- **Low-Cost, High-Performance Fixed-Point DSP – TMS320C6411**
  - **3.3-ns Instruction Cycle Time**
  - **300-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **Twenty-Eight Operations/Cycle**
  - **2400 MIPS**
  - **Fully Software-Compatible With TMS320C62x™**
- **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**
- **L1/L2 Memory Architecture**
  - **128K-Bit (16K-Byte) L1P Program Cache (Direct Mapped)**
  - **128K-Bit (16K-Byte) L1D Data Cache (2-Way Set-Associative)**
  - **2M-Bit (256K-Byte) L2 Unified Mapped RAM/Cache (Flexible RAM/Cache Allocation)**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Asynchronous Memories (SRAM and EPROM) and Synchronous Memories (SDRAM, SBSRAM, ZBT SRAM, and FIFO)**
  - **512M-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)**
  - **Access to Entire Memory Map**
- **32-Bit/33-MHz, 3.3-V Peripheral Component Interconnect (PCI) Master/Slave Interface Conforms to PCI Specification 2.2**
  - **Access to Entire Memory Map**
  - **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**
- **Two Multichannel Buffered Serial Ports (McBSPs)**
  - **Direct Interface to T1/E1, MVIP, SCSA Framers**
  - **ST-Bus-Switching Compatible**
  - **Up to 256 Channels Each**
  - **AC97-Compatible**
  - **Serial Peripheral Interface (SPI) Compatible (Motorola™)**
- **Three 32-Bit General-Purpose Timers**
- **Sixteen General-Purpose I/O (GPIO) Pins**
  - **Programmable Interrupt/Event Generation Modes**
- **Flexible PLL Clock Generator**
- **IEEE-1149.1 (JTAG†) Boundary-Scan-Compatible**
- **532-Pin Ball Grid Array (BGA) Package (GLZ Suffix), 0.8-mm Ball Pitch**
- **0.12-μm/6-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1-V Internal**

PRODUCT PREVIEW

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.

TMS320C62x, 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.

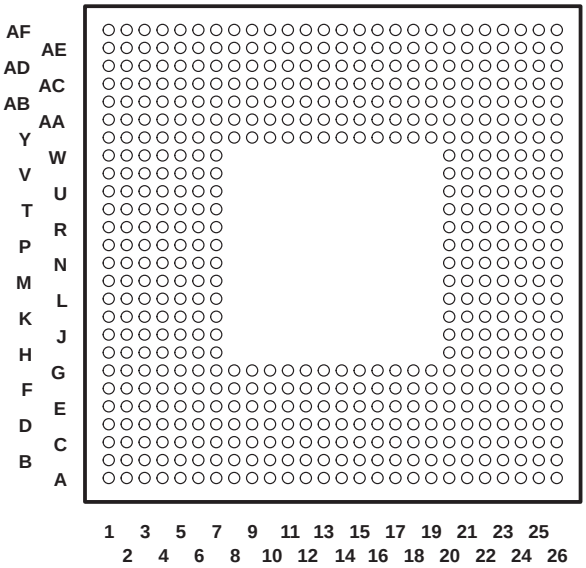
PRODUCT PREVIEW

**Table of Contents**



## GLZ BGA package (bottom view)

**GLZ 532-PIN BALL GRID ARRAY (BGA) PACKAGE**
**(BOTTOM VIEW)**



## description

The TMS320C64x™ DSPs (including the TMS320C6411 device) are the highest-performance fixed-point DSP generation in the TMS320C6000™ DSP platform. The TMS320C6411 (C6411) device is based on the second-generation high-performance, advanced VelociTI™ very-long-instruction-word (VLIW) architecture (VelocTI.2™) developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunction applications. The C64x™ is a code-compatible member of the C6000™ DSP platform.

With performance of up to 2400 million instructions per second (MIPS) at a clock rate of 300 MHz, the C6411 device offers cost-effective solutions to high-performance DSP programming challenges. The C6411 DSP possesses 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 C6411 can produce two 32-bit multiply-accumulates (MACs) per cycle for a total of 600 million MACs per second (MMACS), or eight 8-bit MACs per cycle for a total of 2400 MMACS. The C6411 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals similar to the other C6000™ DSP platform devices.

The C6411 uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 128-Kbit direct mapped cache and the Level 1 data cache (L1D) is a 128-Kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 2-Mbit memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or combinations of the two. The peripheral set includes two multichannel buffered serial ports (McBSPs); three 32-bit general-purpose timers; a user-configurable 16-bit or 32-bit host-port interface (HPI16/HPI32); a peripheral component interconnect (PCI); a general-purpose input/output port (GPIO) with 16 GPIO pins; and a glueless external memory interface (32-bit EMIF), which is capable of interfacing to synchronous and asynchronous memories and peripherals.

The C6411 has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

PRODUCT PREVIEW

## device characteristics

Table 1 provides an overview of the C6411 DSP. The table shows significant features of the C6411 device, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

**Table 1. Characteristics of the C6411 Processor**

| HARDWARE FEATURES | | C6411 |
|---|---|---|
| Peripherals | EMIF (32-bit bus width) | 1 |
| | EDMA (64 independent channels) | 1 |
| | HPI (32- or 16-bit user selectable) | 1 (HPI16 or HPI32) |
| | PCI (32-bit) | 1 |
| | McBSPs (McBSP0 and McBSP1) | 2 |
| | 32-Bit Timers | 3 |
| | General-Purpose Input/Outputs (GPIOs) | 16 |
| On-Chip Memory | Size (Bytes) | 288K |
| | Organization | 16K-Byte (16KB) L1 Program (L1P) Cache<br>16KB L1 Data (L1D) Cache<br>256KB Unified Mapped RAM/Cache (L2) |
| CPU ID + CPU Rev ID | Control Status Register (CSR.[31:16]) | 0x0C01 |
| Frequency | MHz | 300 |
| Cycle Time | ns | 3.3 ns (C6411-300) |
| Voltage | Core (V) | 1 V (-300) |
| | I/O (V) | 3.3 V |
| PLL Options | CLKIN frequency multiplier | Bypass (x1), x6 |
| BGA Package | 23 x 23 mm | 532-Pin BGA (GLZ) |
| Process Technology | μm | 0.12 μm |
| Product Status† | Product Preview (PP)<br>Advance Information (AI)<br>Production Data (PD) | PP |
| Device Part Numbers | (For more details on the C6000™ DSP part numbering, see Figure 4) | TMX320C6411GLZ |

† PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.
ADVANCE INFORMATION concerns new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.
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.

PRODUCT PREVIEW

## device compatibility

The C64x™ family of devices has a diverse and powerful set of peripherals. The common peripheral set that the C6411, 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 C6411 device is a low-cost C64x device which features significant enhancements from the C6211/C6211B devices and can be considered a subset of the C6415 device. Table 2 identifies the C6411 features in comparison with the C6211 and C6415 devices.

**Table 2. C6211, C6411, and C6415 Device Comparison†**

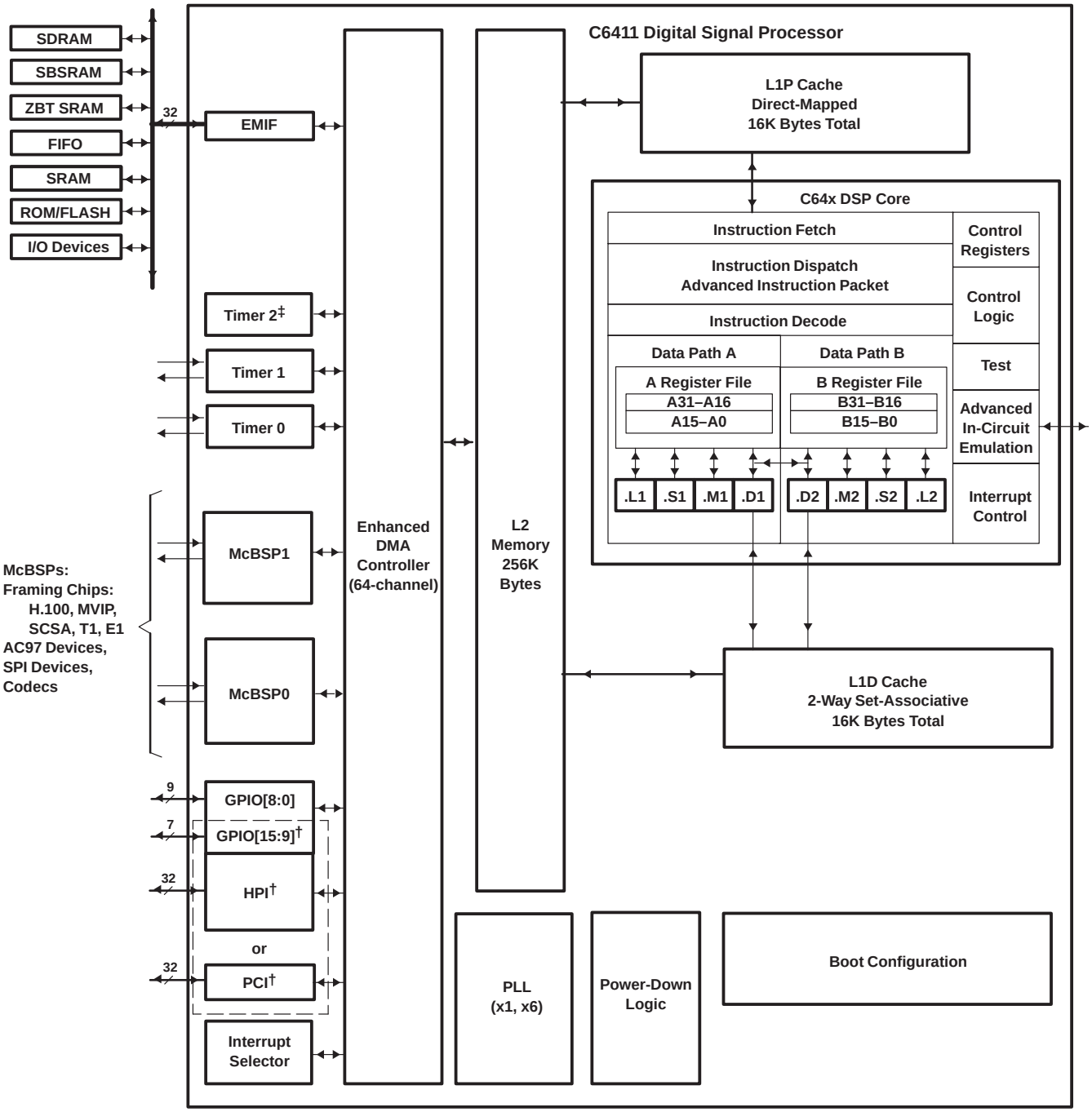
| CPU/PERIPHERALS | C6211/C6211B | C6411 | C6415 |
|---|---|---|---|
| DSP Core | C62x | C64x | C64x |
| L1P (Program Cache) | 4 KB | 16 KB | 16 KB |
| L1D (Data Cache) | 4 KB | 16 KB | 16 KB |
| L2 (Unified Mapped RAM/Cache) | 64 KB | 256 KB | 1024 KB |
| EMIF (64-, 32-, 16-bit bus width) | (1) 32-Bit | (1) 32-Bit programmable synchronous mode | (1) 64-Bit(1) [EMIFA] (1) 16-Bit [EMIFB] programmable synchronous mode |
| EDMA (# of independent channels) | 16 | 64 | 64 |
| HPI (32- or 16-bit user selectable) | 16-Bit | 32-/16-Bit | 32-/16-Bit |
| PCI (32-bit) | — | 32-Bit, 33 MHz | 32-Bit, 33 MHz |
| McBSPs (McBSP0, McBSP1, and McBSP2) | 2 | 2 Enhanced | 3 Enhanced |
| UTOPIA | — | — | (1) Transmit (1) Receive |
| Timers (32-bit) [TIMER0, TIMER1, TIMER2] | 2 | 3 (No TIMER2 pins) | 3 |
| GPIOs (GP[15:0]) | — | 16 | 16 |
| Core Frequency (MHz) | 150-, 167-MHz | 300-MHz | 400-, 500-, 600-MHz |
| Core Voltage (V) | 1.8 V | 1 V | 1.2 V to 1.4 V |
| PLL Modes (x1 [Bypass], x4, x6, x12) | x1, x4 | x1, x6 | x1, x6, x12 |
| Package | 256-pin BGA 27 x 27 mm GFN suffix | 532-pin BGA 23 x 23 mm GLZ suffix | 532-pin BGA 23 x 23 mm GLZ suffix |
| Process Technology | 0.18 µm | 0.12 µm | 0.12 µm |

† — denotes peripheral/coprocessor is *not* available on this device.

For more detailed information on the device compatibility and similarities/differences among the C6211, C6411, C6414, C6415, and C6416 devices, see the *How To Begin Development Today With the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs* application report (literature number SPRA718) and *How To Begin Development Today With the TMS320C6411 DSP* application report (literature number SPRA374).

PRODUCT PREVIEW

## functional block and CPU (DSP core) diagram




† 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.

‡ Timer2 exists internally and is *not* pinned out externally.

## 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, 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)

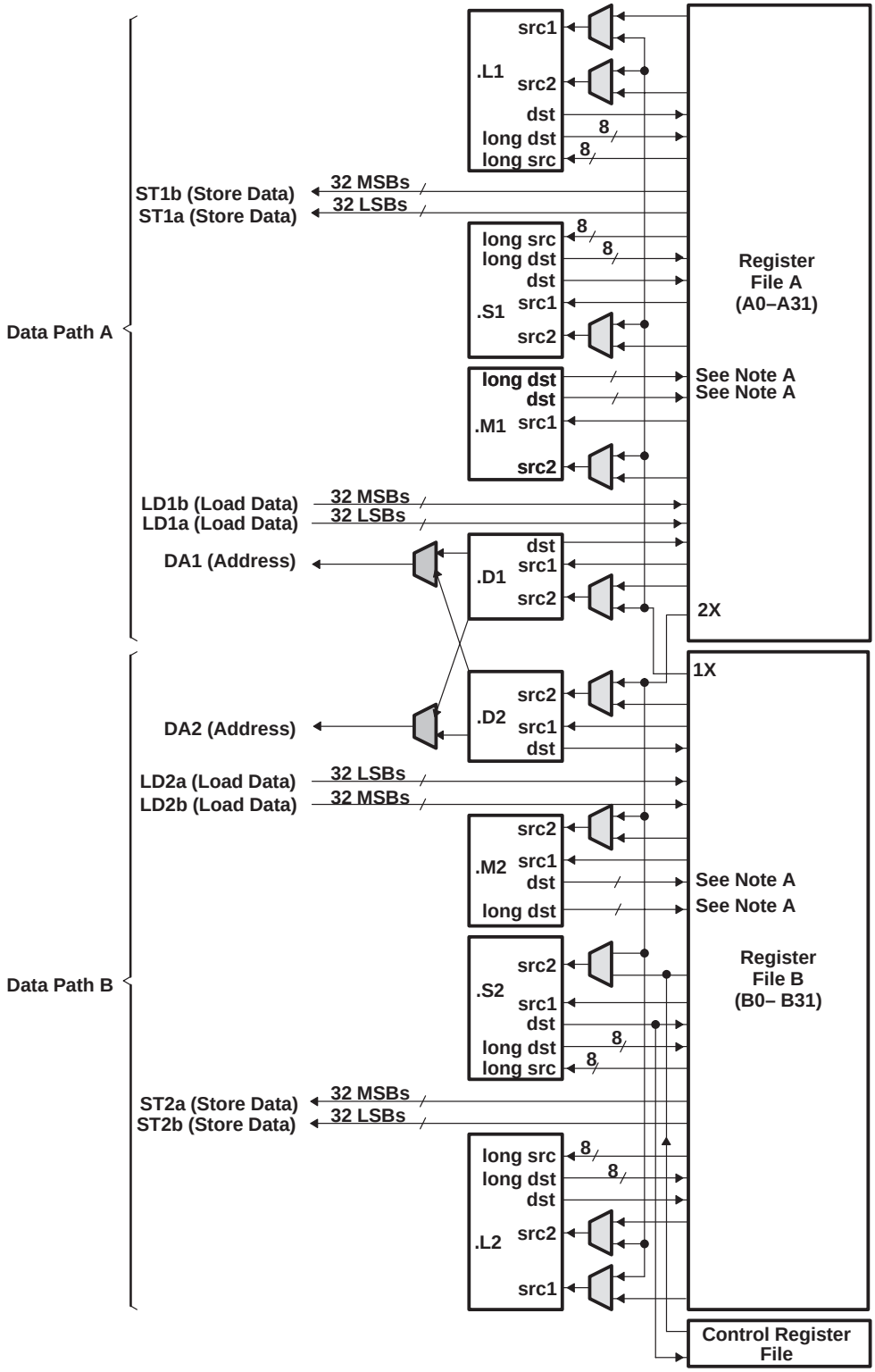
*TMS320C64x Technical Overview* (literature number SPRU395)

*How To Begin Development Today With the TMS320C6411 DSP* application report (literature number SPRA374)



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. TMS320C64x™ CPU (DSP Core) Data Paths**

PRODUCT PREVIEW

## memory map summary

Table 3 shows the memory map address ranges of the C6411 device. Internal memory is always located at address 0 and can be used as both program and data memory. The external memory address range in the C6411 device begins at the hex address location 0x8000 0000 for the EMIF.

**Table 3. TMS320C6411 Memory Map Summary**

| MEMORY BLOCK DESCRIPTION | BLOCK SIZE (BYTES) | HEX ADDRESS RANGE |
|---|---|---|
| Internal RAM (L2) | 256K | 0000 0000 – 0003 FFFF |
| Reserved | 24M – 256K | 0004 0000 – 017F FFFF |
| External Memory Interface (EMIF) 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 |
| Reserved | 512K | 01A4 0000 – 01AB FFFF |
| Timer 2 Registers | 256K | 01AC 0000 – 01AF FFFF |
| GPIO Registers | 256K | 01B0 0000 – 01B3 FFFF |
| Reserved | 768K | 01B4 0000 – 01BF FFFF |
| PCI Registers | 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 |
| Reserved | 1G + 128M | 3800 0000 – 7FFF FFFF |
| EMIF CE0† | 256M | 8000 0000 – 8FFF FFFF |
| EMIF CE1† | 256M | 9000 0000 – 9FFF FFFF |
| EMIF CE2† | 256M | A000 0000 – AFFF FFFF |
| EMIF CE3† | 256M | B000 0000 – BFFF FFFF |
| Reserved | 1G | C000 0000 – FFFF FFFF |

† The number of EMIF address pins (EA[21:2]) limits the maximum addressable memory (SDRAM) to 128MB per CE space. To get 256MB of addressable memory, an additional general-purpose output pin or external logic is required.

## peripheral register descriptions

Table 4 through Table 18 identify the peripheral registers for the C6411 device 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 *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

**Table 4. EMIF Registers**

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

## peripheral register descriptions (continued)

**Table 5. L2 Cache Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0184 0000 | CCFG | Cache configuration register | |
| | – | 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 | |
| | – | 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 | |
| | – | Reserved | |
| 0184 5000 | L2FLUSH | L2 flush register | |
| 0184 5004 | L2CLEAN | L2 clean register | |
| | – | Reserved | |
| 0184 8000 –0184 81FC | MAR0 to MAR127 | Reserved | |
| 0184 8200 | MAR128 | Controls EMIF CE0 range 8000 0000 – 80FF FFFF | |
| 0184 8204 | MAR129 | Controls EMIF CE0 range 8100 0000 – 81FF FFFF | |
| 0184 8208 | MAR130 | Controls EMIF CE0 range 8200 0000 – 82FF FFFF | |
| 0184 820C | MAR131 | Controls EMIF CE0 range 8300 0000 – 83FF FFFF | |
| 0184 8210 | MAR132 | Controls EMIF CE0 range 8400 0000 – 84FF FFFF | |
| 0184 8214 | MAR133 | Controls EMIF CE0 range 8500 0000 – 85FF FFFF | |
| 0184 8218 | MAR134 | Controls EMIF CE0 range 8600 0000 – 86FF FFFF | |
| 0184 821C | MAR135 | Controls EMIF CE0 range 8700 0000 – 87FF FFFF | |
| 0184 8220 | MAR136 | Controls EMIF CE0 range 8800 0000 – 88FF FFFF | |
| 0184 8224 | MAR137 | Controls EMIF CE0 range 8900 0000 – 89FF FFFF | |
| 0184 8228 | MAR138 | Controls EMIF CE0 range 8A00 0000 – 8AFF FFFF | |
| 0184 822C | MAR139 | Controls EMIF CE0 range 8B00 0000 – 8BFF FFFF | |
| 0184 8230 | MAR140 | Controls EMIF CE0 range 8C00 0000 – 8CFF FFFF | |
| 0184 8234 | MAR141 | Controls EMIF CE0 range 8D00 0000 – 8DFF FFFF | |
| 0184 8238 | MAR142 | Controls EMIF CE0 range 8E00 0000 – 8EFF FFFF | |
| 0184 823C | MAR143 | Controls EMIF CE0 range 8F00 0000 – 8FFF FFFF | |
| 0184 8240 | MAR144 | Controls EMIF CE1 range 9000 0000 – 90FF FFFF | |
| 0184 8244 | MAR145 | Controls EMIF CE1 range 9100 0000 – 91FF FFFF | |
| 0184 8248 | MAR146 | Controls EMIF CE1 range 9200 0000 – 92FF FFFF | |

## peripheral register descriptions (continued)

**Table 5. L2 Cache Registers (Continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0184 824C | MAR147 | Controls EMIF CE1 range 9300 0000 – 93FF FFFF | |
| 0184 8250 | MAR148 | Controls EMIF CE1 range 9400 0000 – 94FF FFFF | |
| 0184 8254 | MAR149 | Controls EMIF CE1 range 9500 0000 – 95FF FFFF | |
| 0184 8258 | MAR150 | Controls EMIF CE1 range 9600 0000 – 96FF FFFF | |
| 0184 825C | MAR151 | Controls EMIF CE1 range 9700 0000 – 97FF FFFF | |
| 0184 8260 | MAR152 | Controls EMIF CE1 range 9800 0000 – 98FF FFFF | |
| 0184 8264 | MAR153 | Controls EMIF CE1 range 9900 0000 – 99FF FFFF | |
| 0184 8268 | MAR154 | Controls EMIF CE1 range 9A00 0000 – 9AFF FFFF | |
| 0184 826C | MAR155 | Controls EMIF CE1 range 9B00 0000 – 9BFF FFFF | |
| 0184 8270 | MAR156 | Controls EMIF CE1 range 9C00 0000 – 9CFF FFFF | |
| 0184 8274 | MAR157 | Controls EMIF CE1 range 9D00 0000 – 9DFF FFFF | |
| 0184 8278 | MAR158 | Controls EMIF CE1 range 9E00 0000 – 9EFF FFFF | |
| 0184 827C | MAR159 | Controls EMIF CE1 range 9F00 0000 – 9FFF FFFF | |
| 0184 8280 | MAR160 | Controls EMIF CE2 range A000 0000 – A0FF FFFF | |
| 0184 8284 | MAR161 | Controls EMIF CE2 range A100 0000 – A1FF FFFF | |
| 0184 8288 | MAR162 | Controls EMIF CE2 range A200 0000 – A2FF FFFF | |
| 0184 828C | MAR163 | Controls EMIF CE2 range A300 0000 – A3FF FFFF | |
| 0184 8290 | MAR164 | Controls EMIF CE2 range A400 0000 – A4FF FFFF | |
| 0184 8294 | MAR165 | Controls EMIF CE2 range A500 0000 – A5FF FFFF | |
| 0184 8298 | MAR166 | Controls EMIF CE2 range A600 0000 – A6FF FFFF | |
| 0184 829C | MAR167 | Controls EMIF CE2 range A700 0000 – A7FF FFFF | |
| 0184 82A0 | MAR168 | Controls EMIF CE2 range A800 0000 – A8FF FFFF | |
| 0184 82A4 | MAR169 | Controls EMIF CE2 range A900 0000 – A9FF FFFF | |
| 0184 82A8 | MAR170 | Controls EMIF CE2 range AA00 0000 – AAFF FFFF | |
| 0184 82AC | MAR171 | Controls EMIF CE2 range AB00 0000 – ABFF FFFF | |
| 0184 82B0 | MAR172 | Controls EMIF CE2 range AC00 0000 – ACFF FFFF | |
| 0184 82B4 | MAR173 | Controls EMIF CE2 range AD00 0000 – ADFF FFFF | |
| 0184 82B8 | MAR174 | Controls EMIF CE2 range AE00 0000 – AEFF FFFF | |
| 0184 82BC | MAR175 | Controls EMIF CE2 range AF00 0000 – AFFF FFFF | |
| 0184 82C0 | MAR176 | Controls EMIF CE3 range B000 0000 – B0FF FFFF | |
| 0184 82C4 | MAR177 | Controls EMIF CE3 range B100 0000 – B1FF FFFF | |
| 0184 82C8 | MAR178 | Controls EMIF CE3 range B200 0000 – B2FF FFFF | |
| 0184 82CC | MAR179 | Controls EMIF CE3 range B300 0000 – B3FF FFFF | |
| 0184 82D0 | MAR180 | Controls EMIF CE3 range B400 0000 – B4FF FFFF | |
| 0184 82D4 | MAR181 | Controls EMIF CE3 range B500 0000 – B5FF FFFF | |
| 0184 82D8 | MAR182 | Controls EMIF CE3 range B600 0000 – B6FF FFFF | |
| 0184 82DC | MAR183 | Controls EMIF CE3 range B700 0000 – B7FF FFFF | |
| 0184 82E0 | MAR184 | Controls EMIF CE3 range B800 0000 – B8FF FFFF | |
| 0184 82E4 | MAR185 | Controls EMIF CE3 range B900 0000 – B9FF FFFF | |
| 0184 82E8 | MAR186 | Controls EMIF CE3 range BA00 0000 – BAFF FFFF | |
| 0184 82EC | MAR187 | Controls EMIF CE3 range BB00 0000 – BBFF FFFF | |
| 0184 82F0 | MAR188 | Controls EMIF CE3 range BC00 0000 – BCFF FFFF | |

## peripheral register descriptions (continued)

**Table 5. L2 Cache Registers (Continued)**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 0184 82F4 | MAR189 | Controls EMIF CE3 range BD00 0000 – BDFF FFFF | |
| 0184 82F8 | MAR190 | Controls EMIF CE3 range BE00 0000 – BEFF FFFF | |
| 0184 82FC | MAR191 | Controls EMIF CE3 range BF00 0000 – BFFF FFFF | |
| 0184 8300 –0184 83FC | MAR192 to MAR255 | Reserved | |
| 0184 8400 –0187 FFFF | – | Reserved | |

## peripheral register descriptions (continued)

**Table 6. 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 7. 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 8. 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 psuedo source address register |
| 0200 0028 | QSCNT | QDMA psuedo frame count register |
| 0200 002C | QSDST | QDMA destination address register |
| 0200 0030 | QSIDX | QDMA psuedo index register |

## peripheral register descriptions (continued)

### Table 9. 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 | |
| 019C 0200 | PDCTL | Peripheral power-down control register (see Table 10) | |
| 019C 0204 – 019F FFFF | – | Reserved | |

### Table 10. Peripheral Power-Down Control Register

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 019C 0200 | PDCTL | Peripheral power-down control register |

### 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 Data Bus | |
| 018C 0004 | DXR0 | McBSP0 data transmit register via Configuration Bus | |
| 0x3000 0000 – 0x33FF FFFF | DXR0 | McBSP0 data transmit register via Peripheral Data 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 | 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 Data Bus | |
| 0190 0004 | DXR1 | McBSP1 data transmit register via Configuration Bus | |
| 0x3400 0000 – 0x37FF FFFF | DXR1 | McBSP1 data transmit register via Peripheral Data 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 | |

## peripheral register descriptions (continued)

### Table 13. 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 14. 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 15. 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. |
| 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 16. 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 0001 – 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 17. 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 – 01B3 FFFF | – | Reserved |

## peripheral register descriptions (continued)

**Table 18. PCI Peripheral Registers**

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME |
|---|---|---|
| 01C0 0000 | RSTSRC | DSP Reset source/status register |
| 01C0 0004 | PMDCSR | Power management DSP control/status register |
| 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 – 01C3 FFFF | – | Reserved |

## EDMA channel synchronization events

The C64x EDMA supports up to 64 EDMA channels which service peripheral devices and external memory. Table 19 lists the source of C64x EDMA synchronization events associated with each of the programmable EDMA channels. For the C6411 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 EDMA Controller chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

**Table 19. TMS320C6411 EDMA Channel Synchronization Events†**

| EDMA CHANNEL | EVENT NAME | EVENT DESCRIPTION |
|---|---|---|
| 0 | DSP_INT | HPI/PCI-to-DSP interrupt |
| 1 | TINT0 | Timer 0 interrupt |
| 2 | TINT1 | Timer 1 interrupt |
| 3 | SD_INT | EMIF 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 |
| 16–18 | – | None |
| 19 | TINT2 | Timer 2 interrupt |
| 20–47 | – | 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 EDMA Controller chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

## interrupt sources and interrupt selector

The C64x DSP core supports 16 prioritized interrupts, which are listed in Table 20. 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 20. 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).

