMRXPMDAT-MCMRXPMCLKL)	Setup time - MCMRXPMDAT valid before MCMRXPMCLK low	1		ns
7	th(MCMRXPMCLKL-MCMRXPMDAT)	Hold time - MCMRXPMDAT valid after MCMRXPMCLK low	1		ns
End of Table 8-81					

Table 8-82 HyperLink Peripheral Switching Characteristics (Part 1 of 2)
(see Figure 8-41, Figure 8-42 and Figure 8-43)

No.	Parameter		Min	Max	Unit
FL Interface					
1	tc(MCMRXFLCLK)	Clock period - MCMRXFLCLK (C2)	6.4		ns
2	tw(MCMRXFLCLKH)	High pulse width - MCMRXFLCLK	0.4*C2	0.6*C2	ns
3	tw(MCMRXFLCLKL)	Low pulse width - MCMRXFLCLK	0.4*C2	0.6*C2	ns
4	tosu(MCMRXFLDAT-MCMRXFLCLKH)	Setup time - MCMRXFLDAT valid before MCMRXFLCLK high	1.1		ns
5	toh(MCMRXFLCLKH-MCMRXFLDAT)	Hold time - MCMRXFLDAT valid after MCMRXFLCLK high	1.1		ns
4	tosu(MCMRXFLDAT-MCMRXFLCLKL)	Setup time - MCMRXFLDAT valid before MCMRXFLCLK low	1.1		ns
5	toh(MCMRXFLCLKL-MCMRXFLDAT)	Hold time - MCMRXFLDAT valid after MCMRXFLCLK low	1.1		ns
PM Interface					
1	tc(MCMTXPMCLK)	Clock period - MCMTXPMCLK (C4)	6.4		ns
2	tw(MCMTXPMCLK)	High pulse width - MCMTXPMCLK	0.4*C4	0.6*C4	ns
3	tw(MCMTXPMCLK)	Low pulse width - MCMTXPMCLK	0.4*C4	0.6*C4	ns
4	tosu(MCMTXPMDAT-MCMTXPMCLKH)	Setup time - MCMTXPMDAT valid before MCMTXPMCLK high	1.1		ns

Table 8-82 HyperLink Peripheral Switching Characteristics (Part 2 of 2)
(see Figure 8-41, Figure 8-42 and Figure 8-43)

No.	Parameter		Min	Max	Unit
5	toh(MCMTXPMCLKH-MCMTXPMDAT)	Hold time - MCMTXPMDAT valid after MCMTXPMCLK high	1.1		ns
4	tosu(MCMTXPMDAT-MCMTXPMCLKL)	Setup time - MCMTXPMDAT valid before MCMTXPMCLK low	1.1		ns
5	toh(MCMTXPMCLKL-MCMTXPMDAT)	Hold time - MCMTXPMDAT valid after MCMTXPMCLK low	1.1		ns
End of Table 8-82					

Figure 8-41 HyperLink Station Management Clock Timing

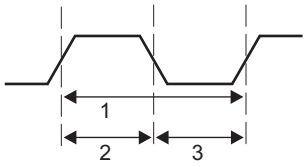


Figure 8-42 HyperLink Station Management Transmit Timing

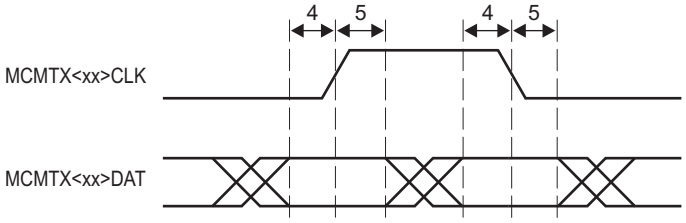
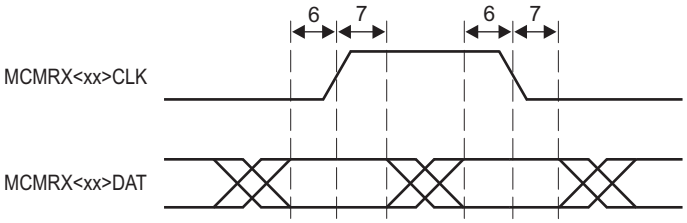


Figure 8-43 HyperLink Station Management Receive Timing



8.16 UART Peripheral

The universal asynchronous receiver/transmitter (UART) module provides an interface between the SoC and UART terminal interface or other UART based peripheral. 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. The SoC can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. The TCI6612 contains two UART modules. For more information on UART, see the *Universal Asynchronous Receiver/Transmitter (UART) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

Table 8-83 UART Timing Requirements
(see Figure 8-44 and Figure 8-45)

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	1.5*(0.96U)	1.5*(1.05U)	ns
6	tw(RXSTOP2)	Pulse width, receive stop bit 2	2*(0.96U)	2*(1.05U)	ns
Autoflow Timing Requirements					
8	td(CTSL-TX)	Delay time, CTS asserted to START bit transmit	P [2]	5 * P	ns
End of Table 8-83					

1 U = UART baud time = 1/programmed baud rate
2 P = 1/SYSCLK7

Figure 8-44 UART Receive Timing Waveform

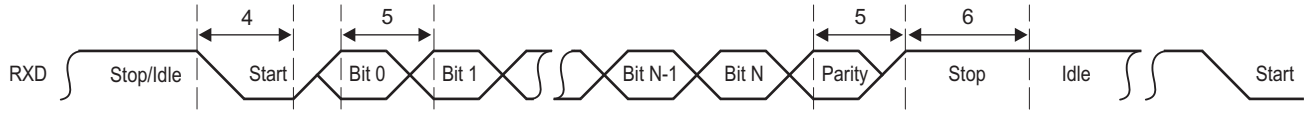


Figure 8-45 UART CTS (Clear-to-Send Input) — Autoflow Timing Waveform

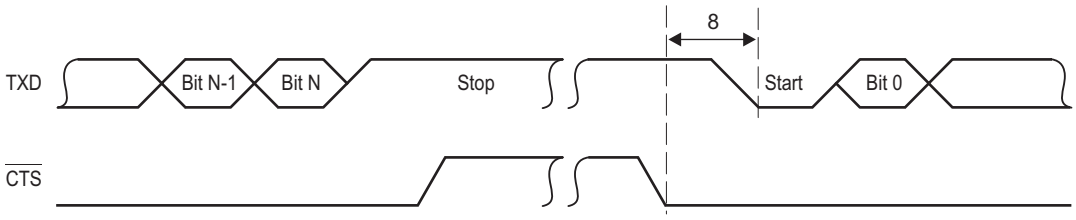


Table 8-84 UART Switching Characteristics
(See Figure 8-46 and Figure 8-47)

No.	Parameter		Min	Max	Unit
Transmit Timing					
1	tw(TXSTART)	Pulse width, transmit start bit	U - 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 (1)	5 * P	ns
End of Table 8-84					

1 P = CPU/6

Figure 8-46 UART Transmit Timing Waveform

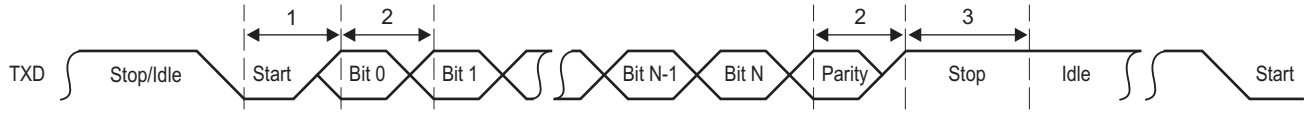
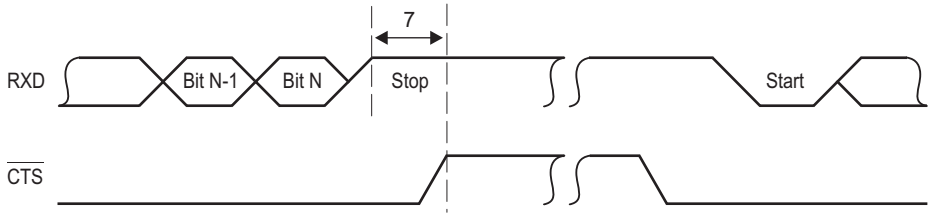


