- **Highest-Performance Fixed-Point Digital Signal Processors (DSPs)**
  - **2-, 1.67-, 1.39-ns Instruction Cycle Time**
  - **600-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **Twenty-Eight Operations/Cycle**
  - **4800 MIPS**
  - **Fully Software-Compatible With C62x™**
  - **C6414/15/16 Devices Pin-Compatible**
- **VelociTI.2™ Extensions to VelociTI™ Advanced Very-Long-Instruction-Word (VLIW) TMS320C64x™ DSP Core**
  - **Eight Highly Independent Functional Units With VelociTI.2™ Extensions:**
    - **Six ALUs (32-/40-Bit), Each Supports Single 32-Bit, Dual 16-Bit, or Quad 8-Bit Arithmetic per Clock Cycle**
    - **Two Multipliers Support Four 16 x 16-Bit Multiplies (32-Bit Results) per Clock Cycle or Eight 8 x 8-Bit Multiplies (16-Bit Results) per Clock Cycle**
  - **Non-Aligned Load-Store Architecture**
  - **64 32-Bit General-Purpose Registers**
  - **Instruction Packing Reduces Code Size**
  - **All Instructions Conditional**
- **Instruction Set Features**
  - **Byte-Addressable (8-/16-/32-/64-Bit Data)**
  - **8-Bit Overflow Protection**
  - **Bit-Field Extract, Set, Clear**
  - **Normalization, Saturation, Bit-Counting**
  - **VelociTI.2™ Increased Orthogonality**
- **Viterbi Decoder Coprocessor (VCP) [C6416]**
  - **Supports Over 500 7.95-Kbps AMR**
  - **Programmable Code Parameters**
- **Turbo Decoder Coprocessor (TCP) [C6416]**
  - **Supports up to Six 2-Mbps 3GPP (6 Iterations)**
  - **Programmable Turbo Code and Decoding Parameters**
- **L1/L2 Memory Architecture**
  - **128K-Bit (16K-Byte) L1P Program Cache (Direct Mapped)**
  - **128K-Bit (16K-Byte) L1D Data Cache (2-Way Set-Associative)**
  - **8M-Bit (1024K-Byte) L2 Unified Mapped RAM/Cache (Flexible Allocation)**
- **Two External Memory Interfaces (EMIFs)**
  - **One 64-Bit (EMIFA), One 16-Bit (EMIFB)**
  - **Glueless Interface to Asynchronous Memories (SRAM and EPROM) and Synchronous Memories (SDRAM, SBSRAM, ZBT SRAM, and FIFO)**
  - **1280M-Byte Total Addressable External Memory Space**
- **Enhanced Direct-Memory-Access (EDMA) Controller (64 Independent Channels)**
- **Host-Port Interface (HPI)**
  - **User-Configurable Bus Width (32-/16-Bit)**
- **32-Bit/33-MHz, 3.3-V PCI Master/Slave Interface Conforms to PCI Specification 2.2 [C6415/C6416]**
  - **Three PCI Bus Address Registers: Prefetchable Memory Non-Prefetchable Memory I/O**
  - **Four-Wire Serial EEPROM Interface**
  - **PCI Interrupt Request Under DSP Program Control**
  - **DSP Interrupt Via PCI I/O Cycle**
- **Three Multichannel Buffered Serial Ports**
  - **Direct Interface to T1/E1, MVIP, SCSA Framers**
  - **Up to 256 Channels Each**
  - **ST-Bus-Switching-, AC97-Compatible**
  - **Serial Peripheral Interface (SPI) Compatible (Motorola™)**
- **Three 32-Bit General-Purpose Timers**
- **Universal Test and Operations PHY Interface for ATM (UTOPIA) [C6415/C6416]**
  - **UTOPIA Level 2 Slave ATM Controller**
  - **8-Bit Transmit and Receive Operations up to 50 MHz per Direction**
  - **User-Defined Cell Format up to 64 Bytes**
- **Sixteen General-Purpose I/O (GPIO) Pins**
- **Flexible PLL Clock Generator**
- **IEEE-1149.1 (JTAG†) Boundary-Scan-Compatible**
- **570-Pin Grid Array (PGA) Package (GAD Suffix)**
- **0.13-μm/6-Level C*u* Metal Process (CMOS)**
- **3.3-V I/Os, 1.4-V Internal**

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

C62x, VelociTI.2, VelociTI, and TMS320C64x are trademarks of Texas Instruments.
Motorola is a trademark of Motorola, Inc.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

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.

Copyright © 2004, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

Table of Contents

REVISION HISTORY

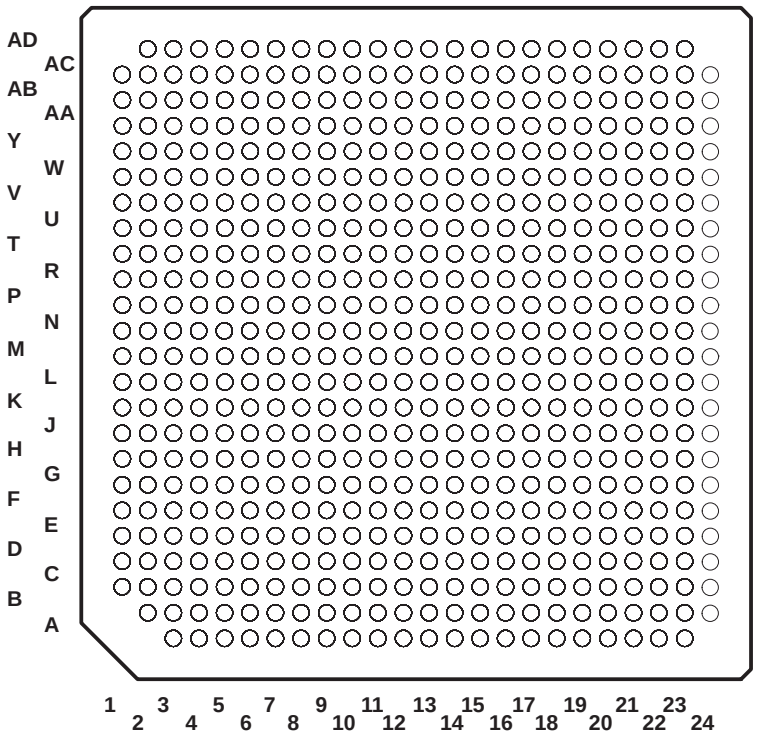
This data sheet revision history highlights the technical changes made to the SMJ320C6414, SMJ320C6415, and SMJ320C6416 device-specific data sheet.

Scope: Applicable updates to the C64x device family, specifically relating to the C6414, C6415, and C6416 devices, have been incorporated.

PAGE(S) NO.	ADDITIONS/CHANGES/DELETIONS
All	Original release
3, 81, 82, 83	Removed pin A24, changed Cycle-to-cycle jitter to Period jitter.

GAD Ceramic PGA package (bottom view)

GAD CERAMIC 570-PIN GRID ARRAY (PGA) PACKAGE
(BOTTOM VIEW)



description

The TMS320C64x™ DSPs (including the SMJ320C6414, SMJ320C6415, and SMJ320C6416 devices) are the highest-performance fixed-point DSP generation in the TMS320C6000™ DSP platform. The TMS320C64x™ (C64x™†) device is based on the second-generation high-performance, advanced VelociTI™ very-long-instruction-word (VLIW) architecture (VelociTI.2™) developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunctional applications. The C64x™ is a code-compatible member of the C6000™ DSP platform.

With performance of up to 5760 million instructions per second (MIPS) at a clock rate of 720 MHz, the C64x devices offer cost-effective solutions to high-performance DSP programming challenges. The C64x DSPs possess the operational flexibility of high-speed controllers and the numerical capability of array processors. The C64x™ DSP core processor has 64 general-purpose registers of 32-bit word length and eight highly independent functional units—two multipliers for a 32-bit result and six arithmetic logic units (ALUs)— with VelociTI.2™ extensions. The VelociTI.2™ extensions in the eight functional units include new instructions to accelerate the performance in key applications and extend the parallelism of the VelociTI™ architecture. The C64x can produce four 32-bit multiply-accumulates (MACs) per cycle for a total of 2400 million MACs per second (MMACS), or eight 8-bit MACs per cycle for a total of 4800 MMACS. The C64x DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals similar to the other C6000™ DSP platform devices.

The C6416 device has two high-performance embedded coprocessors [Viterbi Decoder Coprocessor (VCP) and Turbo Decoder Coprocessor (TCP)] that significantly speed up channel-decoding operations on-chip. The VCP operating at CPU clock divided-by-4 can decode over 500 7.95-Kbps adaptive multi-rate (AMR) [K = 9, R = 1/3] voice channels. The VCP supports constraint lengths K = 5, 6, 7, 8, and 9, rates R = 1/2, 1/3, and 1/4, and flexible polynomials, while generating hard decisions or soft decisions. The TCP operating at CPU clock divided-by-2 can decode up to thirty-six 384-Kbps or six 2-Mbps turbo encoded channels (assuming 6 iterations). The TCP implements the max*log-map algorithm and is designed to support all polynomials and rates required by Third-Generation Partnership Projects (3GPP and 3GPP2), with fully programmable frame length and turbo interleaver. Decoding parameters such as the number of iterations and stopping criteria are also programmable. Communications between the VCP/TCP and the CPU are carried out through the EDMA controller.

The C64x uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 128K-bit direct mapped cache and the Level 1 data cache (L1D) is a 128K-bit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of an 8M-bit memory space that is shared between program and data space. L2 memory can be configured as mapped memory or combinations of cache (up to 256K bytes) and mapped memory. The peripheral set includes three multichannel buffered serial ports (McBSPs); an 8-bit Universal Test and Operations PHY Interface for Asynchronous Transfer Mode (ATM) Slave [UTOPIA Slave] port (C6415/C6416 only); three 32-bit general-purpose timers; a user-configurable 16-bit or 32-bit host-port interface (HPI16/HPI32); a peripheral component interconnect (PCI) [C6415/C6416 only]; a general-purpose input/output port (GPIO) with 16 GPIO pins; and two glueless external memory interfaces (64-bit EMIFA and 16-bit EMIFB‡), both of which are capable of interfacing to synchronous and asynchronous memories and peripherals.

The C64x has a complete set of development tools which includes: an advanced C compiler with C64x-specific enhancements, an assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

TMS320C6000, C64x, and C6000 are trademarks of Texas Instruments.
Windows is a registered trademark of the Microsoft Corporation.
All trademarks are the property of their respective owners.

† Throughout the remainder of this document, the SMJ320C6414, SMJ320C6415, and SMJ320C6416 shall be referred to as SMJ320C64x or C64x where generic, and where specific, their individual full device part numbers will be used or abbreviated as C6414, C6415, or C6416.

‡ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

device characteristics

Table 1 provides an overview of the C6414, C6415, and C6416 DSPs. The table shows significant features of the C64x devices, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

Table 1. Characteristics of the C6414, C6415, and C6416 Processors

HARDWARE FEATURES		C6414, C6415, AND C6416
Peripherals Not all peripherals pins are available at the same time. (For more details, see the Device Configuration section.) Peripheral performance is dependent on chip-level configuration.	EMIFA (64-bit bus width) (default clock source = AECLKIN)	1
	EMIFB (16-bit bus width) (default clock source = BECLKIN)	1
	EDMA (64 independent channels)	1
	HPI (32- or 16-bit user selectable)	1 (HPI16 or HPI32)
	PCI (32-bit) [DeviceID Register value 0xA106]	1 **[C6415/C6416 only]**
	McBSPs (default internal clock source = CPU/4 clock frequency)	3
	UTOPIA (8-bit mode)	1 **[C6415/C6416 only]**
	32-Bit Timers (default internal clock source = CPU/8 clock frequency)	3
	General-Purpose Input/Output 0 (GP0)	16
Decoder Coprocessors	VCP	1 **(C6416 only)**
	TCP	1 **(C6416 only)**
On-Chip Memory	Size (Bytes)	1056K
	Organization	16K-Byte (16KB) L1 Program (L1P) Cache 16KB L1 Data (L1D) Cache 1024KB Unified Mapped RAM/Cache (L2)
CPU ID + CPU Rev ID	Control Status Register (CSR.[31:16])	0x0C01
Device_ID	Silicon Revision Identification Register (DEVICE_REV [19:16]) Address: 0x01B0 0200	DEVICE_REV[19:16] Silicon Revision 1111 1.03 or earlier 0001 1.03 0010 or 0000 1.1
Frequency	MHz	600
Cycle Time	ns	1.67 ns (C6414, C6415, C6416) and (C6414A, C6415A, C6416A) [600-MHz CPU, 133-MHz EMIFA]†
Voltage	Core (V)	1.4 V
	I/O (V)	3.3 V
PLL Options	CLKIN frequency multiplier	Bypass (x1), x6, x12
PGA Package	33 x 33 mm	570-Pin PGA (GAD)
Process Technology	µm	0.13 µm
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	PD

† On these C64x™ devices, the rated EMIF speed affects only the SDRAM interface on EMIFA. For more detailed information, see the EMIF Device Speed section of this data sheet.

device compatibility

The C64x™ generation of devices has a diverse and powerful set of peripherals. The common peripheral set and pin-compatibility that the C6414, C6415, and C6416 devices offer lead to easier system designs and faster time to market. Table 2 identifies the peripherals and coprocessors that are available on the C6414, C6415, and C6416 devices.

The C6414, C6415, and C6416 devices are pin-for-pin compatible, provided the following conditions are met:

- All devices are using the same peripherals.
  The C6414 is pin-for-pin compatible with the C6415/C6416 when the PCI and UTOPIA peripherals on the C6415/C6416 are disabled.
  The C6415 is pin-for-pin compatible with the C6416 when they are in the same peripheral selection mode.
  [For more information on peripheral selection, see the Device Configurations section of this data sheet.]
- The BEA[9:7] pins are properly pulled up/down.
  [For more details on the device-specific BEA[9:7] pin configurations, see the Terminal Functions table of this data sheet.]

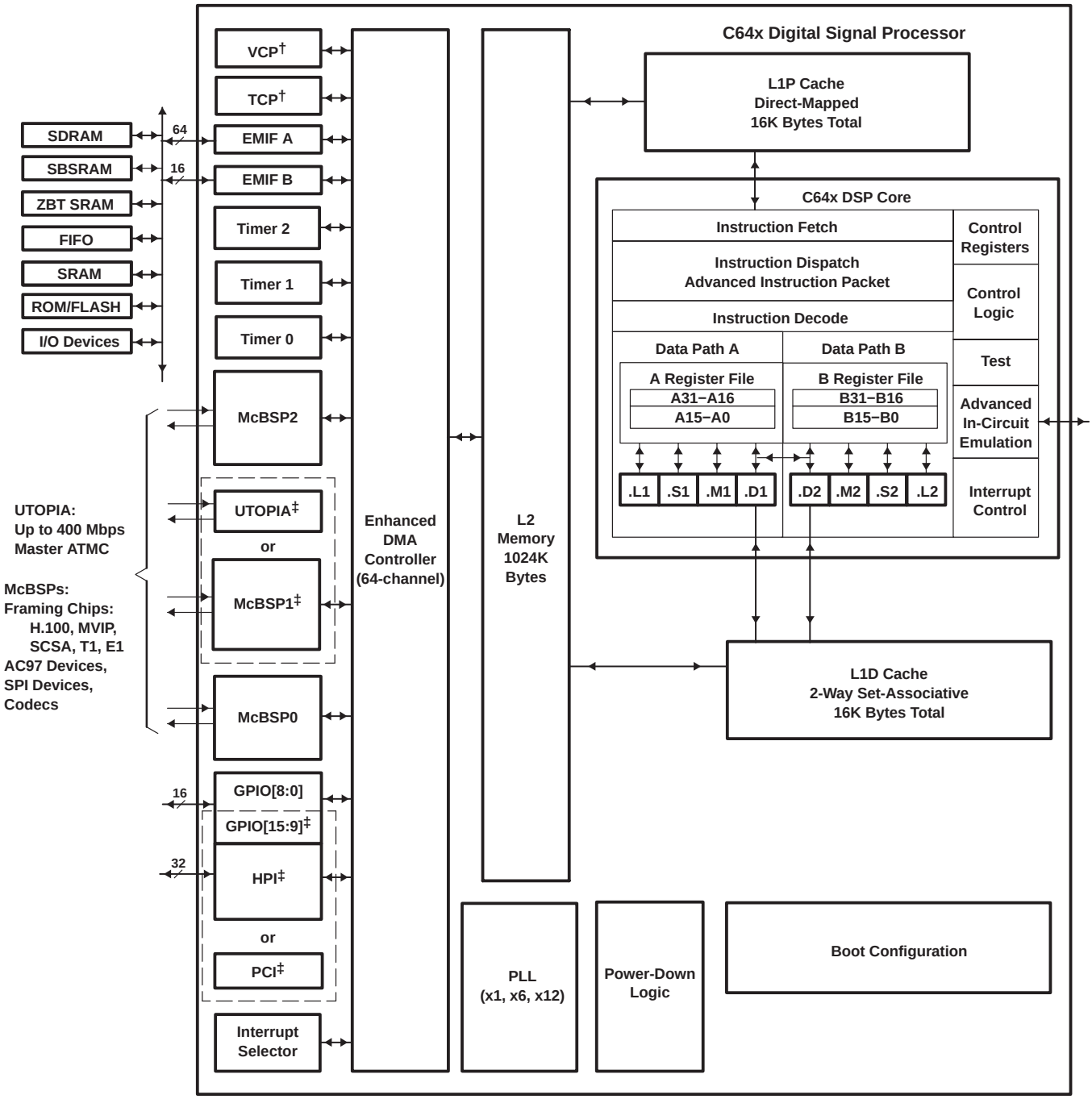
Table 2. Peripherals and Coprocessors Available on the C6414, C6415, and C6416 Devices†‡

PERIPHERALS/COPROCESSORS	C6414	C6415	C6416
EMIFA (64-bit bus width)	√	√	√
EMIFB (16-bit bus width)	√	√	√
EDMA (64 independent channels)	√	√	√
HPI (32- or 16-bit user selectable)	√	√	√
PCI (32-bit) [Specification v2.2]	—	√	√
McBSPs (McBSP0, McBSP1, McBSP2)	√	√	√
UTOPIA (8-bit mode) [Specification v1.0]	—	√	√
Timers (32-bit) [TIMER0, TIMER1, TIMER2]	√	√	√
GPIOs (GP[15:0])	√	√	√
VCP/TCP Coprocessors	—	—	√

† — denotes peripheral/coprocessor is *not* available on this device.
‡ Not all peripherals pins are available at the same time. (For more details, see the Device Configuration section.)

For more detailed information on the device compatibility and similarities/differences among the C6414, C6415, and C6416 devices, see the *How To Begin Development Today With the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs* application report (literature number SPRA718).

functional block and CPU (DSP core) diagram



† VCP and TCP decoder coprocessors are applicable to the C6416 device only.

‡ For the C6415 and C6416 devices, the UTOPIA peripheral is MUXed with McBSP1, and the PCI peripheral is MUXed with the HPI peripheral and the GPIO[15:9] port. For more details on the multiplexed pins of these peripherals, see the Device Configurations section of this data sheet.

CPU (DSP core) description

The CPU fetches VelociTI™ advanced very-long instruction words (VLIWs) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VelociTI™ VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the C64x CPUs from other VLIW architectures. The C64x™ VelociTI.2™ extensions add enhancements to the TMS320C62x™ DSP VelociTI™ architecture. These enhancements include:

- Register file enhancements
- Data path extensions
- Quad 8-bit and dual 16-bit extensions with data flow enhancements
- Additional functional unit hardware
- Increased orthogonality of the instruction set
- Additional instructions that reduce code size and increase register flexibility

The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files each contain 32 32-bit registers for a total of 64 general-purpose registers. In addition to supporting the packed 16-bit and 32-/40-bit fixed-point data types found in the C62x™ VelociTI™ VLIW architecture, the C64x™ register files also support packed 8-bit data and 64-bit fixed-point data types. The two sets of functional units, along with two register files, compose sides A and B of the CPU [see the functional block and CPU (DSP core) diagram, and Figure 1]. The four functional units on each side of the CPU can freely share the 32 registers belonging to that side. Additionally, each side features a “data cross path”—a single data bus connected to all the registers on the other side, by which the two sets of functional units can access data from the register files on the opposite side. The C64x CPU pipelines data-cross-path accesses over multiple clock cycles. This allows the same register to be used as a data-cross-path operand by multiple functional units in the same execute packet. All functional units in the C64x CPU can access operands via the data cross path. Register access by functional units on the same side of the CPU as the register file can service all the units in a single clock cycle. On the C64x CPU, a delay clock is introduced whenever an instruction attempts to read a register via a data cross path if that register was updated in the previous clock cycle.

In addition to the C62x™ DSP fixed-point instructions, the C64x™ DSP includes a comprehensive collection of quad 8-bit and dual 16-bit instruction set extensions. These VelociTI.2™ extensions allow the C64x CPU to operate directly on packed data to streamline data flow and increase instruction set efficiency.

Another key feature of the C64x CPU is the load/store architecture, where all instructions operate on registers (as opposed to data in memory). Two sets of data-addressing units (.D1 and .D2) are responsible for all data transfers between the register files and the memory. The data address driven by the .D units allows data addresses generated from one register file to be used to load or store data to or from the other register file. The C64x .D units can load and store bytes (8 bits), half-words (16 bits), and words (32 bits) with a single instruction. And with the new data path extensions, the C64x .D unit can load and store doublewords (64 bits) with a single instruction. Furthermore, the non-aligned load and store instructions allow the .D units to access words and doublewords on any byte boundary. The C64x CPU supports a variety of indirect addressing modes using either linear- or circular-addressing with 5- or 15-bit offsets. All instructions are conditional, and most can access any one of the 64 registers. Some registers, however, are singled out to support specific addressing modes or to hold the condition for conditional instructions (if the condition is not automatically “true”).

TMS320C62x is a trademark of Texas Instruments.

CPU (DSP core) description (continued)

The two .M functional units perform all multiplication operations. Each of the C64x .M units can perform two 16 × 16-bit multiplies or four 8 × 8-bit multiplies per clock cycle. The .M unit can also perform 16 × 32-bit multiply operations, dual 16 × 16-bit multiplies with add/subtract operations, and quad 8 × 8-bit multiplies with add operations. In addition to standard multiplies, the C64x .M units include bit-count, rotate, Galois field multiplies, and bidirectional variable shift hardware.

The two .S and .L functional units perform a general set of arithmetic, logical, and branch functions with results available every clock cycle. The arithmetic and logical functions on the C64x CPU include single 32-bit, dual 16-bit, and quad 8-bit operations.

The processing flow begins when a 256-bit-wide instruction fetch packet is fetched from a program memory. The 32-bit instructions destined for the individual functional units are “linked” together by “1” bits in the least significant bit (LSB) position of the instructions. The instructions that are “chained” together for simultaneous execution (up to eight in total) compose an execute packet. A “0” in the LSB of an instruction breaks the chain, effectively placing the instructions that follow it in the next execute packet. A C64x™ DSP device enhancement now allows execute packets to cross fetch-packet boundaries. In the TMS320C62x™/TMS320C67x™ DSP devices, if an execute packet crosses the fetch-packet boundary (256 bits wide), the assembler places it in the next fetch packet, while the remainder of the current fetch packet is padded with NOP instructions. In the C64x™ DSP device, the execute boundary restrictions have been removed, thereby, eliminating all of the NOPs added to pad the fetch packet, and thus, decreasing the overall code size. The number of execute packets within a fetch packet can vary from one to eight. Execute packets are dispatched to their respective functional units at the rate of one per clock cycle and the next 256-bit fetch packet is not fetched until all the execute packets from the current fetch packet have been dispatched. After decoding, the instructions simultaneously drive all active functional units for a maximum execution rate of eight instructions every clock cycle. While most results are stored in 32-bit registers, they can be subsequently moved to memory as bytes, half-words, words, or doublewords. All load and store instructions are byte-, half-word-, word-, or doubleword-addressable.

For more details on the C64x CPU functional units enhancements, see the following documents:

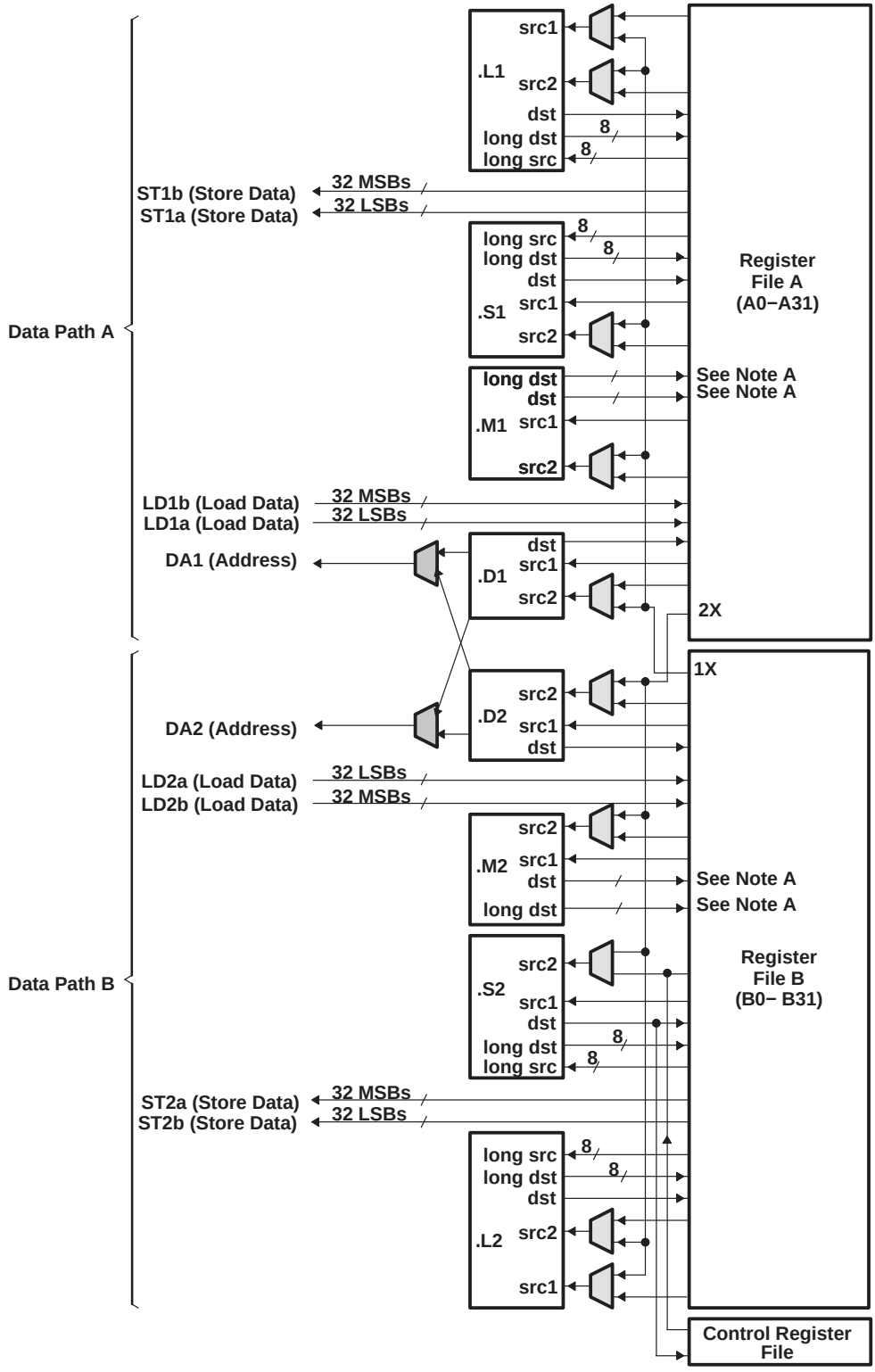
The *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189)

TMS320C64x Technical Overview (literature number SPRU395)

How To Begin Development Today With the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs application report (literature number SPRA718)

TMS320C67x is a trademark of Texas Instruments.

CPU (DSP core) description (continued)



NOTE A: For the .M functional units, the long dst is 32 MSBs and the dst is 32 LSBs.

Figure 1. SMJ320C64x™ CPU (DSP Core) Data Paths

memory map summary

Table 3 shows the memory map address ranges of the SMJ320C64x device. Internal memory is always located at address 0 and can be used as both program and data memory. The external memory address ranges in the C64x device begin at the hex address locations 0x6000 0000 for EMIFB and 0x8000 0000 for EMIFA.

memory map summary (continued)

Table 3. SMJ320C64x Memory Map Summary

MEMORY BLOCK DESCRIPTION	BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
Internal RAM (L2)	1M	0000 0000 – 000F FFFF
Reserved	23M	0010 0000 – 017F FFFF
External Memory Interface A (EMIFA) Registers	256K	0180 0000 – 0183 FFFF
L2 Registers	256K	0184 0000 – 0187 FFFF
HPI Registers	256K	0188 0000 – 018B FFFF
McBSP 0 Registers	256K	018C 0000 – 018F FFFF
McBSP 1 Registers	256K	0190 0000 – 0193 FFFF
Timer 0 Registers	256K	0194 0000 – 0197 FFFF
Timer 1 Registers	256K	0198 0000 – 019B FFFF
Interrupt Selector Registers	256K	019C 0000 – 019F FFFF
EDMA RAM and EDMA Registers	256K	01A0 0000 – 01A3 FFFF
McBSP 2 Registers	256K	01A4 0000 – 01A7 FFFF
EMIFB Registers	256K	01A8 0000 – 01AB FFFF
Timer 2 Registers	256K	01AC 0000 – 01AF FFFF
GPIO Registers	256K	01B0 0000 – 01B3 FFFF
UTOPIA Registers (C6415 and C6416 only)†	256K	01B4 0000 – 01B7 FFFF
TCP/VCP Registers (C6416 only)‡	256K	01B8 0000 – 01BB FFFF
Reserved	256K	01BC 0000 – 01BF FFFF
PCI Registers (C6415 and C6416 only)†	256K	01C0 0000 – 01C3 FFFF
Reserved	4M – 256K	01C4 0000 – 01FF FFFF
QDMA Registers	52	0200 0000 – 0200 0033
Reserved	736M – 52	0200 0034 – 2FFF FFFF
McBSP 0 Data	64M	3000 0000 – 33FF FFFF
McBSP 1 Data	64M	3400 0000 – 37FF FFFF
McBSP 2 Data	64M	3800 0000 – 3BFF FFFF
UTOPIA Queues (C6415 and C6416 only)†	64M	3C00 0000 – 3FFF FFFF
Reserved	256M	4000 0000 – 4FFF FFFF
TCP/VCP (C6416 only)‡	256M	5000 0000 – 5FFF FFFF
EMIFB CE0	64M	6000 0000 – 63FF FFFF
EMIFB CE1	64M	6400 0000 – 67FF FFFF
EMIFB CE2	64M	6800 0000 – 6BFF FFFF
EMIFB CE3	64M	6C00 0000 – 6FFF FFFF
Reserved	256M	7000 0000 – 7FFF FFFF
EMIFA CE0	256M	8000 0000 – 8FFF FFFF
EMIFA CE1	256M	9000 0000 – 9FFF FFFF
EMIFA CE2	256M	A000 0000 – AFFF FFFF
EMIFA CE3	256M	B000 0000 – BFFF FFFF
Reserved	1G	C000 0000 – FFFF FFFF

† For the C6414 device, these memory address locations are reserved. The C6414 device does ***not*** support the UTOPIA and PCI peripherals.
‡ Only the C6416 device supports the VCP/TCP Coprocessors. For the C6414 and C6415 devices, these memory address locations are reserved.

peripheral register descriptions

Table 4 through Table 23 identify the peripheral registers for the C6414, C6415, and C6416 devices by their register names, acronyms, and hex address or hex address range. For more detailed information on the register contents, bit names and their descriptions, see the specific peripheral reference guide listed in the *TMS320C6000 DSP Peripherals Overview Reference Guide* (literature number SPRU190).

Table 4. EMIFA Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
0180 0000	GBLCTL	EMIFA global control
0180 0004	CECTL1	EMIFA CE1 space control
0180 0008	CECTL0	EMIFA CE0 space control
0180 000C	−	Reserved
0180 0010	CECTL2	EMIFA CE2 space control
0180 0014	CECTL3	EMIFA CE3 space control
0180 0018	SDCTL	EMIFA SDRAM control
0180 001C	SDTIM	EMIFA SDRAM refresh control
0180 0020	SDEXT	EMIFA SDRAM extension
0180 0024 − 0180 003C	−	Reserved
0180 0040	PDTCTL	Peripheral device transfer (PDT) control
0180 0044	CESEC1	EMIFA CE1 space secondary control
0180 0048	CESEC0	EMIFA CE0 space secondary control
0180 004C	−	Reserved
0180 0050	CESEC2	EMIFA CE2 space secondary control
0180 0054	CESEC3	EMIFA CE3 space secondary control
0180 0058 − 0183 FFFF	–	Reserved

Table 5. EMIFB Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
01A8 0000	GBLCTL	EMIFB global control
01A8 0004	CECTL1	EMIFB CE1 space control
01A8 0008	CECTL0	EMIFB CE0 space control
01A8 000C	−	Reserved
01A8 0010	CECTL2	EMIFB CE2 space control
01A8 0014	CECTL3	EMIFB CE3 space control
01A8 0018	SDCTL	EMIFB SDRAM control
01A8 001C	SDTIM	EMIFB SDRAM refresh control
01A8 0020	SDEXT	EMIFB SDRAM extension
01A8 0024 − 01A8 003C	−	Reserved
01A8 0040	PDTCTL	Peripheral device transfer (PDT) control
01A8 0044	CESEC1	EMIFB CE1 space secondary control
01A8 0048	CESEC0	EMIFB CE0 space secondary control
01A8 004C	−	Reserved
01A8 0050	CESEC2	EMIFB CE2 space secondary control
01A8 0054	CESEC3	EMIFB CE3 space secondary control
01A8 0058 − 01AB FFFF	–	Reserved

peripheral register descriptions (continued)

Table 6. L2 Cache Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0184 0000	CCFG	Cache configuration register	
0184 0004 − 0184 0FFC	−	Reserved	
0184 1000	EDMAWEIGHT	L2 EDMA access control register	
0184 1004 − 0184 1FFC	−	Reserved	
0184 2000	L2ALLOC0	L2 allocation register 0	
0184 2004	L2ALLOC1	L2 allocation register 1	
0184 2008	L2ALLOC2	L2 allocation register 2	
0184 200C	L2ALLOC3	L2 allocation register 3	
0184 2010 − 0184 3FFC	−	Reserved	
0184 4000	L2FBAR	L2 flush base address register	
0184 4004	L2FWC	L2 flush word count register	
0184 4010	L2CBAR	L2 clean base address register	
0184 4014	L2CWC	L2 clean word count register	
0184 4020	L1PFBAR	L1P flush base address register	
0184 4024	L1PFWC	L1P flush word count register	
0184 4030	L1DFBAR	L1D flush base address register	
0184 4034	L1DFWC	L1D flush word count register	
0184 4038 − 0184 4FFC	−	Reserved	
0184 5000	L2FLUSH	L2 flush register	
0184 5004	L2CLEAN	L2 clean register	
0184 5008 − 0184 7FFC	−	Reserved	
0184 8000 − 0184 817C	MAR0 to MAR95	Reserved	
0184 8180	MAR96	Controls EMIFB CE0 range 6000 0000 − 60FF FFFF	
0184 8184	MAR97	Controls EMIFB CE0 range 6100 0000 − 61FF FFFF	
0184 8188	MAR98	Controls EMIFB CE0 range 6200 0000 − 62FF FFFF	
0184 818C	MAR99	Controls EMIFB CE0 range 6300 0000 − 63FF FFFF	
0184 8190	MAR100	Controls EMIFB CE1 range 6400 0000 − 64FF FFFF	
0184 8194	MAR101	Controls EMIFB CE1 range 6500 0000 − 65FF FFFF	
0184 8198	MAR102	Controls EMIFB CE1 range 6600 0000 − 66FF FFFF	
0184 819C	MAR103	Controls EMIFB CE1 range 6700 0000 − 67FF FFFF	
0184 81A0	MAR104	Controls EMIFB CE2 range 6800 0000 − 68FF FFFF	
0184 81A4	MAR105	Controls EMIFB CE2 range 6900 0000 − 69FF FFFF	
0184 81A8	MAR106	Controls EMIFB CE2 range 6A00 0000 − 6AFF FFFF	
0184 81AC	MAR107	Controls EMIFB CE2 range 6B00 0000 − 6BFF FFFF	
0184 81B0	MAR108	Controls EMIFB CE3 range 6C00 0000 − 6CFF FFFF	
0184 81B4	MAR109	Controls EMIFB CE3 range 6D00 0000 − 6DFF FFFF	
0184 81B8	MAR110	Controls EMIFB CE3 range 6E00 0000 − 6EFF FFFF	
0184 81BC	MAR111	Controls EMIFB CE3 range 6F00 0000 − 6FFF FFFF	
0184 81C0 − 0184 81FC	MAR112 to MAR127	Reserved	
0184 8200	MAR128	Controls EMIFA CE0 range 8000 0000 − 80FF FFFF	
0184 8204	MAR129	Controls EMIFA CE0 range 8100 0000 − 81FF FFFF	

Table 6. L2 Cache Registers (Continued)

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0184 8208	MAR130	Controls EMIFA CE0 range 8200 0000 − 82FF FFFF	
0184 820C	MAR131	Controls EMIFA CE0 range 8300 0000 − 83FF FFFF	
0184 8210	MAR132	Controls EMIFA CE0 range 8400 0000 − 84FF FFFF	
0184 8214	MAR133	Controls EMIFA CE0 range 8500 0000 − 85FF FFFF	
0184 8218	MAR134	Controls EMIFA CE0 range 8600 0000 − 86FF FFFF	
0184 821C	MAR135	Controls EMIFA CE0 range 8700 0000 − 87FF FFFF	
0184 8220	MAR136	Controls EMIFA CE0 range 8800 0000 − 88FF FFFF	
0184 8224	MAR137	Controls EMIFA CE0 range 8900 0000 − 89FF FFFF	
0184 8228	MAR138	Controls EMIFA CE0 range 8A00 0000 − 8AFF FFFF	
0184 822C	MAR139	Controls EMIFA CE0 range 8B00 0000 − 8BFF FFFF	
0184 8230	MAR140	Controls EMIFA CE0 range 8C00 0000 − 8CFF FFFF	
0184 8234	MAR141	Controls EMIFA CE0 range 8D00 0000 − 8DFF FFFF	
0184 8238	MAR142	Controls EMIFA CE0 range 8E00 0000 − 8EFF FFFF	
0184 823C	MAR143	Controls EMIFA CE0 range 8F00 0000 − 8FFF FFFF	
0184 8240	MAR144	Controls EMIFA CE1 range 9000 0000 − 90FF FFFF	
0184 8244	MAR145	Controls EMIFA CE1 range 9100 0000 − 91FF FFFF	
0184 8248	MAR146	Controls EMIFA CE1 range 9200 0000 − 92FF FFFF	
0184 824C	MAR147	Controls EMIFA CE1 range 9300 0000 − 93FF FFFF	
0184 8250	MAR148	Controls EMIFA CE1 range 9400 0000 − 94FF FFFF	
0184 8254	MAR149	Controls EMIFA CE1 range 9500 0000 − 95FF FFFF	
0184 8258	MAR150	Controls EMIFA CE1 range 9600 0000 − 96FF FFFF	
0184 825C	MAR151	Controls EMIFA CE1 range 9700 0000 − 97FF FFFF	
0184 8260	MAR152	Controls EMIFA CE1 range 9800 0000 − 98FF FFFF	
0184 8264	MAR153	Controls EMIFA CE1 range 9900 0000 − 99FF FFFF	
0184 8268	MAR154	Controls EMIFA CE1 range 9A00 0000 − 9AFF FFFF	
0184 826C	MAR155	Controls EMIFA CE1 range 9B00 0000 − 9BFF FFFF	
0184 8270	MAR156	Controls EMIFA CE1 range 9C00 0000 − 9CFF FFFF	
0184 8274	MAR157	Controls EMIFA CE1 range 9D00 0000 − 9DFF FFFF	
0184 8278	MAR158	Controls EMIFA CE1 range 9E00 0000 − 9EFF FFFF	
0184 827C	MAR159	Controls EMIFA CE1 range 9F00 0000 − 9FFF FFFF	
0184 8280	MAR160	Controls EMIFA CE2 range A000 0000 − A0FF FFFF	
0184 8284	MAR161	Controls EMIFA CE2 range A100 0000 − A1FF FFFF	
0184 8288	MAR162	Controls EMIFA CE2 range A200 0000 − A2FF FFFF	
0184 828C	MAR163	Controls EMIFA CE2 range A300 0000 − A3FF FFFF	
0184 8290	MAR164	Controls EMIFA CE2 range A400 0000 − A4FF FFFF	
0184 8294	MAR165	Controls EMIFA CE2 range A500 0000 − A5FF FFFF	
0184 8298	MAR166	Controls EMIFA CE2 range A600 0000 − A6FF FFFF	
0184 829C	MAR167	Controls EMIFA CE2 range A700 0000 − A7FF FFFF	
0184 82A0	MAR168	Controls EMIFA CE2 range A800 0000 − A8FF FFFF	
0184 82A4	MAR169	Controls EMIFA CE2 range A900 0000 − A9FF FFFF	
0184 82A8	MAR170	Controls EMIFA CE2 range AA00 0000 − AAFF FFFF	
0184 82AC	MAR171	Controls EMIFA CE2 range AB00 0000 − ABFF FFFF	
0184 82B0	MAR172	Controls EMIFA CE2 range AC00 0000 − ACFF FFFF	
0184 82B4	MAR173	Controls EMIFA CE2 range AD00 0000 − ADFF FFFF	

Table 6. L2 Cache Registers (Continued)

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0184 82B8	MAR174	Controls EMIFA CE2 range AE00 0000 − AEFF FFFF	
0184 82BC	MAR175	Controls EMIFA CE2 range AF00 0000 − AFFF FFFF	
0184 82C0	MAR176	Controls EMIFA CE3 range B000 0000 − B0FF FFFF	
0184 82C4	MAR177	Controls EMIFA CE3 range B100 0000 − B1FF FFFF	
0184 82C8	MAR178	Controls EMIFA CE3 range B200 0000 − B2FF FFFF	
0184 82CC	MAR179	Controls EMIFA CE3 range B300 0000 − B3FF FFFF	
0184 82D0	MAR180	Controls EMIFA CE3 range B400 0000 − B4FF FFFF	
0184 82D4	MAR181	Controls EMIFA CE3 range B500 0000 − B5FF FFFF	
0184 82D8	MAR182	Controls EMIFA CE3 range B600 0000 − B6FF FFFF	
0184 82DC	MAR183	Controls EMIFA CE3 range B700 0000 − B7FF FFFF	
0184 82E0	MAR184	Controls EMIFA CE3 range B800 0000 − B8FF FFFF	
0184 82E4	MAR185	Controls EMIFA CE3 range B900 0000 − B9FF FFFF	
0184 82E8	MAR186	Controls EMIFA CE3 range BA00 0000 − BAFF FFFF	
0184 82EC	MAR187	Controls EMIFA CE3 range BB00 0000 − BBFF FFFF	
0184 82F0	MAR188	Controls EMIFA CE3 range BC00 0000 − BCFF FFFF	
0184 82F4	MAR189	Controls EMIFA CE3 range BD00 0000 − BDFF FFFF	
0184 82F8	MAR190	Controls EMIFA CE3 range BE00 0000 − BEFF FFFF	
0184 82FC	MAR191	Controls EMIFA CE3 range BF00 0000 − BFFF FFFF	
0184 8300 − 0184 83FC	MAR192 to MAR255	Reserved	
0184 8400 − 0187 FFFF	−	Reserved	

peripheral register descriptions (continued)

Table 7. EDMA Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
01A0 FF9C	EPRH	Event polarity high register
01A0 FFA4	CIPRH	Channel interrupt pending high register
01A0 FFA8	CIERH	Channel interrupt enable high register
01A0 FFAC	CCERH	Channel chain enable high register
01A0 FFB0	ERH	Event high register
01A0 FFB4	EERH	Event enable high register
01A0 FFB8	ECRH	Event clear high register
01A0 FFBC	ESRH	Event set high register
01A0 FFC0	PQAR0	Priority queue allocation register 0
01A0 FFC4	PQAR1	Priority queue allocation register 1
01A0 FFC8	PQAR2	Priority queue allocation register 2
01A0 FFCC	PQAR3	Priority queue allocation register 3
01A0 FFDC	EPRL	Event polarity low register
01A0 FFE0	PQSR	Priority queue status register
01A0 FFE4	CIPRL	Channel interrupt pending low register
01A0 FFE8	CIERL	Channel interrupt enable low register
01A0 FFEC	CCERL	Channel chain enable low register
01A0 FFF0	ERL	Event low register
01A0 FFF4	EERL	Event enable low register
01A0 FFF8	ECRL	Event clear low register
01A0 FFFC	ESRL	Event set low register
01A1 0000 − 01A3 FFFF	–	Reserved

peripheral register descriptions (continued)

Table 8. EDMA Parameter RAM†

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
01A0 0000 − 01A0 0017	−	Parameters for Event 0 (6 words)	
01A0 0018 − 01A0 002F	−	Parameters for Event 1 (6 words)	
01A0 0030 − 01A0 0047	−	Parameters for Event 2 (6 words)	
01A0 0048 − 01A0 005F	−	Parameters for Event 3 (6 words)	
01A0 0060 − 01A0 0077	−	Parameters for Event 4 (6 words)	
01A0 0078 − 01A0 008F	−	Parameters for Event 5 (6 words)	
01A0 0090 − 01A0 00A7	−	Parameters for Event 6 (6 words)	
01A0 00A8 − 01A0 00BF	−	Parameters for Event 7 (6 words)	
01A0 00C0 − 01A0 00D7	−	Parameters for Event 8 (6 words)	
01A0 00D8 − 01A0 00EF	−	Parameters for Event 9 (6 words)	
01A0 00F0 − 01A0 00107	−	Parameters for Event 10 (6 words)	
01A0 0108 − 01A0 011F	−	Parameters for Event 11 (6 words)	
01A0 0120 − 01A0 0137	−	Parameters for Event 12 (6 words)	
01A0 0138 − 01A0 014F	−	Parameters for Event 13 (6 words)	
01A0 0150 − 01A0 0167	−	Parameters for Event 14 (6 words)	
01A0 0168 − 01A0 017F	−	Parameters for Event 15 (6 words)	
01A0 0150 − 01A0 0167	−	Parameters for Event 16 (6 words)	
01A0 0168 − 01A0 017F	−	Parameters for Event 17 (6 words)	
...		...	
...		...	
01A0 05D0 − 01A0 05E7	−	Parameters for Event 62 (6 words)	
01A0 05E8 − 01A0 05FF	−	Parameters for Event 63 (6 words)	
01A0 0600 − 01A0 0617	−	Reload/link parameters for Event M (6 words)	
01A0 0618 − 01A0 062F	−	Reload/link parameters for Event N (6 words)	
...		...	
01A0 07E0 − 01A0 07F7	−	Reload/link parameters for Event Z (6 words)	
01A0 07F8 − 01A0 07FF	−	Scratch pad area (2 words)	

† The C64x device has twenty-one parameter sets [six (6) words each] that can be used to reload/link EDMA transfers.