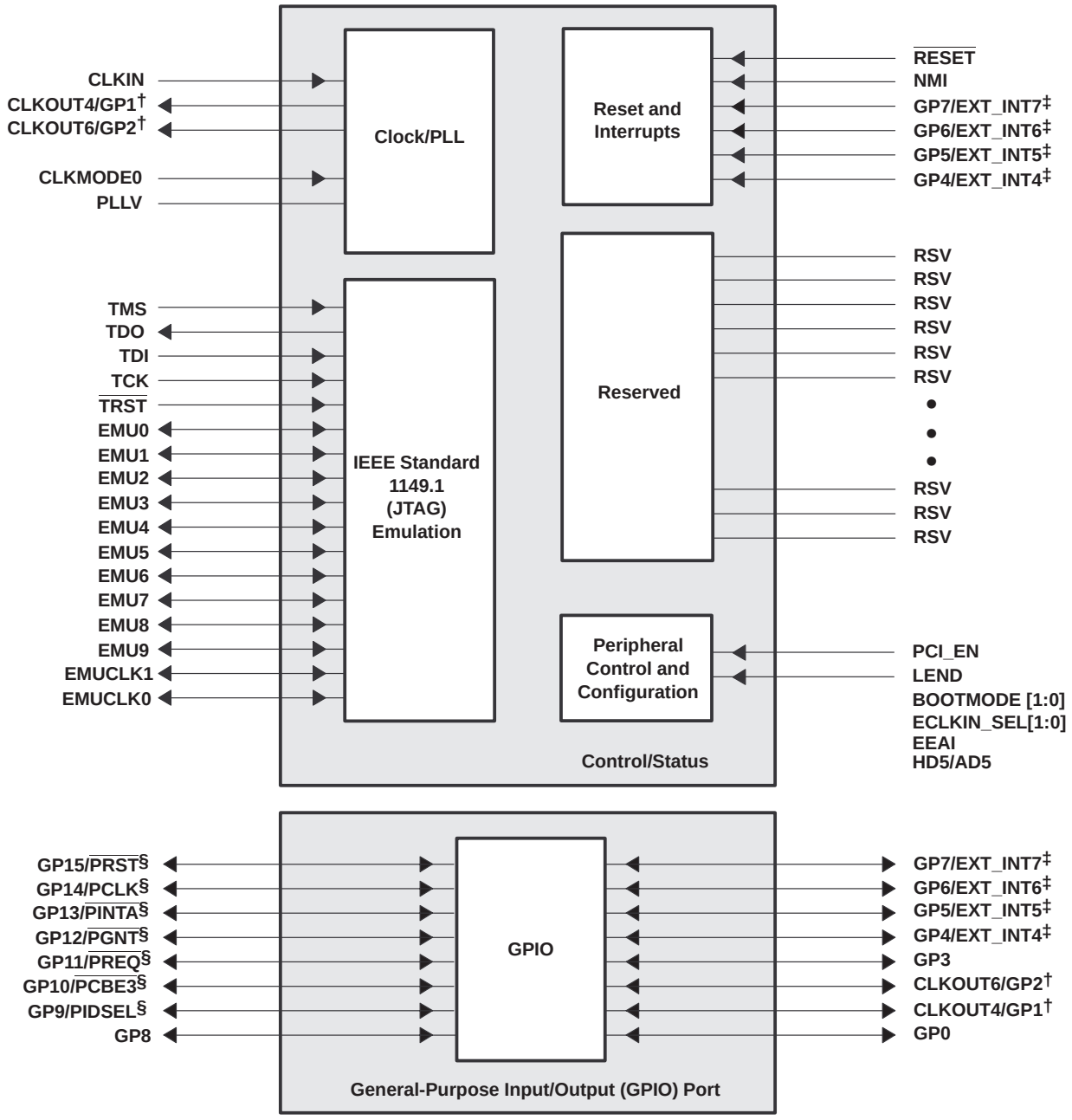
**Table 20. C6411 DSP Interrupts**

| 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_INT | EMIF 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 |
| 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 | Reserved | Reserved. Do not use. |
| – | – | 10010 | Reserved | Reserved. Do not use. |
| – | – | 10011 | TINT2 | Timer 2 interrupt |
| – | – | 10100 – 11111 | Reserved | Reserved. Do not use. |

† 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 20 shows the default interrupt sources for Interrupts INT_04 through INT_15. For more detailed information on interrupt sources and selection, see the Interrupt Selector and External Interrupts chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

## signal groups description




† These pins are muxed with the GPIO port pins and by default these signals function as clocks (CLKOUT4 or CLKOUT6). 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.

§ These GPIO pins are muxed with the PCI peripheral pins and 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.

**Figure 2. CPU and Peripheral Signals**

## signal groups description (continued)

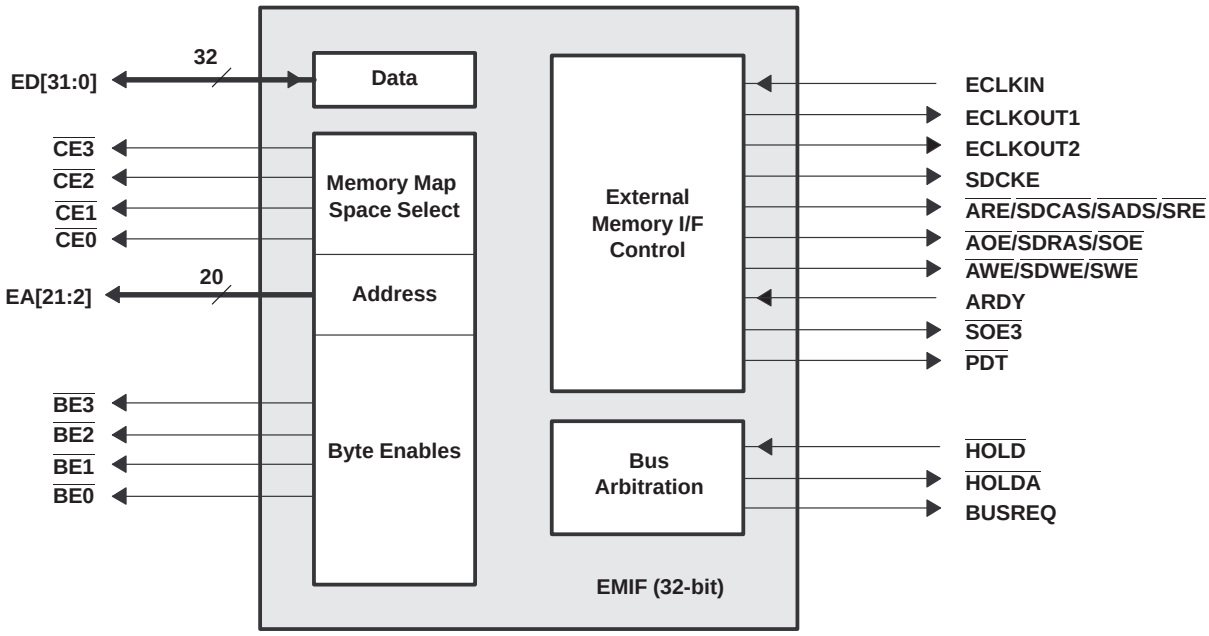
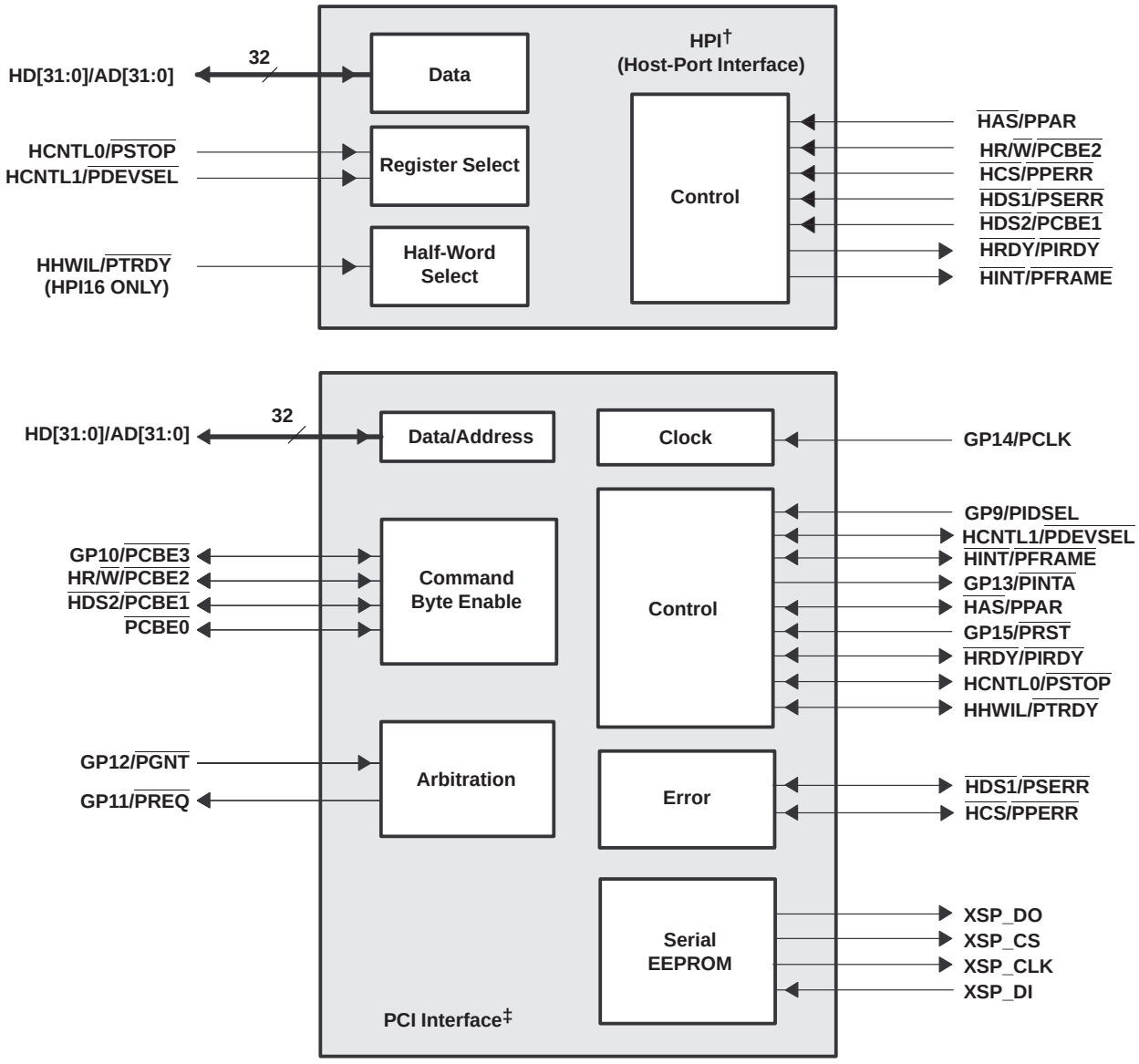



**Figure 3. Peripheral Signals**

PRODUCT PREVIEW

## signal groups description (continued)




† 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.

‡ These PCI pins (excluding PCBE0 , XSP_DO, XSP_CLK, XSP_DI, and XSP_CS) are muxed with the HPI or GPIO peripherals. By default, these signals function as HPI and no function, respectively. For more details on these muxed pins, see the Device Configurations section of this data sheet.

**Figure 3. Peripheral Signals (Continued)**

**signal groups description (continued)**

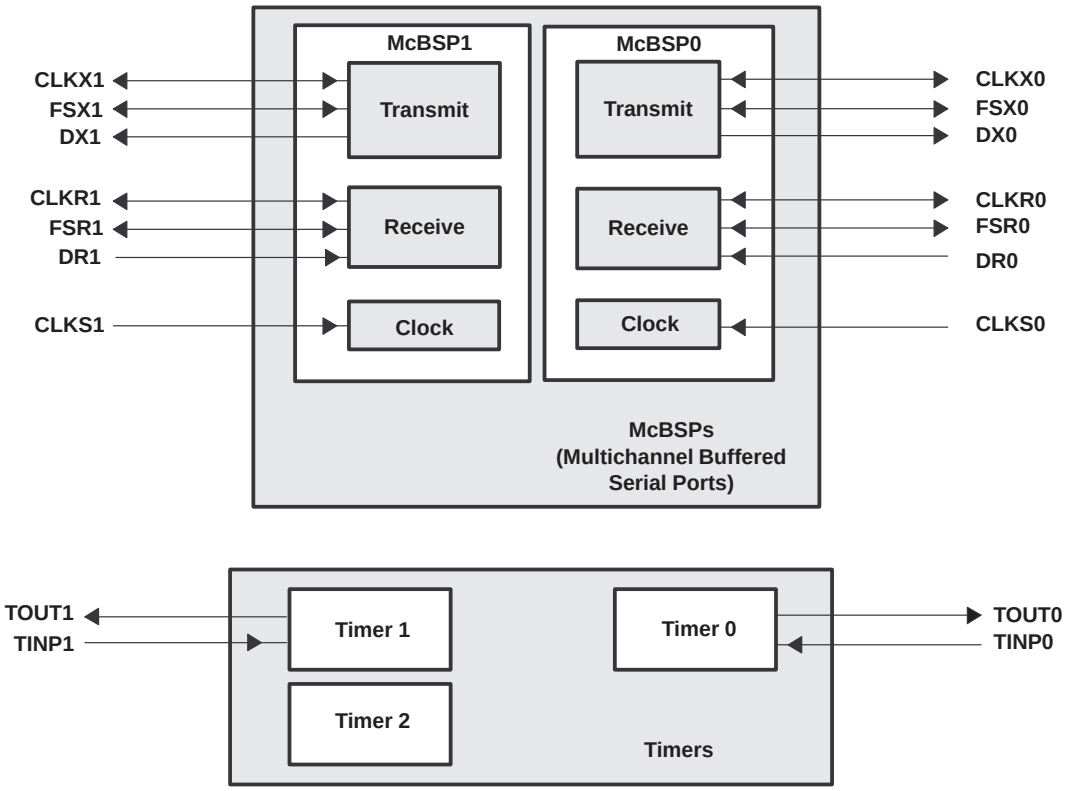



**Figure 3. Peripheral Signals (Continued)**

PRODUCT PREVIEW

## DEVICE CONFIGURATIONS

The C6411 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
  - PCI_EN
- other device configurations
  - LEND
  - BOOTMODE [1:0]
  - ECLKIN_SEL[1:0]
  - EEAI
  - HD5/AD5

### peripherals selection

Some C6411 peripherals share the same pins (internally muxed) and are mutually exclusive (i.e., HPI, general-purpose input/output pins GP[15:9], and PCI). Other C6411 peripherals (i.e., EMIF, three Timers, two McBSPs, and the GP[8:0] pins), are always available.

- HPI/GP[15:9] versus PCI

  The PCI_EN pin is latched at reset. This pin determines the HPI/GP[15:9] versus the PCI peripheral selection, summarized in Table 21.

## DEVICE CONFIGURATIONS (CONTINUED)

**Table 21. PCI_EN Peripheral Selection (HPI/GP[15:9] or PCI)**

| PCI_EN Pin† | PERIPHERALS SELECTED | | | |
|---|---|---|---|---|
| | HPI | GP[15:9] | PCI | DESCRIPTION |
| 0 | √ | √ | | [default] HPI is enabled, GP[15:9] pins can be programmed as GPIO, PCI is disabled.<br><br>This means all multiplexed HPI/PCI pins function as HPI and all standalone PCI pins ($\overline{\text{PCBE0}}$, XSP_DO, XSP_DI, XSP_CLK, 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 (GPxEN) and direction (GPxDIR) registers (for more details, see Table 23). |
| 1 | | | √ | PCI is enabled, HPI/GP[15:9] are disabled.<br>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 23).<br><br>Auto-initialization through PCI EEPROM or initialization via specified PCI default values is controlled by the EEAI pin, see Table 22. |

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

### other device configurations

Table 22 describes the C6411 device configuration pins, which are set up via external pullup/pulldown resistors through the specified pins. For more details on these device configuration pins, see the Terminal Functions table and the Debugging Considerations section.

## DEVICE CONFIGURATIONS (CONTINUED)

**Table 22. Device Configuration Pins (LEND, BOOTMODE[1:0], ECLKIN_SEL[1:0], EEAI, and HD5/AD5)**

| CONFIGURATION PIN | NO. | FUNCTIONAL DESCRIPTION |
|---|---|---|
| LEND | | Device Endian mode (LEND)<br>0 – System operates in Big Endian mode<br>1 – System operates in Little Endian mode (default) |
| BOOTMODE[1:0] | | Bootmode [1:0]. Default is reserved. External pullup and/or pulldown resistors must be used to select a valid bootmode configuration.<br>00 – No boot<br>01 – HPI boot<br>10 – Reserved (default mode)<br>11 – EMIF 8-bit ROM boot with default timings |
| ECLKIN_SEL[1:0] | | EMIF input clock select<br>Clock mode select for EMIF (ECLKIN_SEL[1:0])<br>00 – ECLKIN (default mode)<br>01 – CPU/4 Clock Rate<br>10 – CPU/6 Clock Rate<br>11 – Reserved |
| EEAI | | PCI EEPROM Auto-Initialization (EEAI)<br>PCI auto-initialization via external EEPROM<br>0 – PCI auto-initialization through EEPROM is disabled; the PCI peripheral uses the specified PCI default values (default).<br>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).<br><br>Note: ***If*** the PCI peripheral is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up.<br>For more information on the PCI EEPROM default values, see the PCI chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190). |
| HD5/AD5 | | HPI configuration bus width (HPI_WIDTH)<br>0 – HPI operates as an HPI16.<br>(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.)<br>1 – HPI operates as an HPI32.<br>(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 23 identifies the multiplexed pins on the C6411 device; 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 CLKMODE0, LEND, BOOTMODE[1:0], ECLKIN_SEL[1:0]. EEAI, HD5/AD5, and PCI_EN. Although internal pullup/pulldown resistors exist on these pins, providing external connectivity adds convenience to the user in debugging and flexibility in switching operating modes.

Internal pullup/pulldown resistors also exist on specified reserved (RSV) pins. Do not oppose the internal pullup/pulldown resistors, unless otherwised noted, on these RSV pins.

For the internal pullup/pulldown resistors for all device pins, see the terminal functions table.

## DEVICE CONFIGURATIONS (CONTINUED)

### Table 23. C6411 Device Multiplexed Pins†

| MULTIPLEXED PINS NAME | NO. | DEFAULT FUNCTION | DEFAULT SETTING | DESCRIPTION |
|---|---|---|---|---|
| CLKOUT4/GP1 | | 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.<br>GPxEN = 1: GPx pin enabled<br>GPxDIR = 0: GPx pin is an input<br>GPxDIR = 1: GPx pin is an output |
| CLKOUT6/GP2 | | CLKOUT6 | GP2EN = 0 (disabled) | |
| GP9/PIDSEL | | None | GPxEN = 0 (disabled)<br>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.<br>GPxEN = 1: GPx pin enabled<br>GPxDIR = 0: GPx pin is an input<br>GPxDIR = 1: GPx pin is an output |
| GP10/$\overline{\text{PCBE3}}$ | | | | |
| GP11/$\overline{\text{PREQ}}$ | | | | |
| GP12/$\overline{\text{PGNT}}$ | | | | |
| GP13/$\overline{\text{PINTA}}$ | | | | |
| GP14/PCLK | | | | |
| GP15/$\overline{\text{PRST}}$ | | | | |
| HD[31:0]/AD[31:0] | ‡ | HD[31:0] | PCI_EN = 0 (disabled)† | By default, HPI is enabled upon reset (PCI is disabled).<br>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 and keeping valid “1” after reset). |
| $\overline{\text{HAS}}$/PPAR | | $\overline{\text{HAS}}$ | | |
| HCNTL1/$\overline{\text{PDEVSEL}}$ | | HCNTL1 | | |
| HCNTL0/$\overline{\text{PSTOP}}$ | | HCNTL0 | | |
| $\overline{\text{HDS1}}$/$\overline{\text{PSERR}}$ | | $\overline{\text{HDS1}}$ | | |
| $\overline{\text{HDS2}}$/$\overline{\text{PCBE1}}$ | | $\overline{\text{HDS2}}$ | | |
| HR/$\overline{\text{W}}$/$\overline{\text{PCBE2}}$ | | HR/$\overline{\text{W}}$ | | |
| HWWIL/$\overline{\text{PTRDY}}$ | | HHWIL (HPI16 only) | | |
| $\overline{\text{HINT}}$/$\overline{\text{PFRAME}}$ | | $\overline{\text{HINT}}$ | | |
| $\overline{\text{HCS}}$/$\overline{\text{PPERR}}$ | | $\overline{\text{HCS}}$ | | |
| $\overline{\text{HRDY}}$/PIRDY | | $\overline{\text{HRDY}}$ | | |

† All other standalone PCI pins are tied-off internally (pins in Hi-Z) when the peripheral is disabled [PCI_EN = 0].
‡ 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 | | I | IPD | Clock Input. This clock is the input to the on-chip PLL. |
| CLKOUT4/GP1§ | | 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§ | | 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**). |
| CLKMODE0 | | I | IPD | Clock mode select<br>• Selects whether the CPU clock frequency = input clock frequency x1 (Bypass) [default] or x6.<br>For more details on the CLKMODE0 pin and the PLL multiply factors, see the Clock PLL section of this data sheet. |
| PLLV¶ | | A# | | PLL voltage supply |
| **JTAG EMULATION** | | | | |
| TMS | | I | IPU | JTAG test-port mode select |
| TDO | | O/Z | IPU | JTAG test-port data out |
| TDI | | I | IPU | JTAG test-port data in |
| TCK | | I | IPU | JTAG test-port clock |
| $\overline{\text{TRST}}$ | | I | IPD | JTAG test-port reset |
| EMU9 | | I/O/Z | IPU | Emulation pin 9. Reserved for future use, leave unconnected. |
| EMU8 | | I/O/Z | IPU | Emulation pin 8. Reserved for future use, leave unconnected. |
| EMU7 | | I/O/Z | IPU | Emulation pin 7. Reserved for future use, leave unconnected. |
| EMU6 | | I/O/Z | IPU | Emulation pin 6. Reserved for future use, leave unconnected. |
| EMU5 | | I/O/Z | IPU | Emulation pin 5. Reserved for future use, leave unconnected. |
| EMU4 | | I/O/Z | IPU | Emulation pin 4. Reserved for future use, leave unconnected. |
| EMU3 | | I/O/Z | IPU | Emulation pin 3. Reserved for future use, leave unconnected. |
| EMU2 | | I/O/Z | IPU | Emulation pin 2. Reserved for future use, leave unconnected. |
| EMU1 | | I/O/Z | IPU | Emulation pin 1‖ |
| EMU0 | | I/O/Z | IPU | Emulation pin 0‖ |
| EMUCLK1 | | I/O/Z | IPU | Emulation clock 1. Reserved for future use, leave unconnected. |
| EMUCLK0 | | I/O/Z | IPU | Emulation clock 0. Reserved for future use, leave unconnected. |
| **LITTLE/BIG ENDIAN FORMAT** | | | | |
| LEND | | I/O/Z | IPU | Device Endian mode<br>LEND: 0 – Big Endian<br>1 – Little Endian (default mode) |

† 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, unless otherwise noted.)

§ 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)

‖ The EMU0 and EMU1 pins are internally pulled up with 30-kΩ resistors; therefore, for emulation and normal operation, no external pullup/pulldown resistors are necessary. However, for boundary scan operation, pull down the EMU1 and EMU0 pins with a dedicated 1-kΩ resistor.