Figure 8-47 UART RTS (Request-to-Send Output) – Autoflow Timing Waveform



8.17 PCIe Peripheral

The two-lane PCI express (PCIe) module on TMS320TCI6612 provides an interface between the SoC and other PCIe-compliant devices. The PCI Express module provides low pin count, high reliability, and high-speed data transfer at rates of 5.0 Gbps per lane on the serial links. For more information, see the *Peripheral Component Interconnect Express (PCIe) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.18 Packet Accelerator

The Packet Accelerator provides L2 to L4 classification functionalities. It supports classification for Ethernet, VLAN, MPLS over Ethernet, IPv4/6, GRE over IP, and other session identification over IP such as TCP and UDP ports. It maintains 8K multiple-in, multiple-out hardware queues. It also provides checksum capability as well as some QoS capabilities. It enables a single IP address to be used for a multi-core device. It can process up to 1.5 M pps. For more information, see the *Packet Accelerator (PA) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.19 Security Accelerator

The Security Accelerator provides wire-speed processing on 1-Gbps Ethernet traffic on IPSec, SRTP, and 3GPP Air interface security protocols. It functions on the packet level with the packet and the associated security context being one of these above three types. The security accelerator is coupled with packet accelerator, 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 *Security Accelerator (SA) for KeyStone Devices User Guide* in section 2.13 "Related Documentation from Texas Instruments" on page 75

8.20 Gigabit Ethernet (GbE) Switch Subsystem

The Gigabit Ethernet (GbE) Switch subsystem modules provide an efficient interface between the TMS320TCI6612 SoC and the networked community. The GbE Switch subsystem supports 10Base-T (10 Mbits/second [Mbps]), 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. For more information, see the *Gigabit Ethernet (GbE) Switch for KeyStone Devices* User Guide in 2.13 "Related Documentation from Texas Instruments" on page 75.

Each device has a unique MAC address. There are two registers to hold these values, MACID1 (0x02620110) and MACID2 (0x02600114). All bits of these registers are defined as follows:

Figure 8-48 MACID1 Register

31	0
MACID[31:0]	
R,+xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	

Legend: R = Read only; -x, value is indeterminate

Table 8-85 MACID1 Register Field Descriptions

Bit	Field	Description
31-0	MAC ID[31-0]	MAC ID. A range will be assigned to this device. Each device will consume only one MAC address.
End of Table 8-85		

Figure 8-49 MACID2 Register

31 24	23 18	17	16	15 0
CRC	Reserved	FLOW	BCAST	MACID[47:32]
R+,cccc cccc	R,+rr rrrr	R,+z	R,+y	R,+xxxx xxxx xxxx xxxx

Legend: R = Read only; -x, value is indeterminate

Table 8-86 MACID2 Register Field Descriptions (Part 1 of 2)

Bit	Field	Description
31-24	Reserved	Variable
23-18	Reserved	000000
17	FLOW	MAC Flow Control 0 = Off 1 = On

Table 8-86 MACID2 Register Field Descriptions (Part 2 of 2)

Bit	Field	Description
16	BCAST	Default m/b-cast reception 0 = Broadcast 1 = Disabled
15-0	MAC ID[47-0]	MAC ID. A range will be assigned to this device. Each device will consume only one MAC address.
End of Table 8-86		

There is a time synchronization (CPTS) submodule in the Ethernet switch module for time synchronization. Programming this register selects the clock source for the CPTS_RCLK. See the *Gigabit Ethernet (GbE) Switch Subsystem for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75 for the register address and other details about the time synchronization module. The register CPTS_RFTCLK_SEL for reference clock selection of the time synchronization submodule is shown in Figure 8-50.

Figure 8-50 CPTS_RFTCLK_SEL Register

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

Legend: R = Read only; -x, value is indeterminate

Table 8-87 CPTS_RFTCLK_SEL Register Field Descriptions

Bit	Field	Description
31-3	Reserved	Reserved. Read as 0.
2-0	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. 000 = SYSCLK3 001 = SYSCLK4 010 = TIMI0 011 = TIMI1 1xx = Reserved
End of Table 8-87		

8.21 Management Data Input/Output (MDIO)

The management data input/output (MDIO) module implements the 802.3 serial management interface to interrogate and controls 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 GbE Switch subsystem, retrieve the negotiation results, and configure required parameters in the GbE Switch Subsystem module for correct operation. The module is designed to allow almost transparent operation of the MDIO interface, with very little maintenance from the core processor. For more information, see the *Gigabit Ethernet (GbE) Switch for KeyStone Devices* User Guide in section 2.13 "Related Documentation from Texas Instruments" on page 75.

Table 8-88 MDIO Timing Requirements
(see Figure 8-51)

No.			Min	Max	Unit
1	tc(MDCLK)	Cycle time, MDCLK	400		ns
	tw(MDCLKH)	Pulse duration, MDCLK high	180		ns
	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	0		ns
	tt(MDCLK)	Transition time, MDCLK		5	ns
End of Table 8-88					

Figure 8-51 MDIO Input Timing

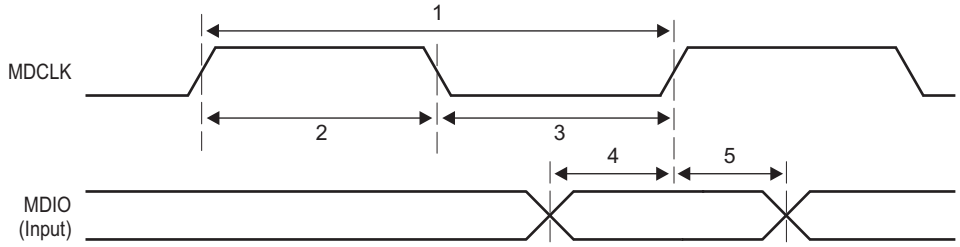
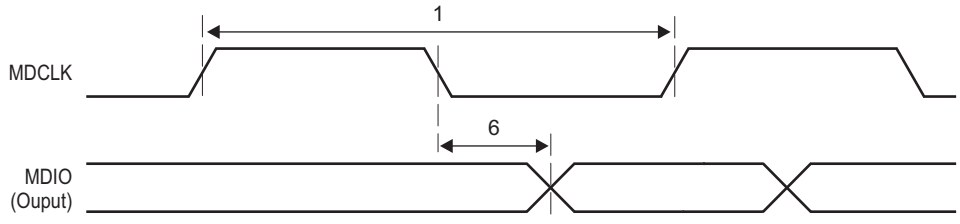


Table 8-89 MDIO Switching Characteristics
(see Figure 8-52)

No.	Parameter		Min	Max	Unit
7	td(MDCLKL-MDIO)	Delay time, MDCLK low to MDIO data output valid		100	ns
End of Table 8-89					

Figure 8-52 MDIO Output Timing



8.22 Timers

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

8.22.1 Timers Device-Specific Information

The TMS320TCI6612 device has ten 64-bit timers in total. Timer0 through Timer1 are dedicated to each of the two CorePacs as a watchdog timer and can also be used as general-purpose timers. Timer8 is dedicated to the ARM as a watchdog timer and cannot be configured as a general-purpose timer. Each of other seven timers can be configured as a general-purpose timer only, with each timer programmed as a 64-bit timer or as two separate 32-bit timers.

Timer 0, 1 and 8 also go through the resetmux block. The NMI event from resetmux for timer 8 is connected to ARM as an interrupt event. The local reset timer event generated by the resetmux for timer 8 is used to trigger device reset, because the TCI6612 does not support local reset to ARM. See the interrupt sections for the timer event connectivity.

In addition, timer 0 and 1 can run only in the 64-bit mode. Each of the rest of the timers can be configured to run as two 32-bit timers or as one 64-bit timer.

