- **Highest-Performance Fixed-Point Digital Signal Processor (DSP) – TMS320C6414**
  - **2.5-, 2-, 1.67-ns Instruction Cycle Time**
  - **400-, 500-, 600-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **Twenty-Eight Operations/Cycle**
  - **3200, 4000, 4800 MIPS**
  - **Fully Software-Compatible With C62x™**
  - **Pin-Compatible With C6415/16 Devices**
- **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 With 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**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit-Counting**
  - **Normalization**
  - **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)**
  - **8M-Bit (1024K-Byte) L2 Unified Mapped RAM/Cache (Flexible RAM/Cache Allocation)**
- **Two External Memory Interfaces (EMIFs)**
  - **One 64-Bit (EMIFA)**
  - **One 16-Bit (EMIFB)**
  - **Glueless Interface to Asynchronous Memories (SRAM and EPROM) and Synchronous Memories (SDRAM, SBSRAM, ZBT SRAM, and FIFO)**
  - **1280M-Byte Total Addressable External Memory Space**
- **Enhanced Direct-Memory-Access (EDMA) Controller (64 Independent Channels)**
- **Host-Port Interface (HPI)**
  - **User-Configurable Bus-Width (32-/16-bit)**
  - **Access to Entire Memory Map**
- **Three 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**
- **General-Purpose I/O (GPIO) Pins**
  - **Thirteen Dedicated GPIO Pins**
  - **Total of Sixteen GPIO Pins**
  - **Programmable Interrupt/Event Generation Modes**
- **Flexible Phase-Locked-Loop (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, 1.2-V Internal (-400, -500 Speeds)**
- **3.3-V I/Os, 1.4-V Internal (-600 Speed)**

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.

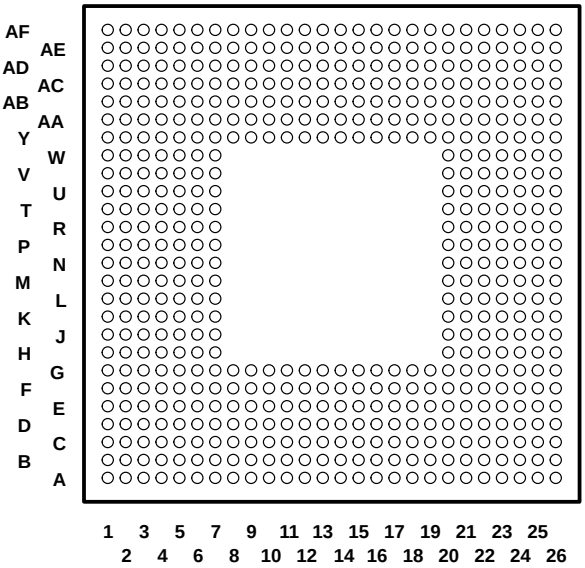
C62x, VelociTI.2, VelociTI, and TMS320C64x are trademarks of Texas Instruments.
Motorola is a trademark of Motorola, Inc.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

Table of Contents

PRODUCT PREVIEW

GLZ BGA package (bottom view)

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



description

The TMS320C64x™ DSPs (including the TMS320C6414 device) are the highest-performance fixed-point DSP generation in the TMS320C6000™ DSP platform. The TMS320C6414 (C6414) 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 4800 million instructions per second (MIPS) at a clock rate of 600 MHz, the C6414 device offers cost-effective solutions to high-performance DSP programming challenges. The C6414 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 C6414 can produce two 32-bit multiply-accumulates (MACs) per cycle for a total of 1200 million MACs per second (MMACS), or eight 8-bit MACs per cycle for a total of 4800 MMACS. The C6414 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals similar to the other C6000™ DSP platform devices.

The C6414 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 an 8-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 three 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 general-purpose input/output port (GPIO) with 16 GPIO pins; and two glueless external memory interfaces (64-bit EMIFA and 16-bit EMIFB†), both of which are capable of interfacing to synchronous and asynchronous memories and peripherals.

The C6414 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

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

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

device characteristics

Table 1 provides an overview of the C6414 DSP. The table shows significant features of the C6414 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 C6414 Processor

HARDWARE FEATURES		C6414
Peripherals	EMIFA (64-bit bus width)	1
	EMIFB (16-bit bus width)	1
	EDMA (64 independent channels)	1
	HPI (32- or 16-bit user selectable)	1 (HPI16 or HPI32)
	McBSPs	3
	32-Bit Timers	3
	General-Purpose Input/Outputs (GPIOs)	16
On-Chip Memory	Size (Bytes)	1056K
	Organization	16K-Byte (16KB) L1 Program (L1P) Cache 16KB L1 Data (L1D) Cache 1024KB Unified Mapped RAM/Cache (L2)
CPU ID + CPU Rev ID	Control Status Register (CSR.[31:16])	0x0C01
Frequency	MHz	400, 500, 600
Cycle Time	ns	2.5 ns (C6414-400) 2 ns (C6414-500) 1.67 ns (C6414-600)
Voltage	Core (V)	1.2 V (-400, -500) 1.4 V (-600)
	I/O (V)	3.3 V
PLL Options	CLKIN frequency multiplier	Bypass (x1), x6, x12
BGA Package	23 x 23 mm	532-Pin BGA (GLZ)
Process Technology	µm	0.12 µm
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	PP
Device Part Numbers	(For more details on the C6000™ DSP part numbering, see Figure 4)	TMX320C6414GLZ

PRODUCT PREVIEW

device compatibility

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

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

- All devices are using the same peripherals.
  The C6414 is pin-for-pin compatible with the C6415/C6416 when the PCI and UTOPIA peripherals on the C6415/C6416 are disabled.
  The C6415 is pin-for-pin compatible with the C6416 when they are in the same peripheral selection mode.
  [For more information on peripheral selection, see the Device Configurations section of the TMS320C6415 and TMS320C6416 device-specific data sheets (literature number SPRS146 and SPRS164, respectively).]
- The BEA[9:7] pins are properly pulled up/down.
  [For more details on the device-specific BEA[9:7] pin configurations, see the Terminal Functions table of the TMS320C6414, TMS320C6415, and TMS320C6416 device-specific data sheets (literature number SPRS134, SPRS146, and SPRS164, respectively).]

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

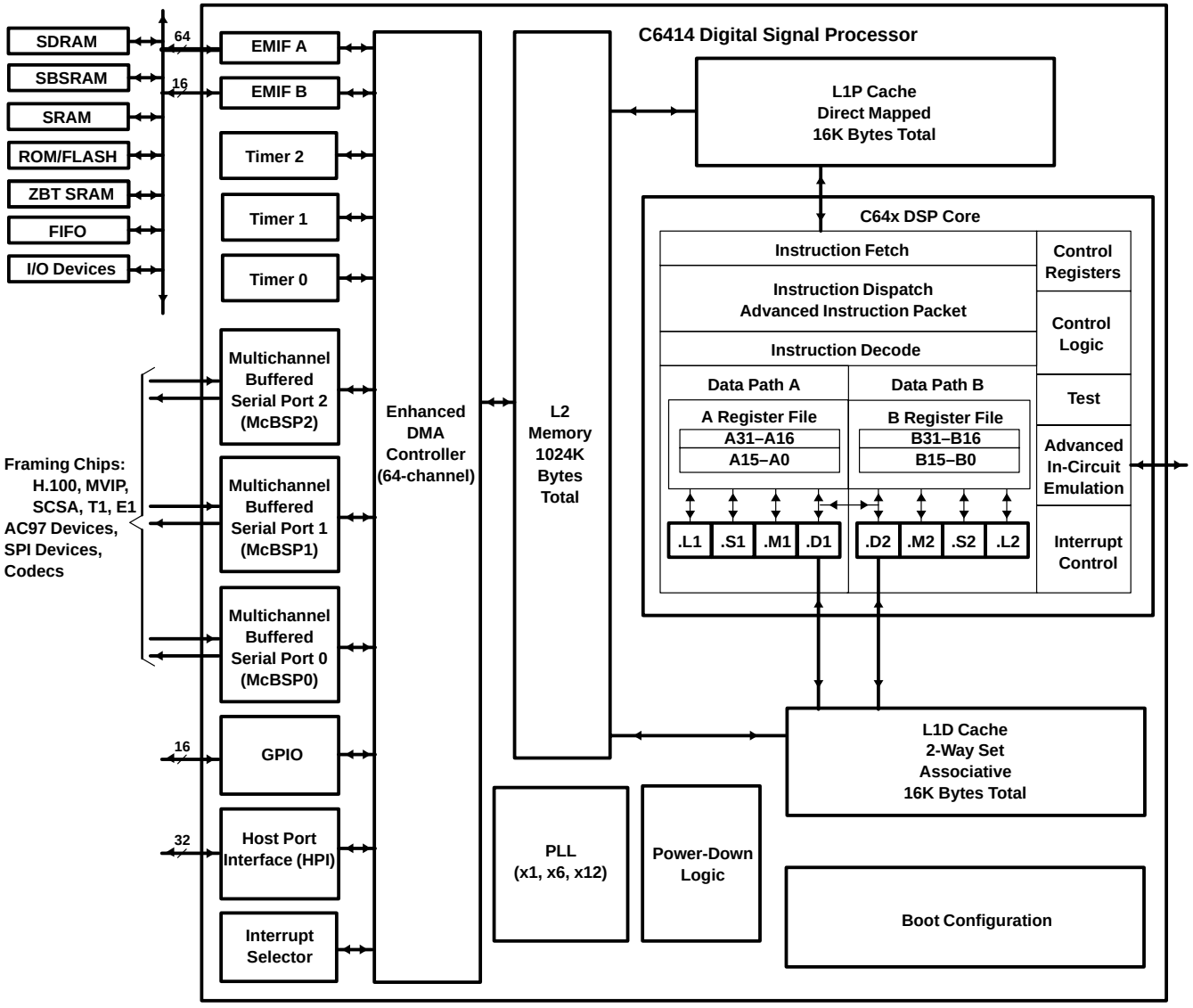
PERIPHERALS/COPROCESSORS	C6414	C6415	C6416
EMIFA (64-bit bus width)	√	√	√
EMIFB (16-bit bus width)	√	√	√
EDMA (64 independent channels)	√	√	√
HPI (32- or 16-bit user selectable)	√	√	√
PCI (32-bit)	—	√	√
McBSPs (McBSP0, McBSP1, McBSP2)	√	√	√
UTOPIA (8-bit mode)	—	√	√
Timers (32-bit) [TIMER0, TIMER1, TIMER2]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

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

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

PRODUCT PREVIEW

functional block and CPU (DSP core) diagram



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:

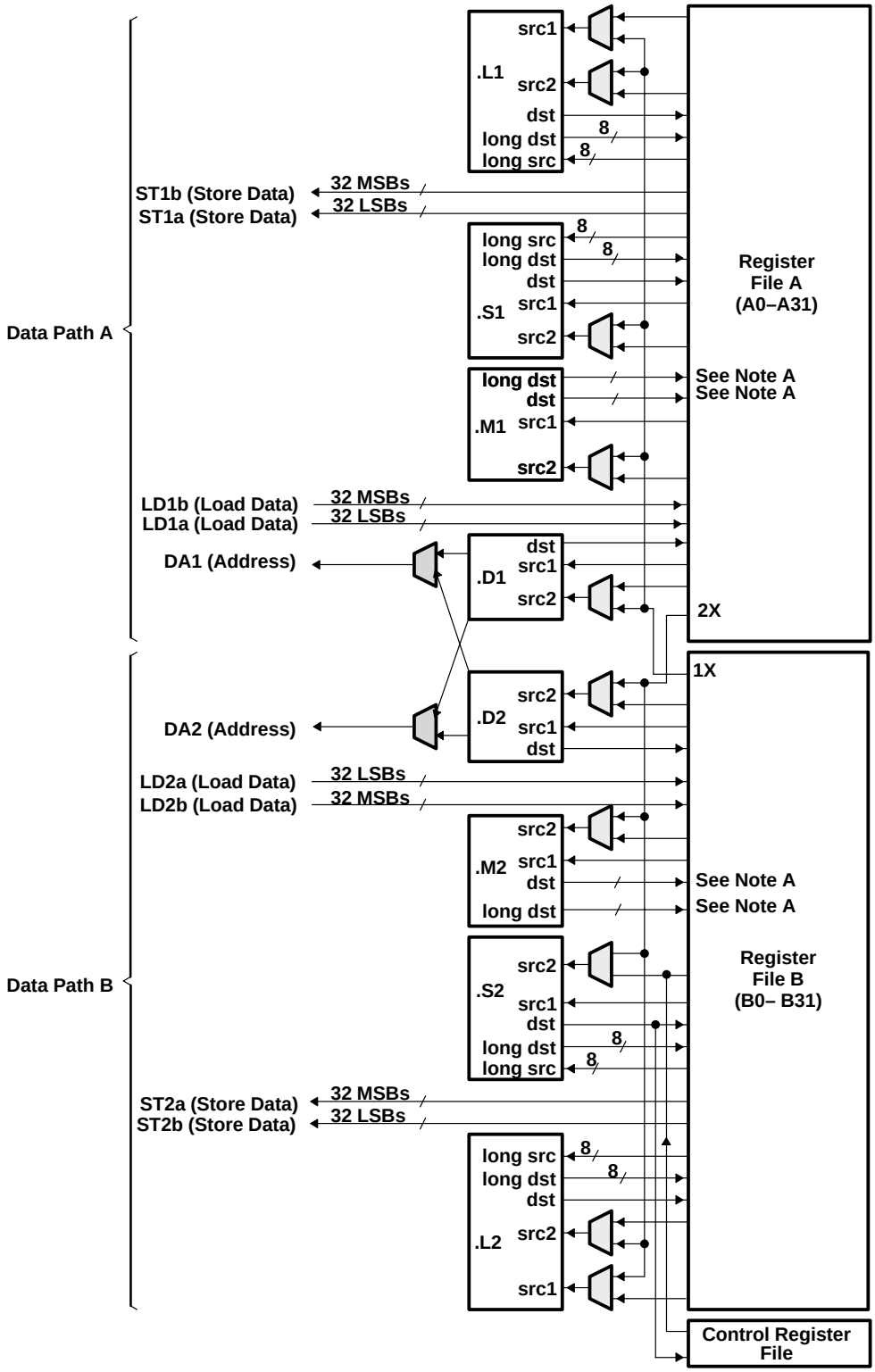
The *TMS320C6000 CPU and Instruction Set Reference Guide* (literature number SPRU189)

TMS320C64x Technical Overview (literature number SPRU395)

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

TMS320C67x is a trademark of Texas Instruments.

CPU (DSP core) description (continued)



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

Figure 1. TMS320C64x™ CPU (DSP Core) Data Paths

PRODUCT PREVIEW

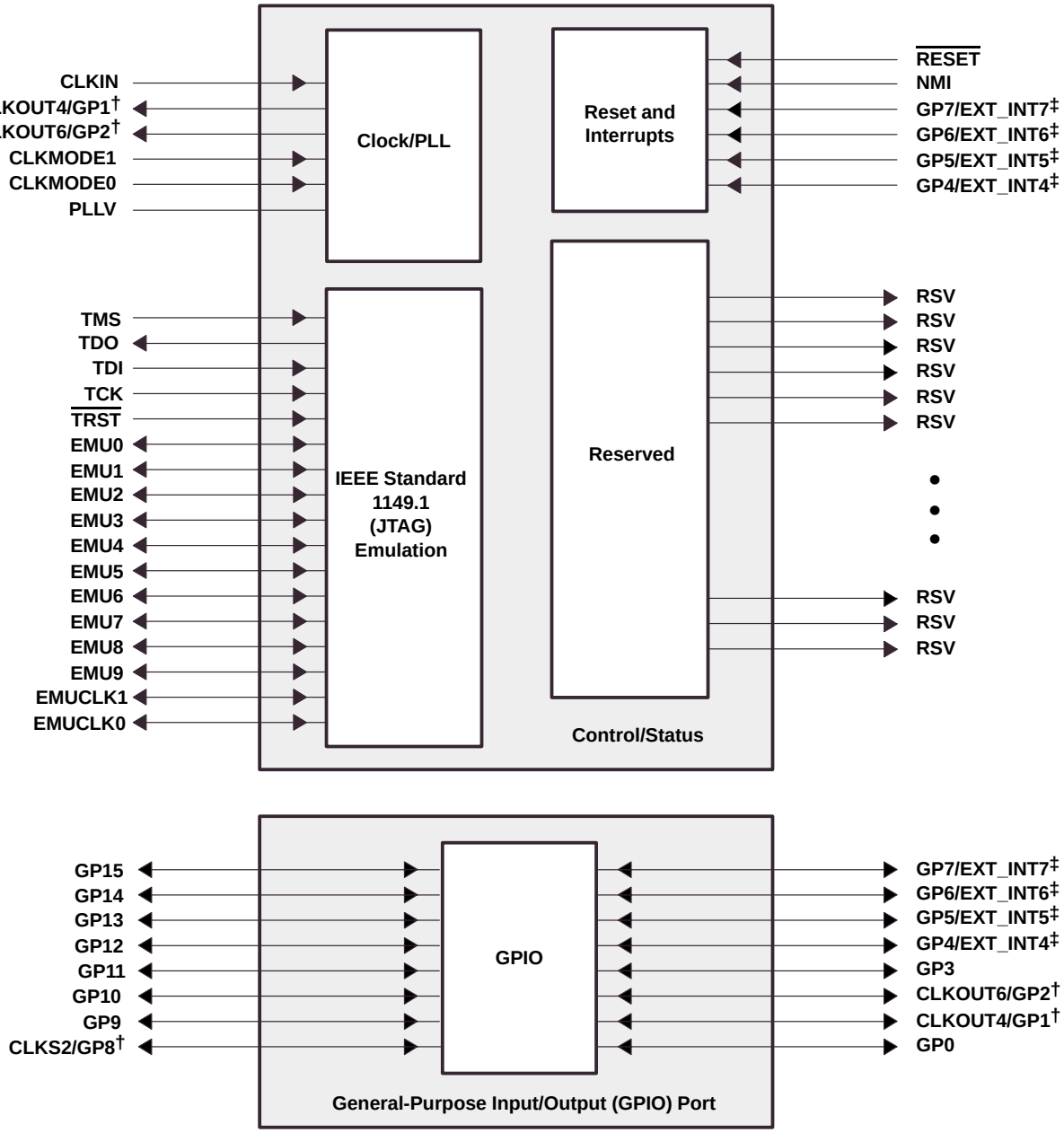
memory map summary

Table 3 shows the memory map address ranges of the C6414 device. Internal memory is always located at address 0 and can be used as both program and data memory. The external memory address ranges in the C6414 device begin at the hex address locations 0x6000 0000 for EMIFB and 0x8000 0000 for EMIFA.

Table 3. TMS320C6414 Memory Map Summary

MEMORY BLOCK DESCRIPTION	BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
Internal RAM (L2)	1M	0000 0000 – 000F FFFF
Reserved	23M	0010 0000 – 017F FFFF
External Memory Interface A (EMIFA) Registers	256K	0180 0000 – 0183 FFFF
L2 Registers	256K	0184 0000 – 0187 FFFF
HPI Registers	256K	0188 0000 – 018B FFFF
McBSP 0 Registers	256K	018C 0000 – 018F FFFF
McBSP 1 Registers	256K	0190 0000 – 0193 FFFF
Timer 0 Registers	256K	0194 0000 – 0197 FFFF
Timer 1 Registers	256K	0198 0000 – 019B FFFF
Interrupt Selector Registers	256K	019C 0000 – 019F FFFF
EDMA RAM and EDMA Registers	256K	01A0 0000 – 01A3 FFFF
McBSP 2 Registers	256K	01A4 0000 – 01A7 FFFF
EMIFB Registers	256K	01A8 0000 – 01AB FFFF
Timer 2 Registers	256K	01AC 0000 – 01AF FFFF
GPIO Registers	256K	01B0 0000 – 01B3 FFFF
Reserved	5M – 256K	01B4 0000 – 01FF FFFF
QDMA Registers	52	0200 0000 – 0200 0033
Reserved	736M – 52	0200 0034 – 2FFF FFFF
McBSP 0 Data	64M	3000 0000 – 33FF FFFF
McBSP 1 Data	64M	3400 0000 – 37FF FFFF
McBSP 2 Data	64M	3800 0000 – 3BFF FFFF
Reserved	576M	3C00 0000 – 5FFF FFFF
EMIFB CE0	64M	6000 0000 – 63FF FFFF
EMIFB CE1	64M	6400 0000 – 67FF FFFF
EMIFB CE2	64M	6800 0000 – 6BFF FFFF
EMIFB CE3	64M	6C00 0000 – 6FFF FFFF
Reserved	256M	7000 0000 – 7FFF FFFF
EMIFA CE0	256M	8000 0000 – 8FFF FFFF
EMIFA CE1	256M	9000 0000 – 9FFF FFFF
EMIFA CE2	256M	A000 0000 – AFFF FFFF
EMIFA CE3	256M	B000 0000 – BFFF FFFF
Reserved	1G	C000 0000 – FFFF FFFF

signal groups description



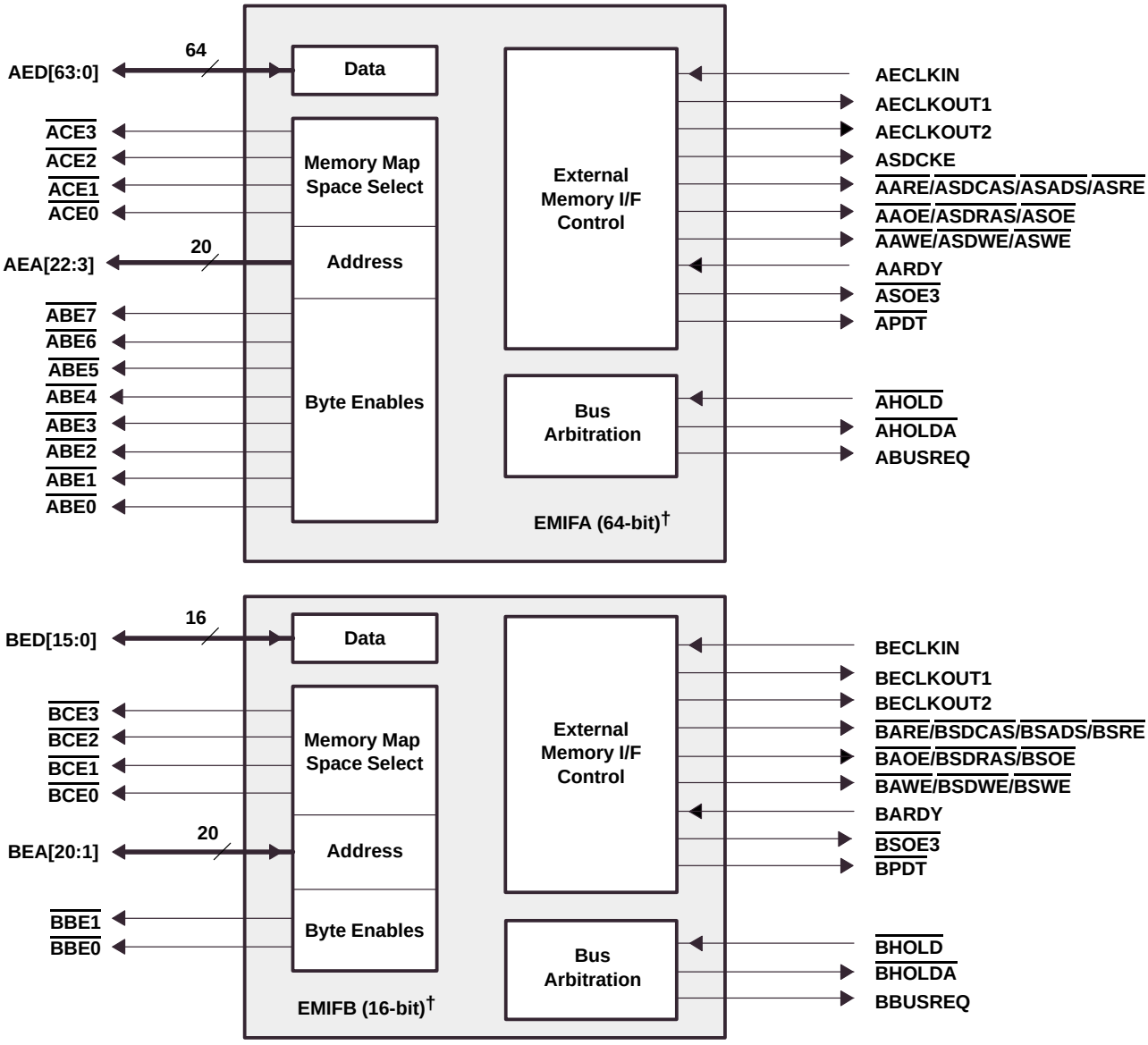
† These pins are muxed with the GPIO port pins and by default these signals function as clocks (CLKOUT4 or CLKOUT6) or McBSP2 clock source (CLKS2). To use these muxed pins as GPIO signals, the appropriate GPIO register bits (GPxEN and GPxDIR) must be properly enabled and configured. For more detail, 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.