PRODUCT PREVIEW

## Terminal Functions (Continued)

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **BOOTMODE** | | | | |
| BOOTMODE1 | | I/O/Z | IPU | Boot mode. Default is reserved. External pullup and/or pulldown resistors must be used to select a valid bootmode configuration.<br>BOOTMODE[1:0]: 00 – No boot<br>01 – HPI boot<br>10 – Reserved (default mode)<br>11 – EMIF 8-bit ROM boot with default timings |
| BOOTMODE0 | | | IPD | |
| **RESETS, INTERRUPTS, AND GENERAL-PURPOSE INPUT/OUTPUTS** | | | | |
| $\overline{\text{RESET}}$ | | I | | Device reset |
| NMI | | I | IPD | Nonmaskable interrupt, edge-driven (rising edge) |
| GP7/EXT_INT7 | | 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.<br>• 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 | | | | |
| GP5/EXT_INT5 | | | | |
| GP4/EXT_INT4 | | | | |
| GP15/$\overline{\text{PRST}}$§ | | I/O/Z | | General-purpose input/output (GPIO) 15 pin (**I/O/Z**) or PCI reset (**I**). No function at default. |
| GP14/PCLK§ | | | | GPIO 14 pin (**I/O/Z**) or PCI clock (**I**). No function at default. |
| GP13/$\overline{\text{PINTA}}$§ | | | | GPIO 13 pin (**I/O/Z**) or PCI interrupt A (**O/Z**). No function at default. |
| GP12/$\overline{\text{PGNT}}$§ | | | | GPIO 12 pin (**I/O/Z**) or PCI bus grant (**I**). No function at default. |
| GP11/$\overline{\text{PREQ}}$§ | | | | GPIO 11 pin (**I/O/Z**) or PCI bus request (**O/Z**). No function at default. |
| GP10/$\overline{\text{PCBE3}}$§ | | | | GPIO 10 pin (**I/O/Z**) or PCI command/byte enable 3 (**I/O/Z**). No function at default. |
| GP9/PIDSEL§ | | | | GPIO 9 pin (**I/O/Z**) or PCI initialization device select (**I**). No function at default. |
| GP3 | | | IPD | GPIO 3 pin (**I/O/Z**). |
| GP0 | | | IPD | GPIO 0 pin.<br>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**). |
| GP8 | | I/O/Z | IPD | This pin has no function at default [default] or this pin can be programmed as a GPIO 8 pin (**I/O/Z**). |
| CLKOUT6/GP2§ | | 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§ | | 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) or PERIPHERAL COMPONENT INTERCONNECT (PCI)** | | | | |
| PCI_EN | | I | IPD | PCI enable pin. This pin controls the selection (enable/disable) of the HPI and GP[15:9], or PCI peripherals. For more details, see the Device Configurations section of this data sheet. |
| EEAI | | I/O/Z | IPD | PCI EEPROM Auto-Initialization (EEAI) via external EEPROM<br>***If*** the PCI peripheral is disabled (PCI_EN pin = 0), this pin must ***not*** be pulled up.<br>EEAI: 0 – PCI auto-initialization through EEPROM is disabled (default).<br>1 – PCI auto-initialization through EEPROM is enabled. |

† 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, unless otherwise noted.)

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

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **HOST-PORT INTERFACE (HPI) or PERIPHERAL COMPONENT INTERCONNECT (PCI) (CONTINUED)** | | | | |
| $\overline{\text{HINT}}$/ $\overline{\text{PFRAME}}$§ | | I/O/Z | | Host interrupt from DSP to host (**O**) [default] or PCI frame (**I/O/Z**) |
| HCNTL1/ $\overline{\text{PDEVSEL}}$§ | | I/O/Z | | Host control – selects between control, address, or data registers (**I**) [default] or PCI device select (**I/O/Z**). |
| HCNTL0/ $\overline{\text{PSTOP}}$§ | | I/O/Z | | Host control – selects between control, address, or data registers (**I**) [default] or PCI stop (**I/O/Z**) |
| HHWIL/ $\overline{\text{PTRDY}}$§ | | 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/$\overline{\text{W}}$/$\overline{\text{PCBE2}}$§ | | I/O/Z | | Host read or write select (**I**) [default] or PCI command/byte enable 2 (**I/O/Z**) |
| $\overline{\text{HAS}}$/PPAR§ | | I/O/Z | | Host address strobe (**I**) [default] or PCI parity (**I/O/Z**) |
| $\overline{\text{HCS}}$/$\overline{\text{PPERR}}$§ | | I/O/Z | | Host chip select (**I**) [default] or PCI parity error (**I/O/Z**) |
| $\overline{\text{HDS1}}$/$\overline{\text{PSERR}}$§ | | I/O/Z | | Host data strobe 1 (**I**) [default] or PCI system error (**I/O/Z**) |
| $\overline{\text{HDS2}}$/$\overline{\text{PCBE1}}$§ | | I/O/Z | | Host data strobe 2 (**I**) [default] or PCI command/byte enable 1 (**I/O/Z**) |
| $\overline{\text{HRDY}}$/$\overline{\text{PIRDY}}$§ | | I/O/Z | | Host ready from DSP to host (**O**) [default] or PCI initiator ready (**I/O/Z**). |
| HD31/AD31§ | | I/O/Z | | Host-port data (**I/O/Z**) [default] or PCI data-address bus (**I/O/Z**)<br><br>As HPI data bus (PCI_EN pin = 0)<br>• Used for transfer of data, address, and control<br>• Host-Port bus width user-configurable at device reset via a 10-kΩ resistor pullup/pulldown resistor on the HD5 pin:<br><br>HD5 pin = 0: HPI operates as an HPI16.<br>(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.)<br><br>HD5 pin = 1: HPI operates as an HPI32.<br>(HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.)<br><br>As PCI data-address bus (PCI_EN pin = 1)<br>• Used for transfer of data and address |
| HD30/AD30§ | | | | |
| HD29/AD29§ | | | | |
| HD28/AD28§ | | | | |
| HD27/AD27§ | | | | |
| HD26/AD26§ | | | | |
| HD25/AD25§ | | | | |
| HD24/AD24§ | | | | |
| HD23/AD23§ | | | | |
| HD22/AD22§ | | | | |
| HD21/AD21§ | | | | |
| HD20/AD20§ | | | | |
| HD19/AD19§ | | | | |
| HD18/AD18§ | | | | |
| HD17/AD17§ | | | | |
| HD16/AD16§ | | | | |
| HD15/AD15§ | | | | |
| HD14/AD14§ | | | | |
| HD13/AD13§ | | | | |
| HD12/AD12§ | | | | |
| HD11/AD11§ | | | | |
| HD10/AD10§ | | | | |
| HD9/AD9§ | | | | |
| HD8/AD8§ | | | | |

† 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, unless otherwise noted.)
§ These pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet.

PRODUCT PREVIEW

PRODUCT PREVIEW

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **HOST-PORT INTERFACE (HPI) or PERIPHERAL COMPONENT INTERCONNECT (PCI) (CONTINUED)** | | | | |
| HD7/AD7§ | | I/O/Z | | Host-port data (**I/O/Z**) [default] or PCI data-address bus (**I/O/Z**)<br><br>As HPI data bus (PCI_EN pin = 0)<br>• Used for transfer of data, address, and control<br>• Host-Port bus width user-configurable at device reset via a 10-kΩ resistor pullup/pulldown resistor on the HD5 pin:<br><br>HD5 pin = 0: HPI operates as an HPI16.<br>(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.)<br><br>HD5 pin = 1: HPI operates as an HPI32.<br>(HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.)<br><br>As PCI data-address bus (PCI_EN pin = 1)<br>• Used for transfer of data and address |
| HD6/AD6§ | | | | |
| HD5/AD5§ | | | | |
| HD4/AD4§ | | | | |
| HD3/AD3§ | | | | |
| HD2/AD2§ | | | | |
| HD1/AD1§ | | | | |
| HD0/AD0§ | | | | |
| $\overline{\text{PCBE0}}$§ | | I/O/Z | | PCI command/byte enable 0 (**I/O/Z**). When PCI is disabled (PCI_EN = 0), this pin is tied-off. |
| XSP_CS | | O | IPD | PCI serial interface chip select (**O**). When PCI is disabled (PCI_EN = 0), this pin is tied-off. |
| XSP_CLK | | I/O/Z | IPD | This pin has no function at default [default] or when PCI is enabled (PCI_EN = 0), this pin is the PCI serial interface clock (**O**). |
| XSP_DI | | I | IPU | This pin has no function at default [default] or when PCI is enabled (PCI_EN = 0), this pin is the PCI serial interface data in (**I**). In PCI mode, this pin is connected to the output data pin of the serial PROM. |
| XSP_DO | | O/Z | IPU | This pin has no function at default [default] or when PCI is enabled (PCI_EN = 0), this pin is the PCI serial interface data out (**O**). In PCI mode, this pin is connected to the input data pin of the serial PROM. |
| GP15/$\overline{\text{PRST}}$§ | | I/O/Z | | General-purpose input/output (GPIO) 15 pin (**I/O/Z**) or PCI reset (**I**). No function at default. |
| GP14/PCLK§ | | | | GPIO 14 pin (**I/O/Z**) or PCI clock (**I**). No function at default. |
| GP13/$\overline{\text{PINTA}}$§ | | | | GPIO 13 pin (**I/O/Z**) or PCI interrupt A (**O/Z**). No function at default. |
| GP12/$\overline{\text{PGNT}}$§ | | | | GPIO 12 pin (**I/O/Z**) or PCI bus grant (**I**). No function at default. |
| GP11/$\overline{\text{PREQ}}$§ | | | | GPIO 11 pin (**I/O/Z**) or PCI bus request (**O/Z**). No function at default. |
| GP10/$\overline{\text{PCBE3}}$§ | | | | GPIO 10 pin (**I/O/Z**) or PCI command/byte enable 3 (**I/O/Z**). No function at default. |
| GP9/PIDSEL§ | | | | GPIO 9 pin (**I/O/Z**) or PCI initialization device select (**I**). No function at default. |
| $\overline{\text{CE3}}$ | | O/Z | IPU | EMIF memory space enables<br>• Enabled by bits 28 through 31 of the word address<br>• Only one pin is asserted during any external data access |
| $\overline{\text{CE2}}$ | | O/Z | IPU | |
| $\overline{\text{CE1}}$ | | O/Z | IPU | |
| $\overline{\text{CE0}}$ | | O/Z | IPU | |

† 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, unless otherwise noted.)
§ These pins are multiplexed pins. For more details, see the Device Configurations section of this data sheet.

## Terminal Functions (Continued)

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **EMIF (32-bit) – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY (CONTINUED)** | | | | |
| $\overline{BE3}$ | | O/Z | IPU | EMIF byte-enable control<br>• Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory.<br>• Byte-write enables for most types of memory<br>• Can be directly connected to SDRAM read and write mask signal (SDQM) |
| $\overline{BE2}$ | | O/Z | IPU | |
| $\overline{BE1}$ | | O/Z | IPU | |
| $\overline{BE0}$ | | O/Z | IPU | |
| $\overline{PDT}$ | | O/Z | IPU | EMIF peripheral data transfer, allows direct transfer between external peripherals |
| **EMIF (32-BIT) – BUS ARBITRATION** | | | | |
| $\overline{HOLDA}$ | | O | IPU | EMIF hold-request-acknowledge to the host |
| $\overline{HOLD}$ | | I | IPU | EMIF hold request from the host |
| BUSREQ | | O | IPU | EMIF bus request output |
| **EMIF (32-BIT) – ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL** | | | | |
| ECLKIN_SEL1<br>ECLKIN_SEL0 | | I/O/Z | IPD | EMIF clock mode select<br>ECLKIN_SEL[1:0]: 00 – ECLKIN (default mode)<br>01 – CPU/4 Clock Rate<br>10 – CPU/6 Clock Rate<br>11 – Reserved |
| ECLKIN | | I | IPD | EMIF external input clock. The EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) is selected at reset via the pullup/pulldown resistors on the ECLKIN_SEL[1:0] pins. ECLKIN is the default for the EMIF input clock. |
| ECLKOUT2 | | O/Z | IPD | EMIF output clock 2. Programmable to be EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) frequency divided-by-1, -2, or -4. |
| ECLKOUT1 | | O/Z | IPD | EMIF output clock 1 at EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) frequency. |
| $\overline{ARE}$/ $\overline{SDCAS}$/ $\overline{SADS}$/$\overline{SRE}$ | | O/Z | IPU | EMIF asynchronous memory read-enable/SDRAM column-address strobe/programmable synchronous interface-address strobe or read-enable<br>• For programmable synchronous interface, the RENEN field in the CE Space Secondary Control Register (CExSEC) selects between $\overline{SADS}$ and $\overline{SRE}$:<br>If RENEN = 0, then the $\overline{SADS}$/$\overline{SRE}$ signal functions as the $\overline{SADS}$ signal.<br>If RENEN = 1, then the $\overline{SADS}$/$\overline{SRE}$ signal functions as the $\overline{SRE}$ signal. |
| $\overline{AOE}$/ $\overline{SDRAS}$/ $\overline{SOE}$ | | O/Z | IPU | EMIF asynchronous memory output-enable/SDRAM row-address strobe/programmable synchronous interface output-enable |
| $\overline{AWE}$/ $\overline{SDWE}$/ $\overline{SWE}$ | | O/Z | IPU | EMIF asynchronous memory write-enable/SDRAM write-enable/programmable synchronous interface write-enable |
| SDCKE | | O/Z | IPU | EMIF SDRAM clock-enable (used for self-refresh mode).<br>• If SDRAM is not in system, SDCKE can be used as a general-purpose output. |
| $\overline{SOE3}$ | | O/Z | IPU | EMIF synchronous memory output-enable for $\overline{CE3}$ (for glueless FIFO interface) |
| ARDY | | I | IPU | 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, unless otherwise noted.)

PRODUCT PREVIEW

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **EMIF (32-BIT) – ADDRESS** | | | | |
| EA21 | | O/Z | IPD | EMIF external address (word address) |
| EA20 | | | | |
| EA19 | | | | |
| EA18 | | | | |
| EA17 | | | | |
| EA16 | | | | |
| EA15 | | | | |
| EA14 | | | | |
| EA13 | | | | |
| EA12 | | | | |
| EA11 | | | | |
| EA10 | | | | |
| EA9 | | | | |
| EA8 | | | | |
| EA7 | | | | |
| EA6 | | | | |
| EA5 | | | | |
| EA4 | | | | |
| EA3 | | | | |
| EA2 | | | | |
| **EMIF (32-bit) – DATA** | | | | |
| ED31 | | I/O/Z | IPU | EMIF external data |
| ED30 | | | | |
| ED29 | | | | |
| ED28 | | | | |
| ED27 | | | | |
| ED26 | | | | |
| ED25 | | | | |
| ED24 | | | | |
| ED23 | | | | |
| ED22 | | | | |
| ED21 | | | | |
| ED20 | | | | |
| ED19 | | | | |
| ED18 | | | | |
| ED17 | | | | |
| ED16 | | | | |

† 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, unless otherwise noted.)

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **EMIF (32-bit) – DATA (CONTINUED)** | | | | |
| ED15 | | I/O/Z | IPU | EMIF external data |
| ED14 | | | | |
| ED13 | | | | |
| ED12 | | | | |
| ED11 | | | | |
| ED10 | | | | |
| ED9 | | | | |
| ED8 | | | | |
| ED7 | | | | |
| ED6 | | | | |
| ED5 | | | | |
| ED4 | | | | |
| ED3 | | | | |
| ED2 | | | | |
| ED1 | | | | |
| ED0 | | | | |
| **TIMER 1** | | | | |
| TOUT1 | | O/Z | IPD | Timer 1 or general-purpose output |
| TINP1 | | I | IPD | Timer 1 or general-purpose input |
| **TIMER 0** | | | | |
| TOUT0 | | O/Z | IPD | Timer 0 or general-purpose output |
| TINP0 | | I | IPD | Timer 0 or general-purpose input |
| **MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)** | | | | |
| CLKS1 | | I | | McBSP1 external clock source (as opposed to internal) |
| CLKR1 | | I/O/Z | | McBSP1 receive clock |
| CLKX1 | | I/O/Z | | McBSP1 transmit clock |
| DR1 | | I | | McBSP1 receive data |
| DX1 | | O/Z | | McBSP1 transmit data |
| FSR1 | | I/O/Z | | McBSP1 receive frame sync |
| FSX1 | | I/O/Z | | McBSP1 transmit frame sync |
| **MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)** | | | | |
| CLKS0 | | I | IPD | McBSP0 external clock source (as opposed to internal) |
| CLKR0 | | I/O/Z | IPD | McBSP0 receive clock |
| CLKX0 | | I/O/Z | IPD | McBSP0 transmit clock |
| DR0 | | I | IPU | McBSP0 receive data |
| DX0 | | O/Z | IPU | McBSP0 transmit data |
| FSR0 | | I/O/Z | IPD | McBSP0 receive frame sync |
| FSX0 | | I/O/Z | IPD | McBSP0 transmit frame sync |

† 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, unless otherwise noted.)

PRODUCT PREVIEW

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **RESERVED FOR TEST** | | | | |
| RSV | | Z | | **For proper device operation, these RSV pins (AD11, AD12) must be externally pulled down to ground with a 10-kΩ resistor.** |
| | | Z | | |
| RSV | | | IPD | **For proper device operation, this RSV pin (D15) must be externally pulled up to $DV_{DD}$ with a 1-kΩ resistor.** |
| RSV | | | | Reserved. These pins must be connected directly to $CV_{DD}$ for proper device operation. |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| RSV | | | | Reserved. This pin must be connected directly to $DV_{DD}$ for proper device operation. |
| RSV | | | | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPD | |
| | | | IPD | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPD | |
| | | Z | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | 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, unless otherwise noted.)

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **RESERVED FOR TEST (CONTINUED)** | | | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | Z | | |
| | | Z | | |
| RSV | | Z | | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | Z | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |

† 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, unless otherwise noted.)

PRODUCT PREVIEW

**Terminal Functions (Continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| RESERVED FOR TEST (CONTINUED) | | | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPD | |
| | | | IPU | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| RSV | | | IPD | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| | | | IPD | |
| | | | IPD | |
| | | | IPD | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPD | |
| | | | IPU | |
| | | | IPD | |
| | | | IPD | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |

† 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, unless otherwise noted.)

**Terminal Functions (Continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE† | IPD/ IPU‡ | DESCRIPTION |
|---|---|---|---|---|
| **RESERVED FOR TEST (CONTINUED)** | | | | |
| RSV | | | IPD | Reserved (leave unconnected, ***do not*** connect to power or ground) |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPD | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |
| | | | IPU | |

† 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, unless otherwise noted.)

**Terminal Functions (Continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE† | DESCRIPTION |
|---|---|---|---|
| **SUPPLY VOLTAGE PINS** | | | |
| $DV_{DD}$ | | S | 3.3-V supply voltage |

† 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)** | | | |
| $DV_{DD}$ | | | 3.3-V supply voltage |
| $CV_{DD}$ | | S | 1-V supply voltage |

† 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)** | | | |
| $CV_{DD}$ | | S | 1-V supply voltage |

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

**Terminal Functions (Continued)**

| SIGNAL NAME | NO. | TYPE† | DESCRIPTION |
|---|---|---|---|
| **GROUND PINS** | | | |
| $V_{SS}$ | | GND | Ground pins |

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

**Terminal Functions (Continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE† | DESCRIPTION |
|---|---|---|---|
| **GROUND PINS (CONTINUED)** | | | |
| $V_{SS}$ | | GND | Ground pins |

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

**Terminal Functions (Continued)**

| SIGNAL NAME | SIGNAL NO. | TYPE† | DESCRIPTION |
|---|---|---|---|
| **GROUND PINS (CONTINUED)** | | | |
| $V_{SS}$ | | GND | Ground pins |

† 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)

The *TMS320 DSP Development Support Reference Guide* (SPRU011) contains information about development-support products for all TMS320™ DSP family member devices, including documentation. See this document for further information on TMS320™ DSP documentation or any TMS320™ DSP support products from Texas Instruments. An additional document, the *TMS320 Third-Party Support Reference Guide* (SPRU052), contains information about TMS320™ DSP-related products from other companies in the industry. To receive TMS320™ DSP literature, contact the Literature Response Center at 800/477-8924.

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.

**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 TMS320™ DSP family member has one of three prefixes: TMX, TMP, or TMS. 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 (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

Device development evolutionary flow:

**TMX** Experimental device that is not necessarily representative of the final device's electrical specifications

**TMP** Final silicon die that conforms to the device's electrical specifications but has not completed quality and reliability verification

**TMS** Fully qualified production device

Support tool development evolutionary flow:

**TMDX** Development-support product that has not yet completed Texas Instruments internal qualification testing.

**TMDS** Fully qualified development-support product

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

"Developmental product is intended for internal evaluation purposes."

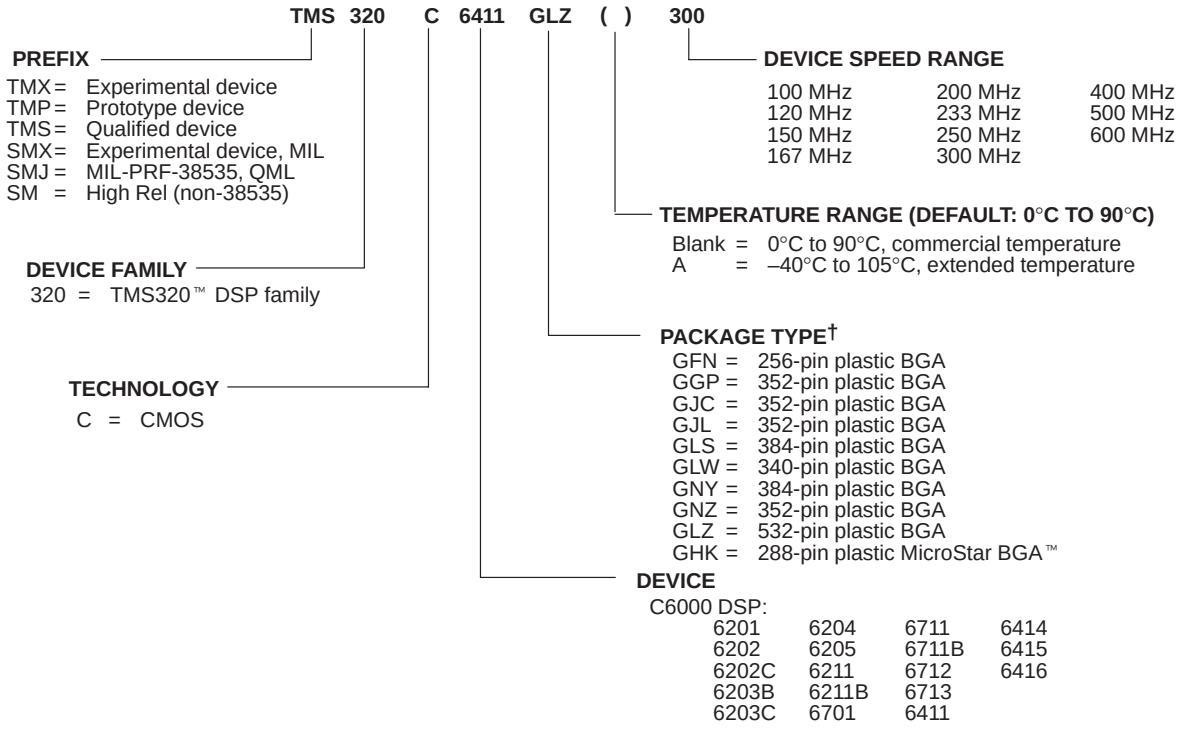
TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

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

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 TMS320™ DSP family member has one of three prefixes: TMX, TMP, or TMS. 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 (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS).

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, GLZ), the temperature range (for example, blank is the default commercial temperature range), and the device speed range in megahertz (for example, -300 is 300 MHz). Figure 4 provides a legend for reading the complete device name for any TMS320C6000™ DSP platform member.

**device and development-support tool nomenclature (continued)**




† BGA = Ball Grid Array

**Figure 4. TMS320C6000™ DSP Device Nomenclature (Including the TMS320C6411 Device)**

PRODUCT PREVIEW

MicroStar BGA is a trademark of Texas Instruments.

## 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 Peripherals Reference Guide* (literature number SPRU190) describes the functionality of the peripherals available on the C6000™ DSP platform of devices, such as the 64-/32-/16-bit external memory interfaces (EMIFs), direct-memory-access (DMA), enhanced direct-memory-access (EDMA) controller, multichannel buffered serial ports (McBSPs), an 8-bit Universal Test and Operations PHY Interface for ATM Slave (UTOPIA Slave) port, 32-/16-bit host-port interfaces (HPIs), a peripheral component interconnect (PCI), expansion bus (XB), peripheral component interconnect (PCI), clocking and phase-locked loop (PLL); general-purpose timers, general-purpose input/output (GPIO) port, and power-down modes. This guide also includes information on internal data and program memories.

The *TMS320C6000 Technical Brief* (literature number SPRU197) gives an introduction to the C62x™/C67x™ devices, associated development tools, and third-party support.

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 Fixed-Point Digital Signal Processor* data sheet (literature number SPRS134) describes the features of the TMS320C6414 fixed-point DSP and provides pinouts, electrical specifications, and timings for the device.

The *TMS320C6416 Fixed-Point Digital Signal Processor* data sheet (literature number SPRS164) describes the features of the TMS320C6416 fixed-point DSP and provides pinouts, electrical specifications, and timings for the device.

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 Texas Instruments for more detailed information on the device compatibility and similarities/differences among the C6211, C6411, C6414, C6415, and C6416 devices, see the *How To Begin Development Today With the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs* application report (literature number SPRA718) and *How To Begin Development Today With the TMS320C6411 DSP* application report (literature number SPRA374).



C67x is a trademark of Texas Instruments.

## 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.

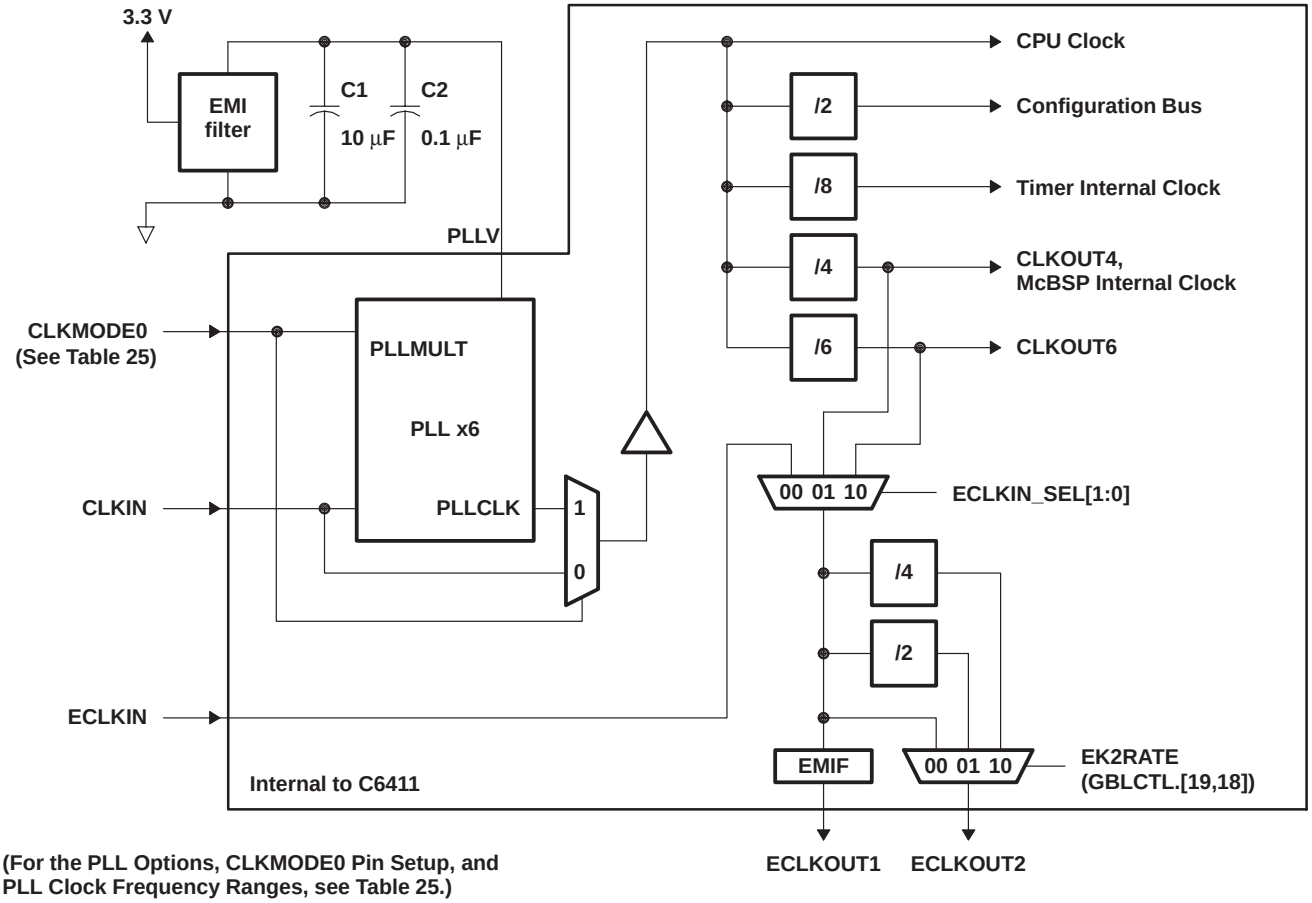
To minimize the clock jitter, a single clean power supply should power both the C64x™ DSP device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see the *input and output clocks* electricals section.