When operating in 64-bit mode, the timer counts either VBUS 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 "Reset Type Status Register (RSTYPE)" on page 153 and the type of reset initiated can set by programming "Reset Configuration Register (RSTCFG)" on page 154. For more information, see the *64-bit Timer (Timer 64) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.22.2 Timers Electrical Data/Timing

Table 8-90, Table 8-91, and Figure 8-53 show the timing requirements and switching characteristics of the Timer peripherals.

Table 8-90 Timer Input Timing Requirements [1]
(see Figure 8-53)

No.			Min	Max	Unit
1	$t_{w(TINPH)}$	Pulse duration, high	12C		ns
2	$t_{w(TINPL)}$	Pulse duration, low	12C		ns
End of Table 8-90					

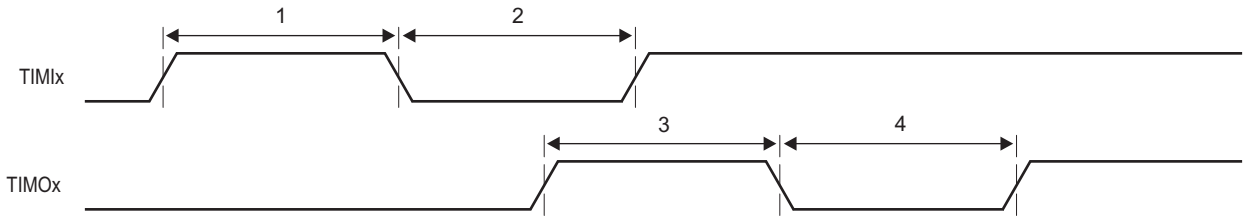
1 If CORECLKSEL = 0, C = 1/CORECLK(NIP) frequency in ns. If CORECLKSEL = 1, C = 1/ALTCORECLK frequency in ns.

Table 8-91 Timer Output Switching Characteristics [1] [2]
(see Figure 8-53)

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
End of Table 8-91					

1 Over recommended operating conditions.
2 If CORECLKSEL = 0, C = 1/CORECLK(NIP) frequency in ns. If CORECLKSEL = 1, C = 1/ALTCORECLK frequency in ns.

Figure 8-53 Timer Timing



8.23 Rake Search Accelerator (RSA)

There are four rake search accelerators (RSAs) on the TCI6612 device. CorePac0 and CorePac1 each have one set of directly-connected RSA pairs. The RSA is an extension of the C66x CPU. The CPU performs send/receive to the RSAs via the .L and .S functional units. For more information, see the *Rake Search Accelerator (RSA) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.24 Enhanced Viterbi-Decoder Coprocessor (VCP2)

The TMS320TCI6612 device has two high-performance embedded Viterbi decoder coprocessors (VCP2) that significantly speed up channel-decoding operations on-chip. Each VCP2, operating at CPU clock divided-by-3, can decode more than 694 7.95-Kbps adaptive multi-rate (AMR) [K = 9, R = 1/3] voice channels. The VCP2 supports constraint lengths K = 5, 6, 7, 8, and 9, rates R = 3/4, 1/2, 1/3, 1/4, and 1/5, and flexible polynomials, while generating hard decisions or soft decisions. Communications between the VCP2 and the CPU are carried out through the EDMA3 controller. The VCP2 supports:

- Unlimited frame sizes
- Code rates 3/4, 1/2, 1/3, 1/4, and 1/5
- Constraint lengths 5, 6, 7, 8, and 9
- Programmable encoder polynomials
- Programmable reliability and convergence lengths
- Hard and soft decoded decisions
- Tail and convergent modes
- Yamamoto logic
- Tail biting logic
- Various input and output FIFO lengths

For more information, see the *Viterbi Coprocessor (VCP2) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.25 Third-Generation Turbo Decoder Coprocessor (TCP3d)

The device has two high-performance embedded turbo-decoder coprocessors (TCP3d) that significantly speed up channel-decoding operations on-chip for WCDMA, HSPA, HSPA+, TD-SCDMA, LTE, and WiMAX. Operating at CPU clock divided-by-2, the TCP3d is capable of processing data channels at a throughput of > 100 Mbps. For more information, see the *Turbo Decoder Coprocessor 3 (TCP3d) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.26 Bit Rate Coprocessor (BCP)

The BCP is a hardware accelerator for wireless infrastructure. It performs most of the uplink and downlink layer 1 bit processing for 3G and 4G wireless standards. It supports LTE, FDD WCDMA, TD-SCDMA, and WiMAX 802.16-2009 standards. It supports various downlink processing blocks like CRC attachment, turbo encoding, rate matching, code block concatenation, scrambling, and modulation. It supports various uplink processing blocks like soft slicer, de-scrambler, de-concatenation, rate de-matching and LLR combining. For more information, see the *Bit Rate Coprocessor for keyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.27 Serial RapidIO (SRIO) Port

The SRIO port on the TMS320TCI6612 device is a high-performance, low pin-count interconnect aimed for embedded markets. The use of the RapidIO interconnect in a baseband board design can create a homogeneous interconnect environment, providing even more connectivity and control among the components. RapidIO is based on the memory and device addressing concepts of processor buses where the transaction processing is managed completely by hardware. This enables the RapidIO interconnect to lower the system cost by providing lower latency, reduced overhead of packet data processing, and higher system bandwidth, all of which are key for wireless interfaces. For more information, see the *Serial RapidIO (SRIO) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.28 General-Purpose Input/Output (GPIO)

8.28.1 GPIO Device-Specific Information

On the TMS320TCI6612, the GPIO peripheral pins GP[31:0] are also used to latch configuration pins. For more detailed information on device/peripheral configuration and the TCI6612 device pin muxing, see "Device Configuration" on page 76.

8.28.2 GPIO Electrical Data/Timing

Table 8-92 GPIO Input Timing Requirements (1)
(see Figure 8-54)

No.			Min	Max	Unit
1	$t_{w(GPOH)}$	Pulse duration, GPOx high	12C		ns
2	$t_{w(GPOL)}$	Pulse duration, GPOx low	12C		ns
End of Table 8-92					

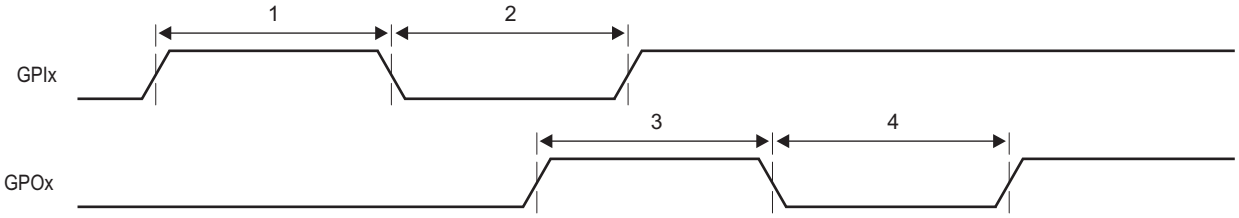
1 If CORECLKSEL = 0, C = 1 ÷ CORECLK(NIP) frequency, in ns. If CORECLKSEL = 1, C = 1 ÷ ALTCORECLK frequency, in ns.

Table 8-93 GPIO Output Switching Characteristics (1) (2)
(see Figure 8-54)

No.	Parameter		Min	Max	Unit
1	$t_{w(GPOH)}$	Pulse duration, GPOx high	36C - 8		ns
2	$t_{w(GPOL)}$	Pulse duration, GPOx low	36C - 8		ns
End of Table 8-93					

1 Over recommended operating conditions.
2 If CORECLKSEL = 0, C = 1 ÷ CORECLK(NIP) frequency, in ns. If CORECLKSEL = 1, C = 1 ÷ ALTCORECLK frequency, in ns.

Figure 8-54 GPIO Timing



8.29 Semaphore2

The device contains an enhanced semaphore module for the management of shared resources of the SoC cores. The Semaphore enforces atomic accesses to shared chip-level resources so that the read-modify-write sequence is not broken. The semaphore block has unique interrupts to each of the cores to identify when that core 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 3 masters and contains 32 semaphores to be used within the system.