Figure 2. CPU and Peripheral Signals

PRODUCT PREVIEW

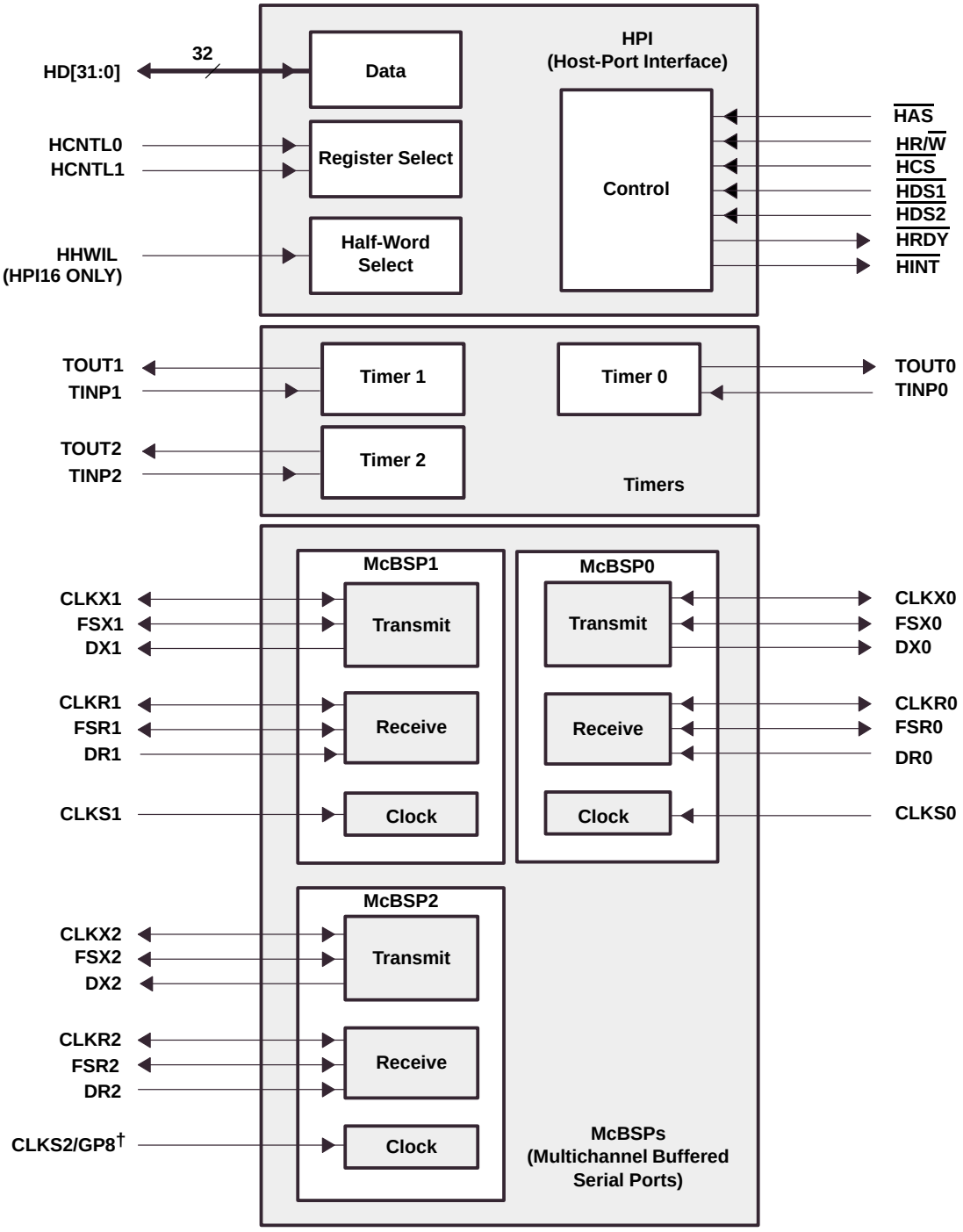
signal groups description (continued)



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

Figure 3. Peripheral Signals

signal groups description (continued)



† These pins are muxed with the GPIO port pins and by default these signals function as clocks (CLKOUT4 or CLKOUT6) or McBSP2 clock source (CLKS2). To use these muxed pins as GPIO signals, the appropriate GPIO register bits (GPxEN and GPxDIR) must be properly enabled and configured. See the Device Configurations section.

Figure 3. Peripheral Signals (Continued)

PRODUCT PREVIEW

DEVICE CONFIGURATIONS

multiplexed pins

Multiplexed pins are pins that are shared by more than one peripheral and are internally multiplexed. On the C6414 device, these multiplexed pins are configured by software and can be programmed to switch operating modes at anytime. Table 4 describes the multiplexed pins on the on the C6414 device.

Table 4. C6414 Device Multiplexed Pins

SIGNAL NAME	SIGNAL NO.	DEFAULT FUNCTION	DEFAULT SETTING	DESCRIPTION
CLKOUT4/GP1	AE6	CLKOUT4	GP1EN = 0	These pins are software-configurable. To use these pins as GPIO pins, the GPxEN bits in the GPIO Enable Register and the GPxDIR bits in the GPIO Direction Register must be properly configured. GPxEN = 1: GPx pin enabled GPxDIR = 0: GPx pin is an input GPxDIR = 1: GPx pin is an output
CLKOUT6/GP2	AD6	CLKOUT6	GP2EN = 0	
CLKS2/GP8	AE4	CLKS2	GP8EN = 0	

debugging considerations

It is recommended that external connections be provided to device configuration pins, including CLKMODE[1:0], BEA[20:14], and HD5. Although internal pullup/pulldown resistors exist on these pins, providing external connectivity adds convenience in debugging and flexibility in switching operating modes.

Internal pullup/pulldown resistors also exist on the non-configuration pins on the BEA bus (BEA[13:1]). Do not oppose the internal pullup/pulldown resistors on these non-configuration pins with external pullup/pulldown resistors. If an external controller provides signals to these non-configuration pins, these signals must be driven to the default state of the pins at reset, or not be driven at all.

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

PRODUCT PREVIEW

Terminal Functions

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
CLOCK/PLL CONFIGURATION				
CLKIN	H4	I	IPD	Clock Input. This clock is the input to the on-chip PLL.
CLKOUT4/GP1	AE6	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 2 pin (**I/O/Z**).
CLKOUT6/GP2	AD6	I/O/Z	IPD	Clock output at 1/6 of the device speed (**O/Z**) [default] or this pin can be programmed as a GPIO 2 pin (**I/O/Z**).
CLKMODE1	G1	I	IPD	Clock mode select • Selects whether the CPU clock frequency = input clock frequency x1 (Bypass), x6, or x12. For more details on CLKMODE pins and the PLL multiply factors, see the *Clock PLL* section of this data sheet.
CLKMODE0	H2	I	IPD	
PLLV§	J6	A¶		PLL voltage supply
JTAG EMULATION				
TMS	AB16	I	IPU	JTAG test-port mode select
TDO	AE19	O/Z	IPU	JTAG test-port data out
TDI	AF18	I	IPU	JTAG test-port data in
TCK	AF16	I	IPU	JTAG test-port clock
$\overline{\text{TRST}}$	AB15	I	IPD	JTAG test-port reset
EMU9	AE18	I/O/Z	IPU	Emulation pin 9. Reserved for future use, leave unconnected.
EMU8	AC17	I/O/Z	IPU	Emulation pin 8. Reserved for future use, leave unconnected.
EMU7	AF17	I/O/Z	IPU	Emulation pin 7. Reserved for future use, leave unconnected.
EMU6	AD17	I/O/Z	IPU	Emulation pin 6. Reserved for future use, leave unconnected.
EMU5	AE17	I/O/Z	IPU	Emulation pin 5. Reserved for future use, leave unconnected.
EMU4	AC16	I/O/Z	IPU	Emulation pin 4. Reserved for future use, leave unconnected.
EMU3	AD16	I/O/Z	IPU	Emulation pin 3. Reserved for future use, leave unconnected.
EMU2	AE16	I/O/Z	IPU	Emulation pin 2. Reserved for future use, leave unconnected.
EMU1	AC15	I/O/Z	IPU	Emulation pin 1#
EMU0	AF15	I/O/Z	IPU	Emulation pin 0#
EMUCLK1	AC18	I/O/Z	IPU	Emulation clock 1. Reserved for future use, leave unconnected.
EMUCLK0	AD18	I/O/Z	IPU	Emulation clock 0. Reserved for future use, leave unconnected.
RESETS, INTERRUPTS, AND GENERAL-PURPOSE INPUT/OUTPUTS				
$\overline{\text{RESET}}$	AC7	I		Device reset
NMI	B4	I	IPD	Nonmaskable interrupt, edge-driven (rising edge)
GP7/EXT_INT7	AF4	I/O/Z	IPU	GPIO (**I/O/Z**) or external interrupts (**input only**). The default after reset setting is GPIO enabled as input-only. • When these pins function as External Interrupts [by selecting the corresponding interrupt enable register bit (IER.[7:4])], they are edge-driven and the polarity can be independently selected via the External Interrupt Polarity Register bits (EXTPOL.[3:0]).
GP6/EXT_INT6	AD5			
GP5/EXT_INT5	AE5			
GP4/EXT_INT4	AF5			

† 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.)

§ PLLV is not part of external voltage supply. See CLOCK/PLL documentation 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	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
RESETS, INTERRUPTS, AND GENERAL-PURPOSE INPUT/OUTPUTS (CONTINUED)				
GP15	G3	I/O/Z		General-purpose input/output (GPIO) pins. By default, these pins are disabled. To enable these pins as GPIO pins, the GPxEN bits in the GPIO Enable Register and the GPxDIR bits in the GPIO Direction Register must be properly configured. GPxEN = 1: GPx pin enabled GPxDIR = 0: GPx pin is an input GPxDIR = 1: GPx pin is an output
GP14	F2			
GP13	G4			
GP12	J3			
GP11	F1			
GP10	L2			
GP9	M3			
GP3	AC6		IPD	General-purpose input/output (GPIO) pin. The default after reset setting is GPIO enabled as input-only.
GP0	AF6		IPD	General-purpose input/output (GPIO) pin.The default after reset setting is GPIO enabled as input-only. The GP0 pin (**I/O/Z**) can be configured as a general-purpose interrupt (GPINT) signal (**output only**).
CLKS2/GP8	AE4	I/O/Z	IPD	McBSP2 external clock source (CLKS2) [**input only**] [default] or this pin can be programmed as a GPIO 8 pin (**I/O/Z**).
CLKOUT6/GP2	AD6	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	AE6	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)				
$\overline{\text{HINT}}$	R4	O		Host interrupt (from DSP to host)
HCNTL1	R1	I		Host control – selects between control, address, or data registers
HCNTL0	T4	I		Host control – selects between control, address, or data registers
HHWIL	R3	I		Host half-word select – first or second half-word (not necessarily high or low order) [For HPI16 bus width selection only]
HR/$\overline{\text{W}}$	P1	I		Host read or write select
$\overline{\text{HAS}}$	T3	I		Host address strobe
$\overline{\text{HCS}}$	R2	I		Host chip select
$\overline{\text{HDS1}}$	T1	I		Host data strobe 1
$\overline{\text{HDS2}}$	T2	I		Host data strobe 2
$\overline{\text{HRDY}}$	P4	O		Host ready (from DSP to host)
HD31	J2	I/O/Z		Host-port data • Used for transfer of data, address, and control • Host-Port bus width is user-configurable at device reset via pullup/pulldown resistor on the HD5 pin: HD5 pin = 0: HPI operates as an HPI16 . (HPI bus is 16 bits wide. HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the high-impedance state.) HD5 pin = 1: HPI operates as an HPI32. (HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.)
HD30	K3			
HD29	J1			
HD28	K4			
HD27	K2			
HD26	L3			
HD25	K1			
HD24	L4			
HD23	L1			
HD22	M4			

† 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.)

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
HOST-PORT INTERFACE (HPI) (CONTINUED)				
HD21	M2	I/O/Z		Host-port data • Used for transfer of data, address, and control • Host-Port bus width is user-configurable at device reset via pullup/pulldown resistor on the HD5 pin: HD5 pin = 0: HPI operates as an HPI16 . (HPI bus is 16 bits wide. HD[15:0] pins are used and the remaining HD[31:16] pins are reserved pins in the high-impedance state.) HD5 pin = 1: HPI operates as an HPI32. (HPI bus is 32 bits wide. All HD[31:0] pins are used for host-port operations.)
HD20	N4			
HD19	M1			
HD18	N5			
HD17	N1			
HD16	P5			
HD15	U4			
HD14	U1			
HD13	U3			
HD12	U2			
HD11	V4			
HD10	V1			
HD9	V3			
HD8	V2			
HD7	W2			
HD6	W4			
HD5	Y1			
HD4	Y3			
HD3	Y2			
HD2	Y4			
HD1	AA1			
HD0	AA3			
EMIFA (64-bit) – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY‖				
$\overline{\text{ACE3}}$	L26	O/Z	IPU	EMIFA memory space enables • Enabled by bits 28 through 31 of the word address • Only one pin is asserted during any external data access
$\overline{\text{ACE2}}$	K23	O/Z	IPU	
$\overline{\text{ACE1}}$	K24	O/Z	IPU	
$\overline{\text{ACE0}}$	K25	O/Z	IPU	
$\overline{\text{ABE7}}$	T23	O/Z	IPU	EMIFA byte-enable control • Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory. • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{\text{ABE6}}$	T24	O/Z	IPU	
$\overline{\text{ABE5}}$	R25	O/Z	IPU	
$\overline{\text{ABE4}}$	R26	O/Z	IPU	
$\overline{\text{ABE3}}$	M25	O/Z	IPU	
$\overline{\text{ABE2}}$	M26	O/Z	IPU	
$\overline{\text{ABE1}}$	L23	O/Z	IPU	
$\overline{\text{ABE0}}$	L24	O/Z	IPU	
$\overline{\text{APDT}}$	M22	O/Z	IPU	EMIFA peripheral data transfer, allows direct transfer between external peripherals

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

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

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

PRODUCT PREVIEW

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-BIT) – BUS ARBITRATION‖				
AHOLDA	N22	O	IPU	EMIFA hold-request-acknowledge to the host
AHOLD	V23	I	IPU	EMIFA hold request from the host
ABUSREQ	P22	O	IPU	EMIFA bus request output
EMIFA (64-BIT) – ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL‖				
AECLKIN	H25	I	IPD	EMIFA external input clock. The EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) is selected at reset via the pullup/pulldown resistors on the BEA[17:16] pins. AECLKIN is the default for the EMIFA input clock.
AECLKOUT2	J23	O/Z	IPD	EMIFA output clock 2. Programmable to be EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) frequency divided-by-1, -2, or -4.
AECLKOUT1	J26	O/Z	IPD	EMIFA output clock 1 [at EMIFA input clock (AECLKIN, CPU/4 clock, or CPU/6 clock) frequency].
AARE/ ASDCAS/ ASADS/ASRE	J25	O/Z	IPU	EMIFA asynchronous memory read-enable/SDRAM column-address strobe/programmable synchronous interface-address strobe or read-enable • For programmable synchronous interface, the RENEN field in the CE Space Secondary Control Register (CExSEC) selects between ASADS and ASRE: If RENEN = 0, then the ASADS/ASRE signal functions as the ASADS signal. If RENEN = 1, then the ASADS/ASRE signal functions as the ASRE signal.
AAOE/ ASDRAS/ ASOE	J24	O/Z	IPU	EMIFA asynchronous memory output-enable/SDRAM row-address strobe/programmable synchronous interface output-enable
AAWE/ ASDWE/ ASWE	K26	O/Z	IPU	EMIFA asynchronous memory write-enable/SDRAM write-enable/programmable synchronous interface write-enable
ASDCKE	L25	O/Z	IPU	EMIFA SDRAM clock-enable (used for self-refresh mode). [EMIFA module only.] • If SDRAM is not in system, ASDCKE can be used as a general-purpose output.
ASOE3	R22	O/Z	IPU	EMIFA synchronous memory output-enable for ACE3 (for glueless FIFO interface)
AARDY	L22	I	IPU	Asynchronous memory ready input
EMIFA (64-BIT) – ADDRESS‖				
AEA22	T22	O/Z	IPD	EMIFA external address (doubleword address)
AEA21	V24			
AEA20	V25			
AEA19	V26			
AEA18	U23			
AEA17	U24			
AEA16	U25			
AEA15	U26			
AEA14	T25			
AEA13	T26			
AEA12	R23			
AEA11	R24			
AEA10	P23			

† 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.)

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

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-BIT) – ADDRESS (CONTINUED)‖				
AEA9	P24	O/Z	IPD	EMIFA external address (doubleword address)
AEA8	P26			
AEA7	N23			
AEA6	N24			
AEA5	N26			
AEA4	M23			
AEA3	M24			
EMIFA (64-bit) – DATA‖				
AED63	AF24	I/O/Z	IPU	EMIFA external data
AED62	AF23			
AED61	AE23			
AED60	AE22			
AED59	AD22			
AED58	AF22			
AED57	AD21			
AED56	AE21			
AED55	AC21			
AED54	AF21			
AED53	AD20			
AED52	AE20			
AED51	AC20			
AED50	AF20			
AED49	AC19			
AED48	AD19			
AED47	W24			
AED46	W23			
AED45	Y26			
AED44	Y23			
AED43	Y25			
AED42	Y24			
AED41	AA26			
AED40	AA23			
AED39	AA25			
AED38	AA24			
AED37	AB26			
AED36	AB24			

† 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.)

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

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFA (64-bit) – DATA‖ (CONTINUED)				
AED35	AB25	I/O/Z	IPU	EMIFA external data
AED34	AC25			
AED33	AC26			
AED32	AD26			
AED31	C26			
AED30	D26			
AED29	D25			
AED28	E25			
AED27	E24			
AED26	E26			
AED25	F24			
AED24	F25			
AED23	F23			
AED22	F26			
AED21	G24			
AED20	G25			
AED19	G23			
AED18	G26			
AED17	H23			
AED16	H24			
AED15	C19			
AED14	D19			
AED13	A20			
AED12	D20			
AED11	B20			
AED10	C20			
AED9	A21			
AED8	D21			
AED7	B21			
AED6	C21			
AED5	A22			
AED4	C22			
AED3	B22			
AED2	B23			
AED1	A23			
AED0	A24			

† 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.)