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 suppy voltage and operating case temperature* table and the *input and output clocks* electricals section). Table 24 lists some examples of compatible CLKIN external clock sources:

**Table 24. 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 | MK1711-S, ICS525-02 | Integrated Circuit Systems |

## clock PLL (continued)




**(For the PLL Options, CLKMODE0 Pin Setup, and PLL Clock Frequency Ranges, see Table 25.)**

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 25. TMS320C6411 PLL Multiply Factor Options, Clock Frequency Ranges, and Typical Lock Time†‡**

| GLZ PACKAGE – 23 x 23 mm BGA | | | | | | |
|---|---|---|---|---|---|---|
| CLKMODE0 | CLKMODE (PLL MULTIPLY FACTORS) | CLKIN RANGE (MHz) | CPU CLOCK FREQUENCY RANGE (MHz) | CLKOUT4 RANGE (MHz) | CLKOUT6 RANGE (MHz) | TYPICAL LOCK TIME (μs)§ |
| 0 | Bypass (x1) [default] | 30–75 | 30–75 | 7.5–18.8 | 5–12.5 | N/A |
| 1 | x6 | 30–50 | 180–300 | 45–75 | 30–50 | 75 |

† These clock frequency range values are applicable to a C6411–300 speed device.
‡ Use an external pullup resistor on the CLKMODE0 pin to set the C6411 device to the valid PLL multiply clock mode (x6). With an internal pulldown resistor on the CLKMODE0 pin, 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.

## 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 if the other supply is below the proper operating voltage.

### system-level design considerations

System-level design considerations, such as bus contention, may require supply sequencing to be implemented. In this case, the core supply should be powered up at the same time as, or prior to (and powered down after), the I/O buffers. This is to ensure that the I/O buffers receive valid inputs from the core before the output buffers are powered up, thus, preventing bus contention with other chips on the board.

### 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.

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.

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

Supply voltage ranges: $CV_{DD}$ (see Note 1) . . . . . – 0.3 V to 2.3 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), $V_{IP}$ . . . . . –0.5 V to $DV_{DD}$ + 0.5 V
Output voltage ranges: (except PCI), $V_O$ . . . . . –0.3 V to 4 V
(PCI), $V_{OP}$ . . . . . –0.5 V to $DV_{DD}$ + 0.5 V
Operating case temperature range, $T_C$ . . . . . 0°C to 90°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‡ | | 0.97 | 1 | 1.03 | 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) | | –0.5 | | $DV_{DD}$ + 0.5 | V |
| $V_{IHP}$ | High-level input voltage (PCI) | | $0.5DV_{DD}$ | | $DV_{DD}$ + 0.5 | V |
| $V_{ILP}$ | Low-level input voltage (PCI) | | –0.5 | | $0.3DV_{DD}$ | V |
| $I_{OH}$ | High-level output current | except CLKOUT4 and CLKOUT6 | | | –8 | mA |
| | | CLKOUT4 and CLKOUT6 | | | –16 | mA |
| $I_{OL}$ | Low-level output current | except CLKOUT4 and CLKOUT6 | | | 8 | mA |
| | | CLKOUT4 and CLKOUT6 | | | 16 | mA |
| $T_C$ | Operating case temperature | | 0 | | 90 | °C |

‡ Future variants of the C6411 DSP may operate at voltages ranging from 1.0 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., 1.0 V, 1.1 V, 1.2 V, 1.3 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.

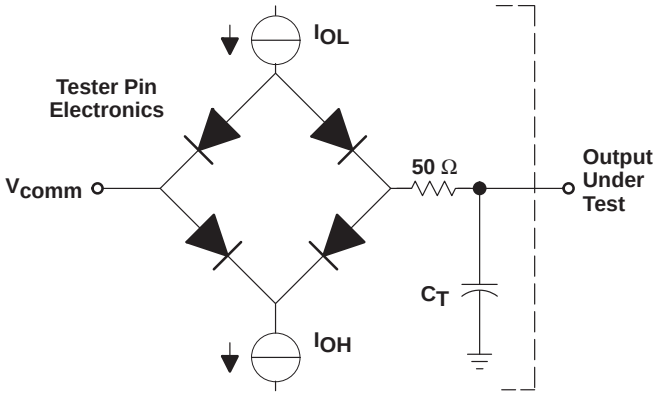
## electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

| 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) | $I_{OHP}$ = –0.5 mA, | 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) | $I_{OLP}$ = 1.5 mA, | 3.3 V | | | $0.1DV_{DD}$ | V |
| $I_I$ | Input current (except PCI) | $V_I = V_{SS}$ to $DV_{DD}$ | | | | ±150 | uA |
| $I_{IP}$ | Input leakage current (PCI)‡ | $0 < V_{IP} < DV_{DD}$, | 3.3 V | | | ±10 | uA |
| $I_{OZ}$ | Off-state output current | $V_O = DV_{DD}$ or 0 V | | | | ±10 | uA |
| $I_{CDD}$ | Core supply current§ | $CV_{DD}$ = 1 V, CPU clock = 300 MHz | | | TBD | | mA |
| $I_{DDD}$ | I/O supply current§ | $DV_{DD}$ = 3.3 V, CPU clock = 300 MHz | | | 125 | | mA |
| $C_i$ | Input capacitance | | | | | 10 | pF |
| $C_o$ | Output capacitance | | | | | 10 | pF |

† For test conditions shown as MIN, MAX, or NOM, use the appropriate value specified in the recommended operating conditions table.
‡ PCI input leakage currents include Hi-Z output leakage for all bidirectional buffers with 3-state outputs.
§ 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 *TMS320C6411 Power Consumption Summary* application report (literature number SPRA373).

PRODUCT PREVIEW

## PARAMETER MEASUREMENT INFORMATION




Where: $I_{OL}$ = 1 μA
$I_{OH}$ = 1 μA
$V_{comm}$ = 0.5 V
$C_T$ = 6.35-pF typical load-circuit capacitance

**Figure 6. Test Load Circuit for AC Timing Measurements**



### signal transition levels

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

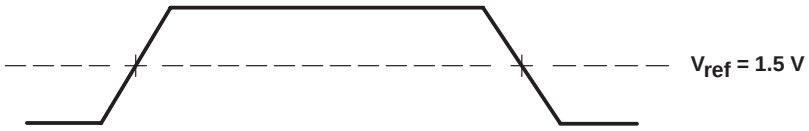



**Figure 7. Input and Output Voltage Reference Levels for AC Timing Measurements**

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.

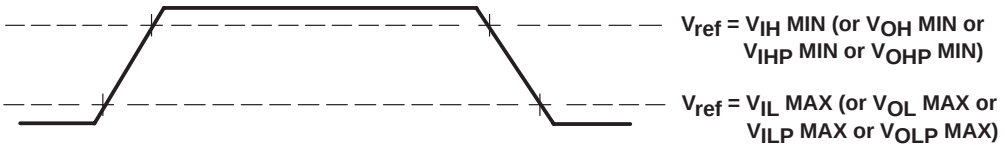



**Figure 8. Rise and Fall Transition Time Voltage Reference Levels**

## PARAMETER MEASUREMENT INFORMATION (CONTINUED)

### timing parameters and board routing analysis

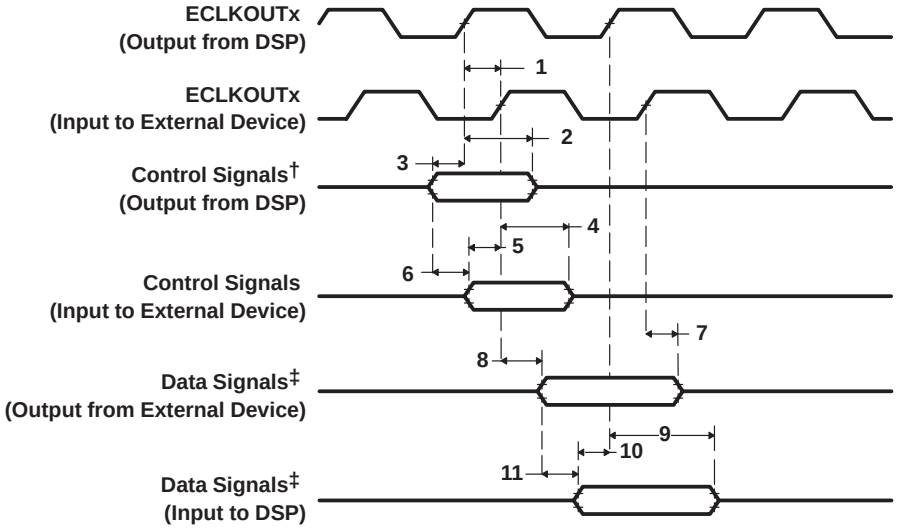
The timing parameter values specified in this data sheet do *not* include delays by board routings. As a good board design practice, such delays must *always* be taken into account. Timing values may be adjusted by increasing/decreasing such delays. TI recommends utilizing the available I/O buffer information 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 26 and Figure 9).

Figure 9 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 26. IBIS Timing Parameters Example (see Figure 9)**

| 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 9. IBIS Input/Output Timings**

## INPUT AND OUTPUT CLOCKS

### timing requirements for CLKIN†‡§ (see Figure 10)

| NO. | | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | PLL MODE x6 | | x1 (BYPASS) | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{c(CLKIN)}$ | Cycle time, CLKIN | | | | | ns |
| 2 | $t_{w(CLKINH)}$ | Pulse duration, CLKIN high | | | | | ns |
| 3 | $t_{w(CLKINL)}$ | Pulse duration, CLKIN low | | | | | ns |
| 4 | $t_{t(CLKIN)}$ | Transition time, CLKIN | | | | | ns |

† 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 factor (x6), 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.

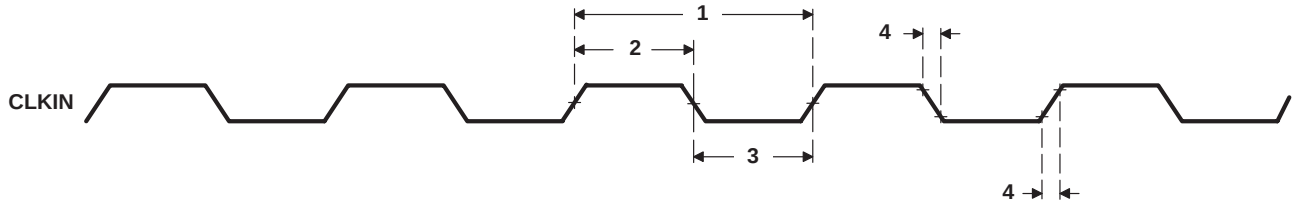



**Figure 10. CLKIN Timing**

PRODUCT PREVIEW

## INPUT AND OUTPUT CLOCKS (CONTINUED)

### switching characteristics over recommended operating conditions for CLKOUT4†‡§ (see Figure 11)

| NO. | PARAMETER | | –300 CLKMODE = x1, x6 MIN | –300 CLKMODE = x1, x6 MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{c(CKO4)}$ | Cycle time, CLKOUT4 | | | ns |
| 2 | $t_{w(CKO4H)}$ | Pulse duration, CLKOUT4 high | | | ns |
| 3 | $t_{w(CKO4L)}$ | Pulse duration, CLKOUT4 low | | | ns |
| 4 | $t_{t(CKO4)}$ | Transition time, CLKOUT4 | | | ns |

† 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)

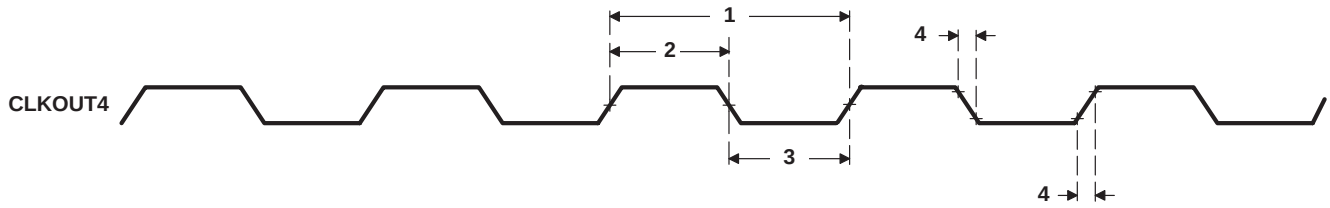



**Figure 11. CLKOUT4 Timing**

### switching characteristics over recommended operating conditions for CLKOUT6†‡§ (see Figure 12)

| NO. | PARAMETER | | –300 CLKMODE = x1, x6 MIN | –300 CLKMODE = x1, x6 MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{c(CKO6)}$ | Cycle time, CLKOUT6 | | | ns |
| 2 | $t_{w(CKO6H)}$ | Pulse duration, CLKOUT6 high | | | ns |
| 3 | $t_{w(CKO6L)}$ | Pulse duration, CLKOUT6 low | | | ns |
| 4 | $t_{t(CKO6)}$ | Transition time, CLKOUT6 | | | ns |

† 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)

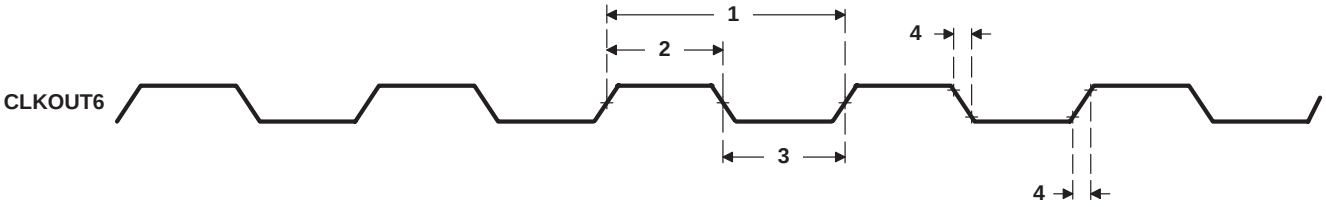



**Figure 12. CLKOUT6 Timing**

PRODUCT PREVIEW

## INPUT AND OUTPUT CLOCKS (CONTINUED)

### timing requirements for ECLKIN†‡ (see Figure 13)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(EKI)}$ | Cycle time, ECLKIN | | | ns |
| 2 | $t_{w(EKIH)}$ | Pulse duration, ECLKIN high | | | ns |
| 3 | $t_{w(EKIL)}$ | Pulse duration, ECLKIN low | | | ns |
| 4 | $t_{t(EKI)}$ | Transition time, ECLKIN | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.
‡ The reference points for the rise and fall transitions are measured at $V_{IL}$ MAX and $V_{IH}$ MIN.

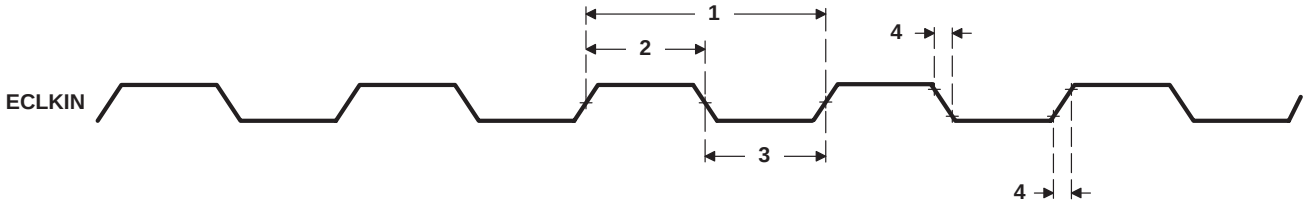



**Figure 13. ECLKIN Timing**

### switching characteristics over recommended operating conditions for ECLKOUT1§¶# (see Figure 14)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(EKO1)}$ | Cycle time, ECLKOUT1 | | | ns |
| 2 | $t_{w(EKO1H)}$ | Pulse duration, ECLKOUT1 high | | | ns |
| 3 | $t_{w(EKO1L)}$ | Pulse duration, ECLKOUT1 low | | | ns |
| 4 | $t_{t(EKO1)}$ | Transition time, ECLKOUT1 | | | ns |
| 5 | $t_{d(EKIH-EKO1H)}$ | Delay time, ECLKIN high to ECLKOUT1 high | | | ns |
| 6 | $t_{d(EKIL-EKO1L)}$ | Delay time, ECLKIN low to ECLKOUT1 low | | | ns |

§ 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.
# 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.

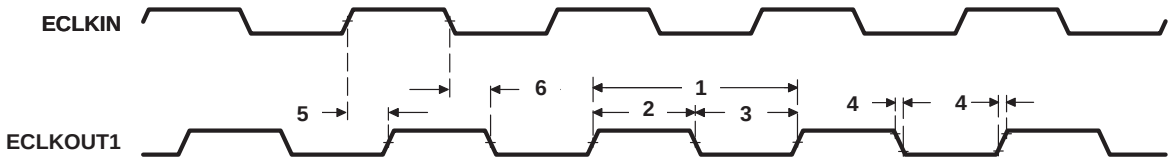



**Figure 14. ECLKOUT1 Timing**

## INPUT AND OUTPUT CLOCKS (CONTINUED)

### switching characteristics over recommended operating conditions for ECLKOUT2†‡ (see Figure 15)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(EKO2)}$ | Cycle time, ECLKOUT2 | | | ns |
| 2 | $t_{w(EKO2H)}$ | Pulse duration, ECLKOUT2 high | | | ns |
| 3 | $t_{w(EKO2L)}$ | Pulse duration, ECLKOUT2 low | | | ns |
| 4 | $t_{t(EKO2)}$ | Transition time, ECLKOUT2 | | | ns |
| 5 | $t_{d(EKIH-EKO2H)}$ | Delay time, ECLKIN high to ECLKOUT2 high | | | ns |
| 6 | $t_{d(EKIH-EKO2L)}$ | Delay time, ECLKIN high to ECLKOUT2 low | | | ns |

† 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.
N = the EMIF input clock divider; N = 1, 2, or 4.

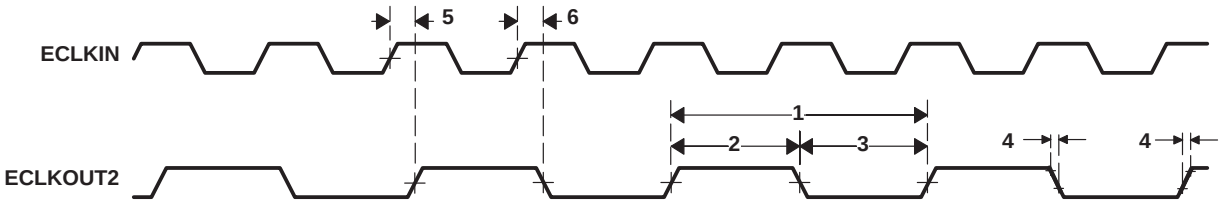



**Figure 15. ECLKOUT2 Timing**

## ASYNCHRONOUS MEMORY TIMING

### timing requirements for asynchronous memory cycles†‡ (see Figure 16 and Figure 17)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{su(EDV-AREH)}$ | Setup time, EDx valid before $\overline{ARE}$ high | | | ns |
| 4 | $t_{h(AREH-EDV)}$ | Hold time, EDx valid after $\overline{ARE}$ high | | | ns |
| 6 | $t_{su(ARDY-EKO1H)}$ | Setup time, ARDY valid before ECLKOUT1 high | | | ns |
| 7 | $t_{h(EKO1H-ARDY)}$ | Hold time, ARDY valid after ECLKOUT1 high | | | ns |

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. The ARDY signal is recognized in the cycle for which the setup and hold time is met. 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.

### switching characteristics over recommended operating conditions for asynchronous memory cycles‡§¶ (see Figure 16 and Figure 17)

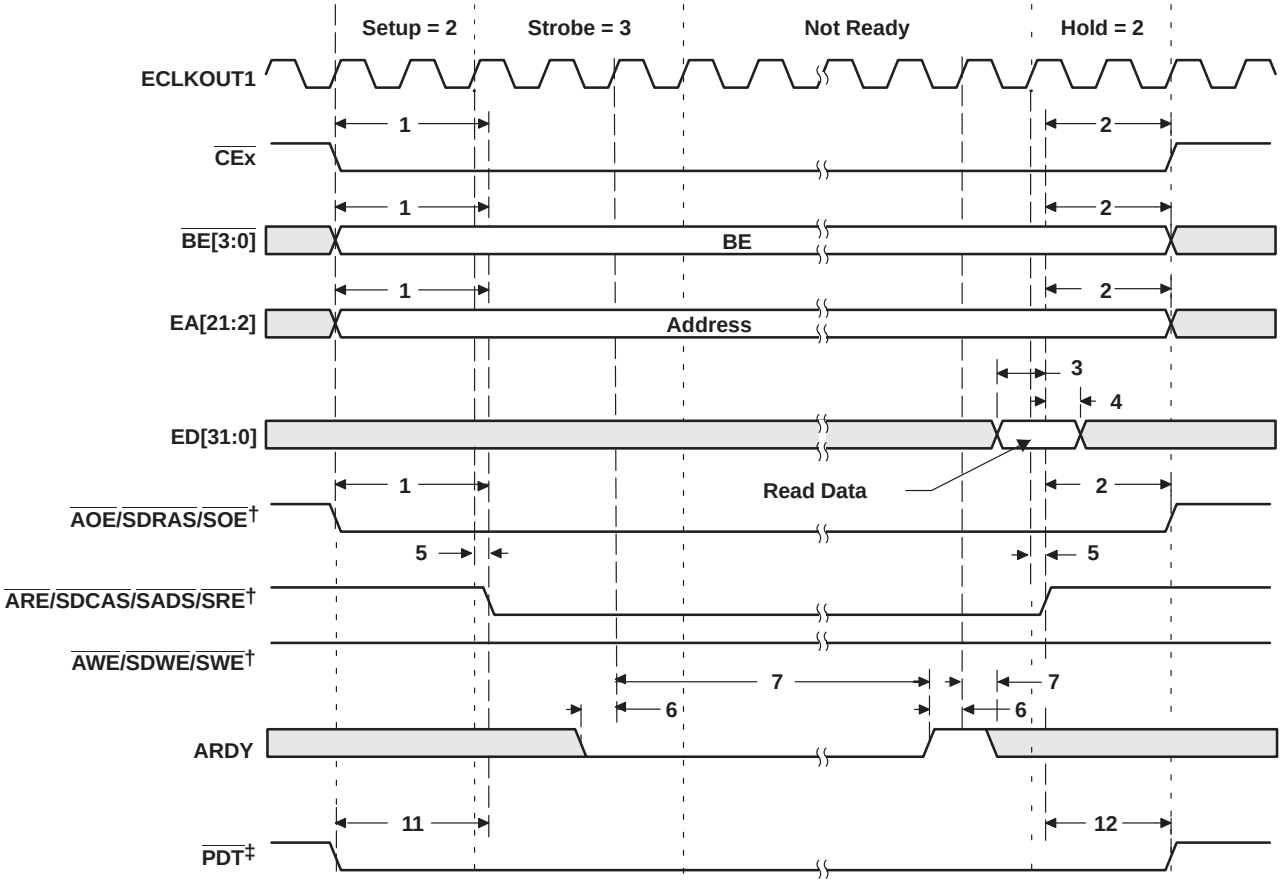
| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{osu(SELV-AREL)}$ | Output setup time, select signals valid to $\overline{ARE}$ low | | | ns |
| 2 | $t_{oh(AREH-SELIV)}$ | Output hold time, $\overline{ARE}$ high to select signals invalid | | | ns |
| 5 | $t_{d(EKO1H-AREV)}$ | Delay time, ECLKOUT1 high to $\overline{ARE}$ vaild | | | ns |
| 8 | $t_{osu(SELV-AWEL)}$ | Output setup time, select signals valid to $\overline{AWE}$ low | | | ns |
| 9 | $t_{oh(AWEH-SELIV)}$ | Output hold time, $\overline{AWE}$ high to select signals invalid | | | ns |
| 10 | $t_{d(EKO1H-AWEV)}$ | Delay time, ECLKOUT1 high to $\overline{AWE}$ vaild | | | ns |
| 11 | $t_{osu(PDTV-AREL)}$ | Output setup time, $\overline{PDT}$ valid to $\overline{ARE}$ low | | | ns |
| 12 | $t_{oh(AREH-PDTIV)}$ | Output hold time, $\overline{ARE}$ high to $\overline{PDT}$ invalid | | | ns |
| 13 | $t_{osu(PDTV-AWEV)}$ | Output setup time, $\overline{PDT}$ valid to $\overline{AWE}$ valid | | | ns |
| 14 | $t_{oh(AWEH-PDTIV)}$ | Output hold time, $\overline{AWE}$ high to $\overline{PDT}$ invalid | | | 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.

§ E = ECLKOUT1 period in ns

¶ Select signals for EMIF include: $\overline{CEx}$, $\overline{BE[3:0]}$, EA[21:2], $\overline{AOE}$; and for EMIF writes, include ED[31:0].