Table 9. Quick DMA (QDMA) and Pseudo Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
0200 0000	QOPT	QDMA options parameter register
0200 0004	QSRC	QDMA source address register
0200 0008	QCNT	QDMA frame count register
0200 000C	QDST	QDMA destination address register
0200 0010	QIDX	QDMA index register
0200 0014 − 0200 001C		Reserved
0200 0020	QSOPT	QDMA pseudo options register
0200 0024	QSSRC	QDMA pseudo source address register
0200 0028	QSCNT	QDMA pseudo frame count register
0200 002C	QSDST	QDMA pseudo destination address register
0200 0030	QSIDX	QDMA pseudo index register

peripheral register descriptions (continued)

Table 10. Interrupt Selector Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
019C 0000	MUXH	Interrupt multiplexer high	Selects which interrupts drive CPU interrupts 10−15 (INT10−INT15)
019C 0004	MUXL	Interrupt multiplexer low	Selects which interrupts drive CPU interrupts 4−9 (INT04−INT09)
019C 0008	EXTPOL	External interrupt polarity	Sets the polarity of the external interrupts (EXT_INT4−EXT_INT7)
019C 000C − 019C 01FF	−	Reserved	

Table 11. McBSP 0 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
018C 0000	DRR0	McBSP0 data receive register via Configuration Bus	The CPU and EDMA controller can only read this register; they cannot write to it.
0x3000 0000 − 0x33FF FFFF	DRR0	McBSP0 data receive register via Peripheral Bus	
018C 0004	DXR0	McBSP0 data transmit register via Configuration Bus	
0x3000 0000 − 0x33FF FFFF	DXR0	McBSP0 data transmit register via Peripheral Bus	
018C 0008	SPCR0	McBSP0 serial port control register	
018C 000C	RCR0	McBSP0 receive control register	
018C 0010	XCR0	McBSP0 transmit control register	
018C 0014	SRGR0	McBSP0 sample rate generator register	
018C 0018	MCR0	McBSP0 multichannel control register	
018C 001C	RCERE00	McBSP0 enhanced receive channel enable register 0	
018C 0020	XCERE00	McBSP0 enhanced transmit channel enable register 0	
018C 0024	PCR0	McBSP0 pin control register	
018C 0028	RCERE10	McBSP0 enhanced receive channel enable register 1	
018C 002C	XCERE10	McBSP0 enhanced transmit channel enable register 1	
018C 0030	RCERE20	McBSP0 enhanced receive channel enable register 2	
018C 0034	XCERE20	McBSP0 enhanced transmit channel enable register 2	
018C 0038	RCERE30	McBSP0 enhanced receive channel enable register 3	
018C 003C	XCERE30	McBSP0 enhanced transmit channel enable register 3	
018C 0040 − 018F FFFF	–	Reserved	

peripheral register descriptions (continued)

Table 12. McBSP 1 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0190 0000	DRR1	McBSP1 data receive register via Configuration Bus	The CPU and EDMA controller can only read this register; they cannot write to it.
0x3400 0000 − 0x37FF FFFF	DRR1	McBSP1 data receive register via Peripheral Bus	
0190 0004	DXR1	McBSP1 data transmit register via Configuration Bus	
0x3400 0000 − 0x37FF FFFF	DXR1	McBSP1 data transmit register via Peripheral Bus	
0190 0008	SPCR1	McBSP1 serial port control register	
0190 000C	RCR1	McBSP1 receive control register	
0190 0010	XCR1	McBSP1 transmit control register	
0190 0014	SRGR1	McBSP1 sample rate generator register	
0190 0018	MCR1	McBSP1 multichannel control register	
0190 001C	RCERE01	McBSP1 enhanced receive channel enable register 0	
0190 0020	XCERE01	McBSP1 enhanced transmit channel enable register 0	
0190 0024	PCR1	McBSP1 pin control register	
0190 0028	RCERE11	McBSP1 enhanced receive channel enable register 1	
0190 002C	XCERE11	McBSP1 enhanced transmit channel enable register 1	
0190 0030	RCERE21	McBSP1 enhanced receive channel enable register 2	
0190 0034	XCERE21	McBSP1 enhanced transmit channel enable register 2	
0190 0038	RCERE31	McBSP1 enhanced receive channel enable register 3	
0190 003C	XCERE31	McBSP1 enhanced transmit channel enable register 3	
0190 0040 − 0193 FFFF	–	Reserved	

Table 13. McBSP 2 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
01A4 0000	DRR2	McBSP2 data receive register via Configuration Bus	The CPU and EDMA controller can only read this register; they cannot write to it.
0x3800 0000 − 0x3BFF FFFF	DRR2	McBSP2 data receive register via Peripheral Bus	
01A4 0004	DXR2	McBSP2 data transmit register via Configuration Bus	
0x3800 0000 − 0x3BFF FFFF	DXR2	McBSP2 data transmit register via Peripheral Bus	
01A4 0008	SPCR2	McBSP2 serial port control register	
01A4 000C	RCR2	McBSP2 receive control register	
01A4 0010	XCR2	McBSP2 transmit control register	
01A4 0014	SRGR2	McBSP2 sample rate generator register	
01A4 0018	MCR2	McBSP2 multichannel control register	
01A4 001C	RCERE02	McBSP2 enhanced receive channel enable register 0	
01A4 0020	XCERE02	McBSP2 enhanced transmit channel enable register 0	
01A4 0024	PCR2	McBSP2 pin control register	
01A4 0028	RCERE12	McBSP2 enhanced receive channel enable register 1	
01A4 002C	XCERE12	McBSP2 enhanced transmit channel enable register 1	
01A4 0030	RCERE22	McBSP2 enhanced receive channel enable register 2	
01A4 0034	XCERE22	McBSP2 enhanced transmit channel enable register 2	
01A4 0038	RCERE32	McBSP2 enhanced receive channel enable register 3	
01A4 003C	XCERE32	McBSP2 enhanced transmit channel enable register 3	
01A4 0040 − 01A7 FFFF	–	Reserved	

peripheral register descriptions (continued)

Table 14. Timer 0 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0194 0000	CTL0	Timer 0 control register	Determines the operating mode of the timer, monitors the timer status, and controls the function of the TOUT pin.
0194 0004	PRD0	Timer 0 period register	Contains the number of timer input clock cycles to count. This number controls the TSTAT signal frequency.
0194 0008	CNT0	Timer 0 counter register	Contains the current value of the incrementing counter.
0194 000C − 0197 FFFF	−	Reserved	

Table 15. Timer 1 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0198 0000	CTL1	Timer 1 control register	Determines the operating mode of the timer, monitors the timer status, and controls the function of the TOUT pin.
0198 0004	PRD1	Timer 1 period register	Contains the number of timer input clock cycles to count. This number controls the TSTAT signal frequency.
0198 0008	CNT1	Timer 1 counter register	Contains the current value of the incrementing counter.
0198 000C − 019B FFFF	−	Reserved	

Table 16. Timer 2 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
01AC 0000	CTL2	Timer 2 control register	Determines the operating mode of the timer, monitors the timer status, and controls the function of the TOUT pin.
01AC 0004	PRD2	Timer 2 period register	Contains the number of timer input clock cycles to count. This number controls the TSTAT signal frequency.
01AC 0008	CNT2	Timer 2 counter register	Contains the current value of the incrementing counter.
01AC 000C − 01AF FFFF	−	Reserved	

peripheral register descriptions (continued)

Table 17. HPI Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
−	HPID	HPI data register	Host read/write access only
0188 0000	HPIC	HPI control register	HPIC has both Host/CPU read/write access
0188 0004	HPIA (HPIAW)†	HPI address register (Write)	HPIA has both Host/CPU read/write access
0188 0008	HPIA (HPIAR)†	HPI address register (Read)	
0188 000C − 0189 FFFF	−	Reserved	
018A 0000	TRCTL	HPI transfer request control register	
018A 0004 − 018B FFFF	−	Reserved	

† Host access to the HPIA register updates both the HPIAW and HPIAR registers. The CPU can access HPIAW and HPIAR independently.

Table 18. GPIO Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
01B0 0000	GPEN	GPIO enable register
01B0 0004	GPDIR	GPIO direction register
01B0 0008	GPVAL	GPIO value register
01B0 000C	−	Reserved
01B0 0010	GPDH	GPIO delta high register
01B0 0014	GPHM	GPIO high mask register
01B0 0018	GPDL	GPIO delta low register
01B0 001C	GPLM	GPIO low mask register
01B0 0020	GPGC	GPIO global control register
01B0 0024	GPPOL	GPIO interrupt polarity register
01B0 0028 − 01B0 01FF	−	Reserved
01B0 0200	DEVICE_REV	Silicon Revision Identification Register (For more details, see the device characteristics listed in Table 1.)
01B0 0204 − 01B3 FFFF	−	Reserved

peripheral register descriptions (continued)

Table 19. PCI Peripheral Registers (C6415 and C6416 Only)†

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
01C0 0000	RSTSRC	DSP Reset source/status register
01C0 0004	−	Reserved
01C0 0008	PCIIS	PCI interrupt source register
01C0 000C	PCIIEN	PCI interrupt enable register
01C0 0010	DSPMA	DSP master address register
01C0 0014	PCIMA	PCI master address register
01C0 0018	PCIMC	PCI master control register
01C0 001C	CDSPA	Current DSP address register
01C0 0020	CPCIA	Current PCI address register
01C0 0024	CCNT	Current byte count register
01C0 0028	−	Reserved
01C0 002C − 01C1 FFEF	–	Reserved
0x01C1 FFF0	HSR	Host status register
0x01C1 FFF4	HDCR	Host-to-DSP control register
0x01C1 FFF8	DSPP	DSP page register
0x01C1 FFFC	−	Reserved
01C2 0000	EEADD	EEPROM address register
01C2 0004	EEDAT	EEPROM data register
01C2 0008	EECTL	EEPROM control register
01C2 000C − 01C2 FFFF	–	Reserved
01C3 0000	TRCTL	PCI transfer request control register
01C3 0004 − 01C3 FFFF	–	Reserved

† These PCI registers are *not* supported on the C6414 device.

peripheral register descriptions (continued)

Table 20. UTOPIA (C6415 and C6416 Only)†

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
01B4 0000	UCR	UTOPIA control register
01B4 0004	−	Reserved
01B4 0008	−	Reserved
01B4 000C	UIER	UTOPIA interrupt enable register
01B4 0010	UIPR	UTOPIA interrupt pending register
01B4 0014	CDR	Clock detect register
01B4 0018	EIER	Error interrupt enable register
01B4 001C	EIPR	Error interrupt pending register
01B4 0020 − 01B7 FFFF	−	Reserved

† These UTOPIA registers are *not* supported on the C6414 device.

Table 21. UTOPIA QUEUES (C6415 and C6416 Only)†

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
3C00 0000	URQ	UTOPIA receive queue
3D00 0000	UXQ	UTOPIA transmit queue
3D00 0004 − 3FFF FFFF	−	Reserved

† These UTOPIA registers are *not* supported on the C6414 device.

peripheral register descriptions (continued)

Table 22. VCP Registers (C6416 Only)†

EDMA BUS HEX ADDRESS RANGE	PERIPHERAL BUS HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
5000 0000	01B8 0000	VCPIC0	VCP input configuration register 0
5000 0004	01B8 0004	VCPIC1	VCP input configuration register 1
5000 0008	01B8 0008	VCPIC2	VCP input configuration register 2
5000 000C	01B8 000C	VCPIC3	VCP input configuration register 3
5000 0010	01B8 0010	VCPIC4	VCP input configuration register 4
5000 0014	01B8 0014	VCPIC5	VCP input configuration register 5
5000 0040	01B8 0024	VCPOUT0	VCP output register 0
5000 0044	01B8 0028	VCPOUT1	VCP output register 1
5000 0080	−	VCPWBM	VCP branch metrics write register
5000 0088	−	VCPRDECS	VCP decisions read register
−	01B8 0018	VCPEXE	VCP execution register
−	01B8 0020	VCPEND	VCP endian register
−	01B8 0040	VCPSTAT0	VCP status register 0
−	01B8 0044	VCPSTAT1	VCP status register 1
−	01B8 0050	VCPERR	VCP error register

† These VCP registers are supported on the C6416 device only.

Table 23. TCP Registers (C6416 Only)‡

EDMA BUS HEX ADDRESS RANGE	PERIPHERAL BUS HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
5800 0000	01BA 0000	TCPIC0	TCP input configuration register 0
5800 0004	01BA 0004	TCPIC1	TCP input configuration register 1
5800 0008	01BA 0008	TCPIC2	TCP input configuration register 2
5800 000C	01BA 000C	TCPIC3	TCP input configuration register 3
5800 0010	01BA 0010	TCPIC4	TCP input configuration register 4
5800 0014	01BA 0014	TCPIC5	TCP input configuration register 5
5800 0018	01BA 0018	TCPIC6	TCP input configuration register 6
5800 001C	01BA 001C	TCPIC7	TCP input configuration register 7
5800 0020	01BA 0020	TCPIC8	TCP input configuration register 8
5800 0024	01BA 0024	TCPIC9	TCP input configuration register 9
5800 0028	01BA 0028	TCPIC10	TCP input configuration register 10
5800 002C	01BA 002C	TCPIC11	TCP input configuration register 11
5800 0030	01BA 0030	TCPOUT	TCP output parameters register
5802 0000	−	TCPSP	TCP systematics and parities memory
5804 0000	−	TCPEXT	TCP extrinsic memory
5806 0000	−	TCPAP	TCP apriori memory
5808 0000	−	TCPINTER	TCP interleaver memory
580A 0000	−	TCPHD	TCP hard decisions memory
−	01BA 0038	TCPEXE	TCP execution register
−	01BA 0040	TCPEND	TCP endian register
−	01BA 0050	TCPERR	TCP error register
−	01BA 0058	TCPSTAT	TCP status register

‡ These TCP registers are supported on the C6416 device only.

EDMA channel synchronization events

The C64x EDMA supports up to 64 EDMA channels which service peripheral devices and external memory. Table 24 lists the source of C64x EDMA synchronization events associated with each of the programmable EDMA channels. For the C64x device, the association of an event to a channel is fixed; each of the EDMA channels has one specific event associated with it. These specific events are captured in the EDMA event registers (ERL, ERH) even if the events are disabled by the EDMA event enable registers (EERL, EERH). The priority of each event can be specified independently in the transfer parameters stored in the EDMA parameter RAM. For more detailed information on the EDMA module and how EDMA events are enabled, captured, processed, linked, chained, and cleared, etc., see the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (literature number SPRU234).

EDMA channel synchronization events (continued)

Table 24. SMJ320C64x EDMA Channel Synchronization Events†

EDMA CHANNEL	EVENT NAME	EVENT DESCRIPTION
0	DSP_INT	HPI/PCI-to-DSP interrupt (PCI peripheral supported on C6415 and C6416 only)‡
1	TINT0	Timer 0 interrupt
2	TINT1	Timer 1 interrupt
3	SD_INTA	EMIFA SDRAM timer interrupt
4	GPINT4/EXT_INT4	GPIO event 4/External interrupt pin 4
5	GPINT5/EXT_INT5	GPIO event 5/External interrupt pin 5
6	GPINT6/EXT_INT6	GPIO event 6/External interrupt pin 6
7	GPINT7/EXT_INT7	GPIO event 7/External interrupt pin 7
8	GPINT0	GPIO event 0
9	GPINT1	GPIO event 1
10	GPINT2	GPIO event 2
11	GPINT3	GPIO event 3
12	XEVT0	McBSP0 transmit event
13	REVT0	McBSP0 receive event
14	XEVT1	McBSP1 transmit event
15	REVT1	McBSP1 receive event
16	–	None
17	XEVT2	McBSP2 transmit event
18	REVT2	McBSP2 receive event
19	TINT2	Timer 2 interrupt
20	SD_INTB	EMIFB SDRAM timer interrupt
21	–	Reserved, for future expansion
22−27	–	None
28	VCPREVT	VCP receive event (C6416 only)§
29	VCPXEVT	VCP transmit event (C6416 only)§
30	TCPREVT	TCP receive event (C6416 only)§
31	TCPXEVT	TCP transmit event (C6416 only)§
32	UREVT	UTOPIA receive event (C6415 and C6416 only)‡
33−39	–	None
40	UXEVT	UTOPIA transmit event (C6415 and C6416 only)‡
41−47	–	None
48	GPINT8	GPIO event 8
49	GPINT9	GPIO event 9
50	GPINT10	GPIO event 10
51	GPINT11	GPIO event 11
52	GPINT12	GPIO event 12
53	GPINT13	GPIO event 13
54	GPINT14	GPIO event 14
55	GPINT15	GPIO event 15
56−63	–	None

† In addition to the events shown in this table, each of the 64 channels can also be synchronized with the transfer completion or alternate transfer completion events. For more detailed information on EDMA event-transfer chaining, see the *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide* (literature number SPRU234).

‡ The PCI and UTOPIA peripherals are not supported on the C6414 device; therefore, these EDMA synchronization events are reserved.

§ The VCP/TCP EDMA synchronization events are supported on the C6416 only. For the C6414 and C6415 devices, these events are reserved.

interrupt sources and interrupt selector

The C64x DSP core supports 16 prioritized interrupts, which are listed in Table 25. The highest-priority interrupt is INT_00 (dedicated to RESET) while the lowest-priority interrupt is INT_15. The first four interrupts (INT_00−INT_03) are non-maskable and fixed. The remaining interrupts (INT_04−INT_15) are maskable and default to the interrupt source specified in Table 25. The interrupt source for interrupts 4−15 can be programmed by modifying the selector value (binary value) in the corresponding fields of the Interrupt Selector Control registers: MUXH (address 0x019C0000) and MUXL (address 0x019C0004).

interrupt sources and interrupt selector (continued)

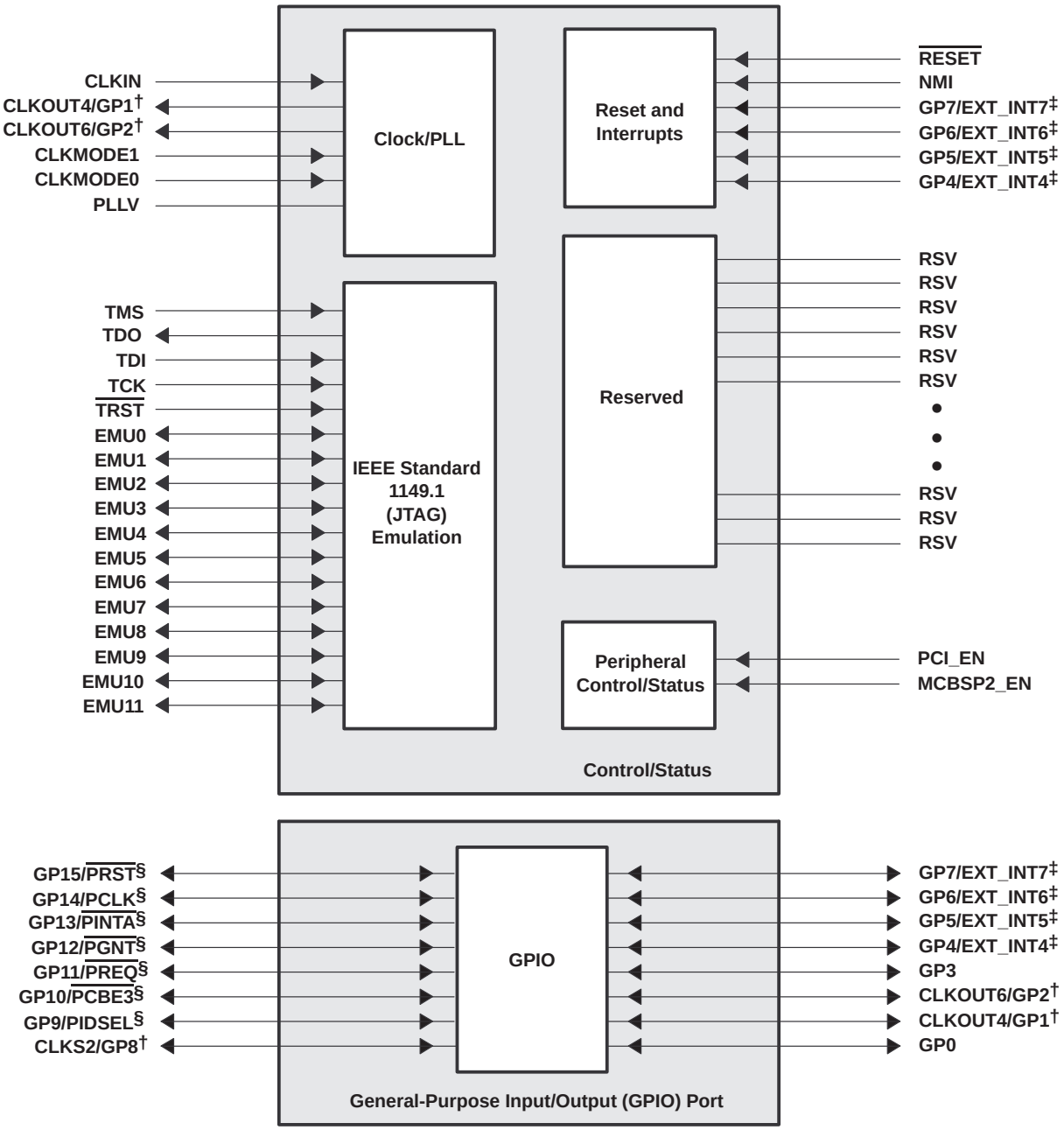
Table 25. C64x DSP Interrupts

CPU INTERRUPT NUMBER	INTERRUPT SELECTOR CONTROL REGISTER	SELECTOR VALUE (BINARY)	INTERRUPT EVENT	INTERRUPT SOURCE
INT_00†	−	−	RESET	
INT_01†	−	−	NMI	
INT_02†	−	−	Reserved	Reserved. Do not use.
INT_03†	−	−	Reserved	Reserved. Do not use.
INT_04‡	MUXL[4:0]	00100	GPINT4/EXT_INT4	GPIO interrupt 4/External interrupt pin 4
INT_05‡	MUXL[9:5]	00101	GPINT5/EXT_INT5	GPIO interrupt 5/External interrupt pin 5
INT_06‡	MUXL[14:10]	00110	GPINT6/EXT_INT6	GPIO interrupt 6/External interrupt pin 6
INT_07‡	MUXL[20:16]	00111	GPINT7/EXT_INT7	GPIO interrupt 7/External interrupt pin 7
INT_08‡	MUXL[25:21]	01000	EDMA_INT	EDMA channel (0 through 63) interrupt
INT_09‡	MUXL[30:26]	01001	EMU_DTDMA	EMU DTDMA
INT_10‡	MUXH[4:0]	00011	SD_INTA	EMIFA SDRAM timer interrupt
INT_11‡	MUXH[9:5]	01010	EMU_RTDXRX	EMU real-time data exchange (RTDX) receive
INT_12‡	MUXH[14:10]	01011	EMU_RTDXTX	EMU RTDX transmit
INT_13‡	MUXH[20:16]	00000	DSP_INT	HPI/PCI-to-DSP interrupt (PCI supported on C6415 and C6416 only)
INT_14‡	MUXH[25:21]	00001	TINT0	Timer 0 interrupt
INT_15‡	MUXH[30:26]	00010	TINT1	Timer 1 interrupt
−	−	01100	XINT0	McBSP0 transmit interrupt
−	−	01101	RINT0	McBSP0 receive interrupt
−	−	01110	XINT1	McBSP1 transmit interrupt
−	−	01111	RINT1	McBSP1 receive interrupt
−	−	10000	GPINT0	GPIO interrupt 0
−	−	10001	XINT2	McBSP2 transmit interrupt
−	−	10010	RINT2	McBSP2 receive interrupt
−	−	10011	TINT2	Timer 2 interrupt
−	−	10100	SD_INTB	EMIFB SDRAM timer interrupt
−	−	10101	Reserved	Reserved. Do not use.
−	−	10110	Reserved	Reserved. Do not use.
−	−	10111	UINT	UTOPIA interrupt (C6415/C6416 only)
−	−	11000 − 11101	Reserved	Reserved. Do not use.
−	−	11110	VCPINT	VCP interrupt (C6416 only)
−	−	11111	TCPINT	TCP interrupt (C6416 only)

† Interrupts INT_00 through INT_03 are non-maskable and fixed.

‡ Interrupts INT_04 through INT_15 are programmable by modifying the binary selector values in the Interrupt Selector Control registers fields. Table 25 shows the default interrupt sources for Interrupts INT_04 through INT_15. For more detailed information on interrupt sources and selection, see the *TMS320C6000 DSP Interrupt Selector Reference Guide* (literature number SPRU646).

signal groups description



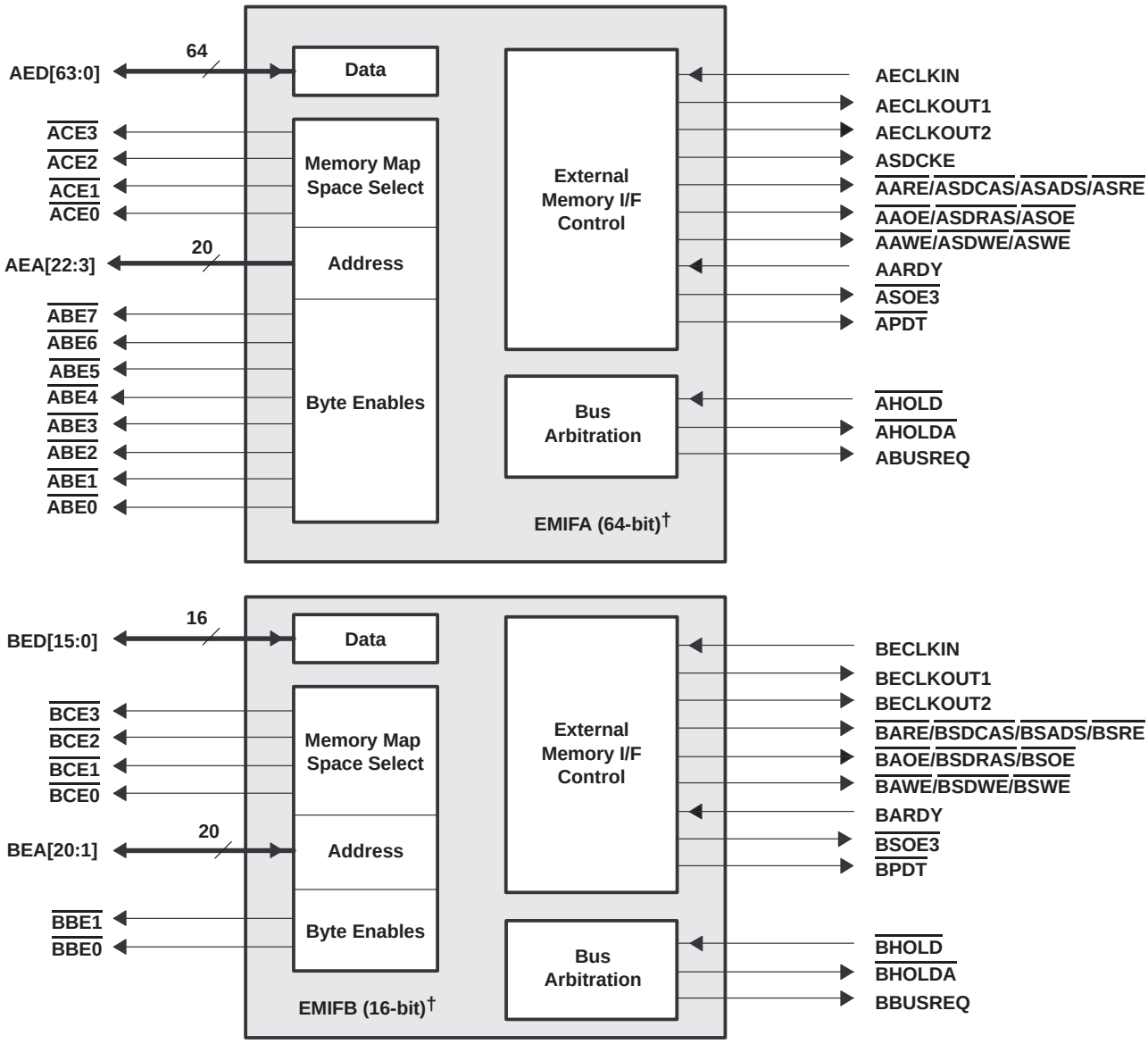
† These pins are MUXed with the GPIO port pins and by default these signals function as clocks (CLKOUT4 or CLKOUT6) or McBSP2 clock source (CLKS2). To use these MUXed pins as GPIO signals, the appropriate GPIO register bits (GPxEN and GPxDIR) must be properly enabled and configured. For more details, see the Device Configurations section of this data sheet.

‡ These pins are GPIO pins that can also function as external interrupt sources (EXT_INT[7:4]). Default after reset is EXT_INTx or GPIO as input-only.

§ For the C6415 and C6416 devices, these GPIO pins are MUXed with the PCI peripheral pins. By default, these signals are set up to no function with both the GPIO and PCI pin functions *disabled*. For more details on these MUXed pins, see the Device Configurations section of this data sheet. For the C6414 device, the GPIO peripheral pins are *not* MUXed; the C6414 device does *not* support the PCI peripheral.

Figure 2. CPU and Peripheral Signals

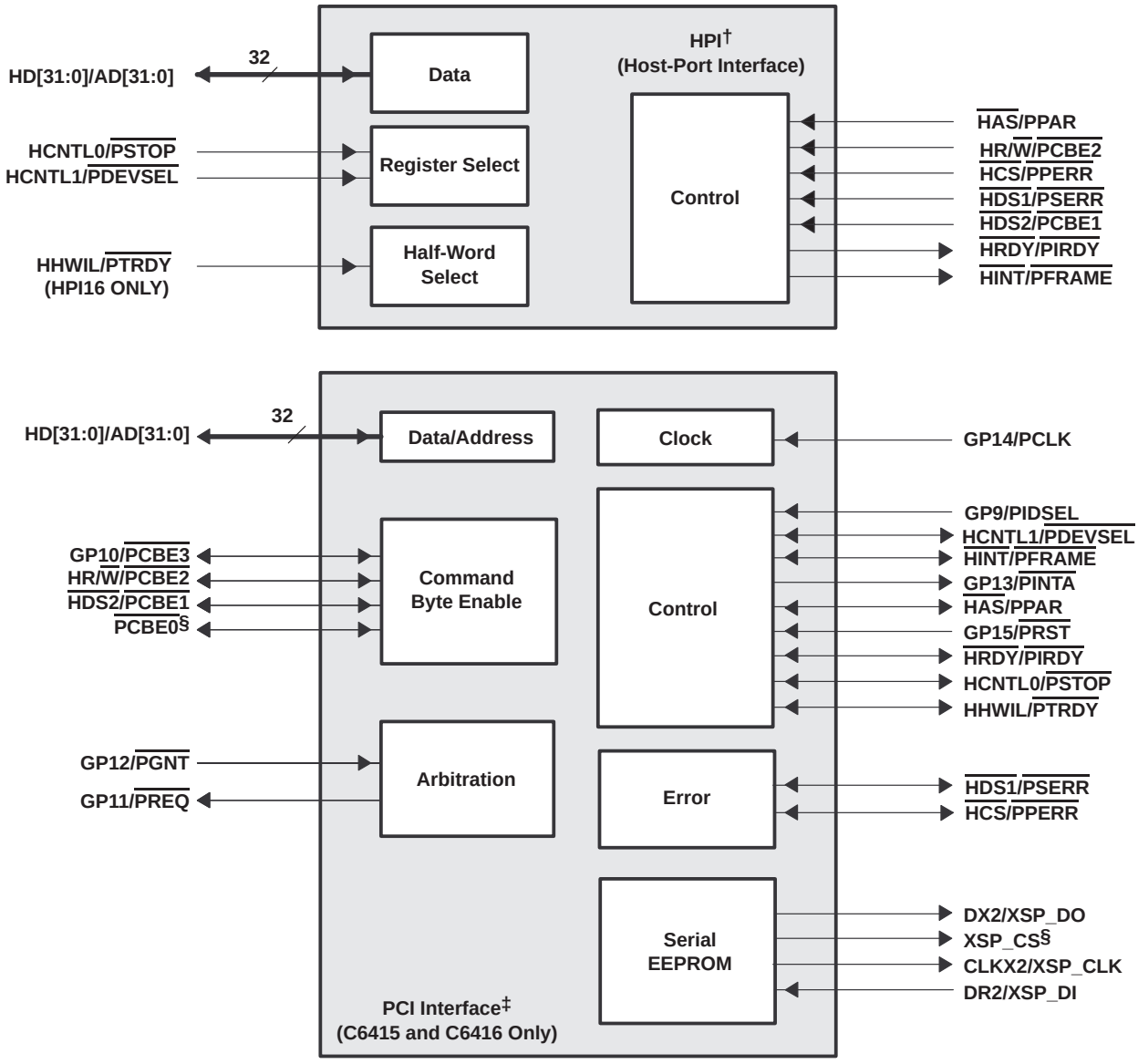
signal groups description (continued)



† These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

Figure 3. Peripheral Signals

signal groups description (continued)



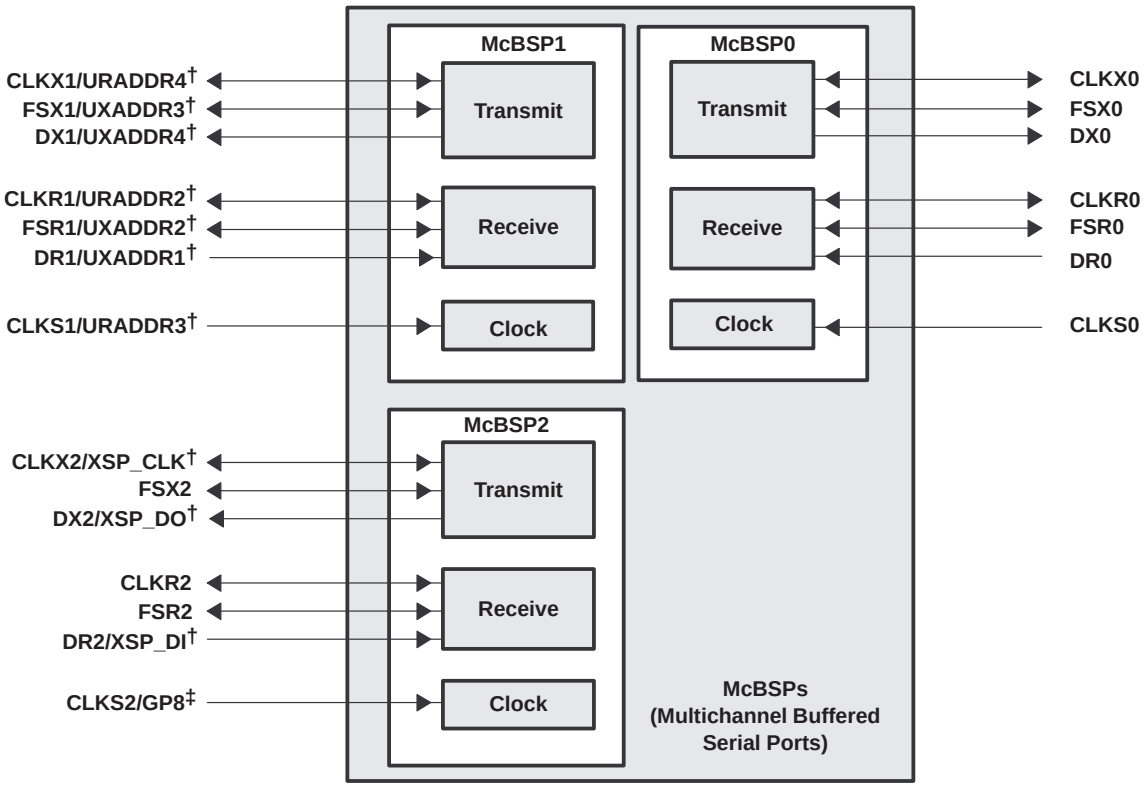
† For the C6415 and C6416 devices, these HPI pins are MUXed with the PCI peripheral. By default, these signals function as HPI. For more details on these MUXed pins, see the Device Configurations section of this data sheet. For the C6414 device, these HPI pins are *not* MUXed; the C6414 device does *not* support the PCI peripheral.

‡ For the C6415 and C6416 devices, these PCI pins (excluding PCBE0 and XSP_CS) are MUXed with the HPI, McBSP2, or GPIO peripherals. By default, these signals function as HPI, McBSP2, and no function, respectively. For more details on these MUXed pins, see the Device Configurations section of this data sheet. For the C6414 device, the HPI, McBSP2, and GPIO peripheral pins are *not* MUXed; the C6414 device does *not* support the PCI peripheral.

§ For the C6414 device, these pins are "Reserved (leave unconnected, ***do not*** connect to power or ground)."