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

Terminal Functions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFB (16-bit) – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY‖				
$\overline{\text{BCE3}}$	A13	O/Z	IPU	EMIFB memory space enables • Enabled by bits 26 through 31 of the word address • Only one pin is asserted during any external data access
$\overline{\text{BCE2}}$	C12	O/Z	IPU	
$\overline{\text{BCE1}}$	B12	O/Z	IPU	
$\overline{\text{BCE0}}$	A12	O/Z	IPU	
$\overline{\text{BBE1}}$	D13	O/Z	IPU	EMIFB byte-enable control • Decoded from the low-order address bits. The number of address bits or byte enables used depends on the width of external memory. • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{\text{BBE0}}$	C13	O/Z	IPU	
$\overline{\text{BPDT}}$	E12	O/Z	IPU	EMIFB peripheral data transfer, allows direct transfer between external peripherals
EMIFB (16-BIT) – BUS ARBITRATION‖				
$\overline{\text{BHOLDA}}$	E13	O	IPU	EMIFB hold-request-acknowledge to the host
$\overline{\text{BHOLD}}$	B19	I	IPU	EMIFB hold request from the host
BBUSREQ	E14	O	IPU	EMIFB bus request output
EMIFB (16-BIT) – ASYNCHRONOUS/SYNCHRONOUS MEMORY CONTROL‖				
BECLKIN	A11	I	IPD	EMIFB external input clock. The EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) is selected at reset via the pullup/pulldown resistors on the BEA[15:14] pins. BECLKIN is the default for the EMIFB input clock.
BECLKOUT2	D11	O/Z	IPD	EMIFB output clock 2. Programmable to be EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) frequency divided by 1, 2, or 4.
BECLKOUT1	D12	O/Z	IPD	EMIFB output clock 1 [at EMIFB input clock (BECLKIN, CPU/4 clock, or CPU/6 clock) frequency].
$\overline{\text{BARE}}$/ $\overline{\text{BSDCAS}}$/ $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$	A10	O/Z	IPU	EMIFB asynchronous memory read-enable/SDRAM column-address strobe/programmable synchronous interface-address strobe or read-enable • For programmable synchronous interface, the RENEN field in the CE Space Secondary Control Register (CExSEC) selects between $\overline{\text{BSADS}}$ and $\overline{\text{BSRE}}$: If RENEN = 0, then the $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$ signal functions as the $\overline{\text{BSADS}}$ signal. If RENEN = 1, then the $\overline{\text{BSADS}}$/$\overline{\text{BSRE}}$ signal functions as the $\overline{\text{BSRE}}$ signal.
$\overline{\text{BAOE}}$/ $\overline{\text{BSDRAS}}$/ $\overline{\text{BSOE}}$	B11	O/Z	IPU	EMIFB asynchronous memory output-enable/SDRAM row-address strobe/programmable synchronous interface output-enable
$\overline{\text{BAWE}}$/$\overline{\text{BSDWE}}$/ $\overline{\text{BSWE}}$	C11	O/Z	IPU	EMIFB asynchronous memory write-enable/SDRAM write-enable/programmable synchronous interface write-enable
$\overline{\text{BSOE3}}$	E15	O/Z	IPU	EMIFB synchronous memory output enable for $\overline{\text{BCE3}}$ (for glueless FIFO interface)
BARDY	E11	I	IPU	EMIFB asynchronous memory ready input

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

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

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

PRODUCT PREVIEW

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIFB (16-BIT) – ADDRESS‖				
BEA20	E16	I/O/Z	IPU	EMIFB external address (half-word address) • Also controls initialization of DSP modes at reset via pullup/pulldown resistors – Device Endian mode BEA20: 0 – Big Endian 1 – Little Endian – Boot mode BEA[19:18]: 00 – No boot 01 – HPI boot 10 – EMIFB 8-bit ROM boot with default timings (default mode) 11 – Reserved – EMIF clock select BEA[17:16]: Clock mode select for EMIFA (AECLKIN_SEL[1:0]) 00 – AECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved BEA[15:14]: Clock mode select for EMIFB (BECLKIN_SEL[1:0]) 00 – BECLKIN (default mode) 01 – CPU/4 Clock Rate 10 – CPU/6 Clock Rate 11 – Reserved
BEA19	D18		IPU	
BEA18	C18		IPD	
BEA17	B18			
BEA16	A18			
BEA15	D17			
BEA14	C17			
BEA13	B17			
BEA12	A17			
BEA11	D16			
BEA10	C16			
BEA9	B16			
BEA8	A16			
BEA7	D15			
BEA6	C15			
BEA5	B15			
BEA4	A15			
BEA3	D14			
BEA2	C14			
BEA1	A14			
EMIFB (16-bit) – DATA‖				
BED15	D7	I/O/Z	IPU	EMIFB external data
BED14	B6			
BED13	C7			
BED12	A6			
BED11	D8			
BED10	B7			
BED9	C8			
BED8	A7			
BED7	C9			
BED6	B8			
BED5	D9			
BED4	B9			
BED3	C10			
BED2	A9			
BED1	D10			
BED0	B10			

† 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.)

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

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
TIMER 2				
TOUT2	A4	O/Z	IPD	Timer 2 or general-purpose output
TINP2	C5	I	IPD	Timer 2 or general-purpose input
TIMER 1				
TOUT1	B5	O/Z	IPD	Timer 1 or general-purpose output
TINP1	A5	I	IPD	Timer 1 or general-purpose input
TIMER 0				
TOUT0	D6	O/Z	IPD	Timer 0 or general-purpose output
TINP0	C6	I	IPD	Timer 0 or general-purpose input
MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP2)				
CLKS2/GP8	AE4	I/O/Z	IPD	McBSP2 external clock source (CLKS2) [**input only**] [default] or this pin can be programmed as a GPIO 8 pin (**I/O/Z**).
CLKR2	AB1	I/O/Z	IPD	McBSP2 receive clock
CLKX2	AC2	I/O/Z	IPD	McBSP2 transmit clock
DR2	AB3	I	IPU	McBSP2 receive data. Connected to output data pin of other
DX2	AA2	O/Z	IPU	McBSP2 transmit data
FSR2	AC1	I/O/Z	IPD	McBSP2 receive frame sync
FSX2	AB2	I/O/Z	IPD	McBSP2 transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)				
CLKS1	AC8	I		McBSP1 external clock source (as opposed to internal)
CLKR1	AC10	I/O/Z		McBSP1 receive clock
CLKX1	AB12	I/O/Z		McBSP1 transmit clock
DR1	AF11	I		McBSP1 receive data
DX1	AB11	O/Z		McBSP1 transmit data
FSR1	AC9	I/O/Z		McBSP1 receive frame sync
FSX1	AB13	I/O/Z		McBSP1 transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)				
CLKS0	F4	I	IPD	McBSP0 external clock source (as opposed to internal)
CLKR0	D1	I/O/Z	IPD	McBSP0 receive clock
CLKX0	E1	I/O/Z	IPD	McBSP0 transmit clock
DR0	D2	I	IPU	McBSP0 receive data
DX0	E2	O/Z	IPU	McBSP0 transmit data
FSR0	C1	I/O/Z	IPD	McBSP0 receive frame sync
FSX0	E3	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.)

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
RESERVED FOR TEST				
RSV	F3			Reserved. This pin **must** be externally pulled up via a 10-kΩ resistor for proper device operation.
	R5			
RSV	G14			Reserved. These pins must be connected directly to CV_{DD} for proper device operation.
	H7			
	N20			
	P7			
	Y13			
RSV	R6			Reserved. This pin must be connected directly to DV_{DD} for proper device operation.
RSV	A3			Reserved (leave unconnected, ***do not*** connect to power or ground)
	G2			
	H3			
	J4			
	K6			
	N3			
	P3			
	W3			
	W25			
	AA4			
	AB14			
	AC11			
	AC12			
	AC13			
	AC14			
	AD1			
	AD7			
	AD8			
	AD9			
	AD10			
	AD11			
	AD12			
	AD13			
	AD14			
	AD15			
	AE7			
	AE8			
	AE9			

† 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.)

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
RESERVED FOR TEST				
RSV	AE10			Reserved (leave unconnected, ***do not*** connect to power or ground)
	AE11			
	AE12			
	AE15			
	AF3			
	AF7			
	AF9			
	AF10			
	AF12			
	AF13			
	AF14			

† 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.)

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS			
DV_{DD}	A2	S	3.3-V supply voltage
	A25		
	B1		
	B14		
	B26		
	E7		
	E8		
	E10		
	E17		
	E19		
	E20		
	F9		
	F12		
	F15		
	F18		
	G5		
	G22		
	H5		
	H22		
	J21		
	K5		
	K22		
	L5		
	M5		
	M6		
	M21		
	N2		
	P25		
	R21		
	T5		
	U5		
	U22		
	V6		
	V21		
	W5		
	W22		
	Y5		
	Y22		

† 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}	AA9	S	3.3-V supply voltage
	AA12		
	AA15		
	AA18		
	AB7		
	AB8		
	AB10		
	AB17		
	AB19		
	AB20		
	AE1		
	AE13		
	AE26		
	AF2		
	AF25		
CV_{DD}	A1	S	1.2-V supply voltage (-400, -500 device speeds) 1.4-V supply voltage (-600 device speed)
	A26		
	B2		
	B25		
	C3		
	C24		
	D4		
	D23		
	E5		
	E22		
	F6		
	F7		
	F20		
	F21		
	G6		
	G7		
	G8		
	G10		
	G11		
	G13		
	G16		
	G17		
	G19		
	G20		

† 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}	G21	S	1.2-V supply voltage (-400, -500 device speeds) 1.4-V supply voltage (-600 device speed)
	H20		
	K7		
	K20		
	L7		
	L20		
	N7		
	P20		
	T7		
	T20		
	U7		
	U20		
	W7		
	W20		
	Y6		
	Y7		
	Y8		
	Y10		
	Y11		
	Y14		
	Y16		
	Y17		
	Y19		
	Y20		
	Y21		
	AA6		
	AA7		
	AA20		
	AA21		
	AB5		
	AB22		
	AC4		
	AC23		
	AD3		
	AD24		
	AE2		
	AE25		
	AF1		
	AF26		

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

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS			
V_{SS}	A8	GND	Ground pins
	A19		
	B3		
	B13		
	B24		
	C2		
	C4		
	C23		
	C25		
	D3		
	D5		
	D22		
	D24		
	E4		
	E6		
	E9		
	E18		
	E21		
	E23		
	F5		
	F8		
	F10		
	F11		
	F13		
	F14		
	F16		
	F17		
	F19		
	F22		
	G9		
	G12		
	G15		
	G18		
	H1		
	H6		
	H21		
	H26		
	J5		
	J7		

† I = Input, O = Output, Z = High iR7mpedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	J20	GND	Ground pins
	J22		
	K21		
	L6		
	L21		
	M7		
	M20		
	N6		
	N21		
	N25		
	P2		
	P6		
	P21		
	R7		
	R20		
	T6		
	T21		
	U6		
	U21		
	V5		
	V7		
	V20		
	V22		
	W1		
	W6		
	W21		
	W26		
	Y9		
	Y12		
	Y15		
	Y18		
	AA5		
	AA8		
	AA10		
	AA11		
	AA13		
	AA14		
	AA16		
	AA17		

† I = Input, O = Output, Z = High iR7mpedance, S = Supply voltage, GND = Ground

Terminal Functions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	AA19	GND	Ground pins
	AA22		
	AB4		
	AB6		
	AB9		
	AB18		
	AB21		
	AB23		
	AC3		
	AC5		
	AC22		
	AC24		
	AD2		
	AD4		
	AD23		
	AD25		
	AE3		
	AE14		
	AE24		
	AF8		
	AF19		

† I = Input, O = Output, Z = High iR7mpedance, 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 device-specific tools, select “Digital Signal Processors”, choose a product family, and select the particular DSP device. 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."

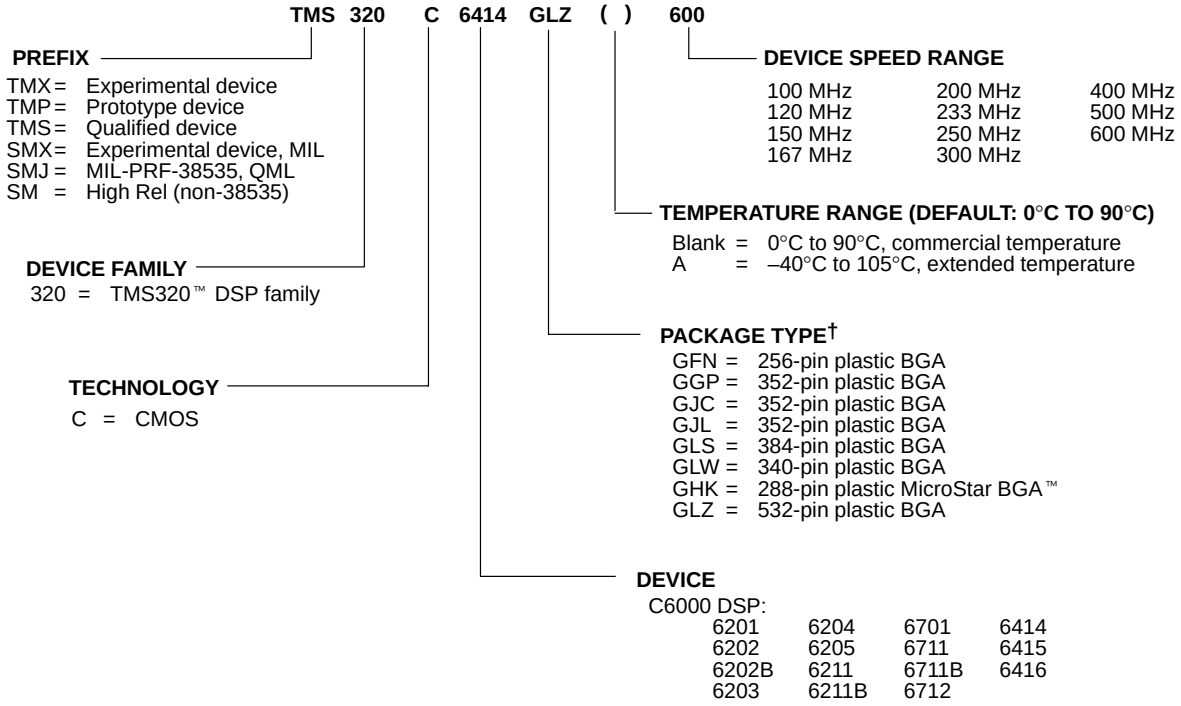
TMS devices and TMDS development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

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, -600 is 600 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 TMS320C6414 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), 32-/16-bit host-port interfaces (HPIs), 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 *TMS320C6415 Fixed-Point Digital Signal Processor* data sheet (literature number SPRS146) describes the features of the TMS320C6415 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 the *How To Begin Development Today with the TMS320C6414, TMS320C6415, and TMS320C6416 DSPs* application report (literature number SPRA718), which describes in more detail the compatibility and similarities/differences among the C6414, C6415, C6416, and C6211 devices.

PRODUCT PREVIEW

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.

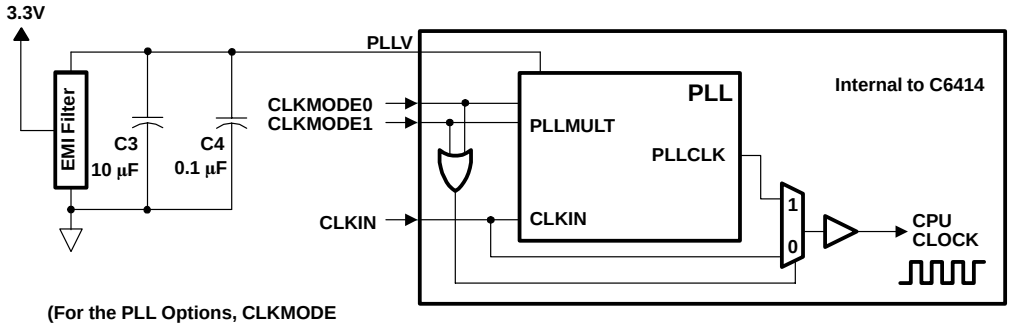
To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 5 shows the external PLL circuitry for either x1 (PLL bypass) or other PLL multiply modes.

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 5 lists some examples of compatible CLKIN external clock sources:

Table 5. 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



(For the PLL Options, CLKMODE Pins Setup, and PLL Clock Frequency Ranges see Table 6.)

NOTES: A. Place all PLL external components (C3, C4, 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 (C3, C4, 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 6. TMS320C6414 PLL Multiply Factor Options, Clock Frequency Ranges, and Typical Lock Time†‡

GLZ PACKAGE – 23 x 23 mm BGA							
CLKMODE1	CLKMODE0	CLKMODE (PLL MULTIPLY FACTORS)	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT4 RANGE (MHz)	CLKOUT6 RANGE (MHz)	TYPICAL LOCK TIME (μs)§
0	0	Bypass (x1)	30–75	30–75	7.5–18.8	5–12.5	N/A
0	1	x6	30–75	180–450	45–112.5	30–75	75
1	0	x12	30–50	360–600	90–150	60–100	
1	1	Reserved	–	–	–	–	–

† These clock frequency range values are applicable to a C6414–600 speed device. For –400 and –500 device speed values, see the CLKIN timing requirements table for the specific device speed.

‡ Use external pullup resistors on the CLKMODE pins (CLKMODE1 and CLKMODE0) to set the C6414 device to one of the valid PLL multiply clock modes (x6 or x12). With internal pulldown resistors on the CLKMODE pins (CLKMODE1, CLKMODE0), the default clock mode is x1 (bypass).

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

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 range, CV_{DD} (see Note 1) . . . . . – 0.3 V to 2.3 V
Supply voltage range, DV_{DD} (see Note 1) . . . . . –0.3 V to 4 V
Input voltage range . . . . . –0.3 V to 4 V
Output voltage range . . . . . –0.3 V to 4 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 (-400, -500 device speeds)‡		1.16	1.2	1.24	V
	Supply voltage, Core (-600 device speed)‡		1.36	1.4	1.44	
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		2			V
V_{IL}	Low-level input voltage				0.8	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 C641x DSPs may operate at voltages ranging from 1.2 V to 1.4 V to provide a range of system power/performance options. TI highly recommends that users design-in a supply that can handle multiple voltages within this range (i.e., 1.2 V, 1.25 V, 1.3 V, 1.35 V, 1.4 V with ±3% tolerances) by implementing simple board changes such as reference resistor values or input pin configuration modifications. Examples of such supplies include the PT4660, PT5500, PT5520, PT6440, and PT6930 series from Power Trends, a subsidiary of Texas Instruments. Not incorporating a flexible supply may limit the system's ability to easily adapt to future versions of C641x devices.

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

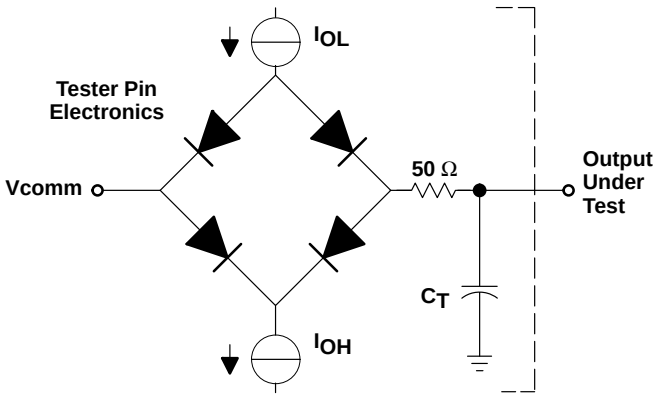
PARAMETER		TEST CONDITIONS‡	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	DV_{DD} = MIN, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN, I_{OL} = MAX			0.4	V
I_I	Input current	V_I = V_{SS} to DV_{DD}			±150	uA
I_{OZ}	Off-state output current	V_O = DV_{DD} or 0 V			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access§	CV_{DD} = NOM, CPU clock = 400 MHz		TBD		mA
I_{DD2V}	Supply current, peripherals§	CV_{DD} = NOM, CPU clock = 400 MHz		TBD		mA
I_{DD3V}	Supply current, I/O pins§	DV_{DD} = NOM, CPU clock = 400 MHz		TBD		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.

§ Measured with average activity (50% high/50% low power). For more details on CPU, peripheral, and I/O activity, refer to the *TMS320C6000 Power Consumption Summary* application report (literature number SPRA486).

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 2 mA
I_{OH} = 2 mA
V_{comm} = 0.8 V
C_T = 10–15-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.