## ASYNCHRONOUS MEMORY TIMING (CONTINUED)

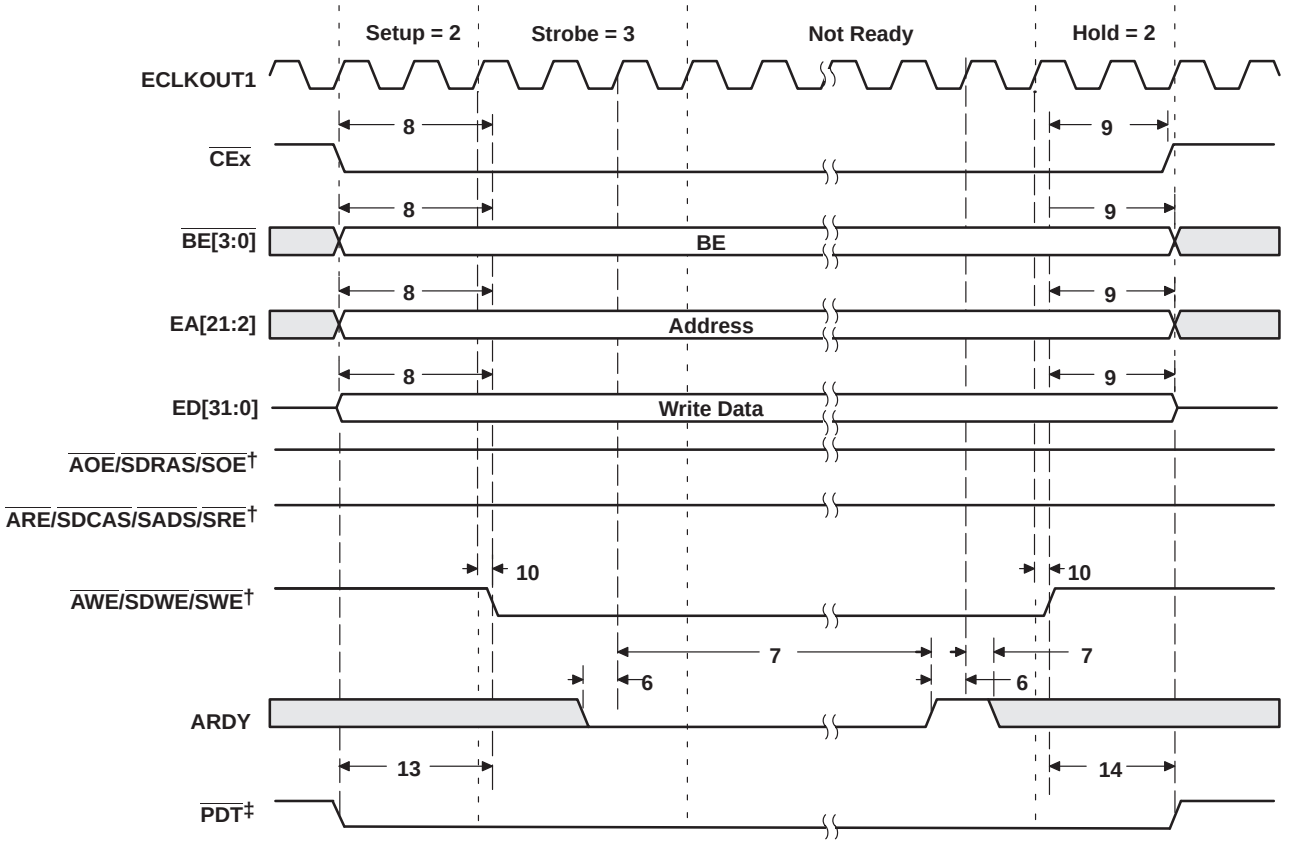



† AOE/SDRAS/SOE, ARE/SDCAS/SADS/SRE, and AWE/SDWE/SWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

‡ 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.

**Figure 16. Asynchronous Memory Read Timing**

## ASYNCHRONOUS MEMORY TIMING (CONTINUED)




† AOE/SDRAS/SOE, ARE/SDCAS/SADS/SRE, and AWE/SDWE/SWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory 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).

**Figure 17. Asynchronous Memory Write Timing**

## PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING

### timing requirements for programmable synchronous interface cycles (see Figure 18)

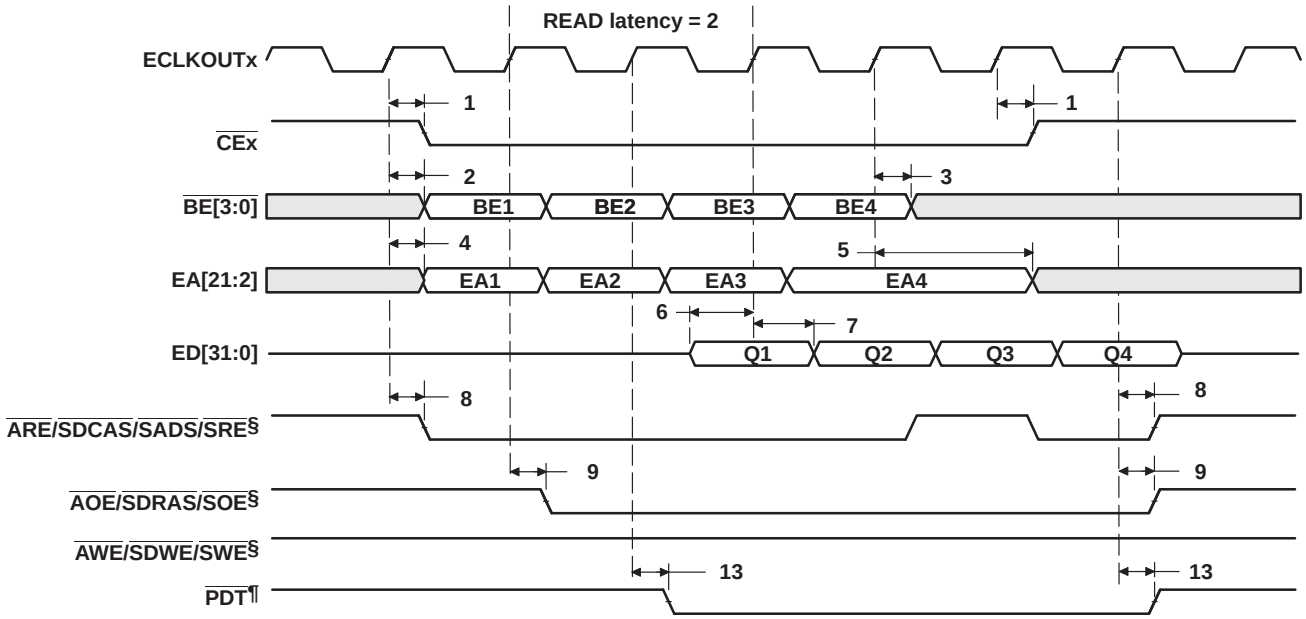
| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 6 | $t_{su(EDV-EKOxH)}$ | Setup time, read EDx valid before ECLKOUTx high | | | ns |
| 7 | $t_{h(EKOxH-EDV)}$ | Hold time, read EDx valid after ECLKOUTx high | | | ns |

### switching characteristics over recommended operating conditions for programmable synchronous interface cycles† (see Figure 18–Figure 20)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(EKOxH-CEV)}$ | Delay time, ECLKOUTx high to $\overline{CEx}$ valid | | | ns |
| 2 | $t_{d(EKOxH-BEV)}$ | Delay time, ECLKOUTx high to $\overline{BEx}$ valid | | | ns |
| 3 | $t_{d(EKOxH-BEIV)}$ | Delay time, ECLKOUTx high to $\overline{BEx}$ invalid | | | ns |
| 4 | $t_{d(EKOxH-EAV)}$ | Delay time, ECLKOUTx high to EAx valid | | | ns |
| 5 | $t_{d(EKOxH-EAIV)}$ | Delay time, ECLKOUTx high to EAx invalid | | | ns |
| 8 | $t_{d(EKOxH-ADSV)}$ | Delay time, ECLKOUTx high to $\overline{SADS}/\overline{SRE}$ valid | | | ns |
| 9 | $t_{d(EKOxH-OEV)}$ | Delay time, ECLKOUTx high to, $\overline{SOE}$ valid | | | ns |
| 10 | $t_{d(EKOxH-EDV)}$ | Delay time, ECLKOUTx high to $\overline{EDx}$ valid | | | ns |
| 11 | $t_{d(EKOxH-EDIV)}$ | Delay time, ECLKOUTx high to $\overline{EDx}$ invalid | | | ns |
| 12 | $t_{d(EKOxH-WEV)}$ | Delay time, ECLKOUTx high to $\overline{SWE}$ valid | | | ns |
| 13 | $t_{d(EKOxH-PDTV)}$ | Delay time, ECLKOUTx high to $\overline{PDT}$ valid | | | ns |

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

## PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)




† The read latency and the length of $\overline{CEx}$ assertion are programmable via the SYNCRL and CEEXT fields, respectively, in the EMIF 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):

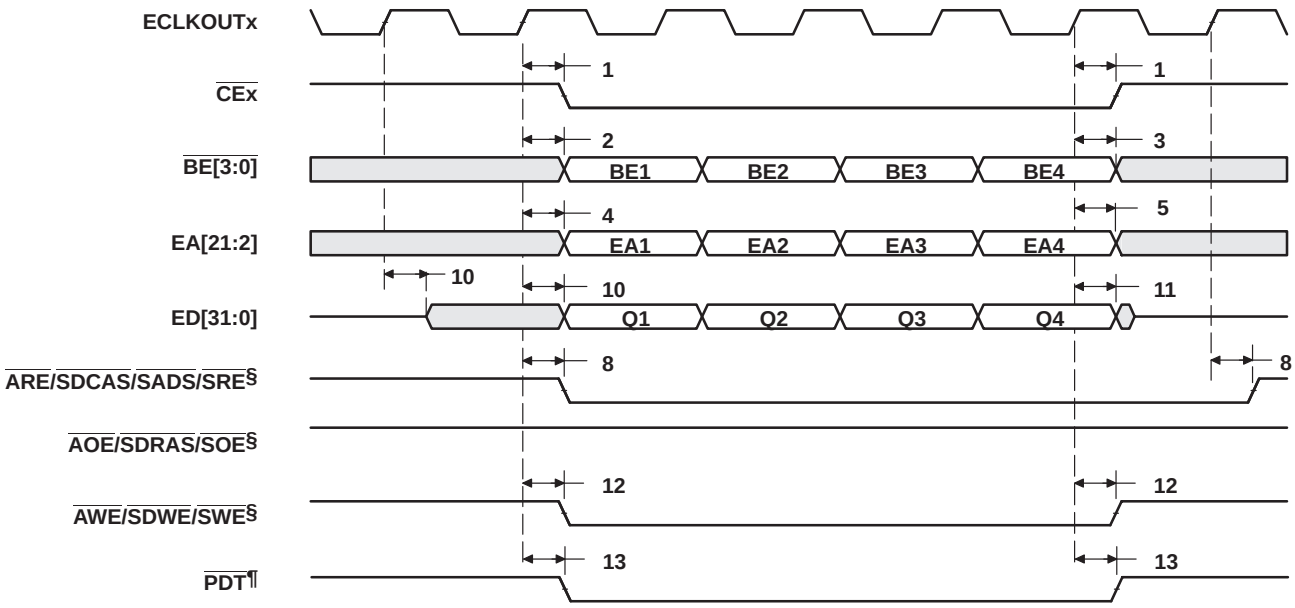
- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{CEx}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{CEx}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous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

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

¶ $\overline{PDT}$ signal is only asserted when the EDMA is in PDT mode (set the PDTS bit to 1 in the EDMA options parameter RAM). For $\overline{PDT}$ read, data is not latched into EMIF.

**Figure 18. Programmable Synchronous Interface Read Timing (With Read Latency = 2)†‡**

## PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)




† The write latency and the length of $\overline{CEx}$ assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIF 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):

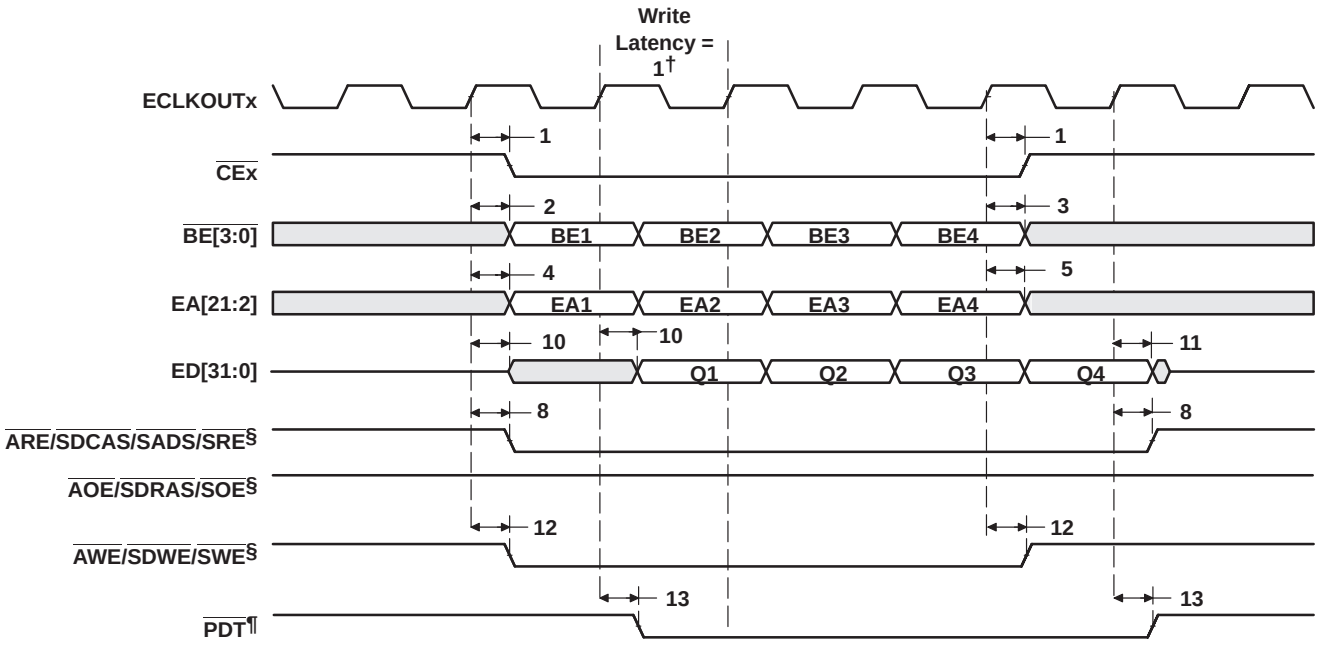
- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{CEx}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{CEx}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous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

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

¶ $\overline{PDT}$ signal is only asserted when the EDMA is in PDT mode (set the PDTD bit to 1 in the EDMA options parameter RAM). For $\overline{PDT}$ write, data is not driven (in High-Z).

**Figure 19. Programmable Synchronous Interface Write Timing (With Write Latency = 0)†‡**

## PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)




† The write latency and the length of $\overline{CEx}$ assertion are programmable via the SYNCWL and CEEXT fields, respectively, in the EMIF 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
- $\overline{CEx}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{CEx}$ goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface with glue, $\overline{CEx}$ is active when $\overline{SOE}$ is active (CEEXT = 1).
- Function of $\overline{SADS}/\overline{SRE}$ (RENEN): For standard SBSRAM or ZBT SRAM interface, $\overline{SADS}/\overline{SRE}$ acts as $\overline{SADS}$ with deselect cycles (RENEN = 0). For FIFO interface, $\overline{SADS}/\overline{SRE}$ acts as $\overline{SRE}$ with NO deselect cycles (RENEN = 1).
- Synchronization clock (SNCCLK): Synchronized to ECLKOUT1 or ECLKOUT2

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

¶ $\overline{PDT}$ signal is only asserted when the EDMA is in PDT mode (set the PDTD bit to 1 in the EDMA options parameter RAM). For $\overline{PDT}$ write, data is not driven (in High-Z).

**Figure 20. Programmable Synchronous Interface Write Timing (With Write Latency = 1)†‡**

## SYNCHRONOUS DRAM TIMING

### timing requirements for synchronous DRAM cycles (see Figure 21)

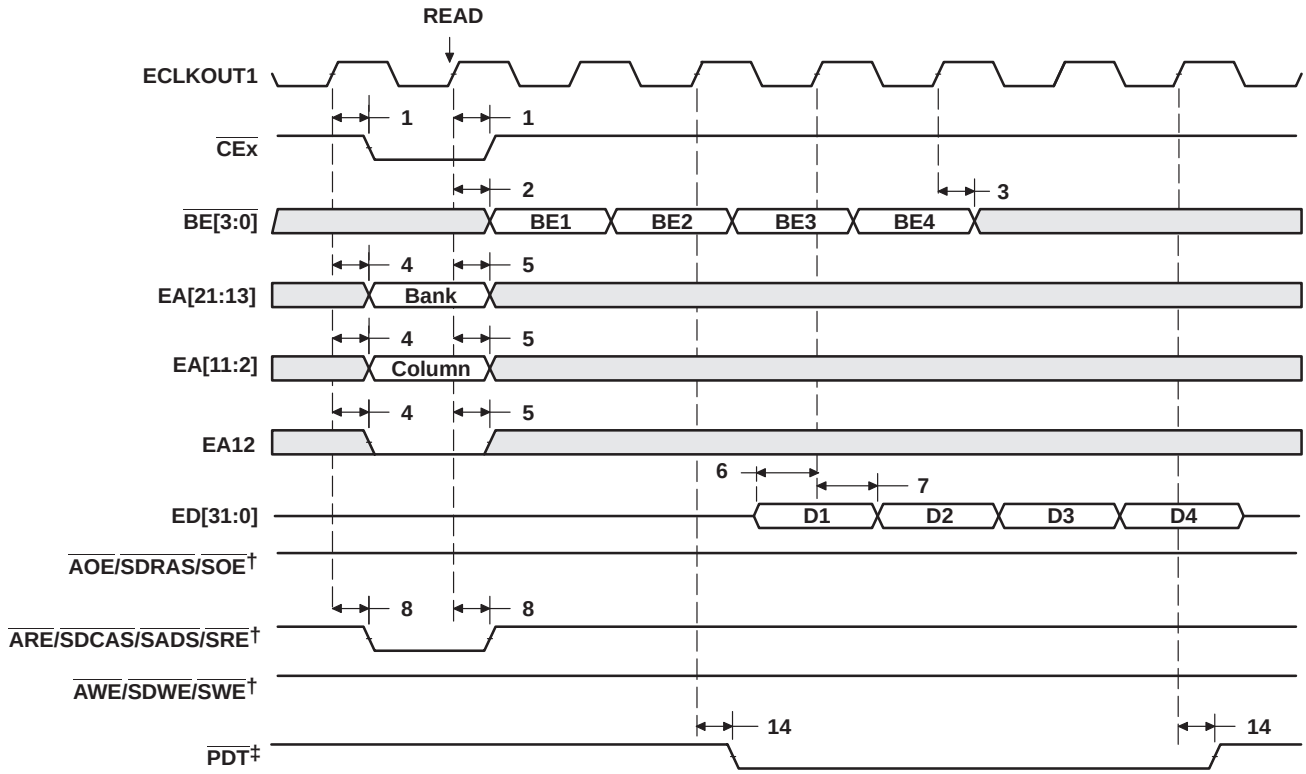
| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 6 | $t_{su(EDV-EKO1H)}$ | Setup time, read EDx valid before ECLKOUT1 high | | | ns |
| 7 | $t_{h(EKO1H-EDV)}$ | Hold time, read EDx valid after ECLKOUT1 high | | | ns |

### switching characteristics over recommended operating conditions for synchronous DRAM cycles (see Figure 21–Figure 28)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(EKO1H-CEV)}$ | Delay time, ECLKOUT1 high to $\overline{CEx}$ valid | | | ns |
| 2 | $t_{d(EKO1H-BEV)}$ | Delay time, ECLKOUT1 high to $\overline{BEx}$ valid | | | ns |
| 3 | $t_{d(EKO1H-BEIV)}$ | Delay time, ECLKOUT1 high to $\overline{BEx}$ invalid | | | ns |
| 4 | $t_{d(EKO1H-EAV)}$ | Delay time, ECLKOUT1 high to EAx valid | | | ns |
| 5 | $t_{d(EKO1H-EAIV)}$ | Delay time, ECLKOUT1 high to EAx invalid | | | ns |
| 8 | $t_{d(EKO1H-CASV)}$ | Delay time, ECLKOUT1 high to $\overline{SDCAS}$ valid | | | ns |
| 9 | $t_{d(EKO1H-EDV)}$ | Delay time, ECLKOUT1 high to $\overline{EDx}$ valid | | | ns |
| 10 | $t_{d(EKO1H-EDIV)}$ | Delay time, ECLKOUT1 high to $\overline{EDx}$ invalid | | | ns |
| 11 | $t_{d(EKO1H-WEV)}$ | Delay time, ECLKOUT1 high to $\overline{SDWE}$ valid | | | ns |
| 12 | $t_{d(EKO1H-RAS)}$ | Delay time, ECLKOUT1 high to $\overline{SDRAS}$ valid | | | ns |
| 13 | $t_{d(EKO1H-ACKEV)}$ | Delay time, ECLKOUT1 high to SDCKE valid | | | ns |
| 14 | $t_{d(EKO1H-PDTV)}$ | Delay time, ECLKOUT1 high to $\overline{PDT}$ valid | | | ns |

PRODUCT PREVIEW

## SYNCHRONOUS DRAM TIMING (CONTINUED)



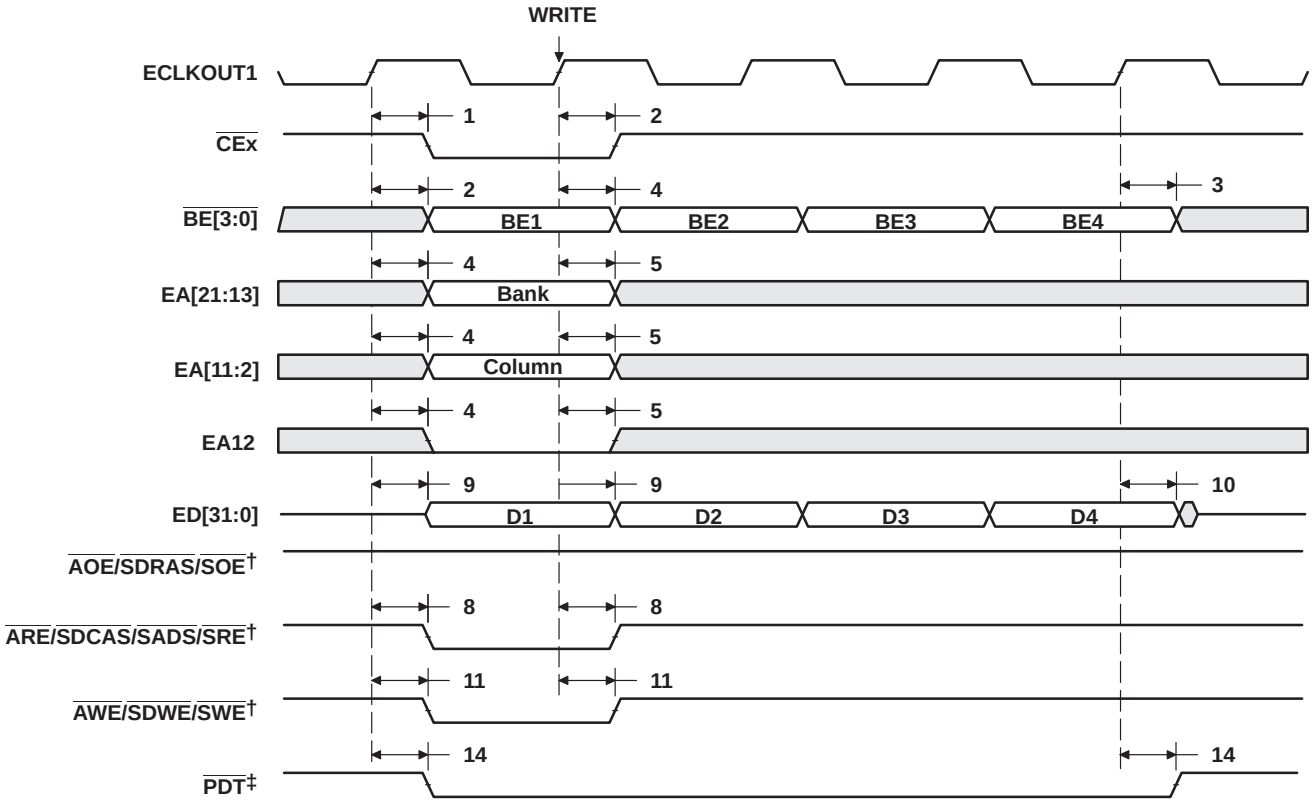

† $\overline{\text{ARE}}/\overline{\text{SDCAS}}/\overline{\text{SADS}}/\overline{\text{SRE}}$, $\overline{\text{AWE}}/\overline{\text{SDWE}}/\overline{\text{SWE}}$, and $\overline{\text{AOE}}/\overline{\text{SDRAS}}/\overline{\text{SOE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$,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

‡ $\overline{\text{PDT}}$ signal is only asserted when the EDMA is in PDT mode (set the PDTS bit to 1 in the EDMA options parameter RAM). For $\overline{\text{PDT}}$ read, data is not latched into EMIF.

**Figure 21. SDRAM Read Command (CAS Latency 3)**

PRODUCT PREVIEW

## SYNCHRONOUS DRAM TIMING (CONTINUED)



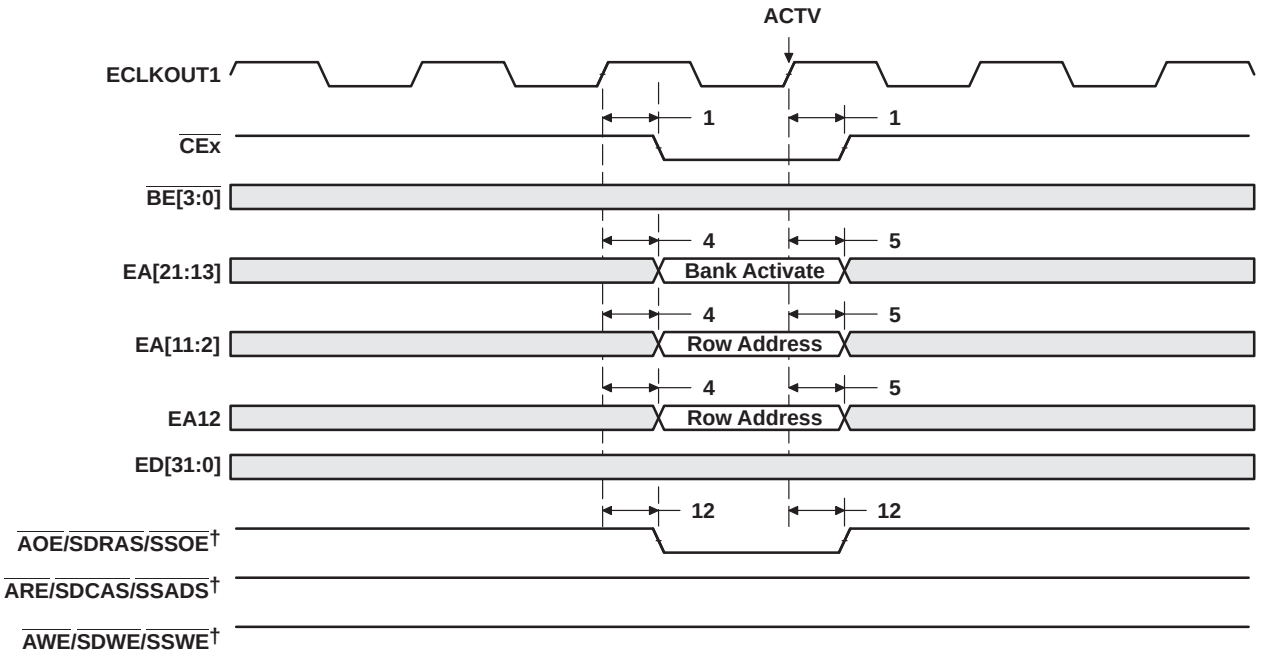

† ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

‡ 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).