The Semaphore module is accessible only by masters of C66x CorePacs (with privID 0 to 1) and ARM CorePac (with privID 7) or the EDMA transactions initiated by C66x CorePacs and ARM CorePac.

If the remote device needs to access the Semaphore module, the HyperLink configuration register must be appropriately configured so the remote device can send transactions with the desired privID value to the local Semaphore module. For more information on HyperLink configuration, see the *HyperLink for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

There are two methods of accessing a semaphore resource:

- **Direct Access:** A core directly accesses a semaphore resource. If free, the semaphore will be granted. If not, the semaphore is not granted.
- **Indirect Access:** A core indirectly accesses a semaphore resource by writing it. Once it is free, an interrupt notifies the CPU that it is available.

8.30 Antenna Interface Subsystem 2 (AIF2)

The enhanced antenna interface subsystem (AIF2) consists of the antenna interface module and two SerDes macros. The AIF2 relies on the performance SerDes macro (high-speed serial link) with a logic layer that supports the CPRI protocol. The AIF is used to connect to the backplane for transmission and reception of antenna data, as well as to connect to additional device peripherals.

The AIF2 has 11 timer synchronization events from the AIF2 Timer (AT) module. Timer synchronization events 0-7 are routed as primary events to the EDMA3CC1 and also as secondary events to the C66x CorePacs via CIC0. Timer synchronization events 8 and 9 are hard-wired to TAC and RAC, respectively.

Table 8-94 AIF2 Timer Module Timing Requirements
See Figure 8-55, Figure 8-56, Figure 8-57, and Figure 8-58

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
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	13.02		ns
11	tc(PHYSYNC)	Cycle time, PHYSYNC pulse to PHYSYNC pulse	10.00		ms
12	tw(RADSYNCH)	Pulse duration, RADSYNC high	13.02		ns
13	tc(RADSYNC)	Cycle time, RADSYNC pulse to RADSYNC pulse	10.00		ms
End of Table 8-94					

1 C1 = tc(RP1CLKN/P)

Figure 8-55 AIF2 RP1 Frame Synchronization Clock Timing

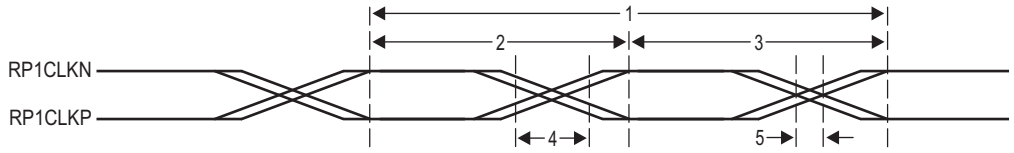


Figure 8-56 AIF2 RP1 Frame Synchronization Burst Timing

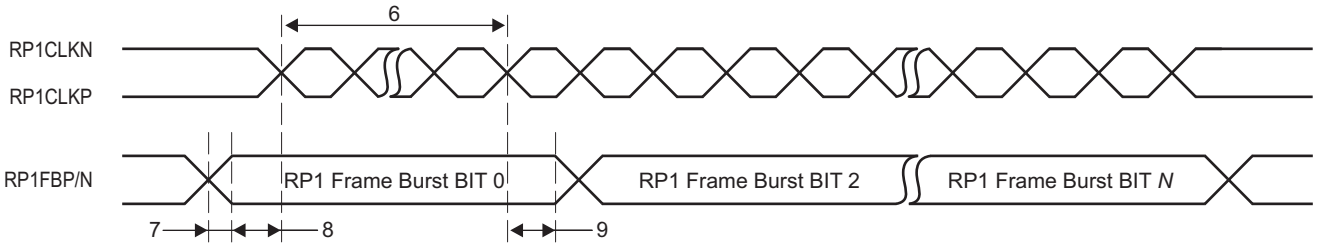


Figure 8-57 AIF2 Physical Layer Synchronization Pulse Timing

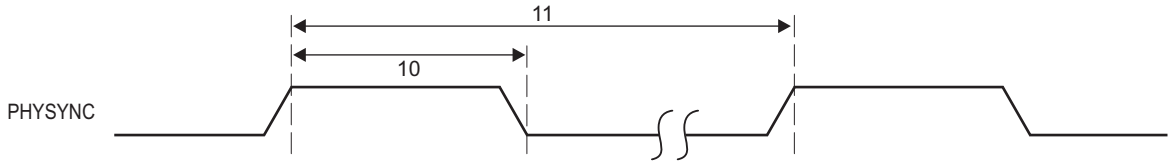


Figure 8-58 AIF2 Radio Synchronization Pulse Timing

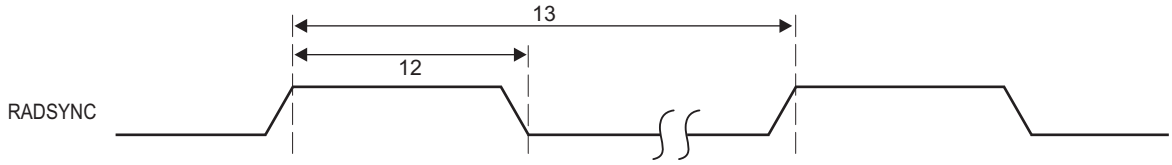
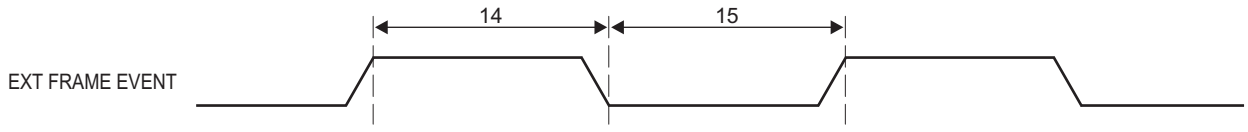


Table 8-95 AIF2 Timer Module Switching Characteristics
(see Figure 8-59)

No.	Parameter		Min	Max	Unit
External Frame Event					
14	tw(EXTFRAMEEVENTH)	Pulse width, EXTFRAMEEVENT output high	4 * C1 (1)		ns
15	tw(EXTFRAMEEVENTL)	Pulse width, EXTFRAMEEVENT output low	4 * C1		ns
End of Table 8-95					

1 C1 = tc(RP1CLKN/P)

Figure 8-59 AIF2 Timer External Frame Event Timing



8.31 Receive Accelerator Coprocessor (RAC)

The TMS320TCI6612 has a receive accelerator coprocessor (RAC) subsystem. The RAC subsystem is a receive chip rate accelerator based on a generic correlator coprocessor (GCCP). It supports UMTS operations; assists in transferring data received from the antenna data to the receive core and performs receive functions targets at W-CDMA macro bits.

The RAC subsystem consists of several components:

- Two GCCP accelerators for finger despread (FD), path monitor (PM), preamble detection (PD), and stream power estimator (SPE).
- Back-end interface (BEI) for management of the RAC configuration and the data output.
- Front-end interface (FEI) for reception of the antenna data for processing and access to all memory mapped registers (MMRs) and memories in the RAC components.

The RAC has a total of three ports connected to the DMA TeraNet:

- BEI includes two master connections to the DMA TeraNet for output data to device memory. One is 128-bit and the other is 64-bit, both are clocked at the same rate as the DMA crossbar.
- The FEI has a slave connection to the DMA TeraNet for input data as well as direct memory access (to facilitate debug).