Figure 3. Peripheral Signals (Continued)

signal groups description (continued)

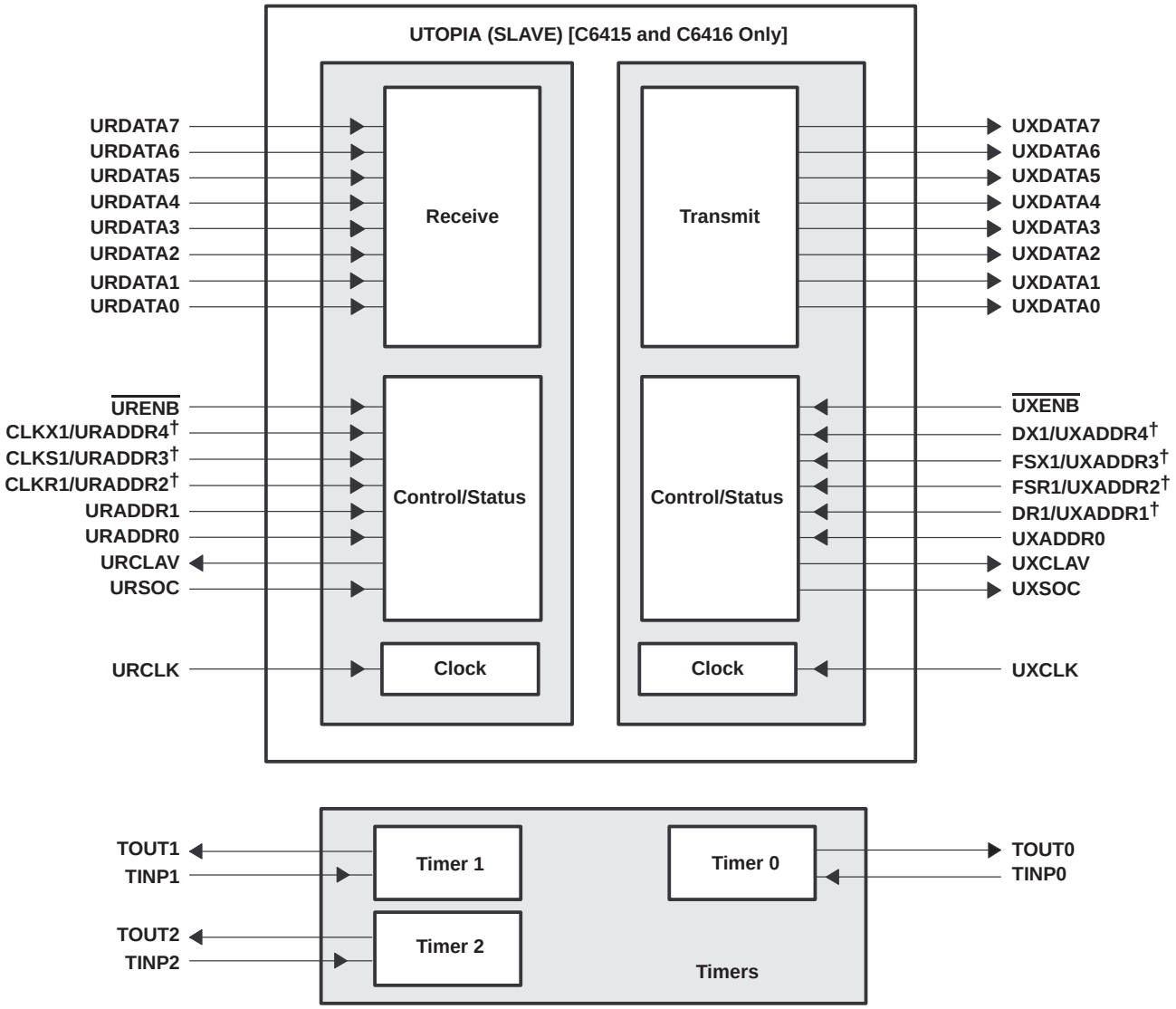


† For the C6415 and C6416 devices, these McBSP2 and McBSP1 pins are MUXed with the PCI and UTOPIA peripherals, respectively. By default, these signals function as McBSP2 and McBSP1, respectively. For more details on these MUXed pins, see the Device Configurations section of this data sheet.
For the C6414 device, these McBSP2 and McBSP1 peripheral pins are *not* MUXed; the C6414 device does *not* support PCI and UTOPIA peripherals.

‡ The McBSP2 clock source pin (CLKS2, default) is MUXed with the GP8 pin. To use this MUXed pin as the GP8 signal, the appropriate GPIO register bits (GP8EN and GP8DIR) must be properly enabled and configured. For more details, see the Device Configurations section of this data sheet.

Figure 3. Peripheral Signals (Continued)

signal groups description (continued)



† For the C6415 and C6416 devices, these UTOPIA pins are MUXed with the McBSP1 peripheral. By default, these signals function as McBSP1. For more details on these MUXed pins, see the Device Configurations section of this data sheet.
For the C6414 device, these McBSP1 peripheral pins are *not* MUXed; the C6414 does *not* support the UTOPIA peripheral.

Figure 3. Peripheral Signals (Continued)

DEVICE CONFIGURATIONS

The C6414, C6415, and C6416 peripheral selections and other device configurations are determined by external pullup/pulldown resistors on the following pins (all of which are latched during device reset):

- peripherals selection (C6415 and C6416 devices)
  - BEA11 (UTOPIA_EN)
  - PCI_EN (for C6415 or C6416, see Table 27 footnotes)
  - MCBSP2_EN (for C6414 or C6416, see Table 27 footnotes)

  The C6414 device does *not* support the PCI and UTOPIA peripherals; for proper operation of the C6414 device, do ***not*** oppose the internal *pulldowns* (IPDs) on the BEA11, PCI_EN, and MCBSP2_EN pins. (For IPUs/IPDs on pins, see the Terminal Functions table of this data sheet.)
- other device configurations (C64x)
  - BEA[20:13, 7]
  - HD5

peripherals selection

Some C6415/C6416 peripherals share the same pins (internally MUXed) and are mutually exclusive (i.e., HPI, general-purpose input/output pins GP[15:9], PCI and its internal EEPROM, McBSP1, McBSP2, and UTOPIA). The VCP/TCP coprocessors (C6416 only) and other C64x peripherals (i.e., the Timers, McBSP0, and the GP[8:0] pins), are always available.

- UTOPIA and McBSP1 peripherals

  The UTOPIA_EN pin (BEA11) is latched at reset. For C6415 and C6416 devices, this pin selects whether the UTOPIA peripheral or McBSP1 peripheral is functionally enabled (see Table 26).

  The C6414 device does *not* support the UTOPIA peripheral; for proper device operation, do ***not*** oppose the internal *pulldown* (IPD) on the BEA11 pin.

Table 26. UTOPIA_EN Peripheral Selection (McBSP1 and UTOPIA) (C6415/C6416 Only)

PERIPHERAL SELECTION	PERIPHERALS SELECTED		DESCRIPTION
UTOPIA_EN (BEA11) Pin [F14]	UTOPIA	McBSP1	
0		√	McBSP1 is enabled and UTOPIA is disabled [default]. This means all multiplexed McBSP1/UTOPIA pins function as McBSP1 and all other standalone UTOPIA pins are tied-off (Hi-Z).
1	√		UTOPIA is enabled and McBSP1 is disabled. This means all multiplexed McBSP1/UTOPIA pins now function as UTOPIA and all other standalone McBSP1 pins are tied-off (Hi-Z).

- HPI, GP[15:9], PCI, EEPROM (internal to PCI), and McBSP2 peripherals

  The PCI_EN and MCBSP2_EN pins are latched at reset. They determine specific peripheral selection for the C6415 and C6416 devices, summarized in Table 27.

  The C6414 device does *not* support the PCI peripheral; for proper device operation, do ***not*** oppose the internal pulldowns (IPDs) on the PCI_EN and MCBSP2_EN pins.

DEVICE CONFIGURATIONS (CONTINUED)

Table 27. PCI_EN and MCBSP2_EN Peripheral Selection (HPI, GP[15:9], PCI, and McBSP2)

PERIPHERAL SELECTION†		PERIPHERALS SELECTED				
PCI_EN Pin [T8]	MCBSP2_EN Pin [AB4]	HPI	GP[15:9]	PCI	EEPROM (Internal to PCI)	McBSP2
0	0	√	√			√
0	1	√	√			√
1	0			√	√	‡
1	1			√		√

† The PCI_EN pin *must* be driven valid at all times and the user *must not* switch values throughout device operation.
The MCBSP2_EN pin *must* be driven valid at all times and the user *can* switch values throughout device operation.

‡ The only time McBSP2 is disabled is when both PCI_EN = 1 and MCBSP2_EN = 0. This configuration enables, at reset, the auto-initialization of the PCI peripheral through the PCI internal EEPROM [provided the PCI EEPROM Auto-Initialization pin (BEA13) is pulled up (EEAI = 1)]. The user can then enable the McBSP2 peripheral (disabling EEPROM) by dynamically changing MCBSP2_EN to a “1” after the device is initialized (out of reset).

- If the PCI is disabled (PCI_EN = 0), the HPI peripheral is enabled and GP[15:9] pins can be programmed as GPIO, provided the GPxEN and GPxDIR bits are properly configured.

  This means all multiplexed HPI/PCI pins function as HPI and all standalone PCI pins ($\overline{\text{PCBE0}}$ and XSP_CS) are tied-off (Hi-Z). Also, the multiplexed GPIO/PCI pins can be used as GPIO with the proper software configuration of the GPIO enable and direction registers (for more details, see Table 29).

- If the PCI is enabled (PCI_EN = 1), the HPI peripheral is disabled.

  This means all multiplexed HPI/PCI pins function as PCI. Also, the multiplexed GPIO/PCI pins function as PCI pins (for more details, see Table 29).

- The MCBSP2_EN pin, in combination with the PCI_EN pin, controls the selection of the McBSP2 peripheral and the PCI internal EEPROM (for more details, see Table 27 and its footnotes).

other device configurations

Table 28 describes the C6414, C6415, and C6416 devices configuration pins, which are set up via external pullup/pulldown resistors through the specified EMIFB address bus pins (BEA[20:13, 11, 9:7]) and the HD5 pin. For more details on these device configuration pins, see the Terminal Functions table and the Debugging Considerations section.

DEVICE CONFIGURATIONS (CONTINUED)

Table 28. Device Configuration Pins (BEA[20:13, 9:7], HD5, and BEA11)

CONFIGURATION PIN	NO.	FUNCTIONAL DESCRIPTION
BEA20	E15	Device Endian mode (LEND) 0 – System operates in Big Endian mode 1 – System operates in Little Endian mode (default)
BEA[19:18]	[D17, J14]	Bootmode [1:0] 00 – No boot 01 – HPI boot 10 – EMIFB 8-bit ROM boot with default timings (default mode) 11 – Reserved
BEA[17:16]	[E16, G15]	EMIFA input clock select Clock mode select for EMIFA (AECLKIN_SEL[1:0]) 00 – AECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved
BEA[15:14]	[C18, F15]	EMIFB input clock select Clock mode select for EMIFB (BECLKIN_SEL[1:0]) 00 – BECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved
BEA13	D16	PCI EEPROM Auto-Initialization (EEAI) [C6415 and C6416 devices only] [The **C6414** device does *not* support the PCI peripheral; for proper device operation, ***do not*** oppose the internal pulldown (IPD) on the BEA13 pin.] PCI auto-initialization via external EEPROM 0 – PCI auto-initialization through EEPROM is disabled; the PCI peripheral uses the specified PCI default values (default). 1 – PCI auto-initialization through EEPROM is enabled; the PCI peripheral is configured through EEPROM provided the PCI peripheral pin is enabled (PCI_EN = 1) and the McBSP2 peripheral pin is disabled (MCBSP2_EN = 0). Note: If the PCI peripheral is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up. For more information on the PCI EEPROM default values, see the *TMS320C6000 DSP Peripheral Component Interconnect (PCI) Reference Guide* (literature number SPRU581).
BEA11	F14	UTOPIA Enable (UTOPIA_EN) [C6415 and C6416 devices only] [The **C6414** device does *not* support the UTOPIA peripheral; for proper device operation, ***do not*** oppose the internal pulldown (IPD) on the BEA11 pin.] UTOPIA peripheral enable (functional) 0 – UTOPIA peripheral disabled (McBSP1 functions are enabled). [default] This means all multiplexed McBSP1/UTOPIA pins function as McBSP1 and all other standalone UTOPIA pins are tied-off (Hi-Z). 1 – UTOPIA peripheral enabled (McBSP1 functions are disabled). This means all multiplexed McBSP1/UTOPIA pins now function as UTOPIA and all other standalone McBSP1 pins are tied-off (Hi-Z).

DEVICE CONFIGURATIONS (CONTINUED)

Table 28. Device Configuration Pins (BEA[20:13, 9:7], HD5, and BEA11) (Continued)

CONFIGURATION PIN	NO.	FUNCTIONAL DESCRIPTION		
		C6414 Devices	**C6415 Devices**	**C6416 Devices**
BEA7	E14	**Do not** oppose internal pulldown (IPD)	Pullup†	**Do not** oppose IPD
BEA8	G14	**Do not** oppose IPD	**Do not** oppose IPD	Pullup†
BEA9	G13	**Do not** oppose IPD	**Do not** oppose IPD	Pullup†
		†For proper device operation, this pin must be externally pulled up with a 1-kΩ resistor.		
HD5	U4	HPI peripheral bus width (HPI_WIDTH) 0 − HPI operates as an HPI16. (HPI bus is 16 bits wide. HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the Hi-Z state.) 1 − HPI operates as an HPI32. (HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.)		

DEVICE CONFIGURATIONS (CONTINUED)

multiplexed pins

Multiplexed pins are pins that are shared by more than one peripheral and are internally multiplexed. Some of these pins are configured by software, and the others are configured by external pullup/pulldown resistors only at reset. Those MUXed pins that are configured by software can be programmed to switch functionalities at any time. Those MUXed pins that are configured by external pullup/pulldown resistors are mutually exclusive; only one peripheral has primary control of the function of these pins after reset. Table 29 identifies the multiplexed pins on the C6414, C6415, and C6416 devices; shows the default (primary) function and the default settings after reset; and describes the pins, registers, etc. necessary to configure specific multiplexed functions.

debugging considerations

It is recommended that external connections be provided to device configuration pins, including CLKMODE[1:0], BEA[20:13, 11, 9:7], HD5/AD5, PCI_EN, and MCBSP2_EN. Although internal pullup/pulldown resistors exist on these pins (except for HD5/AD5), providing external connectivity adds convenience to the user in debugging and flexibility in switching operating modes.

Internal pullup/pulldown resistors also exist on the non-configuration pins on the BEA bus (BEA[12, 10, 6:1]). Do not oppose the internal pullup/pulldown resistors on these non-configuration pins with external pullup/pulldown resistors. If an external controller provides signals to these non-configuration pins, these signals must be driven to the default state of the pins at reset, or not be driven at all.

For the internal pullup/pulldown resistors on the C6414, C6415, and C6416 device pins, see the terminal functions table.

DEVICE CONFIGURATIONS (CONTINUED)

Table 29. C6414, C6415, and C6416 Device Multiplexed Pins†

MULTIPLEXED PINS NAME	MULTIPLEXED PINS NO.	DEFAULT FUNCTION	DEFAULT SETTING	DESCRIPTION
CLKOUT4/GP1‡	Y7	CLKOUT4	GP1EN = 0 (disabled)	These pins are software-configurable. To use these pins as GPIO pins, the GPxEN bits in the GPIO Enable Register and the GPxDIR bits in the GPIO Direction Register must be properly configured. GPxEN = 1: GPx pin enabled GPxDIR = 0: GPx pin is an input GPxDIR = 1: GPx pin is an output
CLKOUT6/GP2‡	T10	CLKOUT6	GP2EN = 0 (disabled)	
CLKS2/GP8‡	W7	CLKS2	GP8EN = 0 (disabled)	
GP9/PIDSEL	K6	None	GPxEN = 0 (disabled) PCI_EN = 0 (disabled)†	To use GP[15:9] as GPIO pins, the PCI needs to be disabled (PCI_EN = 0), the GPxEN bits in the GPIO Enable Register and the GPxDIR bits in the GPIO Direction Register must be properly configured. GPxEN = 1: GPx pin enabled GPxDIR = 0: GPx pin is an input GPxDIR = 1: GPx pin is an output
GP10/$\overline{\text{PCBE3}}$	L7			
GP11/$\overline{\text{PREQ}}$	H6			
GP12/$\overline{\text{PGNT}}$	J7			
GP13/$\overline{\text{PINTA}}$	G4			
GP14/PCLK	G5			
GP15/$\overline{\text{PRST}}$	J8			
DX1/UXADDR4	Y10	DX1	UTOPIA_EN (BEA11) = 0 (disabled)†	By default, McBSP1 is enabled upon reset (UTOPIA is disabled). To enable the UTOPIA peripheral, an external pullup resistor (1 kΩ) must be provided on the BEA11 pin (setting UTOPIA_EN = 1 at reset).
FSX1/UXADDR3	AA11	FSX1		
FSR1/UXADDR2	AA8	FSR1		
DR1/UXADDR1	V11	DR1		
CLKX1/URADDR4	T12	CLKX1		
CLKS1/URADDR3	Y8	CLKS1		
CLKR1/URADDR2	AB7	CLKR1		
CLKX2/XSP_CLK	W5	CLKX2	PCI_EN = 0 (disabled)†	By default, HPI is enabled upon reset (PCI is disabled). To enable the PCI peripheral an external pullup resistor (1 kΩ) must be provided on the PCI_EN pin (setting PCI_EN = 1 at reset).
DR2/XSP_DI	Y4	DR2		
DX2/XSP_DO	R9	DX2		
HD[31:0]/AD[31:0]	§	HD[31:0]		
$\overline{\text{HAS}}$/PPAR	N7	$\overline{\text{HAS}}$		
HCNTL1/$\overline{\text{PDEVSEL}}$	N8	HCNTL1		
HCNTL0/$\overline{\text{PSTOP}}$	P5	HCNTL0		
$\overline{\text{HDS1}}$/$\overline{\text{PSERR}}$	R5	$\overline{\text{HDS1}}$		
$\overline{\text{HDS2}}$/$\overline{\text{PCBE1}}$	P6	$\overline{\text{HDS2}}$		
HR/$\overline{\text{W}}$/$\overline{\text{PCBE2}}$	N6	HR/$\overline{\text{W}}$		
HWWIL/$\overline{\text{PTRDY}}$	N5	HHWIL (HPI16 only)		
$\overline{\text{HINT}}$/$\overline{\text{PFRAME}}$	P4	$\overline{\text{HINT}}$		
$\overline{\text{HCS}}$/$\overline{\text{PPERR}}$	N9	$\overline{\text{HCS}}$		
$\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$	N4	$\overline{\text{HRDY}}$		

† For the C6415 and C6416 devices, all other standalone UTOPIA and PCI pins are tied-off internally (pins in Hi-Z) when the peripheral is disabled [UTOPIA_EN (BEA11) = 0 or PCI_EN = 0].

‡ The C6414 device does not support the PCI and UTOPIA peripherals. These are the ***only*** multiplexed pins on the C6414 device, all other pins are standalone peripheral functions and are not MUXed.

§ For the HD[31:0]/AD[31:0] multiplexed pins pin numbers, see the *Terminal Functions* table.

Terminal Functions

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
CLOCK/PLL CONFIGURATION				
CLKIN	H4	I	IPD	Clock Input. This clock is the input to the on-chip PLL.
CLKOUT4/GP1§	Y7	I/O/Z	IPD	Clock output at 1/4 of the device speed (**O/Z**) [default] or this pin can be programmed as a GPIO 1 pin (**I/O/Z**).
CLKOUT6/GP2§	T10	I/O/Z	IPD	Clock output at 1/6 of the device speed (**O/Z**) [default] or this pin can be programmed as a GPIO 2 pin (**I/O/Z**).
CLKMODE1	K8	I	IPD	Clock mode select • Selects whether the CPU clock frequency = input clock frequency x1 (Bypass), x6, or x12. For more details on the CLKMODE pins and the PLL multiply factors, see the Clock PLL section of this data sheet.
CLKMODE0	E3	I	IPD	
PLLV¶	L9	A#		PLL voltage supply
JTAG EMULATION				
TMS	V14	I	IPU	JTAG test-port mode select
TDO	W16	O/Z	IPU	JTAG test-port data out
TDI	AA17	I	IPU	JTAG test-port data in
TCK	Y15	I	IPU	JTAG test-port clock
$\overline{\text{TRST}}$	Y14	I	IPD	JTAG test-port reset. For IEEE 1149.1 JTAG compatibility, see the IEEE 1149.1 JTAG Compatibility Statement section of this data sheet.
EMU11	V15	I/O/Z	IPU	Emulation pin 11. Reserved for future use, leave unconnected.
EMU10	Y16	I/O/Z	IPU	Emulation pin 10. Reserved for future use, leave unconnected.
EMU9	T14	I/O/Z	IPU	Emulation pin 9. Reserved for future use, leave unconnected.
EMU8	U14	I/O/Z	IPU	Emulation pin 8. Reserved for future use, leave unconnected.
EMU7	AB18	I/O/Z	IPU	Emulation pin 7. Reserved for future use, leave unconnected.
EMU6	AA16	I/O/Z	IPU	Emulation pin 6. Reserved for future use, leave unconnected.
EMU5	W15	I/O/Z	IPU	Emulation pin 5. Reserved for future use, leave unconnected.
EMU4	AB17	I/O/Z	IPU	Emulation pin 4. Reserved for future use, leave unconnected.
EMU3	W14	I/O/Z	IPU	Emulation pin 3. Reserved for future use, leave unconnected.
EMU2	AA15	I/O/Z	IPU	Emulation pin 2. Reserved for future use, leave unconnected.
EMU1 EMU0	T13 V13	I/O/Z	IPU	Emulation [1:0] pins • Select the device functional mode of operation EMU[1:0] Operation 00 Boundary Scan/Normal Mode (see Note) 01 Reserved 10 Reserved 11 Emulation/Normal Mode [default] (see the IEEE 1149.1 JTAG Compatibility Statement section of this data sheet) Normal mode refers to the DSPs normal operational mode, when the DSP is free running. The DSP can be placed in normal operational mode when the EMU[1:0] pins are configured for either Boundary Scan or Emulation. Note: When the EMU[1:0] pins are configured for Boundary Scan mode, the internal pulldown (IPD) on the $\overline{\text{TRST}}$ signal must ***not*** be opposed in order to operate in Normal mode. For the Boundary Scan mode pulldown EMU[1:0] pins with a dedicated 1-kΩ resister.

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)
§ These pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet.
¶ PLLV is not part of external voltage supply. See the Clock PLL section for information on how to connect this pin.
# A = Analog signal (PLL Filter)

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
RESETS, INTERRUPTS, AND GENERAL-PURPOSE INPUT/OUTPUTS				
$\overline{\text{RESET}}$	AB5	I		Device reset
NMI	E6	I	IPD	Nonmaskable interrupt, edge-driven (rising edge)
GP7/EXT_INT7	Y6	I/O/Z	IPU	General-purpose input/output (GPIO) pins (**I/O/Z**) or external interrupts (**input only**). The default after reset setting is GPIO enabled as input-only. • When these pins function as External Interrupts [by selecting the corresponding interrupt enable register bit (IER.[7:4])], they are edge-driven and the polarity can be independently selected via the External Interrupt Polarity Register bits (EXTPOL.[3:0]).
GP6/EXT_INT6	V8			
GP5/EXT_INT5	AA5			
GP4/EXT_INT4	U9			
GP15/$\overline{\text{PRST}}$§	J8	I/O/Z		General-purpose input/output (GPIO) 15 pin (**I/O/Z**) or PCI reset (**I**). No function at default.
GP14/PCLK§	G5			GPIO 14 pin (**I/O/Z**) or PCI clock (**I**). No function at default.
GP13/$\overline{\text{PINTA}}$§	G4			GPIO 13 pin (**I/O/Z**) or PCI interrupt A (**O/Z**). No function at default.
GP12/$\overline{\text{PGNT}}$§	J7			GPIO 12 pin (**I/O/Z**) or PCI bus grant (**I**). No function at default.
GP11/$\overline{\text{PREQ}}$§	H6			GPIO 11 pin (**I/O/Z**) or PCI bus request (**O/Z**). No function at default.
GP10/$\overline{\text{PCBE3}}$§	L7			GPIO 10 pin (**I/O/Z**) or PCI command/byte enable 3 (**I/O/Z**). No function at default.
GP9/PIDSEL§	K6			GPIO 9 pin (**I/O/Z**) or PCI initialization device select (**I**). No function at default.
GP3	AA6		IPD	GPIO 3 pin (**I/O/Z**). The default after reset setting is GPIO 3 enabled as input-only.
GP0	W8		IPD	GPIO 0 pin. The general-purpose I/O 0 pin (GPIO 0) (**I/O/Z**) can be programmed as GPIO 0 (**input only**) [default] or as GPIO 0 (**output only**) pin or output as a general-purpose interrupt (GP0INT) signal (**output only**).
CLKS2/GP8§¶	W7	I/O/Z	IPD	McBSP2 external clock source (CLKS2) [**input only**] [default] or this pin can be programmed as a GPIO 8 pin (**I/O/Z**).
CLKOUT6/GP2§¶	T10	I/O/Z	IPD	Clock output at 1/6 of the device speed (**O/Z**) [default] or this pin can be programmed as a GPIO 2 pin (**I/O/Z**).
CLKOUT4/GP1§¶	Y7	I/O/Z	IPD	Clock output at 1/4 of the device speed (**O/Z**) [default] or this pin can be programmed as a GPIO 1 pin (**I/O/Z**).
HOST-PORT INTERFACE (HPI) [C64x] or PERIPHERAL COMPONENT INTERCONNECT (PCI) [C6415 or C6416 devices only]				
PCI_EN	T8	I	IPD	PCI enable pin. This pin controls the selection (enable/disable) of the HPI and GP[15:9], or PCI peripherals (for the C6415 and C6416 devices). This pin works in conjunction with the MCBSP2_EN pin to enable/disable other peripherals (for more details, see the Device Configurations section of this data sheet). The C6414 device does *not* support the PCI peripheral; for proper device operation, do ***not*** oppose the internal pulldown (IPD) on this pin.
$\overline{\text{HINT}}$/$\overline{\text{PFRAME}}$§	P4	I/O/Z		Host interrupt from DSP to host (**O**) [default] or PCI frame (**I/O/Z**)
HCNTL1/ $\overline{\text{PDEVSEL}}$§	N8	I/O/Z		Host control − selects between control, address, or data registers (**I**) [default] or PCI device select (**I/O/Z**).
HCNTL0/ $\overline{\text{PSTOP}}$§	P5	I/O/Z		Host control − selects between control, address, or data registers (**I**) [default] or PCI stop (**I/O/Z**)
HHWIL/$\overline{\text{PTRDY}}$§	N5	I/O/Z		Host half-word select − first or second half-word (not necessarily high or low order) [For HPI16 bus width selection only] (**I**) [default] or PCI target ready (**I/O/Z**)
HR/$\overline{\text{W}}$/$\overline{\text{PCBE2}}$§	N6	I/O/Z		Host read or write select (**I**) [default] or PCI command/byte enable 2 (**I/O/Z**)

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins are standalone peripheral functions for this device.

¶ For the C6414 device, only these pins are multiplexed pins.

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
HOST-PORT INTERFACE (HPI) [C64x] or PERIPHERAL COMPONENT INTERCONNECT (PCI) [C6415 or C6416 devices only] (CONTINUED)				
$\overline{\text{HAS}}$/PPAR§	N7	I/O/Z		Host address strobe (**I**) [default] or PCI parity (**I/O/Z**)
$\overline{\text{HCS}}$/$\overline{\text{PPERR}}$§	N9	I/O/Z		Host chip select (**I**) [default] or PCI parity error (**I/O/Z**)
$\overline{\text{HDS1}}$/$\overline{\text{PSERR}}$§	R5	I/O/Z		Host data strobe 1 (**I**) [default] or PCI system error (**I/O/Z**)
$\overline{\text{HDS2}}$/$\overline{\text{PCBE1}}$§	P6	I/O/Z		Host data strobe 2 (**I**) [default] or PCI command/byte enable 1 (**I/O/Z**)
$\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$§	N4	I/O/Z		Host ready from DSP to host (**O**) [default] or PCI initiator ready (**I/O/Z**).
HD31/AD31§	J4	I/O/Z		Host-port data (**I/O/Z**) [default] (C64x) or PCI data-address bus (**I/O/Z**) [C6415 and C6416] As HPI data bus (PCI_EN pin = 0) • Used for transfer of data, address, and control • Host-Port bus width user-configurable at device reset via a 10-kΩ resistor pullup/pulldown resistor on the HD5 pin: HD5 pin = 0: HPI operates as an HPI16. (HPI bus is 16 bits wide. HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the high-impedance state.) HD5 pin = 1: HPI operates as an HPI32. (HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.) As PCI data-address bus (PCI_EN pin = 1) [C6415 and C6416 devices only] • Used for transfer of data and address The C6414 device does *not* support the PCI peripheral; therefore, the HPI peripheral pins are standalone peripheral functions, not MUXed.
HD30/AD30§	K7			
HD29/AD29§	J5			
HD28/AD28§	K4			
HD27/AD27§	K5			
HD26/AD26§	L6			
HD25/AD25§	L8			
HD24/AD24§	J6			
HD23/AD23§	L5			
HD22/AD22§	M5			
HD21/AD21§	M6			
HD20/AD20§	M8			
HD19/AD19§	L4			
HD18/AD18§	M4			
HD17/AD17§	M9			
HD16/AD16§	M7			
HD15/AD15§	P8			
HD14/AD14§	R6			
HD13/AD13§	R4			
HD12/AD12§	P7			
HD11/AD11§	R7			
HD10/AD10§	T5			
HD9/AD9§	T4			
HD8/AD8§	P9			
HD7/AD7§	T6			
HD6/AD6§	R8			
HD5/AD5§	U4			
HD4/AD4§	U5			
HD3/AD3§	T7			
HD2/AD2§	U6			
HD1/AD1§	V4			
HD0/AD0§	V5			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins are standalone peripheral functions for this device.

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
HOST-PORT INTERFACE (HPI) [C64x] or PERIPHERAL COMPONENT INTERCONNECT (PCI) [C6415 or C6416 devices only] (CONTINUED)				
$\overline{\text{PCBE0}}$	Y3	I/O/Z		PCI command/byte enable 0 (**I/O/Z**). When PCI is disabled (PCI_EN = 0), this pin is tied-off. For the C6414 device this pin is "Reserved (leave unconnected, ***do not*** connect to power or ground)."
XSP_CS	AA3	O	IPD	PCI serial interface chip select (**O**). When PCI is disabled (PCI_EN = 0), this pin is tied-off. For the C6414 device this pin is "Reserved (leave unconnected, ***do not*** connect to power or ground)."
CLKX2/ XSP_CLK§	W5	I/O/Z	IPD	McBSP2 transmit clock (**I/O/Z**) [default] or PCI serial interface clock (**O**).
DR2/XSP_DI§	Y4	I	IPU	McBSP2 receive data (**I**) [default] or PCI serial interface data in (**I**). In PCI mode, this pin is connected to the output data pin of the serial PROM.
DX2/XSP_DO§	R9	O/Z	IPU	McBSP2 transmit data (**O/Z**) [default] or PCI serial interface data out (**O**). In PCI mode, this pin is connected to the input data pin of the serial PROM.
GP15/$\overline{\text{PRST}}$§	J8	I/O/Z		General-purpose input/output (GPIO) 15 pin (**I/O/Z**) or PCI reset (**I**). No function at default.
GP14/PCLK§	G5			GPIO 14 pin (**I/O/Z**) or PCI clock (**I**). No function at default.
GP13/$\overline{\text{PINTA}}$§	G4			GPIO 13 pin (**I/O/Z**) or PCI interrupt A (**O/Z**). No function at default.
GP12/$\overline{\text{PGNT}}$§	J7			GPIO 12 pin (**I/O/Z**) or PCI bus grant (**I**). No function at default.
GP11/$\overline{\text{PREQ}}$§	H6			GPIO 11 pin (**I/O/Z**) or PCI bus request (**O/Z**). No function at default.
GP10/$\overline{\text{PCBE3}}$§	L7			GPIO 10 pin (**I/O/Z**) or PCI command/byte enable 3 (**I/O/Z**). No function at default.
GP9/PIDSEL§	K6			GPIO 9 pin (**I/O/Z**) or PCI initialization device select (**I**). No function at default.
EMIFA (64-bit) − CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY‖☆				
$\overline{\text{ACE3}}$	K20	O/Z	IPU	EMIFA memory space enables • Enabled by bits 28 through 31 of the word address • Only one pin is asserted during any external data access
$\overline{\text{ACE2}}$	L17	O/Z	IPU	
$\overline{\text{ACE1}}$	J21	O/Z	IPU	
$\overline{\text{ACE0}}$	K19	O/Z	IPU	
$\overline{\text{ABE7}}$	P19	O/Z	IPU	EMIFA byte-enable control • Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory. • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{\text{ABE6}}$	U22	O/Z	IPU	
$\overline{\text{ABE5}}$	T22	O/Z	IPU	
$\overline{\text{ABE4}}$	R21	O/Z	IPU	
$\overline{\text{ABE3}}$	M17	O/Z	IPU	
$\overline{\text{ABE2}}$	M18	O/Z	IPU	
$\overline{\text{ABE1}}$	H22	O/Z	IPU	
$\overline{\text{ABE0}}$	L19	O/Z	IPU	
$\overline{\text{APDT}}$	L20	O/Z	IPU	EMIFA peripheral data transfer, allows direct transfer between external peripherals

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins are standalone peripheral functions for this device.

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix "A" in front of a signal name indicates it is an EMIFA signal whereas a prefix "B" in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted from the signal name.

☆ To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-BIT) − BUS ARBITRATION‖☆				
AHOLDA	M19	O	IPU	EMIFA hold-request-acknowledge to the host
AHOLD	U21	I	IPU	EMIFA hold request from the host
ABUSREQ	P21	O	IPU	EMIFA bus request output
EMIFA (64-BIT) − ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL‖☆				
AECLKIN	J19	I	IPD	EMIFA external input clock. The EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) is selected at reset via the pullup/pulldown resistors on the BEA[17:16] pins. AECLKIN is the default for the EMIFA input clock.
AECLKOUT2	K18	O/Z	IPD	EMIFA output clock 2. Programmable to be EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) frequency divided-by-1, -2, or -4.
AECLKOUT1	H21	O/Z	IPD	EMIFA output clock 1 [at EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) frequency].
AARE/ ASDCAS/ ASADS/ASRE	L16	O/Z	IPU	EMIFA asynchronous memory read-enable/SDRAM column-address strobe/programmable synchronous interface-address strobe or read-enable • For programmable synchronous interface, the RENEN field in the CE Space Secondary Control Register (CExSEC) selects between ASADS and ASRE: If RENEN = 0, then the ASADS/ASRE signal functions as the ASADS signal. If RENEN = 1, then the ASADS/ASRE signal functions as the ASRE signal.
AAOE/ ASDRAS/ ASOE	J20	O/Z	IPU	EMIFA asynchronous memory output-enable/SDRAM row-address strobe/programmable synchronous interface output-enable
AAWE/ ASDWE/ ASWE	G22	O/Z	IPU	EMIFA asynchronous memory write-enable/SDRAM write-enable/programmable synchronous interface write-enable
ASDCKE	K21	O/Z	IPU	EMIFA SDRAM clock-enable (used for self-refresh mode). [EMIFA module only.] • If SDRAM is not in system, ASDCKE can be used as a general-purpose output.
ASOE3	N16	O/Z	IPU	EMIFA synchronous memory output-enable for ACE3 (for glueless FIFO interface)
AARDY	L18	I	IPU	Asynchronous memory ready input
EMIFA (64-BIT) − ADDRESS‖☆				
AEA22	R20	O/Z	IPD	EMIFA external address (doubleword address)
AEA21	P16			
AEA20	T20			
AEA19	R18			
AEA18	V22			
AEA17	R19			
AEA16	T21			
AEA15	P17			
AEA14	N18			
AEA13	P18			
AEA12	P20			
AEA11	N17			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-BIT) − ADDRESS‖☆ (CONTINUED)				
AEA10	N20	O/Z	IPD	EMIFA external address (doubleword address)
AEA9	N21			
AEA8	N19			
AEA7	M21			
AEA6	M20			
AEA5	L21			
AEA4	M16			
AEA3	J22			
EMIFA (64-bit) − DATA‖☆				
AED63	AB21	I/O/Z	IPU	EMIFA external data
AED62	W18			
AED61	Y19			
AED60	V17			
AED59	AA20			
AED58	AA19			
AED57	Y18			
AED56	T15			
AED55	U16			
AED54	AB20			
AED53	AA18			
AED52	V16			
AED51	W17			
AED50	Y17			
AED49	U15			
AED48	AB19			
AED47	T19			
AED46	U20			
AED45	R17			
AED44	Y22			
AED43	V21			
AED42	T18			
AED41	U19			
AED40	W21			
AED39	V20			
AED38	R16			
AED37	T17			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-bit) − DATA‖☆ (CONTINUED)				
AED36	U18	I/O/Z	IPU	EMIFA external data
AED35	Y21			
AED34	V19			
AED33	W20			
AED32	AA22			
AED31	D22			
AED30	G19			
AED29	F20			
AED28	H18			
AED27	E21			
AED26	F21			
AED25	G20			
AED24	K16			
AED23	J17			
AED22	E22			
AED21	G21			
AED20	J18			
AED19	H19			
AED18	H20			
AED17	K17			
AED16	F22			
AED15	F16			
AED14	E17			
AED13	H15			
AED12	C20			
AED11	D18			
AED10	G16			
AED9	F17			
AED8	D19			
AED7	E18			
AED6	J15			
AED5	H16			
AED4	G17			
AED3	D20			
AED2	F18			
AED1	E19			
AED0	C21			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆ To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFB (16-bit) − CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY‖☆				
$\overline{\text{BCE3}}$	D11	O/Z	IPU	EMIFB memory space enables • Enabled by bits 26 through 31 of the word address • Only one pin is asserted during any external data access
$\overline{\text{BCE2}}$	C9	O/Z	IPU	
$\overline{\text{BCE1}}$	H12	O/Z	IPU	
$\overline{\text{BCE0}}$	G12	O/Z	IPU	
$\overline{\text{BBE1}}$	D12	O/Z	IPU	EMIFB byte-enable control • Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory. • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{\text{BBE0}}$	E12	O/Z	IPU	
$\overline{\text{BPDT}}$	E11	O/Z	IPU	EMIFB peripheral data transfer, allows direct transfer between external peripherals
EMIFB (16-BIT) − BUS ARBITRATION‖☆				
$\overline{\text{BHOLDA}}$	F12	O	IPU	EMIFB hold-request-acknowledge to the host
$\overline{\text{BHOLD}}$	C19	I	IPU	EMIFB hold request from the host
BBUSREQ	D14	O	IPU	EMIFB bus request output
EMIFB (16-BIT) − ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL‖☆				
BECLKIN	E10	I	IPD	EMIFB external input clock. The EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) is selected at reset via the pullup/pulldown resistors on the BEA[15:14] pins. BECLKIN is the default for the EMIFB input clock.
BECLKOUT2	C8	O/Z	IPD	EMIFB output clock 2. Programmable to be EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) frequency divided by 1, 2, or 4.
BECLKOUT1	J12	O/Z	IPD	EMIFB output clock 1 [at EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) frequency].
$\overline{\text{BARE}}$/ $\overline{\text{BSDCAS}}$/ $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$	C7	O/Z	IPU	EMIFB asynchronous memory read-enable/SDRAM column-address strobe/programmable synchronous interface-address strobe or read-enable • For programmable synchronous interface, the RENEN field in the CE Space Secondary Control Register (CExSEC) selects between $\overline{\text{BSADS}}$ and $\overline{\text{BSRE}}$: If RENEN = 0, then the $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$ signal functions as the $\overline{\text{BSADS}}$ signal. If RENEN = 1, then the $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$ signal functions as the $\overline{\text{BSRE}}$ signal.
$\overline{\text{BAOE}}$/ $\overline{\text{BSDRAS}}$/ $\overline{\text{BSOE}}$	D10	O/Z	IPU	EMIFB asynchronous memory output-enable/SDRAM row-address strobe/programmable synchronous interface output-enable
$\overline{\text{BAWE}}$/$\overline{\text{BSDWE}}$/ $\overline{\text{BSWE}}$	F11	O/Z	IPU	EMIFB asynchronous memory write-enable/SDRAM write-enable/programmable synchronous interface write-enable
$\overline{\text{BSOE3}}$	J13	O/Z	IPU	EMIFB synchronous memory output enable for $\overline{\text{BCE3}}$ (for glueless FIFO interface)
BARDY	G11	I	IPU	EMIFB asynchronous memory ready input

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFB (16-BIT) − ADDRESS‖☆				
BEA20	E15	I/O/Z	IPU	EMIFB external address (half-word address) (**O/Z**) • Also controls initialization of DSP modes at reset (**I**) via pullup/pulldown resistors − Device Endian mode BEA20: 0 – Big Endian 1 – Little Endian (default mode) − Boot mode BEA[19:18]: 00 – No boot 01 – HPI boot 10 – EMIFB 8-bit ROM boot with default timings (default mode) 11 – Reserved − EMIF clock select BEA[17:16]: Clock mode select for EMIFA (AECLKIN_SEL[1:0]) 00 – AECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved BEA[15:14]: Clock mode select for EMIFB (BECLKIN_SEL[1:0]) 00 – BECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved − PCI EEPROM Auto-Initialization (EEAI) [C6415 and C6416 devices only] BEA13: PCI auto-initialization via external EEPROM If the PCI peripheral is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up. 0 – PCI auto-initialization through EEPROM is disabled (default). 1 – PCI auto-initialization through EEPROM is enabled. − UTOPIA Enable (UTOPIA_EN) [C6415 and C6416 devices only] BEA11: UTOPIA peripheral enable (functional) 0 – UTOPIA disabled (McBSP1 enabled) [default] 1 – UTOPIA enabled (McBSP1 disabled) The **C6414** device does ***not*** support the PCI and UTOPIA peripherals; for proper device operation, ***do not*** oppose the internal pulldowns (IPDs) on the BEA13 and BEA11 pins. **Also for proper C6414 device operation, *do not* oppose the IPDs on the BEA7, BEA8, and BEA9 pins.** **For proper C6415 device operation, the BEA7 pin must be externally pulled up with a 1-kΩ resistor.** **For proper C6416 device operation, the BEA8 and BEA9 pins must be externally pulled up with a 1-kΩ resistor.** For more details, see the Device Configurations section of this data sheet.
BEA19	D17		IPU	
BEA18	J14		IPD	
BEA17	E16			
BEA16	G15			
BEA15	C18			
BEA14	F15			
BEA13	D16			
BEA12	H14			
BEA11	F14			
BEA10	C17			
BEA9	G13			
BEA8	G14			
BEA7	E14			
BEA6	H13			
BEA5	C16			
BEA4	D15			
BEA3	E13			
BEA2	D13			
BEA1	F13			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFB (16-bit) − DATA‖☆				
BED15	F8	I/O/Z	IPU	EMIFB external data
BED14	J10			
BED13	D7			
BED12	C5			
BED11	H10			
BED10	G9			
BED9	C6			
BED8	E8			
BED7	E9			
BED6	F9			
BED5	G10			
BED4	J11			
BED3	D9			
BED2	D8			
BED1	H11			
BED0	F10			
MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP2)				
MCBSP2_EN	AB4	I	IPD	McBSP2 enable pin. This pin works in conjunction with the PCI_EN pin to enable/disable other peripherals (for more details, see the Device Configurations section of this data sheet).
CLKS2/GP8§	W7	I/O/Z	IPD	McBSP2 external clock source (CLKS2) [**input only**] [default] or this pin can also be programmed as a GPIO 8 pin (**I/O/Z**).
CLKR2	W4	I/O/Z	IPD	McBSP2 receive clock. When McBSP2 is disabled (PCI_EN = 1 and MCBSP2_EN pin = 0), this pin is tied-off.
CLKX2/ XSP_CLK§	W5	I/O/Z	IPD	McBSP2 transmit clock (**I/O/Z**) [default] or PCI serial interface clock (**O**).
DR2/XSP_DI§	Y4	I	IPU	McBSP2 receive data (**I**) [default] or PCI serial interface data in (**I**). In PCI mode, this pin is connected to the output data pin of the serial PROM.
DX2/XSP_DO§	R9	O/Z	IPU	McBSP2 transmit data (**O/Z**) [default] or PCI serial interface data out (**O**). In PCI mode, this pin is connected to the input data pin of the serial PROM.
FSR2	V6	I/O/Z	IPD	McBSP2 receive frame sync. When McBSP2 is disabled (PCI_EN = 1 and MCBSP2_EN pin = 0), this pin is tied-off.
FSX2	U7	I/O/Z	IPD	McBSP2 transmit frame sync. When McBSP2 is disabled (PCI_EN = 1 and MCBSP2_EN pin = 0), this pin is tied-off.