**Figure 22. SDRAM Write Command**

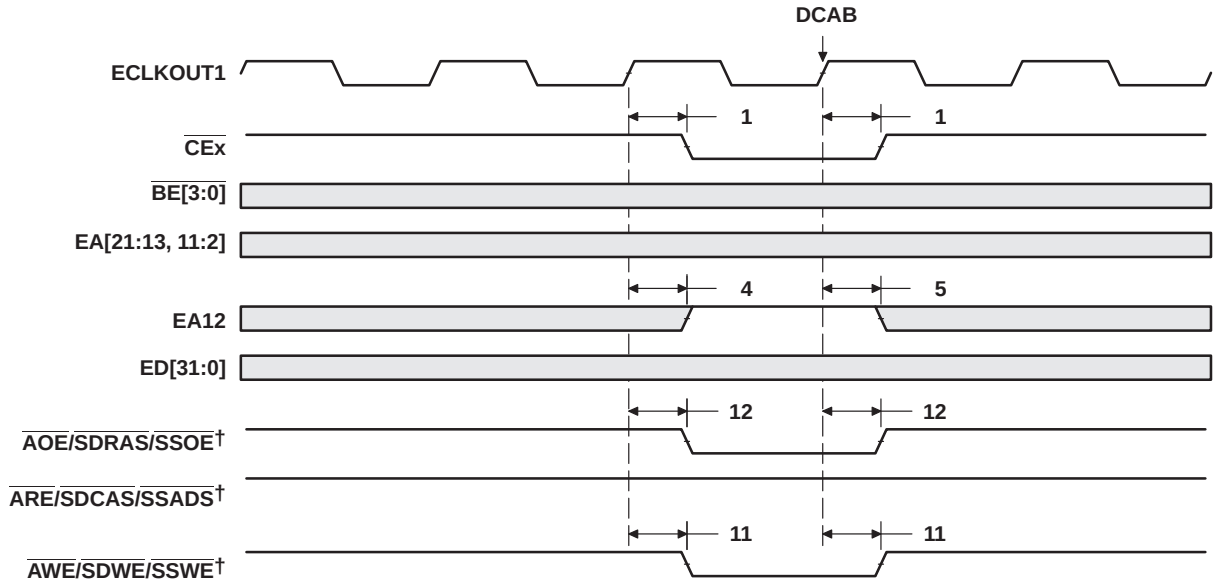
PRODUCT PREVIEW

## SYNCHRONOUS DRAM TIMING (CONTINUED)




† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

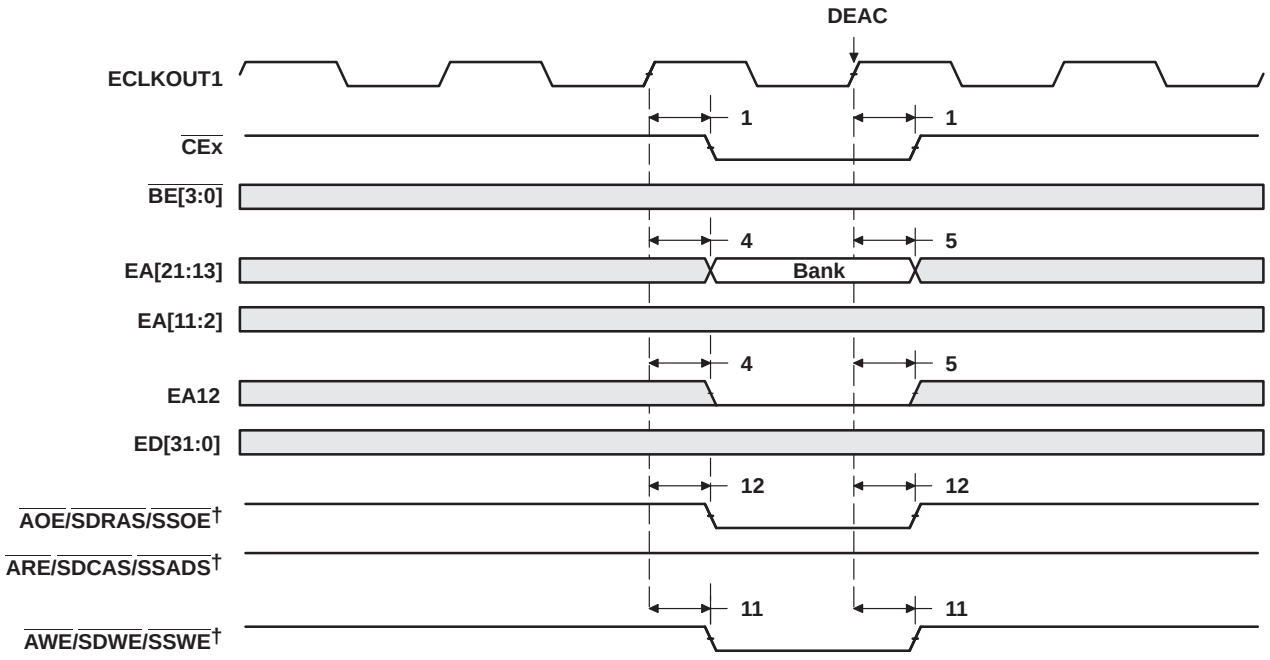
**Figure 23. SDRAM ACTV Command**




† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

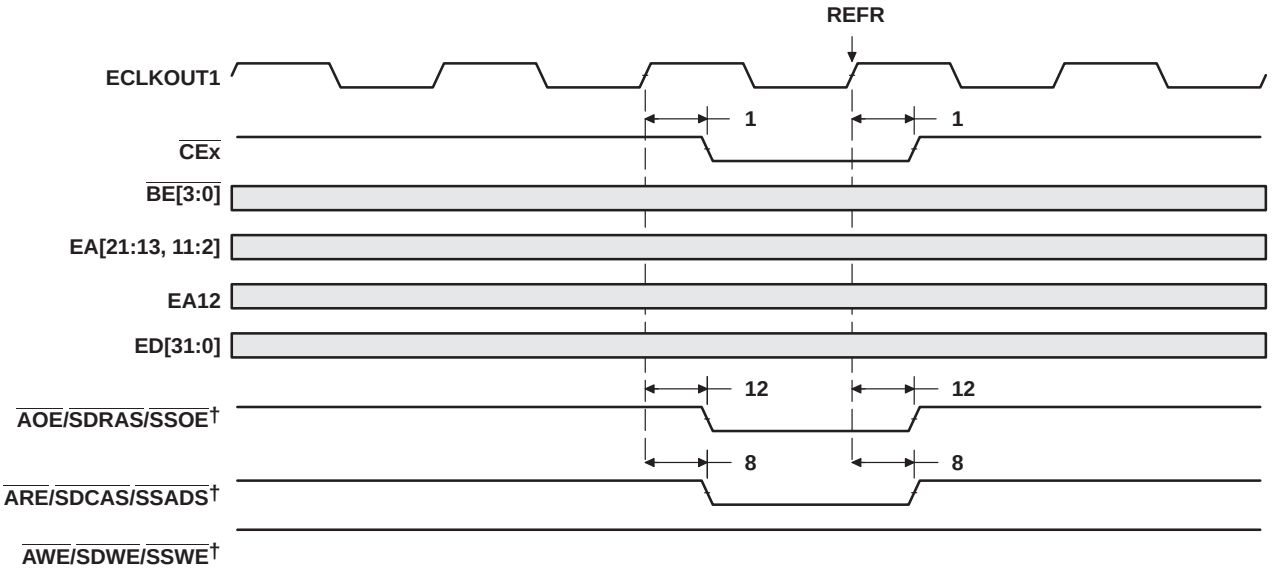
**Figure 24. SDRAM DCAB Command**

## SYNCHRONOUS DRAM TIMING (CONTINUED)




† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

**Figure 25. SDRAM DEAC Command**

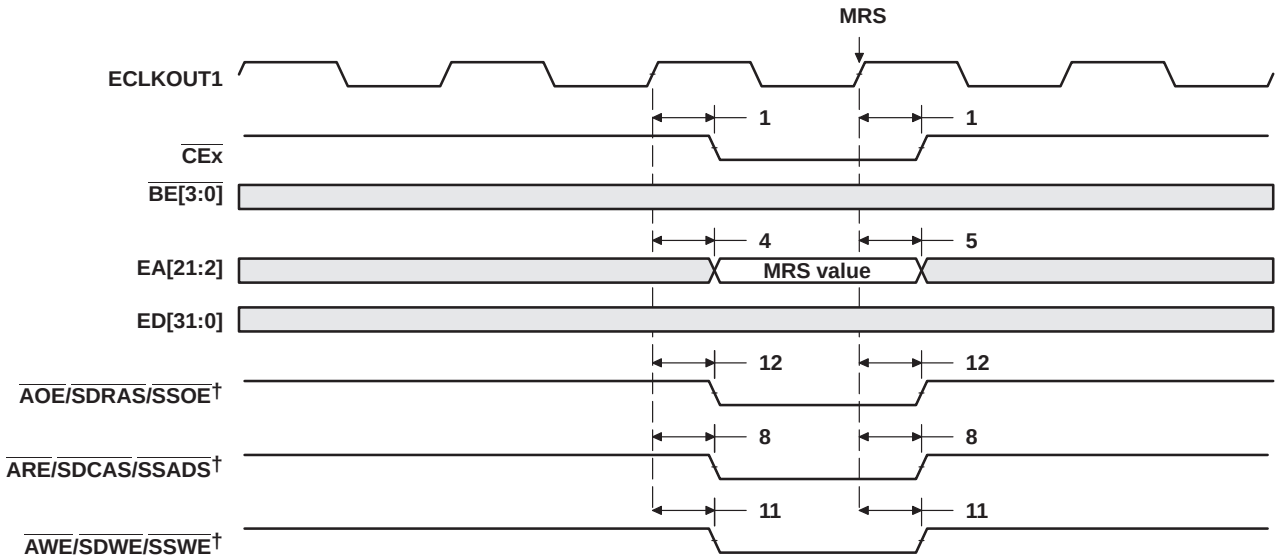



† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

**Figure 26. SDRAM REFR Command**

PRODUCT PREVIEW

## SYNCHRONOUS DRAM TIMING (CONTINUED)

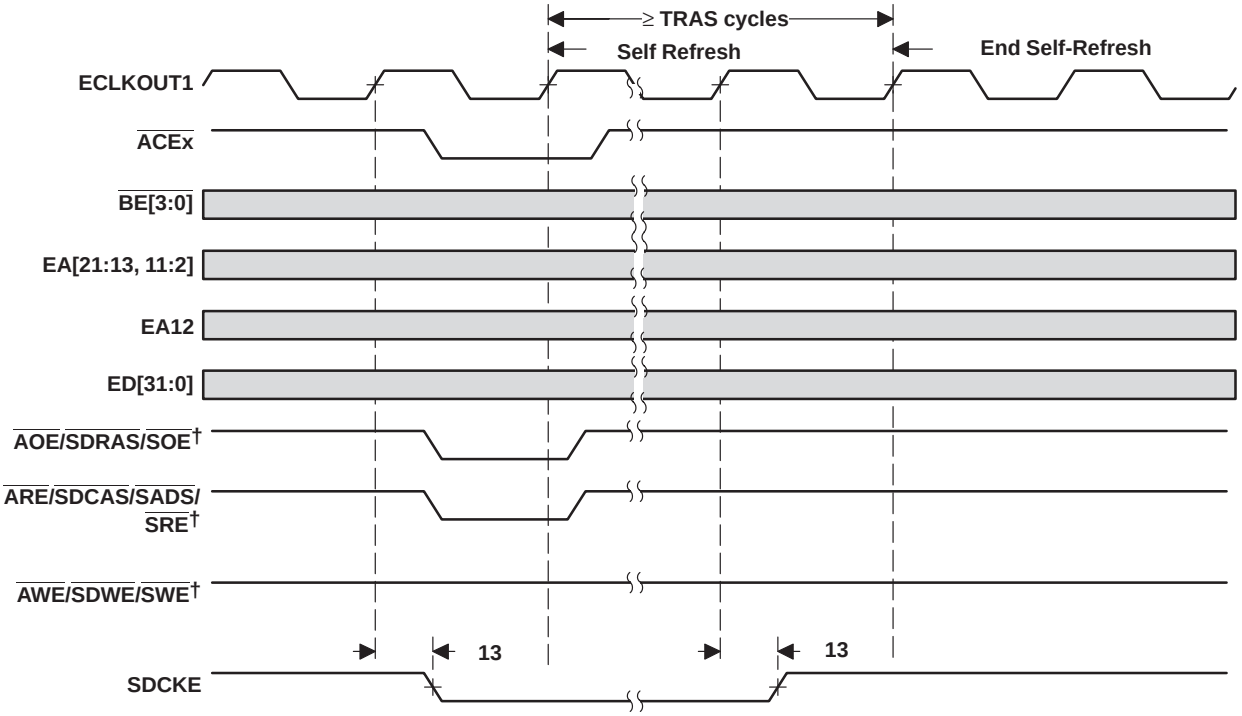



† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

**Figure 27. SDRAM MRS Command**

PRODUCT PREVIEW

## SYNCHRONOUS DRAM TIMING (CONTINUED)




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

**Figure 28. SDRAM Self-Refresh Timing**

## HOLD/HOLDA TIMING

### timing requirements for the HOLD/HOLDA cycles† (see Figure 29)

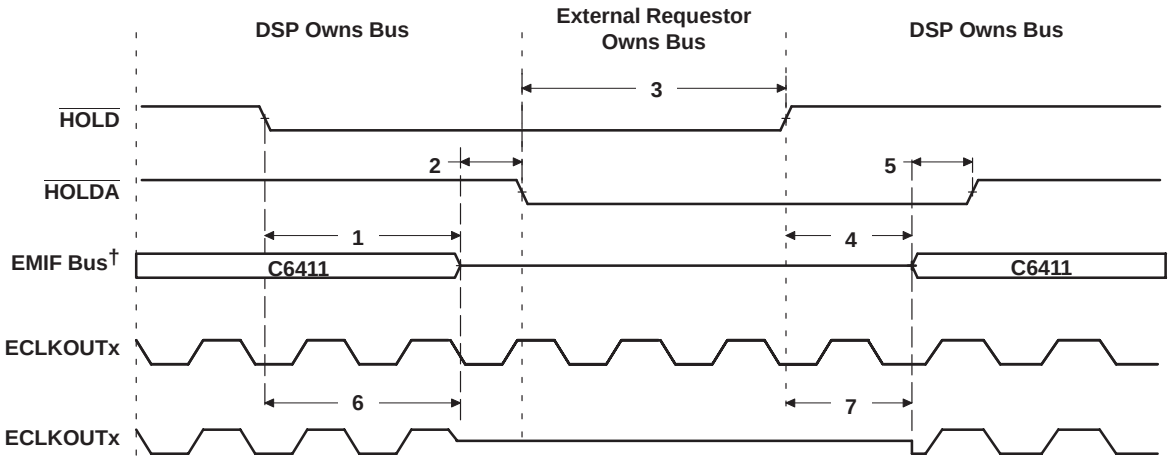
| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{oh(HOLDAL-HOLDL)}$ | Hold time, HOLD low after HOLDA low | | | ns |

† E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns.

### switching characteristics over recommended operating conditions for the HOLD/HOLDA cycles†‡§ (see Figure 29)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(HOLDL-EMHZ)}$ | Delay time, HOLD low to EMIF Bus high impedance | | ¶ | ns |
| 2 | $t_{d(EMHZ-HOLDAL)}$ | Delay time, EMIF Bus high impedance to HOLDA low | | | ns |
| 4 | $t_{d(HOLDH-EMLZ)}$ | Delay time, HOLD high to EMIF Bus low impedance | | | ns |
| 5 | $t_{d(EMLZ-HOLDAH)}$ | Delay time, EMIF Bus low impedance to HOLDA high | | | ns |
| 6 | $t_{d(HOLDL-EKOHZ)}$ | Delay time, HOLD low to ECLKOUTx high impedance | | ¶ | ns |
| 7 | $t_{d(HOLDH-EKOLZ)}$ | Delay time, HOLD high to ECLKOUTx low impedance | | | ns |

† E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns.
‡ EMIF Bus consists of: CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE/SDCAS/SADS/SRE, AOE/SDRAS/SOE, and AWE/SDWE/SWE , SDCKE, SOE3, and PDT.
§ 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 29.
¶ 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.




† EMIF Bus consists of: CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE/SDCAS/SADS/SRE, AOE/SDRAS/SOE, and AWE/SDWE/SWE, SDCKE, SOE3, and PDT.

**Figure 29. HOLD/HOLDA Timing**

## BUSREQ TIMING

**switching characteristics over recommended operating conditions for the BUSREQ cycles (see Figure 30)**

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(AEKO1H-ABUSRV)}$ | Delay time, AECLKOUT1 high to ABUSREQ valid | | | ns |

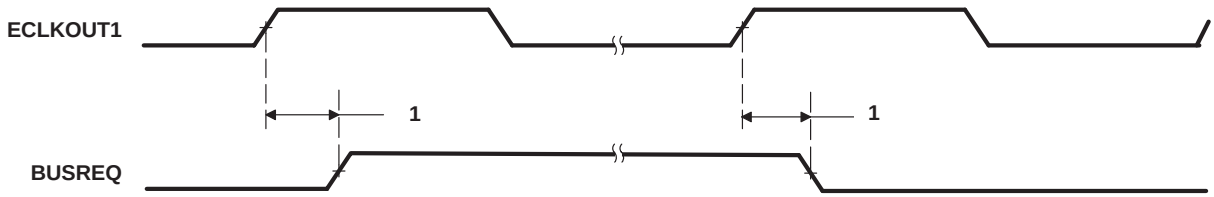



**Figure 30. BUSREQ Timing**

PRODUCT PREVIEW

## RESET TIMING

### timing requirements for reset† (see Figure 31)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(RST)}$ | Width of the RESET pulse (PLL stable)‡ | | | ns |
| | | Width of the RESET pulse (PLL needs to sync up)§ | | | µs |
| 16 | $t_{su(boot)}$ | Setup time, boot configuration bits valid before RESET high¶ | | | ns |
| 17 | $t_{h(boot)}$ | Hold time, boot configuration bits valid after RESET high¶ | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.
‡ This parameter applies to CLKMODE x1 when CLKIN is stable, and applies to CLKMODE x6 when CLKIN and PLL are stable.
§ This parameter applies to CLKMODE x6 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.
¶ LEND, BOOTMODE[1:0], ECLKIN_SEL[1:0], EEAI, and HD5/AD5 are the boot configuration pins during device reset.

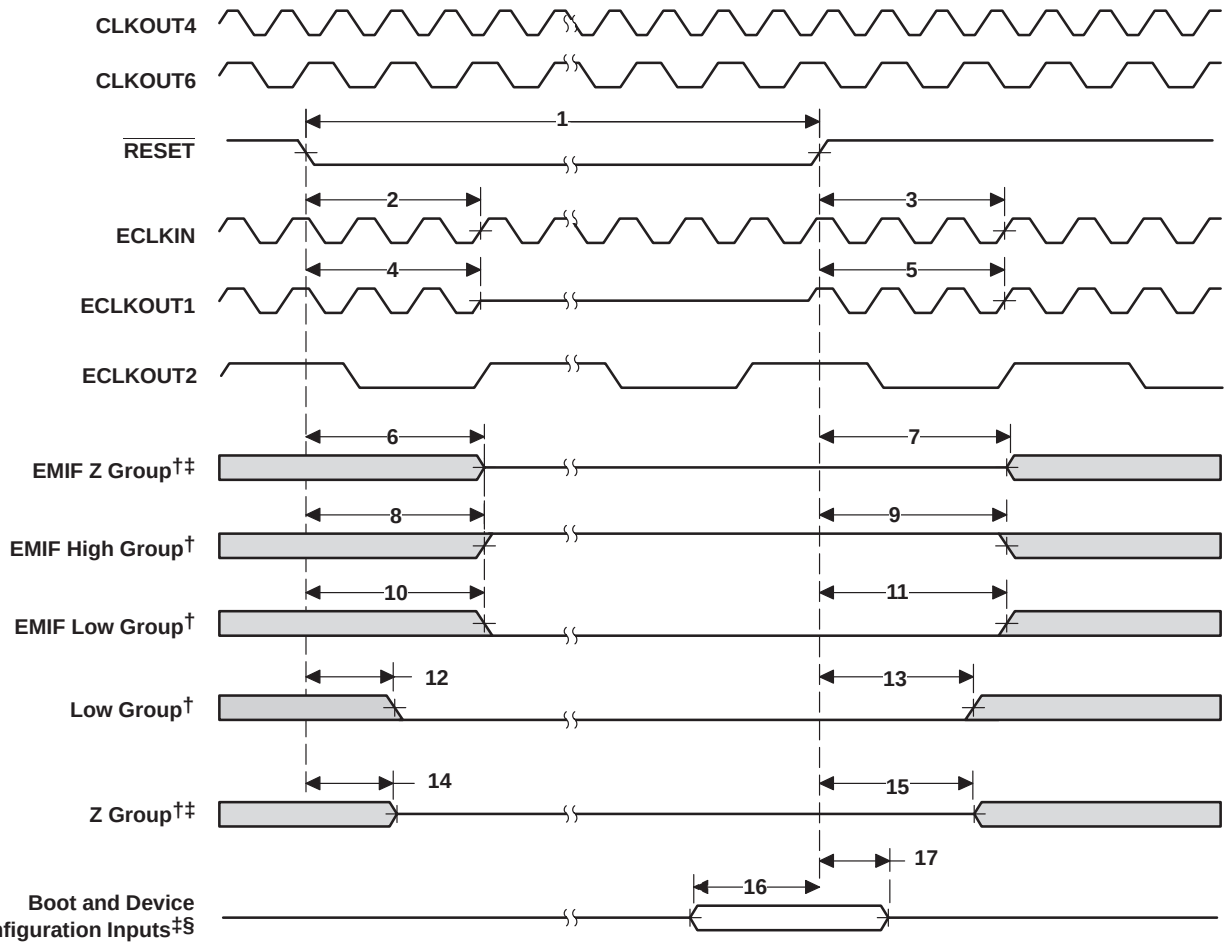
### switching characteristics over recommended operating conditions during reset†#‖ (see Figure 31)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 2 | $t_{d(RSTL-ECKI)}$ | Delay time, RESET low to ECLKIN synchronized internally | | | ns |
| 3 | $t_{d(RSTH-ECKI)}$ | Delay time, RESET high to ECLKIN synchronized internally | | | ns |
| 4 | $t_{d(RSTL-ECKO1HZ)}$ | Delay time, RESET low to ECLKOUT1 high impedance | | | ns |
| 5 | $t_{d(RSTH-ECKO1V)}$ | Delay time, RESET high to ECLKOUT1 valid | | | ns |
| 6 | $t_{d(RSTL-EMIFZHZ)}$ | Delay time, RESET low to EMIF Z high impedance | | | ns |
| 7 | $t_{d(RSTH-EMIFZV)}$ | Delay time, RESET high to EMIF Z valid | | | ns |
| 8 | $t_{d(RSTL-EMIFHIV)}$ | Delay time, RESET low to EMIF high group invalid | | | ns |
| 9 | $t_{d(RSTH-EMIFHV)}$ | Delay time, RESET high to EMIF high group valid | | | ns |
| 10 | $t_{d(RSTL-EMIFLIV)}$ | Delay time, RESET low to EMIF low group invalid | | | ns |
| 11 | $t_{d(RSTH-EMIFLV)}$ | Delay time, RESET high to EMIF low group valid | | | ns |
| 12 | $t_{d(RSTL-LOWIV)}$ | Delay time, RESET low to low group invalid | | | ns |
| 13 | $t_{d(RSTH-LOWV)}$ | Delay time, RESET high to low group valid | | | ns |
| 14 | $t_{d(RSTL-ZHZ)}$ | Delay time, RESET low to Z group high impedance | | | ns |
| 15 | $t_{d(RSTH-ZV)}$ | Delay time, RESET high to Z group valid | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.
# E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns.
‖ EMIF Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE, SOE3, SDCKE, and PDT.
EMIF high group consists of: HOLDA (when the corresponding HOLD input is high)
EMIF low group consists of: BUSREQ; HOLDA (when the corresponding HOLD input is low)
Low group consists of: XSP_CS, XSP_CLK, and XSP_DO; all of which apply only when PCI EEPROM (EEAI) is enabled (with PCI_EN = 1). Otherwise, the XSP_CLK and 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, XSP_CLK, FSX0, FSX1, DX0, DX1, XSP_DO, CLKR0, CLKR1, FSR0, FSR1, TOUT0, TOUT1, 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, and HINT/PFRAME.

PRODUCT PREVIEW

## RESET TIMING (CONTINUED)




† EMIF Z group consists of: EA[21:2], ED[31:0], $\overline{CE}$[3:0], $\overline{BE}$[3:0], $\overline{ARE}$/$\overline{SDCAS}$/$\overline{SADS}$/$\overline{SRE}$, $\overline{AWE}$/$\overline{SDWE}$/$\overline{SWE}$, and $\overline{AOE}$/$\overline{SDRAS}$/$\overline{SOE}$, $\overline{SOE3}$, SDCKE, and $\overline{PDT}$.

EMIF high group consists of: $\overline{HOLDA}$ (when the corresponding $\overline{HOLD}$ input is high)

EMIF low group consists of: BUSREQ; $\overline{HOLDA}$ (when the corresponding $\overline{HOLD}$ input is low)

Low group consists of: XSP_CS, XSP_CLK, and XSP_DO; all of which apply only when PCI EEPROM (EEAI) is enabled (with PCI_EN = 1). Otherwise, the XSP_CLK and 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, XSP_CLK, FSX0, FSX1, DX0, DX1, XSP_DO, CLKR0, CLKR1, FSR0, FSR1, TOUT0, TOUT1, GP[8:0], GP10/$\overline{PCBE3}$, HR/$\overline{W}$/$\overline{PCBE2}$, $\overline{HDS2}$/$\overline{PCBE1}$, PCBE0, GP13/$\overline{PINTA}$, GP11/$\overline{PREQ}$, $\overline{HDS1}$/$\overline{PSERR}$, $\overline{HCS}$/$\overline{PPERR}$, HCNTL1/$\overline{PDEVSEL}$, $\overline{HAS}$/$\overline{PPAR}$, HCNTL0/$\overline{PSTOP}$, HHWIL/$\overline{PTRDY}$ (16-bit HPI mode only), $\overline{HRDY}$/$\overline{PIRDY}$, and $\overline{HINT}$/$\overline{PFRAME}$.