8.32 Transmit Accelerator Coprocessor (TAC)

The transmit accelerator coprocessor (TAC) subsystem is a transmit chip-rate accelerator intended to support UMTS applications. For more information, see the *Transmit Accelerator (TAC) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.33 Fast Fourier Transform Coprocessor (FFTC)

There are two fast Fourier transform coprocessors (FFTC) intended to accelerate FFT, IFFT, DFT, and IDFT operations. For more information, see the *Fast Fourier Transform Coprocessor (FFTC) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.34 Universal Subscriber Identity Module (USIM)

The TMS320TCI6612 is equipped with a Universal Subscriber Identity Module (USIM) for user authentication. The TCI6612 USIM supports the following features for compliance with ISO, ETSI/GSM, 3GPP standards:

- General:
  - Voltage supply: Class C (Vcc = 1.8 V) mandatory
  - External clock frequency between 1 MHz and 5 MHz during ATR sequence
  - Mandatory transmission factor values: Di/Fi = 1/372, 8/512, and 16/512
  - Start-bit detection logic activated at 11 ETU (T = 0) or at 10 ETU (T = 1)
    (Feature strengthened by comparison to ISO: min 12 ETU for T = 0)
  - Over-sampling frequency in reception equal to FETU × 8

- Answer to reset (ATR) procedure for power-up status and optimization
  - Hardware error handling of character parity check
  - Hardware error handling of ATR time-out
  - Hardware identification of character coding convention (direct/inverse)

- T = 0 and T = 1 transmission protocol support
  - T = 0 protocol
    - Hardware error handling of WWT timer time-out
    - Delayed release of I/O data line after successful completion of last byte transmission
  - T = 1 protocol
    - Transmit protocol (T) of type 1 (TD1 character equal to xH01) (asynchronous half-duplex block transmission protocol)
    - Character waiting time (CWT) management (hardware-timer based)
    - Block waiting time (BWT) management (hardware-timer based)
    - Block guard time (BGT) management (hardware-timer based)
    - Need to keep parity checking at character level without acknowledgement (i.e. no repeat)
- Provide an external clock (FSCK) between 1 MHz and 5 MHz
- ATR procedure
  - Hardware identification of character coding convention (direct/inverse) from TS character with update for processing of subsequent characters (T0, Tai...etc)
  - Software processing of PPS/PTS procedure
  - Software identification of procedure bytes of PTS0 and PTS1 with update of protocol type T and transmission factor value Fi/Di T = 1 protocol
  - Hardware management of block length (interpretation of LEN character)
  - Hardware checking of EDC error code, if LRC is used.
  - Hardware error handling of:
    - LEN block length error (in Rx mode)
    - Character parity check in complement to EDC error.
    - CWT, BWT and BGT timers time-out (with IT generation)

Note—Note—If CRC used, EDC error code checking should be done by software. Handling of PCB erroneous encoding should be done by software.

8.35 EMIF16 Peripheral

The EMIF16 module provides an interface between SoC and external memories such as NAND and NOR flash. For more information, see the *External Memory Interface (EMIF16) for KeyStone Devices User Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.35.1 EMIF16 Electrical Data/Timing

Table 8-96 EMIF16 Asynchronous Memory Timing Requirements [1] (Part 1 of 2)
(see Figure 8-60 and Figure 8-61)

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

Table 8-96 EMIF16 Asynchronous Memory Timing Requirements [1] (Part 2 of 2)
(see Figure 8-60 and Figure 8-61)

No.			Min	Max	Unit
3	t_C(CEL)	EMIF read cycle time when ew =1, meaning extended wait mode enabled	(RS+RST+RH+3) *E-3	(RS+RST+RH+3) *E+3	ns
4	t_{osu}(CEL-OEL)	Output setup time from CE low to OE low. SS = 0, not in select strobe mode	(RS+1) * E - 3	(RS+1) * E + 3	ns
5	t_{oh}(OEH-CEH)	Output hold time from OE high to CE high. SS = 0, not in select strobe mode	(RH+1) * E - 3	(RH+1) * E + 3	ns
4	t_{osu}(CEL-OEL)	Output setup time from CE low to OE low in select strobe mode, SS = 1	(RS+1) * E - 3	(RS+1) * E + 3	ns
5	t_{oh}(OEH-CEH)	Output hold time from OE high to CE high in select strobe mode, SS = 1	(RH+1) * E - 3	(RH+1) * E + 3	ns
6	t_{osu}(BAV-OEL)	Output setup time from BA valid to OE low	(RS+1) * E - 3	(RS+1) * E + 3	ns
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
End of Table 8-96					