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins except CLKS2/GP8 are standalone peripheral functions for this device.

‖ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). The prefix “A” in front of a signal name indicates it is an EMIFA signal whereas a prefix “B” in front of a signal name indicates it is an EMIFB signal. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted from the signal name.

☆ To maintain signal integrity for the EMIF signals, serial termination resistors should be inserted into all EMIF output signal lines.

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)				
CLKS1/ URADDR3§	Y8	I		McBSP1 external clock source (as opposed to internal) (**I**) [default] or UTOPIA receive address 3 pin (**I**)
CLKR1/ URADDR2§	AB7	I/O/Z		McBSP1 receive clock (**I/O/Z**) [default] or UTOPIA receive address 2 pin (**I**)
CLKX1/ URADDR4§	T12	I/O/Z		McBSP1 transmit clock (**I/O/Z**) [default] or UTOPIA receive address 4 pin (**I**)
DR1/ UXADDR1§	V11	I		McBSP1 receive data (**I**) [default] or UTOPIA transmit address 1 pin (**I**)
DX1/ UXADDR4§	Y10	I/O/Z		McBSP1 transmit data (**O/Z**) [default] or UTOPIA transmit address 4 pin (**I**)
FSR1/ UXADDR2§	AA8	I/O/Z		McBSP1 receive frame sync (**I/O/Z**) [default] or UTOPIA transmit address 2 pin (**I**)
FSX1/ UXADDR3§	AA11	I/O/Z		McBSP1 transmit frame sync (**I/O/Z**) [default] or UTOPIA transmit address 3 pin (**I**)
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)				
CLKS0	F4	I	IPD	McBSP0 external clock source (as opposed to internal)
CLKR0	F5	I/O/Z	IPD	McBSP0 receive clock
CLKX0	K9	I/O/Z	IPD	McBSP0 transmit clock
DR0	G6	I	IPU	McBSP0 receive data
DX0	E4	O/Z	IPU	McBSP0 transmit data
FSR0	D3	I/O/Z	IPD	McBSP0 receive frame sync
FSX0	H7	I/O/Z	IPD	McBSP0 transmit frame sync
TIMER 2				
TOUT2	F7	O/Z	IPD	Timer 2 or general-purpose output
TINP2	D5	I	IPD	Timer 2 or general-purpose input
TIMER 1				
TOUT1	G8	O/Z	IPD	Timer 1 or general-purpose output
TINP1	D6	I	IPD	Timer 1 or general-purpose input
TIMER 0				
TOUT0	H9	O/Z	IPD	Timer 0 or general-purpose output
TINP0	E7	I	IPD	Timer 0 or general-purpose input

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins are standalone peripheral functions for this device.

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
UNIVERSAL TEST AND OPERATIONS PHY INTERFACE FOR ASYNCHRONOUS TRANSFER MODE (ATM) [UTOPIA SLAVE] [C6415 and C6416 devices only]				
UTOPIA SLAVE (ATM CONTROLLER) − TRANSMIT INTERFACE				
UXCLKΨ	AB8	I	□	Source clock for UTOPIA transmit driven by Master ATM Controller. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
UXCLAVΨ	AA13	O/Z		Transmit cell available status output signal from UTOPIA Slave. 0 indicates a complete cell is NOT available for transmit 1 indicates a complete cell is available for transmit When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
$\overline{\text{UXENB}}$Ψ	U13	I	◊	UTOPIA transmit interface enable input signal. Asserted by the Master ATM Controller to indicate that the UTOPIA Slave should put out on the Transmit Data Bus the first byte of valid data and the UXSOC signal in the next clock cycle. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
UXSOCΨ	Y12	O/Z		Transmit Start-of-Cell signal. This signal is output by the UTOPIA Slave on the rising edge of the UXCLK, indicating that the first valid byte of the cell is available on the 8-bit Transmit Data Bus (UXDATA[7:0]). When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
DX1/ UXADDR4§	Y10	I/O/Z	◊	McBSP1 [default] or UTOPIA transmit address pins As UTOPIA transmit address pins UXADDR[4:0] (**I**), UTOPIA_EN (BEA11 pin) = 1: • 5-bit Slave transmit address input pins driven by the Master ATM Controller to identify and select one of the Slave devices (up to 31 possible) in the ATM System. • UXADDR0 pin is tied off when the UTOPIA peripheral is disabled [UTOPIA_EN (BEA11 pin) = 0] For the McBSP1 pin functions (UTOPIA_EN (BEA11 pin) = 0 [default]), see the MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1) section of this table.
FSX1/ UXADDR3§	AA11	I/O/Z	◊	McBSP1 [default] or UTOPIA transmit address pins As UTOPIA transmit address pins UXADDR[4:0] (**I**), UTOPIA_EN (BEA11 pin) = 1: • 5-bit Slave transmit address input pins driven by the Master ATM Controller to identify and select one of the Slave devices (up to 31 possible) in the ATM System. • UXADDR0 pin is tied off when the UTOPIA peripheral is disabled [UTOPIA_EN (BEA11 pin) = 0] For the McBSP1 pin functions (UTOPIA_EN (BEA11 pin) = 0 [default]), see the MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1) section of this table.
FSR1/ UXADDR2§	AA8	I/O/Z	◊	
DR1/ UXADDR1§	V11	I	◊	
UXADDR0Ψ	Y9	I	◊	

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ For the C6415 and C6416 devices, these pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet. The C6414 device does *not* support the PCI or UTOPIA peripherals; therefore, these MUXed peripheral pins are standalone peripheral functions for this device.

□ For the C6415 and C6416 devices, external pulldowns required: *If* UTOPIA is selected (BEA11 = 1) *and* these pins are connected to other devices, then a 10-kΩ resistor must be used to **externally** pull down each of these pins. If these pins are “no connects”, then only UXCLK and URCLK need to be pulled down and other pulldowns are not necessary.

◊ For the C6415 and C6416 devices, external pullups required: *If* UTOPIA is selected (BEA11 = 1) *and* these pins are connected to other devices, then a 10-kΩ resistor must be used to **externally** pull up each of these pins. If these pins are “no connects”, then the pullups are not necessary.

Ψ The C6414 device does not support the UTOPIA peripheral; therefore, these standalone UTOPIA pins are **Reserved** (leave unconnected, ***do not*** connect to power or ground) with the exception of UXCLK and URCLK which should be connected to a 10-kΩ pulldown resistor (see the square [□] footnote).

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
UTOPIA SLAVE (ATM CONTROLLER) − TRANSMIT INTERFACE (CONTINUED)				
UXDATA7Ψ	W10	O/Z		8-bit Transmit Data Bus Using the Transmit Data Bus, the UTOPIA Slave (on the rising edge of the UXCLK) transmits the 8-bit ATM cells to the Master ATM Controller. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), these pins are tied-off.
UXDATA6Ψ	T11			
UXDATA5Ψ	W9			
UXDATA4Ψ	AB6			
UXDATA3Ψ	V10			
UXDATA2Ψ	U10			
UXDATA1Ψ	AA7			
UXDATA0Ψ	V9			
UTOPIA SLAVE (ATM CONTROLLER) − RECEIVE INTERFACE				
URCLKΨ	U12	I	□	Source clock for UTOPIA receive driven by Master ATM Controller. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
URCLAVΨ	AA14	O/Z		Receive cell available status output signal from UTOPIA Slave. 0 indicates NO space is available to receive a cell from Master ATM Controller 1 indicates space is available to receive a cell from Master ATM Controller When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
$\overline{\text{URENB}}$Ψ	AB16	I	◊	UTOPIA receive interface enable input signal. Asserted by the Master ATM Controller to indicate to the UTOPIA Slave to sample the Receive Data Bus (URDATA[7:0]) and URSOC signal in the next clock cycle or thereafter. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
URSOCΨ	W13	I	□	Receive Start-of-Cell signal. This signal is output by the Master ATM Controller to indicate to the UTOPIA Slave that the first valid byte of the cell is available to sample on the 8-bit Receive Data Bus (URDATA[7:0]). When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), this pin is tied-off.
CLKX1/ URADDR4§	T12	I/O/Z	◊	McBSP1 [default] or UTOPIA receive address pins As UTOPIA receive address pins URADDR[4:0] (**I**), UTOPIA_EN (BEA11 pin) = 1: • 5-bit Slave receive address input pins driven by the Master ATM Controller to identify and select one of the Slave devices (up to 31 possible) in the ATM System. • URADDR1 and URADDR0 pins are tied off when the UTOPIA peripheral is disabled [UTOPIA_EN (BEA11 pin) = 0] For the McBSP1 pin functions (UTOPIA_EN (BEA11 pin) = 0 [default]), see the MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1) section of this table.
CLKS1/ URADDR3§	Y8	I	◊	
CLKR1/ URADDR2§	AB7	I/O/Z	◊	
URADDR1Ψ	U11	I	◊	
URADDR0Ψ	AA9	I	◊	

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

§ These pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet.

□ External pulldowns required: *If* UTOPIA is selected (BEA11 = 1) *and* these pins are connected to other devices, then a 10-kΩ resistor must be used to **externally** pull down each of these pins. If these pins are “no connects”, then only UXCLK and URCLK need to be pulled down and other pulldowns are not necessary.

◊ External pullups required: *If* UTOPIA is selected (BEA11 = 1) *and* these pins are connected to other devices, then a 10-kΩ resistor must be used to **externally** pull up each of these pins. If these pins are “no connects”, then the pullups are not necessary.

Ψ The C6414 device does not support the UTOPIA peripheral; therefore, these standalone UTOPIA pins are **Reserved** (leave unconnected, ***do not*** connect to power or ground) with the exception of UXCLK and URCLK which should be connected to a 10-kΩ pulldown resistor (see the square [□] footnote).

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
UTOPIA SLAVE (ATM CONTROLLER) − RECEIVE INTERFACE (CONTINUED)				
URDATA7Ψ	AA10	I	□	8-bit Receive Data Bus. Using the Receive Data Bus, the UTOPIA Slave (on the rising edge of the URCLK) can receive the 8-bit ATM cell data from the Master ATM Controller. When the UTOPIA peripheral is disabled (UTOPIA_EN [BEA11 pin] = 0), these pins are tied-off.
URDATA6Ψ	V12			
URDATA5Ψ	W12			
URDATA4Ψ	W11			
URDATA3Ψ	Y11			
URDATA2Ψ	AB9			
URDATA1Ψ	Y13			
URDATA0Ψ	AA12			
RESERVED FOR TEST				
RSV	C4			Reserved (leave unconnected, ***do not*** connect to power or ground)
	H5			
	H3			
	N3			
	P3			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

‡ IPD = Internal pulldown, IPU = Internal pullup. (These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a 1-kΩ resistor should be used.)

□ External pulldowns required: *If* UTOPIA is selected (BEA11 = 1) *and* these pins are connected to other devices, then a 10-kΩ resistor must be used to **externally** pull down each of these pins. If these pins are “no connects”, then only UXCLK and URCLK need to be pulled down and other pulldowns are not necessary.

Ψ The C6414 device does not support the UTOPIA peripheral; therefore, these standalone UTOPIA pins are **Reserved** (leave unconnected, ***do not*** connect to power or ground) with the exception of UXCLK and URCLK which should be connected to a 10-kΩ pulldown resistor (see the square [□] footnote).

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS			
DV_{DD}	AA23	S	3.3-V supply voltage (see the Power-Supply Decoupling section of this data sheet)
	AB10		
	AB11		
	AB14		
	AB15		
	AB2		
	AB23		
	AC10		
	AC12		
	AC14		
	AC16		
	AC19		
	AC21		
	AC22		
	AC3		
	AC4		
	AC6		
	AC8		
	B11		
	B13		
	B15		
	B17		
	B19		
	B21		
	B22		
	B3		
	B4		
	B6		
	B9		
	C10		
	C11		
	C14		
	C15		
	C23		
	D2		
	D23		
	F2		
	F23		
	F3		
	G3		
	H2		
	J23		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
DV_{DD}	J3	S	3.3-V supply voltage (see the Power-Supply Decoupling section of this data sheet)
	K2		
	K22		
	K3		
	L22		
	L23		
	M2		
	N23		
	P2		
	P22		
	R22		
	R23		
	T2		
	T3		
	U23		
	U3		
	V2		
	V3		
	W23		
	W3		
	Y2		
CV_{DD}	A10	S	1.4 V supply voltage (see the Power-Supply Decoupling section of this data sheet)
	A12		
	A14		
	A16		
	A18		
	A20		
	A22		
	A4		
	A6		
	A8		
	AA1		
	AB12		
	AB13		
	AB24		
	AC18		
	AC20		
	AC5		
	AD11		
	AD13		
	AD15		
	AD17		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD}	AD19	S	1.4 V supply voltage (see the Power-Supply Decoupling section of this data sheet)
	AD21		
	AD23		
	AD3		
	AD5		
	AD7		
	AD9		
	B20		
	B24		
	B5		
	B7		
	C1		
	C12		
	C13		
	C2		
	D24		
	E1		
	E23		
	F24		
	G1		
	G23		
	H24		
	J1		
	K10		
	K12		
	K14		
	K24		
	L1		
	L11		
	L13		
	L15		
	L3		
	M10		
	M12		
	M14		
	M22		
	M24		
	M3		
	N1		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED			
CV_{DD}	N11	S	1.4 V supply voltage (see the Power-Supply Decoupling section of this data sheet)
	N13		
	N15		
	N22		
	P10		
	P12		
	P14		
	P24		
	R1		
	R11		
	R13		
	R15		
	R3		
	T24		
	U1		
	V24		
	W1		
	Y23		
	Y24		
GROUND PINS			
V_{SS}	A11	GND	Ground pins
	A13		
	A15		
	A17		
	A19		
	A21		
	A23		
	A3		
	A5		
	A7		
	A9		
	AA2		
	AA21		
	AA24		
	AA4		
	AB1		
	AB22		
	AB3		
	AC1		
	AC11		
	AC13		
	AC15		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS			
V_{SS}	AC17	GND	Ground pins
	AC2		
	AC23		
	AC24		
	AC7		
	AC9		
	AD10		
	AD12		
	AD14		
	AD16		
	AD18		
	AD2		
	AD20		
	AD22		
	AD4		
	AD6		
	AD8		
	B10		
	B12		
	B14		
	B16		
	B18		
	B2		
	B23		
	B8		
	C22		
	C24		
	C3		
	D1		
	D21		
	D4		
	E2		
	E20		
	E24		
	E5		
	F1		
	F19		
	F6		
	G18		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	G2	GND	Ground pins
	G24		
	G7		
	H1		
	H17		
	H23		
	H8		
	J16		
	J2		
	J24		
	J9		
	K1		
	K11		
	K13		
	K15		
	K23		
	L10		
	L12		
	L14		
	L2		
	L24		
	M1		
	M11		
	M13		
	M15		
	M23		
	N10		
	N12		
	N14		
	N2		
	N24		
	P1		
	P11		
	P13		
	P15		
	P23		
	R10		
	R12		
	R14		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	R2	GND	Ground pins
	R24		
	T1		
	T16		
	T23		
	T9		
	U17		
	U2		
	U24		
	U8		
	V1		
	V18		
	V23		
	V7		
	W19		
	W2		
	W24		
	W6		
	Y1		
	Y20		
	Y5		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

development support

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 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)
EVM (Evaluation Module)

For a complete listing of development-support tools for the TMS320C6000™ DSP platform, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL). For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Code Composer Studio, DSP/BIOS, and XDS are trademarks of Texas Instruments.

device and development-support tool nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ DSP devices and support tools. Each military TMS320™ DSP family member has one of three prefixes: SMX, TMP, or SMJ. Texas Instruments recommends two of three possible prefix designators for support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (SMX/TMDX) through fully qualified production devices/tools (SMJ/TMDS).

Device development evolutionary flow:

SMX Preproduction 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

SMJ 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

SMX and TMP devices and TMDX development-support tools are shipped with appropriate disclaimers describing their limitations and intended uses. Experimental devices (SMX) may not be representative of a final product and Texas Instruments reserves the right to change or discontinue these products without notice.

SMJ 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 preproduction or prototype devices (SMX 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, GAD), the temperature range (for example, blank is the default commercial temperature range), and the device speed range in megahertz (for example, -6E3 is 600-MHz CPU, 133-MHz EMIFA). Figure 4 provides a legend for reading the complete device name for any SMJ320C64x™ DSP generation member.

TMS320 is a trademark of Texas Instruments.

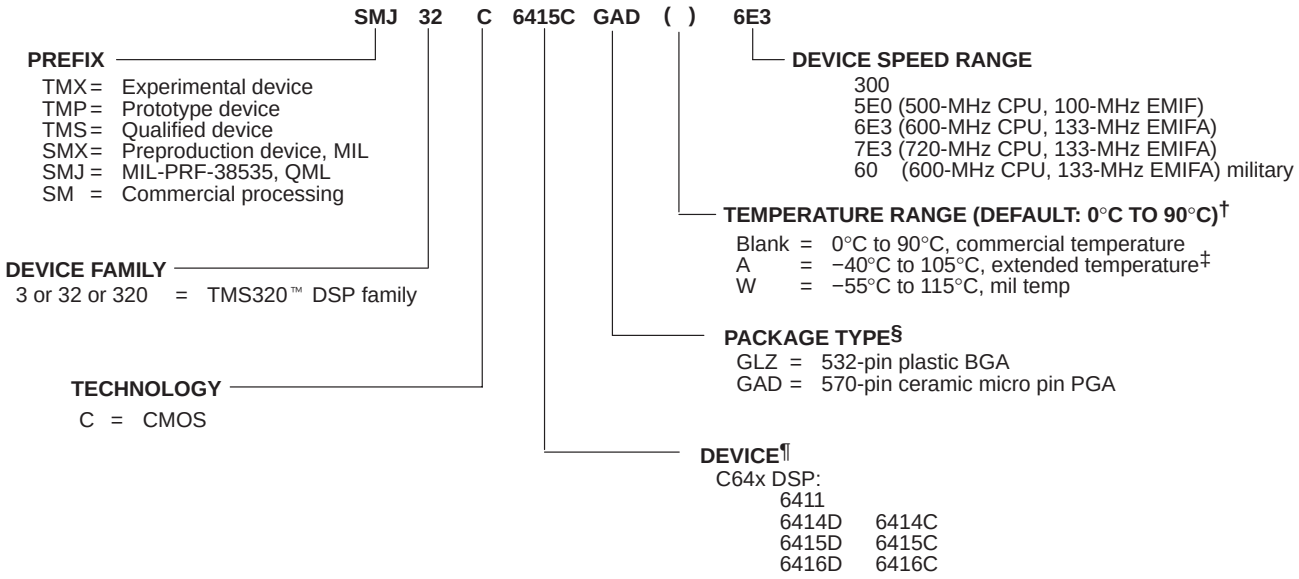
device and development-support tool nomenclature (continued)

Table 30. SMJ320C6414/C6415/C6416 Device Part Numbers (P/Ns) and Ordering Information

DEVICE ORDERABLE P/N	DEVICE SPEED, EMIF SPEED, SILICON REVISION	CV_{DD} (CORE VOLTAGE)	DV_{DD} (I/O VOLTAGE)	OPERATING CASE TEMPERATURE RANGE
C6414				
SMJ320C6414DGADW60†	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C
SM320C6414DGADW60†	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C
C6415				
SMJ320C6415DGADW60	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C
SM320C6415DGADW60	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C
C6416				
SMJ320C6416DGADW60†	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C
SM320C6416DGADW60†	600 MHz/4800 MIPS, 133-MHz EMIFA, Silicon Rev. 1.1	1.4 V	3.3 V	−55°C to 115°C

† Product Preview

device and development-support tool nomenclature (continued)



† See the Recommended Operating Conditions section of this data sheet for more details.
‡ The extended temperature “A version” devices may have different operating conditions than the commercial temperature devices. See the Recommended Operating Conditions section of this data sheet for more details.
§ BGA = Ball Grid Array.
PGA = Pin Grid Array.
¶ For the actual device part numbers (P/Ns) and ordering information, see Table 30 of this data sheet.

Figure 4. SMJ320C64x™ DSP Device Nomenclature (Including the C6414, C6415, and C6416 Devices)

documentation support

Extensive documentation supports all TMS320™ DSP family generations of devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's reference guides for all devices and tools; technical briefs; development-support tools; on-line help; and hardware and software applications. The following is a brief, descriptive list of support documentation specific to the C6000™ DSP devices:

The *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189) describes the C6000™ DSP CPU (core) architecture, instruction set, pipeline, and associated interrupts.

The *TMS320C6000 DSP Peripherals Overview Reference Guide* (literature number SPRU190) provides an overview and briefly describes the functionality of the peripherals available on the C6000™ DSP platform of devices. This document also includes a table listing the peripherals available on the C6000 devices along with literature numbers and hyperlinks to the associated peripheral documents.

The *TMS320C64x Technical Overview* (literature number SPRU395) gives an introduction to the C64x™ digital signal processor, and discusses the application areas that are enhanced by the C64x™ DSP VelociTI.2™ VLIW architecture.

The *TMS320C6414, TMS320C6415, and TMS320C6416 Digital Signal Processors Silicon Errata* (literature number SPRZ011) describes the known exceptions to the functional specifications for the SMJ320C6414, SMJ320C6415, and SMJ320C6416 devices.

The *TMS320C6414/15/16 Power Consumption Summary* application report (literature number SPRA811) discusses the power consumption for user applications with the SMJ320C6414, SMJ320C6415, and SMJ320C6416 DSP devices.

The *Using IBIS Models for Timing Analysis* application report (literature number SPRA839) describes how to properly use IBIS models to attain accurate timing analysis for a given system.

The tools support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE). For a complete listing of C6000™ DSP latest documentation, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

See the Worldwide Web URL for the *How To Begin Development Today With the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs* application report (literature number SPRA718), which describes in more details the compatibility and similarities/differences among the C6414, C6415, C6416, and C6211 devices.

clock PLL

Most of the internal C64x™ DSP clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock.

To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 5 shows the external PLL circuitry for either x1 (PLL bypass) or other PLL multiply modes.

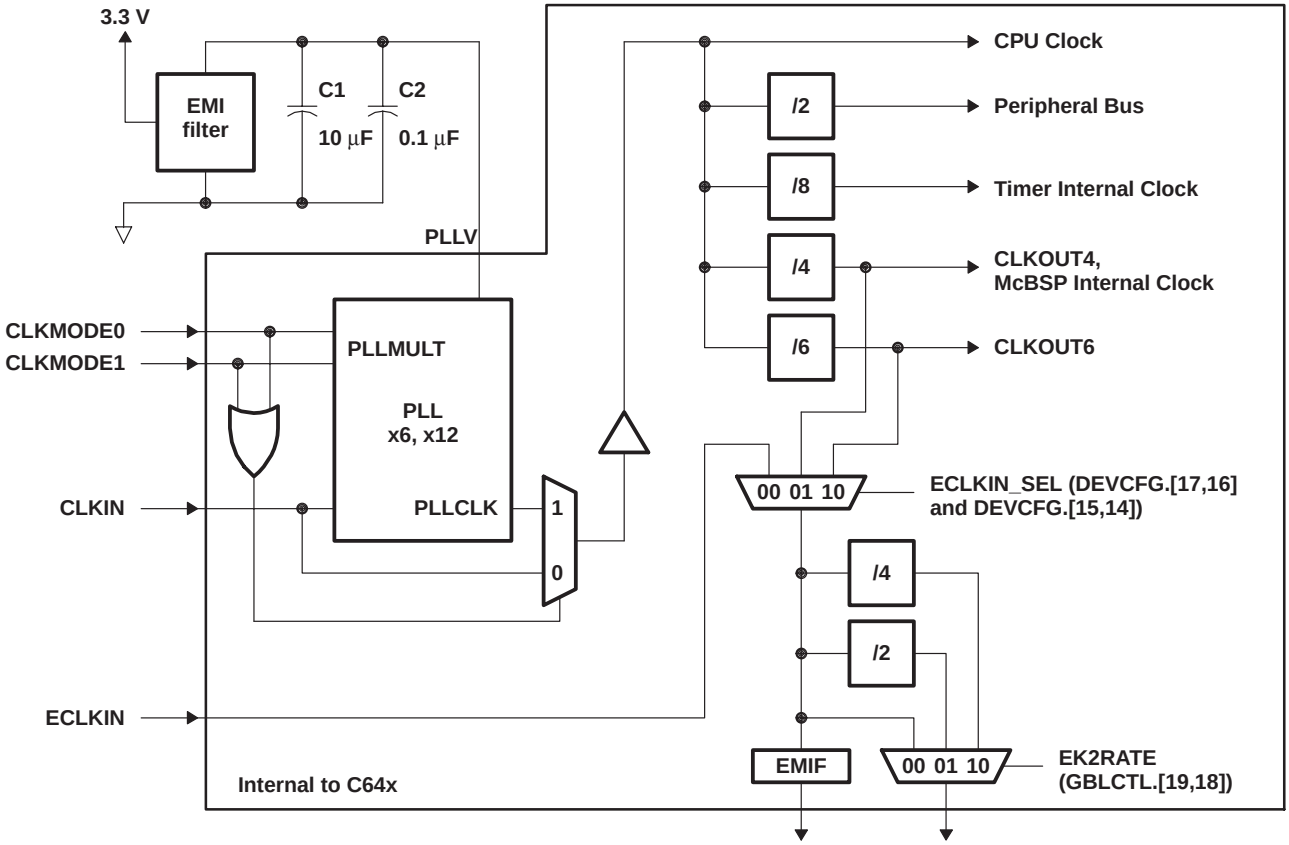
To minimize the clock jitter, a single clean power supply should power both the C64x™ DSP device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see the *input and output clocks* electrical section.

Rise/fall times, duty cycles (high/low pulse durations), and the load capacitance of the external clock source must meet the DSP requirements in this data sheet (see the *electrical characteristics over recommended ranges of supply voltage and operating case temperature* table and the *input and output clocks* electrical section). Table 31 lists some examples of compatible CLKIN external clock sources:

Table 31. Compatible CLKIN External Clock Sources

COMPATIBLE PARTS FOR EXTERNAL CLOCK SOURCES (CLKIN)	PART NUMBER	MANUFACTURER
Oscillators	JITO-2	Fox Electronix
	STA series, ST4100 series	SaRonix Corporation
	SG-636	Epson America
	342	Corning Frequency Control
PLL	ICS525-02	Integrated Circuit Systems

clock PLL (continued)



(For the PLL Options, CLKMODE Pins Setup, and PLL Clock Frequency Ranges, see Table 32.)

NOTES: A. Place all PLL external components (C1, C2, and the EMI Filter) as close to the C6000™ DSP device as possible. 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 the ones shown.

B. For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (C1, C2, and the EMI Filter).

C. The 3.3-V supply for the EMI filter must be from the same 3.3-V power plane supplying the I/O voltage, DV_{DD}.

D. EMI filter manufacturer TDK part number ACF451832-333, -223, -153, -103. Panasonic part number EXCCET103U.

Figure 5. External PLL Circuitry for Either PLL Multiply Modes or x1 (Bypass) Mode

clock PLL (continued)

Table 32. SMJ320C64x PLL Multiply Factor Options, Clock Frequency Ranges, and Typical Lock Time†‡

GAD PACKAGE − Micro Pin PGA							
CLKMODE1	CLKMODE0	CLKMODE (PLL MULTIPLY FACTORS)	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT4 RANGE (MHz)	CLKOUT6 RANGE (MHz)	TYPICAL LOCK TIME (μs)§
0	0	Bypass (x1)	30−75	30−75	7.5−18.8	5−12.5	N/A
0	1	x6	30−75	180−450	45−112.5	30−75	75
1	0	x12	30−60	360−720	90−180	60−120	
1	1	Reserved	−	−	−	−	−

† These clock frequency range values are applicable to a C64x−60 speed device..

‡ Use external pullup resistors on the CLKMODE pins (CLKMODE1 and CLKMODE0) to set the C64x device to one of the valid PLL multiply clock modes (x6 or x12). With internal pulldown resistors on the CLKMODE pins (CLKMODE1, CLKMODE0), the default clock mode is x1 (bypass).

§ Under some operating conditions, the maximum PLL lock time may vary by as much as 150% from the specified typical value. For example, if the typical lock time is specified as 100 μs, the maximum value may be as long as 250 μs.

general-purpose input/output (GPIO)

To use the GP[15:0] software-configurable GPIO pins, the GPxEN bits in the GP Enable (GPEN) Register and the GPxDIR bits in the GP Direction (GPDIR) Register must be properly configured.

GPxEN = 1 GP[x] pin is enabled

GPxDIR = 0 GP[x] pin is an input

GPxDIR = 1 GP[x] pin is an output

where “x” represents one of the 15 through 0 GPIO pins

Figure 6 shows the GPIO enable bits in the GPEN register for the C6414/C6415/C6416 device. To use any of the GPx pins as general-purpose input/output functions, the corresponding GPxEN bit must be set to “1” (enabled). Default values are device-specific, so refer to Figure 6 for the C6414/15/16 default configuration.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP15 EN	GP14 EN	GP13 EN	GP12 EN	GP11 EN	GP10 EN	GP9 EN	GP8 EN	GP7 EN	GP6 EN	GP5 EN	GP4 EN	GP3 EN	GP2 EN	GP1 EN	GP0 EN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-1

Legend: R/W = Readable/Writable; *-n* = value after reset, -x = undefined value after reset

Figure 6. GPIO Enable Register (GPEN) [Hex Address: 01B0 0000]

Figure 7 shows the GPIO direction bits in the GPDIR register. This register determines if a given GPIO pin is an input or an output providing the corresponding GPxEN bit is enabled (set to “1”) in the GPEN register. By default, all the GPIO pins are configured as input pins.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP15 DIR	GP14 DIR	GP13 DIR	GP12 DIR	GP11 DIR	GP10 DIR	GP9 DIR	GP8 DIR	GP7 DIR	GP6 DIR	GP5 DIR	GP4 DIR	GP3 DIR	GP2 DIR	GP1 DIR	GP0 DIR
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

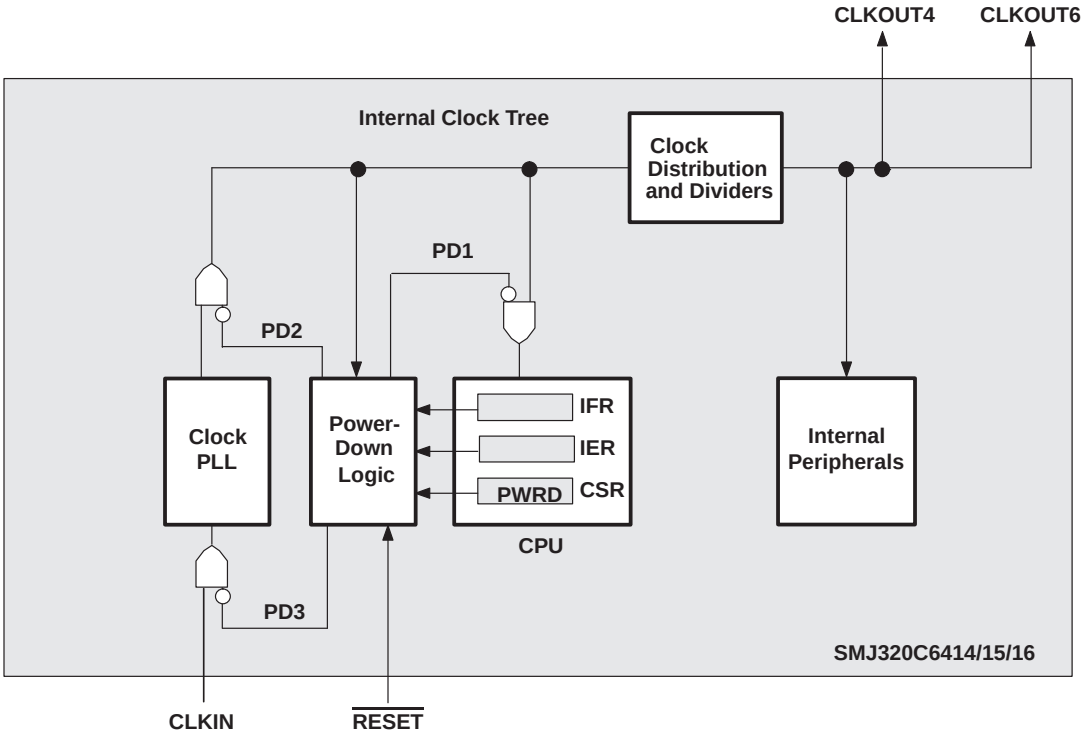
Legend: R/W = Readable/Writable; *-n* = value after reset, -x = undefined value after reset

Figure 7. GPIO Direction Register (GPDIR) [Hex Address: 01B0 0004]

For more detailed information on general-purpose inputs/outputs (GPIOs), see the *TMS320C6000 DSP General-Purpose Input/Output (GPIO) Reference Guide* (literature number SPRU584).

power-down mode logic

Figure 8 shows the power-down mode logic on the C6414/C6415/C6416.



† External input clocks, with the exception of CLKIN, are *not* gated by the power-down mode logic.

Figure 8. Power-Down Mode Logic†

triggering, wake-up, and effects

The power-down modes and their wake-up methods are programmed by setting the PWRD field (bits 15−10) of the control status register (CSR). The PWRD field of the CSR is shown in Figure 9 and described in Table 33. When writing to the CSR, all bits of the PWRD field should be set at the same time. Logic 0 should be used when writing to the reserved bit (bit 15) of the PWRD field. The CSR is discussed in detail in the *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189).

31							16

15	14	13	12	11	10	9	8
Reserved	Enable or Non-Enabled Interrupt Wake	Enabled Interrupt Wake	PD3	PD2	PD1		
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		

7							0

Legend: R/W−x = Read/write reset value

NOTE: The shadowed bits are not part of the power-down logic discussion and therefore are not covered here. For information on these other bit fields in the CSR register, see the *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189).

Figure 9. PWRD Field of the CSR Register

Power-down mode PD1 takes effect eight to nine clock cycles after the instruction that sets the PWRD bits in the CSR. If PD1 mode is terminated by a non-enabled interrupt, the program execution returns to the instruction where PD1 took effect. If PD1 mode is terminated by an enabled interrupt, the interrupt service routine will be executed first, then the program execution returns to the instruction where PD1 took effect.

PD2 and PD3 modes can only be aborted by device reset. Table 33 summarizes all the power-down modes.

Table 33. Characteristics of the Power-Down Modes

PRWD FIELD (BITS 15−10)	POWER-DOWN MODE	WAKE-UP METHOD	EFFECT ON CHIP'S OPERATION
000000	No power-down	—	—
001001	PD1	Wake by an enabled interrupt	CPU halted (except for the interrupt logic) Power-down mode blocks the internal clock inputs at the boundary of the CPU, preventing most of the CPU's logic from switching. During PD1, EDMA transactions can proceed between peripherals and internal memory.
010001	PD1	Wake by an enabled or non-enabled interrupt	
011010	PD2†	Wake by a device reset	Output clock from PLL is halted, stopping the internal clock structure from switching and resulting in the entire chip being halted. All register and internal RAM contents are preserved. All functional I/O "freeze" in the last state when the PLL clock is turned off.
011100	PD3†	Wake by a device reset	Input clock to the PLL stops generating clocks. All register and internal RAM contents are preserved. All functional I/O "freeze" in the last state when the PLL clock is turned off. Following reset, the PLL needs time to re-lock, just as it does following power-up. Wake-up from PD3 takes longer than wake-up from PD2 because the PLL needs to be re-locked, just as it does following power-up.
All others	Reserved	—	—

† When entering PD2 and PD3, all functional I/O remains in the previous state. However, for peripherals which are asynchronous in nature or peripherals with an external clock source, output signals may transition in response to stimulus on the inputs. Under these conditions, peripherals will not operate according to specifications.

C64x power-down mode with an emulator

If user power-down modes are programmed, and an emulator is attached, the modes will be masked to allow the emulator access to the system. This condition prevails until the emulator is reset or the cable is removed from the header. If power measurements are to be performed when in a power-down mode, the emulator cable should be removed.

When the DSP is in power-down mode PD2 or PD3, emulation logic will force any emulation execution command (such as Step or Run) to spin in IDLE. For this reason, PC writes (such as loading code) will fail. A DSP reset will be required to get the DSP out of PD2/PD3.

power-supply sequencing

TI DSPs do not require specific power sequencing between the core supply and the I/O supply. However, systems should be designed to ensure that neither supply is powered up for extended periods of time (>1 second) if the other supply is below the proper operating voltage.

power-supply design considerations

A dual-power supply with simultaneous sequencing can be used to eliminate the delay between core and I/O power up. A Schottky diode can also be used to tie the core rail to the I/O rail (see Figure 10).

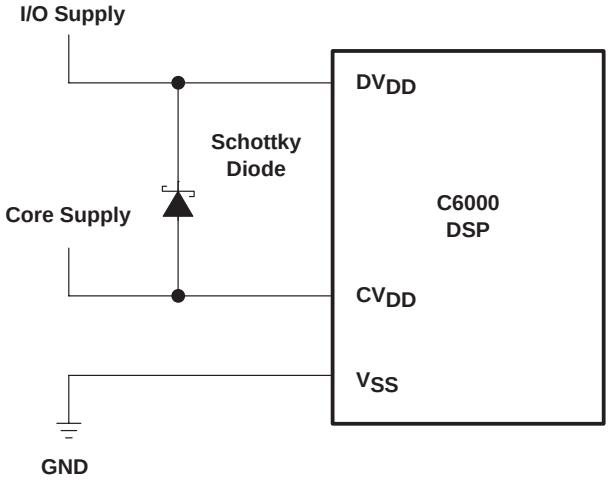


Figure 10. Schottky Diode Diagram

Core and I/O supply voltage regulators should be located close to the DSP (or DSP array) to minimize inductance and resistance in the power delivery path. Additionally, when designing for high-performance applications utilizing the C6000™ platform of DSPs, the PC board should include separate power planes for core, I/O, and ground, all bypassed with high-quality low-ESL/ESR capacitors.

power-supply decoupling

In order to properly decouple the supply planes from system noise, place as many capacitors (caps) as possible close to the DSP. Assuming 0603 caps, the user should be able to fit a total of 60 caps, 30 for the core supply and 30 for the I/O supply. These caps need to be close to the DSP, no more than 1.25 cm maximum distance to be effective. Physically smaller caps are better, such as 0402, but need to be evaluated from a yield/manufacturing point-of-view. Parasitic inductance limits the effectiveness of the decoupling capacitors, therefore physically smaller capacitors should be used while maintaining the largest available capacitance value. As with the selection of any component, verification of capacitor availability over the product's production lifetime should be considered.

IEEE 1149.1 JTAG compatibility statement

The SMJ320C6414/15/16 DSP requires that both $\overline{\text{TRST}}$ and $\overline{\text{RESET}}$ be asserted upon power up to be properly initialized. While $\overline{\text{RESET}}$ initializes the DSP core, $\overline{\text{TRST}}$ initializes the DSP's emulation logic. Both resets are required for proper operation.

While both $\overline{\text{TRST}}$ and $\overline{\text{RESET}}$ need to be asserted upon power up, only $\overline{\text{RESET}}$ needs to be released for the DSP to boot properly. $\overline{\text{TRST}}$ may be asserted indefinitely for normal operation, keeping the JTAG port interface and DSP's emulation logic in the reset state.

$\overline{\text{TRST}}$ only needs to be released when it is necessary to use a JTAG controller to debug the DSP or exercise the DSP's boundary scan functionality.

For maximum reliability, the SMJ320C6414/15/16 DSP includes an internal pulldown (IPD) on the $\overline{\text{TRST}}$ pin to ensure that $\overline{\text{TRST}}$ will always be asserted upon power up and the DSP's internal emulation logic will always be properly initialized.

JTAG controllers from Texas Instruments actively drive $\overline{\text{TRST}}$ high. However, some third-party JTAG controllers may not drive $\overline{\text{TRST}}$ high but expect the use of a pullup resistor on $\overline{\text{TRST}}$.

When using this type of JTAG controller, assert $\overline{\text{TRST}}$ to initialize the DSP after power up and externally drive $\overline{\text{TRST}}$ high before attempting any emulation or boundary scan operations. Following the release of $\overline{\text{RESET}}$, the low-to-high transition of $\overline{\text{TRST}}$ must be "seen" to latch the state of EMU1 and EMU0. The EMU[1:0] pins configure the device for either Boundary Scan mode or Emulation mode. For more detailed information, see the terminal functions section of this data sheet.