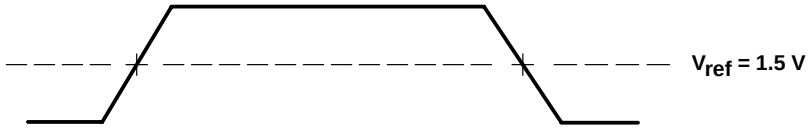


Figure 7. Input and Output Voltage Reference Levels for ac Timing Measurements

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

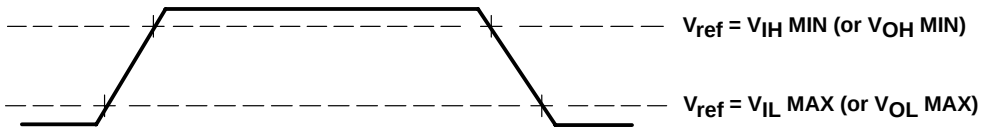


Figure 8. Rise and Fall Transition Time Voltage Reference Levels

PARAMETER MEASUREMENT INFORMATION (CONTINUED)

timing parameters and board routing analysis

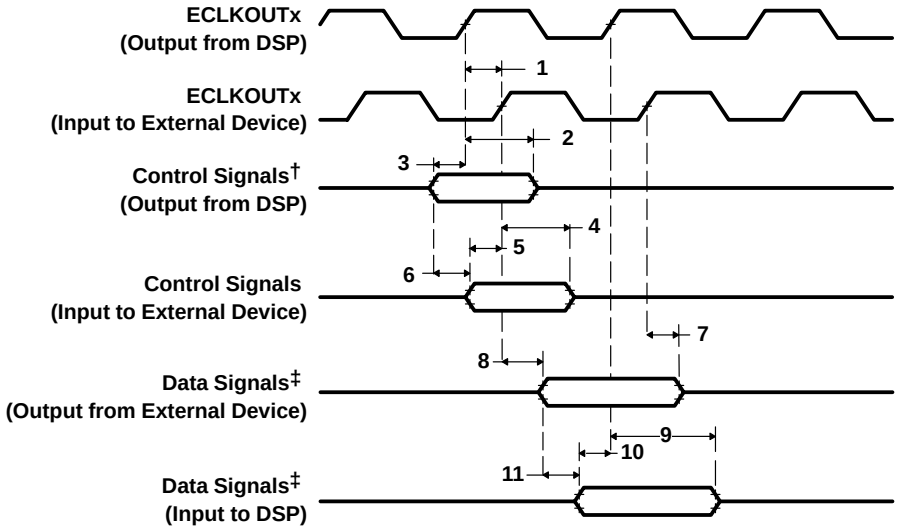
The timing parameter values specified in this data sheet do *not* include delays by board routings. As a good board design practice, such delays must *always* be taken into account. Timing values may be adjusted by increasing/decreasing such delays. TI recommends utilizing the available I/O buffer information 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 7 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 7. 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 for –400 speed devices†‡§ (see Figure 10)

NO.			–400						UNIT
			PLL MODE x12		PLL MODE x6		x1 (BYPASS)		
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	30	33.3	15	33.3	13.3	33.3	ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.4C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.4C		0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		5		1	ns

† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ For more details on the PLL multiplier factors (x6, x12), see the *Clock PLL* section of this data sheet.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

timing requirements for CLKIN for –500 speed devices†‡§ (see Figure 10)

NO.			–500						UNIT
			PLL MODE x12		PLL MODE x6		x1 (BYPASS)		
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	24	33.3	13.3	33.3	13.3	33.3	ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.4C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.4C		0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		5		1	ns

† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ For more details on the PLL multiplier factors (x6, x12), see the *Clock PLL* section of this data sheet.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

timing requirements for CLKIN for –600 speed devices†‡§ (see Figure 10)

NO.			–600						UNIT
			PLL MODE x12		PLL MODE x6		x1 (BYPASS)		
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	20	33.3	13.3	33.3	13.3	33.3	ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.4C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.4C		0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		5		1	ns

† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ For more details on the PLL multiplier factors (x6, x12), see the *Clock PLL* section of this data sheet.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

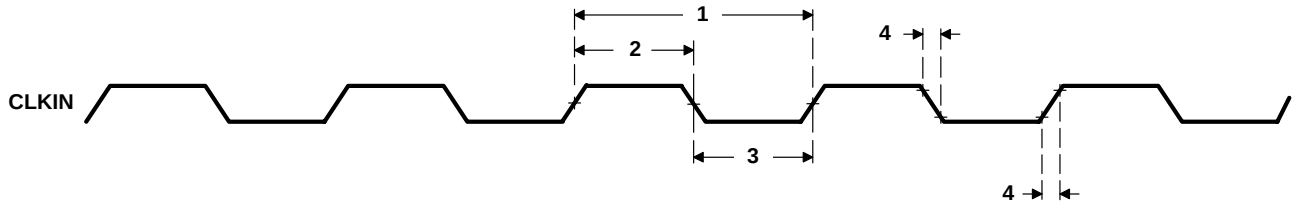


Figure 10. CLKIN Timing

INPUT AND OUTPUT CLOCKS (CONTINUED)

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

NO.	PARAMETER		–400 –500 –600		UNIT
			CLKMODE = x1, x6, x12		
			MIN	MAX	
1	$t_{c(CKO4)}$	Cycle time, CLKOUT4	4P – 0.7	4P + 0.7	ns
2	$t_{w(CKO4H)}$	Pulse duration, CLKOUT4 high	2P – 0.7	2P + 0.7	ns
3	$t_{w(CKO4L)}$	Pulse duration, CLKOUT4 low	2P – 0.7	2P + 0.7	ns
4	$t_{t(CKO4)}$	Transition time, CLKOUT4		1	ns

† 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)

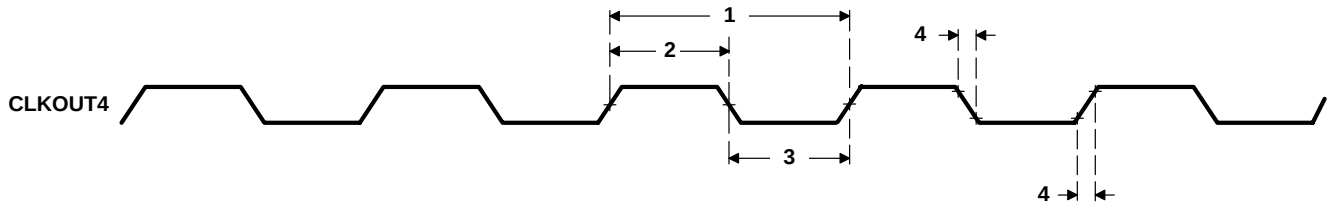


Figure 11. CLKOUT4 Timing

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

NO.	PARAMETER		–400 –500 –600		UNIT
			CLKMODE = x1, x6, x12		
			MIN	MAX	
1	$t_{c(CKO6)}$	Cycle time, CLKOUT6	6P – 0.7	6P + 0.7	ns
2	$t_{w(CKO6H)}$	Pulse duration, CLKOUT6 high	3P – 0.7	3P + 0.7	ns
3	$t_{w(CKO6L)}$	Pulse duration, CLKOUT6 low	3P – 0.7	3P + 0.7	ns
4	$t_{t(CKO6)}$	Transition time, CLKOUT6		1	ns

† 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)

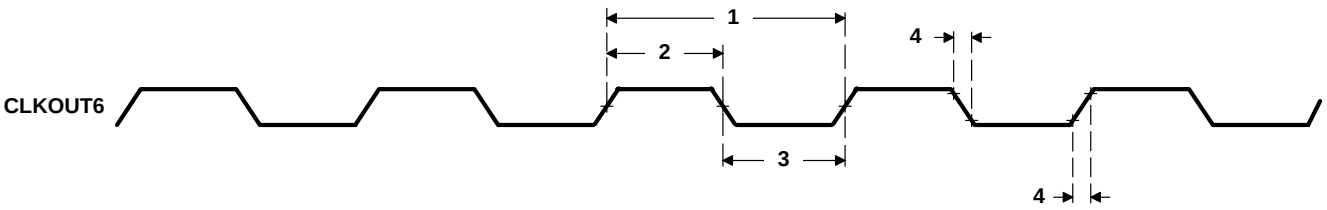


Figure 12. CLKOUT6 Timing

INPUT AND OUTPUT CLOCKS (CONTINUED)

timing requirements for ECLKIN for EMIFA and EMIFB†‡§ (see Figure 13)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{c(EKI)}$	Cycle time, ECLKIN	7.5	16P	ns
2	$t_{w(EKIH)}$	Pulse duration, ECLKIN high	3.38		ns
3	$t_{w(EKIL)}$	Pulse duration, ECLKIN low	3.38		ns
4	$t_{t(EKI)}$	Transition time, ECLKIN		2	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
§ The C64x™ has two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.

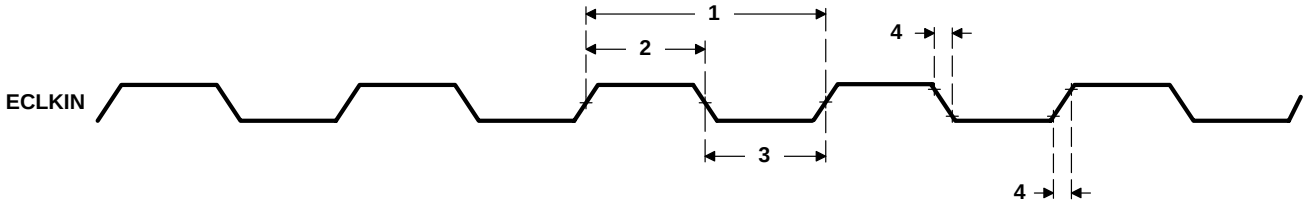


Figure 13. ECLKIN Timing for EMIFA and EMIFB

switching characteristics over recommended operating conditions for ECLKOUT1 for EMIFA and EMIFB modules§¶#‖ (see Figure 14)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{c(EKO1)}$	Cycle time, ECLKOUT1	E – 0.7	E + 0.7	ns
2	$t_{w(EKO1H)}$	Pulse duration, ECLKOUT1 high	EH – 0.7	EH + 0.7	ns
3	$t_{w(EKO1L)}$	Pulse duration, ECLKOUT1 low	EL – 0.7	EL + 0.7	ns
4	$t_{t(EKO1)}$	Transition time, ECLKOUT1		1	ns
5	$t_{d(EKIH-EKO1H)}$	Delay time, ECLKIN high to ECLKOUT1 high	3	8	ns
6	$t_{d(EKIL-EKO1L)}$	Delay time, ECLKIN low to ECLKOUT1 low	3	8	ns

§ The C64x™ has two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.
¶ The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
# E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
‖ EH is the high period of E (EMIF input clock period) in ns and EL is the low period of E (EMIF input clock period) in ns for EMIFA or EMIFB.

INPUT AND OUTPUT CLOCKS (CONTINUED)

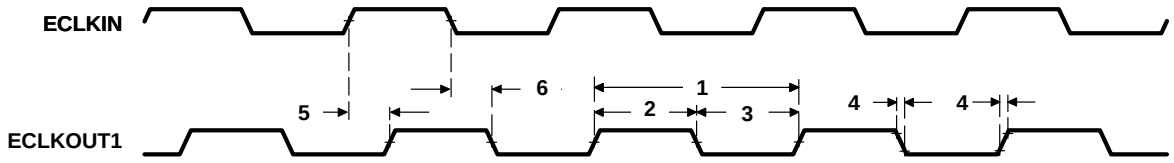


Figure 14. ECLKOUT1 Timing for EMIFA and EMIFB Modules

switching characteristics over recommended operating conditions for ECLKOUT2 for the EMIFA and EMIFB modules†‡§ (see Figure 15)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{c(EKO2)}$	Cycle time, ECLKOUT2	NE – 0.7	NE + 0.7	ns
2	$t_{w(EKO2H)}$	Pulse duration, ECLKOUT2 high	0.5NE – 0.7	0.5NE + 0.7	ns
3	$t_{w(EKO2L)}$	Pulse duration, ECLKOUT2 low	0.5NE – 0.7	0.5NE + 0.7	ns
4	$t_{t(EKO2)}$	Transition time, ECLKOUT2		1	ns
5	$t_{d(EKIH-EKO2H)}$	Delay time, ECLKIN high to ECLKOUT2 high	3	8	ns
6	$t_{d(EKIH-EKO2L)}$	Delay time, ECLKIN high to ECLKOUT2 low	3	8	ns

† The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
‡ The C64x™ has two EMIFs (64-bit EMIFA and 16-bit EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted.
§ E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
N = the EMIF input clock divider; N = 1, 2, or 4.

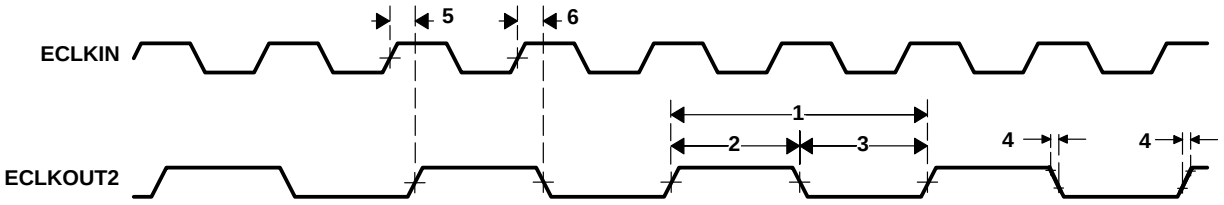


Figure 15. ECLKOUT2 Timing for the EMIFA and EMIFB Modules

PRODUCT PREVIEW

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles for EMIFA and EMIFB modules[†‡§] (see Figure 16 and Figure 17)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{ARE}$ high	6		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{ARE}$ high	1		ns
6	$t_{su(ARDY-EKO1H)}$	Setup time, ARDY valid before ECLKOUT1 high	3		ns
7	$t_{h(EKO1H-ARDY)}$	Hold time, ARDY valid after ECLKOUT1 high	1		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.

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

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

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS * E – 1.5		ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	RH * E – 1.5		ns
5	$t_{d(EKO1H-AREV)}$	Delay time, ECLKOUT1 high to $\overline{ARE}$ vaild	1.5	5	ns
8	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS * E – 1.5		ns
9	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	WH * E – 1.5		ns
10	$t_{d(EKO1H-AWEV)}$	Delay time, ECLKOUT1 high to $\overline{AWE}$ vaild	1.5	5	ns
11	$t_{osu(PDTV-AREL)}$	Output setup time, $\overline{PDT}$ valid to $\overline{ARE}$ low	RS * E – 1.5		ns
12	$t_{oh(AREH-PDTIV)}$	Output hold time, $\overline{ARE}$ high to $\overline{PDT}$ invalid	RH * E – 1.5		ns
13	$t_{osu(PDTV-AWEV)}$	Output setup time, $\overline{PDT}$ valid to $\overline{AWE}$ valid	WS * E – 1.5		ns
14	$t_{oh(AWEH-PDTIV)}$	Output hold time, $\overline{AWE}$ high to $\overline{PDT}$ invalid	WS * E – 1.5		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.

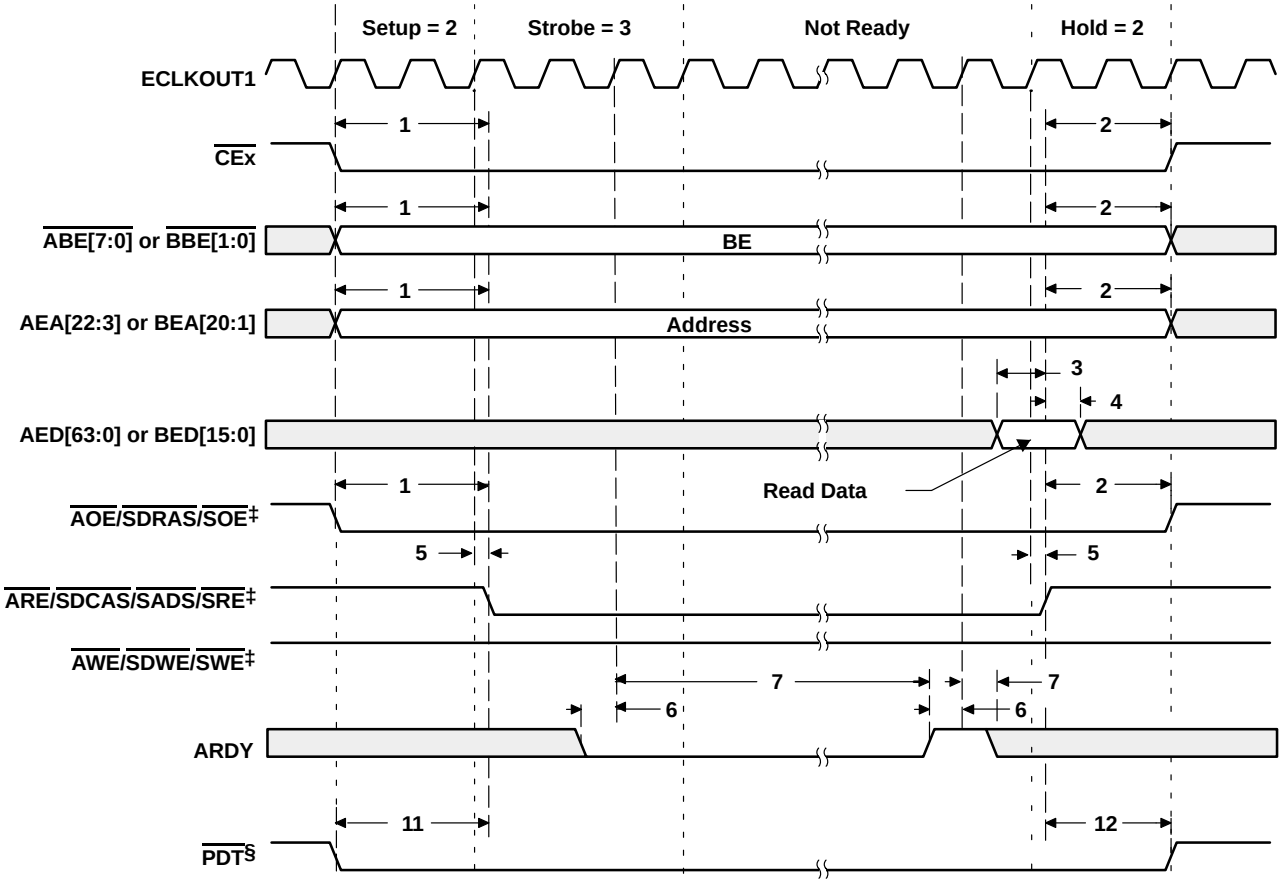
§ The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the asynchronous memory access signals are shown as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

¶ E = ECLKOUT1 period in ns for EMIFA or EMIFB

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

PRODUCT PREVIEW

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



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

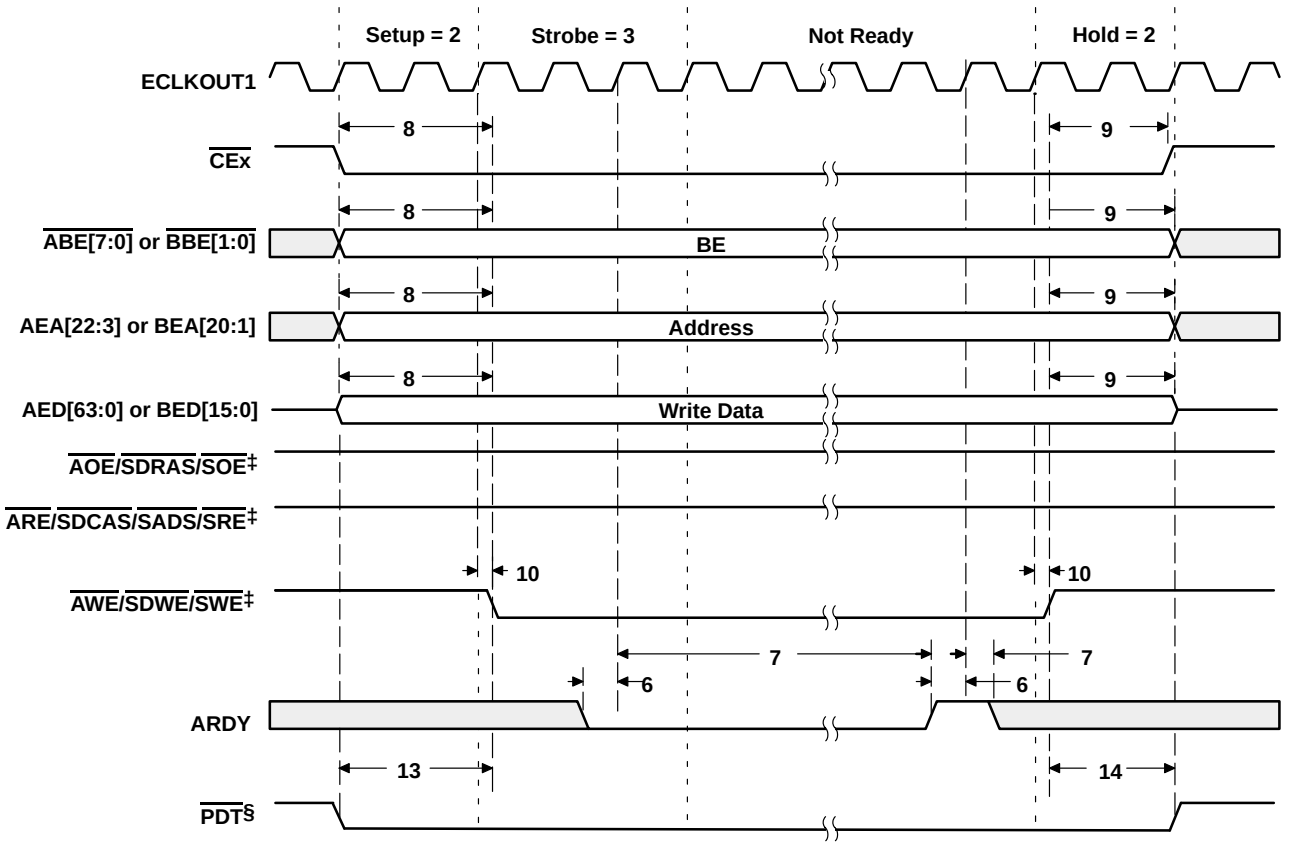
‡ AOE/SDRAS/SOE, ARE/SDCAS/SADS/SRE, and AWE/SDWE/SWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

§ 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 for EMIFA and EMIFB†

PRODUCT PREVIEW

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



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

‡ 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 for EMIFA and EMIFB†

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING

timing requirements for programmable synchronous interface cycles for EMIFA and EMIFB modules† (see Figure 18)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
6	$t_{su(EDV-EKOxH)}$	Setup time, read EDx valid before ECLKOUTx high	2		ns
7	$t_{h(EKOxH-EDV)}$	Hold time, read EDx valid after ECLKOUTx high	1.5		ns

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

switching characteristics over recommended operating conditions for programmable synchronous interface cycles for EMIFA and EMIFB modules†‡ (see Figure 18–Figure 20)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{d(EKOxH-CEV)}$	Delay time, ECLKOUTx high to $\overline{CEx}$ valid	1	5	ns
2	$t_{d(EKOxH-BEV)}$	Delay time, ECLKOUTx high to $\overline{BEx}$ valid		5	ns
3	$t_{d(EKOxH-BEIV)}$	Delay time, ECLKOUTx high to $\overline{BEx}$ invalid	1		ns
4	$t_{d(EKOxH-EAV)}$	Delay time, ECLKOUTx high to EAx valid		5	ns
5	$t_{d(EKOxH-EAIV)}$	Delay time, ECLKOUTx high to EAx invalid	1		ns
8	$t_{d(EKOxH-ADSV)}$	Delay time, ECLKOUTx high to $\overline{SADS}/\overline{SRE}$ valid	1	5	ns
9	$t_{d(EKOxH-OEV)}$	Delay time, ECLKOUTx high to, $\overline{SOE}$ valid	1	5	ns
10	$t_{d(EKOxH-EDV)}$	Delay time, ECLKOUTx high to $\overline{EDx}$ valid		5	ns
11	$t_{d(EKOxH-EDIV)}$	Delay time, ECLKOUTx high to $\overline{EDx}$ invalid	1		ns
12	$t_{d(EKOxH-WEV)}$	Delay time, ECLKOUTx high to $\overline{SWE}$ valid	1	5	ns
13	$t_{d(EKOxH-PDTV)}$	Delay time, ECLKOUTx high to $\overline{PDT}$ valid	1	5	ns