1 E = 1/SYSCLK7

Figure 8-60 EMIF16 Asynchronous Memory Read Timing Diagram

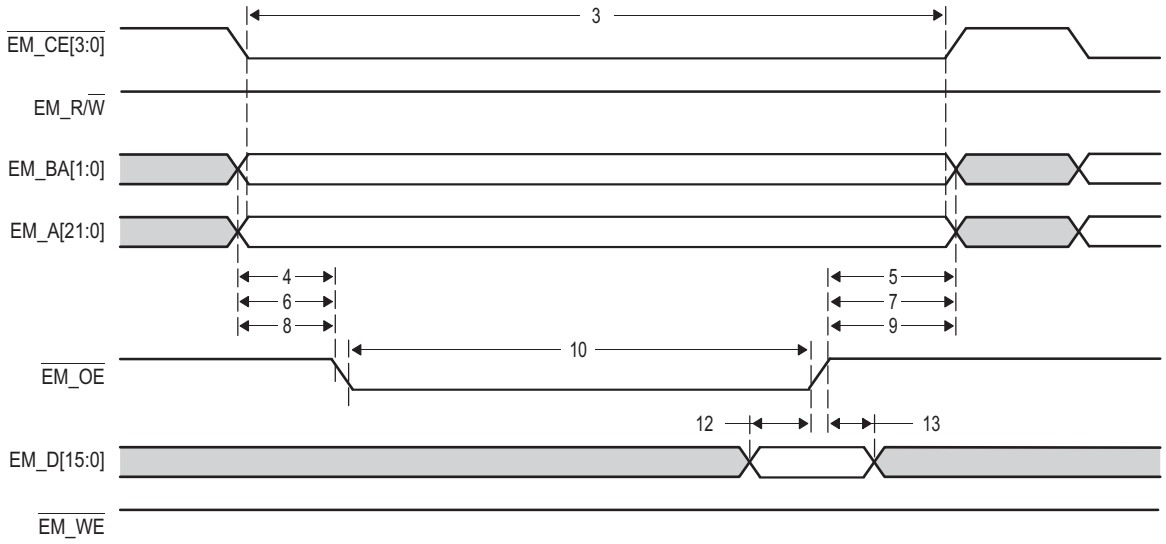


Figure 8-61 EMIF16 Asynchronous Memory Write Timing Diagram

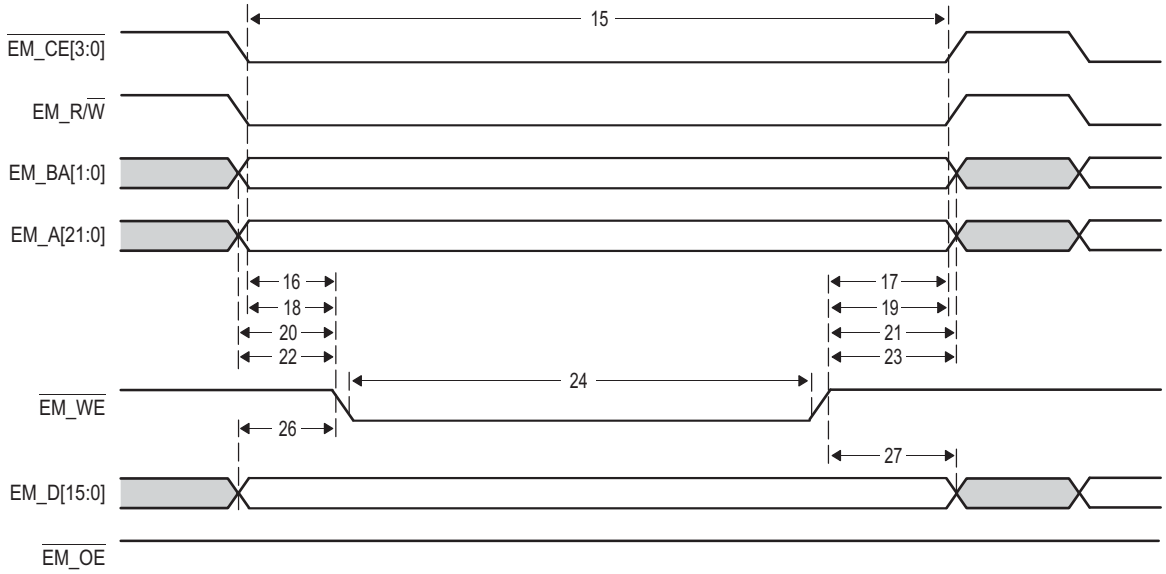


Figure 8-62 EMIF16 EM_WAIT Read Timing Diagram

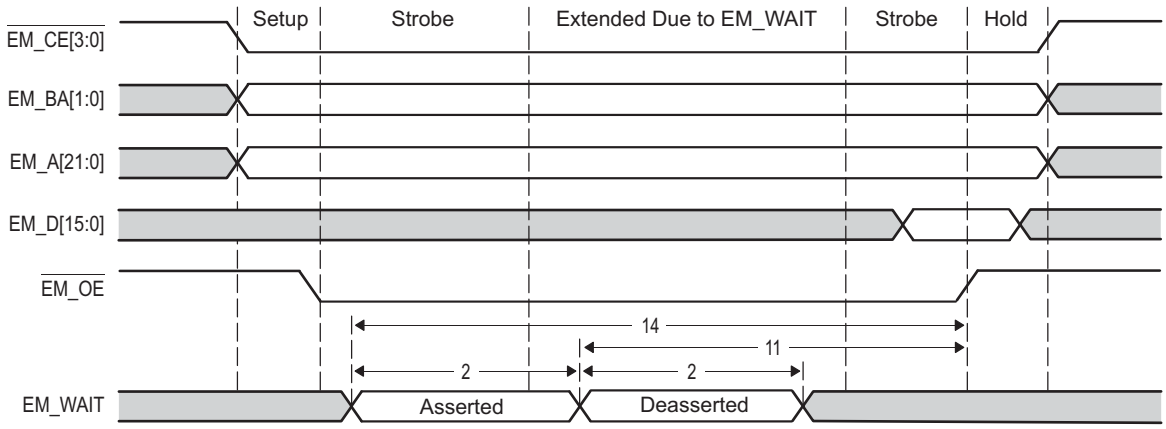
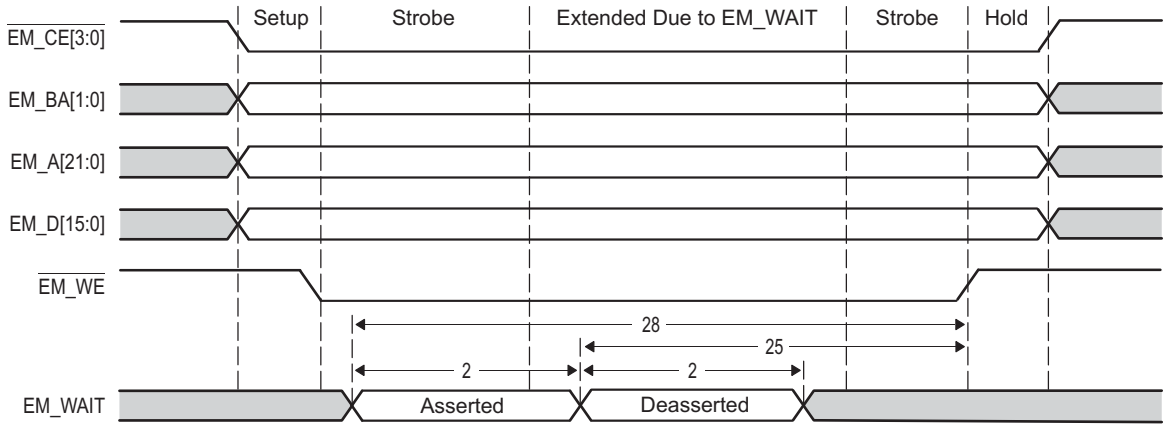


Figure 8-63 EMIF16 EM_WAIT Write Timing Diagram



8.36 Emulation Features and Capability

8.36.1 Advanced Event Triggering (AET)

The TMS320TCI6612 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 in 2.13 "Related Documentation from Texas Instruments" on page 75:

- *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

8.36.2 Trace

The TCI6612 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.

The TCI6612 supports the following 3 levels of trace:

- DSP trace for each individual CorePac (AET is a part of DSP trace)
- CoreSight trace for ARM A8 Core
- System trace

In addition the TCI6612 has the capability of storing trace data on-chip to embedded trace buffers (ETBs) for all levels of trace, or exporting data to an external trace receiver through the supported JTAG interface. For more information on trace, see the *Keystone Embedded Trace User's Guide* in 2.13 "Related Documentation from Texas Instruments" on page 75.

For more information on board design guidelines for trace advanced emulation, see the *Emulation and Trace Headers Technical Reference* in 2.13 "Related Documentation from Texas Instruments" on page 75.

8.36.2.1 Trace Electrical Data/Timing

Table 8-97 DSP Trace Switching Characteristics (1)
(see Figure 8-64)

No.	Parameter		Min	Max	Unit
1	t_w(DPnH)	Pulse duration, DPn/EMUn high detected at 50% Voh	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 detected at 50% Voh	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
End of Table 8-97					

1 Over recommended operating conditions.

Table 8-98 STM Trace Switching Characteristics (1)
(see Figure 8-64)

No.	Parameter		Min	Max	Unit
1	t_w(DPnH)	Pulse duration, DPn/EMUn high detected at 50% Voh with 60/40 duty cycle	5-1		ns
1	t_w(DPnH)90%	Pulse duration, DPn/EMUn high detected at 90% Voh	3.5		ns
2	t_w(DPnL)	Pulse duration, DPn/EMUn low detected at 50% Voh with 60/40 duty cycle	5-1		ns
2	t_w(DPnL)10%	Pulse duration, DPn/EMUn low detected at 10% Voh	3.5		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.		1	ns
	t_{sldp_o}(DPn)	Output slew rate DPn/EMUn	3.3		V/ns
End of Table 8-98					

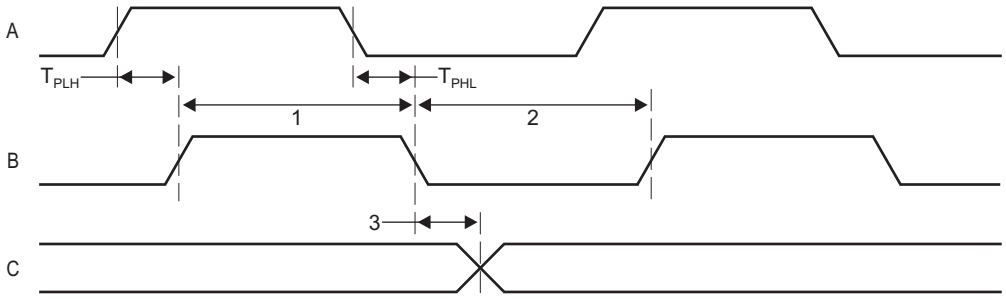
1 Over recommended operating conditions.

Table 8-99 CoreSight Trace Switching Characteristics (1)
(see Figure 8-64)

No.	Parameter		Min	Max	Unit
1	t_w(DPnH)	Pulse duration, DPn/EMUn high detected at 50% Voh with 60/40 duty cycle	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 detected at 50% Voh with 60/40 duty cycle	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
End of Table 8-99					

1 Over recommended operating conditions.

Figure 8-64 Trace Timing



8.36.3 IEEE 1149.1 JTAG

The JTAG interface is used to support boundary scan and emulation of the device. The boundary scan supported allows for an asynchronous TRST and only the 5 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 (SRIO and 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).

8.36.3.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the TCI6612 SoC includes an internal pulldown (IPD) on the TRST pin to ensure that TRST will always be asserted upon power up and the SoC's internal emulation logic will always be properly initialized when this pin is not routed out. JTAG controllers from Texas Instruments actively drive TRST high. However, some third-party JTAG controllers may not drive TRST high but expect the use of an external pullup resistor on TRST. When using this type of JTAG controller, assert TRST to initialize the SoC after powerup and externally drive TRST high before attempting any emulation or boundary scan operations.

8.36.3.2 JTAG Electrical Data/Timing

Table 8-100 JTAG Test Port Timing Requirements
(see Figure 8-65)

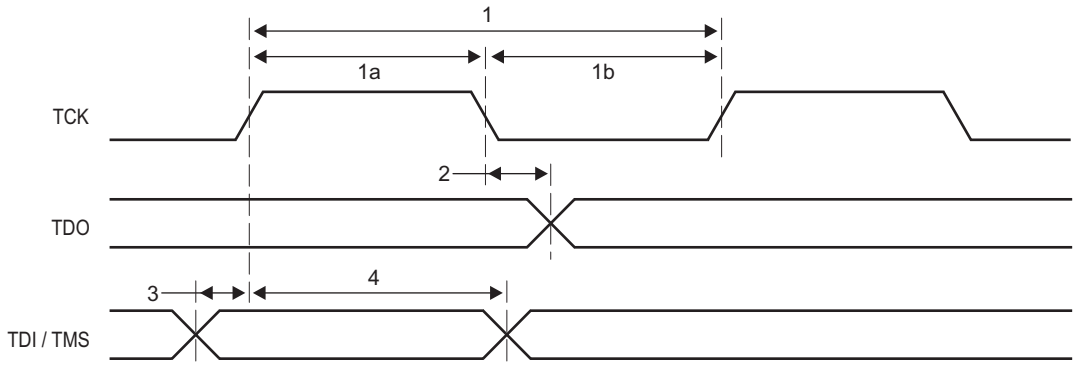
No.			Min	Max	Unit
1	$t_{c(TCK)}$	Cycle time, TCK	28		ns
1a	tw(TCKH)	Pulse duration, TCK high (40% of tc)	11.2		ns
1b	tw(TCKL)	Pulse duration, TCK low(40% of tc)	11.2		ns
3	tsu(TDI-TCK)	Input setup time, TDI valid to TCK high	2.8		ns
3	tsu(TMS-TCK)	Input setup time, TMS valid to TCK high	2.8		ns
4	th(TCK-TDI)	Input hold time, TDI valid from TCK high	14		ns
4	th(TCK-TMS)	Input hold time, TMS valid from TCK high	14		ns
End of Table 8-100					

Table 8-101 JTAG Test Port Switching Characteristics (1)
(see Figure 8-65)

No.	Parameter		Min	Max	Unit
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid		11.2	ns
End of Table 8-101					

1 Over recommended operating conditions.

Figure 8-65 JTAG Test-Port Timing



A Revision History

Revision D

Revised Main PLL diagram to show 0 input to PLL Controller coming from output of PLL input MUX (Page 146)
Updated vusr to HyperLink in interrupt event tables (Page 182)
Updated UART_A to be UART_0 and UART_B to be UART_1 in Memory Map Summary table (Page 25)
Updated VCP[0-3] to VCP[A-D] in CIC0 Event Input table (Page 183)
Updated the description of the escalated priority field in ARM Priority register (Page 108)
Renamed EFUSE to OTP Memory in Memory Map Summary table (Page 25)
Changed Timer number from 12 to 10 for 6612 device (Page 13)
Added MPU Registers Reset Values section (Page 224)
Added Initial Startup row for CVDD in Recommended Operating Conditions table (Page 124)
Updated all SerDes clocks to discrete frequencies in the Clock Input Timing Requirements table (Page 157)
Corrected tj(ALTCORECLKN/P) max value from 100 to 0.02*tc(ALTCORECLKN/P) (Page 157)
Corrected tj(DDRCLKN/P) max value from 0.025*tc(DDRCLKN/P) to 0.02*tc(DDRCLKN/P) (Page 162)
Corrected tj(PASSCLKN/P) max value from 100 to 0.02*tc(PASSCLKN/P) (Page 165)
Corrected tj(SYSCLKN/P) max value from 100 to 0.02*tc(SYSCLKN/P) (Page 157)
Updated the descriptions of how Semaphore module is accessible (Page 247)
Added HOUT timing diagram in Host Interrupt Output section (Page 208)
Updated PLL clock formula in PLL Settings section (Page 40)
Added note to DDR3 PLL initialization sequence (Page 161)
Corrected MPU0 Memory Protection End Address from 0x026203FF to 0x026207FF (Page 209)
Removed SmartReflex Class0 from Device State Control Registers table (Page 78)
Revised IPCGRH register description (Page 89)
Corrected DDR3 transfer rate from 1033 MTS to 1066 MTS (Page 228)
Corrected pin number AD21 to AK21 for PCIESSMODE[1:0] in Device Configuration Pins table (Page 76)
Added CVDD and SmartReflex voltage parameter in SmartReflex switching table (Page 134)
Removed DDR3 PLL initialization sequence from data manual to PLL controller user guide (Page 161)
Removed PASS PLL initialization sequence from data manual to PLL controller user guide (Page 165)
Updated EMIF16 CS to CE in the timing table (Page 252)
Updated EMIF16 CS[5:2] to CE[3:0] (Page 29)
Added footnote for DDR3 EMIF data in memory map summary table (Page 29)
Updated CVDD core supply range from 0.9v-1.1V to 0.85-1.1V (Page 124)
Fixed AIF_SEVTn numbering and descriptions in C66x CorePac Primary Interrupts table (Page 176)
Updated Tracer descriptions across the data manual (Page 21)
Corrected PASSCLK(N/P) max cycle time from 6.4ns to 25ns (Page 165)
Corrected PASSCLK(N/P) max cycle time from 6.4ns to 25ns (Page 165)
Corrected performance numbers (for 1.2GHz) in Device Description (Page 14)
Added priority related registers and updated the host interrupt register numbers (Page 202)
Corrected CORECLKSEL pin number from AB25 to AC26 (Page 76)
Updated the DDR3 MMR descriptions and deleted the unrelated PCIe MMR descriptions for soft reset. (Page 142)
Added event #180 PONIRQ in the CIC0 event inputs table (Page 184)
Added Boot Mode Pins Mapping table with CorePac as Boot Master (Page 31)
Corrected the bits values of Boot Device Field (Page 31)
Corrected the description of NAND boot that DEVSTAT values are used (Page 37)
Updated bits values of No Boot/EMIF16 configuration field (Page 32)
Removed Packet Accelerator from the descriptions of Queue Manager pend event (Page 183)
Added note for VCNTLID register that it is available for debug purpose only (Page 137)
Updated tw(DPnH) and tw(DPnL) descriptions in Trace Switching Characteristics tables (Page 256)

Revision C

Updated JTAG timing (Page 258)
Updated power domain table notes (Page 135)
Updated UART delay time max boundary (Page 239)

Added clarification for RESETSTATz input current (Page 125)
Updated address ranges (Page 209)
Updated hold time for MDIO to 0 (Page 243)
Added CoreSight Trace timing table (Page 257)
Added STM Trace timing table (Page 256)
Updated CPTS_RFTCLK_SEL values (Page 242)
Updated HyperLink timing (Page 237)
Updated JTAG timing (Page 258)
Updated max value (Page 165)
Updated max value (Page 165)
Updated skew timing value (Page 256)
Updated Trace output skew time (Page 257)
Updated Trace output skew time (Page 256)
Updated Trace output skew time (Page 256)
Added DEVSPEED register information (Page 93)
Added DEVSPEED register location (Page 80)
Changed HyperLink address range (Page 28)
Changed PLL lock time (Page 148)
Changed SPI range (Page 28)
Changed the output divide range (Page 150)
Updated DDR3 address range (Page 29)
Updated EMIF address range (Page 28)
Updated PHY Sync and Rad Sync timing numbers (Page 249)
Updated TBD parameters (Page 207)
Updated timing numbers (Page 258)
Added FFTC-B configuration space for 6612 device in memory map table (Page 23)
Updated DEVSPEED Register (Page 93)
Removed PBIST columns (Page 99)

B Mechanical Data

B.1 Thermal Data

Table B-1 shows the thermal resistance characteristics for the mechanical package.

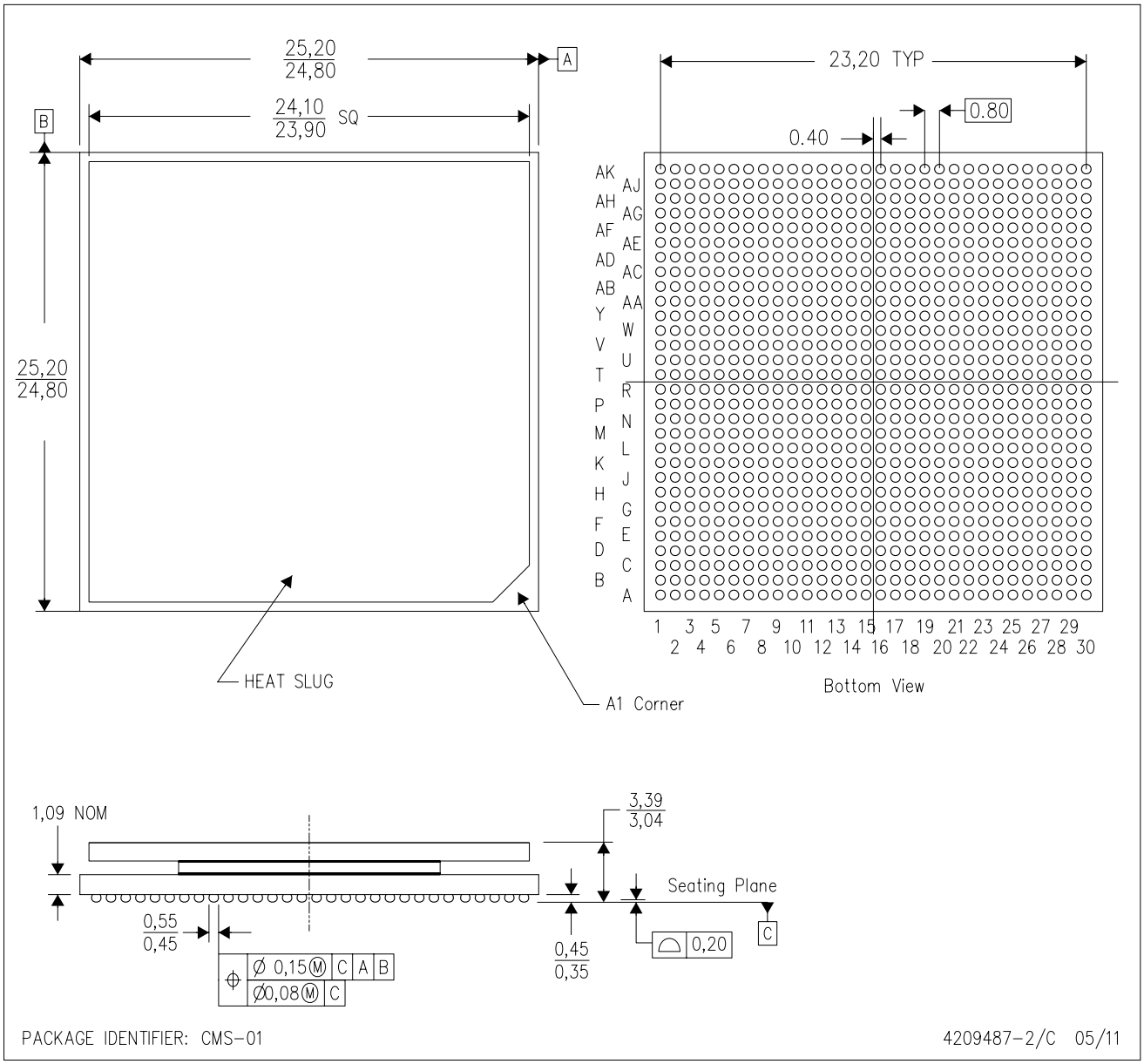
Table B-1 Thermal Resistance Characteristics for the CMS 900-Pin Plastic BGA Package (PBGA Package)

No.			°C/W
1	$R\theta_{JC}$	Junction-to-case	0.14
2	$R\theta_{JB}$	Junction-to-board	3.00
End of Table B-1			

B.2 Packaging Information

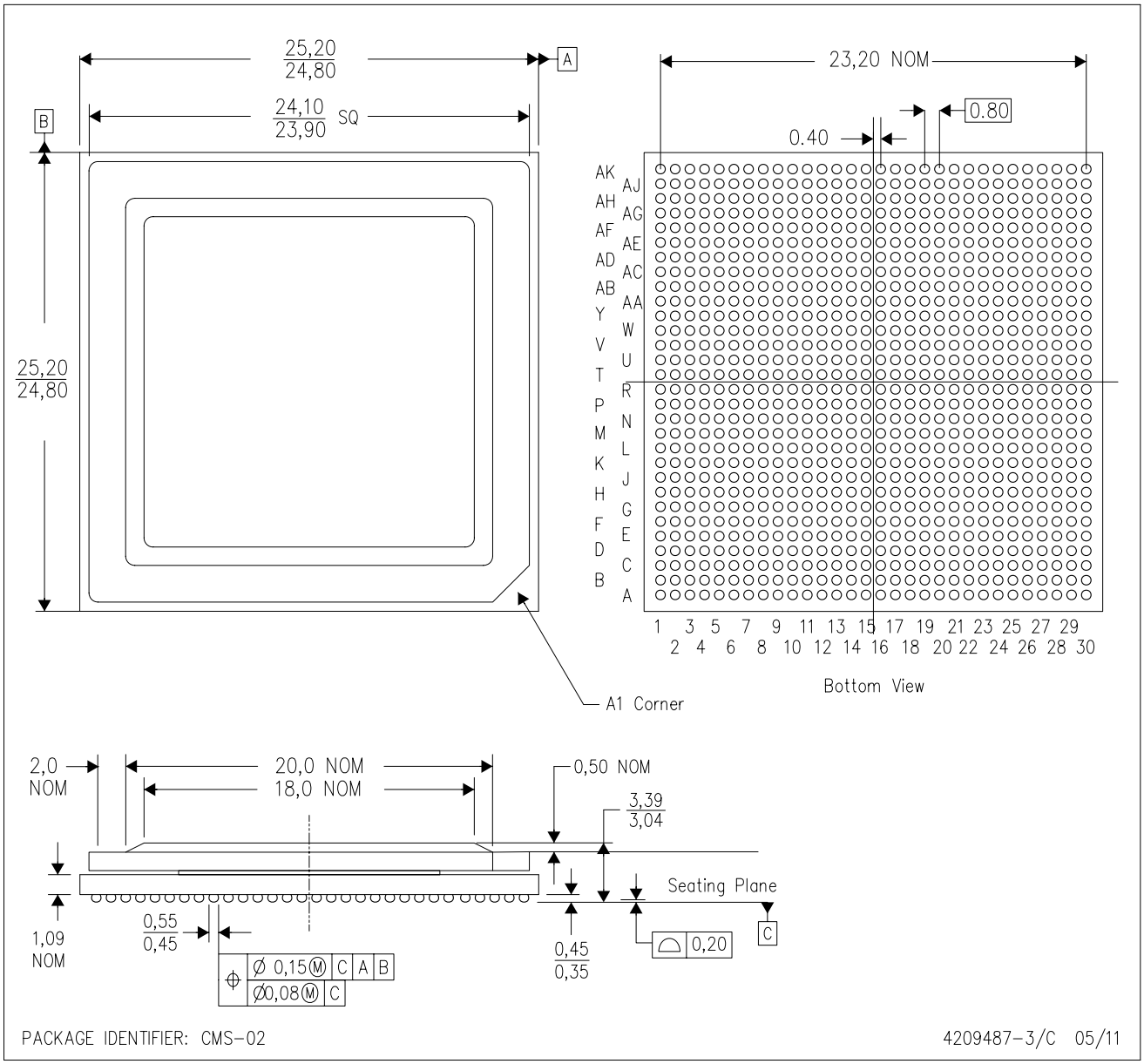
The following packaging information reflects the most current released data available for the designated device(s). This data is subject to change without notice and without revision of this document.

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.

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

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265