EMIF device speed

The rated EMIF speed, referring to both EMIFA and EMIFB, of these devices only applies to the SDRAM interface when in a system that meets the following requirements:

- 1 chip-enable (CE) space (maximum of 2 chips) of SDRAM connected to EMIF
- up to 1 CE space of buffers connected to EMIF
- EMIF trace lengths between 1 and 3 inches
- 166-MHz SDRAM for 133-MHz operation (applies only to EMIFA)
- 143-MHz SDRAM for 100-MHz operation

Other configurations may be possible, but timing analysis must be done to verify all AC timings are met. Verification of AC timings is mandatory when using configurations other than those specified above. TI recommends utilizing I/O buffer information specification (IBIS) to analyze all AC timings.

To properly use IBIS models to attain accurate timing analysis for a given system, see the *Using IBIS Models for Timing Analysis* application report (literature number SPRA839).

To maintain signal integrity, serial termination resistors should be inserted into all EMIF output signal lines (see the Terminal Functions table for the EMIF output signals).

bootmode

The C6414/15/16 device resets using the active-low signal $\overline{\text{RESET}}$. While $\overline{\text{RESET}}$ is low, the device is held in reset and is initialized to the prescribed reset state. Refer to reset timing for reset timing characteristics and states of device pins during reset. The release of $\overline{\text{RESET}}$ starts the processor running with the prescribed device configuration and boot mode.

The C6414/C6415/C6416 has three types of boot modes:

- Host boot

  If host boot is selected, upon release of $\overline{\text{RESET}}$, the CPU is internally “stalled” while the remainder of the device is released. During this period, an external host can initialize the CPU’s memory space as necessary through the host interface, including internal configuration registers, such as those that control the EMIF or other peripherals. For the C6414 device, the HPI peripheral is used for host boot. For the C6415/C6416 device, the HPI peripheral is used for host boot if PCI_EN = 0, and the PCI peripheral is used for host boot if PCI_EN = 1. Once the host is finished with all necessary initialization, it must set the DSPINT bit in the HPIC register to complete the boot process. This transition causes the boot configuration logic to bring the CPU out of the “stalled” state. The CPU then begins execution from address 0. The DSPINT condition is not latched by the CPU, because it occurs while the CPU is still internally “stalled”. Also, DSPINT brings the CPU out of the “stalled” state only if the host boot process is selected. All memory may be written to and read by the host. This allows for the host to verify what it sends to the DSP if required. After the CPU is out of the “stalled” state, the CPU needs to clear the DSPINT, otherwise, no more DSPINTs can be received.

- EMIF boot (using default ROM timings)

  Upon the release of $\overline{\text{RESET}}$, the 1K-Byte ROM code located in the beginning of $\overline{\text{CE1}}$ is copied to address 0 by the EDMA using the default ROM timings, while the CPU is internally “stalled”. The data should be stored in the endian format that the system is using. In this case, the EMIF automatically assembles consecutive 8-bit bytes to form the 32-bit instruction words to be copied. The transfer is automatically done by the EDMA as a single-frame block transfer from the ROM to address 0. After completion of the block transfer, the CPU is released from the “stalled” state and starts running from address 0.

- No boot

  With no boot, the CPU begins direct execution from the memory located at address 0. Note: operation is undefined if invalid code is located at address 0.

absolute maximum ratings over operating case temperature range (unless otherwise noted)†

Supply voltage ranges: CV_{DD} (see Note 1) . . . . . − 0.3 V to 1.8 V
DV_{DD} (see Note 1) . . . . . −0.3 V to 4 V
Input voltage ranges: (except PCI), V_I . . . . . −0.3 V to 4 V
(PCI), V_{IP} [C6415 and C6416 only] . . . . . −0.5 V to DV_{DD} + 0.5 V
Output voltage ranges: (except PCI), V_O . . . . . −0.3 V to 4 V
(PCI), V_{OP} [C6415 and C6416 only] . . . . . −0.5 V to DV_{DD} + 0.5 V
Operating case temperature ranges, T_C: . . . . . −55°C to 115°C
Storage temperature range, T_{stg} . . . . . −65°C to 150°C

† 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.

NOTE 1: All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
CV_{DD}	Supply voltage, Core‡		1.36	1.4	1.44	V
DV_{DD}	Supply voltage, I/O		3.14	3.3	3.46	V
V_{SS}	Supply ground		0	0	0	V
V_{IH}	High-level input voltage (except PCI)		2			V
V_{IL}	Low-level input voltage (except PCI)				0.8	V
V_{IP}	Input voltage (PCI) [C6415 and C6416 only]		−0.5		DV_{DD} + 0.5	V
V_{IHP}	High-level input voltage (PCI) [C6415 and C6416 only]		$0.5DV_{DD}$		DV_{DD} + 0.5	V
V_{ILP}	Low-level input voltage (PCI) [C6415 and C6416 only]		−0.5		$0.3DV_{DD}$	V
V_{OS}	Maximum voltage during overshoot/undershoot		−1.0§		4.3§	V
T_C	Operating case temperature	W version	–55		115	°C

‡ Future variants of the C641x DSPs may operate at voltages ranging from 0.9 V to 1.4 V to provide a range of system power/performance options. TI highly recommends that users design-in a supply that can handle multiple voltages within this range (i.e., 1.2 V, 1.25 V, 1.3 V, 1.35 V, 1.4 V with ± 3% tolerances) by implementing simple board changes such as reference resistor values or input pin configuration modifications. Examples of such supplies include the PT4660, PT5500, PT5520, PT6440, and PT6930 series from Power Trends, a subsidiary of Texas Instruments. Not incorporating a flexible supply may limit the system’s ability to easily adapt to future versions of C641x devices.

§ The absolute maximum ratings should *not* be exceeded for more than 30% of the cycle period.

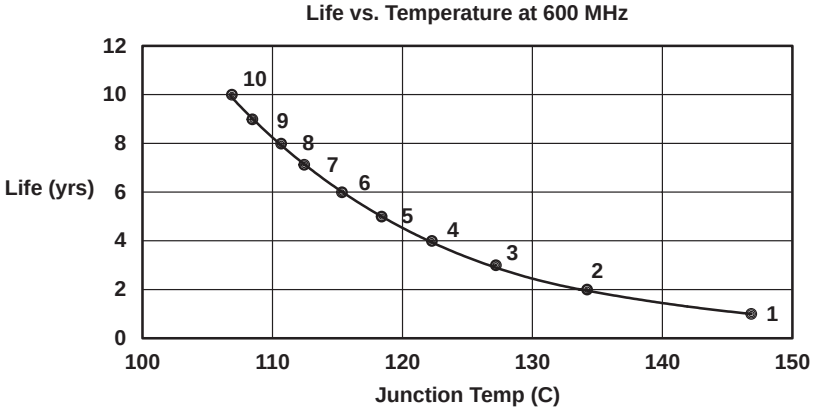


Figure 11. Impact of Elevated Temperature on Device Life

electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

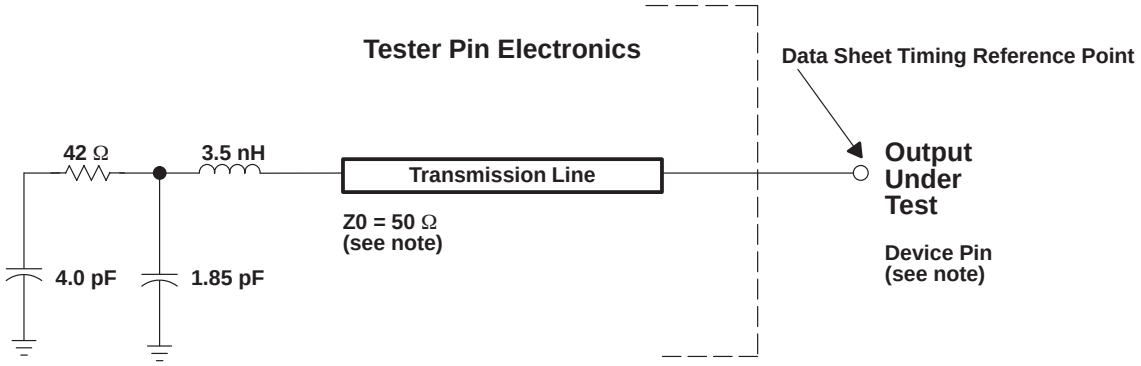
PARAMETER		TEST CONDITIONS†		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage (except PCI)	DV_{DD} = MIN,	I_{OH} = MAX	2.4			V
V_{OHP}	High-level output voltage (PCI) [C6415/C6416 only]	I_{OHP} = −0.5 mA,	DV_{DD} = 3.3 V	0.9DV_{DD}¶			V
V_{OL}	Low-level output voltage (except PCI)	DV_{DD} = MIN,	I_{OL} = MAX			0.4	V
V_{OLP}	Low-level output voltage (PCI) [C6415/C6416 only]	I_{OLP} = 1.5 mA,	DV_{DD} = 3.3 V			0.1DV_{DD}¶	V
I_I	Input current (except PCI)	V_I = V_{SS} to DV_{DD} no opposing internal resistor				±10	uA
		V_I = V_{SS} to DV_{DD} opposing internal pullup resistor‡		50	100	150	uA
		V_I = V_{SS} to DV_{DD} opposing internal pulldown resistor‡		−150	−100	−50	uA
I_{IP}	Input leakage current (PCI) [C6415/C6416 only]§	0 < V_{IP} < DV_{DD} = 3.3 V				±10	uA
I_{OH}	High-level output current	EMIF, CLKOUT4, CLKOUT6, EMUx				−16	mA
		Timer, UTOPIA, TDO, GPIO (Excluding GP[15:9, 2, 1]), McBSP				−8	mA
		PCI/HPI				−0.5¶	mA
I_{OL}	Low-level output current	EMIF, CLKOUT4, CLKOUT6, EMUx				16	mA
		Timer, UTOPIA, TDO, GPIO (Excluding GP[15:9, 2, 1]), McBSP				8	mA
		PCI/HPI				1.5¶	mA
I_{OZ}	Off-state output current	V_O = DV_{DD} or 0 V				±10	uA
I_{CDD}	Core supply current#	CV_{DD} = 1.4 V, CPU clock = 600 MHz			750		mA
I_{DDD}	I/O supply current#	DV_{DD} = 3.3 V, CPU clock = 600 MHz			125		mA
C_i	Input capacitance					12	pF
C_o	Output capacitance					12	pF

† For test conditions shown as MIN, MAX, or NOM, use the appropriate value specified in the recommended operating conditions table.
‡ Applies only to pins with an internal pullup (IPU) or pulldown (IPD) resistor.
§ PCI input leakage currents include Hi-Z output leakage for all bidirectional buffers with 3-state outputs.
¶ These rated numbers are from the PCI specification version 2.3. The DC specification and AC specification are defined in Tables 4-3 and 4-4, respectively.
# Measured with average activity (50% high/50% low power). The actual current draw is highly application-dependent. For more details on core and I/O activity, refer to the *TMS320C6414/15/16 Power Consumption Summary* application report (literature number SPRA811).

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.

PARAMETER MEASUREMENT INFORMATION



NOTE: The data sheet provides timing at the device pin. For output timing analysis, the tester pin electronics and its transmission line effects must be taken into account. A transmission line with a delay of 2 ns or longer can be used to produce the desired transmission line effect. The transmission line is intended as a load only. It is not necessary to add or subtract the transmission line delay (2 ns or longer) from the data sheet timings.

Input requirements in this data sheet are tested with an input slew rate of < 4 Volts per nanosecond (4 V/ns) at the device pin.

Figure 12. Test Load Circuit for AC Timing Measurements

The tester load circuit is for characterization and measurement of AC timing signals. This load does not indicate the maximum load the device is capable of driving.

signal transition levels

All input and output timing parameters are referenced to 1.5 V for both “0” and “1” logic levels.

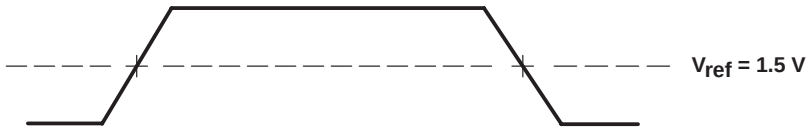


Figure 13. Input and Output Voltage Reference Levels for AC Timing Measurements

All rise and fall transition timing parameters are referenced to V_{IL} MAX and V_{IH} MIN for input clocks, V_{OL} MAX and V_{OH} MIN for output clocks, V_{ILP} MAX and V_{IHP} MIN for PCI input clocks, and V_{OLP} MAX and V_{OHP} MIN for PCI output clocks.

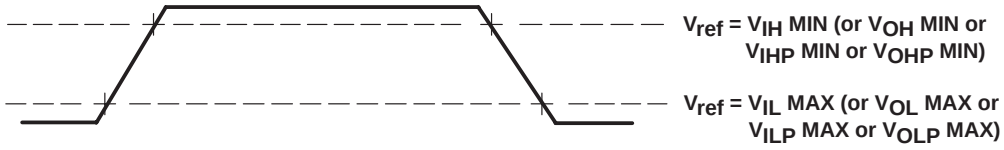


Figure 14. Rise and Fall Transition Time Voltage Reference Levels

signal transition rates

All timings are tested with an input edge rate of 4 Volts per nanosecond (4 V/ns).

PARAMETER MEASUREMENT INFORMATION (CONTINUED)

timing parameters and board routing analysis

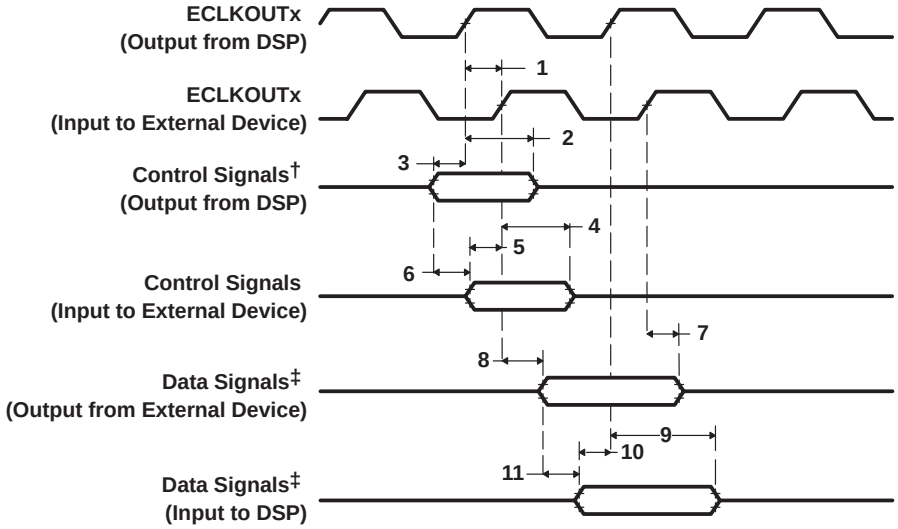
The timing parameter values specified in this data sheet do *not* include delays by board routings. As a good board design practice, such delays must *always* be taken into account. Timing values may be adjusted by increasing/decreasing such delays. TI recommends utilizing the available I/O buffer information specification (IBIS) models to analyze the timing characteristics correctly. If needed, external logic hardware such as buffers may be used to compensate any timing differences.

For inputs, timing is most impacted by the round-trip propagation delay from the DSP to the external device and from the external device to the DSP. This round-trip delay tends to negatively impact the input setup time margin, but also tends to improve the input hold time margins (see Table 34 and Figure 15).

Figure 15 represents a general transfer between the DSP and an external device. The figure also represents board route delays and how they are perceived by the DSP and the external device.

Table 34. Board-Level Timings Example (see Figure 15)

NO.	DESCRIPTION
1	Clock route delay
2	Minimum DSP hold time
3	Minimum DSP setup time
4	External device hold time requirement
5	External device setup time requirement
6	Control signal route delay
7	External device hold time
8	External device access time
9	DSP hold time requirement
10	DSP setup time requirement
11	Data route delay



† Control signals include data for Writes.
‡ Data signals are generated during Reads from an external device.

Figure 15. Board-Level Input/Output Timings

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN†‡§ (see Figure 16)

NO.			PLL MODE x12		PLL MODE x6		x1 (BYPASS)		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	20	33.3	13.3	33.3	13.3*	33.3	ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C*		0.4C*		0.45C*		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C*		0.4C*		0.45C*		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5*		5*		1*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ For more details on the PLL multiplier factors (x6, x12), see the *Clock PLL* section of this data sheet.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

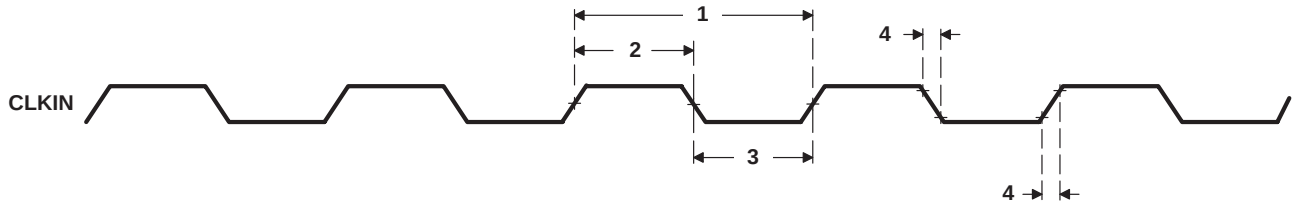


Figure 16. CLKIN Timing

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics over recommended operating conditions for CLKOUT4†‡§ (see Figure 17)

NO.	PARAMETER		CLKMODE = x1, x6, x12		UNIT
			MIN	MAX	
1	$t_{J(CKO4)}$	Period jitter, CLKOUT4	0*	±175*	ps
2	$t_{w(CKO4H)}$	Pulse duration, CLKOUT4 high	2P − 0.7*	2P + 0.7*	ns
3	$t_{w(CKO4L)}$	Pulse duration, CLKOUT4 low	2P − 0.7*	2P + 0.7*	ns
4	$t_{t(CKO4)}$	Transition time, CLKOUT4		1*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‡ PH is the high period of CLKIN in ns and PL is the low period of CLKIN in ns.
§ P = 1/CPU clock frequency in nanoseconds (ns)

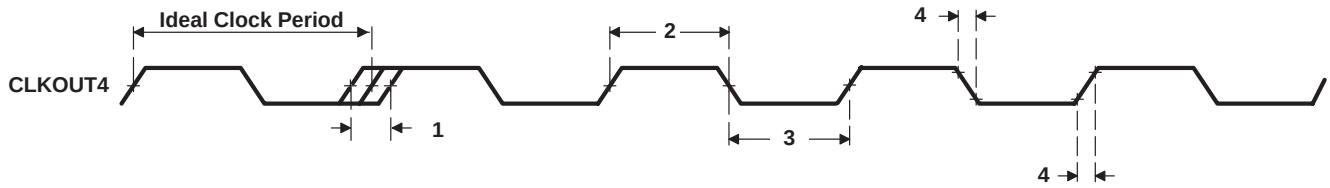


Figure 17. CLKOUT4 Timing

switching characteristics over recommended operating conditions for CLKOUT6†‡§ (see Figure 18)

NO.	PARAMETER		CLKMODE = x1, x6, x12		UNIT
			MIN	MAX	
1	$t_{J(CKO6)}$	Period jitter, CLKOUT6	0*	±175*	ps
2	$t_{w(CKO6H)}$	Pulse duration, CLKOUT6 high	3P − 0.7*	3P + 0.7*	ns
3	$t_{w(CKO6L)}$	Pulse duration, CLKOUT6 low	3P − 0.7*	3P + 0.7*	ns
4	$t_{t(CKO6)}$	Transition time, CLKOUT6		1*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‡ PH is the high period of CLKIN in ns and PL is the low period of CLKIN in ns.
§ P = 1/CPU clock frequency in nanoseconds (ns)

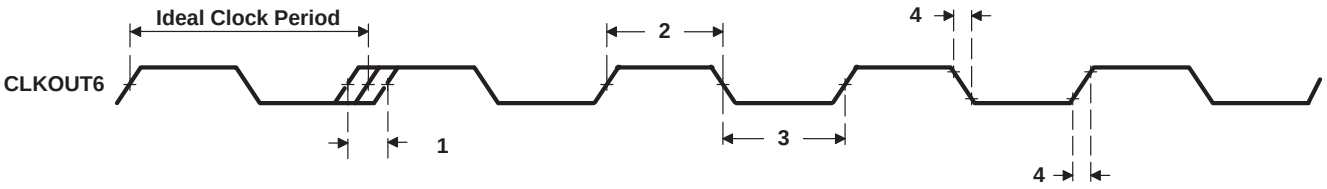


Figure 18. CLKOUT6 Timing

INPUT AND OUTPUT CLOCKS (CONTINUED)

timing requirements for ECLKIN for EMIFA and EMIFB†‡§ (see Figure 19)

NO.			MIN	MAX	UNIT
1	$t_{c(EKI)}$	Cycle time, ECLKIN	6¶*	16P*	ns
2	$t_{w(EKIH)}$	Pulse duration, ECLKIN high	2.7*		ns
3	$t_{w(EKIL)}$	Pulse duration, ECLKIN low	2.7*		ns
4	$t_{t(EKI)}$	Transition time, ECLKIN		2*	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
§ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.
¶ Minimum ECLKIN times are based on internal logic speed; the maximum useable speed of the EMIF may be lower due to AC timing requirements. On the 7E3 and 6E3 devices, 133-MHz operation is achievable if the requirements of the EMIF Device Speed section are met. On the 5E0 devices, 100-MHz operation is achievable if the requirements of the EMIF Device Speed section are met.

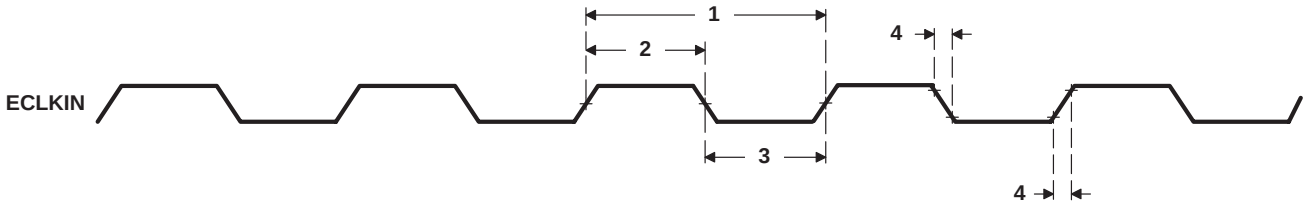


Figure 19. ECLKIN Timing for EMIFA and EMIFB

switching characteristics over recommended operating conditions for ECLKOUT1 for EMIFA and EMIFB modules§#‖☆ (see Figure 20)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{J(EKO1)}$	Period jitter, ECLKOUT1	0*	±175□*	ps
2	$t_{w(EKO1H)}$	Pulse duration, ECLKOUT1 high	EH − 0.7*	EH + 0.7*	ns
3	$t_{w(EKO1L)}$	Pulse duration, ECLKOUT1 low	EL − 0.7*	EL + 0.7*	ns
4	$t_{t(EKO1)}$	Transition time, ECLKOUT1		1*	ns
5	$t_{d(EKIH-EKO1H)}$	Delay time, ECLKIN high to ECLKOUT1 high	1*	8*	ns
6	$t_{d(EKIL-EKO1L)}$	Delay time, ECLKIN low to ECLKOUT1 low	1*	8*	ns

*This parameter is not production tested.
§ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.
# The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‖ E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
☆EH is the high period of E (EMIF input clock period) in ns and EL is the low period of E (EMIF input clock period) in ns for EMIFA or EMIFB.
□This period jitter specification was measured with CPU/4 or CPU/6 as the source of the EMIF input clock.

INPUT AND OUTPUT CLOCKS (CONTINUED)

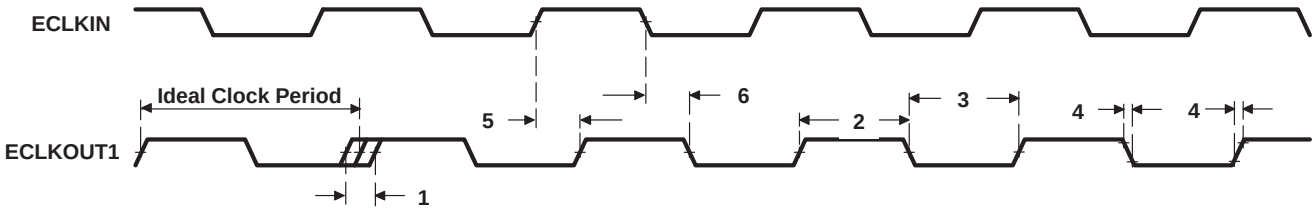


Figure 20. ECLKOUT1 Timing for EMIFA and EMIFB Modules

switching characteristics over recommended operating conditions for ECLKOUT2 for the EMIFA and EMIFB modules†‡§ (see Figure 21)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{J(EKO2)}$	Period jitter, ECLKOUT2	0*	±175¶*	ps
2	$t_{w(EKO2H)}$	Pulse duration, ECLKOUT2 high	0.5NE − 0.7*	0.5NE + 0.7*	ns
3	$t_{w(EKO2L)}$	Pulse duration, ECLKOUT2 low	0.5NE − 0.7*	0.5NE + 0.7*	ns
4	$t_{t(EKO2)}$	Transition time, ECLKOUT2		1*	ns
5	$t_{d(EKIH-EKO2H)}$	Delay time, ECLKIN high to ECLKOUT2 high	1*	8*	ns
6	$t_{d(EKIH-EKO2L)}$	Delay time, ECLKIN high to ECLKOUT2 low	1*	8*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‡ These C64x™ devices have two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.
§ E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
N = the EMIF input clock divider; N = 1, 2, or 4.
¶ This period jitter specification was measured with CPU/4 or CPU/6 as the source of the EMIF input clock.

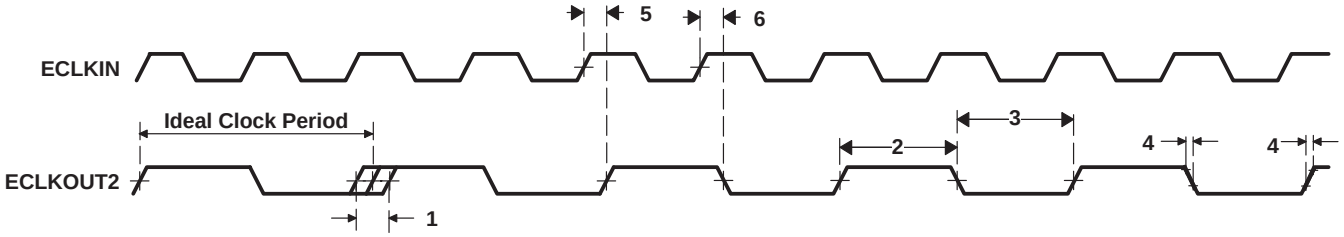


Figure 21. ECLKOUT2 Timing for the EMIFA and EMIFB Modules

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles for EMIFA module†‡§ (see Figure 22 and Figure 23)

NO.			MIN	MAX	UNIT
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{ARE}$ high	6.5		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{ARE}$ high	1		ns
6	$t_{su(ARDY-EKO1H)}$	Setup time, ARDY valid before ECLKOUTx high	3		ns
7	$t_{h(EKO1H-ARDY)}$	Hold time, ARDY valid after ECLKOUTx high	1		ns

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. The ARDY signal is *only* recognized two cycles before the end of the programmed strobe time and while ARDY is low, the strobe time is extended cycle-by-cycle. When ARDY is recognized low, the end of the strobe time is two cycles after ARDY is recognized high. To use ARDY as an asynchronous input, the pulse width of the ARDY signal should be wide enough (e.g., pulse width = 2E) to ensure setup and hold time is met.

‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

§ These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{AOE}$, $\overline{ARE}$, and $\overline{AWE}$) instead of $\overline{AAOE}$, $\overline{AARE}$, and $\overline{AAWE}$ (for EMIFA) and $\overline{BAOE}$, $\overline{BARE}$, and $\overline{BAWE}$ (for EMIFB)].

switching characteristics over recommended operating conditions for asynchronous memory cycles for EMIFA module‡§¶# (see Figure 22 and Figure 23)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS * E − 2		ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	RH * E − 1.9		ns
5	$t_{d(EKO1H-AREV)}$	Delay time, ECLKOUTx high to $\overline{ARE}$ valid	1	7	ns
8	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS * E − 1.7		ns
9	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	WH * E − 1.8		ns
10	$t_{d(EKO1H-AWEV)}$	Delay time, ECLKOUTx high to $\overline{AWE}$ valid	1.3	7.1	ns

‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

§ These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{AOE}$, $\overline{ARE}$, and $\overline{AWE}$) instead of $\overline{AAOE}$, $\overline{AARE}$, and $\overline{AAWE}$ (for EMIFA) and $\overline{BAOE}$, $\overline{BARE}$, and $\overline{BAWE}$ (for EMIFB)].

¶ E = ECLKOUT1 period in ns for EMIFA or EMIFB

Select signals for EMIFA include: $\overline{ACE}$x, $\overline{ABE}$[7:0], AEA[22:3], $\overline{AAOE}$; and for EMIFA writes, include AED[63:0].
Select signals EMIFB include: $\overline{BCE}$x, $\overline{BBE}$[1:0], BEA[20:1], $\overline{BAOE}$; and for EMIFB writes, include BED[15:0].

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

timing requirements for asynchronous memory cycles for EMIFB module†‡§ (see Figure 22 and Figure 23)

NO.			MIN	MAX	UNIT
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{ARE}$ high	6.2		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{ARE}$ high	1		ns
6	$t_{su(ARDY-EKO1H)}$	Setup time, ARDY valid before ECLKOUTx high	3		ns
7	$t_{h(EKO1H-ARDY)}$	Hold time, ARDY valid after ECLKOUTx high	1.2		ns

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. The ARDY signal is *only* recognized two cycles before the end of the programmed strobe time and while ARDY is low, the strobe time is extended cycle-by-cycle. When ARDY is recognized low, the end of the strobe time is two cycles after ARDY is recognized high. To use ARDY as an asynchronous input, the pulse width of the ARDY signal should be wide enough (e.g., pulse width = 2E) to ensure setup and hold time is met.

‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

§ These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{AOE}$, $\overline{ARE}$, and $\overline{AWE}$) instead of $\overline{AAOE}$, $\overline{AARE}$, and $\overline{AAWE}$ (for EMIFA) and $\overline{BAOE}$, $\overline{BARE}$, and $\overline{BAWE}$ (for EMIFB)].

switching characteristics over recommended operating conditions for asynchronous memory cycles for EMIFB module‡§¶# (see Figure 22 and Figure 23)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS * E − 2		ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	RH * E − 1.7		ns
5	$t_{d(EKO1H-AREV)}$	Delay time, ECLKOUTx high to $\overline{ARE}$ valid	0.8	6.6	ns
8	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS * E − 1.9		ns
9	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	WH * E − 1.7		ns
10	$t_{d(EKO1H-AWEV)}$	Delay time, ECLKOUTx high to $\overline{AWE}$ valid	0.9	6.7	ns

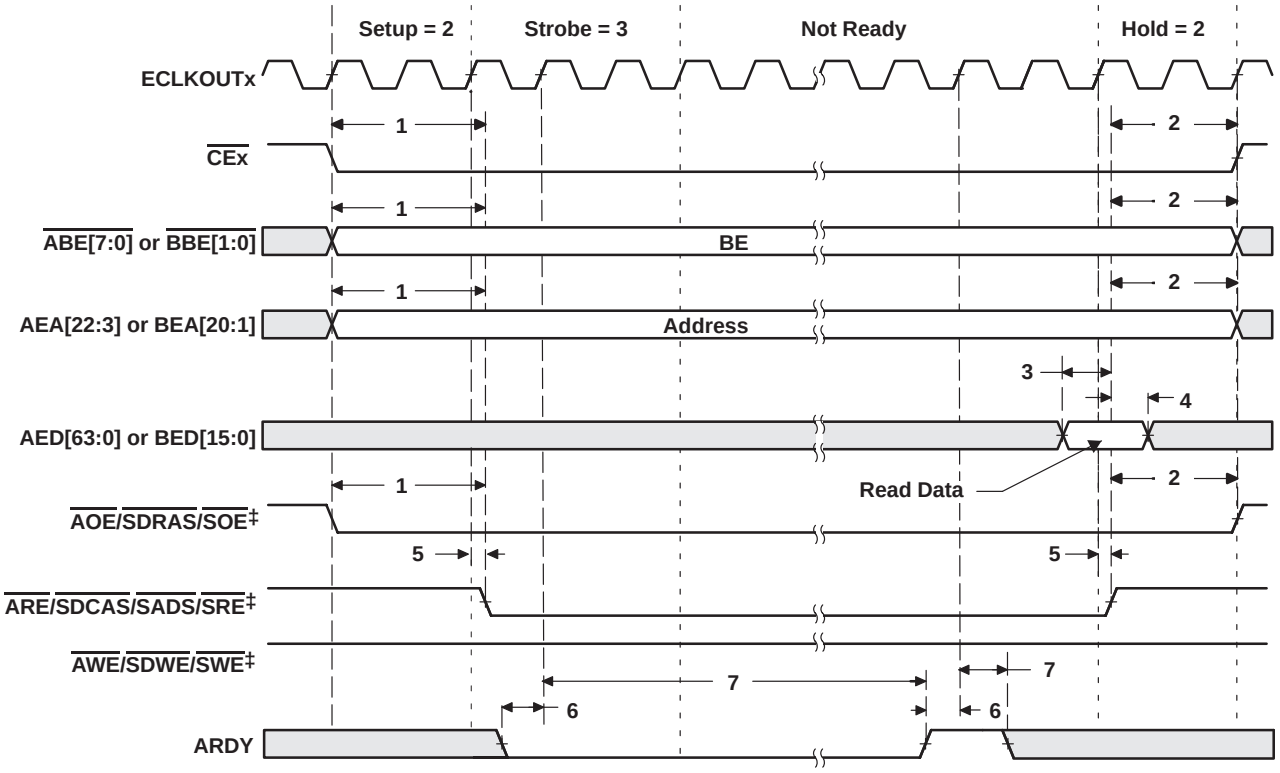
‡ RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed via the EMIF CE space control registers.

§ These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{AOE}$, $\overline{ARE}$, and $\overline{AWE}$) instead of $\overline{AAOE}$, $\overline{AARE}$, and $\overline{AAWE}$ (for EMIFA) and $\overline{BAOE}$, $\overline{BARE}$, and $\overline{BAWE}$ (for EMIFB)].

¶ E = ECLKOUT1 period in ns for EMIFA or EMIFB

Select signals for EMIFA include: $\overline{ACEx}$, $\overline{ABE[7:0]}$, AEA[22:3], $\overline{AAOE}$; and for EMIFA writes, include AED[63:0].
Select signals EMIFB include: $\overline{BCEx}$, $\overline{BBE[1:0]}$, BEA[20:1], $\overline{BAOE}$; and for EMIFB writes, include BED[15:0].

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

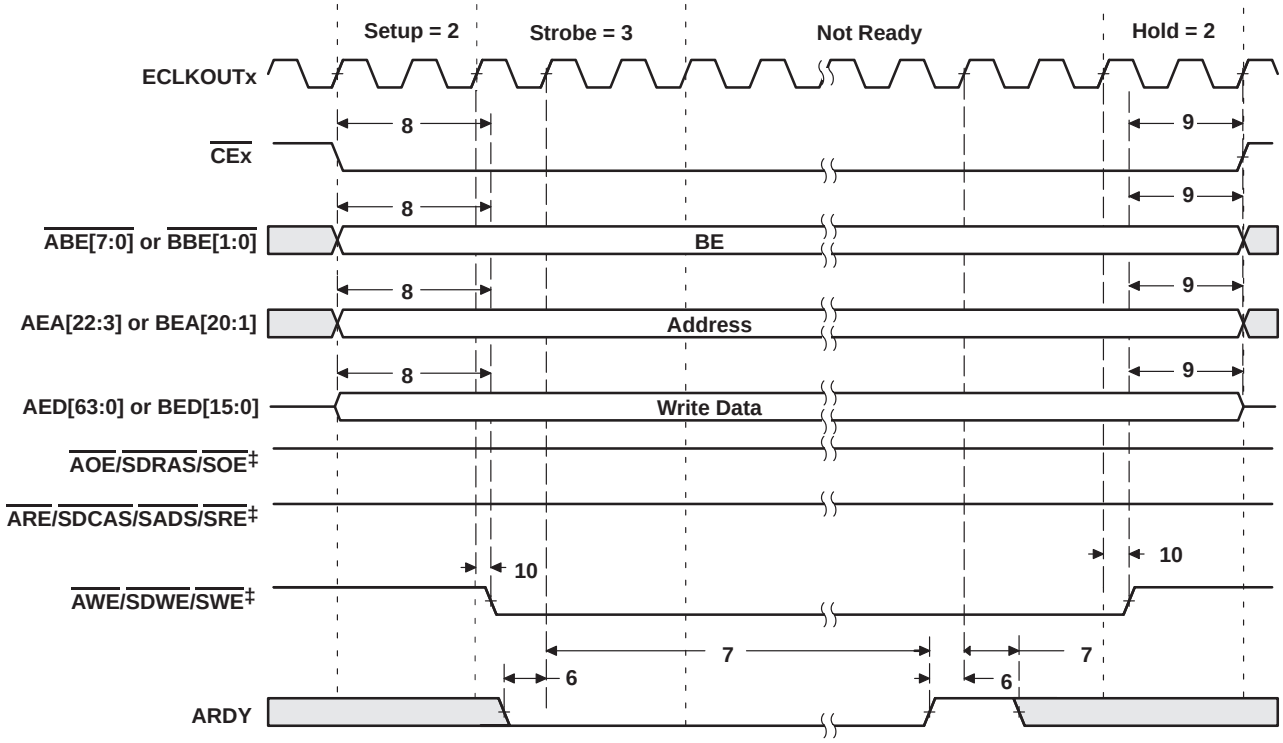


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{\text{AOE}}$, $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$) instead of $\overline{\text{AAOE}}$, $\overline{\text{AARE}}$, and $\overline{\text{AAWE}}$ (for EMIFA) and $\overline{\text{BAOE}}$, $\overline{\text{BARE}}$, and $\overline{\text{BAWE}}$ (for EMIFB)].

‡ $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SOE}}$, $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SADS}}$/$\overline{\text{SRE}}$, and $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SWE}}$ operate as $\overline{\text{AOE}}$ (identified under select signals), $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$, respectively, during asynchronous memory accesses.

Figure 22. Asynchronous Memory Read Timing for EMIFA and EMIFB†

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the asynchronous memory access signals are shown as generic ($\overline{\text{AOE}}$, $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$) instead of $\overline{\text{AAOE}}$, $\overline{\text{AARE}}$, and $\overline{\text{AAWE}}$ (for EMIFA) and $\overline{\text{BAOE}}$, $\overline{\text{BARE}}$, and $\overline{\text{BAWE}}$ (for EMIFB)].

‡ $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SOE}}$, $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SADS}}$/$\overline{\text{SRE}}$, and $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SWE}}$ operate as $\overline{\text{AOE}}$ (identified under select signals), $\overline{\text{ARE}}$, and $\overline{\text{AWE}}$, respectively, during asynchronous memory accesses.

Figure 23. Asynchronous Memory Write Timing for EMIFA and EMIFB†

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING

timing requirements for programmable synchronous interface cycles for EMIFA module† (see Figure 24)

NO.			MIN	MAX	UNIT
6	$t_{su(EDV-EKOxH)}$	Setup time, read EDx valid before ECLKOUTx high	2		ns
7	$t_{h(EKOxH-EDV)}$	Hold time, read EDx valid after ECLKOUTx high	1.5		ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic ($\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{SOE}}$, and $\overline{\text{SWE}}$) instead of $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$, $\overline{\text{ASOE}}$, and $\overline{\text{ASWE}}$ (for EMIFA) and $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$, $\overline{\text{BSOE}}$, and $\overline{\text{BSWE}}$ (for EMIFB)].

switching characteristics over recommended operating conditions for programmable synchronous interface cycles for EMIFA module†‡ (see Figure 24−Figure 26)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(EKOxH-CEV)}$	Delay time, ECLKOUTx high to $\overline{\text{CEx}}$ valid	1.3	4.9	ns
2	$t_{d(EKOxH-BEV)}$	Delay time, ECLKOUTx high to $\overline{\text{BEx}}$ valid		5.1	ns
3	$t_{d(EKOxH-BEIV)}$	Delay time, ECLKOUTx high to $\overline{\text{BEx}}$ invalid	1.3		ns
4	$t_{d(EKOxH-EAV)}$	Delay time, ECLKOUTx high to EAx valid		4.9	ns
5	$t_{d(EKOxH-EAIV)}$	Delay time, ECLKOUTx high to EAx invalid	1.3		ns
8	$t_{d(EKOxH-ADSV)}$	Delay time, ECLKOUTx high to $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ valid	1.3	4.9	ns
9	$t_{d(EKOxH-OEV)}$	Delay time, ECLKOUTx high to, $\overline{\text{SOE}}$ valid	1.3	4.9	ns
10	$t_{d(EKOxH-EDV)}$	Delay time, ECLKOUTx high to $\overline{\text{EDx}}$ valid		4.9	ns
11	$t_{d(EKOxH-EDIV)}$	Delay time, ECLKOUTx high to $\overline{\text{EDx}}$ invalid	1.3		ns
12	$t_{d(EKOxH-WEV)}$	Delay time, ECLKOUTx high to $\overline{\text{SWE}}$ valid	1.3	4.9	ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic ($\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{SOE}}$, and $\overline{\text{SWE}}$) instead of $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$, $\overline{\text{ASOE}}$, and $\overline{\text{ASWE}}$ (for EMIFA) and $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$, $\overline{\text{BSOE}}$, and $\overline{\text{BSWE}}$ (for EMIFB)].