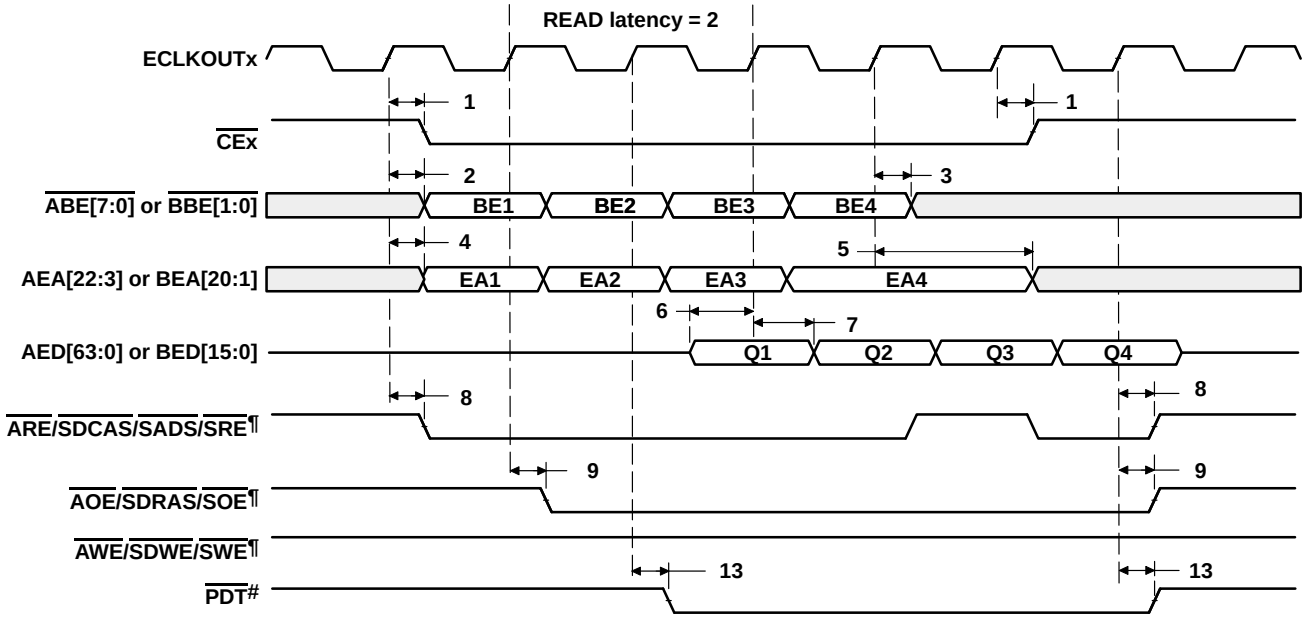
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the programmable synchronous interface access signals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

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

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

PRODUCT PREVIEW

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



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

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

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

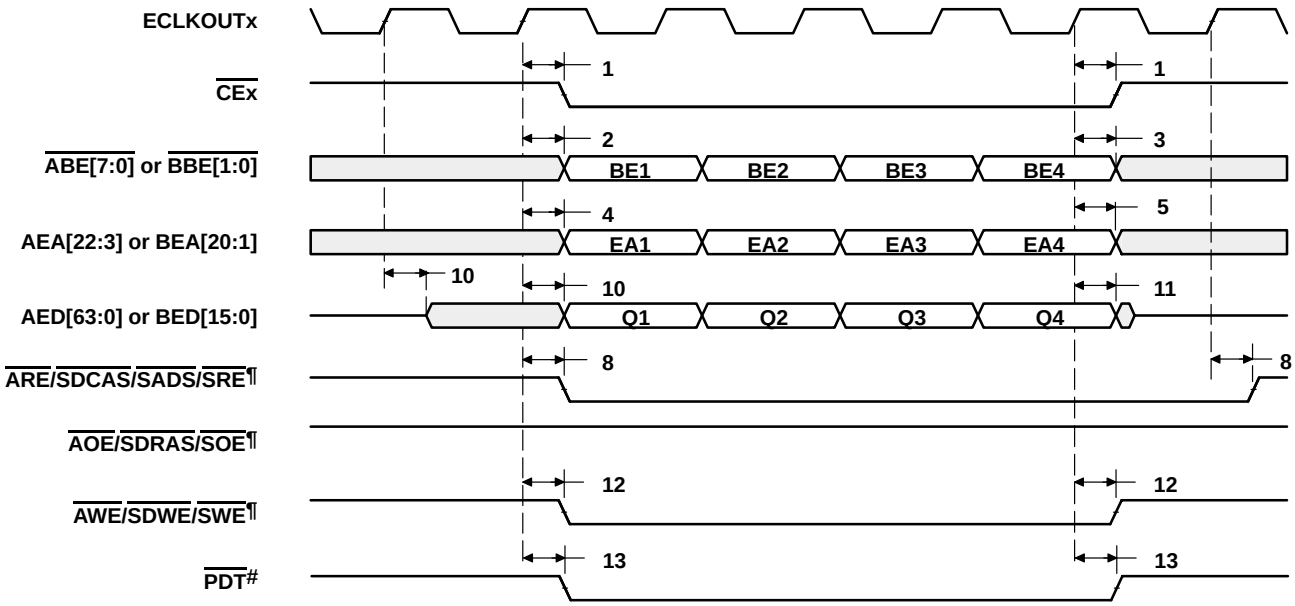
- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- CEx assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, CEx goes inactive after the final command has been issued (CEEXT = 0). For synchronous FIFO interface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

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

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

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



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

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

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

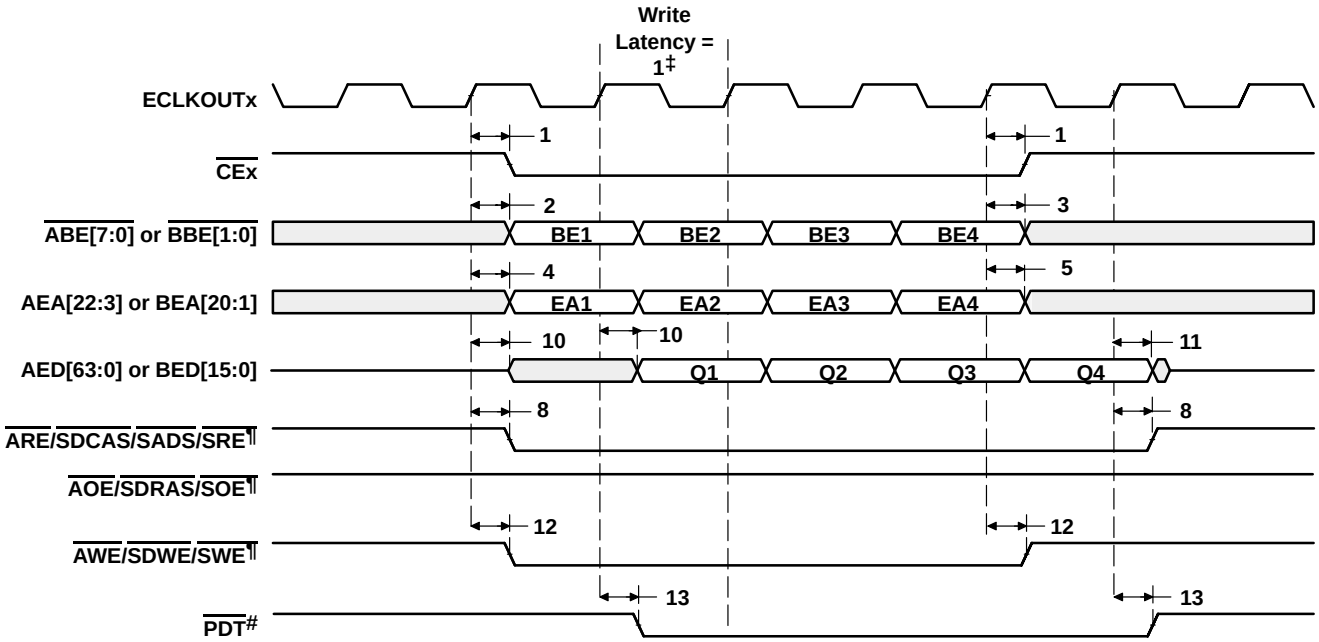
- Read latency (SYNCRL): 0-, 1-, 2-, or 3-cycle read latency
- Write latency (SYNCWL): 0-, 1-, 2-, or 3-cycle write latency
- $\overline{\text{CEx}}$ assertion length (CEEXT): For standard SBSRAM or ZBT SRAM interface, $\overline{\text{CEx}}$ goes inactive after the final command has been issued (CEEXT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

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

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

Figure 19. Programmable Synchronous Interface Write Timing for EMIFA and EMIFB (With Write Latency = 0)†‡§

PROGRAMMABLE SYNCHRONOUS INTERFACE TIMING (CONTINUED)



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

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

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

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

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

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

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles for EMIFA and EMIFB modules† (see Figure 21)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
6	$t_{su(EDV-EKO1H)}$	Setup time, read EDx valid before ECLKOUT1 high	0.5		ns
7	$t_{h(EKO1H-EDV)}$	Hold time, read EDx valid after ECLKOUT1 high	2		ns

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

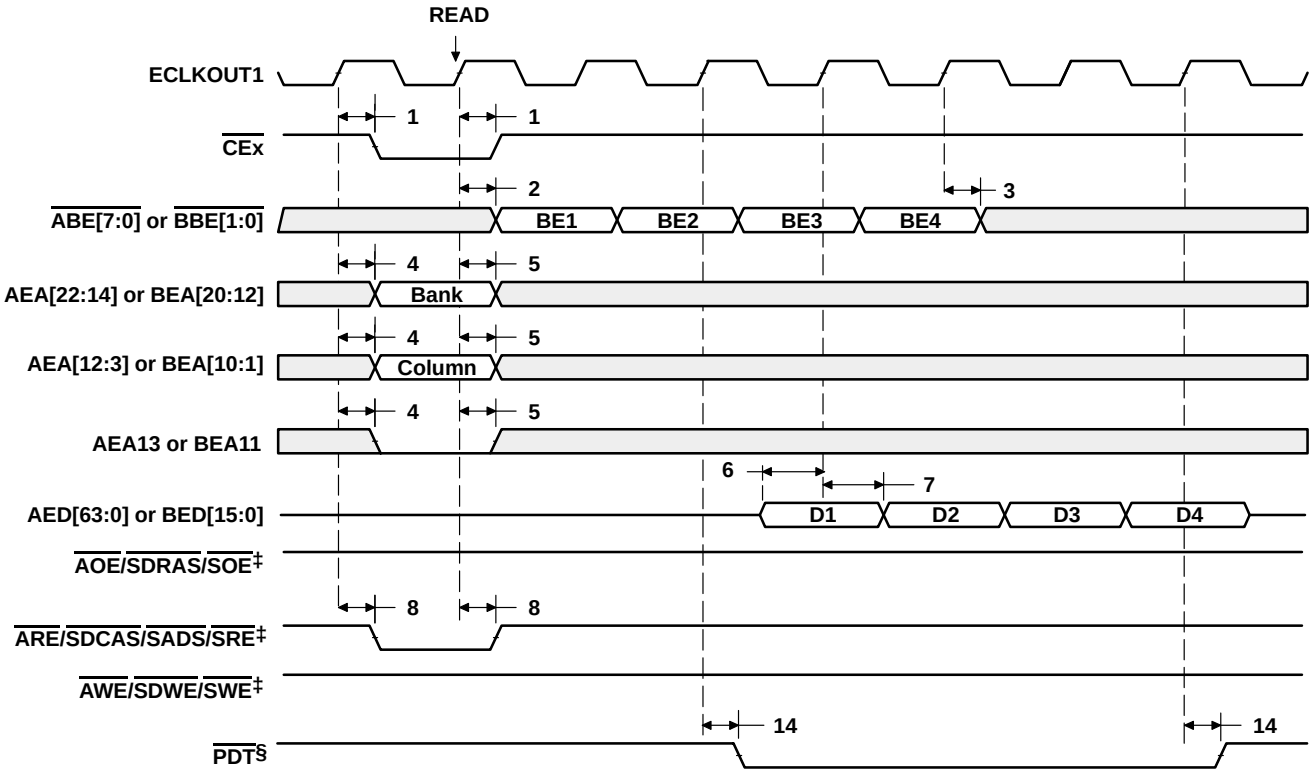
switching characteristics over recommended operating conditions for synchronous DRAM cycles for EMIFA and EMIFB modules† (see Figure 21–Figure 28)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{d(EKO1H-CEV)}$	Delay time, ECLKOUT1 high to $\overline{\text{CEx}}$ valid	1	5	ns
2	$t_{d(EKO1H-BEV)}$	Delay time, ECLKOUT1 high to $\overline{\text{BEx}}$ valid		5	ns
3	$t_{d(EKO1H-BEIV)}$	Delay time, ECLKOUT1 high to $\overline{\text{BEx}}$ invalid	1		ns
4	$t_{d(EKO1H-EAV)}$	Delay time, ECLKOUT1 high to EAx valid		5	ns
5	$t_{d(EKO1H-EAIV)}$	Delay time, ECLKOUT1 high to EAx invalid	1		ns
8	$t_{d(EKO1H-CASV)}$	Delay time, ECLKOUT1 high to $\overline{\text{SDCAS}}$ valid	1	5	ns
9	$t_{d(EKO1H-EDV)}$	Delay time, ECLKOUT1 high to $\overline{\text{EDx}}$ valid		5	ns
10	$t_{d(EKO1H-EDIV)}$	Delay time, ECLKOUT1 high to $\overline{\text{EDx}}$ invalid	1		ns
11	$t_{d(EKO1H-WEV)}$	Delay time, ECLKOUT1 high to $\overline{\text{SDWE}}$ valid	1	5	ns
12	$t_{d(EKO1H-RAS)}$	Delay time, ECLKOUT1 high to $\overline{\text{SDRAS}}$ valid	1	5	ns
13	$t_{d(EKO1H-ACKEV)}$	Delay time, ECLKOUT1 high to ASDCKE valid (EMIFA only)	1	5	ns
14	$t_{d(EKO1H-PDTV)}$	Delay time, ECLKOUT1 high to $\overline{\text{PDT}}$ valid	1	5	ns

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

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

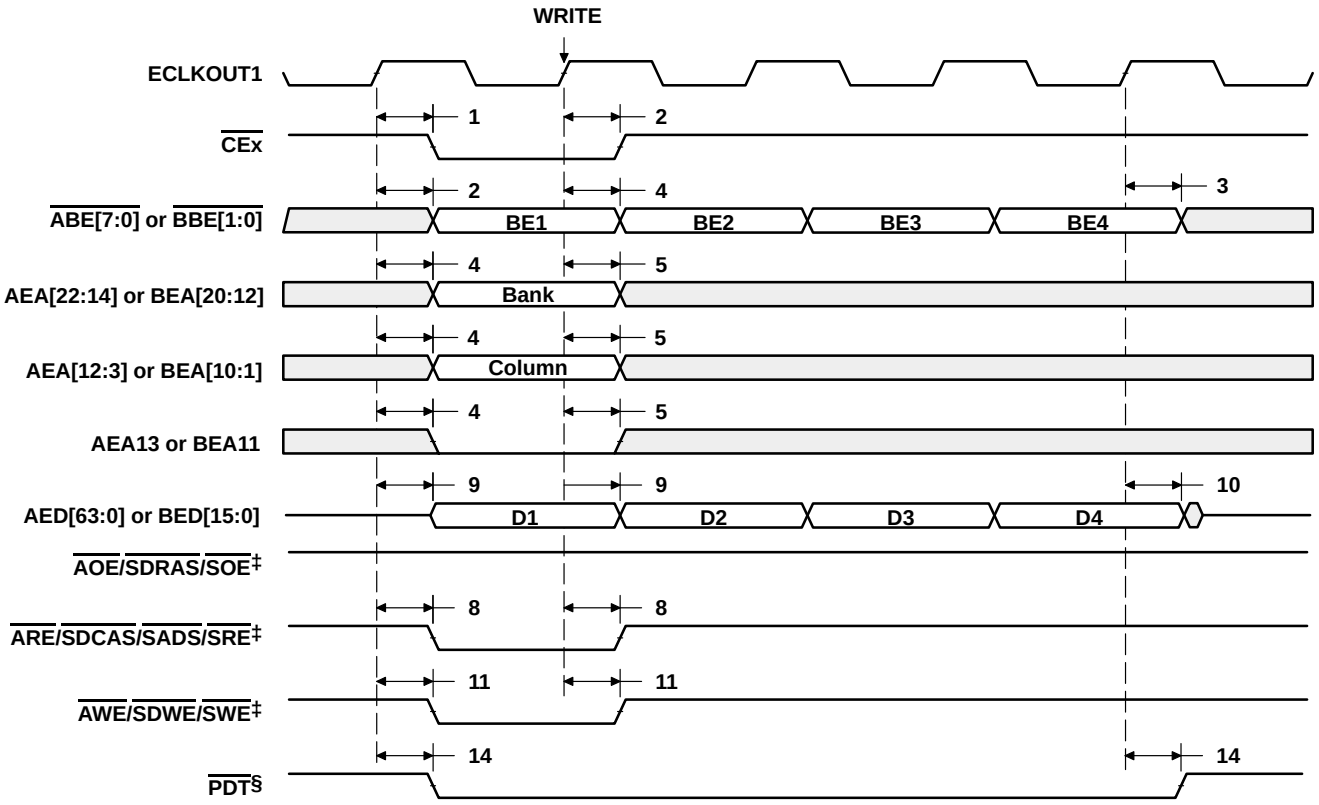
‡ $\overline{\text{ARE}}/\overline{\text{SDCAS}}/\overline{\text{SADS}}/\overline{\text{SRE}}$, $\overline{\text{AWE}}/\overline{\text{SDWE}}/\overline{\text{SWE}}$, and $\overline{\text{AOE}}/\overline{\text{SDRAS}}/\overline{\text{SOE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$,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

§ $\overline{\text{PDT}}$ 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) for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

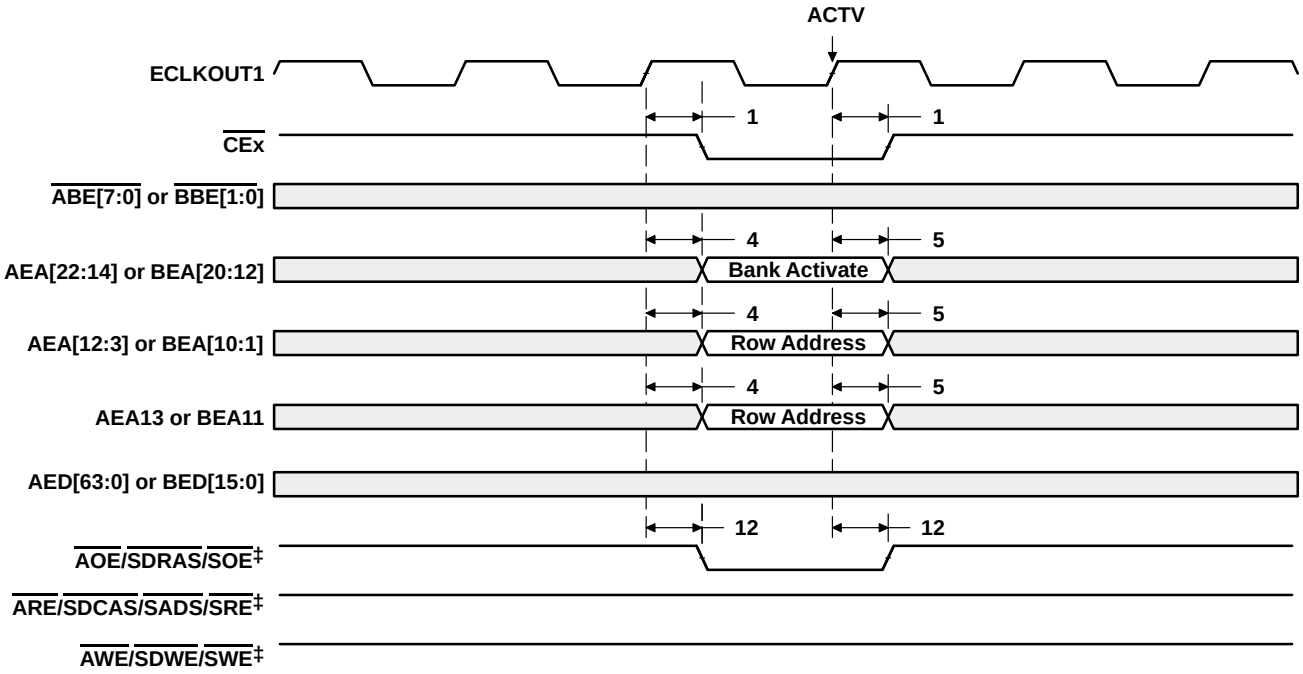
‡ ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

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

Figure 22. SDRAM Write Command for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



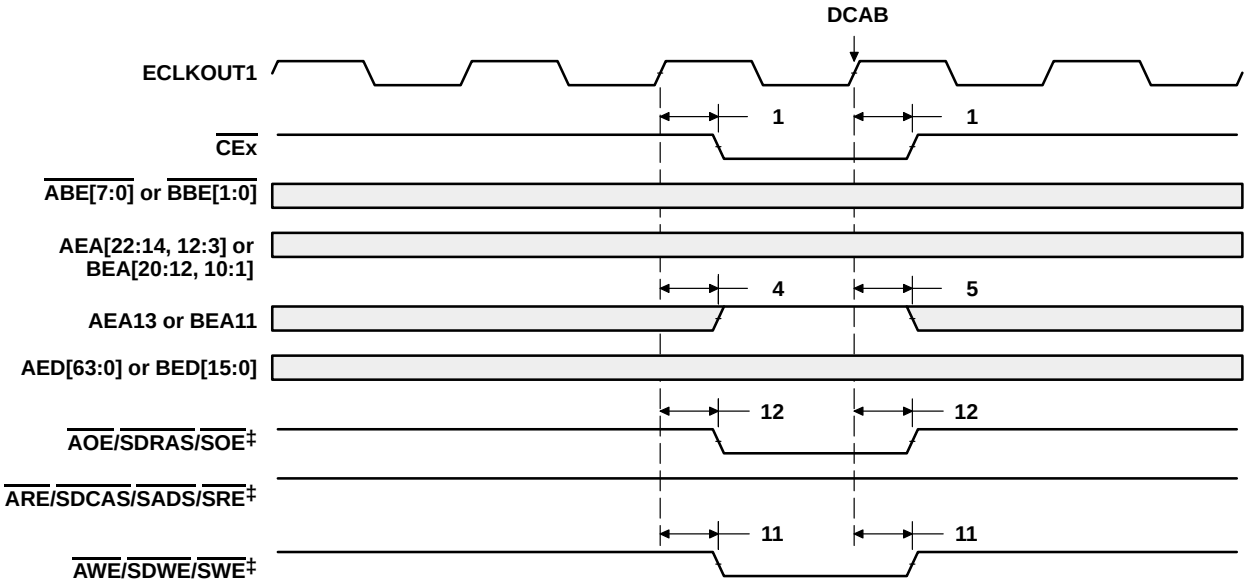
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an “A” and all EMIFB signals are prefixed by a “B”. Throughout the rest of this document, in generic EMIF areas of discussion, the prefix “A” or “B” may be omitted [e.g., the synchronous DRAM memory access signals are shown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

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

Figure 23. SDRAM ACTV Command for EMIFA and EMFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



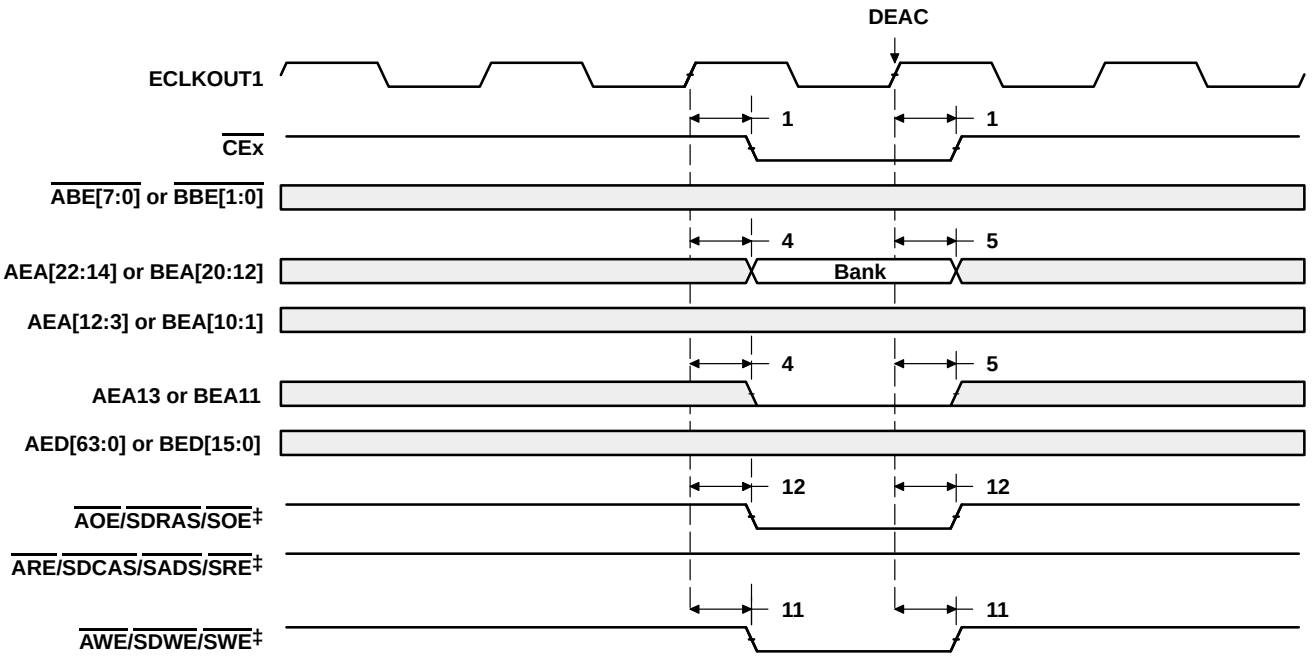
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the synchronous DRAM memory access signals are shown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

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

Figure 24. SDRAM DCAB Command for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



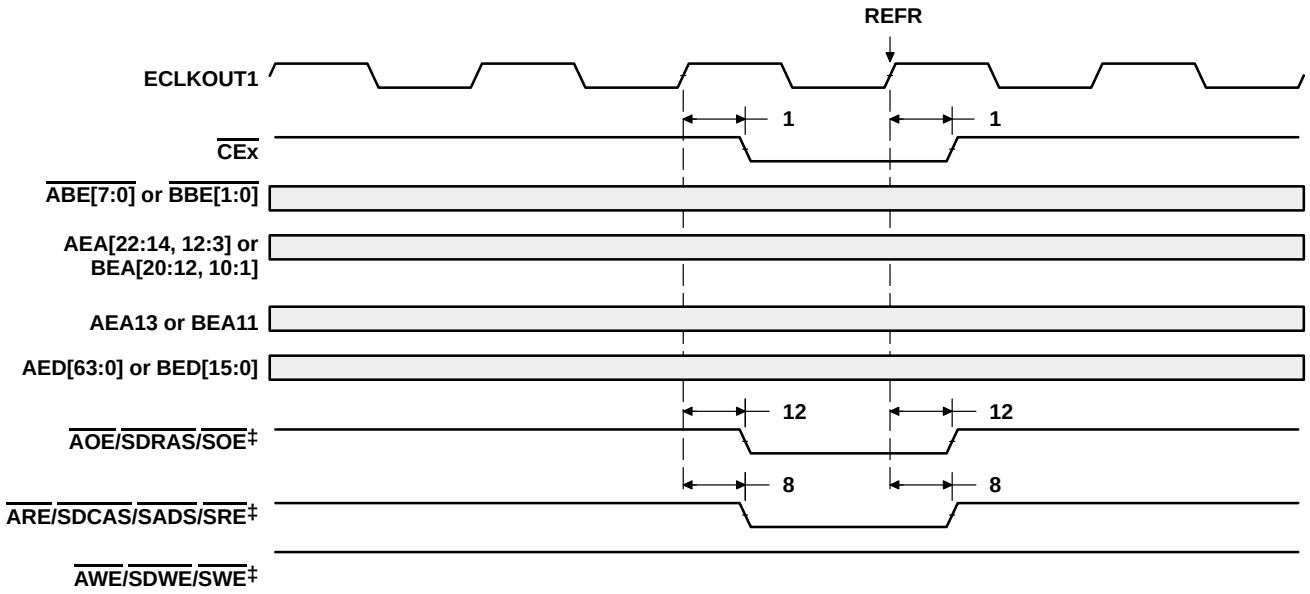
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the synchronous DRAM memory access signals are shown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

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

Figure 25. SDRAM DEAC Command for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



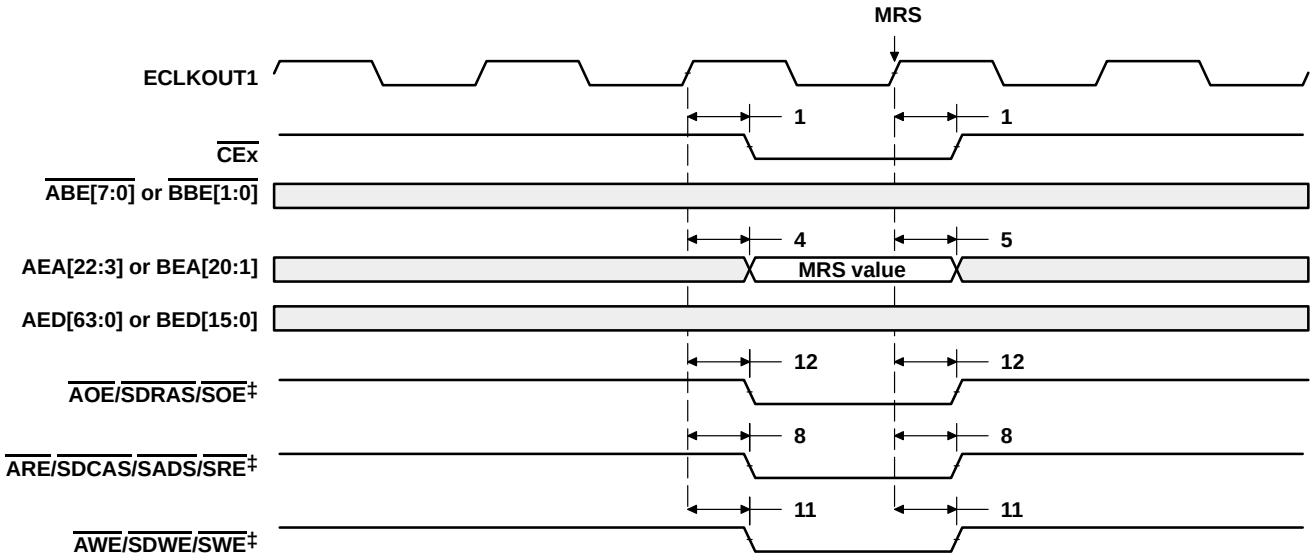
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the synchronous DRAM memory access signals are shown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

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

Figure 26. SDRAM REFR Command for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



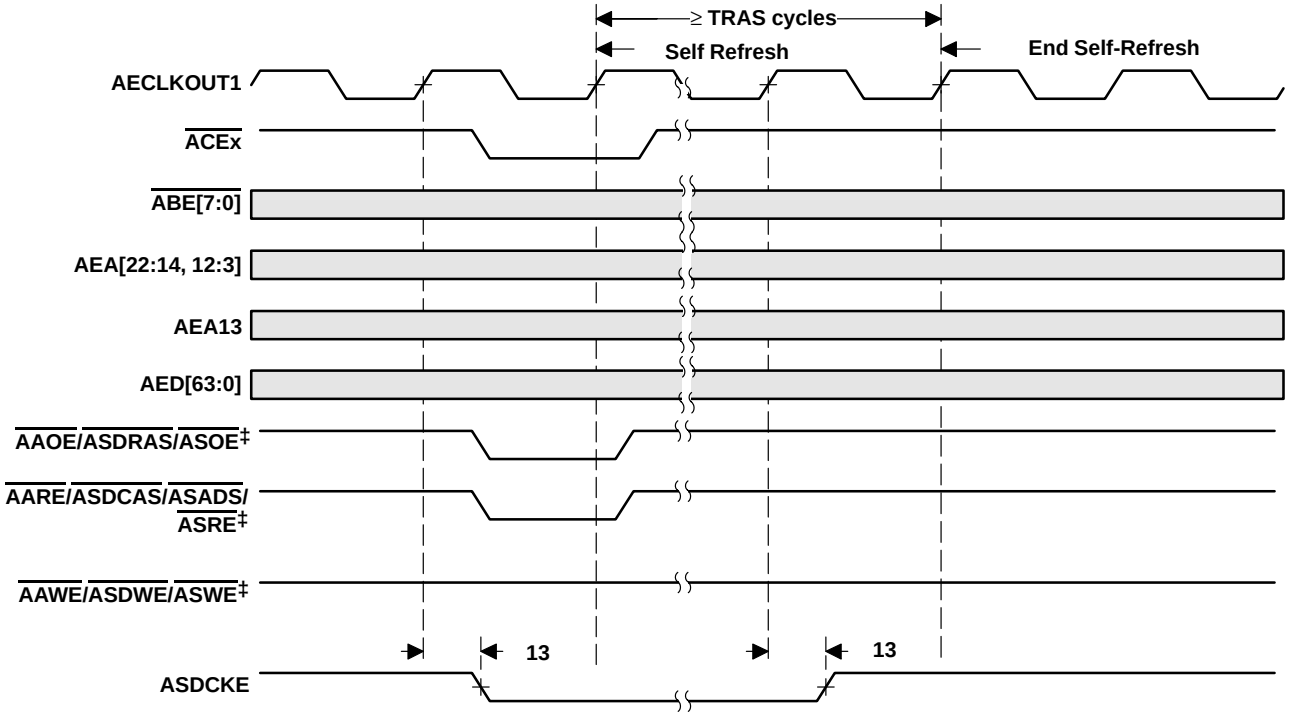
† The C64x™ has two EMIFs (EMIFA and EMIFB). All EMIFA signals are prefixed by an "A" and all EMIFB signals are prefixed by a "B". Throughout the rest of this document, in generic EMIF areas of discussion, the prefix "A" or "B" may be omitted [e.g., the synchronous DRAM memory access signals are shown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

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

Figure 27. SDRAM MRS Command for EMIFA and EMIFB†

PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

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

Figure 28. SDRAM Self-Refresh Timing for EMIFA Only†

HOLD/HOLDA TIMING

timing requirements for the HOLD/HOLDA cycles for EMIFA and EMIFB modules† (see Figure 29)

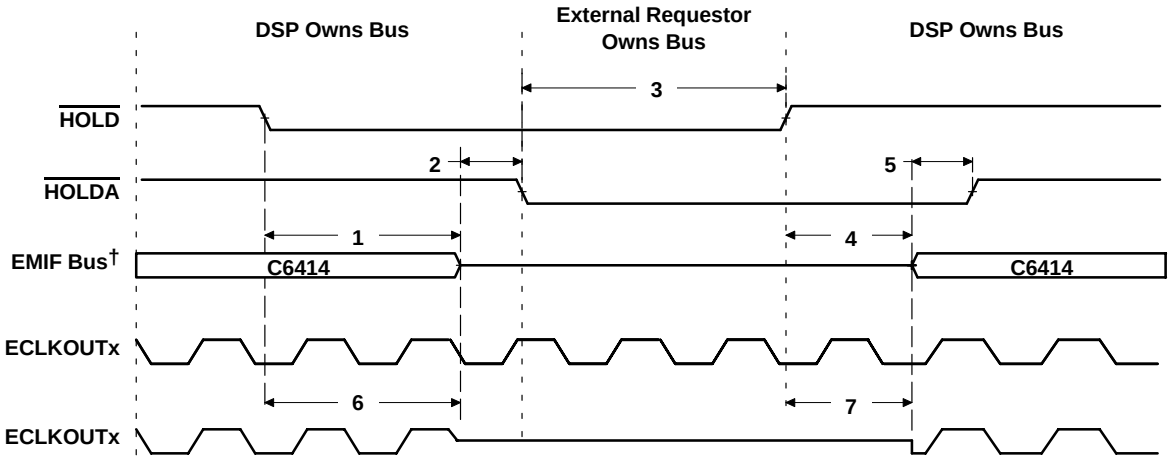
NO.		–400 –500 –600		UNIT
		MIN	MAX	
3	$t_{oh(HOLDAL-HOLDL)}$ Hold time, HOLD low after HOLDA low	E		ns

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

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

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{d(HOLDL-EMHZ)}$	Delay time, HOLD low to EMIF Bus high impedance	2E	¶	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	0	2E	ns
4	$t_{d(HOLDH-EMLZ)}$	Delay time, HOLD high to EMIF Bus low impedance	2E	7E	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	0	2E	ns
6	$t_{d(HOLDL-EKOHZ)}$	Delay time, HOLD low to ECLKOUTx high impedance	2E	¶	ns
7	$t_{d(HOLDH-EKOLZ)}$	Delay time, HOLD high to ECLKOUTx low impedance	2E	7E	ns

† E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
‡ For EMIFA, EMIF Bus consists of: ACE[3:0], ABE[7:0], AED[63:0], AEA[22:3], AARE/ASDCAS/ASADS/ASRE, AAOE/ASDRAS/ASOE, and AAWE/ASDWE/ASWE, ASDCKE, ASOE3, and APDT.
For EMIFB, EMIF Bus consists of: BCE[3:0], BBE[1:0], BED[15:0], BEA[20:1], BARE/BSDCAS/BSADS/BSRE, BAOE/BSDRAS/BSOE, and BAWE/BSDWE/BSWE, BSOE3, and BPDT.
§ The EKxHZ bits in the EMIF Global Control register (GBLCTL) determine the state of the ECLKOUTx signals during HOLDA. If EKxHZ = 0, ECLKOUTx continues clocking during Hold mode. If EKxHZ = 1, ECLKOUTx goes to high impedance during Hold mode, as shown in Figure 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.



† For EMIFA, EMIF Bus consists of: ACE[3:0], ABE[7:0], AED[63:0], AEA[22:3], AARE/ASDCAS/ASADS/ASRE, AAOE/ASDRAS/ASOE, and AAWE/ASDWE/ASWE, ASDCKE, ASOE3, and APDT.
For EMIFB, EMIF Bus consists of: BCE[3:0], BBE[1:0], BED[15:0], BEA[20:1], BARE/BSDCAS/BSADS/BSRE, BAOE/BSDRAS/BSOE, and BAWE/BSDWE/BSWE, BSOE3, and BPDT.

Figure 29. HOLD/HOLDA Timing for EMIFA and EMIFB

BUSREQ TIMING

switching characteristics over recommended operating conditions for the BUSREQ cycles for EMIFA and EMIFB modules (see Figure 30)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{d(AEKO1H-ABUSRV)}$	Delay time, AECLKOUT1 high to ABUSREQ valid	1	5.5	ns
2	$t_{d(BEKO1H-BBUSRV)}$	Delay time, BECLKOUT1 high to BBUSREQ valid	1	5.5	ns

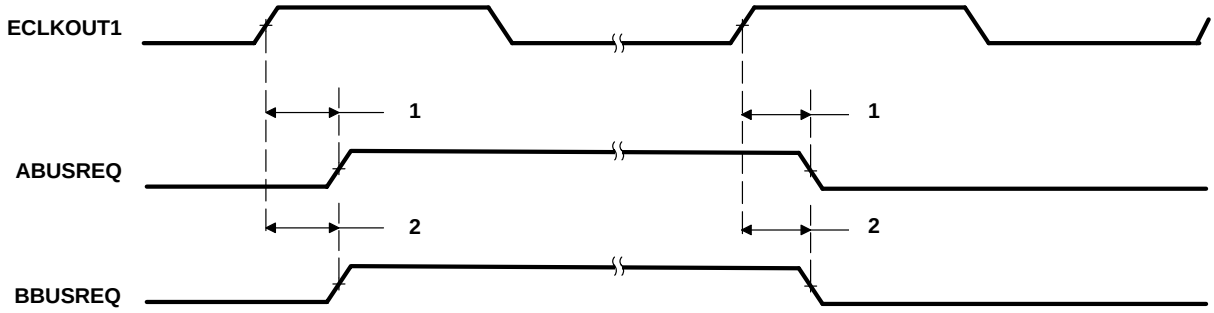


Figure 30. BUSREQ Timing for EMIFA and EMIFB

RESET TIMING

timing requirements for reset† (see Figure 31)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{w(RST)}$	Width of the RESET pulse (PLL stable)‡	10P		ns
		Width of the RESET pulse (PLL needs to sync up)§	250		µs
14	$t_{su(boot)}$	Setup time, boot configuration bits valid before RESET high¶	4P		ns
15	$t_{h(boot)}$	Hold time, boot configuration bits valid after RESET high¶	4P		ns

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

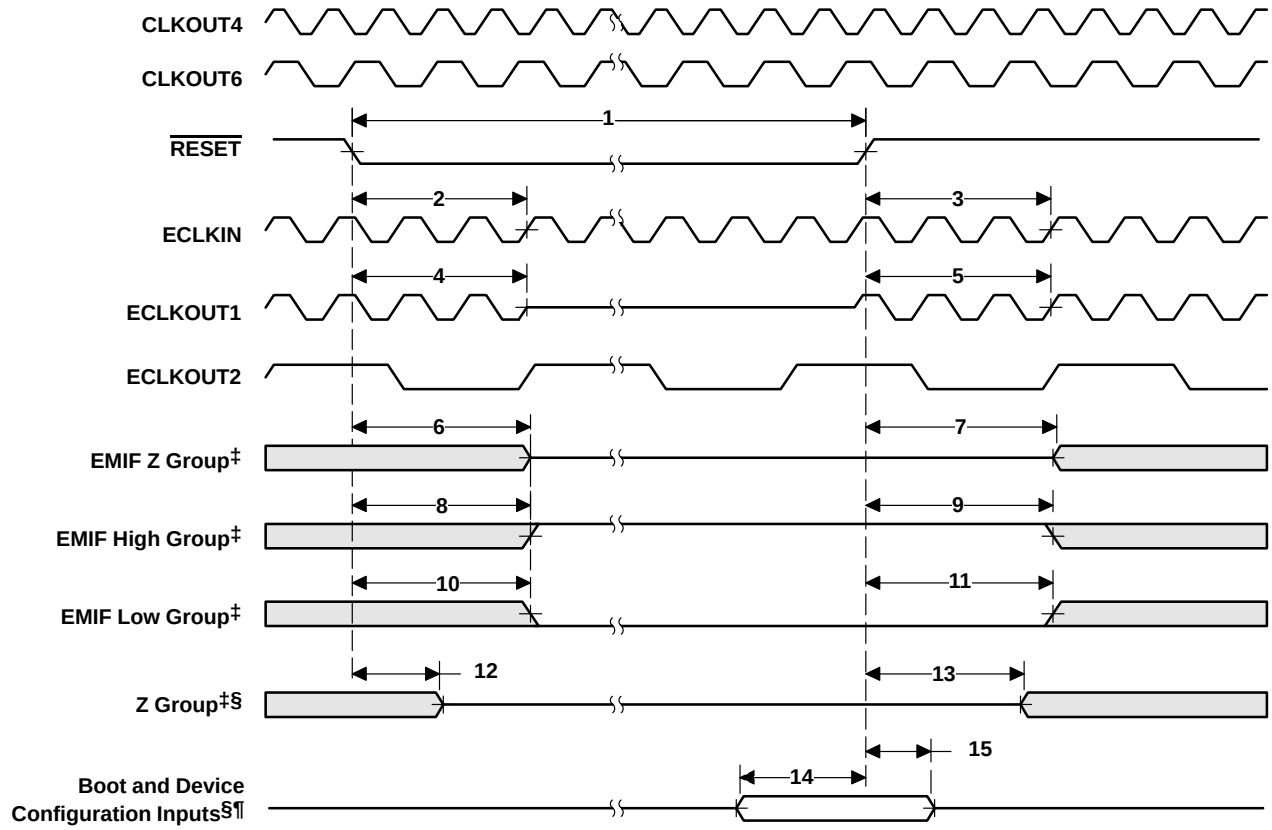
switching characteristics over recommended operating conditions during reset†#‖ (see Figure 31)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
2	$t_{d(RSTL-ECKI)}$	Delay time, RESET low to ECLKIN synchronized internally	2E	3P + 20E	ns
3	$t_{d(RSTH-ECKI)}$	Delay time, RESET high to ECLKIN synchronized internally	2E	3P + 20E	ns
4	$t_{d(RSTL-ECKO1HZ)}$	Delay time, RESET low to ECLKOUT1 high impedance	2E		ns
5	$t_{d(RSTH-ECKO1V)}$	Delay time, RESET high to ECLKOUT1 valid		3P + 20E	ns
6	$t_{d(RSTL-EMIFZHZ)}$	Delay time, RESET low to EMIF Z high impedance	2E	2P + 5E	ns
7	$t_{d(RSTH-EMIFZV)}$	Delay time, RESET high to EMIF Z valid	16E	3P + 20E	ns
8	$t_{d(RSTL-EMIFHIV)}$	Delay time, RESET low to EMIF high group invalid	2E		ns
9	$t_{d(RSTH-EMIFHV)}$	Delay time, RESET high to EMIF high group valid		3P + 20E	ns
10	$t_{d(RSTL-EMIFLIV)}$	Delay time, RESET low to EMIF low group invalid	2E		ns
11	$t_{d(RSTH-EMIFLV)}$	Delay time, RESET high to EMIF low group valid		3P + 20E	ns
12	$t_{d(RSTL-ZHZ)}$	Delay time, RESET low to Z group high impedance	0		ns
13	$t_{d(RSTH-ZV)}$	Delay time, RESET high to Z group valid	2P	6P	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
# E = the EMIF input clock (ECLKIN, CPU/4 clock, or CPU/6 clock) period in ns for EMIFA or EMIFB.
‖ EMIF Z group consists of: AEA[22:3], BEA[20:1], AED[63:0], BED[15:0], CE[3:0], ABE[7:0], BBE[1:0], ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE, SOE3, ASDCKE, and PDT.
EMIF high group consists of: AHOLDA and BHOLDA (when the corresponding HOLD input is high)
EMIF low group consists of: ABUSREQ and BBUSREQ; AHOLDA and BHOLDA (when the corresponding HOLD input is low)
Z group consists of: HD[31:0], CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, TOUT0, TOUT1, TOUT2, GP[15:0], HRDY, and HINT.

PRODUCT PREVIEW

RESET TIMING (CONTINUED)



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

‡ EMIF Z group consists of: AEA[22:3], BEA[20:1], AED[63:0], BED[15:0], CE[3:0], ABE[7:0], BBE[1:0], ARE/SDCAS/SADS/SRE, AWE/SDWE/SWE, and AOE/SDRAS/SOE, SOE3, ASDCKE, and PDT.
EMIF high group consists of: AHOLDA and BHOLDA (when the corresponding HOLD input is high)
EMIF low group consists of: ABUSREQ and BBUSREQ; AHOLDA and BHOLDA (when the corresponding HOLD input is low)
Z group consists of: HD[31:0], CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, TOUT0, TOUT1, TOUT2, GP[15:0], HRDY, and HINT.

§ If BEA[20:14] and HD5 pins are actively driven, care must be taken to ensure no timing contention between parameters 6, 7, 12, 13, 14, and 15.

¶ Boot and Device Configuration Inputs (during reset) include: EMIFB address pins BEA[20:14] and HD5.

Figure 31. Reset Timing†

EXTERNAL INTERRUPT TIMING

timing requirements for external interrupts† (see Figure 32)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{w(ILOW)}$	Width of the interrupt pulse low	4P		ns
2	$t_{w(IHIGH)}$	Width of the interrupt pulse high	4P		ns

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

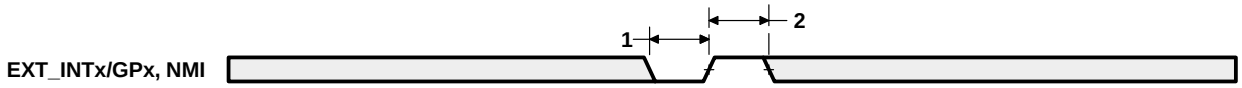


Figure 32. External/NMI Interrupt Timing

PRODUCT PREVIEW

HOST-PORT INTERFACE (HPI) TIMING

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

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{su(SELV-HSTBL)}$	Setup time, select signals§ valid before $\overline{\text{HSTROBE}}$ low	5		ns
2	$t_{h(HSTBL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HSTROBE}}$ low	2		ns
3	$t_{w(HSTBL)}$	Pulse duration, $\overline{\text{HSTROBE}}$ low	4P		ns
4	$t_{w(HSTBH)}$	Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses	4P		ns
10	$t_{su(SELV-HASL)}$	Setup time, select signals§ valid before $\overline{\text{HAS}}$ low	5		ns
11	$t_{h(HASL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HAS}}$ low	2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before $\overline{\text{HSTROBE}}$ high	5		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after $\overline{\text{HSTROBE}}$ high	2		ns
14	$t_{h(HRDYL-HSTBL)}$	Hold time, $\overline{\text{HSTROBE}}$ low after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly.	2		ns
18	$t_{su(HASL-HSTBL)}$	Setup time, $\overline{\text{HAS}}$ low before $\overline{\text{HSTROBE}}$ low	2		ns
19	$t_{h(HSTBL-HASL)}$	Hold time, $\overline{\text{HAS}}$ low after $\overline{\text{HSTROBE}}$ low	2		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 600 MHz, use P = 1.67 ns.
§ Select signals include: HCNTL[1:0] and HR/$\overline{\text{W}}$. For HPI16 mode only, select signals also include HHWIL.

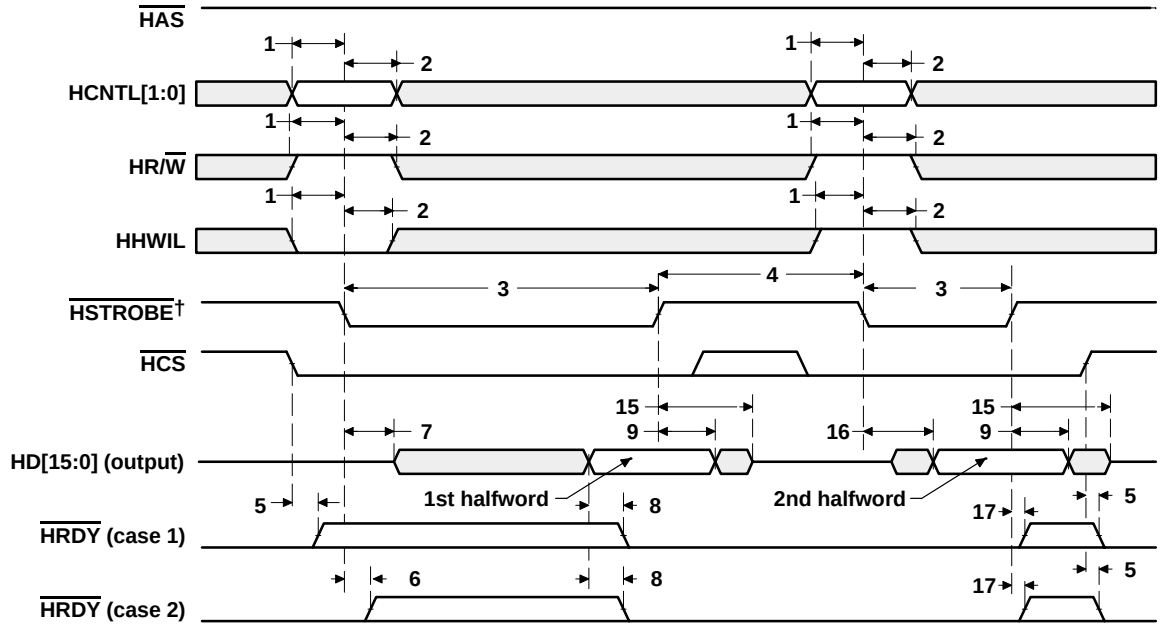
switching characteristics over recommended operating conditions during host-port interface cycles†‡ (see Figure 33 through Figure 40)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
5	$t_{d(HCS-HRDY)}$	Delay time, $\overline{\text{HCS}}$ to $\overline{\text{HRDY}}$¶	1	7	ns
6	$t_{d(HSTBL-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high#	3	12	ns
7	$t_{d(HSTBL-HDLZ)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD low impedance for an HPI read	2		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to $\overline{\text{HRDY}}$ low	2P – 6		ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high	3		ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance		12	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD valid (HPI16 only)		12	ns
17	$t_{d(HSTBH-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ high to $\overline{\text{HRDY}}$ high‖	3	12	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 600 MHz, use P = 1.67 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 an HPID read. 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.
‖ 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.

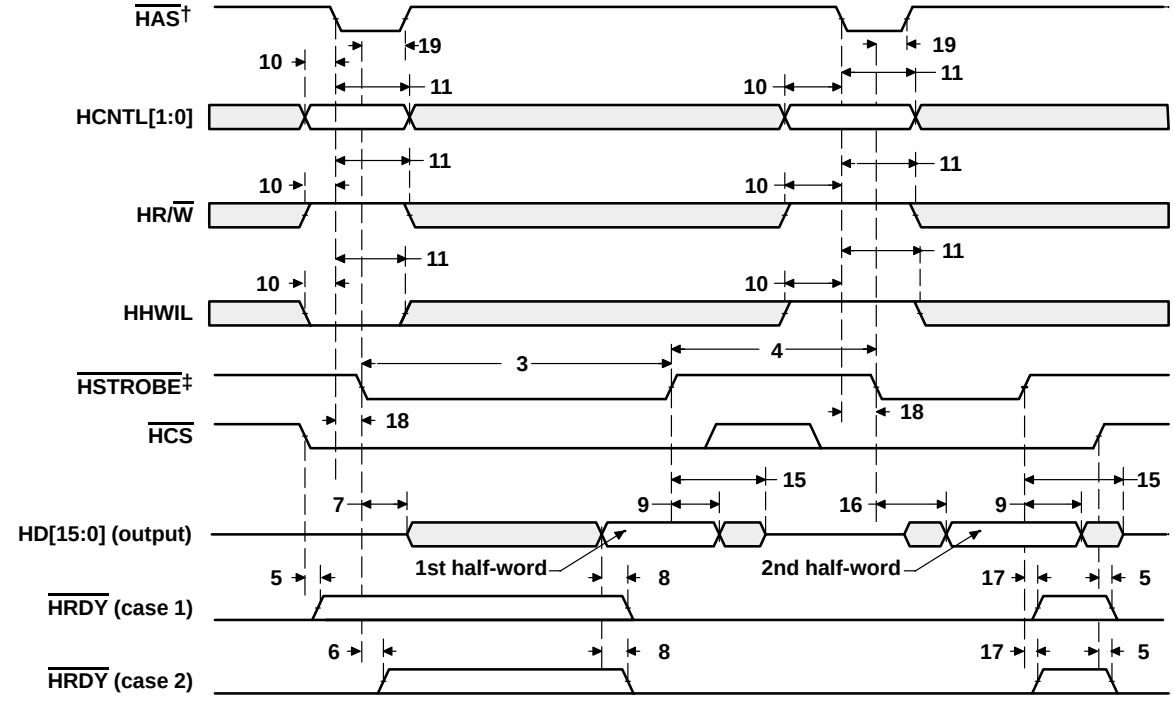
PRODUCT PREVIEW

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



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

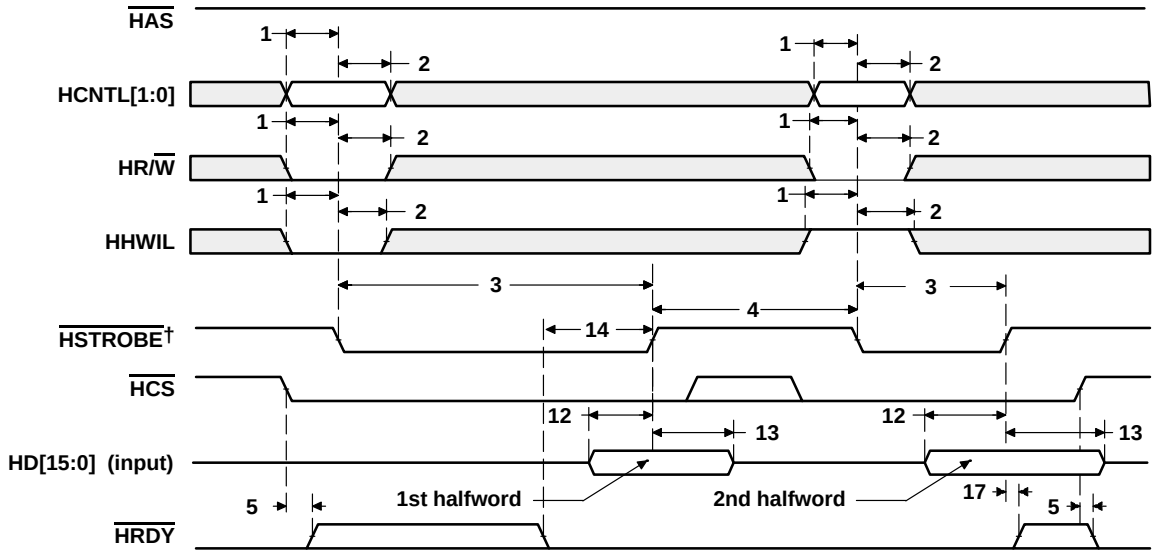
Figure 33. HPI16 Read Timing (HAS Not Used, Tied High)



† 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.

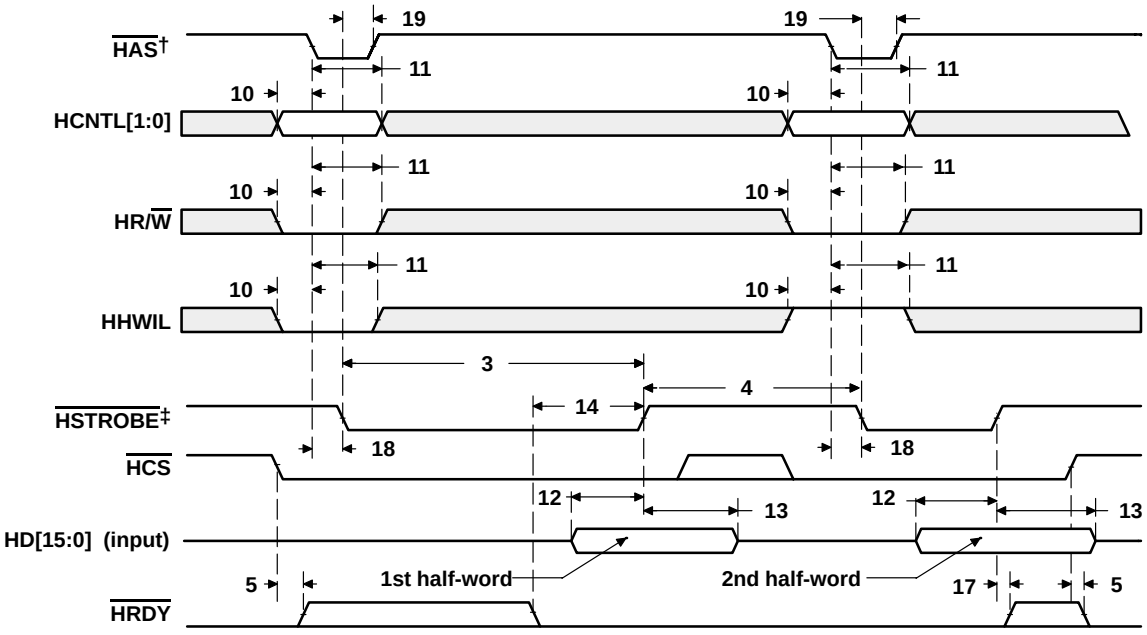
Figure 34. HPI16 Read Timing (HAS Used)

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)

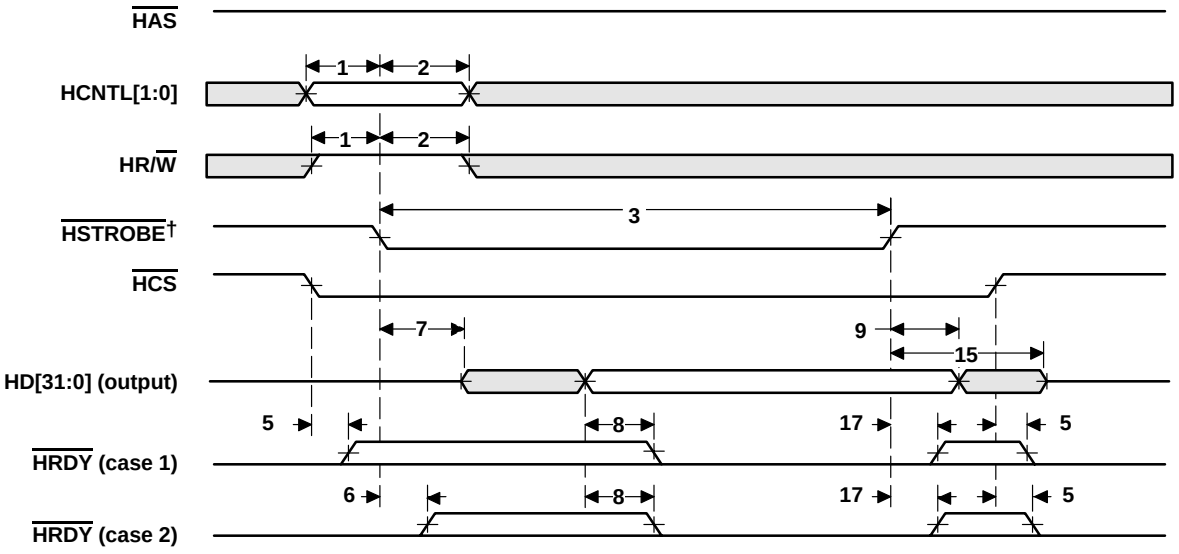


† For correct operation, strobe the HAS signal only once per HSTROBE active cycle.
‡ HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

Figure 36. HPI16 Write Timing (HAS Used)

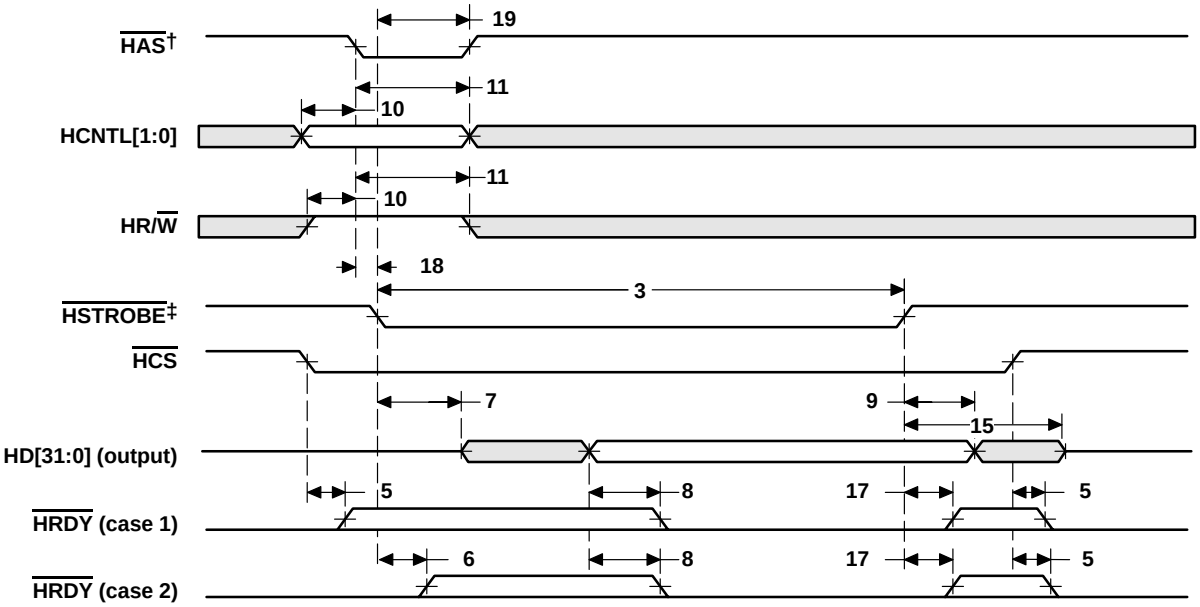
PRODUCT PREVIEW

HOST-PORT INTERFACE (HPI) TIMING (CONTINUED)



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

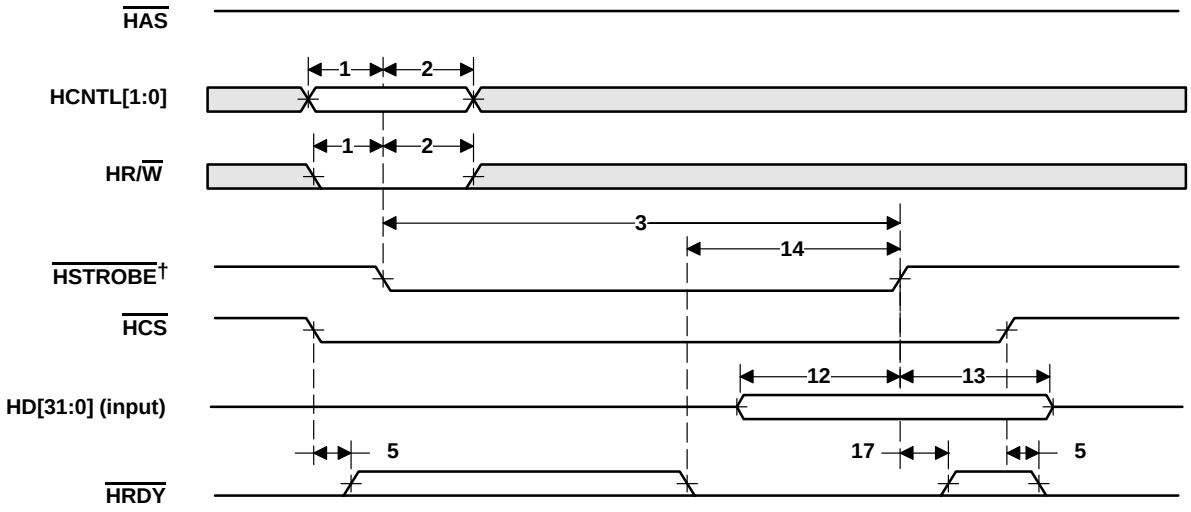
Figure 37. HPI32 Read Timing (HAS Not Used, Tied High)



† 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.

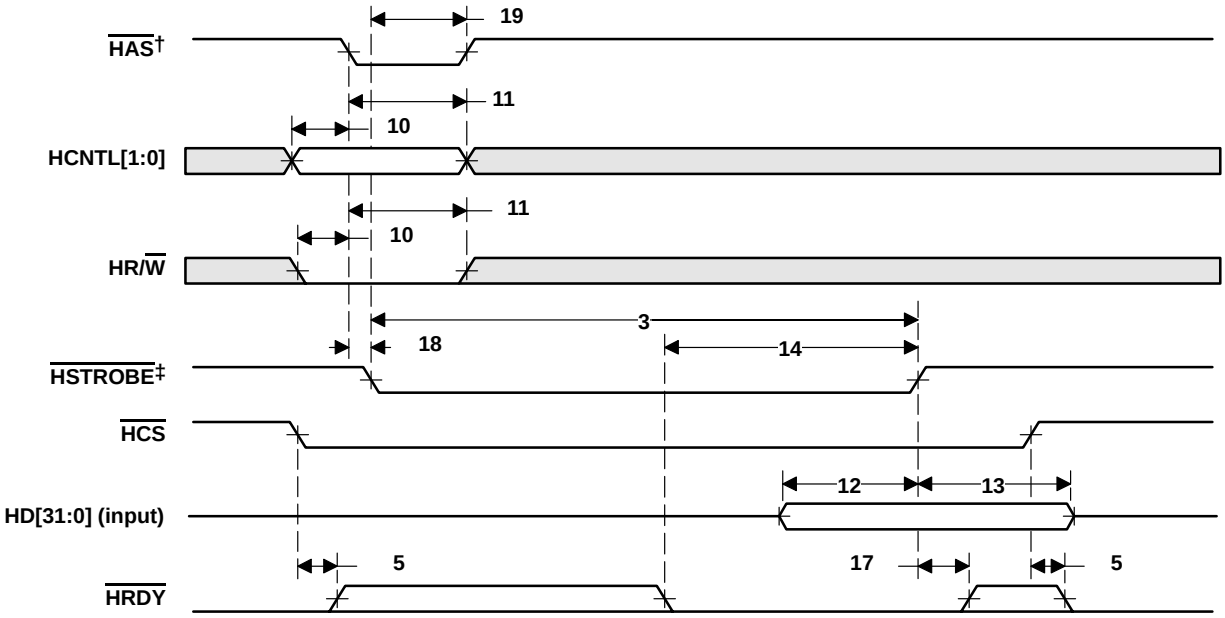
Figure 38. HPI32 Read Timing (HAS Used)

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

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING

timing requirements for McBSP†‡ (see Figure 41)

NO.				–400 –500 –600	UNIT
				MIN MAX	
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	4P§	ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	$0.5t_{c(CKRX)} - 1$	ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9	ns
			CLKR ext	1	
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	6	ns
			CLKR ext	3	
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	8	ns
			CLKR ext	0	
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	3	ns
			CLKR ext	3	
10	$t_{su(FXH-CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	9	ns
			CLKX ext	1	
11	$t_{h(CKXL-FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	6	ns
			CLKX ext	3	

† 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 600 MHz, use P = 1.67 ns.
§ The maximum bit rate for McBSP-to-McBSP communications is 100 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 600 MHz (P = 1.67 ns), use 10 ns 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 41)

NO.	PARAMETER			–400 –500 –600		UNIT
				MIN	MAX	
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		4	10	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	4P§¶		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C – 1#	C + 1#	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	–1	3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	–1	3	ns
			CLKX ext	3	9	
12	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	–1	4	ns
			CLKX ext	3	9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	–1+ D1‖	4 + D2‖	ns
			CLKX ext	3 + D1‖	9 + D2‖	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode	FSX int	–1	3	ns
			FSX ext	3	9	

† 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 600 MHz, use P = 1.67 ns.

¶ The maximum bit rate for McBSP-to-McBSP communications is 100 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 600 MHz (P = 1.67 ns), use 10 ns 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 = 2P, D2 = 4P

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

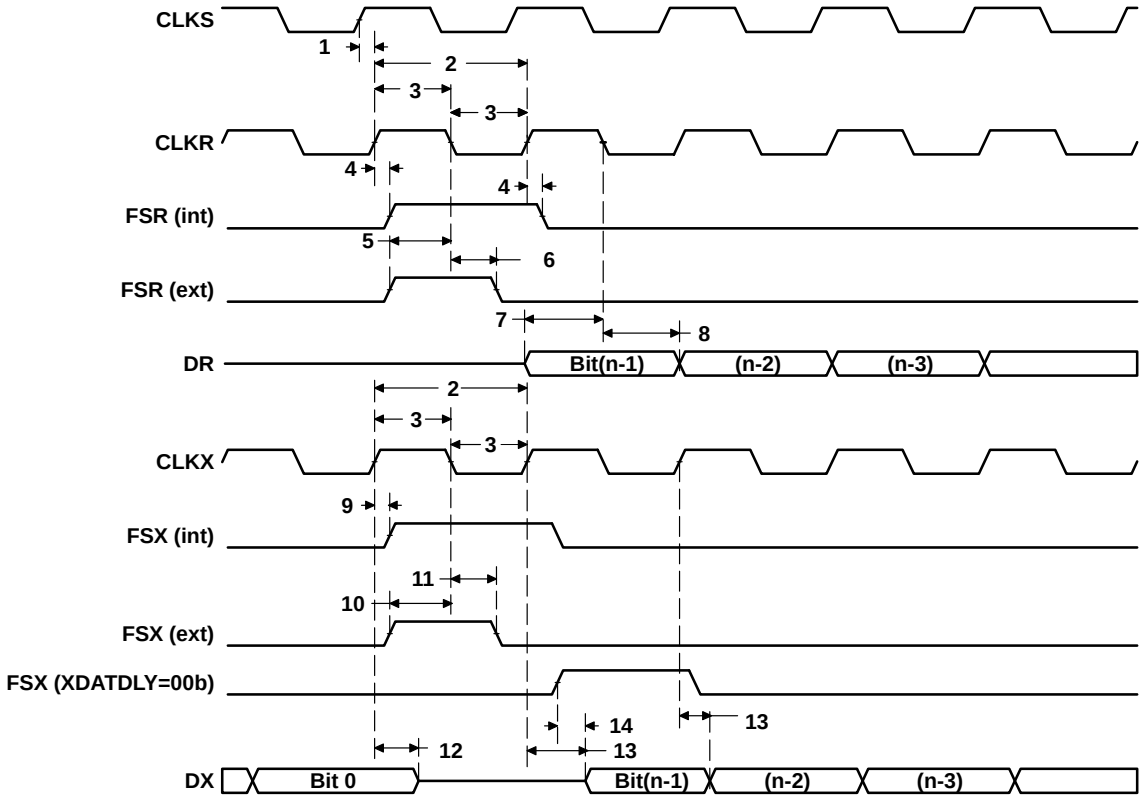


Figure 41. McBSP Timing

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 42)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{su(FRH-CKSH)}$	Setup time, FSR high before CLKS high	4		ns
2	$t_{h(CKSH-FRH)}$	Hold time, FSR high after CLKS high	4		ns

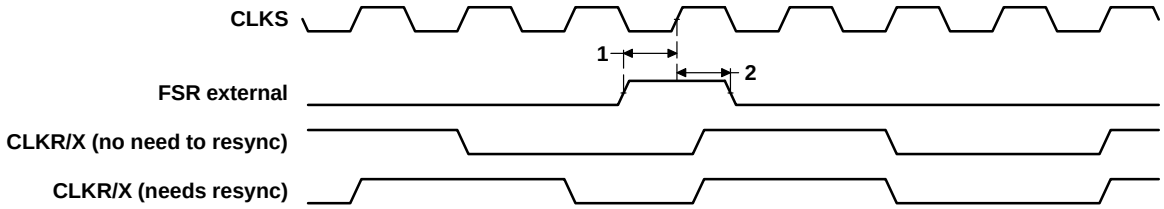


Figure 42. FSR Timing When GSYNC = 1

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 43)

NO.			–400 –500 –600				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 12P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 24P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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 43)

NO.	PARAMETER		–400 –500 –600				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L – 2	L + 3			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	12P + 4	20P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	L – 2	L + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			4P + 3	12P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			8P + 2	16P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

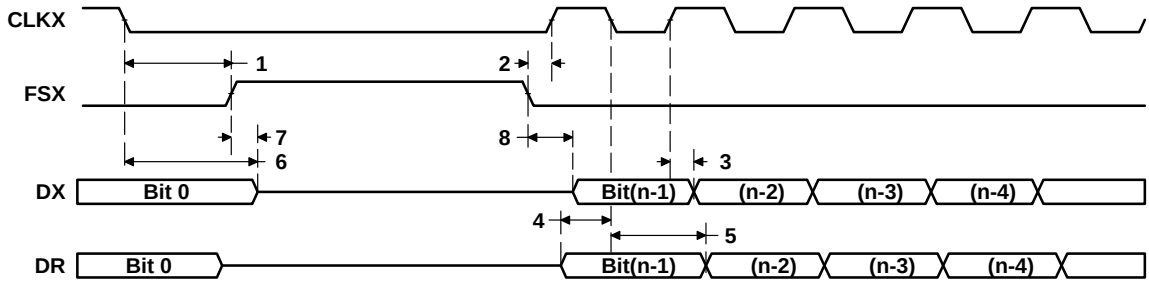


Figure 43. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 44)

NO.			–400 –500 –600				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 12P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 24P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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 44)

NO.	PARAMETER		–400 –500 –600				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	L – 2	L + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	T – 2	T + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–2	4	12P + 4	20P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	–2	4	12P + 3	20P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H – 2	H + 4	8P + 2	16P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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).

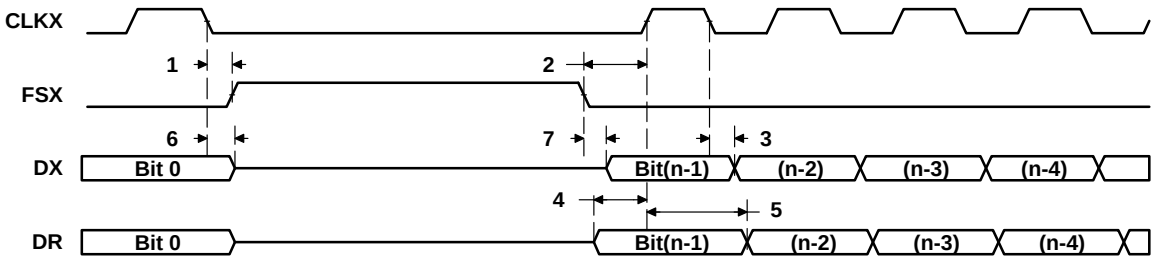


Figure 44. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 45)

NO.			–400 –500 –600				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 12P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 24P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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 45)

NO.	PARAMETER		–400 –500 –600				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	H – 2	H + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–2	4	12P + 4	20P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	H – 2	H + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			4P + 3	12P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			8P + 2	16P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

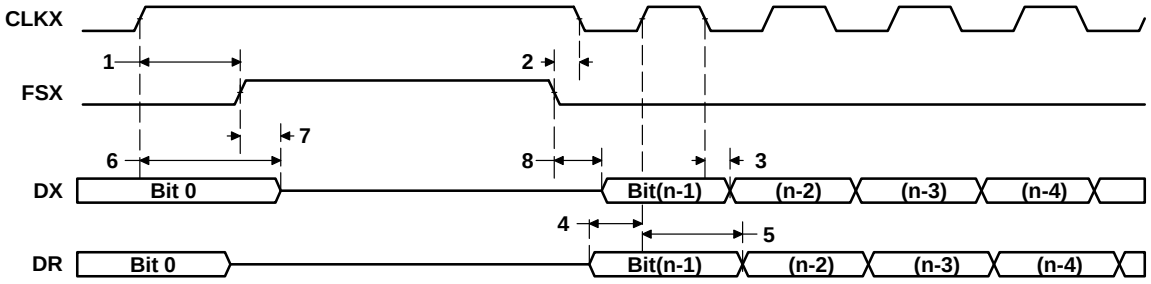


Figure 45. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

timing requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 46)

NO.			–400 –500 –600				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 12P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 24P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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 46)

NO.	PARAMETER		–400 –500 –600				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	H – 2	H + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	T – 2	T + 1			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	12P + 4	20P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	4	12P + 3	20P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L – 2	L + 4	8P + 2	16P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 600 MHz, use P = 1.67 ns.
‡ For all SPI Slave modes, CLKG is programmed as 1/2 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).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT (McBSP) TIMING (CONTINUED)

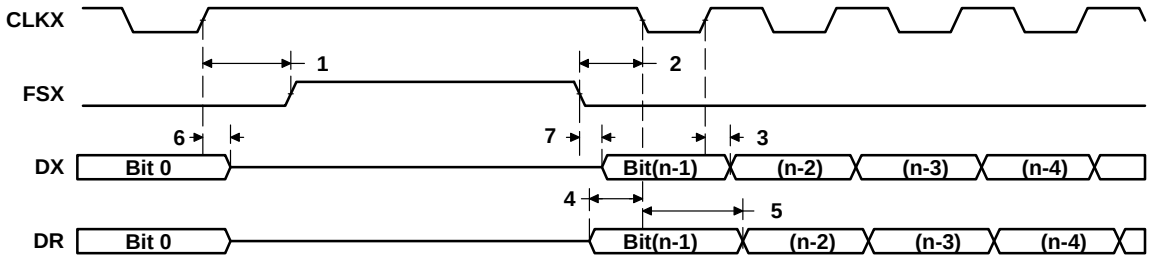


Figure 46. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

TIMER TIMING

timing requirements for timer inputs† (see Figure 47)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{w(TINPH)}$	Pulse duration, TINP high	4P		ns
2	$t_{w(TINPL)}$	Pulse duration, TINP low	4P		ns

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

switching characteristics over recommended operating conditions for timer outputs† (see Figure 47)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
3	$t_{w(TOUTH)}$	Pulse duration, TOUT high	8P–3		ns
4	$t_{w(TOUTL)}$	Pulse duration, TOUT low	8P–3		ns

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

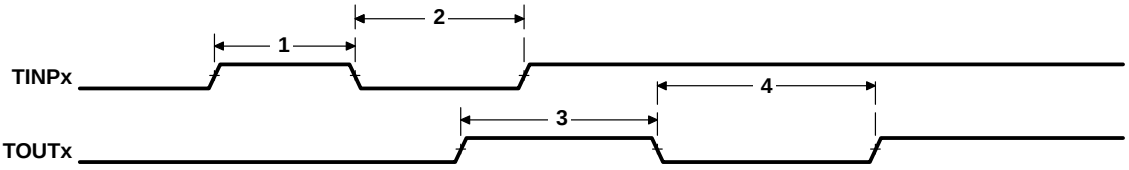


Figure 47. Timer Timing

GENERAL-PURPOSE INPUT/OUTPUT (GPIO) PORT TIMING

timing requirements for GPIO inputs† (see Figure 48)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{w(GPIH)}$	Pulse duration, GPIx high	4P		ns
2	$t_{w(GPIL)}$	Pulse duration, GPIx low	4P		ns

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

switching characteristics over recommended operating conditions for GPIO outputs† (see Figure 48)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
3	$t_{w(GPOH)}$	Pulse duration, GPOx high	8P–3		ns
4	$t_{w(GPOL)}$	Pulse duration, GPOx low	8P–3		ns

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

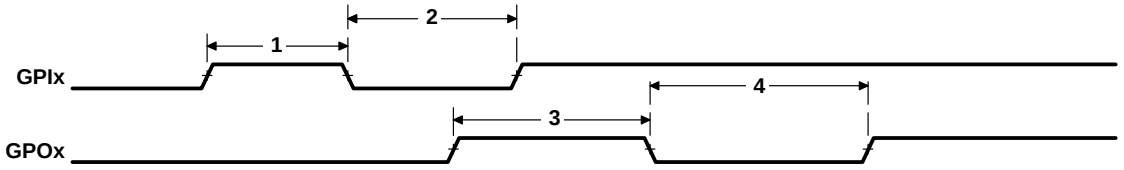


Figure 48. GPIO Port Timing

PRODUCT PREVIEW

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 49)

NO.			–400 –500 –600		UNIT
			MIN	MAX	
1	$t_{c(TCK)}$	Cycle time, TCK	35		ns
3	$t_{su(TDIV-TCKH)}$	Setup time, TDI/TMS/$\overline{TRST}$ valid before TCK high	10		ns
4	$t_{h(TCKH-TDIV)}$	Hold time, TDI/TMS/$\overline{TRST}$ valid after TCK high	9		ns

switching characteristics over recommended operating conditions for JTAG test port (see Figure 49)

NO.	PARAMETER		–400 –500 –600		UNIT
			MIN	MAX	
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid	–3	18	ns

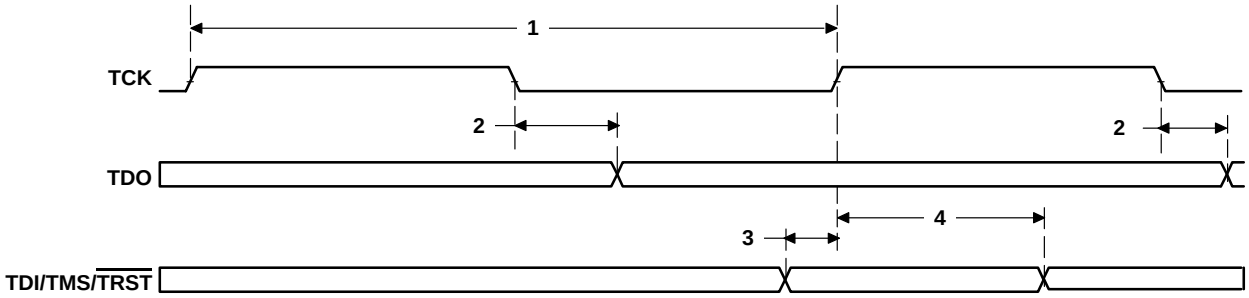
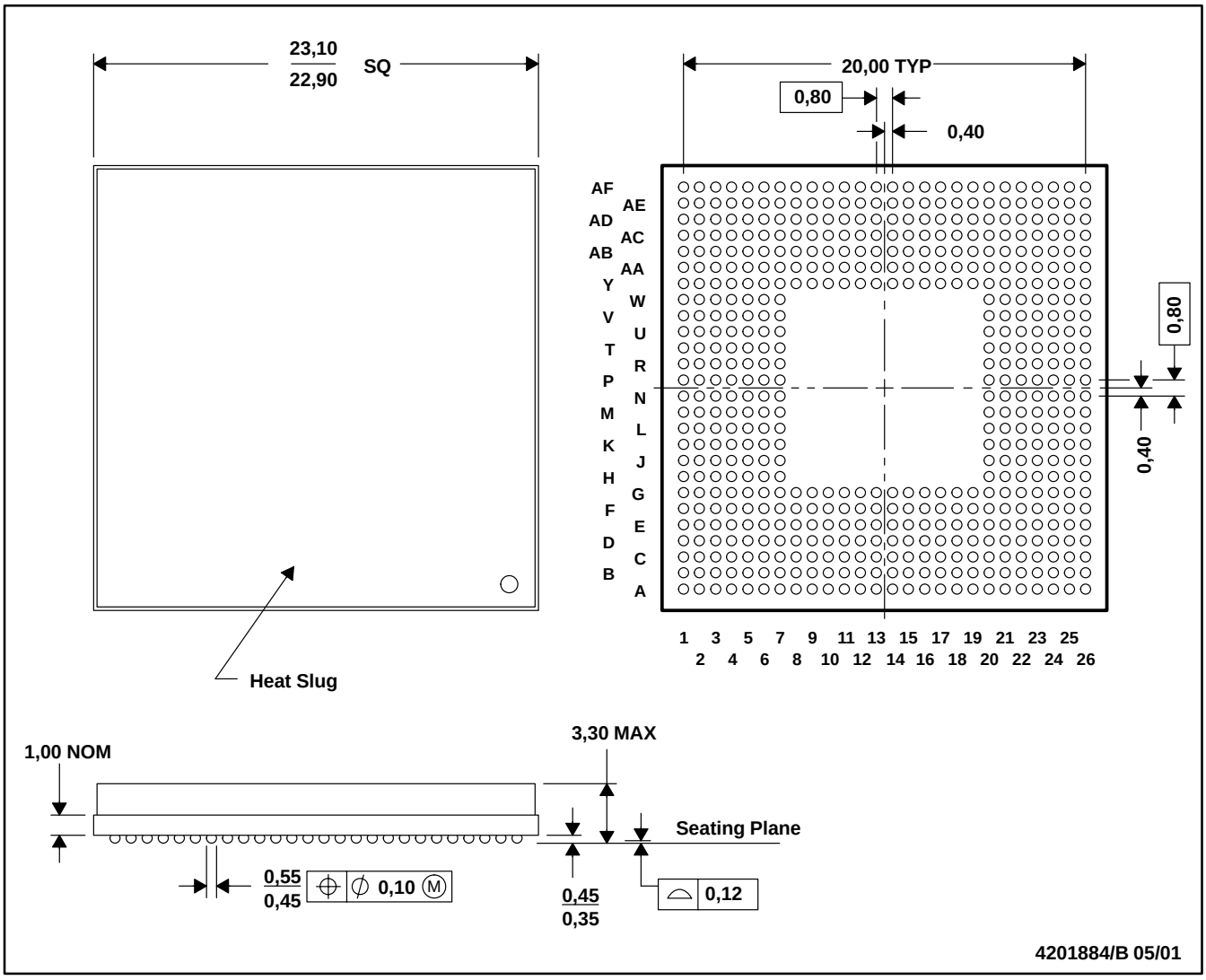


Figure 49. JTAG Test-Port Timing

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	3.54	N/A
2	$R\Theta_{JA}$	Junction-to-free air	16.9	0.00
3	$R\Theta_{JA}$	Junction-to-free air	15.3	0.50
4	$R\Theta_{JA}$	Junction-to-free air	14.0	1.00
5	$R\Theta_{JA}$	Junction-to-free air	12.9	2.00

† 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