‡ If LEND, BOOTMODE[1:0], ECLKIN_SEL[1:0], EEAI, 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: LEND, BOOTMODE[1:0], ECLKIN_SEL[1:0], EEAI, and HD5/AD5. The PCI_EN pin *must* be driven valid at all times and the user *must not* switch values throughout device operation.

**Figure 31. Reset Timing†**

PRODUCT PREVIEW

## EXTERNAL INTERRUPT TIMING

### timing requirements for external interrupts† (see Figure 32)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(ILOW)}$ | Width of the interrupt pulse low | | | ns |
| 2 | $t_{w(IHIGH)}$ | Width of the interrupt pulse high | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

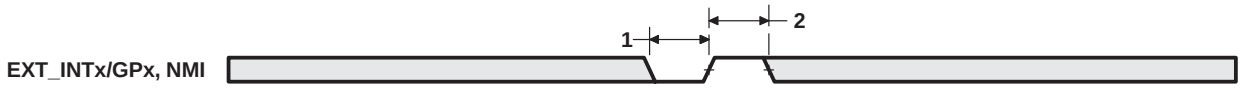



**Figure 32. External/NMI Interrupt Timing**

PRODUCT PREVIEW

## HOST-PORT INTERFACE (HPI) TIMING

### timing requirements for host-port interface cycles†‡ (see Figure 33 through Figure 40)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(SELV-HSTBL)}$ | Setup time, select signals§ valid before $\overline{\text{HSTROBE}}$ low | | | ns |
| 2 | $t_{h(HSTBL-SELV)}$ | Hold time, select signals§ valid after $\overline{\text{HSTROBE}}$ low | | | ns |
| 3 | $t_{w(HSTBL)}$ | Pulse duration, $\overline{\text{HSTROBE}}$ low | ¶ | | ns |
| 4 | $t_{w(HSTBH)}$ | Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses | | | ns |
| 10 | $t_{su(SELV-HASL)}$ | Setup time, select signals§ valid before $\overline{\text{HAS}}$ low | | | ns |
| 11 | $t_{h(HASL-SELV)}$ | Hold time, select signals§ valid after $\overline{\text{HAS}}$ low | | | ns |
| 12 | $t_{su(HDV-HSTBH)}$ | Setup time, host data valid before $\overline{\text{HSTROBE}}$ high | | | ns |
| 13 | $t_{h(HSTBH-HDV)}$ | Hold time, host data valid after $\overline{\text{HSTROBE}}$ high | | | 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. | | | ns |
| 18 | $t_{su(HASL-HSTBL)}$ | Setup time, $\overline{\text{HAS}}$ low before $\overline{\text{HSTROBE}}$ low | | | ns |
| 19 | $t_{h(HSTBL-HASL)}$ | Hold time, $\overline{\text{HAS}}$ low after $\overline{\text{HSTROBE}}$ low | | | ns |

† $\overline{\text{HSTROBE}}$ 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 300 MHz, use P = 3.33 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.

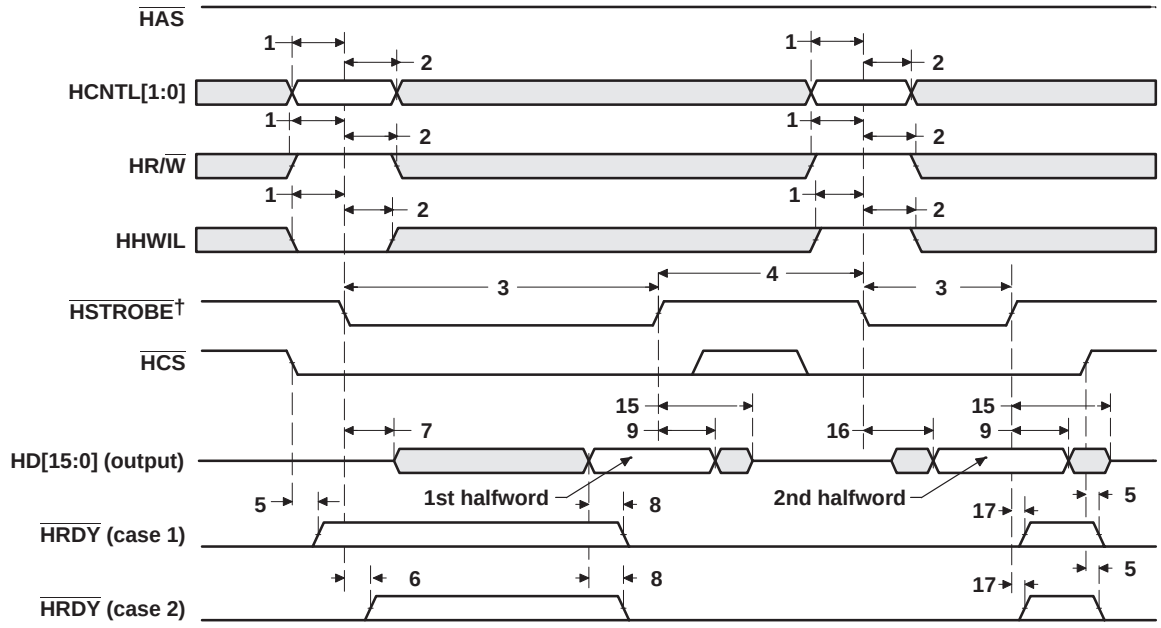
### switching characteristics over recommended operating conditions during host-port interface cycles†‡ (see Figure 33 through Figure 40)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{d(HCS-HRDY)}$ | Delay time, $\overline{\text{HCS}}$ to $\overline{\text{HRDY}}$# | | | ns |
| 6 | $t_{d(HSTBL-HRDYH)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high‖ | | | ns |
| 7 | $t_{d(HSTBL-HDLZ)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to HD low impedance for an HPI read | | | ns |
| 8 | $t_{d(HDV-HRDYL)}$ | Delay time, HD valid to $\overline{\text{HRDY}}$ low | | | ns |
| 9 | $t_{oh(HSTBH-HDV)}$ | Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high | | | ns |
| 15 | $t_{d(HSTBH-HDHZ)}$ | Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance | | | ns |
| 16 | $t_{d(HSTBL-HDV)}$ | Delay time, $\overline{\text{HSTROBE}}$ low to HD valid (HPI16 only) | | | ns |
| 17 | $t_{d(HSTBH-HRDYH)}$ | Delay time, $\overline{\text{HSTROBE}}$ high to $\overline{\text{HRDY}}$ high☆ | | | ns |

† $\overline{\text{HSTROBE}}$ 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 300 MHz, use P = 3.33 ns.
# $\overline{\text{HCS}}$ enables $\overline{\text{HRDY}}$, and $\overline{\text{HRDY}}$ is always low when $\overline{\text{HCS}}$ is high. The case where $\overline{\text{HRDY}}$ goes high when $\overline{\text{HCS}}$ falls indicates that HPI is busy completing a previous HPID write or READ with autoincrement.
‖ 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.
☆ This parameter is used after a word (HPI32) or the second half-word (HPI16) of an HPID write or autoincrement read. $\overline{\text{HRDY}}$ remains low if the access is not an HPID write or autoincrement read. Reading or writing to HPIC or HPIA does not affect the $\overline{\text{HRDY}}$ signal.

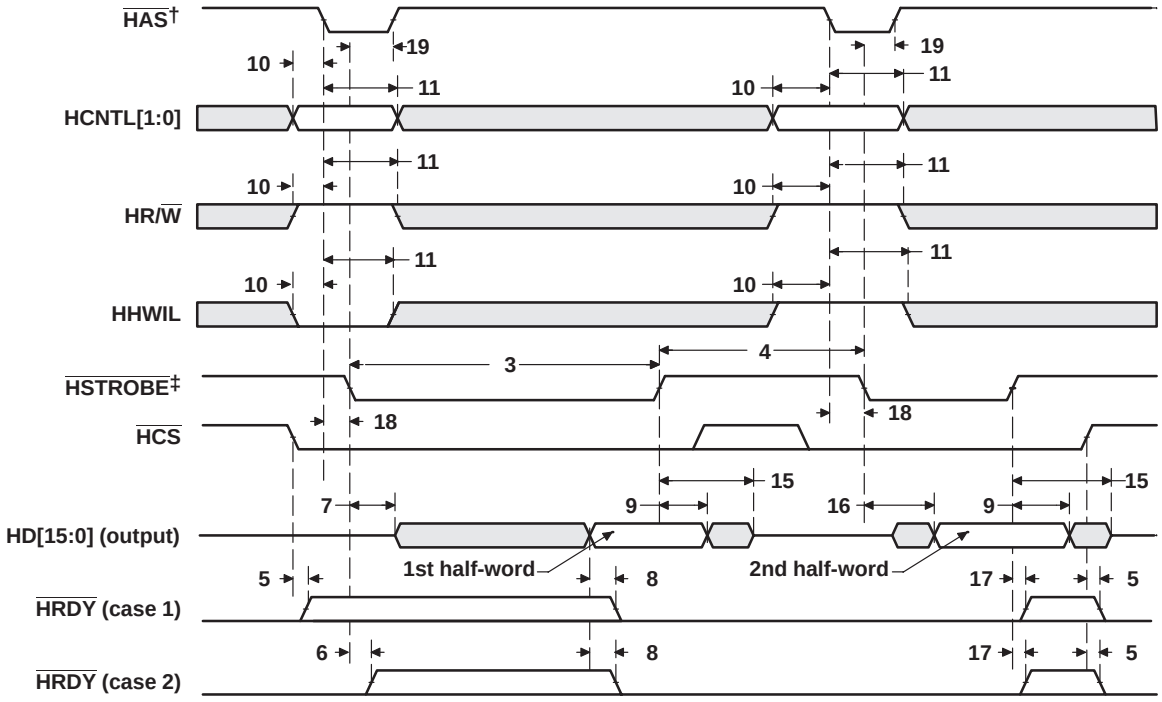
PRODUCT PREVIEW

## HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)




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

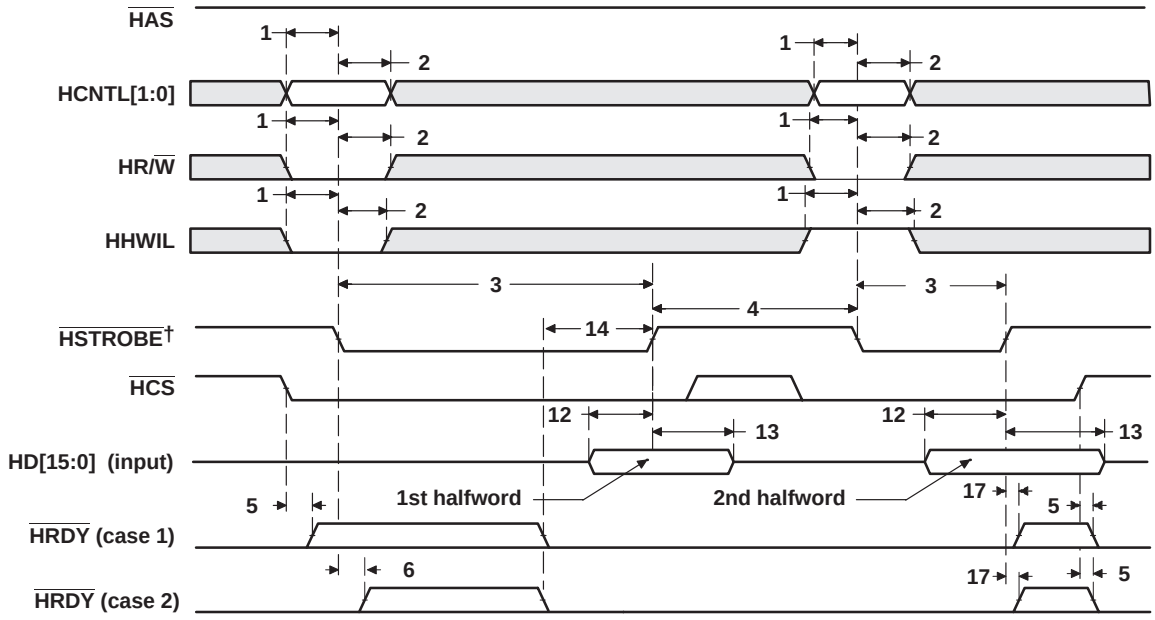
**Figure 33. HPI16 Read Timing (HAS Not Used, Tied High)**




† 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.

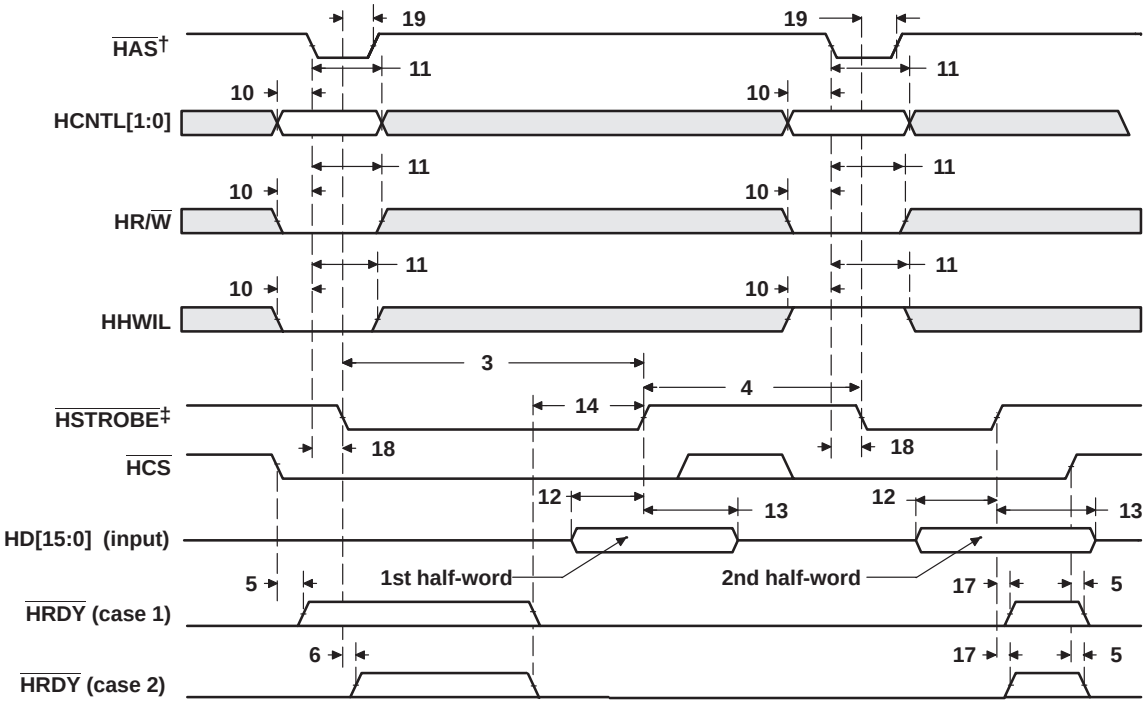
**Figure 34. HPI16 Read Timing (HAS Used)**

## 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 35. HPI16 Write Timing (HAS Not Used, Tied High)**

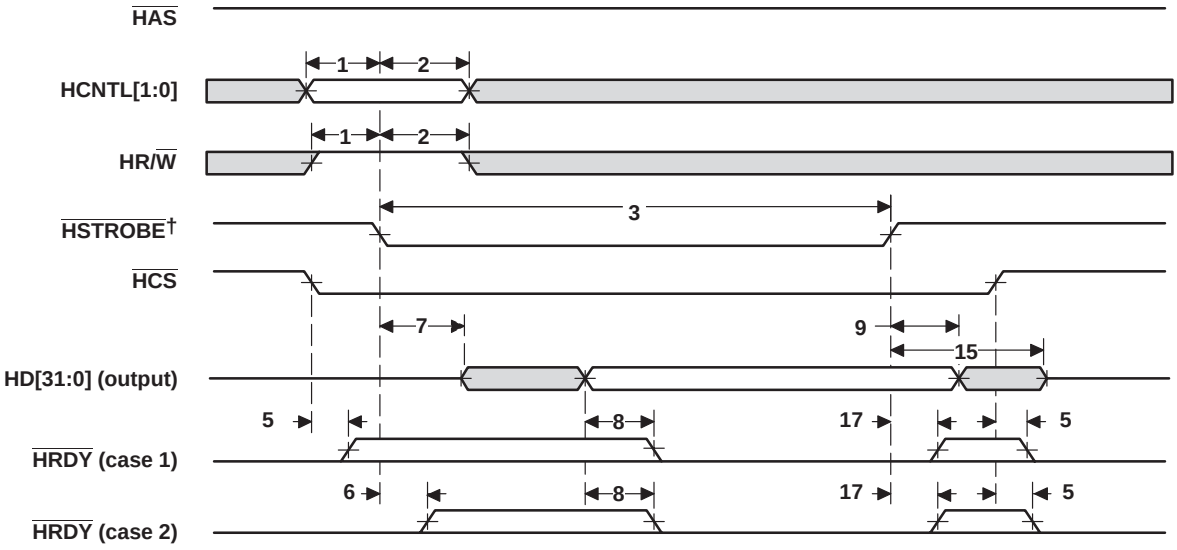



† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

**Figure 36. HPI16 Write Timing (HAS Used)**

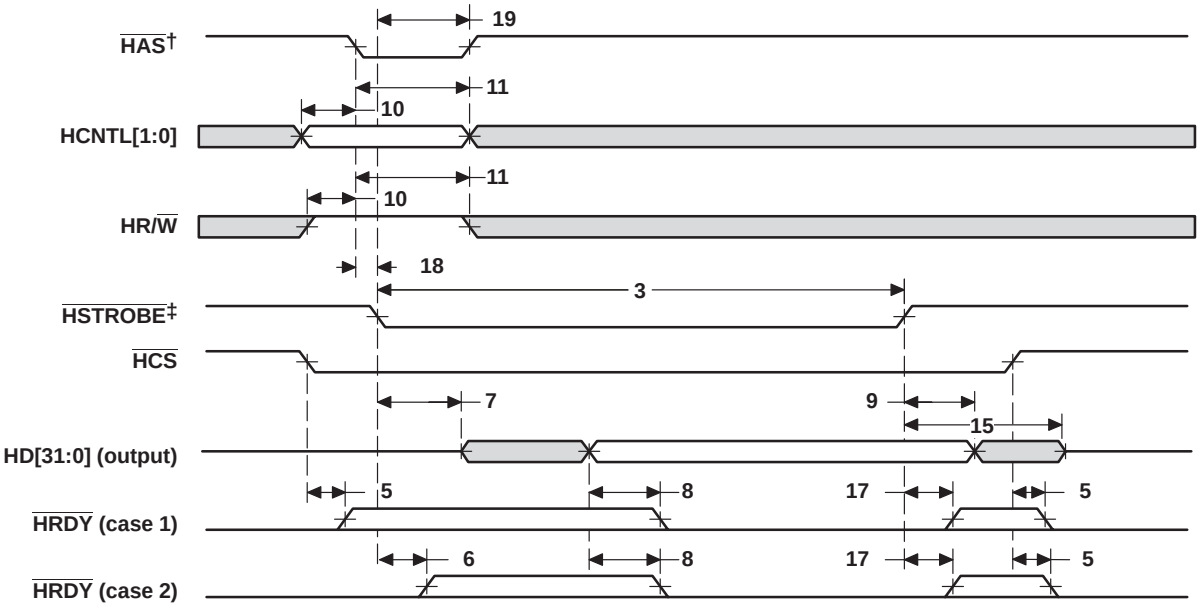
PRODUCT PREVIEW

## HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)




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

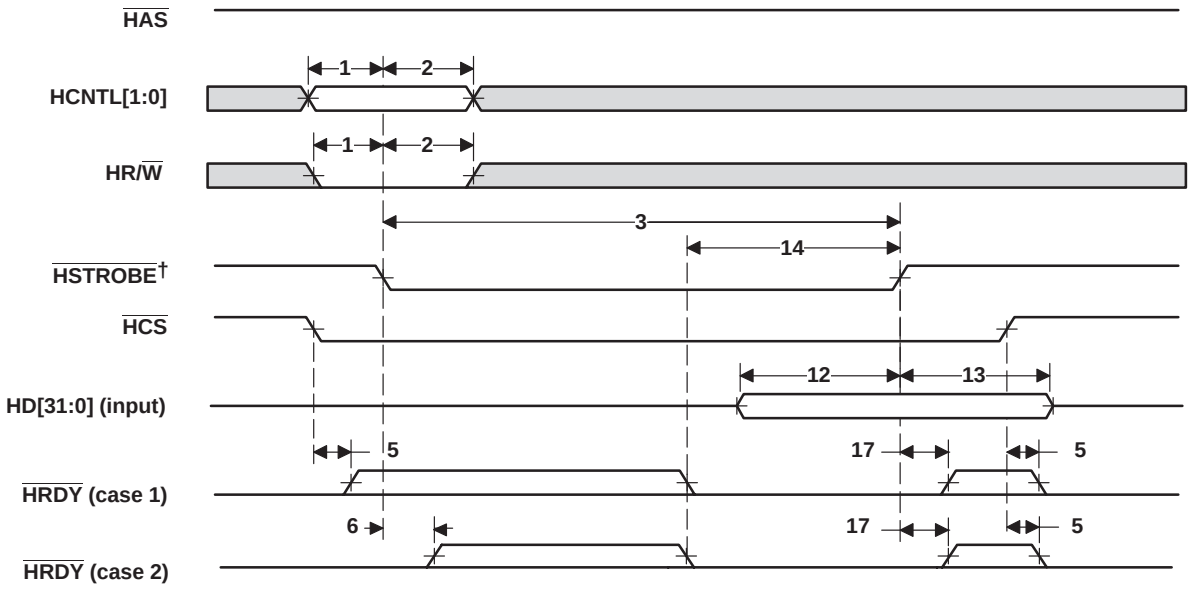
**Figure 37. HPI32 Read Timing (HAS Not Used, Tied High)**




† 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.

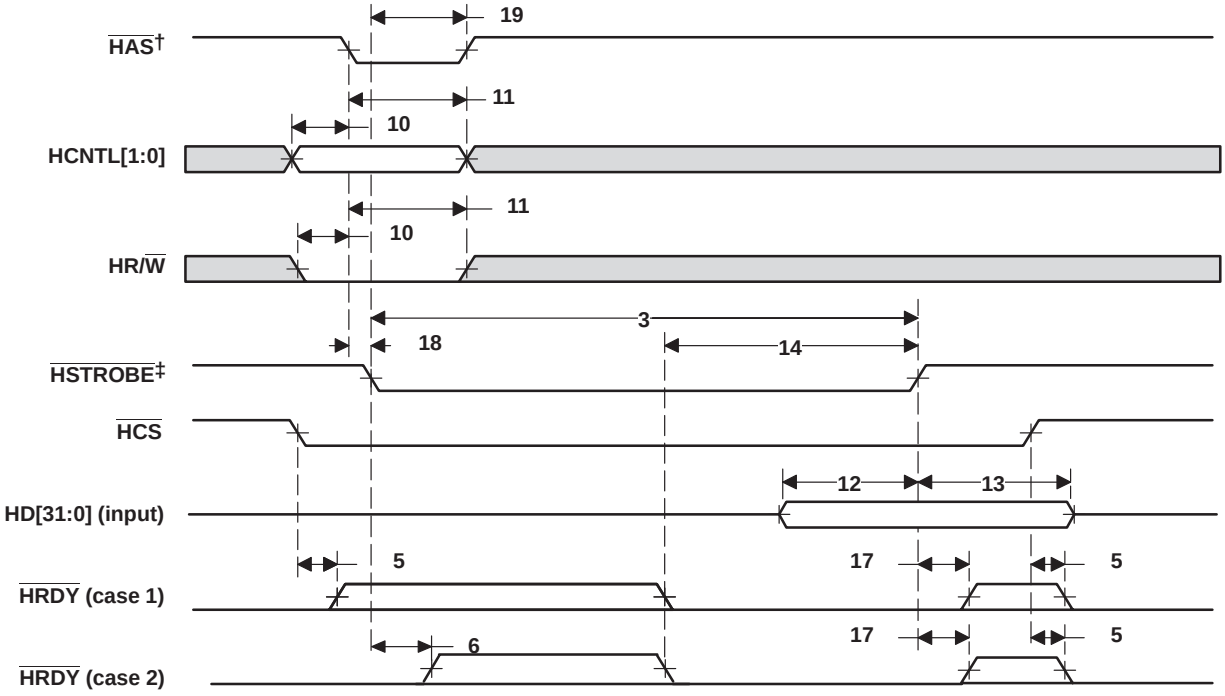
**Figure 38. HPI32 Read Timing (HAS Used)**

## 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 39. 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 40. HPI32 Write Timing (HAS Used)**

PRODUCT PREVIEW

## PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING

### timing requirements for PCLK† (see Figure 41)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(PCLK)}$ | Cycle time, PCLK | | | ns |
| 2 | $t_{w(PCLKH)}$ | Pulse duration, PCLK high | | | ns |
| 3 | $t_{w(PCLKL)}$ | Pulse duration, PCLK low | | | ns |
| 4 | $t_{sr(PCLK)}$ | Δv/Δt slew rate, PCLK | | | V/ns |

† For 3.3 V operation, the reference points for the rise and fall transitions are measured at $V_{ILP}$ MAX and $V_{IHP}$ MIN.