‡ The following parameters are programmable via the EMIF CE Space Secondary Control register (CExSEC):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{\text{CEx}}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{CEx}}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{\text{CEx}}$ is active when $\overline{\text{SOE}}$ is active (CEEXT = 1).
- Function of $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SADS}}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SRE}}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)

timing requirements for programmable synchronous interface cycles for EMIFB module† (see Figure 24)

NO.			MIN	MAX	UNIT
6	$t_{su(EDV-EKOxH)}$	Setup time, read EDx valid before ECLKOUTx high	3.1		ns
7	$t_{h(EKOxH-EDV)}$	Hold time, read EDx valid after ECLKOUTx high	1.5		ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic (SADS/SRE, SOE, and SWE) instead of ASADS/ASRE, ASOE, and ASWE (for EMIFA) and BSADS/BSRE, BSOE, and BSWE (for EMIFB)].

switching characteristics over recommended operating conditions for programmable synchronous interface cycles for EMIFB module†‡ (see Figure 24−Figure 26)

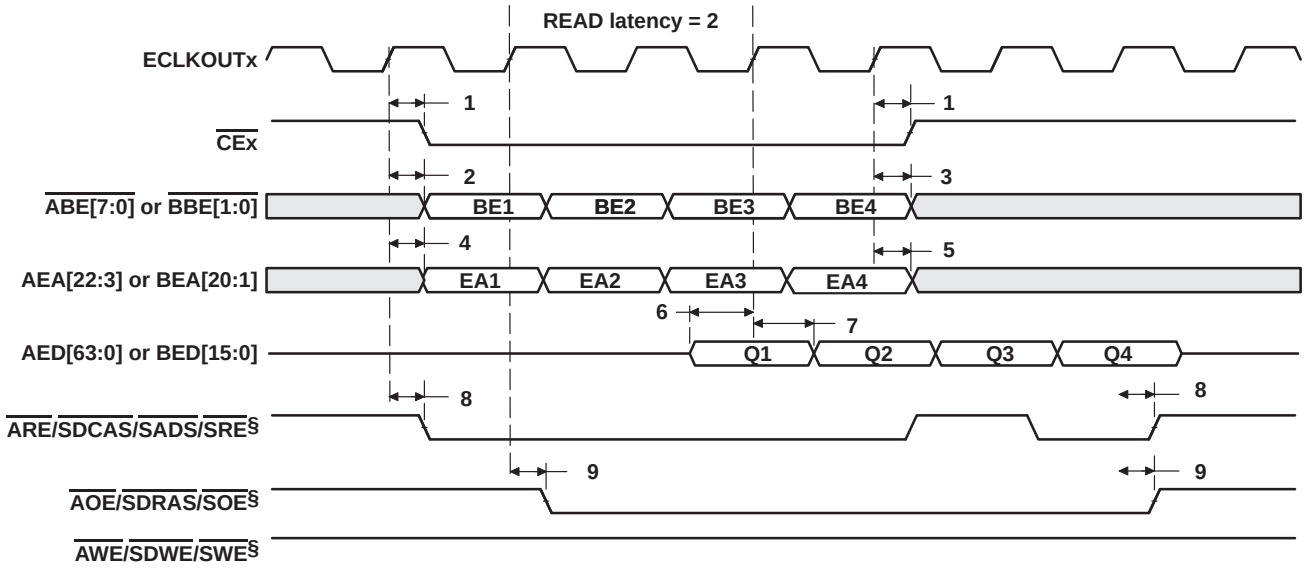
NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(EKOxH-CEV)}$	Delay time, ECLKOUTx high to CEx valid	1.3	6.4	ns
2	$t_{d(EKOxH-BEV)}$	Delay time, ECLKOUTx high to BEx valid		6.4	ns
3	$t_{d(EKOxH-BEIV)}$	Delay time, ECLKOUTx high to BEx invalid	1.3		ns
4	$t_{d(EKOxH-EAV)}$	Delay time, ECLKOUTx high to EAx valid		6.4	ns
5	$t_{d(EKOxH-EAIV)}$	Delay time, ECLKOUTx high to EAx invalid	1.3		ns
8	$t_{d(EKOxH-ADSV)}$	Delay time, ECLKOUTx high to SADS/SRE valid	1.3	6.4	ns
9	$t_{d(EKOxH-OEV)}$	Delay time, ECLKOUTx high to, SOE valid	1.3	6.4	ns
10	$t_{d(EKOxH-EDV)}$	Delay time, ECLKOUTx high to EDx valid		6.4	ns
11	$t_{d(EKOxH-EDIV)}$	Delay time, ECLKOUTx high to EDx invalid	1.3		ns
12	$t_{d(EKOxH-WEV)}$	Delay time, ECLKOUTx high to SWE valid	1.3	6.4	ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic (SADS/SRE, SOE, and SWE) instead of ASADS/ASRE, ASOE, and ASWE (for EMIFA) and BSADS/BSRE, BSOE, and BSWE (for EMIFB)].

‡ The following parameters are programmable via the EMIF CE Space Secondary Control register (CExSEC):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- CEx assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, CEx goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, CEx is active when SOE is active (CEEXT = 1).
- Function of SADS/SRE (RENEN): For standard SBSRAM or ZBT SRAM interface, SADS/SRE acts as SADS with deselect cycles (RENEN = 0). For FIFO interface, SADS/SRE acts as SRE with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic ($\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{SOE}}$, and $\overline{\text{SWE}}$) instead of $\overline{\text{ASADS}}$/$\overline{\text{ASRE}}$, $\overline{\text{ASOE}}$, and $\overline{\text{ASWE}}$ (for EMIFA) and $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$, $\overline{\text{BSOE}}$, and $\overline{\text{BSWE}}$ (for EMIFB)].

‡ The read latency and the length of $\overline{\text{CEx}}$ assertion are programmable via the SYNCRL and CEEXT fields, respectively, in the EMIFx CE Space Secondary Control register (CExSEC). In this figure, SYNCRL = 2 and CEEXT = 0.

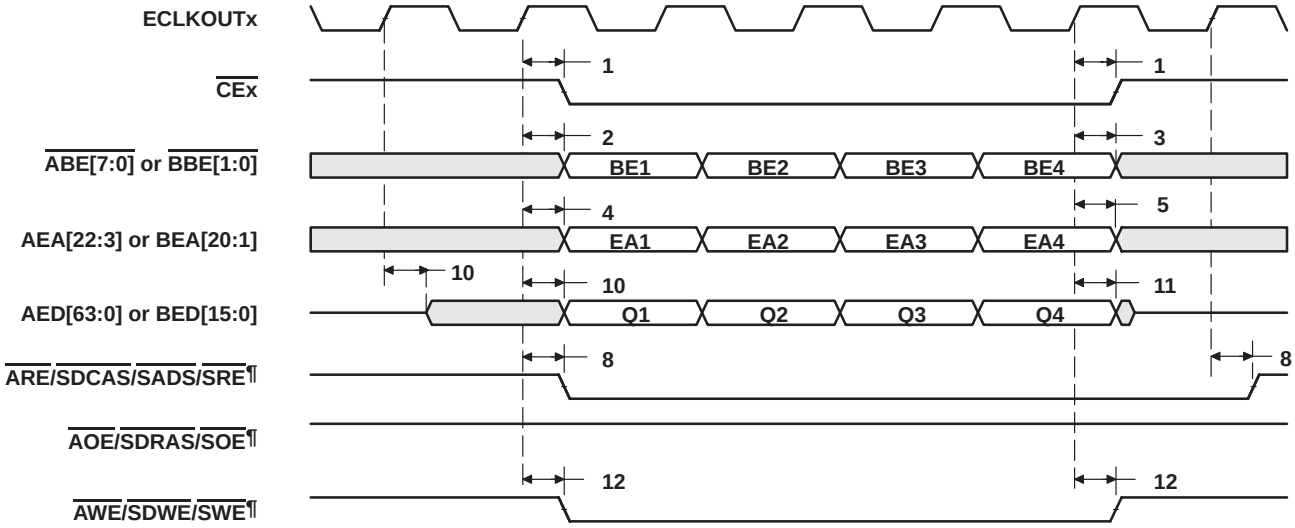
§ The following parameters are programmable via the EMIF CE Space Secondary Control register (CExSEC):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{\text{CEx}}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{CEx}}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{\text{CEx}}$ is active when $\overline{\text{SOE}}$ is active (CEEXT = 1).
- Function of $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SADS}}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{\text{SADS}}$/$\overline{\text{SRE}}$ acts as $\overline{\text{SRE}}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

¶ $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SOE}}$, and $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SWE}}$ operate as $\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{SOE}}$, and $\overline{\text{SWE}}$, respectively, during programmable synchronous interface accesses.

Figure 24. Programmable Synchronous Interface Read Timing for EMIFA and EMIFB (With Read Latency = 2)†‡§

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic (SADS/SRE, SOE, and SWE) instead of ASADS/ASRE, ASOE, and ASWE (for EMIFA) and BSADS/BSRE, BSOE, and BSWE (for EMIFB)].

‡ The write latency and the length of CEx assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIFx CE Space Secondary Control register (CExSEC). In this figure, SYNCWL = 0 and CEEXT = 0.

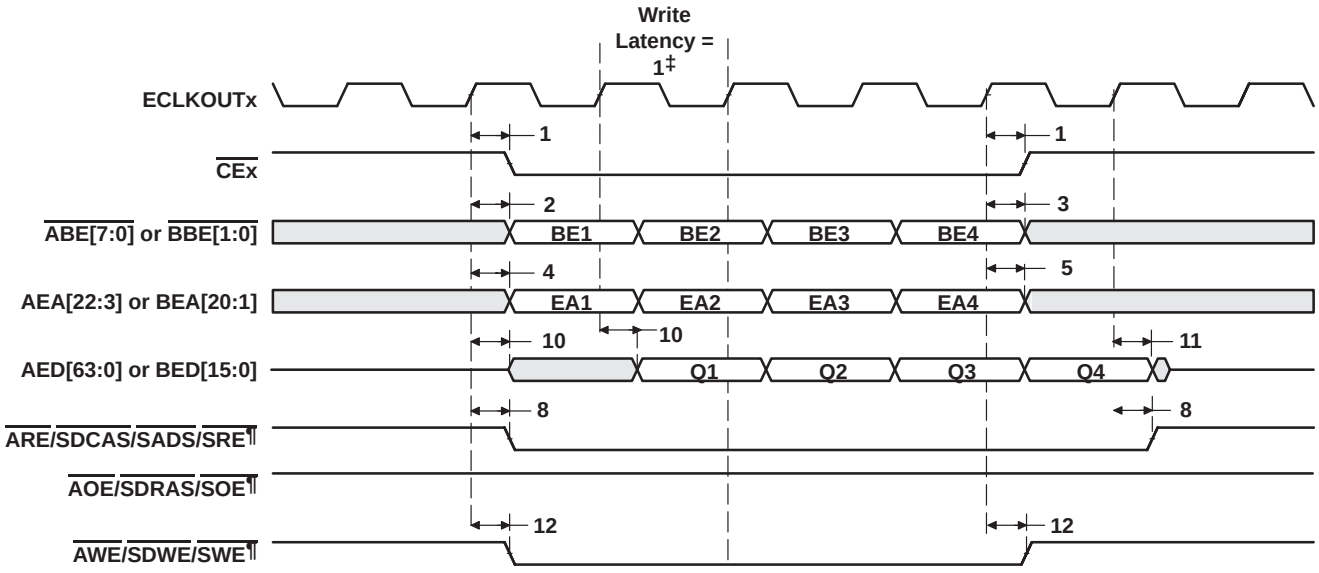
§ The following parameters are programmable via the EMIF CE Space Secondary Control register (CExSEC):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- CEx assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, CEx goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, CEx is active when SOE is active (CEEXT = 1).
- Function of SADS/SRE (RENEN): For standard SBSRAM or ZBT SRAM interface, SADS/SRE acts as SADS with deselect cycles (RENEN = 0). For FIFO interface, SADS/SRE acts as SRE with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

¶ ARE/SDCAS/SADS/SRE, AOE/SDRAS/SOE, and AWE/SDWE/SWE operate as SADS/SRE, SOE, and SWE, respectively, during programmable synchronous interface accesses.

Figure 25. Programmable Synchronous Interface Write Timing for EMIFA and EMIFB (With Write Latency = 0)†‡§

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the programmable synchronous interface access signals are shown as generic (SADS/SRE, SOE, and SWE) instead of ASADS/ASRE, ASOE, and ASWE (for EMIFA) and BSADS/BSRE, BSOE, and BSWE (for EMIFB)].

‡ The write latency and the length of CEx assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIFx CE Space Secondary Control register (CExSEC). In this figure, SYNCWL = 1 and CEEXT = 0.

§ The following parameters are programmable via the EMIF CE Space Secondary Control register (CExSEC):

- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- CEx assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, CEx goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, CEx is active when SOE is active (CEEXT = 1).
- Function of SADS/SRE (RENEN): For standard SBSRAM or ZBT SRAM interface, SADS/SRE acts as SADS with deselect cycles (RENEN = 0). For FIFO interface, SADS/SRE acts as SRE with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

¶ ARE/SDCAS/SADS/SRE, AOE/SDRAS/SOE, and AWE/SDWE/SWE operate as SADS/SRE, SOE, and SWE, respectively, during programmable synchronous interface accesses.

Figure 26. Programmable Synchronous Interface Write Timing for EMIFA and EMIFB (With Write Latency = 1)†‡§

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles for EMIFA module† (see Figure 27)

NO.			MIN	MAX	UNIT
6	$t_{su(EDV-EKO1H)}$	Setup time, read EDx valid before ECLKOUTx high	0.6		ns
7	$t_{h(EKO1H-EDV)}$	Hold time, read EDx valid after ECLKOUTx high	1.8		ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic ($\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$) instead of $\overline{\text{ASDCAS}}$, $\overline{\text{ASDWE}}$, and $\overline{\text{ASDRAS}}$ (for EMIFA) and $\overline{\text{BSDCAS}}$, $\overline{\text{BSDWE}}$, and $\overline{\text{BSDRAS}}$ (for EMIFB)].

switching characteristics over recommended operating conditions for synchronous DRAM cycles for EMIFA module† (see Figure 27−Figure 34)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(EKO1H-CEV)}$	Delay time, ECLKOUTx high to $\overline{\text{CE}}$x valid	1.3	4.9	ns
2	$t_{d(EKO1H-BEV)}$	Delay time, ECLKOUTx high to $\overline{\text{BE}}$x valid		4.9	ns
3	$t_{d(EKO1H-BEIV)}$	Delay time, ECLKOUTx high to $\overline{\text{BE}}$x invalid	1.3		ns
4	$t_{d(EKO1H-EAV)}$	Delay time, ECLKOUTx high to EAx valid		4.9	ns
5	$t_{d(EKO1H-EAIV)}$	Delay time, ECLKOUTx high to EAx invalid	1.3		ns
8	$t_{d(EKO1H-CASV)}$	Delay time, ECLKOUTx high to $\overline{\text{SDCAS}}$ valid	1.3	4.9	ns
9	$t_{d(EKO1H-EDV)}$	Delay time, ECLKOUTx high to $\overline{\text{ED}}$x valid		4.9	ns
10	$t_{d(EKO1H-EDIV)}$	Delay time, ECLKOUTx high to $\overline{\text{ED}}$x invalid	1.3		ns
11	$t_{d(EKO1H-WEV)}$	Delay time, ECLKOUTx high to $\overline{\text{SDWE}}$ valid	1.3	4.9	ns
12	$t_{d(EKO1H-RAS)}$	Delay time, ECLKOUTx high to $\overline{\text{SDRAS}}$ valid	1.3	4.9	ns
13	$t_{d(EKO1H-ACKEV)}$	Delay time, ECLKOUTx high to ASDCKE valid (EMIFA only)	1.3	4.9	ns
14	$t_{d(EKO1H-PDTV)}$	Delay time, ECLKOUTx high to $\overline{\text{PDT}}$ valid	1.3	4.9	ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic ($\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$) instead of $\overline{\text{ASDCAS}}$, $\overline{\text{ASDWE}}$, and $\overline{\text{ASDRAS}}$ (for EMIFA) and $\overline{\text{BSDCAS}}$, $\overline{\text{BSDWE}}$, and $\overline{\text{BSDRAS}}$ (for EMIFB)].

SYNCHRONOUS DRAM TIMING (CONTINUED)

timing requirements for synchronous DRAM cycles for EMIFB module† (see Figure 27)

NO.			MIN	MAX	UNIT
6	$t_{su(EDV-EKO1H)}$	Setup time, read EDx valid before ECLKOUTx high	2.1		ns
7	$t_{h(EKO1H-EDV)}$	Hold time, read EDx valid after ECLKOUTx high	2.5		ns

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic ($\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$) instead of $\overline{\text{ASDCAS}}$, $\overline{\text{ASDWE}}$, and $\overline{\text{ASDRAS}}$ (for EMIFA) and $\overline{\text{BSDCAS}}$, $\overline{\text{BSDWE}}$, and $\overline{\text{BSDRAS}}$ (for EMIFB)].

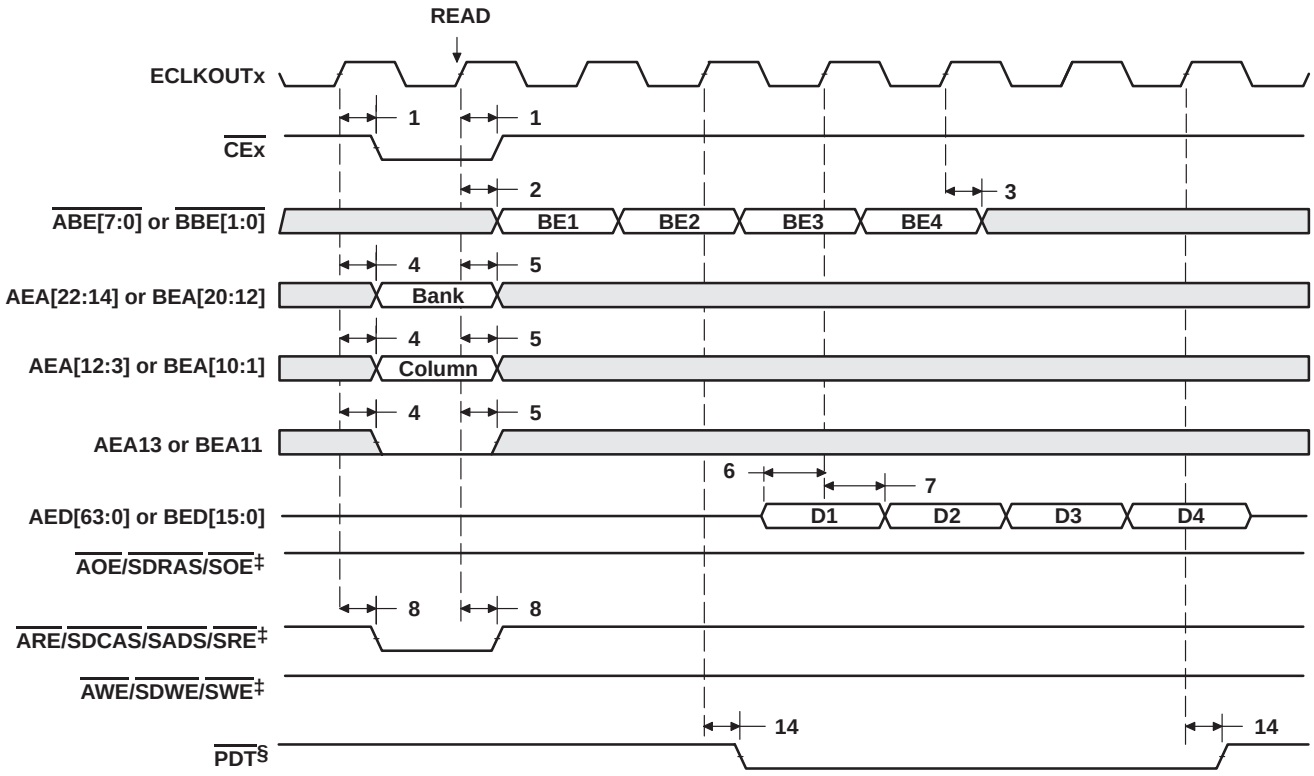
switching characteristics over recommended operating conditions for synchronous DRAM cycles for EMIFB module† (see Figure 27−Figure 34)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(EKO1H-CEV)}$	Delay time, ECLKOUTx high to $\overline{\text{CEx}}$ valid	1.3	6.4	ns
2	$t_{d(EKO1H-BEV)}$	Delay time, ECLKOUTx high to $\overline{\text{BEx}}$ valid		6.4	ns
3	$t_{d(EKO1H-BEIV)}$	Delay time, ECLKOUTx high to $\overline{\text{BEx}}$ invalid	1.3		ns
4	$t_{d(EKO1H-EAV)}$	Delay time, ECLKOUTx high to EAx valid		6.4	ns
5	$t_{d(EKO1H-EAIV)}$	Delay time, ECLKOUTx high to EAx invalid	1.3		ns
8	$t_{d(EKO1H-CASV)}$	Delay time, ECLKOUTx high to $\overline{\text{SDCAS}}$ valid	1.3	6.4	ns
9	$t_{d(EKO1H-EDV)}$	Delay time, ECLKOUTx high to $\overline{\text{EDx}}$ valid		6.4	ns
10	$t_{d(EKO1H-EDIV)}$	Delay time, ECLKOUTx high to $\overline{\text{EDx}}$ invalid	1.3		ns
11	$t_{d(EKO1H-WEV)}$	Delay time, ECLKOUTx high to $\overline{\text{SDWE}}$ valid	1.3	6.4	ns
12	$t_{d(EKO1H-RAS)}$	Delay time, ECLKOUTx high to $\overline{\text{SDRAS}}$ valid	1.3	6.4	ns
13	$t_{d(EKO1H-ACKEV)}$	Delay time, ECLKOUTx high to ASDCKE valid (EMIFA only)	1.3*	6.4*	ns
14	$t_{d(EKO1H-PDTV)}$	Delay time, ECLKOUTx high to $\overline{\text{PDT}}$ valid	1.3	6.4	ns

*This parameter is not production tested.

† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic ($\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$) instead of $\overline{\text{ASDCAS}}$, $\overline{\text{ASDWE}}$, and $\overline{\text{ASDRAS}}$ (for EMIFA) and $\overline{\text{BSDCAS}}$, $\overline{\text{BSDWE}}$, and $\overline{\text{BSDRAS}}$ (for EMIFB)].

SYNCHRONOUS DRAM TIMING (CONTINUED)



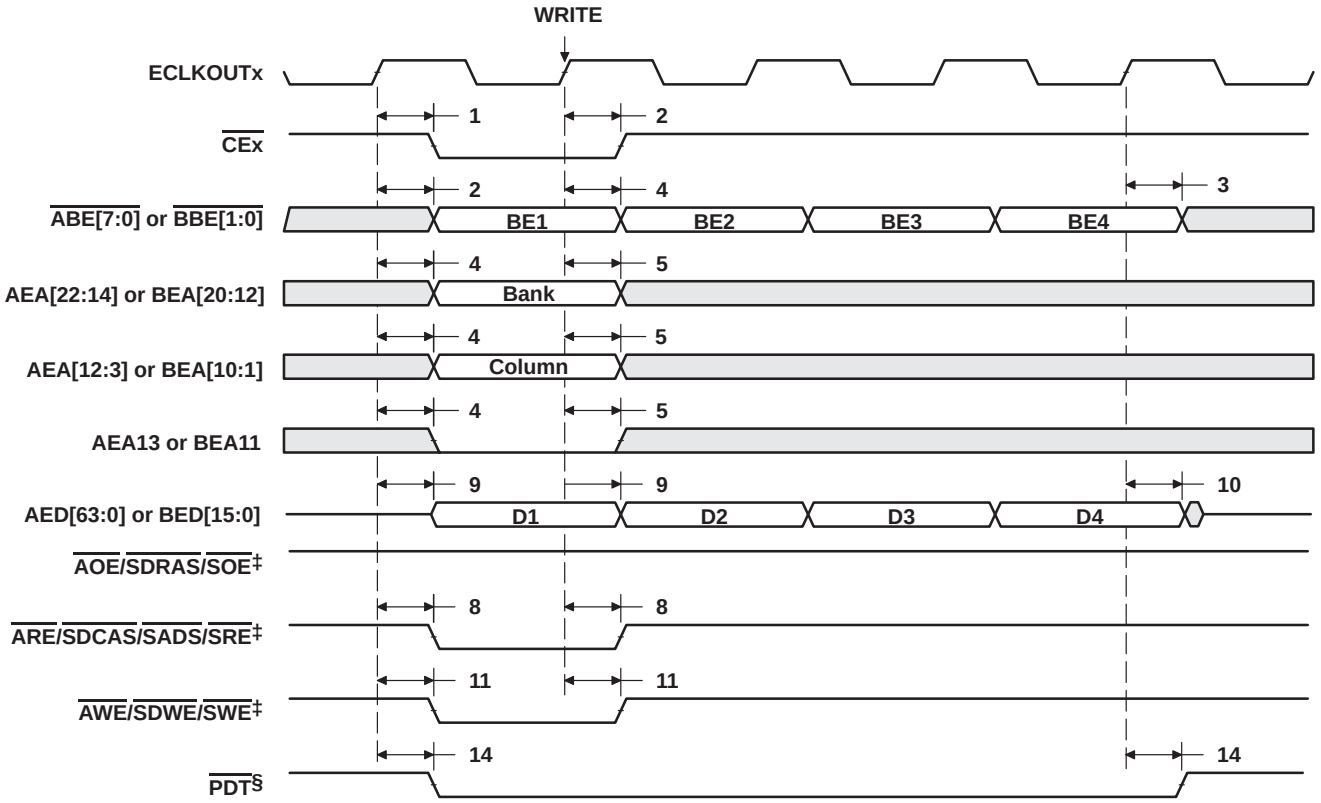
† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

§ PDT signal is only asserted when the EDMA is in PDT mode (set the PDTS bit to 1 in the EDMA options parameter RAM). For PDT read, data is not latched into EMIF. The PDTRL field in the PDT control register (PDTCTL) configures the latency of the PDT signal with respect to the data phase of a read transaction. The latency of the PDT signal for a read can be programmed to 0, 1, 2, or 3 by setting PDTRL to 00, 01, 10, or 11, respectively. PDTRL equals 00 (zero latency) in Figure 27.

Figure 27. SDRAM Read Command (CAS Latency 3) for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)



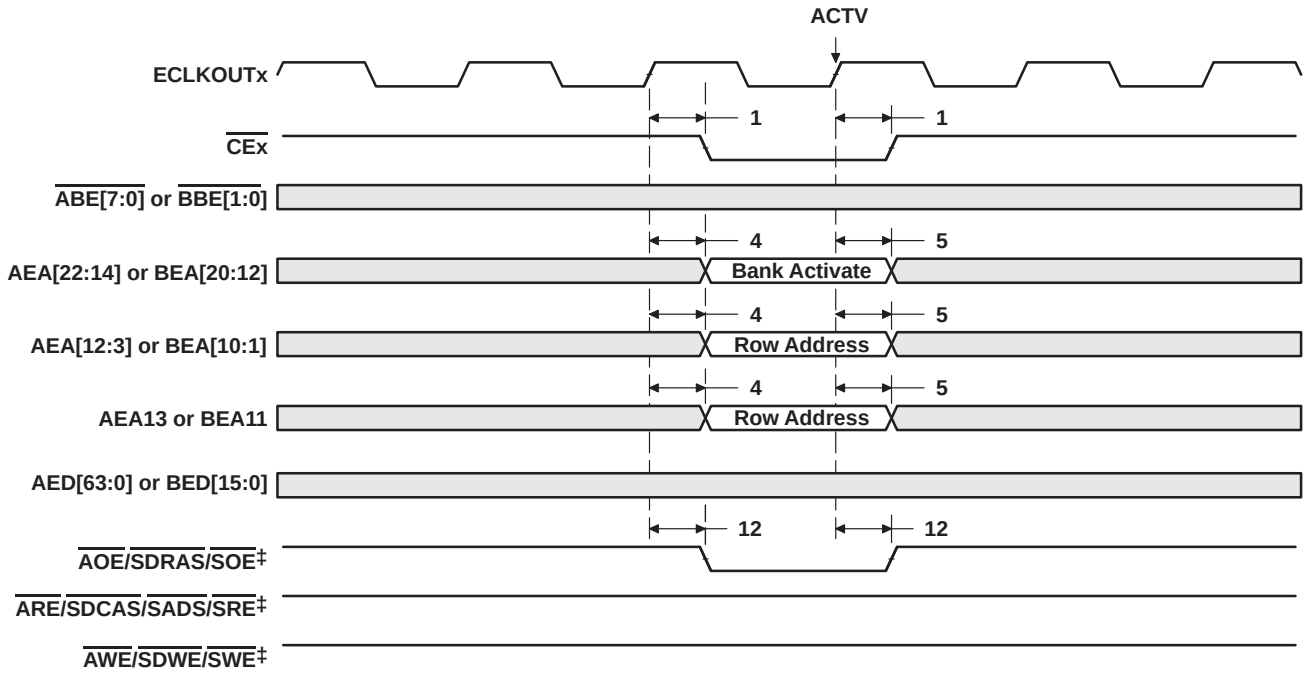
† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

§ PDT signal is only asserted when the EDMA is in PDT mode (set the PDTD bit to 1 in the EDMA options parameter RAM). For PDT write, data is not driven (in High-Z). The PDTWL field in the PDT control register (PDTCTL) configures the latency of the PDT signal with respect to the data phase of a write transaction. The latency of the PDT signal for a write transaction can be programmed to 0, 1, 2, or 3 by setting PDTWL to 00, 01, 10, or 11, respectively. PDTWL equals 00 (zero latency) in Figure 28.

Figure 28. SDRAM Write Command for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)

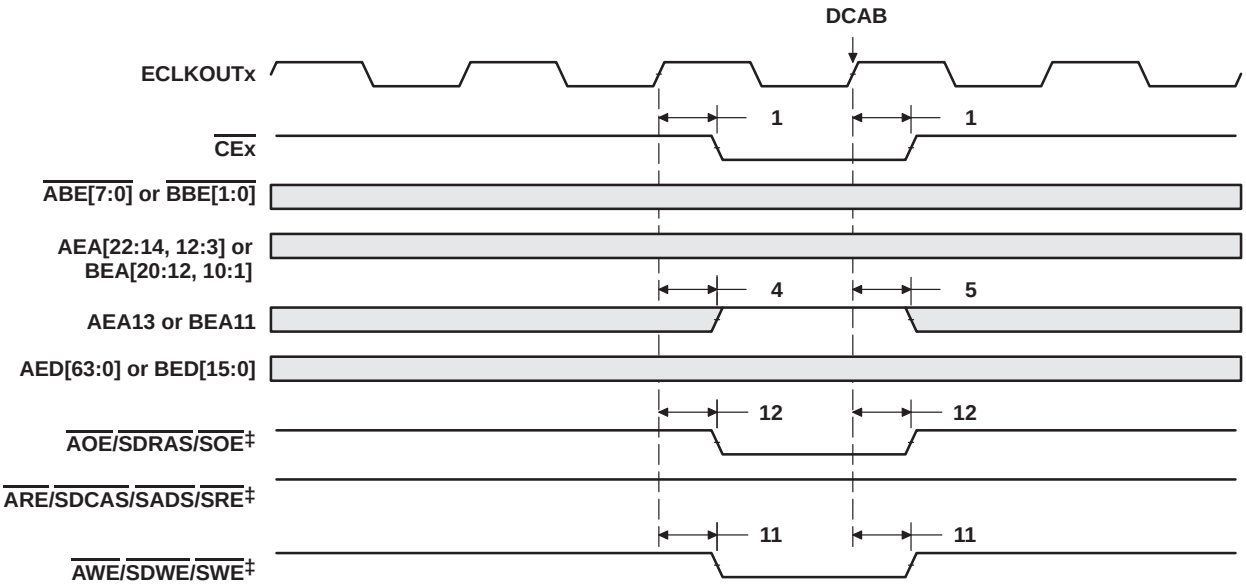


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 29. SDRAM ACTV Command for EMIFA and EMFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)

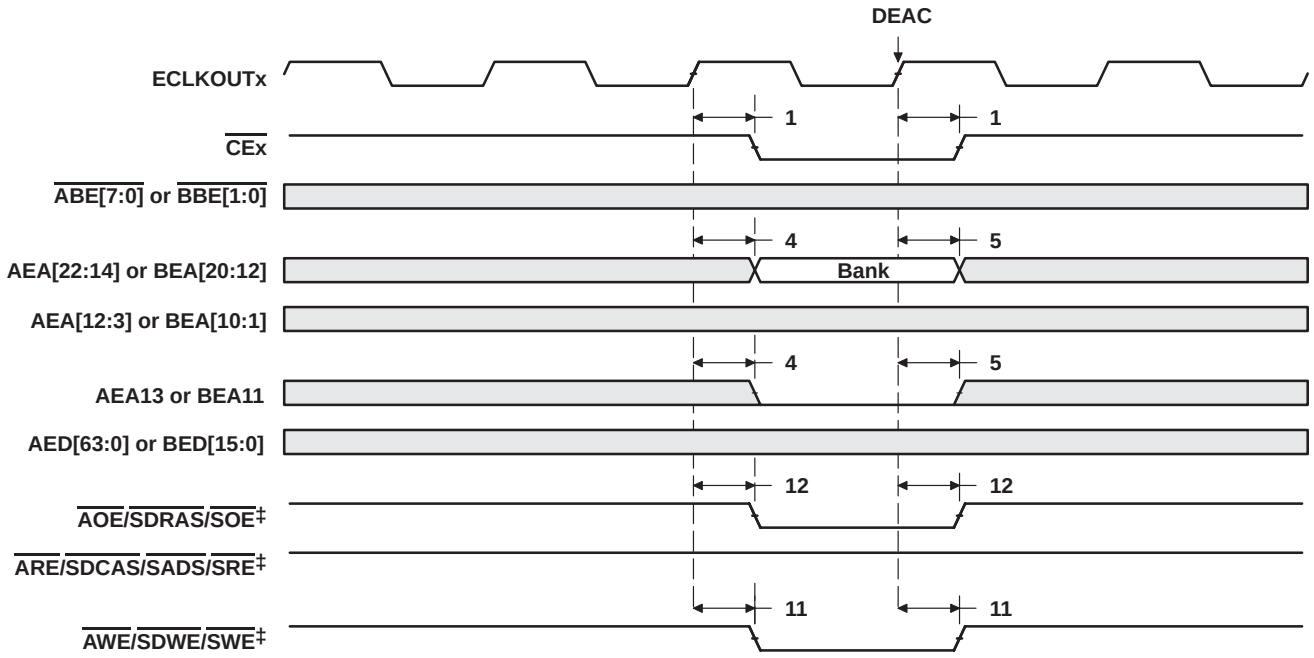


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic ($\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$) instead of $\overline{\text{ASDCAS}}$, $\overline{\text{ASDWE}}$, and $\overline{\text{ASDRAS}}$ (for EMIFA) and $\overline{\text{BSDCAS}}$, $\overline{\text{BSDWE}}$, and $\overline{\text{BSDRAS}}$ (for EMIFB)].

‡ $\overline{\text{ARE}}/\overline{\text{SDCAS}}/\overline{\text{SADS}}/\overline{\text{SRE}}$, $\overline{\text{AWE}}/\overline{\text{SDWE}}/\overline{\text{SWE}}$, and $\overline{\text{AOE}}/\overline{\text{SDRAS}}/\overline{\text{SOE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$, respectively, during SDRAM accesses.

Figure 30. SDRAM DCAB Command for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)

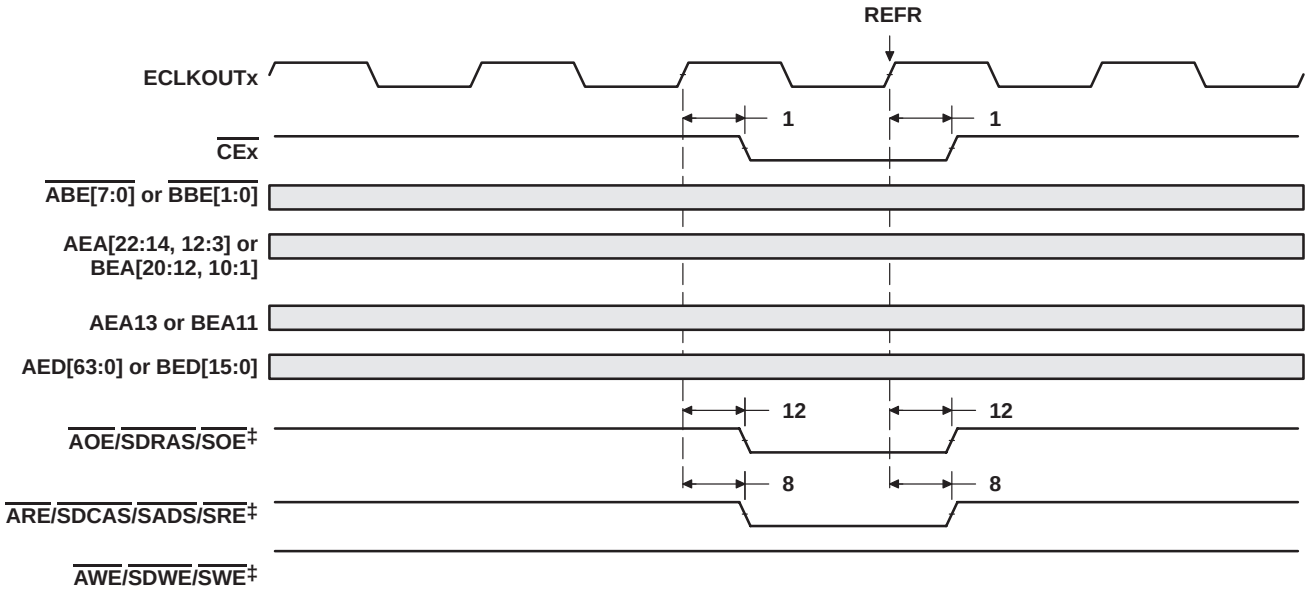


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 31. SDRAM DEAC Command for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)

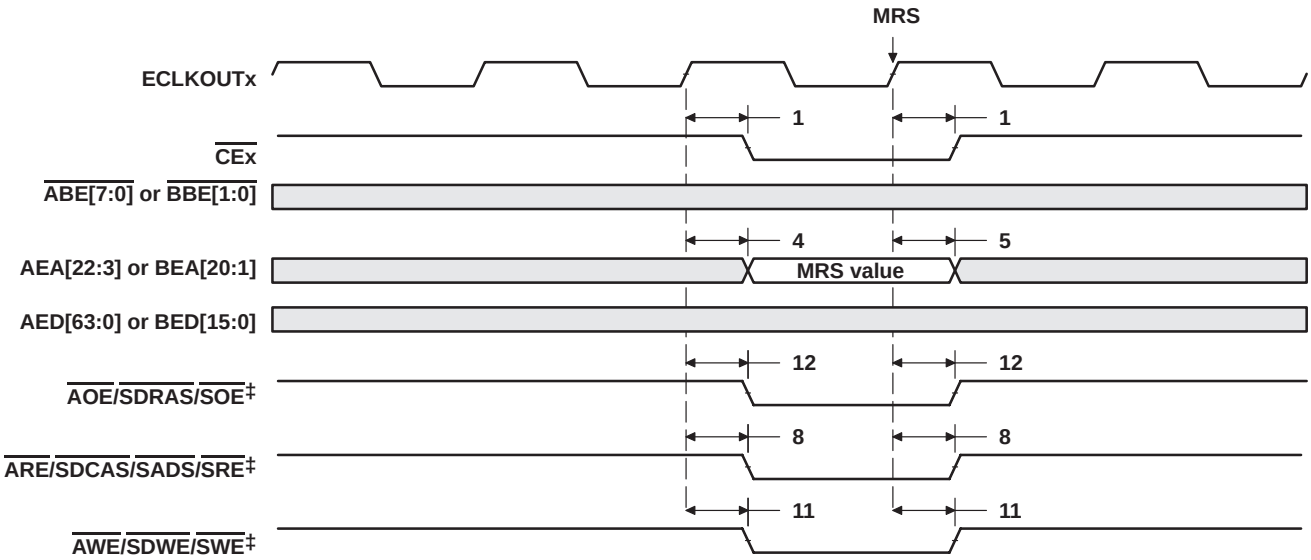


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 32. SDRAM REFR Command for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)

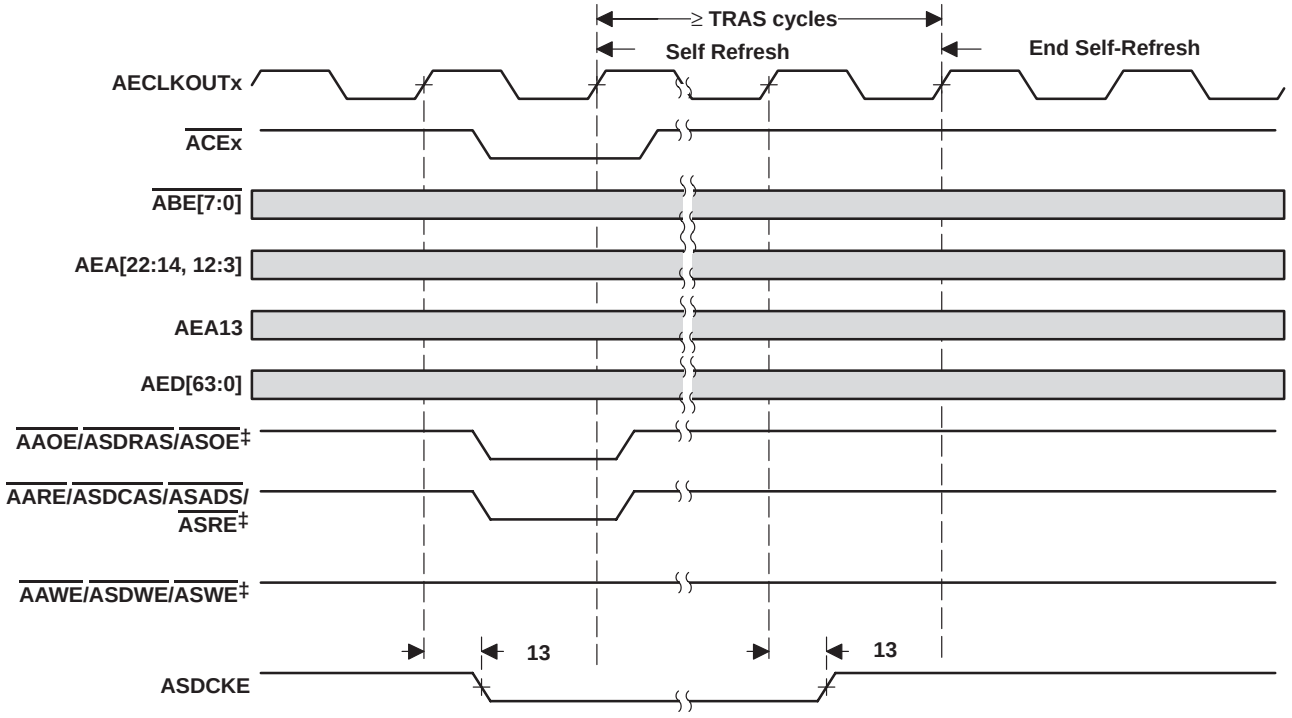


† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 33. SDRAM MRS Command for EMIFA and EMIFB†

SYNCHRONOUS DRAM TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown as generic (SDCAS, SDWE, and SDRAS) instead of ASDCAS, ASDWE, and ASDRAS (for EMIFA) and BSDCAS, BSDWE, and BSDRAS (for EMIFB)].

‡ AARE/ASDCAS/ASADS/ASRE, AAWE/ASDWE/ASWE, and AAOE/ASDRAS/ASOE operate as ASDCAS, ASDWE, and ASDRAS, respectively, during SDRAM accesses.

Figure 34. SDRAM Self-Refresh Timing for EMIFA Only†

HOLD/HOLDA TIMING

timing requirements for the HOLD/HOLDA cycles for EMIFA and EMIFB modules† (see Figure 35)

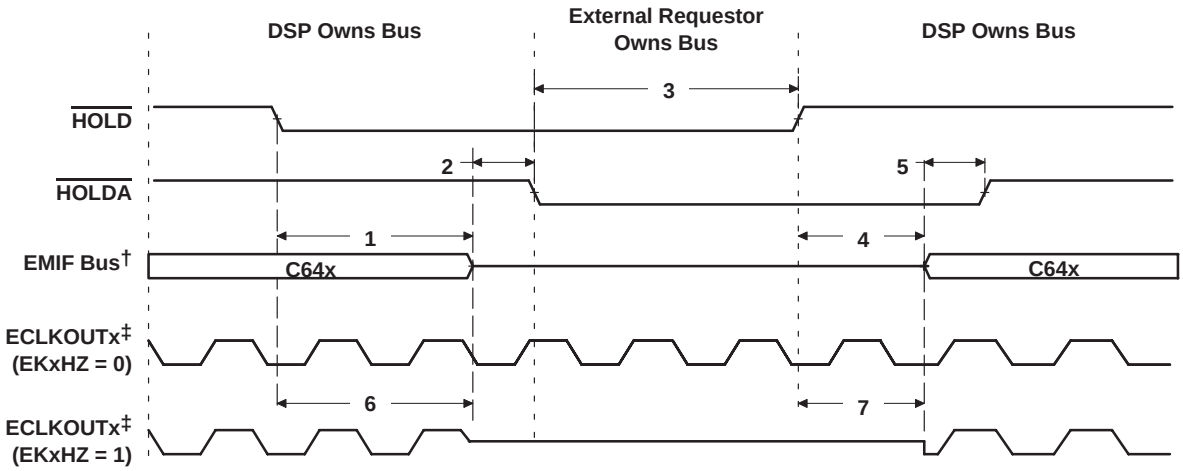
NO.			MIN	MAX	UNIT
3	$t_{oh(HOLDAL-HOLDL)}$	Hold time, HOLD low after HOLDA low	E*		ns

*This parameter is not production tested.
† E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.

switching characteristics over recommended operating conditions for the HOLD/HOLDA cycles for EMIFA and EMIFB modules†‡§ (see Figure 35)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(HOLDL-EMHZ)}$	Delay time, HOLD low to EMIF Bus high impedance	2E*	¶*	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	0*	2E*	ns
4	$t_{d(HOLDH-EMLZ)}$	Delay time, HOLD high to EMIF Bus low impedance	2E*	7E*	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	0*	2E*	ns
6	$t_{d(HOLDL-EKOHZ)}$	Delay time, HOLD low to ECLKOUTx high impedance	2E*	¶*	ns
7	$t_{d(HOLDH-EKOLZ)}$	Delay time, HOLD high to ECLKOUTx low impedance	2E*	7E*	ns

*This parameter is not production tested.
† E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
‡ For EMIFA, EMIF Bus consists of: ACE[3:0], ABE[7:0], AED[63:0], AEA[22:3], AARE/ASDCAS/ASADS/ASRE, AAOE/ASDRAS/ASOE, and AAWE/ASDWE/ASWE , ASDCKE, ASOE3, and APDT.
For EMIFB, EMIF Bus consists of: BCE[3:0], BBE[1:0], BED[15:0], BEA[20:1], BARE/BSDCAS/BSADS/BSRE, BAOE/BSDRAS/BSOE, and BAWE/BSDWE/BSWE, BSOE3, and BPDT.
§ The EKxHZ bits in the EMIF Global Control register (GBLCTL) determine the state of the ECLKOUTx signals during HOLDA. If EKxHZ = 0, ECLKOUTx continues clocking during Hold mode. If EKxHZ = 1, ECLKOUTx goes to high impedance during Hold mode, as shown in Figure 35.
¶ All pending EMIF transactions are allowed to complete before HOLDA is asserted. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.



† For EMIFA, EMIF Bus consists of: ACE[3:0], ABE[7:0], AED[63:0], AEA[22:3], AARE/ASDCAS/ASADS/ASRE, AAOE/ASDRAS/ASOE, and AAWE/ASDWE/ASWE, ASDCKE, ASOE3, and APDT.
For EMIFB, EMIF Bus consists of: BCE[3:0], BBE[1:0], BED[15:0], BEA[20:1], BARE/BSDCAS/BSADS/BSRE, BAOE/BSDRAS/BSOE, and BAWE/BSDWE/BSWE, BSOE3, and BPDT.
‡ The EKxHZ bits in the EMIF Global Control register (GBLCTL) determine the state of the ECLKOUTx signals during HOLDA. If EKxHZ = 0, ECLKOUTx continues clocking during Hold mode. If EKxHZ = 1, ECLKOUTx goes to high impedance during Hold mode, as shown in Figure 35.

Figure 35. HOLD/HOLDA Timing for EMIFA and EMIFB

BUSREQ TIMING

switching characteristics over recommended operating conditions for the BUSREQ cycles for EMIFA and EMIFB modules (see Figure 36)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(AEKO1H-ABUSRV)}$	Delay time, AECLKOUTx high to ABUSREQ valid	1	5.5	ns
2	$t_{d(BEKO1H-BBUSRV)}$	Delay time, BECLKOUTx high to BBUSREQ valid	0.9	5.5	ns

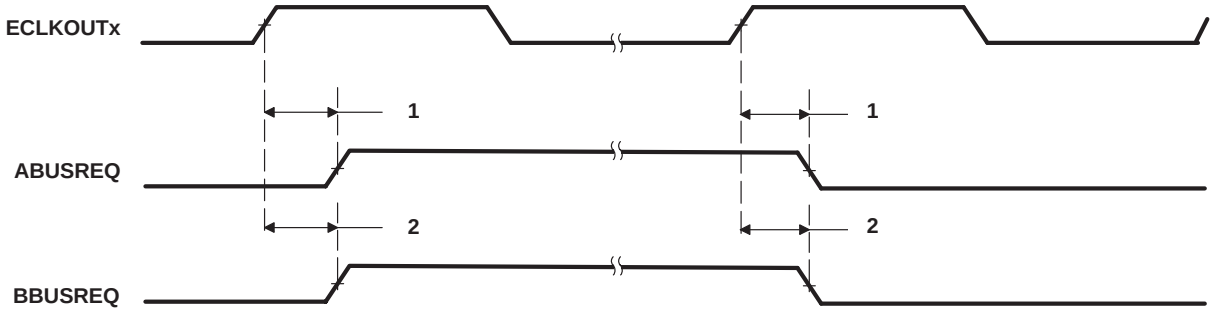


Figure 36. BUSREQ Timing for EMIFA and EMIFB

RESET TIMING

timing requirements for reset† (see Figure 37)

NO.			MIN	MAX	UNIT
1	$t_{w(RST)}$	Width of the RESET pulse (PLL stable)‡	10P*		ns
		Width of the RESET pulse (PLL needs to sync up)§	250*		µs
16	$t_{su(boot)}$	Setup time, boot configuration bits valid before RESET high¶	4P*		ns
17	$t_{h(boot)}$	Hold time, boot configuration bits valid after RESET high¶	4P*		ns

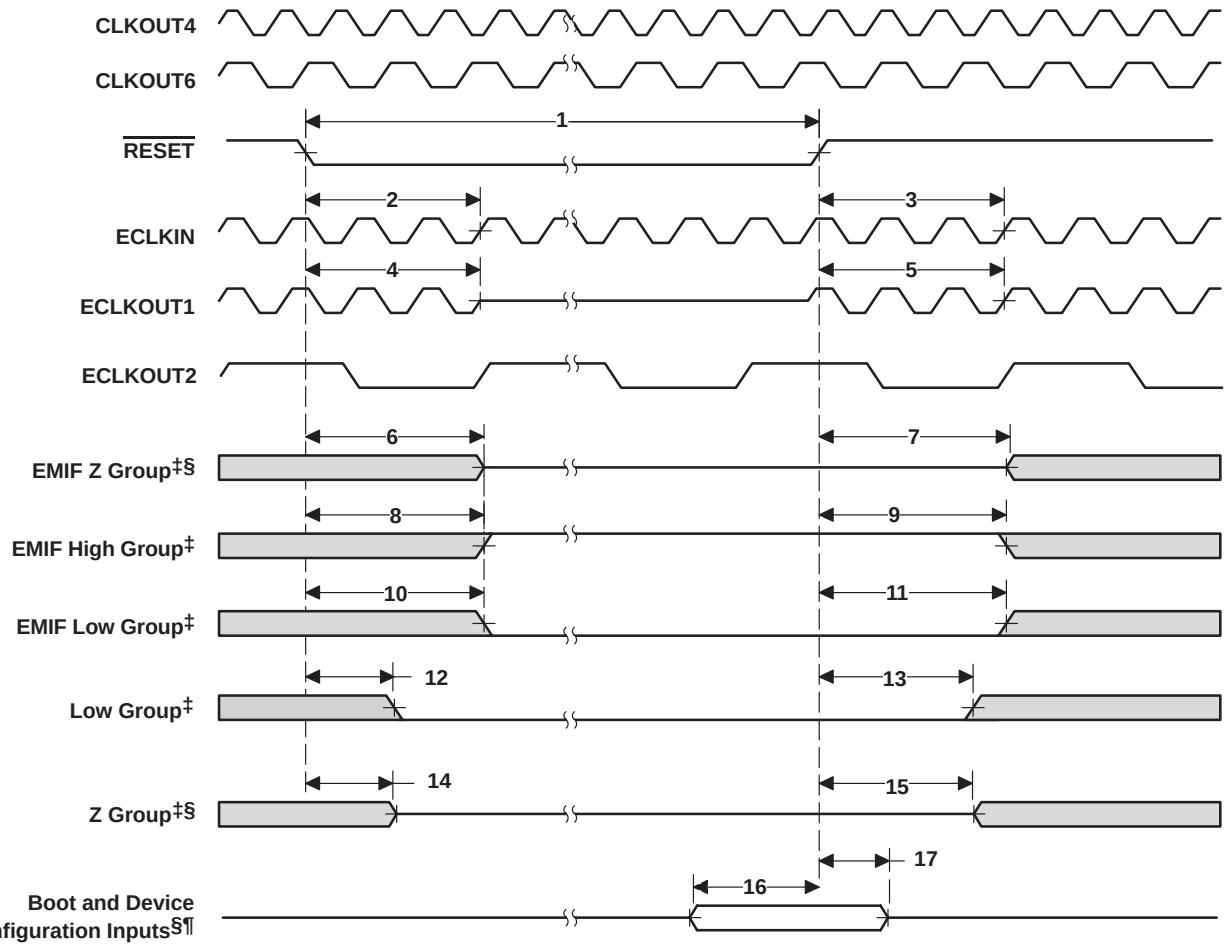
*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ This parameter applies to CLKMODE x1 when CLKIN is stable, and applies to CLKMODE x6, x12 when CLKIN and PLL are stable.
§ This parameter applies to CLKMODE x6, x12 only (it does not apply to CLKMODE x1). The RESET signal is not connected internally to the clock PLL circuit. The PLL, however, may need up to 250 µs to stabilize following device power up or after PLL configuration has been changed. During that time, RESET must be asserted to ensure proper device operation. See the *clock PLL* section for PLL lock times.
¶ EMIFB address pins BEA[20:13, 11, 7] are the boot configuration pins during device reset.

switching characteristics over recommended operating conditions during reset†#‖ (see Figure 37)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(RSTL-ECKI)}$	Delay time, RESET low to ECLKIN synchronized internally	2E*	3P + 20E*	ns
3	$t_{d(RSTH-ECKI)}$	Delay time, RESET high to ECLKIN synchronized internally	2E*	8P + 20E*	ns
4	$t_{d(RSTL-ECKO1HZ)}$	Delay time, RESET low to ECLKOUT1 high impedance	2E*		ns
5	$t_{d(RSTH-ECKO1V)}$	Delay time, RESET high to ECLKOUT1 valid		8P + 20E*	ns
6	$t_{d(RSTL-EMIFZHZ)}$	Delay time, RESET low to EMIF Z high impedance	2E*	3P + 4E*	ns
7	$t_{d(RSTH-EMIFZV)}$	Delay time, RESET high to EMIF Z valid	16E*	8P + 20E*	ns
8	$t_{d(RSTL-EMIFHIV)}$	Delay time, RESET low to EMIF high group invalid	2E*		ns
9	$t_{d(RSTH-EMIFHV)}$	Delay time, RESET high to EMIF high group valid		8P + 20E*	ns
10	$t_{d(RSTL-EMIFLIV)}$	Delay time, RESET low to EMIF low group invalid	2E*		ns
11	$t_{d(RSTH-EMIFLV)}$	Delay time, RESET high to EMIF low group valid		8P + 20E*	ns
12	$t_{d(RSTL-LOWIV)}$	Delay time, RESET low to low group invalid	0*		ns
13	$t_{d(RSTH-LOWV)}$	Delay time, RESET high to low group valid		11P*	ns
14	$t_{d(RSTL-ZHZ)}$	Delay time, RESET low to Z group high impedance	0*		ns
15	$t_{d(RSTH-ZV)}$	Delay time, RESET high to Z group valid	2P*	8P*	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
# E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
‖ EMIF Z group consists of: AEA[22:3], BEA[20:1], AED[63:0], BED[15:0], CE[3:0], ABE[7:0], BBE[1:0], ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE, SOE3, ASDCKE, and PDT.
EMIF high group consists of: AHOLDA and BHOLDA (when the corresponding HOLD input is high)
EMIF low group consists of: ABUSREQ and BBUSREQ; AHOLDA and BHOLDA (when the corresponding HOLD input is low)
Low group consists of: XSP_CS, CLKX2/XSP_CLK, and DX2/XSP_DO; all of which apply only when PCI EEPROM (BEA13) is enabled (with PCI_EN = 1 and MCBSP2_EN = 0). Otherwise, the CLKX2/XSP_CLK and DX2/XSP_DO pins are in the Z group. For more details on the PCI configuration pins, see the Device Configurations section of this data sheet.
Z group consists of: HD[31:0]/AD[31:0], CLKX0, CLKX1/URADDR4, CLKX2/XSP_CLK, FSX0, FSX1/UXADDR3, FSX2, DX0, DX1/UXADDR4, DX2/XSP_DO, CLKR0, CLKR1/URADDR2, CLKR2, FSR0, FSR1/UXADDR2, FSR2, TOUT0, TOUT1, TOUT2, GP[8:0], GP10/PCBE3, HR/W/PCBE2, HDS2/PCBE1, PCBE0, GP13/PINTA, GP11/PREQ, HDS1/PSERR, HCS/PPERR, HCNTL1/PDEVSEL, HAS/PPAR, HCNTL0/PSTOP, HHWIL/PTRDY (16-bit HPI mode only), HRDY/PIRDY, HINT/PFRAME, UXDATA[7:0], UXSOC, UXCLAV, and URCLAV.

RESET TIMING (CONTINUED)



† These C64x™ devices have two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., ECLKIN, ECLKOUT1, and ECLKOUT2].

‡ EMIF Z group consists of: AEA[22:3], BEA[20:1], AED[63:0], BED[15:0], $\overline{\text{CE}}$[3:0], $\overline{\text{ABE}}$[7:0], $\overline{\text{BBE}}$[1:0], $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SADS}}$/$\overline{\text{SRE}}$, $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SWE}}$, and AOE/$\overline{\text{SDRAS}}$/$\overline{\text{SOE}}$, $\overline{\text{SOE3}}$, ASDCKE, and $\overline{\text{PDT}}$.

EMIF high group consists of: $\overline{\text{AHOLDA}}$ and $\overline{\text{BHOLDA}}$ (when the corresponding $\overline{\text{HOLD}}$ input is high)

EMIF low group consists of: ABUSREQ and BBUSREQ; $\overline{\text{AHOLDA}}$ and $\overline{\text{BHOLDA}}$ (when the corresponding $\overline{\text{HOLD}}$ input is low)

Low group consists of: XSP_CS, CLKX2/XSP_CLK, and DX2/XSP_DO; all of which apply only when PCI EEPROM (BEA13) is enabled (with PCI_EN = 1 and MCBSP2_EN = 0). Otherwise, the CLKX2/XSP_CLK and DX2/XSP_DO pins are in the Z group. For more details on the PCI configuration pins, see the Device Configurations section of this data sheet.

Z group consists of: HD[31:0]/AD[31:0], CLKX0, CLKX1/URADDR4, CLKX2/XSP_CLK, FSX0, FSX1/UXADDR3, FSX2, DX0, DX1/UXADDR4, DX2/XSP_DO, CLKR0, CLKR1/URADDR2, CLKR2, FSR0, FSR1/UXADDR2, FSR2, TOUT0, TOUT1, TOUT2, GP[8:0], GP10/$\overline{\text{PCBE3}}$, HR/$\overline{\text{W}}$/$\overline{\text{PCBE2}}$, $\overline{\text{HDS2}}$/$\overline{\text{PCBE1}}$, $\overline{\text{PCBE0}}$, GP13/$\overline{\text{PINTA}}$, GP11/$\overline{\text{PREQ}}$, $\overline{\text{HDS1}}$/$\overline{\text{PSERR}}$, $\overline{\text{HCS}}$/$\overline{\text{PPERR}}$, $\overline{\text{HCNTL1}}$/$\overline{\text{PDEVSEL}}$, $\overline{\text{HAS}}$/PPAR, HCNTL0/$\overline{\text{PSTOP}}$, HHWIL/$\overline{\text{PTRDY}}$ (16-bit HPI mode only), $\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$, $\overline{\text{HINT}}$/$\overline{\text{PFRAME}}$, UXDATA[7:0], UXSOC, UXCLAV, and URCLAV.

§ If BEA[20:13, 11, 7] and HD5/AD5 pins are actively driven, care must be taken to ensure no timing contention between parameters 6, 7, 14, 15, 16, and 17.

¶ Boot and Device Configurations Inputs (during reset) include: EMIFB address pins BEA[20:13, 11, 7] and HD5/AD5.
The PCI_EN pin *must* be driven valid at all times and the user *must not* switch values throughout device operation.
The MCBSP2_EN pin *must* be driven valid at all times and the user *can* switch values throughout device operation.

Figure 37. Reset Timing†

EXTERNAL INTERRUPT TIMING

timing requirements for external interrupts† (see Figure 38)

			MIN	MAX	
1	$t_{w(ILOW)}$	Width of the NMI interrupt pulse low	4P*		ns
		Width of the EXT_INT interrupt pulse low	8P*		ns
2	$t_{w(IHIGH)}$	Width of the NMI interrupt pulse high	4P*		ns
		Width of the EXT_INT interrupt pulse high	8P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.

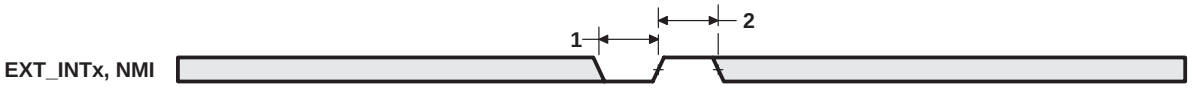


Figure 38. External/NMI Interrupt Timing

HOST-PORT INTERFACE (HPI) TIMING

timing requirements for host-port interface cycles†‡ (see Figure 39 through Figure 46)

NO.			MIN	MAX	UNIT
1	$t_{su(SELV-HSTBL)}$	Setup time, select signals§ valid before $\overline{\text{HSTROBE}}$ low	5		ns
2	$t_{h(HSTBL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HSTROBE}}$ low	2.4		ns
3	$t_{w(HSTBL)}$	Pulse duration, $\overline{\text{HSTROBE}}$ low	4P¶*		ns
4	$t_{w(HSTBH)}$	Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses	4P*		ns
10	$t_{su(SELV-HASL)}$	Setup time, select signals§ valid before $\overline{\text{HAS}}$ low	5		ns
11	$t_{h(HASL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HAS}}$ low	2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before $\overline{\text{HSTROBE}}$ high	5		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after $\overline{\text{HSTROBE}}$ high	2.8		ns
14	$t_{h(HRDYL-HSTBL)}$	Hold time, $\overline{\text{HSTROBE}}$ low after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly.	2*		ns
18	$t_{su(HASL-HSTBL)}$	Setup time, $\overline{\text{HAS}}$ low before $\overline{\text{HSTROBE}}$ low	2		ns
19	$t_{h(HSTBL-HASL)}$	Hold time, $\overline{\text{HAS}}$ low after $\overline{\text{HSTROBE}}$ low	2.1		ns

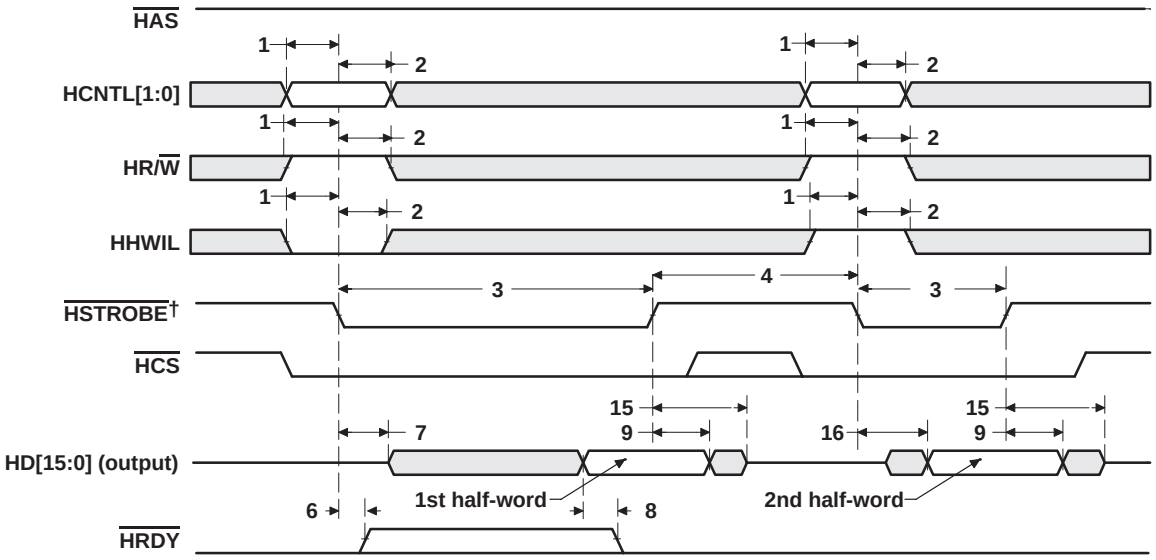
*This parameter is not production tested.
† $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
§ Select signals include: HCNTL[1:0] and HR/$\overline{\text{W}}$. For HPI16 mode only, select signals also include HHWIL.
¶ Select the parameter value of 4P or 12.5 ns, whichever is greater.

switching characteristics over recommended operating conditions during host-port interface cycles†‡ (see Figure 39 through Figure 46)

NO.	PARAMETER		MIN	MAX	UNIT
6	$t_{d(HSTBL-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high#	1.3	4P + 8	ns
7	$t_{d(HSTBL-HDLZ)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD low impedance for an HPI read	2*		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to $\overline{\text{HRDY}}$ low	−3		ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high	1.5		ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance		12*	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD valid (HPI16 only)		4P + 8	ns

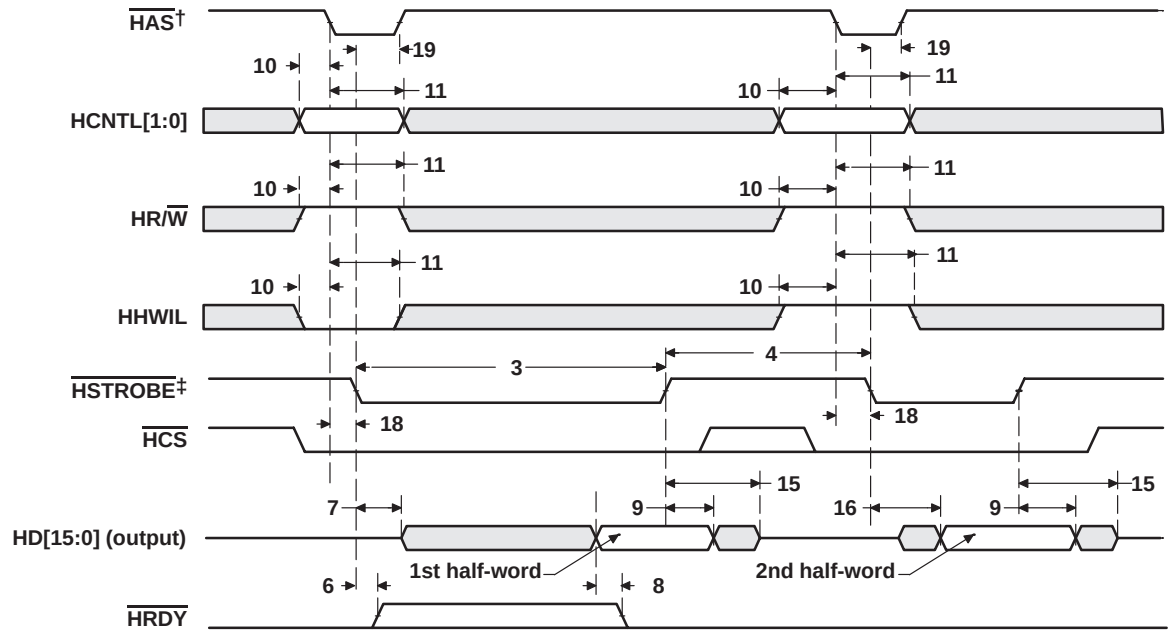
*This parameter is not production tested.
† $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
# This parameter is used during HPID reads and writes. For reads, at the beginning of a word transfer (HPI32) or the first half-word transfer (HPI16) on the falling edge of $\overline{\text{HSTROBE}}$, the HPI sends the request to the EDMA internal address generation hardware, and $\overline{\text{HRDY}}$ remains high until the EDMA internal address generation hardware loads the requested data into HPID. For writes, $\overline{\text{HRDY}}$ goes high if the internal write buffer is full.

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

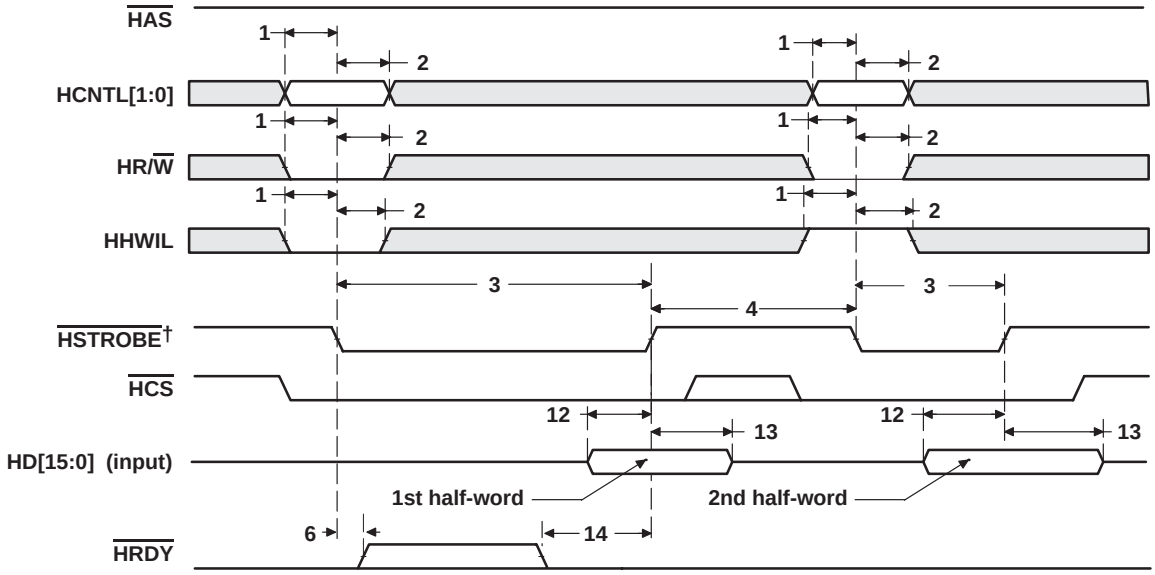
Figure 39. HPI16 Read Timing (HAS Not Used, Tied High)



† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

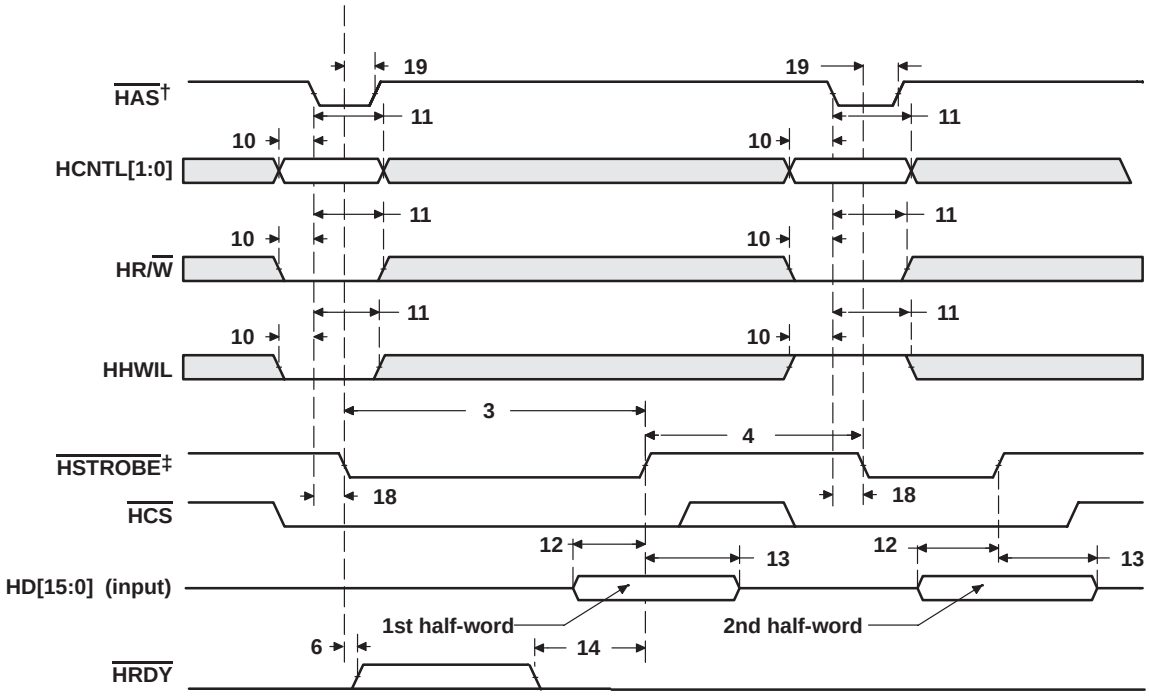
Figure 40. HPI16 Read Timing (HAS Used)

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

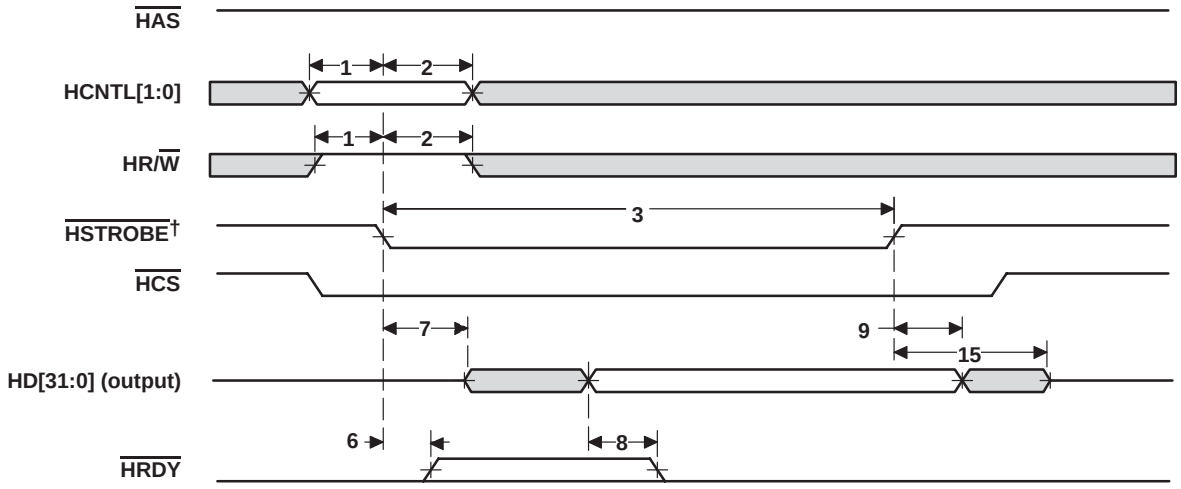
Figure 41. HPI16 Write Timing (HAS Not Used, Tied High)



† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

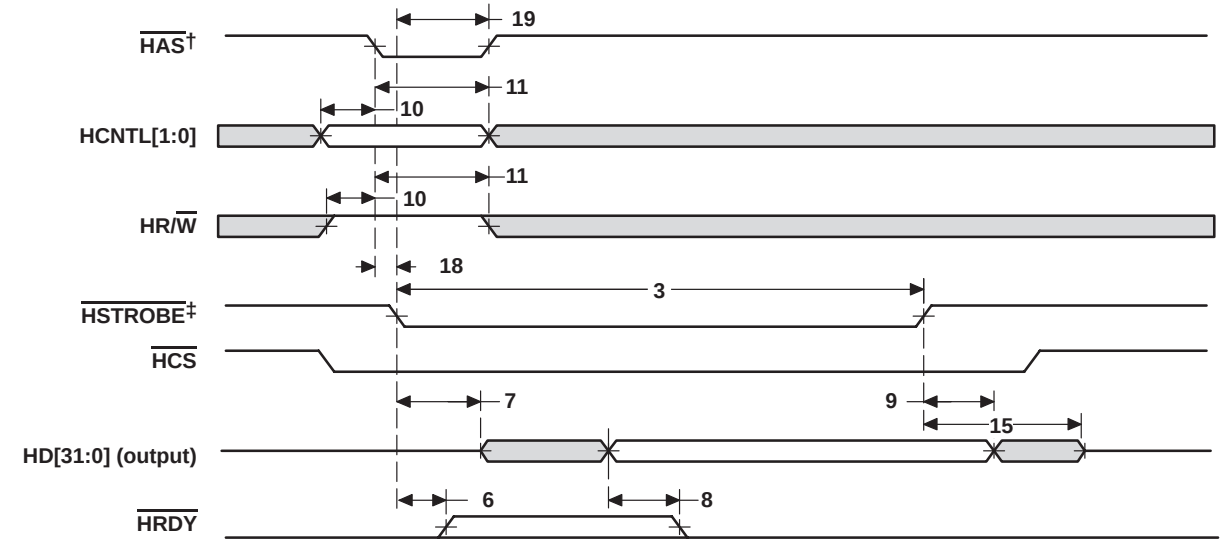
Figure 42. HPI16 Write Timing (HAS Used)

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

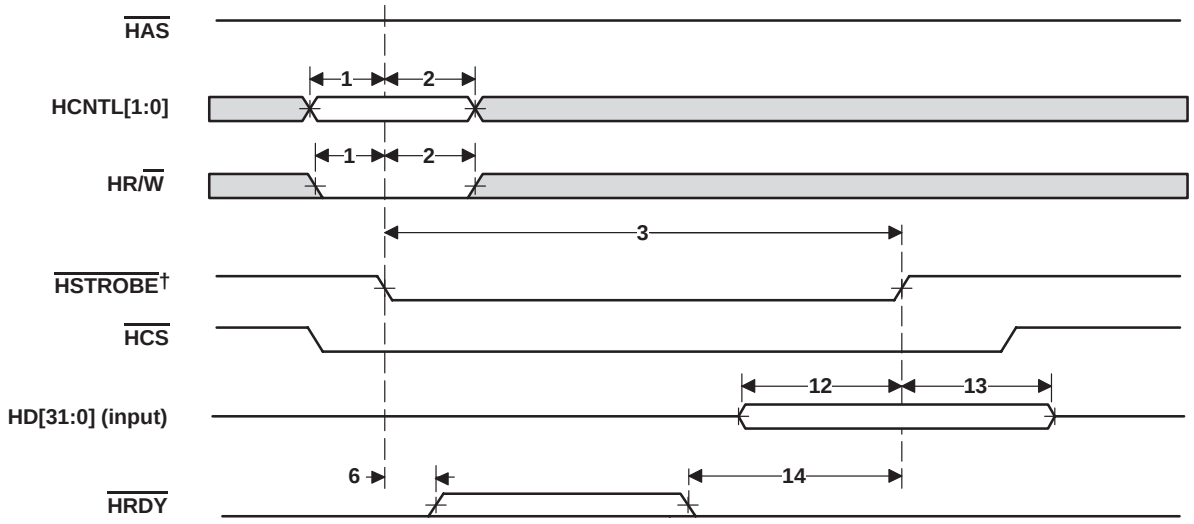
Figure 43. HPI32 Read Timing (HAS Not Used, Tied High)



† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

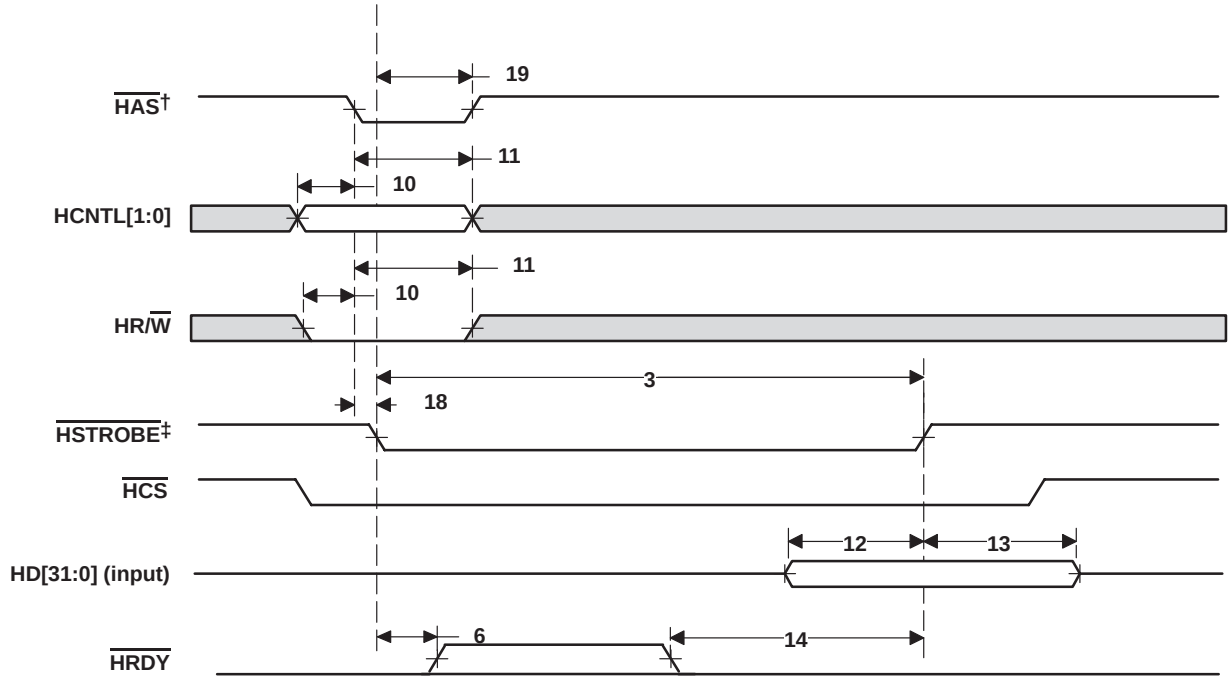
Figure 44. HPI32 Read Timing (HAS Used)

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

Figure 45. HPI32 Write Timing (HAS Not Used, Tied High)



† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

Figure 46. HPI32 Write Timing (HAS Used)

PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING [C6415 AND C6416 ONLY]

timing requirements for PCLK†‡ (see Figure 47)

NO.			MIN	MAX	UNIT
1	$t_{c(PCLK)}$	Cycle time, PCLK	30 (or 8P§)*		ns
2	$t_{w(PCLKH)}$	Pulse duration, PCLK high	11*		ns
3	$t_{w(PCLKL)}$	Pulse duration, PCLK low	11*		ns
4	$t_{sr(PCLK)}$	Δv/Δt slew rate, PCLK	1*	4*	V/ns

*This parameter is not production tested.
† For 3.3-V operation, the reference points for the rise and fall transitions are measured at V_{ILP} MAX and V_{IHP} MIN.
‡ P = 1/CPU clock frequency in ns. For example when running parts at 600 MHz, use P = 1.67 ns.
§ Select the parameter value of 30 ns or 8P, whichever is greater.

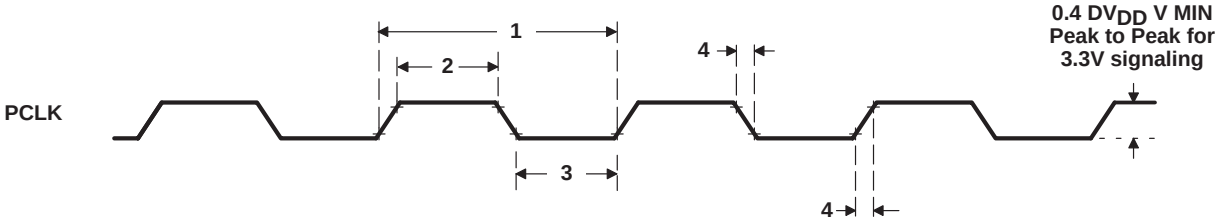


Figure 47. PCLK Timing

timing requirements for PCI reset (see Figure 48)

NO.			MIN	MAX	UNIT
1	$t_{w(PRST)}$	Pulse duration, $\overline{PRST}$	1*		ms
2	$t_{su(PCLKA-PRSTH)}$	Setup time, PCLK active before $\overline{PRST}$ high	100*		µs

*This parameter is not production tested.