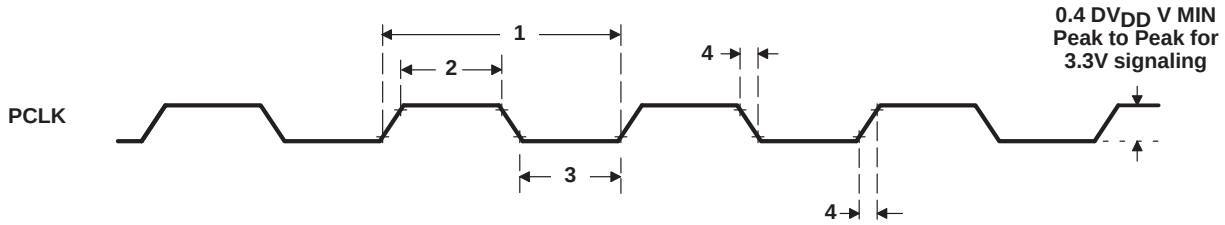



**Figure 41. PCLK Timing**

### timing requirements for PCI reset (see Figure 42)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(PRST)}$ | Pulse duration, $\overline{PRST}$ | | | ms |
| 2 | $t_{su(PCLKA-PRSTH)}$ | Setup time, PCLK active before $\overline{PRST}$ high | | | µs |

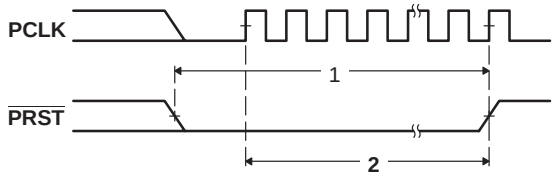



**Figure 42. PCI Reset ($\overline{PRST}$) Timing**

PRODUCT PREVIEW

## PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING (CONTINUED)

### timing requirements for PCI inputs (see Figure 43)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 5 | $t_{su(IV-PCLKH)}$ | Setup time, input valid before PCLK high | | | ns |
| 6 | $t_{h(IV-PCLKH)}$ | Hold time, input valid after PCLK high | | | ns |

### switching characteristics over recommended operating conditions for PCI outputs (see Figure 43)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{d(PCLKH-OV)}$ | Delay time, PCLK high to output valid | | | ns |
| 2 | $t_{d(PCLKH-OIV)}$ | Delay time, PCLK high to output invalid | | | ns |
| 3 | $t_{d(PCLKH-OLZ)}$ | Delay time, PCLK high to output low impedance | | | ns |
| 4 | $t_{d(PCLKH-OHZ)}$ | Delay time, PCLK high to output high impedance | | | ns |

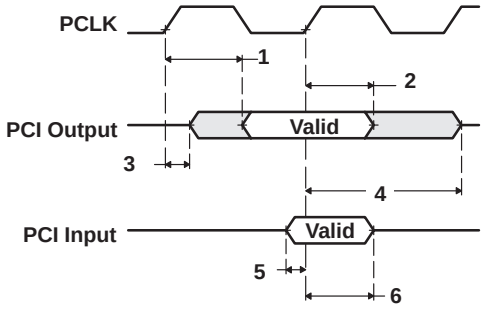



**Figure 43. PCI Intput/Output Timing**

PRODUCT PREVIEW

## PERIPHERAL COMPONENT INTERCONNECT (PCI) TIMING (CONTINUED)

### timing requirements for serial EEPROM interface (see Figure 44)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 8 | $t_{su(DIV-CLKH)}$ | Setup time, XSP_DI valid before XSP_CLK high | | | ns |
| 9 | $t_{h(CLKH-DIV)}$ | Hold time, XSP_DI valid after XSP_CLK high | | | ns |

### switching characteristics over recommended operating conditions for serial EEPROM interface† (see Figure 44)

| NO. | PARAMETER | | –300 | | | UNIT |
|---|---|---|---|---|---|---|
| | | | MIN | NOM | MAX | |
| 1 | $t_{w(CSL)}$ | Pulse duration, XSP_CS low | | | | ns |
| 2 | $t_{d(CLKL-CSL)}$ | Delay time, XSP_CLK low to XSP_CS low | | | | ns |
| 3 | $t_{d(CSH-CLKH)}$ | Delay time, XSP_CS high to XSP_CLK high | | | | ns |
| 4 | $t_{w(CLKH)}$ | Pulse duration, XSP_CLK high | | | | ns |
| 5 | $t_{w(CLKL)}$ | Pulse duration, XSP_CLK low | | | | ns |
| 6 | $t_{osu(DOV-CLKH)}$ | Output setup time, XSP_DO valid after XSP_CLK high | | | | ns |
| 7 | $t_{oh(CLKH-DOV)}$ | Output hold time, XSP_DO valid after XSP_CLK high | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

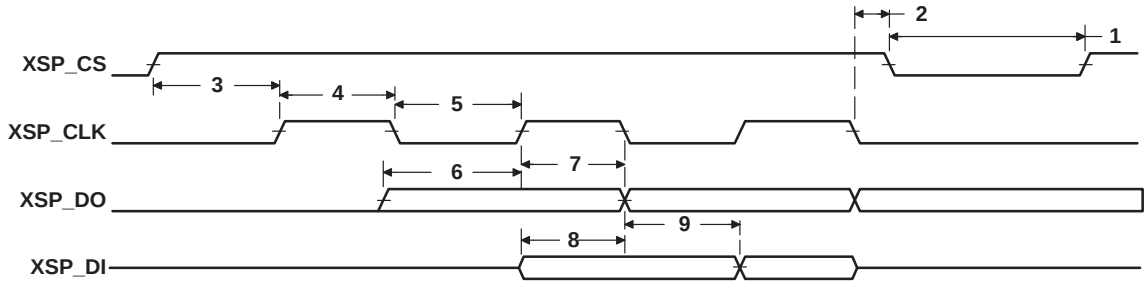



**Figure 44. PCI Serial EEPROM Interface Timing**

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING

### timing requirements for McBSP†‡ (see Figure 45)

| NO. | | | | –300 | | UNIT |
|---|---|---|---|---|---|---|
| | | | | MIN | MAX | |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X ext | § | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X ext | | | ns |
| 5 | $t_{su(FRH-CKRL)}$ | Setup time, external FSR high before CLKR low | CLKR int | | | ns |
| | | | CLKR ext | | | |
| 6 | $t_{h(CKRL-FRH)}$ | Hold time, external FSR high after CLKR low | CLKR int | | | ns |
| | | | CLKR ext | | | |
| 7 | $t_{su(DRV-CKRL)}$ | Setup time, DR valid before CLKR low | CLKR int | | | ns |
| | | | CLKR ext | | | |
| 8 | $t_{h(CKRL-DRV)}$ | Hold time, DR valid after CLKR low | CLKR int | | | ns |
| | | | CLKR ext | | | |
| 10 | $t_{su(FXH-CKXL)}$ | Setup time, external FSX high before CLKX low | CLKX int | | | ns |
| | | | CLKX ext | | | |
| 11 | $t_{h(CKXL-FXH)}$ | Hold time, external FSX high after CLKX low | CLKX int | | | ns |
| | | | CLKX ext | | | |

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

‡ P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

§ The maximum bit rate for McBSP-to-McBSP communications is 75 MHz; therefore, the minimum CLKR/X clock cycle is either four times the CPU cycle time (4P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 300 MHz (P = 3.33 ns), use 13.33 ns (75 MHz) as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 200 MHz (P = 5 ns), use 4P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies to the following hardware configuration: the serial port is a Master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a Slave.

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### switching characteristics over recommended operating conditions for McBSP†‡ (see Figure 45)

| NO. | PARAMETER | | | –300 MIN | –300 MAX | UNIT |
|---|---|---|---|---|---|---|
| 1 | $t_{d(CKSH-CKRXH)}$ | Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input | | | | ns |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X int | §¶ | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | # | # | ns |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | | | ns |
| 9 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | CLKX int | | | ns |
| | | | CLKX ext | | | |
| 12 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | CLKX int | | | ns |
| | | | CLKX ext | | | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX int | ‖ | ‖ | ns |
| | | | CLKX ext | ‖ | ‖ | |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid<br>ONLY applies when in data delay 0 (XDATDLY = 00b) mode | FSX int | | | ns |
| | | | FSX ext | | | |

† 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.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.
¶ The maximum bit rate for McBSP-to-McBSP communications is 75 MHz; therefore, the minimum CLKR/X clock cycle is either four times the CPU cycle time (4P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 300 MHz (P = 3.33 ns), use 13.33 ns (75 MHz) as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 200 MHz (P = 5 ns), use 4P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies to the following hardware configuration: the serial port is a Master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a Slave.
# 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

PRODUCT PREVIEW

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

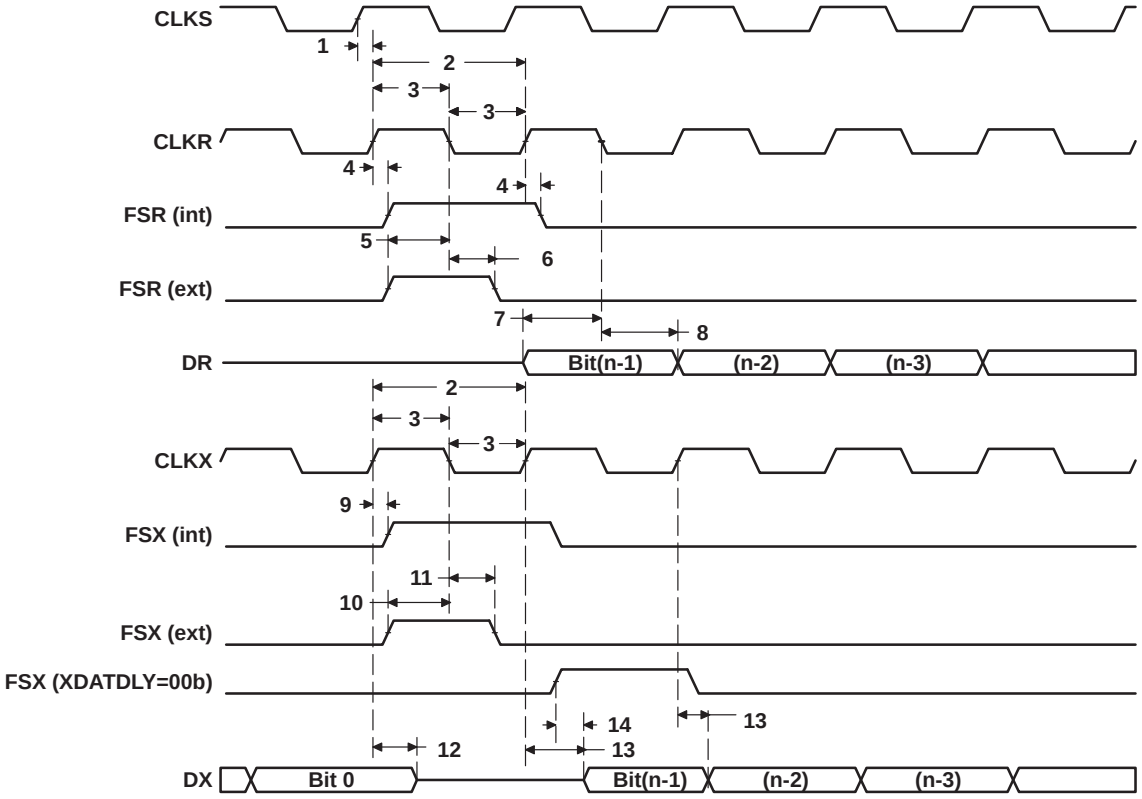



**Figure 45. McBSP Timing**

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### timing requirements for FSR when GSYNC = 1 (see Figure 46)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{su(FRH-CKSH)}$ | Setup time, FSR high before CLKS high | | | ns |
| 2 | $t_{h(CKSH-FRH)}$ | Hold time, FSR high after CLKS high | | | ns |

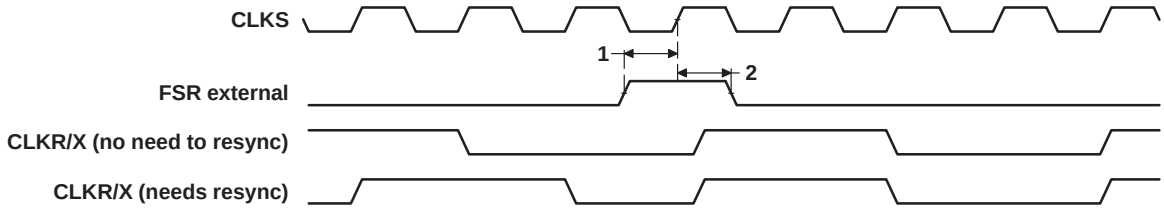



**Figure 46. FSR Timing When GSYNC = 1**

PRODUCT PREVIEW

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 47)

| NO. | | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXL)}$ | Setup time, DR valid before CLKX low | | | | | ns |
| 5 | $t_{h(CKXL-DRV)}$ | Hold time, DR valid after CLKX low | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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 47)

| NO. | PARAMETER | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low¶ | | | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high# | | | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | | | | | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | | | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | | | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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).

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

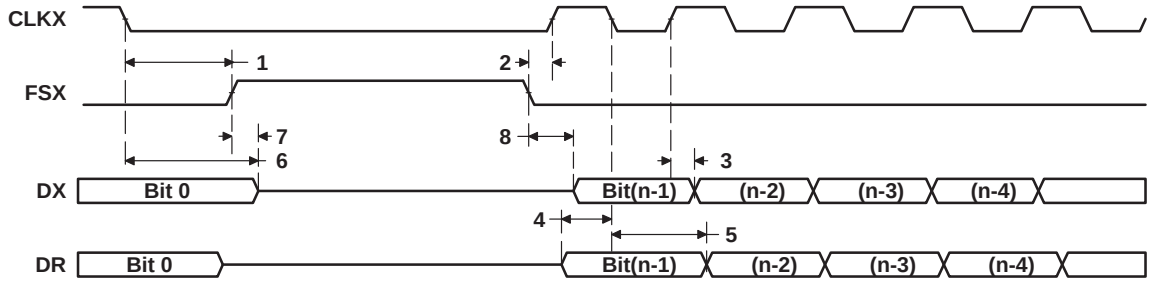



**Figure 47. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0**

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 48)

| NO. | | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ | Setup time, DR valid before CLKX high | | | | | ns |
| 5 | $t_{h(CKXH-DRV)}$ | Hold time, DR valid after CLKX high | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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 48)

| NO. | PARAMETER | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low¶ | | | | | ns |
| 2 | $t_{d(FXL-CKXH)}$ | Delay time, FSX low to CLKX high# | | | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | | | | | ns |
| 6 | $t_{dis(CKXL-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX low | | | | | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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).

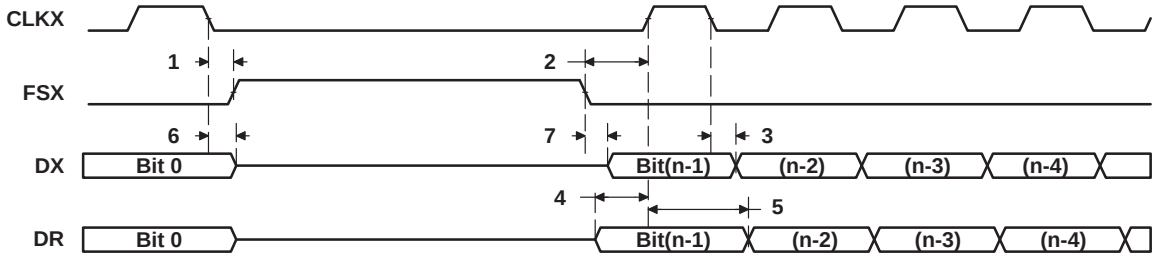



**Figure 48. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0**

PRODUCT PREVIEW

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 49)

| NO. | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ Setup time, DR valid before CLKX high | | | | | ns |
| 5 | $t_{h(CKXH-DRV)}$ Hold time, DR valid after CLKX high | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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 49)

| NO. | PARAMETER | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high¶ | | | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low# | | | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | | | | | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | | | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | | | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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).

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

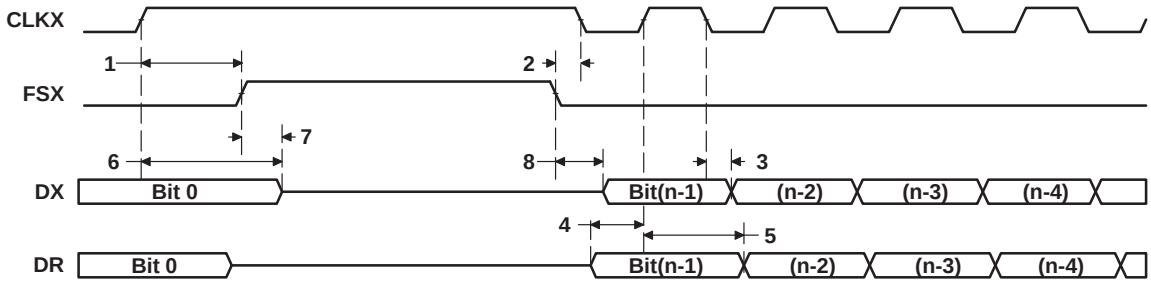



**Figure 49. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1**

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

### timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 50)

| NO. | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|
| | | MASTER | | SLAVE | | |
| | | MIN | MAX | MIN | MAX | |
| 4 | $t_{su(DRV-CKXH)}$ Setup time, DR valid before CLKX high | | | | | ns |
| 5 | $t_{h(CKXH-DRV)}$ Hold time, DR valid after CLKX high | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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 50)

| NO. | PARAMETER | | –300 | | | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MASTER§ | | SLAVE | | |
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high¶ | | | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low# | | | | | ns |
| 3 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | | | | | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | | | | | ns |
| 7 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 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).

## MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

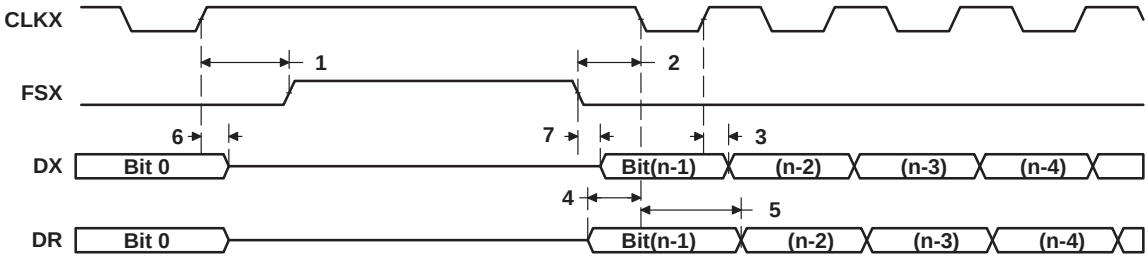



**Figure 50. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1**

## TIMER TIMING

### timing requirements for timer inputs† (see Figure 51)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(TINPH)}$ | Pulse duration, TINP high | | | ns |
| 2 | $t_{w(TINPL)}$ | Pulse duration, TINP low | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

### switching characteristics over recommended operating conditions for timer outputs† (see Figure 51)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{w(TOUTH)}$ | Pulse duration, TOUT high | | | ns |
| 4 | $t_{w(TOUTL)}$ | Pulse duration, TOUT low | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

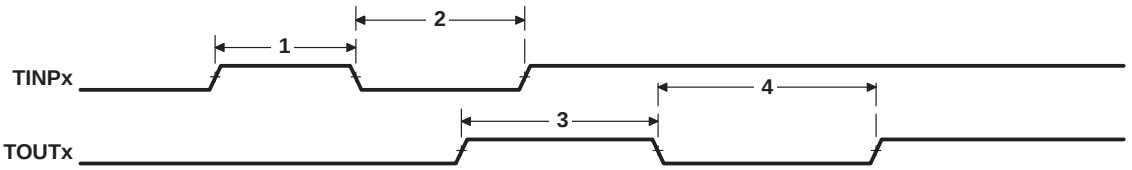



**Figure 51. Timer Timing**

## GENERAL-PURPOSE INPUT/OUTPUT (GPIO) PORT TIMING

### timing requirements for GPIO inputs† (see Figure 52)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{w(GPIH)}$ | Pulse duration, GPIx high | | | ns |
| 2 | $t_{w(GPIL)}$ | Pulse duration, GPIx low | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

### switching characteristics over recommended operating conditions for GPIO outputs† (see Figure 52)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 3 | $t_{w(GPOH)}$ | Pulse duration, GPOx high | | | ns |
| 4 | $t_{w(GPOL)}$ | Pulse duration, GPOx low | | | ns |

† P = 1/CPU clock frequency in ns. For example, when running parts at 300 MHz, use P = 3.33 ns.

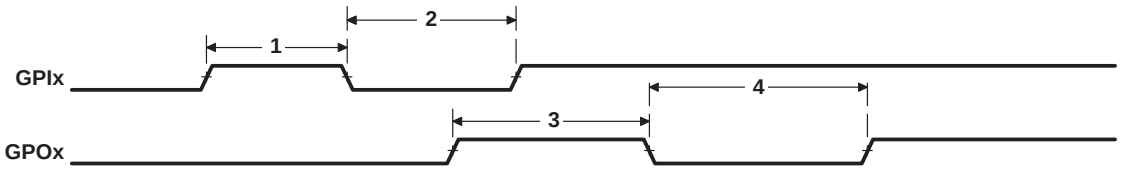



**Figure 52. GPIO Port Timing**

PRODUCT PREVIEW

## JTAG TEST-PORT TIMING

### timing requirements for JTAG test port (see Figure 53)

| NO. | | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 1 | $t_{c(TCK)}$ | Cycle time, TCK | | | ns |
| 3 | $t_{su(TDIV-TCKH)}$ | Setup time, TDI/TMS/$\overline{TRST}$ valid before TCK high | | | ns |
| 4 | $t_{h(TCKH-TDIV)}$ | Hold time, TDI/TMS/$\overline{TRST}$ valid after TCK high | | | ns |

### switching characteristics over recommended operating conditions for JTAG test port (see Figure 53)

| NO. | PARAMETER | | –300 | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| 2 | $t_{d(TCKL-TDOV)}$ | Delay time, TCK low to TDO valid | | | ns |

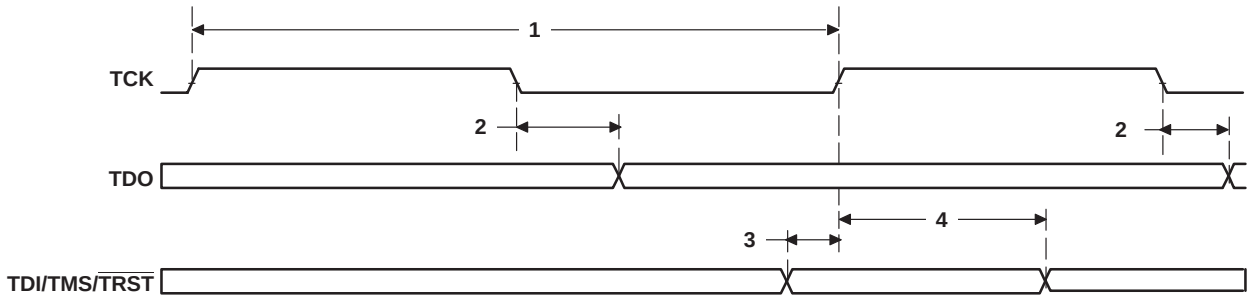
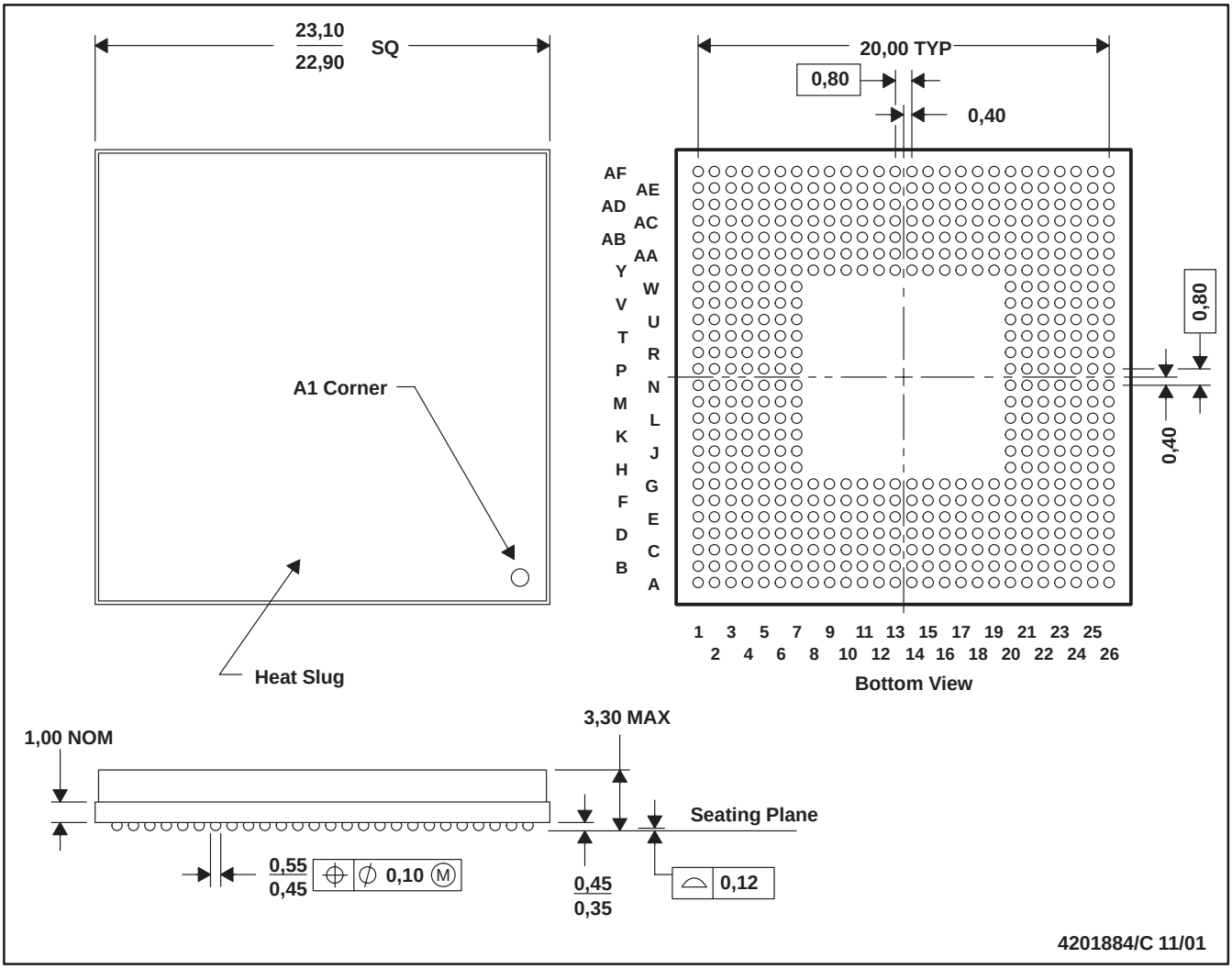



**Figure 53. JTAG Test-Port Timing**

PRODUCT PREVIEW

## MECHANICAL DATA

**GLZ (S-PBGA-N532)** **PLASTIC BALL GRID ARRAY**




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL)
D. Flip chip application only

### thermal resistance characteristics (S-PBGA package)

| NO | | | °C/W | Air Flow (m/s†) |
|---|---|---|---|---|
| 1 | $R\Theta_{JC}$ | Junction-to-case | 1.4 | N/A |
| 2 | $R\Theta_{JB}$ | Junction-to-board | 4.9 | 0.00 |
| 3 | $R\Theta_{JA}$ | Junction-to-free air | 17.5 | 0.00 |
| 4 | $R\Theta_{JA}$ | Junction-to-free air | 14.5 | 0.75 |
| 5 | $R\Theta_{JA}$ | Junction-to-free air | 12.5 | 1.25 |
| 6 | $R\Theta_{JA}$ | Junction-to-free air | 10.4 | 2.00 |
| 7 | $Psi_{JT}$ | Junction-to-package top | 0.7 | N/A |
| 8 | $Psi_{JB}$ | Junction-to-board | 4.5 | N/A |

† m/s = meters per second

PRODUCT PREVIEW

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265