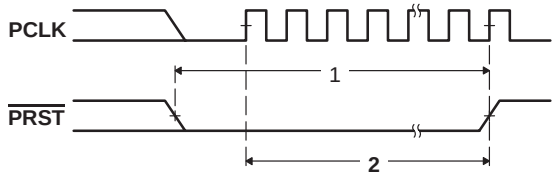


Figure 48. PCI Reset ($\overline{PRST}$) Timing

PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING [C6415 AND C6416 ONLY] (CONTINUED)

timing requirements for PCI inputs (see Figure 49)

NO.			MIN	MAX	UNIT
5	$t_{su(IV-PCLKH)}$	Setup time, input valid before PCLK high	7		ns
6	$t_{h(IV-PCLKH)}$	Hold time, input valid after PCLK high	0		ns

switching characteristics over recommended operating conditions for PCI outputs (see Figure 49)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(PCLKH-OV)}$	Delay time, PCLK high to output valid		11	ns
2	$t_{d(PCLKH-OIV)}$	Delay time, PCLK high to output invalid	2		ns
3	$t_{d(PCLKH-OLZ)}$	Delay time, PCLK high to output low impedance	2*		ns
4	$t_{d(PCLKH-OHZ)}$	Delay time, PCLK high to output high impedance		28*	ns

*This parameter is not production tested.

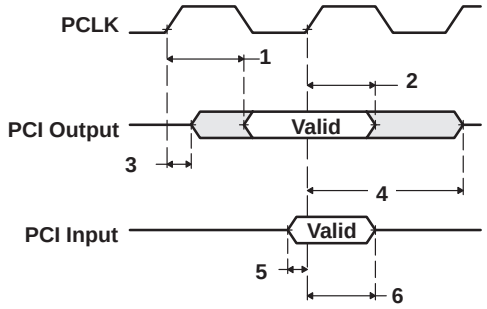


Figure 49. PCI Input/Output Timing

PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING [C6415 AND C6416 ONLY] (CONTINUED)

timing requirements for serial EEPROM interface (see Figure 50)

NO.			MIN	MAX	UNIT
8	$t_{su(DIV-CLKH)}$	Setup time, XSP_DI valid before XSP_CLK high	50*		ns
9	$t_{h(CLKH-DIV)}$	Hold time, XSP_DI valid after XSP_CLK high	0*		ns

*This parameter is not production tested.

switching characteristics over recommended operating conditions for serial EEPROM interface† (see Figure 50)

NO.	PARAMETER		MIN	TYP	MAX	UNIT
1	$t_{w(CSL)}$	Pulse duration, XSP_CS low		4092P		ns
2	$t_{d(CLKL-CSL)}$	Delay time, XSP_CLK low to XSP_CS low		0		ns
3	$t_{d(CSH-CLKH)}$	Delay time, XSP_CS high to XSP_CLK high		2046P		ns
4	$t_{w(CLKH)}$	Pulse duration, XSP_CLK high		2046P		ns
5	$t_{w(CLKL)}$	Pulse duration, XSP_CLK low		2046P		ns
6	$t_{osu(DOV-CLKH)}$	Output setup time, XSP_DO valid after XSP_CLK high		2046P		ns
7	$t_{oh(CLKH-DOV)}$	Output hold time, XSP_DO valid after XSP_CLK high		2046P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.

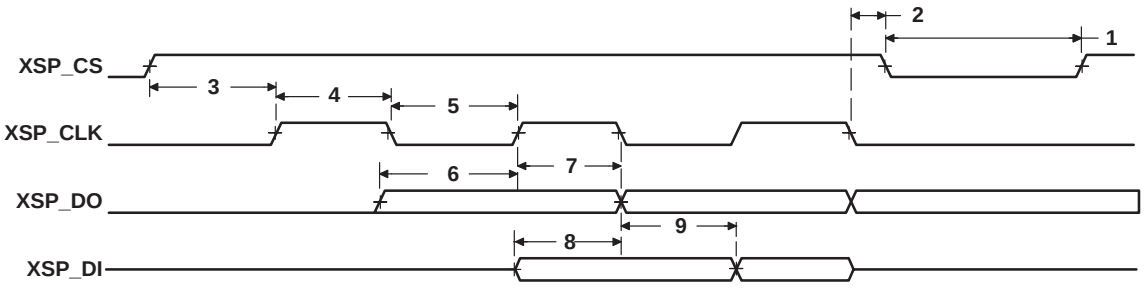


Figure 50. PCI Serial EEPROM Interface Timing

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING

timing requirements for McBSP† (see Figure 51)

NO.				MIN	MAX	UNIT
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	6.67‡*		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	$0.5t_{c(CKRX)}-1$§*		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9		ns
			CLKR ext	1.3		
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	6		ns
			CLKR ext	3		
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	8		ns
			CLKR ext	0.9		
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	3		ns
			CLKR ext	3.1		
10	$t_{su(FXH-CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	9		ns
			CLKX ext	1.3		
11	$t_{h(CKXL-FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	6		ns
			CLKX ext	3		

*This parameter is not production tested.
† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum CLKR/X cycle times are based on internal logic speed; the maximum usable speed may be lower due to EDMA limitations and ac timing requirements.
§ This parameter applies to the maximum McBSP frequency. Operate serial clocks (CLKR/X) in the reasonable range of 40/60 duty cycle.

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

switching characteristics over recommended operating conditions for McBSP†‡ (see Figure 51)

NO.	PARAMETER			MIN	MAX	UNIT
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		1.4	10	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	6.67§*		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C − 1¶*	C + 1¶*	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	−2.1	3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	−1.7	3	ns
			CLKX ext	1.7	9	
12	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	−3.9*	4*	ns
			CLKX ext	−2.1*	9*	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	−3.9 + D1#	4 + D2#	ns
			CLKX ext	−2.1 + D1#	9 + D2#	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode	FSX int	−2.3	5.6	ns
			FSX ext	1.9	9	

*This parameter is not production tested.
† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum delay times also represent minimum output hold times.
§ Minimum CLKR/X cycle times are based on internal logic speed; the maximum usable speed may be lower due to EDMA limitations and AC timing requirements.
¶ C = H or L
S = sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the maximum limit (see ¶ footnote above).
Extra delay from CLKX high to DX valid applies *only* to the first data bit of a device, if and only if DXENA = 1 in SPCR.
if DXENA = 0, then D1 = D2 = 0
if DXENA = 1, then D1 = 4P, D2 = 8P

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

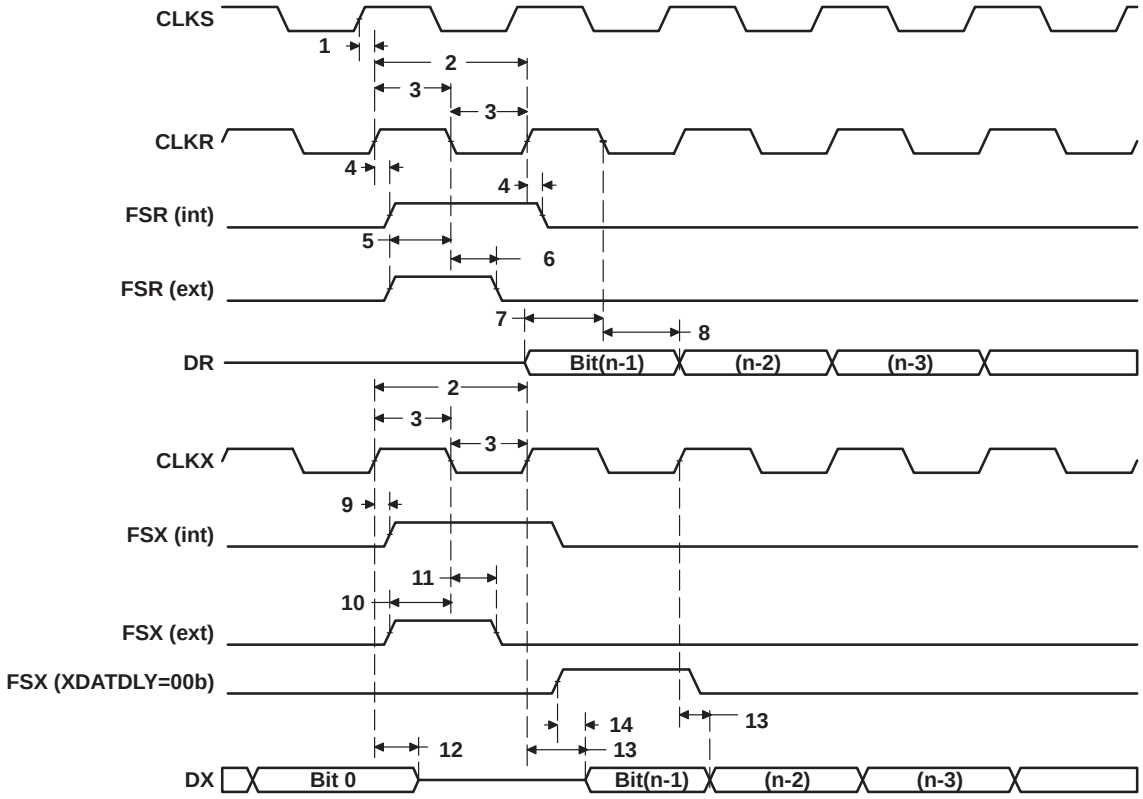


Figure 51. McBSP Timing

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 52)

NO.			MIN	MAX	UNIT
1	$t_{su(FRH-CKSH)}$	Setup time, FSR high before CLKS high	4*		ns
2	$t_{h(CKSH-FRH)}$	Hold time, FSR high after CLKS high	4*		ns

*This parameter is not production tested.

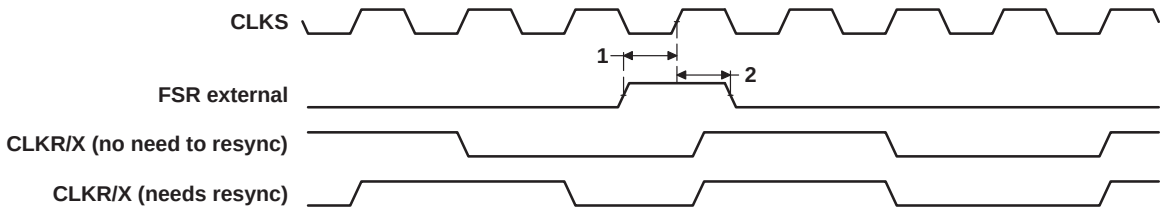


Figure 52. FSR Timing When GSYNC = 1

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 53)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 − 12P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 24P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 53)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T − 2	T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L − 2*	L + 3*			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	−2	4	12P + 2.8	20P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	L − 2*	L + 3*			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			4P + 3*	12P + 17*	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			8P + 1.8	16P + 17	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = Sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency)
= Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
# FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

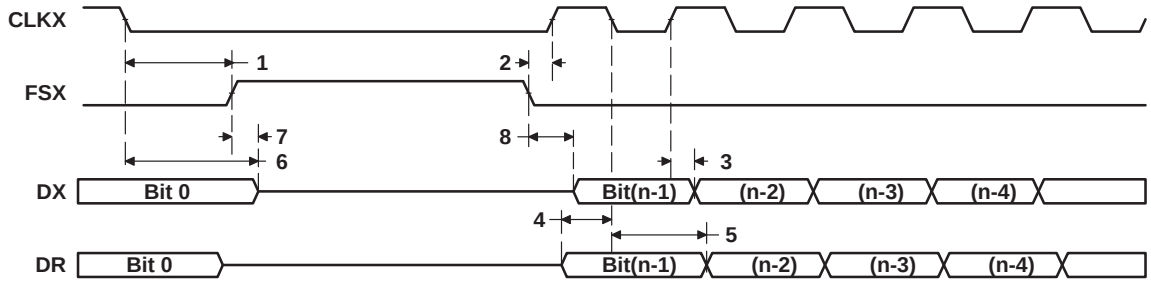


Figure 53. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 54)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12*		2 − 12P*		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4*		5 + 24P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 54)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	L − 2*	L + 3*			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	T − 2*	T + 3*			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	−2*	4*	12P + 4*	20P + 17*	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	−2*	4*	12P + 3*	20P + 17*	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H − 2*	H + 4*	8P + 2*	16P + 17*	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = Sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency)
= Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
# FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

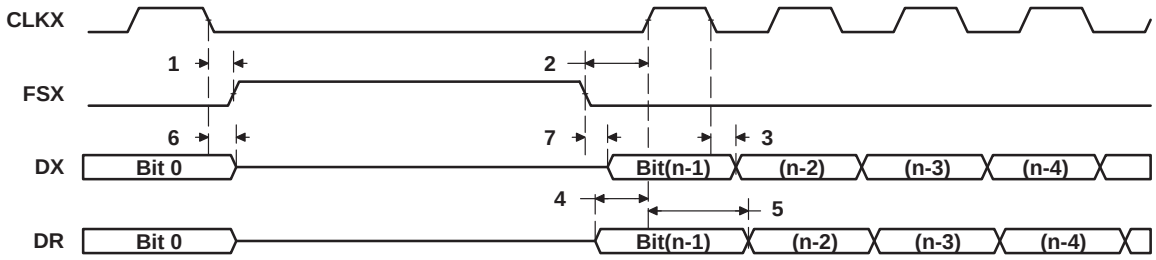


Figure 54. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 55)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12*		2 − 12P*		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4*		5 + 24P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 55)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	T − 2*	T + 3*			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	H − 2*	H + 3*			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	−2*	4*	12P + 4*	20P + 17*	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	H − 2*	H + 3*			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			4P + 3*	12P + 17*	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			8P + 2*	16P + 17*	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = Sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency)
= Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
# FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

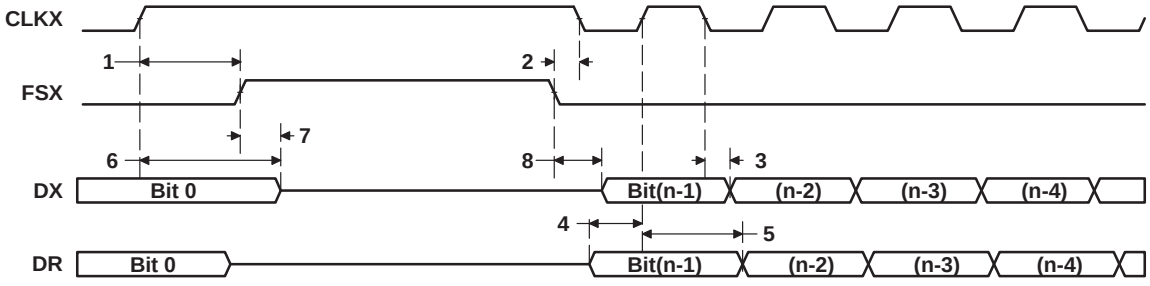


Figure 55. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 56)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12*		2 − 12P*		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4*		5 + 24P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 56)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	H − 2*	H + 3*			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	T − 2*	T + 1*			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	−2*	4*	12P + 4*	20P + 17*	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	−2*	4*	12P + 3*	20P + 17*	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L − 2*	L + 4*	8P + 2*	16P + 17*	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/4 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = Sample rate generator input clock = 4P if CLKSM = 1 (P = 1/CPU clock frequency)
= Sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI Master, FSX is inverted to provide active-low slave-enable output. As a Slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for Master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for Slave McBSP
# FSX should be low before the rising edge of clock to enable Slave devices and then begin a SPI transfer at the rising edge of the Master clock (CLKX).

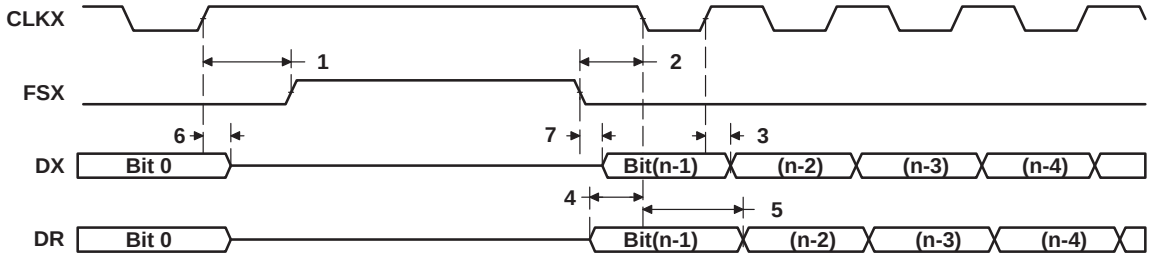


Figure 56. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

UTOPIA SLAVE TIMING [C6415 AND C6416 ONLY]

timing requirements for UXCLK† (see Figure 57)

NO.			MIN	MAX	UNIT
1	$t_{c(UXCK)}$	Cycle time, UXCLK	20*		ns
2	$t_{w(UXCKH)}$	Pulse duration, UXCLK high	$0.4t_{c(UXCK)}$*	$0.6t_{c(UXCK)}$*	ns
3	$t_{w(UXCKL)}$	Pulse duration, UXCLK low	$0.4t_{c(UXCK)}$*	$0.6t_{c(UXCK)}$*	ns
4	$t_{t(UXCK)}$	Transition time, UXCLK		2*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.

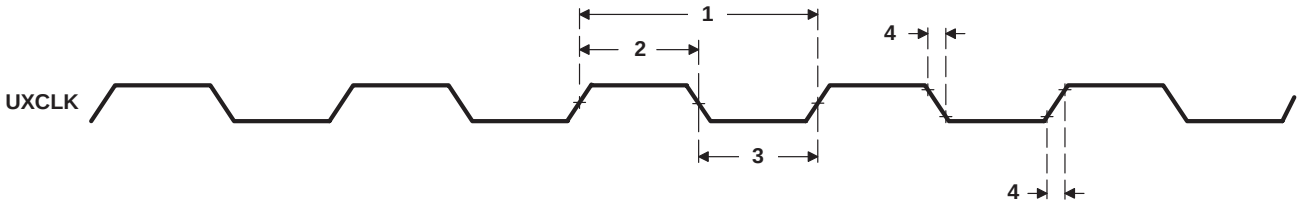


Figure 57. UXCLK Timing

timing requirements for URCLK† (see Figure 58)

NO.			MIN	MAX	UNIT
1	$t_{c(URCK)}$	Cycle time, URCLK	20*		ns
2	$t_{w(URCKH)}$	Pulse duration, URCLK high	$0.4t_{c(URCK)}$*	$0.6t_{c(URCK)}$*	ns
3	$t_{w(URCKL)}$	Pulse duration, URCLK low	$0.4t_{c(URCK)}$*	$0.6t_{c(URCK)}$*	ns
4	$t_{t(URCK)}$	Transition time, URCLK		2*	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.

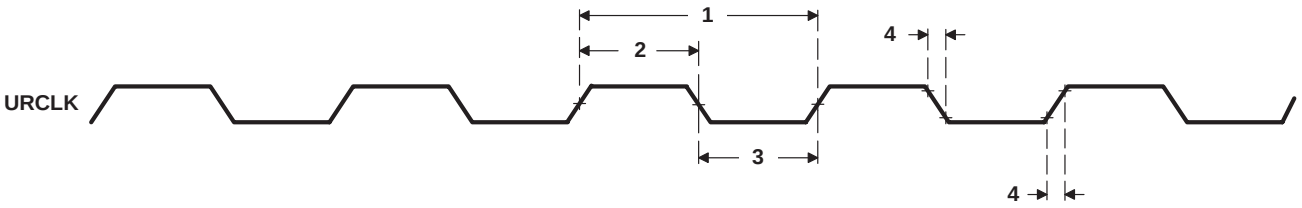


Figure 58. URCLK Timing

UTOPIA SLAVE TIMING [C6415 AND C6416 ONLY] (CONTINUED)

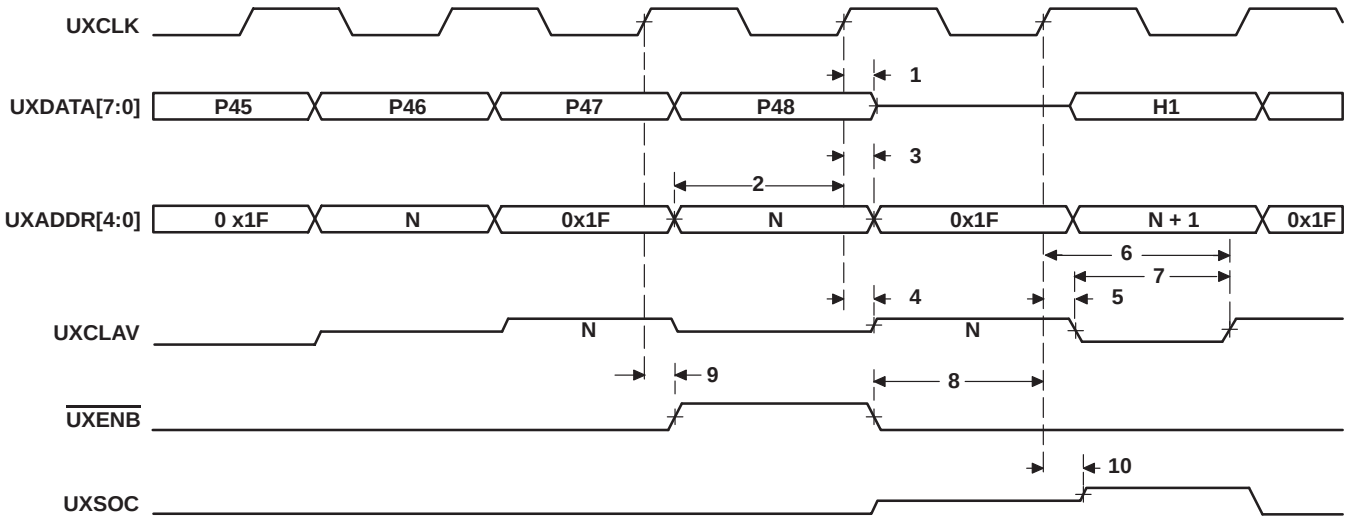
timing requirements for UTOPIA Slave transmit (see Figure 59)

NO.			MIN	MAX	UNIT
2	$t_{su(UXAV-UXCH)}$	Setup time, UXADDR valid before UXCLK high	4		ns
3	$t_{h(UXCH-UXAV)}$	Hold time, UXADDR valid after UXCLK high	1		ns
8	$t_{su(UXENBL-UXCH)}$	Setup time, $\overline{\text{UXENB}}$ low before UXCLK high	4		ns
9	$t_{h(UXCH-UXENBL)}$	Hold time, $\overline{\text{UXENB}}$ low after UXCLK high	1		ns

switching characteristics over recommended operating conditions for UTOPIA Slave transmit (see Figure 59)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(UXCH-UXDV)}$	Delay time, UXCLK high to UXDATA valid	3	12	ns
4	$t_{d(UXCH-UXCLAV)}$	Delay time, UXCLK high to UXCLAV driven active value	3	12	ns
5	$t_{d(UXCH-UXCLAVL)}$	Delay time, UXCLK high to UXCLAV driven inactive low	3*	12*	ns
6	$t_{d(UXCH-UXCLAVHZ)}$	Delay time, UXCLK high to UXCLAV going Hi-Z	9*	18.5*	ns
7	$t_{w(UXCLAVL-UXCLAVHZ)}$	Pulse duration (low), UXCLAV low to UXCLAV Hi-Z	3*		ns
10	$t_{d(UXCH-UXSV)}$	Delay time, UXCLK high to UXSOC valid	3	12	ns

*This parameter is not production tested.



† The UTOPIA Slave module has signals that are middle-level signals indicating a high-impedance state (i.e., the UXCLAV and UXSOC signals).

Figure 59. UTOPIA Slave Transmit Timing†

UTOPIA SLAVE TIMING [C6415 AND C6416 ONLY] (CONTINUED)

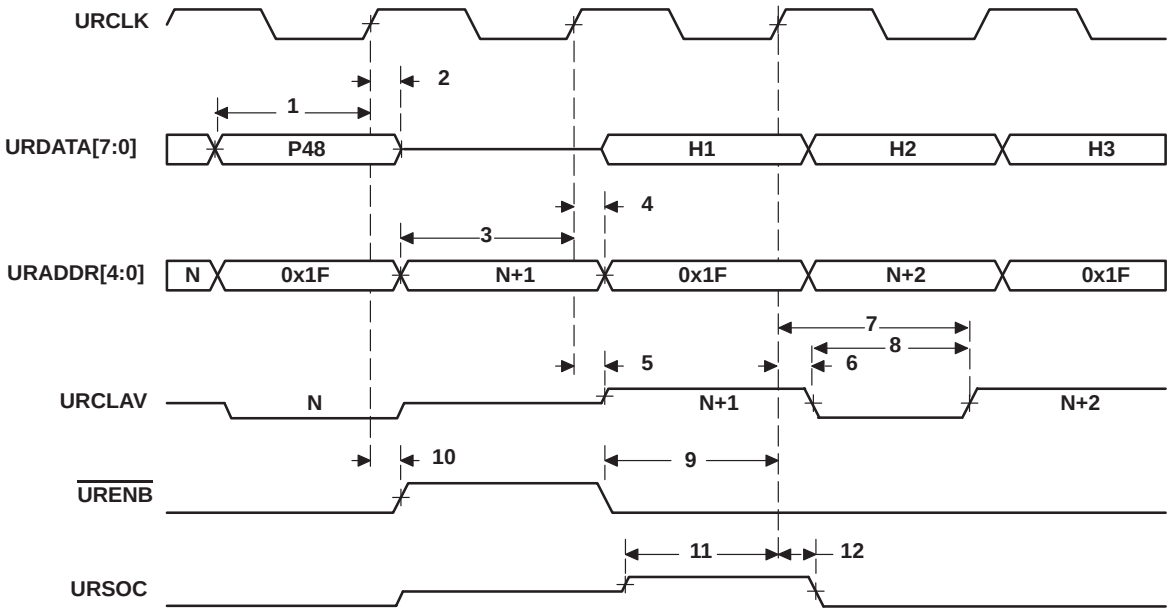
timing requirements for UTOPIA Slave receive (see Figure 60)

NO.			MIN	MAX	UNIT
1	$t_{su(URDV-URCH)}$	Setup time, URDATA valid before URCLK high	4		ns
2	$t_{h(URCH-URDV)}$	Hold time, URDATA valid after URCLK high	1		ns
3	$t_{su(URAV-URCH)}$	Setup time, URADDR valid before URCLK high	4		ns
4	$t_{h(URCH-URAV)}$	Hold time, URADDR valid after URCLK high	1		ns
9	$t_{su(URENBL-URCH)}$	Setup time, $\overline{\text{URENB}}$ low before URCLK high	4		ns
10	$t_{h(URCH-URENBL)}$	Hold time, $\overline{\text{URENB}}$ low after URCLK high	1		ns
11	$t_{su(URSH-URCH)}$	Setup time, URSOC high before URCLK high	4		ns
12	$t_{h(URCH-URSH)}$	Hold time, URSOC high after URCLK high	1		ns

switching characteristics over recommended operating conditions for UTOPIA Slave receive (see Figure 60)

NO.	PARAMETER		MIN	MAX	UNIT
5	$t_{d(URCH-URCLAV)}$	Delay time, URCLK high to URCLAV driven active value	3	12	ns
6	$t_{d(URCH-URCLAVL)}$	Delay time, URCLK high to URCLAV driven inactive low	3*	12*	ns
7	$t_{d(URCH-URCLAVHZ)}$	Delay time, URCLK high to URCLAV going Hi-Z	9*	18.5*	ns
8	$t_{w(URCLAVL-URCLAVHZ)}$	Pulse duration (low), URCLAV low to URCLAV Hi-Z	3*		ns

*This parameter is not production tested.



† The UTOPIA Slave module has signals that are middle-level signals indicating a high-impedance state (i.e., the URCLAV and URSOC signals).

Figure 60. UTOPIA Slave Receive Timing†

TIMER TIMING

timing requirements for timer inputs† (see Figure 61)

NO.			MIN	MAX	UNIT
1	$t_{w(TINPH)}$	Pulse duration, TINP high	8P*		ns
2	$t_{w(TINPL)}$	Pulse duration, TINP low	8P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.

switching characteristics over recommended operating conditions for timer outputs† (see Figure 61)

NO.	PARAMETER		MIN	MAX	UNIT
3	$t_{w(TOUTH)}$	Pulse duration, TOUT high	8P − 3*		ns
4	$t_{w(TOUTL)}$	Pulse duration, TOUT low	8P − 3*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.

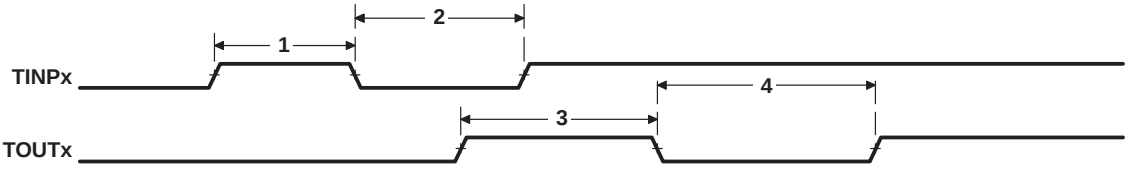


Figure 61. Timer Timing

GENERAL-PURPOSE INPUT/OUTPUT (GPIO) PORT TIMING

timing requirements for GPIO inputs†‡ (see Figure 62)

NO.			MIN	MAX	UNIT
1	$t_{w(GPIH)}$	Pulse duration, GPIx high	8P*		ns
2	$t_{w(GPIL)}$	Pulse duration, GPIx low	8P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ The pulse width given is sufficient to generate a CPU interrupt or an EDMA event. However, if a user wants to have the DSP recognize the GPIx changes through software polling of the GPIO register, the GPIx duration must be extended to at least 12P to allow the DSP enough time to access the GPIO register through the CFGBUS.

switching characteristics over recommended operating conditions for GPIO outputs† (see Figure 62)

NO.	PARAMETER		MIN	MAX	UNIT
3	$t_{w(GPOH)}$	Pulse duration, GPOx high	32P*		ns
4	$t_{w(GPOL)}$	Pulse duration, GPOx low	32P*		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.

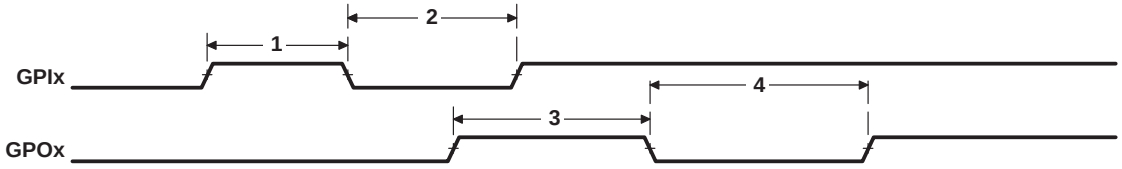


Figure 62. GPIO Port Timing

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 63)

NO.			MIN	MAX	UNIT
1	$t_{c(TCK)}$	Cycle time, TCK	35*		ns
3	$t_{su(TDIV-TCKH)}$	Setup time, TDI/TMS/$\overline{TRST}$ valid before TCK high	10*		ns
4	$t_{h(TCKH-TDIV)}$	Hold time, TDI/TMS/$\overline{TRST}$ valid after TCK high	9*		ns

*This parameter is not production tested.

switching characteristics over recommended operating conditions for JTAG test port (see Figure 63)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid	–3*	18*	ns

*This parameter is not production tested.

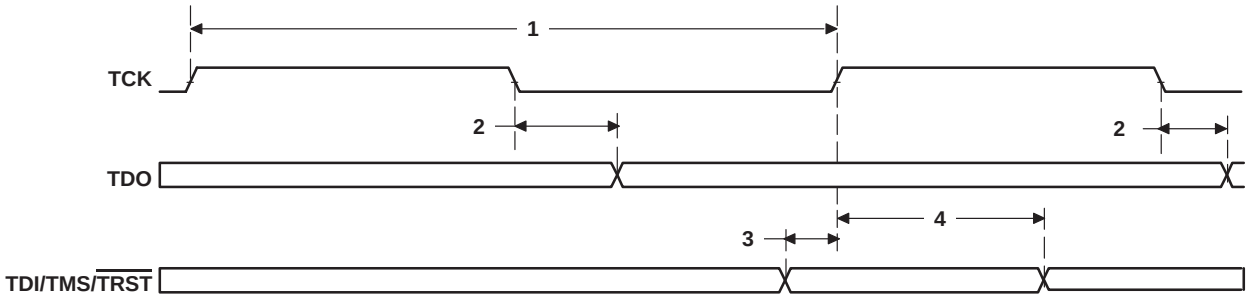
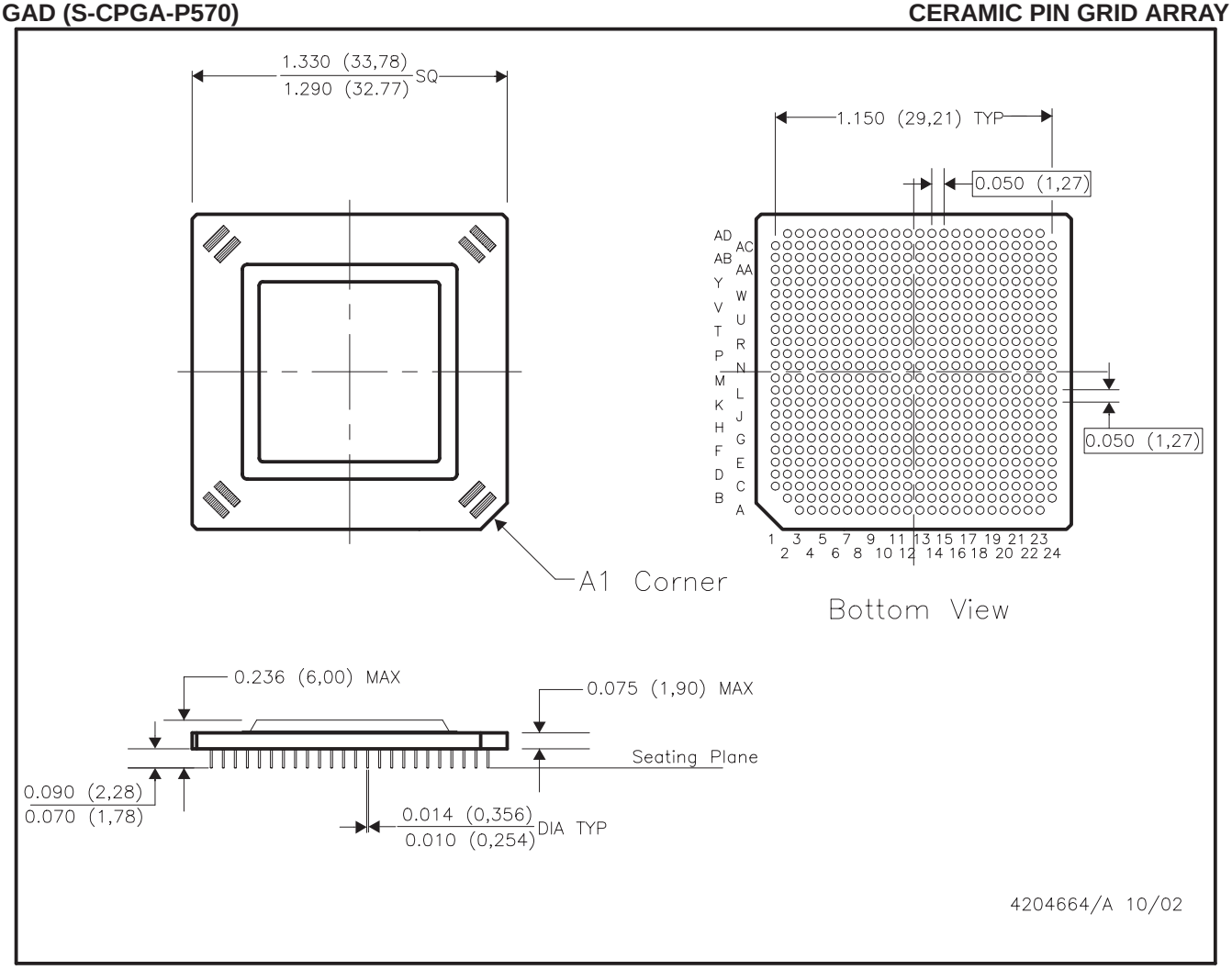


Figure 63. JTAG Test-Port Timing

MECHANICAL DATA

GAD (S-CPGA-P570) **CERAMIC PIN GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Flip chip application only
D. This package is hermetically sealed with a metal lid.

thermal resistance characteristics (S-CPGA package)

NO			°C/W
1	$R\Theta_{JA}$	Junction-to-free air (Low K JEDEC PCB)	20.96
2	$R\Theta_{JA}$	Junction-to-free air (High K JEDEC PCB)	16.5
3	$R\Theta_{JC}$	Junction-to-case (High K JEDEC PCB with heat sink on lid)	10.9
4	$R\Theta_{JC}$	Junction-to-case (High K JEDEC PCB with heat sink on body)	7.69
5	$R\Theta_{JB}$	Junction-to-board (High K JEDEC PCB)	9
6	$R\Theta_{JB}$	Junction-to-board (High K JEDEC PCB with thermal compound on top of the die)	2.93

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
5962-0324501QXA	LIFEBUY	FCPGA	GAD	570	8	TBD	Call TI	Call TI	-55 to 115	(5962-0324501QX, 5962-0324501QXA) (A, SMJ320C6415DGADW60) SMJ320C6415DGADW60	
SM320C6415DGADW60	LIFEBUY	FCPGA	GAD	570	8	TBD	Call TI	Call TI	-55 to 115	SM320C6415DGADW60 SM320C6415DGADW60	
SMJ320C6415DGADW60	LIFEBUY	FCPGA	GAD	570	8	TBD	Call TI	Call TI	-55 to 115	(5962-0324501QX, 5962-0324501QXA) (A, SMJ320C6415DGADW60) SMJ320C6415DGADW60	

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

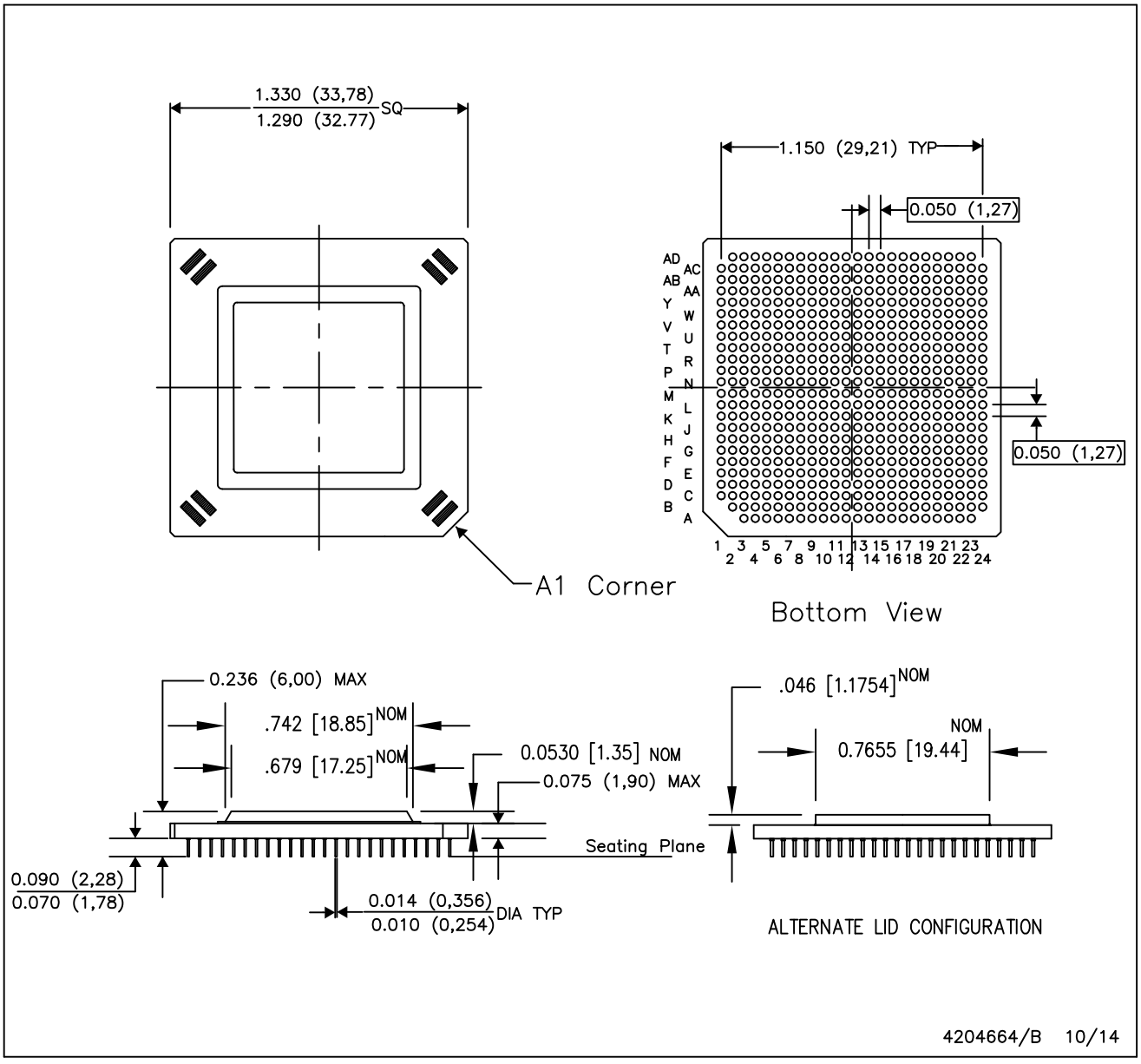
OTHER QUALIFIED VERSIONS OF SM320C6415, SMJ320C6415 :

- Catalog : TMS320C6415, TMS320C6415
- Enhanced Product : SM320C6415-EP
- Military : SMJ320C6415

NOTE: Qualified Version Definitions:

- Catalog - TI's standard catalog product
- Enhanced Product - Supports Defense, Aerospace and Medical Applications
- Military - QML certified for Military and Defense Applications

GAD (S–CPGA–P570) CERAMIC PIN GRID ARRAY



NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Flip chip application only
D. This package is hermetically sealed with a metal lid.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated