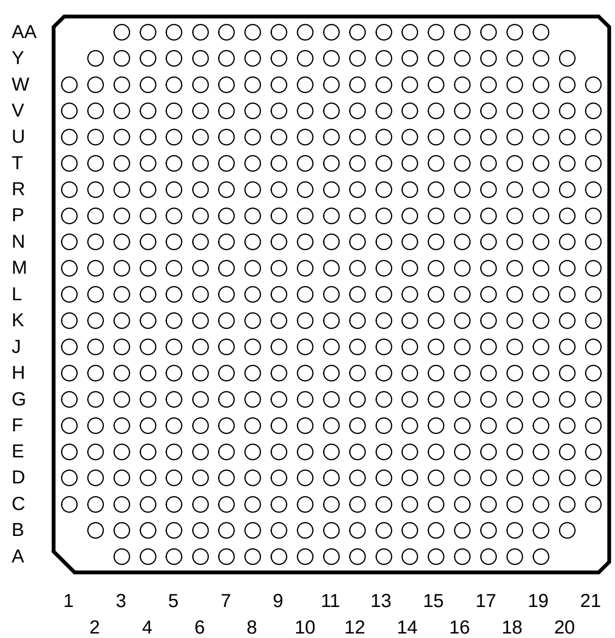
- **High-Performance Fixed-Point Digital Signal Processor (DSP) – SMJ320C62x™**
  - **5-ns Instruction Cycle Time**
  - **200-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **1600 Million Instructions Per Second (MIPS)**
- **429-Pin Ball Grid Array (BGA) Package (GLP Suffix)**
- **VelociTI™ Advanced Very-Long-Instruction-Word (VLIW) C62x™ DSP Core**
  - **Eight Highly Independent Functional Units:**
    - **Six Arithmetic Logic Units (ALUs) (32-/40-Bit)**
    - **Two 16-Bit Multipliers (32-Bit Result)**
  - **Load-Store Architecture With 32 32-Bit General-Purpose Registers**
  - **Instruction Packing Reduces Code Size**
  - **All Instructions Conditional**
- **Instruction Set Features**
  - **Byte-Addressable (8-, 16-, 32-Bit Data)**
  - **8-Bit Overflow Protection**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit-Counting**
  - **Normalization**
- **7M-Bit On-Chip SRAM**
  - **3M-Bit Internal Program/Cache (96K 32-Bit Instructions)**
  - **4M-Bit Dual-Access Internal Data (512K Bytes)**
    - **Organized as Two 256K-Byte Blocks for Improved Concurrency**
- **Flexible Phase-Locked-Loop (PLL) Clock Generator**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Synchronous Memories: SDRAM or SBSRAM**
  - **Glueless Interface to Asynchronous Memories: SRAM and EPROM**
  - **52M-Byte Addressable External Memory Space**
- **Four-Channel Bootloading Direct-Memory-Access (DMA) Controller With an Auxiliary Channel**
- **32-Bit Expansion Bus**
  - **Glueless/Low-Glue Interface to Popular PCI Bridge Chips**
  - **Glueless/Low-Glue Interface to Popular Synchronous or Asynchronous Microprocessor Buses**
  - **Master/Slave Functionality**
  - **Glueless Interface to Synchronous FIFOs and Asynchronous Peripherals**
- **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™)**
- **Two 32-Bit General-Purpose Timers**
- **IEEE-1149.1 (JTAG†) Boundary-Scan-Compatible**
- **0.15-μm/5-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1.5-V Internal**

SMJ320C62x, VelociTI, and C62x are trademarks of Texas Instruments.
Motorola is a trademark of Motorola, Inc.
All trademarks are the property of their respective owners.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

Table of Contents

GLP BGA package (bottom view)

GLP 429-PIN BALL GRID ARRAY (BGA) PACKAGE (BOTTOM VIEW)



description

The SMJ320C6203 device is part of the SMJ320C62x™ fixed-point DSP generation in the SMJ320C6000™ DSP platform. The C62x™ DSP devices are based on the high-performance, advanced VelociTI™ very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunction applications.

The SMJ320C62x™ DSP offers cost-effective solutions to high-performance DSP-programming challenges. The SMJ320C6203 has a performance capability of up to 1600 MIPS at a clock rate of 200 MHz. The C6203 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. This processor has 32 general-purpose registers of 32-bit word length and eight highly independent functional units. The eight functional units provide six arithmetic logic units (ALUs) for a high degree of parallelism and two 16-bit multipliers for a 32-bit result. The C6203 can produce two multiply-accumulates (MACs) per cycle for a total of 400 million MACs per second (MMACS). The C6203 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The C6203 device program memory consists of two blocks, with a 256K-byte block configured as memory-mapped program space, and the other 128K-byte block user-configurable as cache or memory-mapped program space. Data memory for the C6203 consists of two 256K-byte blocks of RAM.

The C6203 device has a powerful and diverse set of peripherals. The peripheral set includes three multichannel buffered serial ports (McBSPs), two general-purpose timers, a 32-bit expansion bus that offers ease of interface to synchronous or asynchronous industry-standard host bus protocols, and a glueless 32-bit external memory interface (EMIF) capable of interfacing to SDRAM or SBSRAM and asynchronous peripherals.

The C62x™ devices have a complete set of development tools that includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

device characteristics

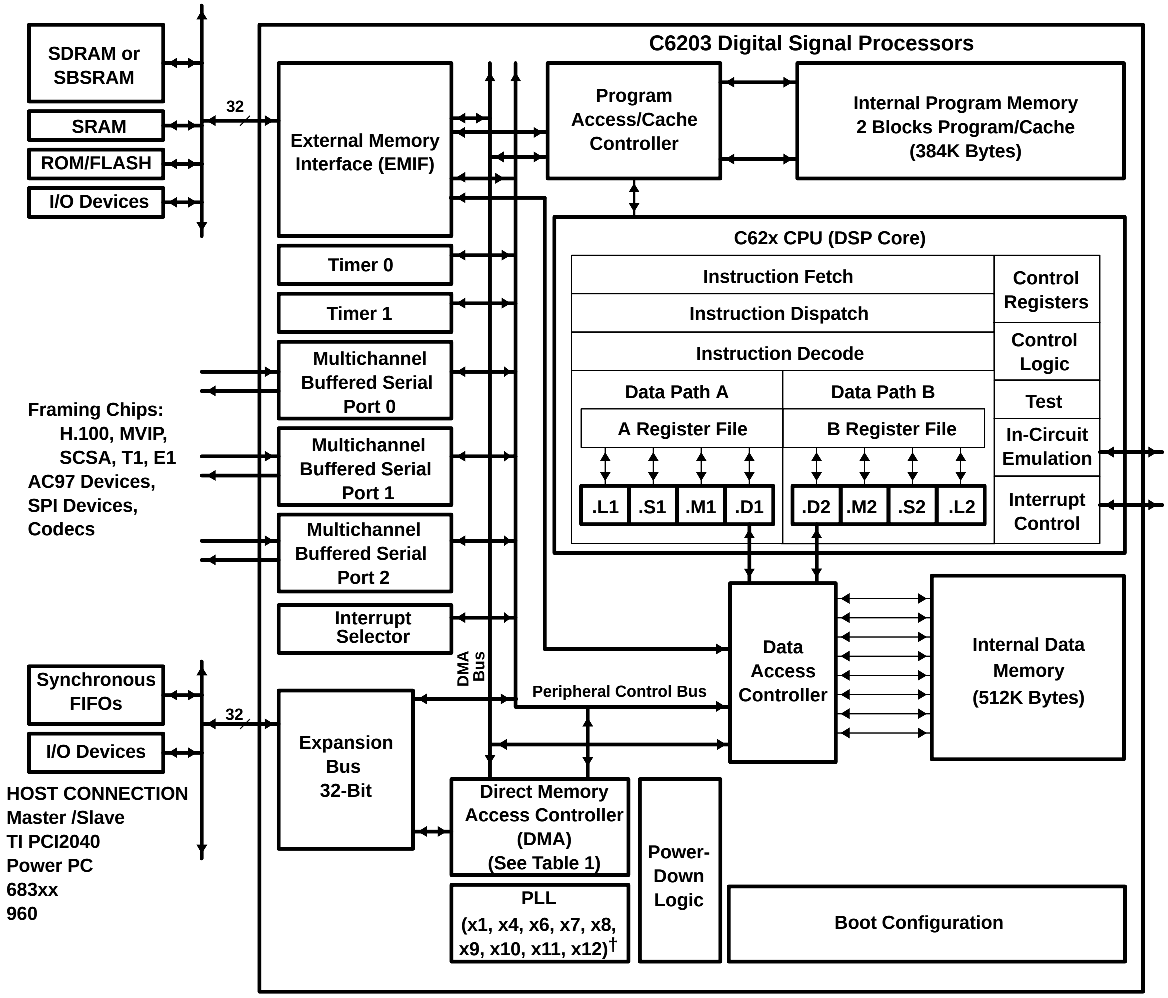
Table 1 provides an overview of the SMJ320C6203 DSP. The table shows significant features of the device, including the capacity of on-chip RAM, the peripherals, the execution time, and the package type with pin count. This data sheet focuses on the functionality of the SMJ320C6203 device. For more details on the C6000™ DSP part numbering, see Figure 4.

device characteristics (continued)

Table 1. Characteristics of the C6203 DSP

HARDWARE FEATURES		C6203
Peripherals	EMIF	√
	DMA	4-Channel With Throughput Enhancements
	Expansion Bus	√
	McBSPs	3
	32-Bit Timers	2
Internal Program Memory	Size (Bytes)	384K
	Organization	Block 0: 256K-Byte Mapped Program Block 1: 128K-Byte Cache/Mapped Program
Internal Data Memory	Size (Bytes)	512K
	Organization	2 Blocks: Four 16-Bit Banks per Block 50/50 Split
CPU ID + CPU Rev ID	Control Status Register (CSR.[31:16])	0x0003
Frequency	MHz	200
Cycle Time	ns	5 ns (6203-200)
Voltage	Core (V)	1.5
	I/O (V)	3.3
PLL Options	CLKIN frequency multiplier [Bypass (x1), x4, x6, x7, x8, x9, x10, and x11]	Bypass (x1), x4, x6, x7, x8, x9, x10, and x11
BGA Package	27 x 27 mm	GLP
Process Technology	μm	0.15 μm
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	PD

functional and CPU (DSP core) block diagram



† For additional details on the PLL clock module and specific options for the C6203 device, see Table 1 and the *Clock PLL* section of this data sheet.

CPU (DSP core) description

The CPU fetches VelociTI™ advanced very-long instruction words (VLIW) (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 C62x CPU from other VLIW architectures.

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 16 32-bit registers for a total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU [see the functional and CPU (DSP core) block diagram and Figure 1]. The four functional units on each side of the CPU can freely share the 16 registers belonging to that side. Additionally, each side features 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. While 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, register access using the register file across the CPU supports one read and one write per cycle.

Another key feature of the C62x 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 C62x CPU supports a variety of indirect addressing modes using either linear- or circular-addressing modes with 5- or 15-bit offsets. All instructions are conditional, and most can access any one of the 32 registers. Some registers, however, are singled out to support specific addressing or to hold the condition for conditional instructions (if the condition is not automatically “true”). The two .M functional units are dedicated for multiplies. The two .S and .L functional units perform a general set of arithmetic, logical, and branch functions with results available every clock cycle.

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. If an execute packet crosses the 256-bit-wide fetch-packet boundary, the assembler places it in the next fetch packet, while the remainder of the current fetch packet is padded with NOP instructions. 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 or half-words as well. All load and store instructions are byte-, half-word, or word-addressable.

CPU (DSP core) description (continued)

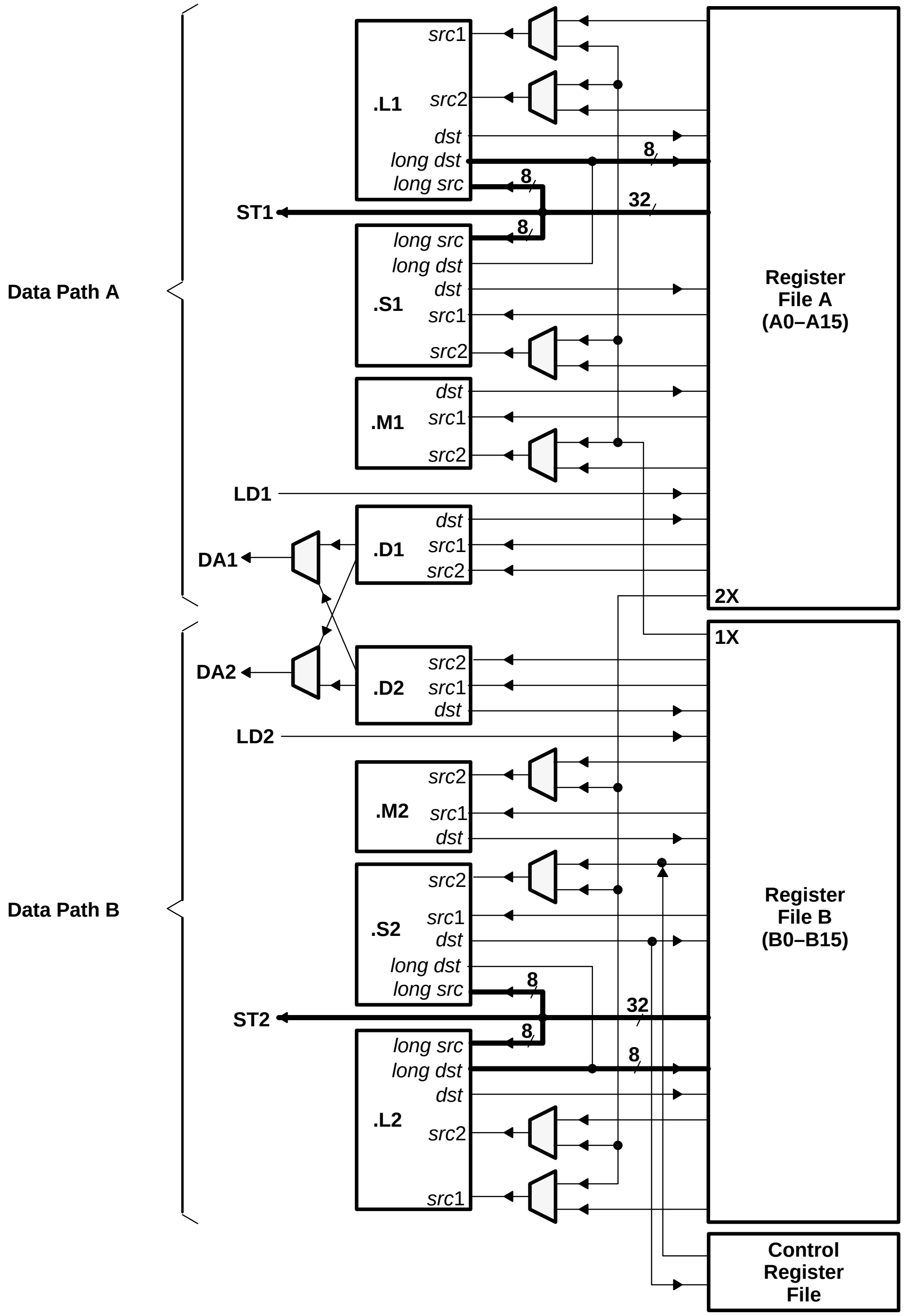


Figure 1. SMJ320C62x CPU (DSP Core) Data Paths

memory map summary

Table 2 shows the memory map address ranges of the C6203 device. The C6203 device has the capability of a MAP 0 or MAP 1 memory block configuration. These memory block configurations are set up at reset by the boot configuration pins (generically called BOOTMODE[4:0]). For the C6203 device, the BOOTMODE configuration is handled, at reset, by the expansion bus module (specifically XD[4:0] pins). For more detailed information on the C6203 device settings, which include the device boot mode configuration at reset and other device-specific configurations, see the Boot Configuration section and the Boot Configuration Summary table of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

Table 2. 320C6203 Memory Map Summary

MEMORY BLOCK DESCRIPTION		BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
MAP 0	MAP 1		
External Memory Interface (EMIF) CE0	Internal Program RAM	384K	0000_0000 – 0005_FFFF
EMIF CE0	Reserved	4M – 384K	0006_0000 – 003F_FFFF
EMIF CE0	EMIF CE0	12M	0040_0000 – 00FF_FFFF
EMIF CE1	EMIF CE0	4M	0100_0000 – 013F_FFFF
Internal Program RAM	EMIF CE1	384K	0140_0000 – 0145_FFFF
Reserved	EMIF CE1	4M – 384K	0146_0000 – 017F_FFFF
EMIF Registers		256K	0180_0000 – 0183_FFFF
DMA Controller Registers		256K	0184_0000 – 0187_FFFF
Expansion Bus 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		512	019C_0000 – 019C_01FF
Power-Down Registers		256K – 512	019C_0200 – 019F_FFFF
Reserved		256K	01A0_0000 – 01A3_FFFF
McBSP 2 Registers		256K	01A4_0000 – 01A7_FFFF
Reserved		5.5M	01A8_0000 – 01FF_FFFF
EMIF CE2		16M	0200_0000 – 02FF_FFFF
EMIF CE3		16M	0300_0000 – 03FF_FFFF
Reserved		1G – 64M	0400_0000 – 3FFF_FFFF
Expansion bus XCE0		256M	4000_0000 – 4FFF_FFFF
Expansion bus XCE1		256M	5000_0000 – 5FFF_FFFF
Expansion bus XCE2		256M	6000_0000 – 6FFF_FFFF
Expansion bus XCE3		256M	7000_0000 – 7FFF_FFFF
Internal Data RAM		512K	8000_0000 – 8007_FFFF
Reserved		2G – 512K	8008_0000 – FFFF_FFFF

peripheral register descriptions

Table 3 through Table 12 identify the peripheral registers for the C6203 device by their register names, acronyms, and hex address or hex address range. For more detailed information on the register contents, bit names, and their descriptions, see the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

Table 3. EMIF Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0180 0000	GBLCTL	EMIF global control	
0180 0004	CECTL1	EMIF CE1 space control	External or internal; dependent on MAP0 or MAP1 configuration (selected by the MAP bit in the EMIF GBLCTL register
0180 0008	CECTL0	EMIF CE0 space control	External or internal; dependent on MAP0 or MAP1 configuration (selected by the MAP bit in the EMIF GBLCTL register
0180 000C	–	Reserved	
0180 0010	CECTL2	EMIF CE2 space control	Corresponds to EMIF CE2 memory space: [0200 0000 – 02FF FFFF]
0180 0014	CECTL3	EMIF CE3 space control	Corresponds to EMIF CE3 memory space: [0300 0000 – 03FF FFFF]
0180 0018	SDCTL	EMIF SDRAM control	
0180 001C	SDTIM	EMIF SDRAM refresh control	
0180 0020 – 0180 0054	–	Reserved	
0180 0058 – 0183 FFFF	–	Reserved	

peripheral register descriptions (continued)

Table 4. DMA Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
0184 0000	PRICTL0	DMA channel 0 primary control
0184 0004	PRICTL2	DMA channel 2 primary control
0184 0008	SECCTL0	DMA channel 0 secondary control
0184 000C	SECCTL2	DMA channel 2 secondary control
0184 0010	SRC0	DMA channel 0 source address
0184 0014	SRC2	DMA channel 2 source address
0184 0018	DST0	DMA channel 0 destination address
0184 001C	DST2	DMA channel 2 destination address
0184 0020	XFRCNT0	DMA channel 0 transfer counter
0184 0024	XFRCNT2	DMA channel 2 transfer counter
0184 0028	GBLCNTA	DMA global count reload register A
0184 002C	GBLCNTB	DMA global count reload register B
0184 0030	GBLIDXA	DMA global index register A
0184 0034	GBLIDXB	DMA global index register B
0184 0038	GBLADDRA	DMA global address register A
0184 003C	GBLADDRB	DMA global address register B
0184 0040	PRICTL1	DMA channel 1 primary control
0184 0044	PRICTL3	DMA channel 3 primary control
0184 0048	SECCTL1	DMA channel 1 secondary control
0184 004C	SECCTL3	DMA channel 3 secondary control
0184 0050	SRC1	DMA channel 1 source address
0184 0054	SRC3	DMA channel 3 source address
0184 0058	DST1	DMA channel 1 destination address
0184 005C	DST3	DMA channel 3 destination address
0184 0060	XFRCNT1	DMA channel 1 transfer counter
0184 0064	XFRCNT3	DMA channel 3 transfer counter
0184 0068	GBLADDRC	DMA global address register C
0184 006C	GBLADDRD	DMA global address register D
0184 0070	AUXCTL	DMA auxiliary control register
0184 0074 – 0187 FFFF	–	Reserved

peripheral register descriptions (continued)

Table 5. Expansion Bus Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0188 0000	XBGC	Expansion bus global control register	
0188 0004	XCECTL1	XCE1 space control register	Corresponds to expansion bus XCE0 memory space: [4000 0000 – 4FFF FFFF]
0188 0008	XCECTL0	XCE0 space control register	Corresponds to expansion bus XCE1 memory space: [5000 0000 – 5FFF FFFF]
0188 000C	XBHC	Expansion bus host port interface control register	DSP read/write access only
0188 0010	XCECTL2	XCE2 space control register	Corresponds to expansion bus XCE2 memory space: [6000 0000 – 6FFF FFFF]
0188 0014	XCECTL3	XCE3 space control register	Corresponds to expansion bus XCE3 memory space: [7000 0000 – 7FFF FFFF]
0188 0018	–	Reserved	
0188 001C	–	Reserved	
0188 0020	XBIMA	Expansion bus internal master address register	DSP read/write access only
0188 0024	XBEA	Expansion bus external address register	DSP read/write access only
0188 0028 – 018B FFFF	–	Reserved	
–	XBISA	Expansion bus internal slave address	
–	XBD	Expansion bus data	

Table 6. Interrupt Selector Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
019C 0000	MUXH	Interrupt multiplexer high	Selects which interrupts drive CPU interrupts 10–15 (INT10–INT15)
019C 0004	MUXL	Interrupt multiplexer low	Selects which interrupts drive CPU interrupts 4–9 (INT04–INT09)
019C 0008	EXTPOL	External interrupt polarity	Sets the polarity of the external interrupts (EXT_INT4–EXT_INT7)
019C 000C – 019C 01FF	–	Reserved	
019C 0200	PDCTL	Peripheral power-down control register	
019C 0204 – 019F FFFF	–	Reserved	

Table 7. Peripheral Power-Down Control Register

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

peripheral register descriptions (continued)

Table 8. McBSP 0 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
018C 0000	DRR0	McBSP0 data receive register	The CPU and DMA/EDMA controller can only read this register; they cannot write to it.
018C 0004	DXR0	McBSP0 data transmit register	
018C 0008	SPCR0	McBSP0 serial port control register	
018C 000C	RCR0	McBSP0 receive control register	
018C 0010	XCR0	McBSP0 transmit control register	
018C 0014	SRGR0	McBSP0 sample rate generator register	
018C 0018	MCR0	McBSP0 multichannel control register	
018C 001C	RCER0	McBSP0 receive channel enable register	
018C 0020	XCER0	McBSP0 transmit channel enable register	
018C 0024	PCR0	McBSP0 pin control register	
018C 0028 – 018F FFFF	–	Reserved	

Table 9. McBSP 1 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
0190 0000	DRR1	Data receive register	The CPU and DMA/EDMA controller can only read this register; they cannot write to it.
0190 0004	DXR1	McBSP1 data transmit register	
0190 0008	SPCR1	McBSP1 serial port control register	
0190 000C	RCR1	McBSP1 receive control register	
0190 0010	XCR1	McBSP1 transmit control register	
0190 0014	SRGR1	McBSP1 sample rate generator register	
0190 0018	MCR1	McBSP1 multichannel control register	
0190 001C	RCER1	McBSP1 receive channel enable register	
0190 0020	XCER1	McBSP1 transmit channel enable register	
0190 0024	PCR1	McBSP1 pin control register	
0190 0028 – 0193 FFFF	–	Reserved	

peripheral register descriptions (continued)

Table 10. McBSP 2 Registers

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME	COMMENTS
01A4 0000	DRR2	McBSP2 data receive register	The CPU and DMA/EDMA controller can only read this register; they cannot write to it.
01A4 0004	DXR2	McBSP2 data transmit register	
01A4 0008	SPCR2	McBSP2 serial port control register	
01A4 000C	RCR2	McBSP2 receive control register	
01A4 0010	XCR2	McBSP2 transmit control register	
01A4 0014	SRGR2	McBSP2 sample rate generator register	
01A4 0018	MCR2	McBSP2 multichannel control register	
01A4 001C	RCER2	McBSP2 receive channel enable register	
01A4 0020	XCER2	McBSP2 transmit channel enable register	
01A4 0024	PCR2	McBSP2 pin control register	
01A4 0028 – 01A7 FFFF	–	Reserved	

Table 11. Timer 0 Registers

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

Table 12. Timer 1 Registers

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

DMA synchronization events

The C6203 DMA supports up to four independent programmable DMA channels, plus an auxiliary channel used for servicing the HPI module. The four main DMA channels can be read/write synchronized based on the events shown in Table 13. Selection of these events is done via the RSYNC and WSYNC fields in the Primary Control registers of the specific DMA channel. For more detailed information on the DMA module, associated channels, and event-synchronization, see the Direct Memory Access (DMA) Controller chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

Table 13. 320C6203 DMA Synchronization Events

DMA EVENT NUMBER (BINARY)	EVENT NAME	EVENT DESCRIPTION
00000	Reserved	Reserved
00001	TINT0	Timer 0 interrupt
00010	TINT1	Timer 1 interrupt
00011	SD_INT	EMIF SDRAM timer interrupt
00100	EXT_INT4	External interrupt pin 4
00101	EXT_INT5	External interrupt pin 5
00110	EXT_INT6	External interrupt pin 6
00111	EXT_INT7	External interrupt pin 7
01000	DMA_INT0	DMA channel 0 interrupt
01001	DMA_INT1	DMA channel 1 interrupt
01010	DMA_INT2	DMA channel 2 interrupt
01011	DMA_INT3	DMA channel 3 interrupt
01100	XEVT0	McBSP0 transmit event
01101	REVT0	McBSP0 receive event
01110	XEVT1	McBSP1 transmit event
01111	REVT1	McBSP1 receive event
10000	DSP_INT	Host processor-to-DSP interrupt
10001	XEVT2	McBSP2 transmit event
10010	REVT2	McBSP2 receive event
10011 – 11111	Reserved	Reserved. Not used.

interrupt sources and interrupt selector

The C62x DSP core supports 16 prioritized interrupts, which are listed in Table 14. The highest-priority interrupt is INT_00 (dedicated to RESET) while the lowest-priority interrupt is INT_15. The first four interrupts (INT_00–INT_03) are non-maskable and fixed. The remaining interrupts (INT_04–INT_15) are maskable and default to the interrupt source specified in Table 14. The interrupt source for interrupts 4–15 can be programmed by modifying the selector value (binary value) in the corresponding fields of the Interrupt Selector Control registers: MUXH (address 0x019C0000) and MUXL (address 0x019C0004).

Table 14. C6203 DSP Interrupts

CPU INTERRUPT NUMBER	INTERRUPT SELECTOR CONTROL REGISTER	SELECTOR VALUE (BINARY)	INTERRUPT EVENT	INTERRUPT SOURCE
INT_00†	–	–	RESET	
INT_01†	–	–	NMI	
INT_02†	–	–	Reserved	Reserved. Do not use.
INT_03†	–	–	Reserved	Reserved. Do not use.
INT_04‡	MUXL[4:0]	00100	EXT_INT4	External interrupt pin 4
INT_05‡	MUXL[9:5]	00101	EXT_INT5	External interrupt pin 5
INT_06‡	MUXL[14:10]	00110	EXT_INT6	External interrupt pin 6
INT_07‡	MUXL[20:16]	00111	EXT_INT7	External interrupt pin 7
INT_08‡	MUXL[25:21]	01000	DMA_INT0	DMA channel 0 interrupt
INT_09‡	MUXL[30:26]	01001	DMA_INT1	DMA channel 1 interrupt
INT_10‡	MUXH[4:0]	00011	SD_INT	EMIF SDRAM timer interrupt
INT_11‡	MUXH[9:5]	01010	DMA_INT2	DMA channel 2 interrupt
INT_12‡	MUXH[14:10]	01011	DMA_INT3	DMA channel 3 interrupt
INT_13‡	MUXH[20:16]	00000	DSP_INT	Host-processor-to-DSP interrupt
INT_14‡	MUXH[25:21]	00001	TINT0	Timer 0 interrupt
INT_15‡	MUXH[30:26]	00010	TINT1	Timer 1 interrupt
–	–	01100	XINT0	McBSP0 transmit interrupt
–	–	01101	RINT0	McBSP0 receive interrupt
–	–	01110	XINT1	McBSP1 transmit interrupt
–	–	01111	RINT1	McBSP1 receive interrupt
–	–	10000	Reserved	Reserved. Not used.
–	–	10001	XINT2	McBSP2 transmit interrupt
–	–	10010	RINT2	McBSP2 receive interrupt
–	–	10011 – 11111	Reserved	Reserved. Do not use.

† Interrupts INT_00 through INT_03 are non-maskable and fixed.
‡ Interrupts INT_04 through INT_15 are programmable by modifying the binary selector values in the Interrupt Selector Control registers fields. Table 14 shows the default interrupt sources for Interrupts INT_04 through INT_15. For more detailed information on interrupt sources and selection, see the Interrupt Selector and External Interrupts chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190).

signal groups description

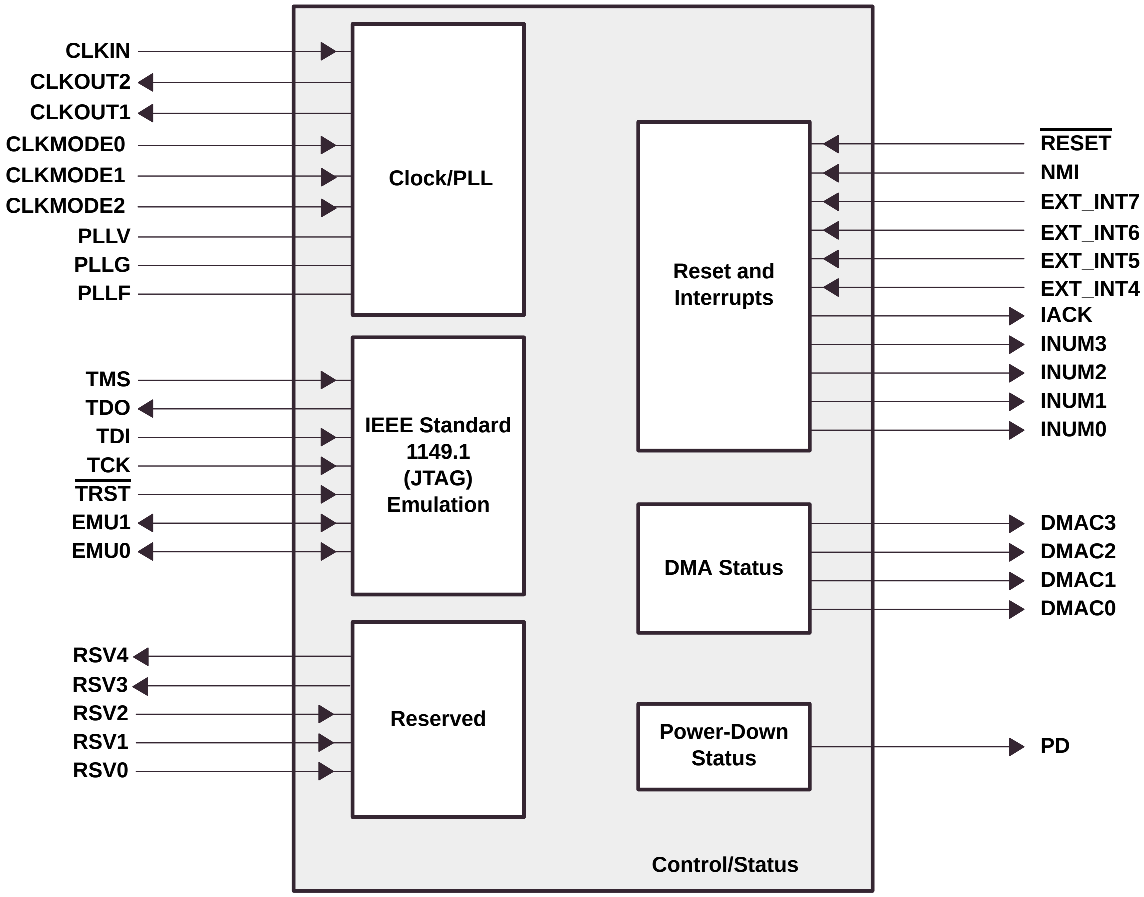


Figure 2. CPU (DSP Core) Signals

signal groups description (continued)

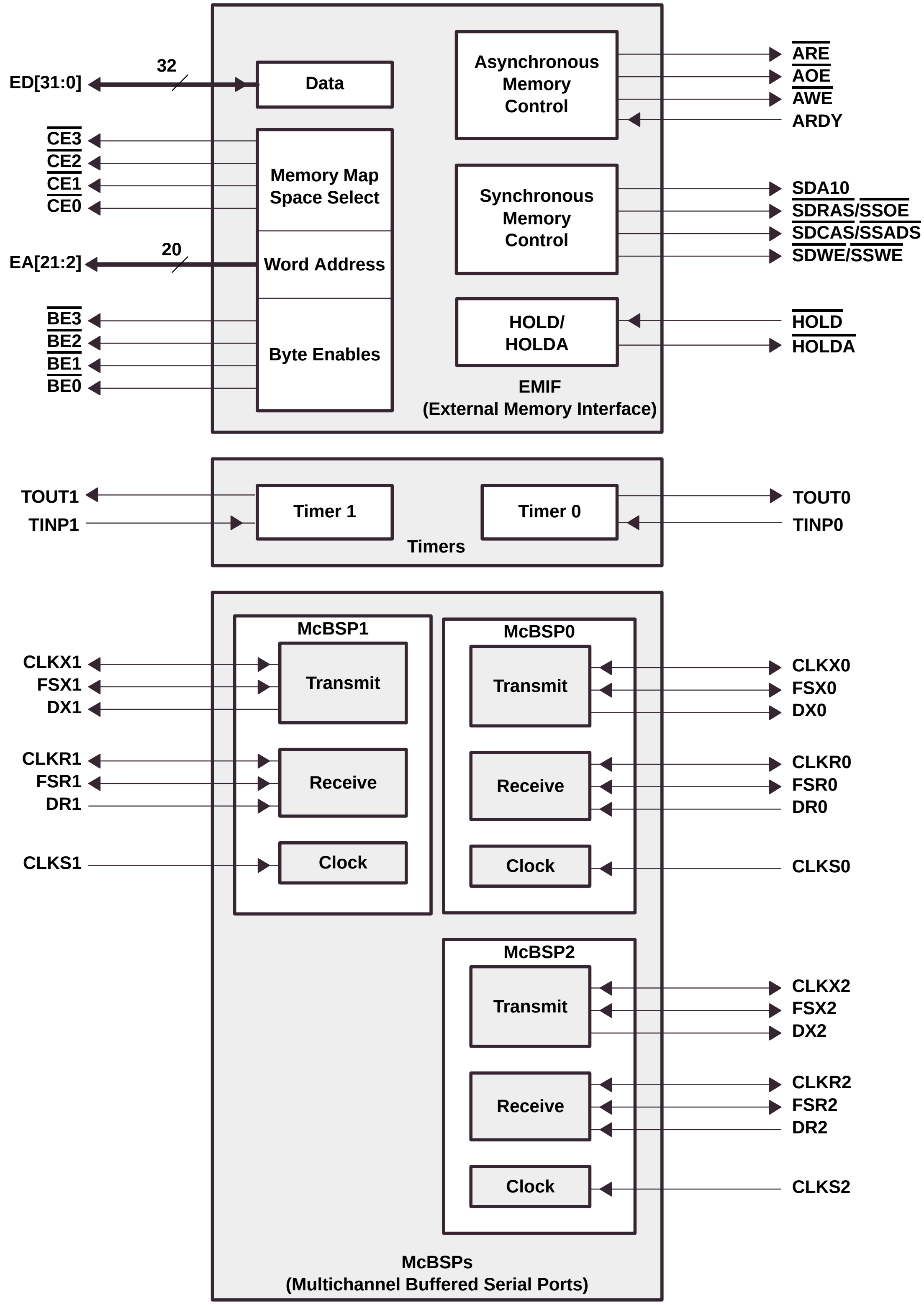


Figure 3. Peripheral Signals

signal groups description (continued)

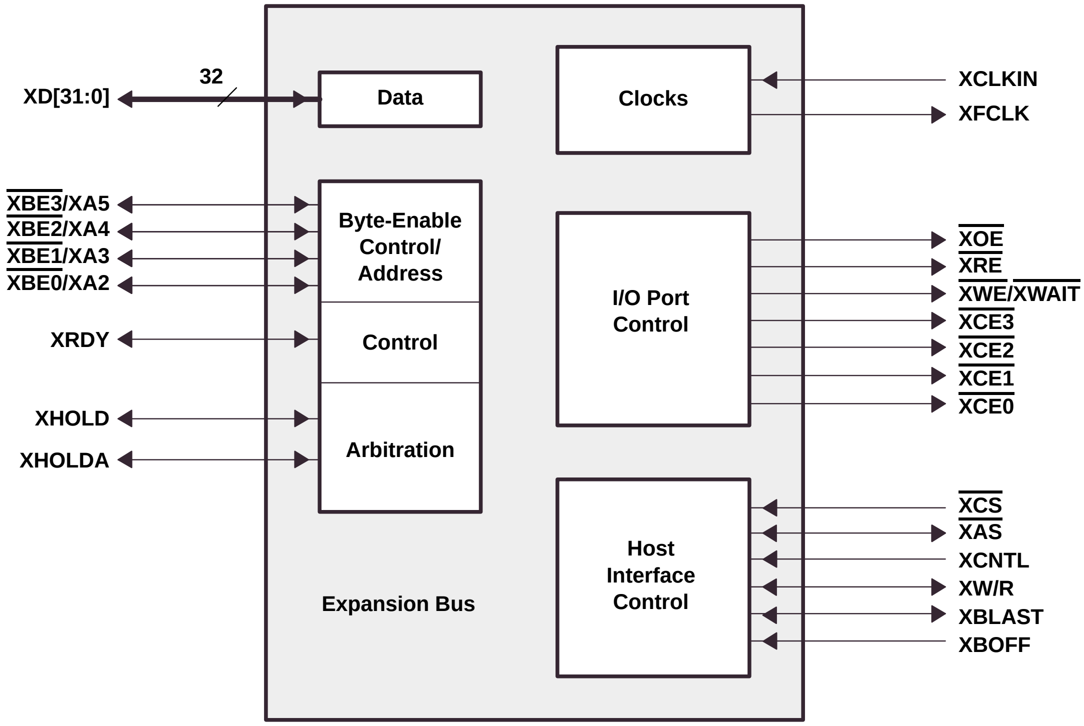


Figure 3. Peripheral Signals (Continued)

Signal Descriptions

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
CLOCK/PLL			
CLKIN	D10	I	Clock input
CLKOUT1	Y17	O	Clock output at full device speed
CLKOUT2	Y16	O	Clock output at half (1/2) of device speed • Used for synchronous memory interface
CLKMODE0	C12	I	Clock mode selects • Selects what multiply factors of the input clock frequency the CPU frequency equals. For more details on the CLKMODE pins and the PLL multiply factors for the C6203 device, see the *Clock PLL* section of this data sheet.
CLKMODE1	G10	I	
CLKMODE2	G12	I	
PLLV‡	B11	A§	PLL analog V_{CC} connection for the low-pass filter
PLLG‡	A11	A§	PLL analog GND connection for the low-pass filter
PLLF‡	G11	A§	PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION			
TMS	W5	I	JTAG test-port mode select (features an internal pullup)
TDO	R8	O/Z	JTAG test-port data out
TDI	W4	I	JTAG test-port data in (features an internal pullup)
TCK	V5	I	JTAG test-port clock
$\overline{\text{TRST}}$	R7	I	JTAG test-port reset (features an internal pulldown)
EMU1	T7	I/O/Z	Emulation pin 1, pullup with a dedicated 20-kΩ resistor¶
EMU0	Y5	I/O/Z	Emulation pin 0, pullup with a dedicated 20-kΩ resistor¶
RESET AND INTERRUPTS			
$\overline{\text{RESET}}$	J4	I	Device reset
NMI	K2	I	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	R4	I	External interrupts • Edge-driven • Polarity independently selected via the External Interrupt Polarity Register bits (EXTPOL.[3:0])
EXT_INT6	P6		
EXT_INT5	T2		
EXT_INT4	T3		
IACK	R2	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	P4	O	Active interrupt identification number • Valid during IACK for all active interrupts (not just external) • Encoding order follows the interrupt-service fetch-packet ordering
INUM2	P1		
INUM1	P2		
INUM0	N6		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground
‡ PLLV, PLLG, and PLLF are not part of external voltage supply or ground. See the *clock PLL* section for information on how to connect these pins.
§ A = Analog Signal (PLL Filter)
¶ For emulation and normal operation, pull up EMU1 and EMU0 with a dedicated 20-kΩ resistor. For boundary scan, pull down EMU1 and EMU0 with a dedicated 20-kΩ resistor.

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GNP	TYPE†	DESCRIPTION
POWER-DOWN STATUS			
PD	V3	O	Power-down modes 2 or 3 (active if high)
EXPANSION BUS			
XCLKIN	C9	I	Expansion bus synchronous host interface clock input
XFCLK	B9	O	Expansion bus FIFO interface clock output
XD31	D11	I/O/Z	Expansion bus data • Used for transfer of data, address, and control • Also controls initialization of DSP modes and expansion bus at reset [Note: For more information on pin control and boot configuration fields, see the Boot Modes and Configuration chapter of the *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190)] XD[30:16]– $\overline{\text{XCE[3:0]}}$ memory type XD13 – XBLAST polarity XD12 – XW/R polarity XD11 – Asynchronous or synchronous host operation XD10 – Arbitration mode (internal or external) XD9 – FIFO mode XD8 – Little endian/big endian XD7 – SCRT select XD[4:0] – Boot mode All other expansion bus data pins not listed should be pulled down. For proper operation, XD7 must be pulled down with a 10-kΩ resistor. The board design should be wired such that a pullup or pulldown resistor can be used on XD7 for future applications.
XD30	B13		
XD29	F12		
XD28	C13		
XD27	D12		
XD26	A14		
XD25	B14		
XD24	F13		
XD23	B15		
XD22	C15		
XD21	D13		
XD20	B16		
XD19	B17		
XD18	D14		
XD17	F15		
XD16	C17		
XD15	G14		
XD14	D17		
XD13	C18		
XD12	E18		
XD11	D18		
XD10	G15		
XD9	D19		
XD8	F16		
XD7	F19		
XD6	E20		
XD5	G16		
XD4	H19		
XD3	G20		
XD2	J18		
XD1	H20		
XD0	H21		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
EXPANSION BUS (CONTINUED)			
$\overline{XCE3}$	D3	O/Z	Expansion bus I/O port memory space enables • Enabled by bits 28, 29, and 30 of the word address • Only one asserted during any I/O port data access
$\overline{XCE2}$	G6		
$\overline{XCE1}$	D4		
$\overline{XCE0}$	E4		
$\overline{XBE3}$/XA5	F6	I/O/Z	Expansion bus multiplexed byte-enable control/address signals • Act as byte-enable for host-port operation • Act as address for I/O port operation
$\overline{XBE2}$/XA4	F7		
$\overline{XBE1}$/XA3	B5		
$\overline{XBE0}$/XA2	C7		
$\overline{XOE}$	B7	O/Z	Expansion bus I/O port output-enable
$\overline{XRE}$	B8	O/Z	Expansion bus I/O port read-enable
$\overline{XWE}$/$\overline{XWAIT}$	D7	O/Z	Expansion bus I/O port write-enable and host-port wait signals
$\overline{XCS}$	D8	I	Expansion bus host-port chip-select input
$\overline{XAS}$	G9	I/O/Z	Expansion bus host-port address strobe
XCNTL	A9	I	Expansion bus host control. XCNTL selects between expansion bus address or data register.
XW/R	F9	I/O/Z	Expansion bus host-port write/read-enable. XW/R polarity is selected at reset.
XRDY	F4	I/O/Z	Expansion bus host-port ready (active low) and I/O port ready (active high)
XBLAST	C5	I/O/Z	Expansion bus host-port burst last-polarity selected at reset
XBOFF	C10	I	Expansion bus back off
XHOLD	C4	I/O/Z	Expansion bus hold request
XHOLDA	D6	I/O/Z	Expansion bus hold acknowledge
EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY			
$\overline{CE3}$	V18	O/Z	Memory space enables • Enabled by bits 24 and 25 of the word address • Only one asserted during any external data access
$\overline{CE2}$	W18		
$\overline{CE1}$	T15		
$\overline{CE0}$	U18		
$\overline{BE3}$	R15	O/Z	Byte-enable control • Decoded from the two lowest bits of the internal address • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{BE2}$	V19		
$\overline{BE1}$	U20		
$\overline{BE0}$	V16		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
EMIF – ADDRESS			
EA21	K18	O/Z	External address (word address)
EA20	K16		
EA19	J20		
EA18	K19		
EA17	J21		
EA16	K20		
EA15	M19		
EA14	L16		
EA13	K21		
EA12	M18		
EA11	L21		
EA10	N18		
EA9	M20		
EA8	M16		
EA7	R18		
EA6	M21		
EA5	N21		
EA4	N16		
EA3	P20		
EA2	T18		
EMIF – DATA			
ED31	V6	I/O/Z	External data
ED30	Y6		
ED29	T8		
ED28	Y7		
ED27	Y8		
ED26	V7		
ED25	T9		
ED24	AA8		
ED23	V8		
ED22	Y9		
ED21	AA9		
ED20	V9		
ED19	T10		
ED18	Y10		
ED17	W9		
ED16	V10		
ED15	T11		
ED14	AA10		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
EMIF – DATA (CONTINUED)			
ED13	W10	I/O/Z	External data
ED12	W12		
ED11	Y11		
ED10	Y12		
ED9	T12		
ED8	AA13		
ED7	R12		
ED6	V13		
ED5	Y13		
ED4	Y14		
ED3	T13		
ED2	Y15		
ED1	R13		
ED0	V14		
EMIF – ASYNCHRONOUS MEMORY CONTROL			
$\overline{\text{ARE}}$	T20	O/Z	Asynchronous memory read-enable
$\overline{\text{AOE}}$	P16	O/Z	Asynchronous memory output-enable
$\overline{\text{AWE}}$	R20	O/Z	Asynchronous memory write-enable
ARDY	R16	I	Asynchronous memory ready input
EMIF – SYNCHRONOUS DRAM (SDRAM)/SYNCHRONOUS BURST SRAM (SBSRAM) CONTROL			
SDA10	T14	O/Z	SDRAM address 10 (separate for deactivate command)
$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$	V17	O/Z	SDRAM column-address strobe/SBSRAM address strobe
$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$	W17	O/Z	SDRAM row-address strobe/SBSRAM output-enable
$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$	W15	O/Z	SDRAM write-enable/SBSRAM write-enable
EMIF – BUS ARBITRATION			
$\overline{\text{HOLD}}$	T19	I	Hold request from the host
$\overline{\text{HOLDA}}$	T16	O	Hold-request-acknowledge to the host
TIMER 0			
TOUT0	F2	O	Timer 0 or general-purpose output
TINP0	E2	I	Timer 0 or general-purpose input
TIMER 1			
TOUT1	G4	O	Timer 1 or general-purpose output
TINP1	H6	I	Timer 1 or general-purpose input
DMA ACTION COMPLETE STATUS			
DMAC3	R6	O	DMA action complete
DMAC2	U2		
DMAC1	T6		
DMAC0	V4		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)			
CLKS0	K6	I	External clock source (as opposed to internal)
CLKR0	L1	I/O/Z	Receive clock
CLKX0	K3	I/O/Z	Transmit clock
DR0	M1	I	Receive data
DX0	L6	O/Z	Transmit data
FSR0	L2	I/O/Z	Receive frame sync
FSX0	L3	I/O/Z	Transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)			
CLKS1	G2	I	External clock source (as opposed to internal)
CLKR1	H2	I/O/Z	Receive clock
CLKX1	H4	I/O/Z	Transmit clock
DR1	J2	I	Receive data
DX1	H3	O/Z	Transmit data
FSR1	J6	I/O/Z	Receive frame sync
FSX1	J1	I/O/Z	Transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP2)			
CLKS2	L4	I	External clock source (as opposed to internal)
CLKR2	M2	I/O/Z	Receive clock
CLKX2	N4	I/O/Z	Transmit clock
DR2	P3	I	Receive data
DX2	N2	O/Z	Transmit data
FSR2	M6	I/O/Z	Receive frame sync
FSX2	N1	I/O/Z	Transmit frame sync
RESERVED FOR TEST			
RSV0	K1	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV1	F3	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV2	A10	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV3	F11	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV4	D9	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
N/C	R11	-	No connect
	R9	-	
	W7	-	

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS			
DV_{DD} - 3.3 V	C8 C14 E3 E19 H9 H11 H13 J3 J8 J10 J12 J14 J19 K7 K9 K11 K13 K15 L8 L10 L12 L14 M7 M9 M11 M13 M15 N3 N8 N10 N12 N14 N19 P9 P11 P13 U3 U19 W8 W14	S	3.3-V supply voltage (I/O)

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
	A3		
	A5		
	A7		
	A12		
	A13		
	A16		
	A18		
	B2		
	B4		
	B6		
	B10		
	B12		
	B19		
	C1		
	C3		
	C20		
	D2		
	D15		
	D16		
	D21		
CV_{DD} - 1.5 V	E1	S	1.5-V supply voltage (core)
	E6		
	E8		
	E10		
	E12		
	E14		
	E16		
	F5		
	F8		
	F10		
	F14		
	F17		
	F20		
	F21		
	G1		
	G7		
	G8		
	G13		
	G18		
	H5		
	H16		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD} - 1.5 V	H17 H18 K4 K5 K17 L18 L19 L20 M3 M4 M5 M17 N20 P5 P17 P18 P19 R10 R14 R21 T1 T5 T17 U4 U6 U8 U10 U12 U14 U16 U21 V1 V11 V12 V15 V20 W2 W13 W19 W21 Y3	S	1.5-V supply voltage (core)

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD} - 1.5 V	Y18	S	1.5-V supply voltage (core)
	Y20		
	AA4		
	AA6		
	AA11		
	AA12		
	AA15		
	AA17		
	AA19		
GROUND PINS			
V_{SS}	A4	GND	Ground pins
	A6		
	A8		
	A15		
	A17		
	A19		
	B3		
	B18		
	B20		
	C2		
	C6		
	C11		
	C16		
	C19		
	C21		
	D1		
	D5		
	D20		
	E5		
	E7		
	E9		
	E11		
	E13		
	E15		
	E17		
	E21		
	F1		
	F18		
	G3		
	G5		
	G17		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	G19	GND	Ground pins
	G21		
	H1		
	H7		
	H8		
	H10		
	H12		
	H14		
	H15		
	J5		
	J7		
	J9		
	J11		
	J13		
	J15		
	J16		
	J17		
	K8		
	K10		
	K12		
	K14		
	L5		
	L7		
	L9		
	L11		
	L13		
	L15		
	L17		
	M8		
	M10		
	M12		
	M14		
	N5		
	N7		
	N9		
	N11		
	N13		
	N15		
	N17		
	P7		
	P8		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GLP	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	P10	GND	Ground pins
	P12		
	P14		
	P15		
	P21		
	R1		
	R3		
	R5		
	R17		
	R19		
	T4		
	T21		
	U1		
	U5		
	U7		
	U9		
	U11		
	U13		
	U15		
	U17		
	V2		
	V21		
	W1		
	W3		
	W6		
	W11		
	W16		
	W20		
	Y2		
	Y4		
	Y19		
	AA3		
	AA5		
	AA7		
	AA14		
	AA16		
	AA18		

† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

development support

TI offers an extensive line of development tools for the TMS320C6000™ DSP platform, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules.

The following products support development of C6000™ DSP-based applications:

Software Development Tools:
Code Composer Studio™ Integrated Development Environment (IDE) including Editor
C/C++/Assembly Code Generation, and Debug plus additional development tools
Scalable, Real-Time Foundation Software (DSP/BIOS™), which provides the basic run-time target software needed to support any DSP application.

Hardware Development Tools:
Extended Development System (XDS™) Emulator (supports C6000™ DSP multiprocessor system debug)
EVM (Evaluation Module)

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

For a complete listing of development-support tools for the TMS320C6000™ DSP platform, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL) and select “Find Development Tools”. For device-specific tools, under “Semiconductor Products” 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.

development support (Continued)

device and development-support tool nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all SMJ320™ DSP devices and support tools. Each SMJ320™ DSP commercial family member has one of three prefixes: SMX, SM, or SMJ. Texas Instruments recommends two of three possible prefix designators for support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (SMX/TMDX) through fully qualified production devices/tools (SMJ/TMDS).

Device development evolutionary flow:

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

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

SMJ Fully qualified production device processed to MIL-PRF-38535

Support tool development evolutionary flow:

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

TMDS Fully qualified development-support product

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

“Developmental product is intended for internal evaluation purposes.”

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

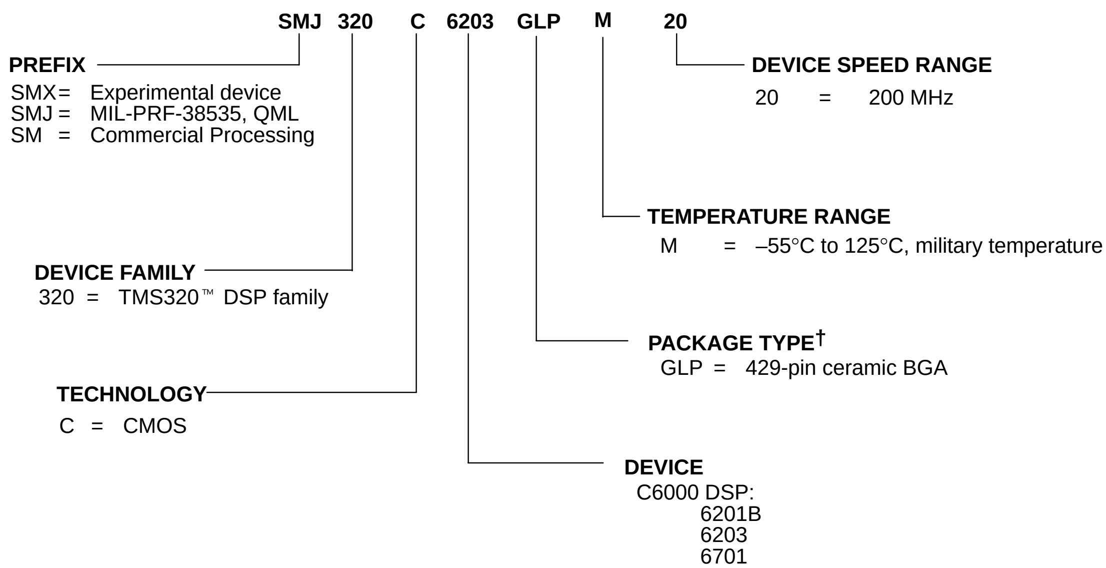
Predictions show that prototype devices (SMX or SM) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices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

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, GLP), the temperature range, and the device speed range in megahertz (for example, 20 is 200 MHz).

Figure 4 provides a legend for reading the complete device name. For the C6203 device orderable part numbers (P/Ns), see the Texas Instruments web site on the Worldwide web at http://www.ti.com URL, or contact the nearest TI field sales office, or authorized distributor.

TMS320 is a trademark of Texas Instruments.

device and development-support tool nomenclature (continued)



† BGA = Ball Grid Array

Figure 4. SMJ320C6000™ DSP Platform Device Nomenclature

documentation support

Extensive documentation supports all SMJ320™ DSP family 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™ CPU (DSP 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), 32-/16-bit host-port interfaces (HPIs), multichannel buffered serial ports (McBSPs), direct memory access (DMA), enhanced direct-memory-access (EDMA) controller, expansion bus, peripheral component interconnect (PCI), clocking and phase-locked loop (PLL); 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 TMS320C62x™/TMS320C67x™ devices, associated development tools, and third-party support.

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

TMS320C67x is a trademark of Texas Instruments.

clock PLL

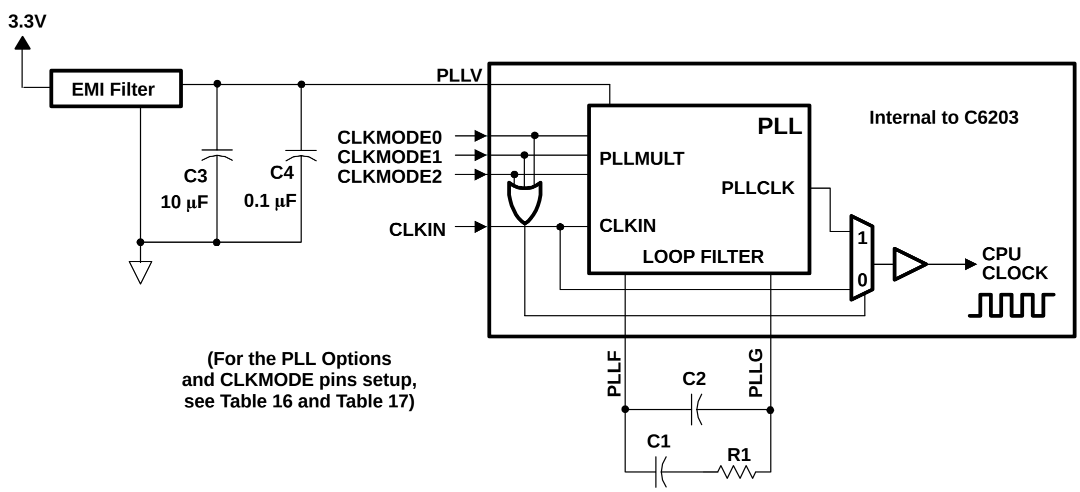
Most of the internal C6203 clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock in frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock.

To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 5, and Table 16 through Table 17 show the external PLL circuitry for either x1 (PLL bypass) or x4 PLL multiply modes. Figure 6 shows the external PLL circuitry for a system with ONLY x1 (PLL bypass) mode.

To minimize the clock jitter, a single clean power supply should power both the C6203 device and the external clock oscillator circuit. Noise coupling into PLLF directly impacts PLL clock jitter. 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. Table 15 lists some examples of compatible CLKIN external clock sources:

Table 15. 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



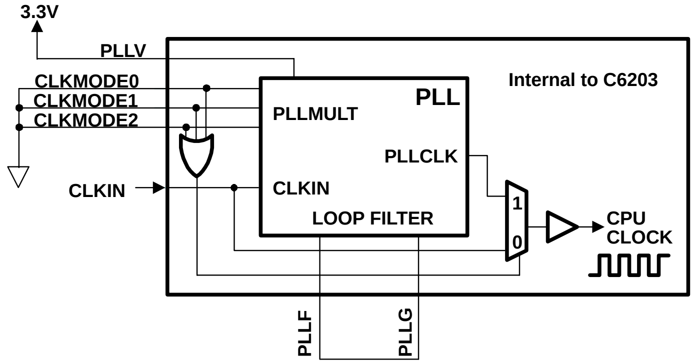
NOTES: A. Keep the lead length and the number of vias between pin PLLF, pin PLLG, R1, C1, and C2 to a minimum. In addition, place all PLL components (R1, C1, C2, C3, C4, and EMI Filter) as close to the C6000™ DSP device as possible. Best performance is achieved with the PLL components on a single side 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 (R1, C1, C2, 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}.

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

clock PLL (continued)



NOTES: A. For a system with ONLY PLL x1 (bypass) mode, short the PLLF to PLLG.
B. The 3.3-V supply for PLLV must be from the same 3.3-V power plane supplying the I/O voltage, DV_{DD}.

Figure 6. External PLL Circuitry for x1 (Bypass) PLL Mode Only

Table 16. PLL Multiply and Bypass (x1) Options†

BIT (PIN NO.)	CLKMODE2 (G12)	CLKMODE1 (G10)	CLKMODE0 (C12)	DEVICES AND PLL CLOCK OPTIONS C6203 (GLP)
Value	0	0	0	Bypass (x1)
	0	0	1	x4
	0	1	0	x8
	0	1	1	x10
	1	0	0	x6
	1	0	1	x9
	1	1	0	x7
	1	1	1	x11

† f(CPU Clock) = f(CLKIN) x (PLL mode)

Table 17. SMJ320C6203 PLL Component Selection Table†

CLKMODE	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 [±1%] (Revision No.)	C1 [±10%] (Revision No.)	C2 [±10%] (Revision No.)	TYPICAL LOCK TIME (µs)
x4	32.5–75	130–300	65–150	45.3 Ω	47 nF	10 pF	75
x6	21.7–50						
x7	18.6–42.9						
x8	16.3–37.5						
x9	14.4–33.3						
x10	13–30						
x11	11.8–27.3						

† 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

For systems using the C6000™ DSP platform of devices, the core supply may be required to provide in excess of 2 A per DSP until the I/O supply is powered up. This extra current condition is a result of uninitialized logic within the DSP(s) and is corrected once the CPU sees an internal clock pulse. With the PLL enabled, as the I/O supply is powered on, a clock pulse is produced stopping the extra current draw from the supply. With the PLL disabled, as many as five external clock cycle pulses may be required to stop this extra current draw. A normal current state returns once the I/O power supply is turned on and the CPU sees a clock pulse. Decreasing the amount of time between the core supply power up and the I/O supply power up can minimize the effects of this current draw.

A dual-power supply with simultaneous sequencing, such as available with TPS563xx controllers or PT69xx plug-in power modules, can be used to eliminate the delay between core and I/O power up [see the *Using the TPS56300 to Power DSPs* application report (literature number SLVA088)]. A Schottky diode can also be used to tie the core rail to the I/O rail, effectively pulling up the I/O power supply to a level that can help initialize the logic within the DSP.

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 ranges (unless otherwise noted)†

Supply voltage range, CV_{DD} (see Note 1)	– 0.3 V to 1.8 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 ranges, T_C	–55°C to 125°C
Storage temperature range, T_{stg}	–65°C to 150°C
Temperature cycle range, (1000-cycle performance)	–55°C to 125°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

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

recommended operating conditions

		MIN	NOM	MAX	UNIT
CV_{DD}	Supply voltage, Core	1.43	1.5	1.57	V
DV_{DD}	Supply voltage, I/O	3.14	3.3	3.46	V
V_{SS}	Supply ground	0	0	0	V
V_{IH}	High-level input voltage‡	2			V
V_{IL}	Low-level input voltage§			0.8	V
I_{OH}	High-level output current			–8	mA
I_{OL}	Low-level output current			8	mA
T_C	Operating case temperature	–55		125	°C

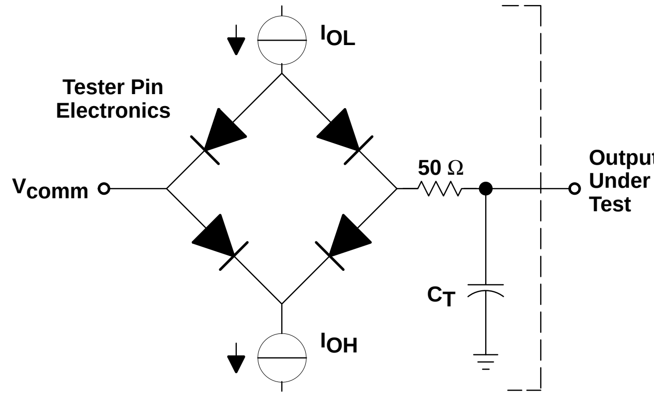
‡ VIH is not production tested for: CLKMODE [2:0], CLKIN, XCLKIN, XCS.
§ VIL is not production tested for: CLKIN, TRST.

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			0.6	V
I_I	Input current#	$V_I = V_{SS}$ to DV_{DD}				±10	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 = 200 MHz		340		mA
I_{DD2V}	Supply current, peripherals☆	CV_{DD} = NOM,	CPU clock = 200 MHz		235		mA
I_{DD3V}	Supply current, I/O pins☆	CV_{DD} = NOM,	CPU clock = 200 MHz		45		mA
C_i	Input capacitance					12	pF
C_o	Output capacitance					15	pF

¶ VOH and VOL are not production tested for: CLKOUT1, EMU0, EMU1.
# TMS and TDI are not included due to internal pullups. TRST is not included due to internal pulldown.
‖ TDO is not production tested.
☆ Measured with average activity (50% high / 50% low power). For more details on CPU, peripheral, and I/O activity, see the *TMS320C6000 Power Consumption Summary* application report (literature number SPRA486).

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 2 mA
I_{OH} = 2 mA
V_{comm} = 2.1 V
C_T = 15-pF typical load-circuit capacitance

Figure 7. 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.

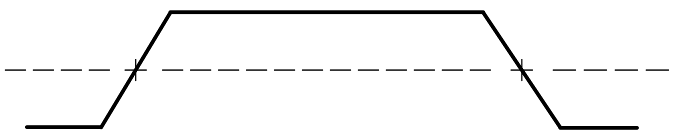


Figure 8. Input and Output Voltage Reference Levels for ac Timing Measurements

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.

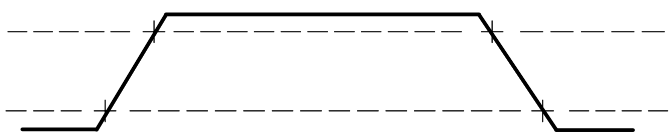


Figure 9. Rise and Fall Transition Time Voltage Reference Levels

PARAMETER MEASUREMENT INFORMATION (CONTINUED)

timing parameters and board routing analysis

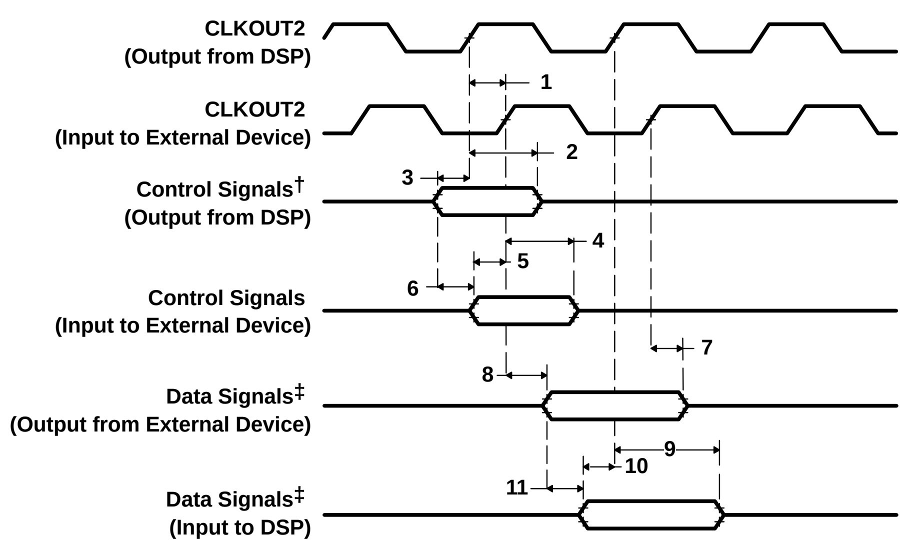
The timing parameter values specified in this data sheet do *not* include delays by board routings. As a good board design practice, such delays must *always* be taken into account. Timing values may be adjusted by increasing/decreasing such delays. TI recommends utilizing the available I/O buffer information 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 18 and Figure 10).

Figure 10 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 18. IBIS Timing Parameters Example (see Figure 10)

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 10. IBIS Input/Output Timings

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN (PLL used)†‡§ (see Figure 11)

NO.			MIN	MAX	UNIT
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	5 x M		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	*0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	*0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		*0.5	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ M = the PLL multiplier factor (x4, x6, x7, x8, x9, x10, or x11).
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

timing requirements for CLKIN [PLL bypassed (x1)]†¶ (see Figure 11)

NO.			MIN	MAX	UNIT
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	5		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	*0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	*0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		*0.6	ns

*This parameter is not production tested.
† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
¶ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns. The maximum CLKIN cycle time in PLL bypass mode (x1) is 200 MHz.

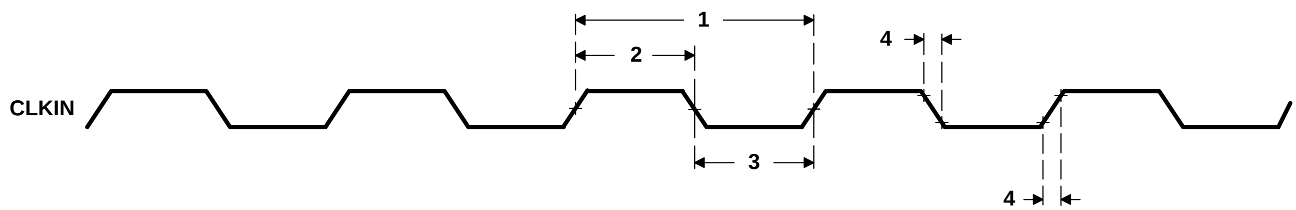


Figure 11. CLKIN Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

timing requirements for XCLKIN† (see Figure 12)

NO.			MIN	MAX	UNIT
1	$t_{c(XCLKIN)}$	Cycle time, XCLKIN	4P		ns
2	$t_{w(XCLKINH)}$	Pulse duration, XCLKIN high	*1.8P		ns
3	$t_{w(XCLKINL)}$	Pulse duration, XCLKIN low	*1.8P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in nanoseconds (ns).

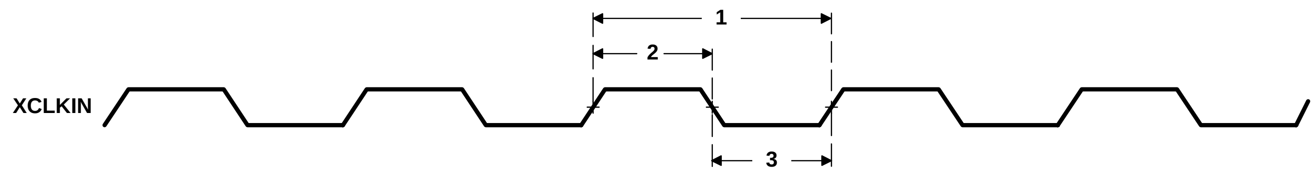


Figure 12. XCLKIN Timings

switching characteristics over recommended operating conditions for CLKOUT2‡§ (see Figure 13)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	*2P – 0.7	*2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	*P – 0.7	*P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	*P – 0.7	*P + 0.7	ns

*This parameter is not production tested.
‡ P = 1/CPU clock frequency in ns.
§ The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.

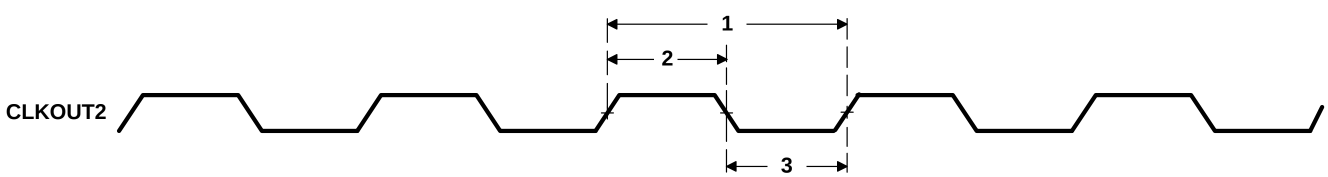


Figure 13. CLKOUT2 Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics over recommended operating conditions for XFCLK†‡ (see Figure 14)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{c(XFCK)}$	Cycle time, XFCLK	*D x P – 0.7	*D x P + 0.7	ns
2	$t_{w(XFCKH)}$	Pulse duration, XFCLK high	*(D/2) x P – 0.7	*(D/2) x P + 0.7	ns
3	$t_{w(XFCKL)}$	Pulse duration, XFCLK low	*(D/2) x P – 0.7	*(D/2) x P + 0.7	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns.
‡ D = 8, 6, 4, or 2; FIFO clock divide ratio, user-programmable

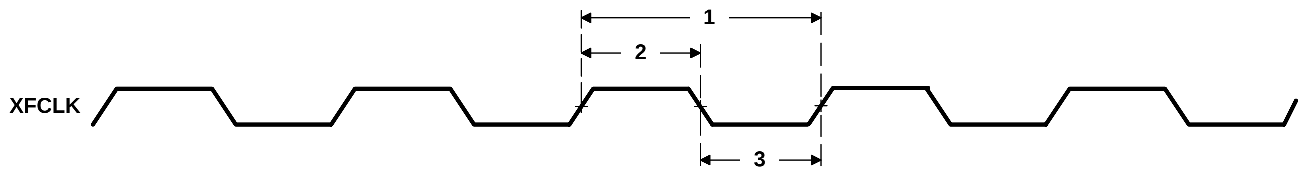


Figure 14. XFCLK Timings

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles§¶#‖ (see Figure 15 – Figure 18)

NO.			MIN	MAX	UNIT
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{\text{ARE}}$ high	1		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{\text{ARE}}$ high	4.9		ns
6	$t_{su(ARDYH-AREL)}$	Setup time, ARDY high before $\overline{\text{ARE}}$ low	–[(RST – 3) x P – 6]		ns
7	$t_{h(AREL-ARDYH)}$	Hold time, ARDY high after $\overline{\text{ARE}}$ low	(RST – 3) x P + 2		ns
9	$t_{su(ARDYL-AREL)}$	Setup time, ARDY low before $\overline{\text{ARE}}$ low	–[(RST – 3) x P – 6]		ns
10	$t_{h(AREL-ARDYL)}$	Hold time, ARDY low after $\overline{\text{ARE}}$ low	(RST – 3) x P + 2		ns
11	$t_{w(ARDYH)}$	Pulse width, ARDY high	*2P		ns
15	$t_{su(ARDYH-AWEL)}$	Setup time, ARDY high before $\overline{\text{AWE}}$ low	–[(WST – 3) x P – 6]		ns
16	$t_{h(AWEL-ARDYH)}$	Hold time, ARDY high after $\overline{\text{AWE}}$ low	(WST – 3) x P + 2		ns
18	$t_{su(ARDYL-AWEL)}$	Setup time, ARDY low before $\overline{\text{AWE}}$ low	–[(WST – 3) x P – 6]		ns
19	$t_{h(AWEL-ARDYL)}$	Hold time, ARDY low after $\overline{\text{AWE}}$ low	(WST – 3) x P + 2		ns

*This parameter is not production tested.
§ To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. If ARDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, ARDY can be an asynchronous input.
¶ 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.
# P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‖ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use ARDY input to extend strobe width.

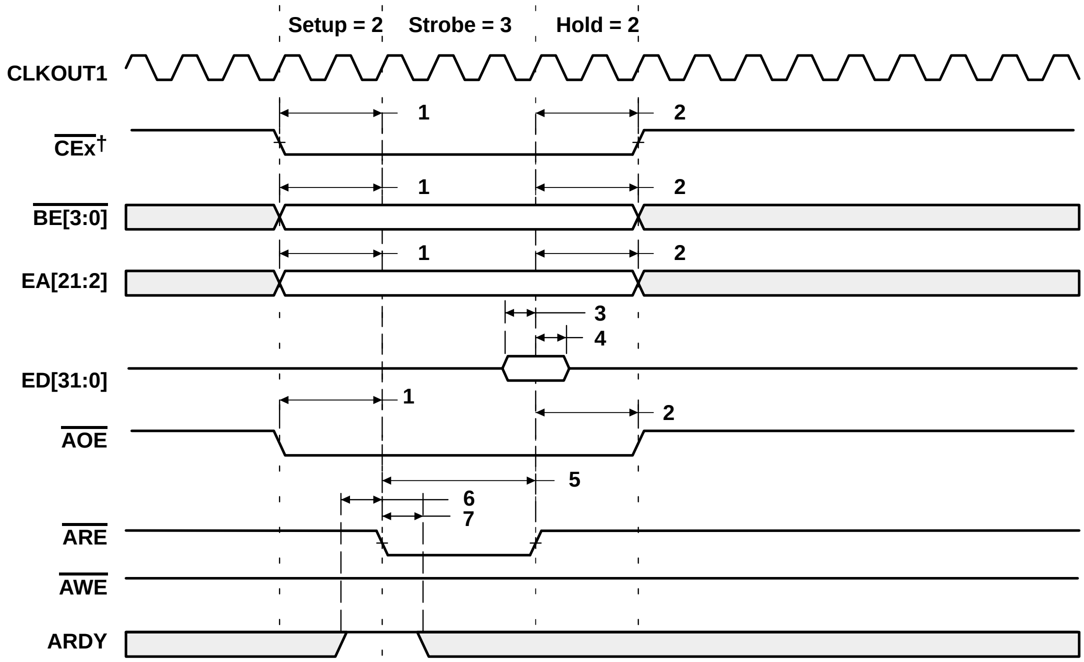
ASYNCHRONOUS MEMORY TIMING (CONTINUED)

switching characteristics over recommended operating conditions for asynchronous memory cycles†‡§¶ (see Figure 15 – Figure 18)

NO.	PARAMETER		MIN	TYP	MAX	UNIT
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS x P – 2			ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	*RH x P – 2			ns
5	$t_{w(AREL)}$	Pulse width, $\overline{ARE}$ low		RST x P		ns
8	$t_{d(ARDYH-AREH)}$	Delay time, ARDY high to $\overline{ARE}$ high	*3P		*4P + 5	ns
12	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS x P – 3			ns
13	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	*WH x P – 2			ns
14	$t_{w(AWEL)}$	Pulse width, $\overline{AWE}$ low		WST x P		ns
17	$t_{d(ARDYH-AWEH)}$	Delay time, ARDY high to $\overline{AWE}$ high	*3P		*4P + 5	ns

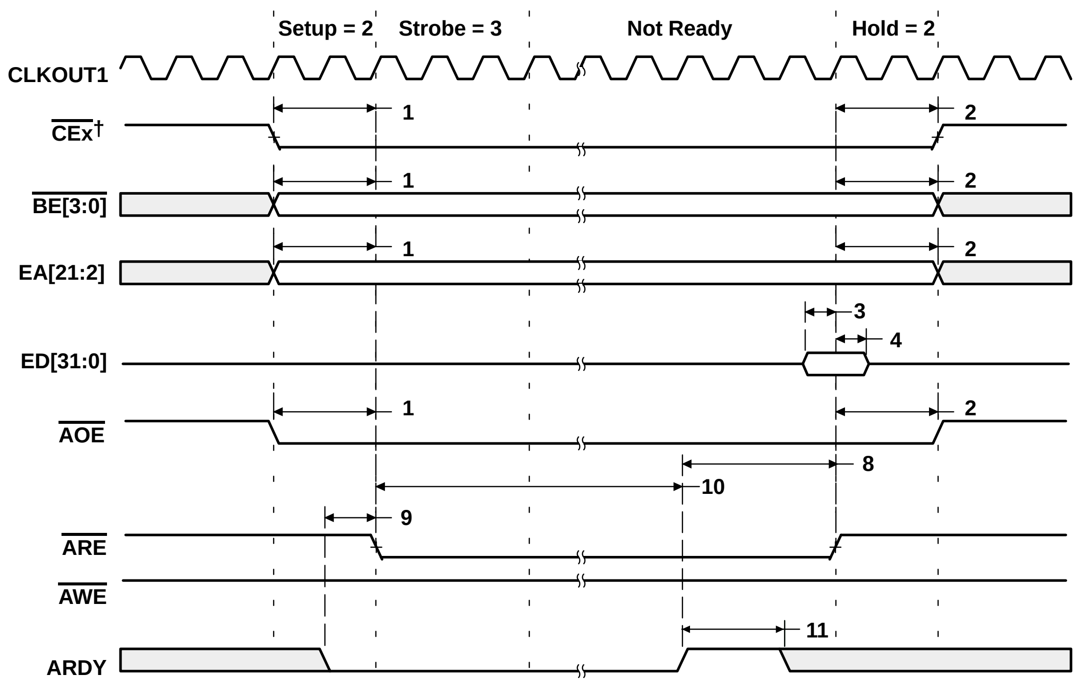
*This parameter is not production tested.
† RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the EMIF CE space control registers.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
§ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use ARDY input to extend strobe width.
¶ Select signals include: $\overline{CEx}$, $\overline{BE[3:0]}$, EA[21:2], $\overline{AOE}$; and for writes, include ED[31:0], with the exception that $\overline{CEx}$ can stay active for an additional 7P ns following the end of the cycle.

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



† CEx stays active for seven minus the value of Read Hold cycles after the last access (DMA transfer or CPU access). For example, if read HOLD = 1, then CEx stays active for six more cycles. This does not affect performance, it merely reflects the EMIF's overhead.

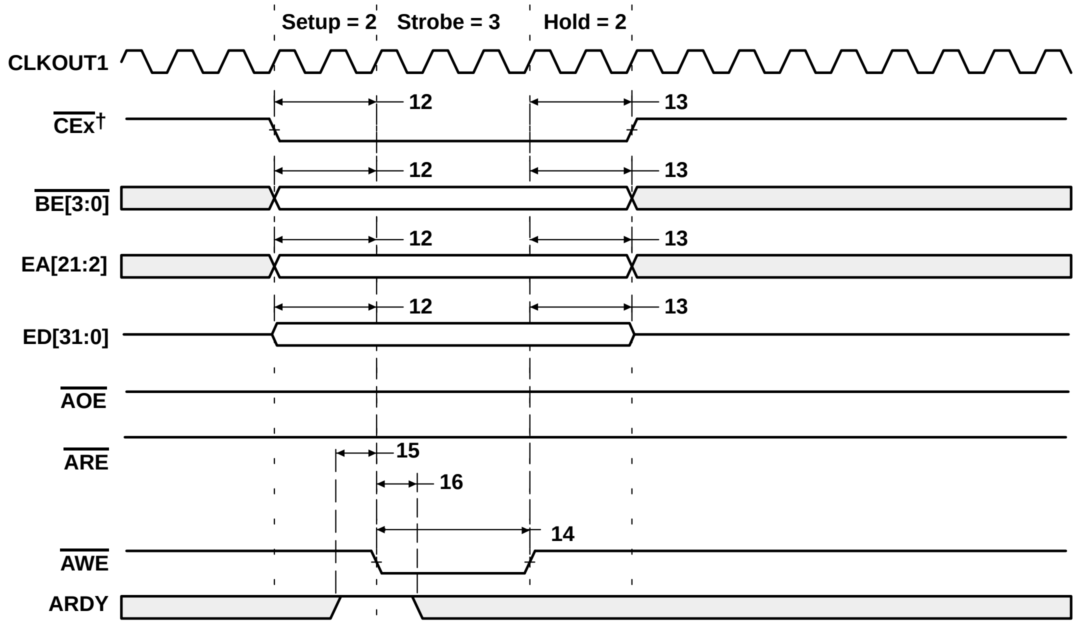
Figure 15. Asynchronous Memory Read Timing (ARDY Not Used)



† CEx stays active for seven minus the value of Read Hold cycles after the last access (DMA transfer or CPU access). For example, if read HOLD = 1, then CEx stays active for six more cycles. This does not affect performance, it merely reflects the EMIF's overhead.

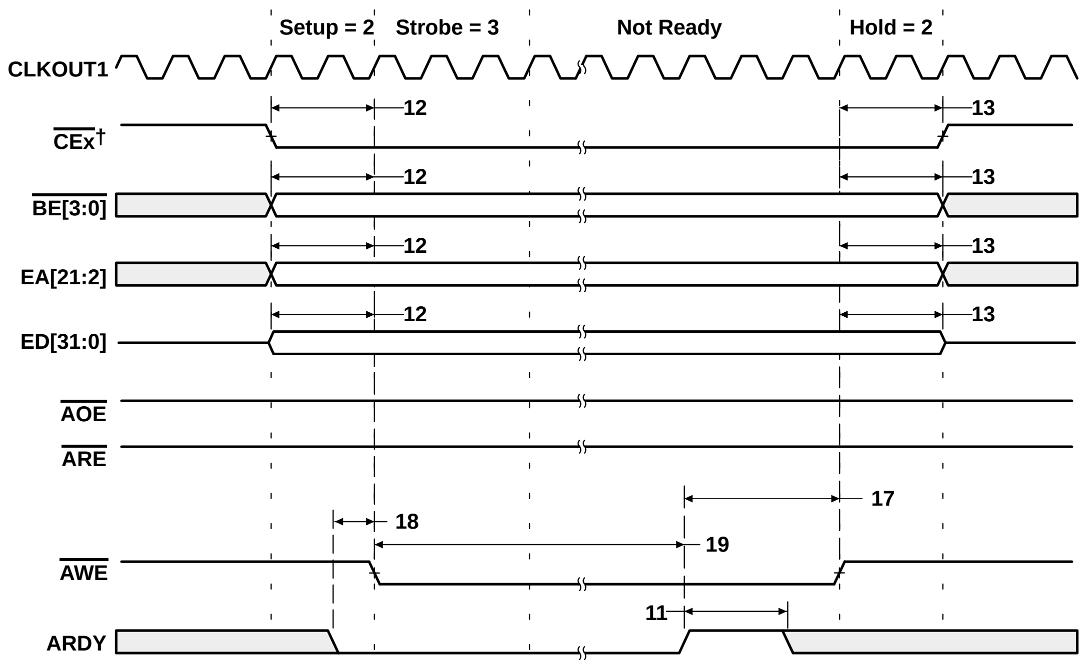
Figure 16. Asynchronous Memory Read Timing (ARDY Used)

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



† If no write accesses are scheduled for the next cycle and write hold is set to 1 or greater, then CEx stays active for three cycles after the value of the programmed hold period. If write hold is set to 0, then CEx stays active for four more cycles. This does not affect performance, it merely reflects the EMIF’s overhead.

Figure 17. Asynchronous Memory Write Timing (ARDY Not Used)



† If no write accesses are scheduled for the next cycle and write hold is set to 1 or greater, then CEx stays active for three cycles after the value of the programmed hold period. If write hold is set to 0, then CEx stays active for four more cycles. This does not affect performance, it merely reflects the EMIF’s overhead.

Figure 18. Asynchronous Memory Write Timing (ARDY Used)

SYNCHRONOUS-BURST MEMORY TIMING

timing requirements for synchronous-burst SRAM cycles (see Figure 19)

NO.			MIN	MAX	UNIT
7	$t_{su(EDV\text{-}CKO2H)}$	Setup time, read EDx valid before CLKOUT2 high	2.0		ns
8	$t_{h(CKO2H\text{-}EDV)}$	Hold time, read EDx valid after CLKOUT2 high	2.0		ns

switching characteristics over recommended operating conditions for synchronous-burst SRAM cycles†‡ (see Figure 19 and Figure 20)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{osu(CEV\text{-}CKO2H)}$	Output setup time, $\overline{\text{CEx}}$ valid before CLKOUT2 high	P – 0.8		ns
2	$t_{oh(CKO2H\text{-}CEV)}$	Output hold time, $\overline{\text{CEx}}$ valid after CLKOUT2 high	*P – 4		ns
3	$t_{osu(BEV\text{-}CKO2H)}$	Output setup time, $\overline{\text{BEx}}$ valid before CLKOUT2 high	P – 0.8		ns
4	$t_{oh(CKO2H\text{-}BEIV)}$	Output hold time, $\overline{\text{BEx}}$ invalid after CLKOUT2 high	*P – 4		ns
5	$t_{osu(EAV\text{-}CKO2H)}$	Output setup time, EAx valid before CLKOUT2 high	P – 0.8		ns
6	$t_{oh(CKO2H\text{-}EAIV)}$	Output hold time, EAx invalid after CLKOUT2 high	*P – 4		ns
9	$t_{osu(ADSV\text{-}CKO2H)}$	Output setup time, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid before CLKOUT2 high	P – 0.8		ns
10	$t_{oh(CKO2H\text{-}ADSV)}$	Output hold time, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid after CLKOUT2 high	*P – 4		ns
11	$t_{osu(OEV\text{-}CKO2H)}$	Output setup time, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid before CLKOUT2 high	P – 0.8		ns
12	$t_{oh(CKO2H\text{-}OEV)}$	Output hold time, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid after CLKOUT2 high	*P – 4		ns
13	$t_{osu(EDV\text{-}CKO2H)}$	Output setup time, EDx valid before CLKOUT2 high§	P – 1.2		ns
14	$t_{oh(CKO2H\text{-}EDIV)}$	Output hold time, EDx invalid after CLKOUT2 high	*P – 4		ns
15	$t_{osu(WEV\text{-}CKO2H)}$	Output setup time, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid before CLKOUT2 high	P – 0.8		ns
16	$t_{oh(CKO2H\text{-}WEV)}$	Output hold time, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid after CLKOUT2 high	*P – 4		ns

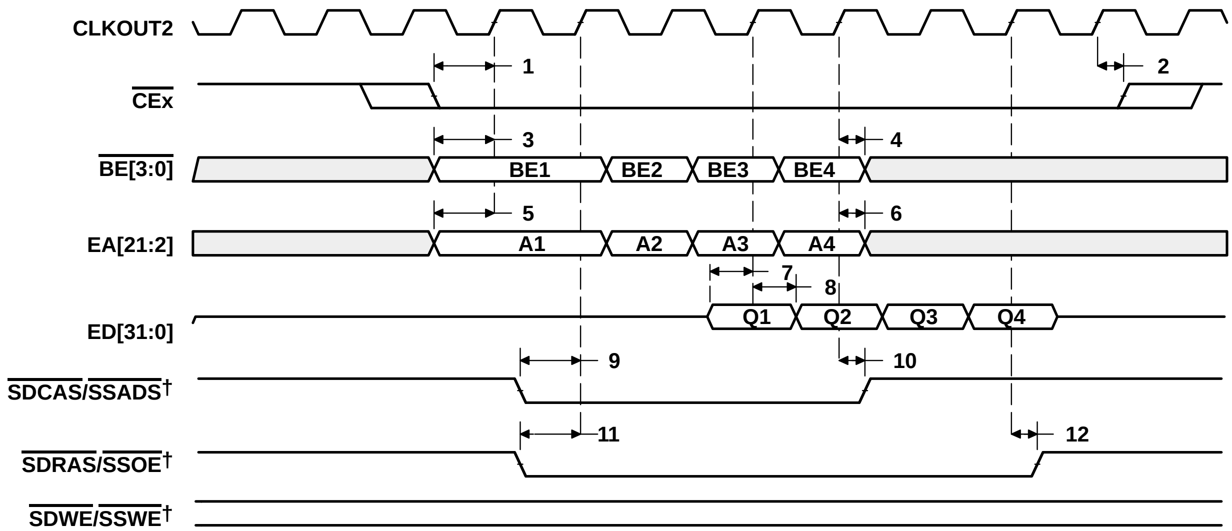
*This parameter is not production tested.

† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

‡ $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

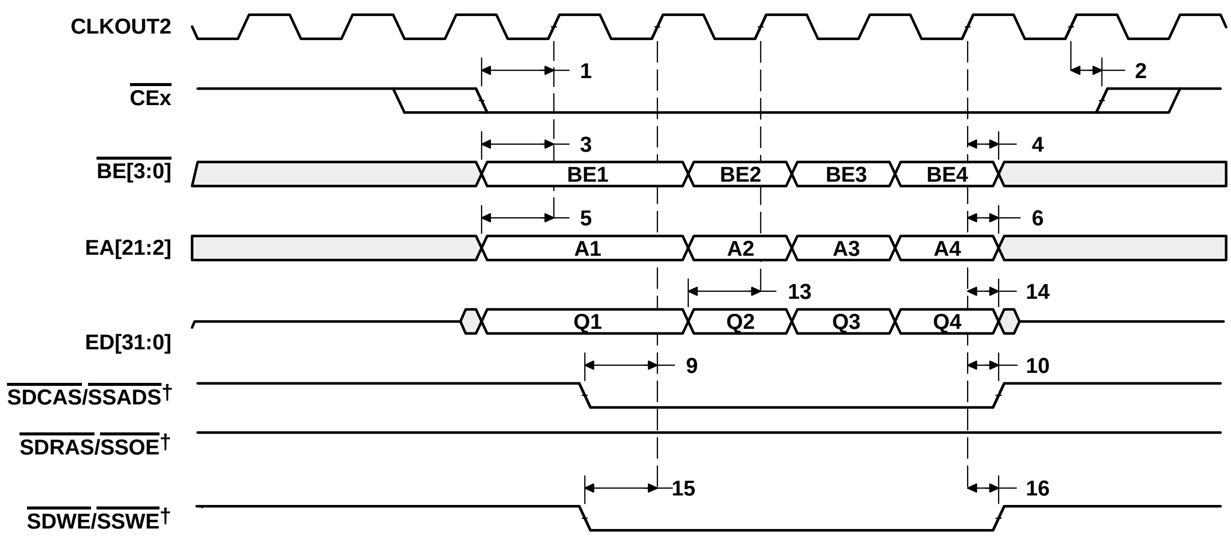
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)



† $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

Figure 19. SBSRAM Read Timing



† $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

Figure 20. SBSRAM Write Timing

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles (see Figure 21)

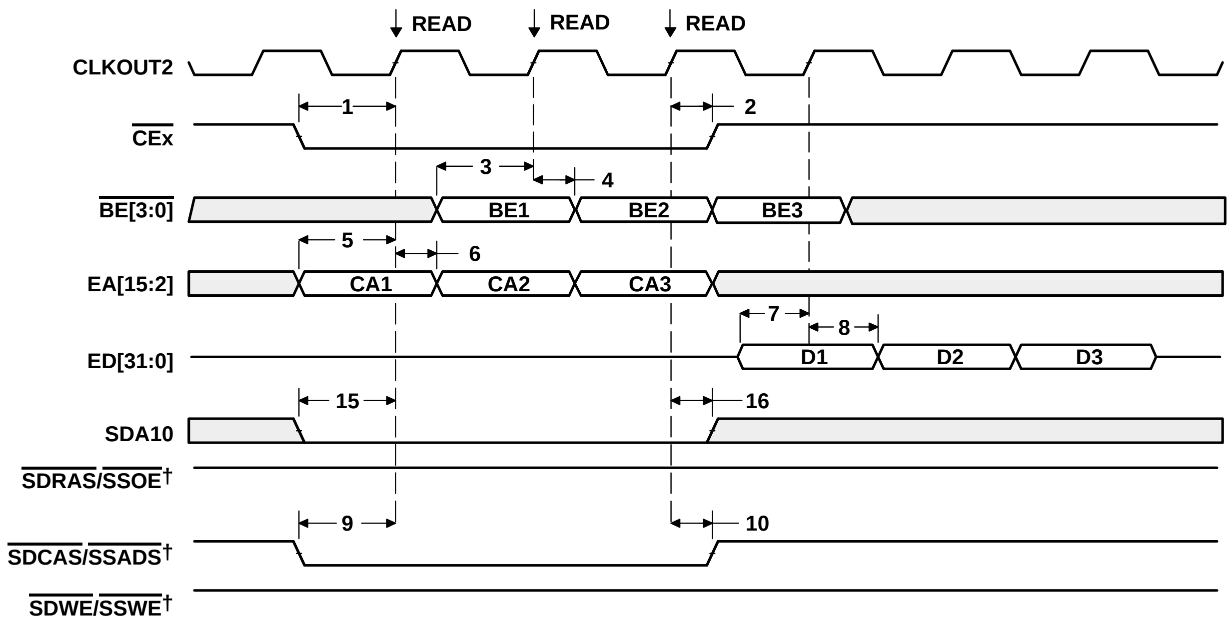
NO.			MIN	MAX	UNIT
7	$t_{su(EDV-CKO2H)}$	Setup time, read EDx valid before CLKOUT2 high	1.2		ns
8	$t_{h(CKO2H-EDV)}$	Hold time, read EDx valid after CLKOUT2 high	2.9		ns

switching characteristics over recommended operating conditions for synchronous DRAM cycles for C6203B Rev. 2†‡ (see Figure 21–Figure 26)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{osu(CEV-CKO2H)}$	Output setup time, $\overline{\text{CEx}}$ valid before CLKOUT2 high	P – 0.9		ns
2	$t_{oh(CKO2H-CEV)}$	Output hold time, $\overline{\text{CEx}}$ valid after CLKOUT2 high	*P – 4		ns
3	$t_{osu(BEV-CKO2H)}$	Output setup time, $\overline{\text{BEx}}$ valid before CLKOUT2 high	P – 0.9		ns
4	$t_{oh(CKO2H-BEIV)}$	Output hold time, $\overline{\text{BEx}}$ invalid after CLKOUT2 high	*P – 4		ns
5	$t_{osu(EAV-CKO2H)}$	Output setup time, EAx valid before CLKOUT2 high	P – 0.9		ns
6	$t_{oh(CKO2H-EAIV)}$	Output hold time, EAx invalid after CLKOUT2 high	*P – 4		ns
9	$t_{osu(CASV-CKO2H)}$	Output setup time, $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$ valid before CLKOUT2 high	P – 0.9		ns
10	$t_{oh(CKO2H-CASV)}$	Output hold time, $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$ valid after CLKOUT2 high	*P – 4		ns
11	$t_{osu(EDV-CKO2H)}$	Output setup time, EDx valid before CLKOUT2 high§	P – 1.5		ns
12	$t_{oh(CKO2H-EDIV)}$	Output hold time, EDx invalid after CLKOUT2 high	*P – 4		ns
13	$t_{osu(WEV-CKO2H)}$	Output setup time, $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ valid before CLKOUT2 high	P – 0.9		ns
14	$t_{oh(CKO2H-WEV)}$	Output hold time, $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ valid after CLKOUT2 high	*P – 4		ns
15	$t_{osu(SDA10V-CKO2H)}$	Output setup time, SDA10 valid before CLKOUT2 high	P – 0.9		ns
16	$t_{oh(CKO2H-SDA10IV)}$	Output hold time, SDA10 invalid after CLKOUT2 high	*P – 4		ns
17	$t_{osu(RASV-CKO2H)}$	Output setup time, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$ valid before CLKOUT2 high	P – 0.9		ns
18	$t_{oh(CKO2H-RASV)}$	Output hold time, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$ valid after CLKOUT2 high	*P – 4		ns

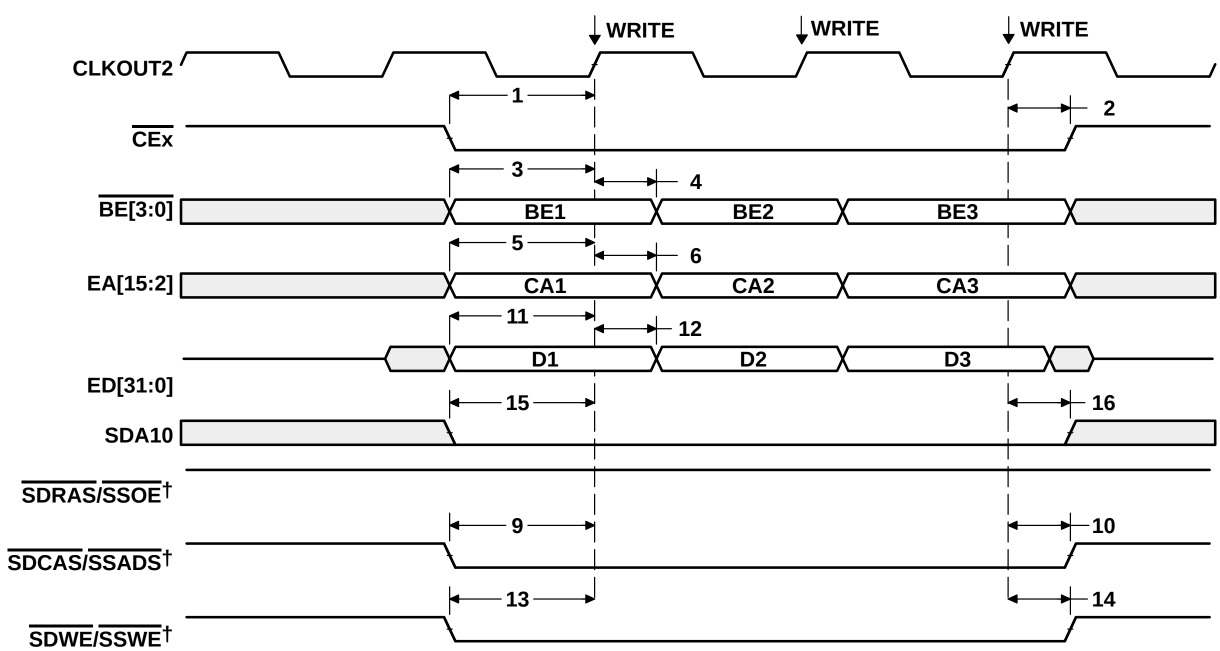
*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

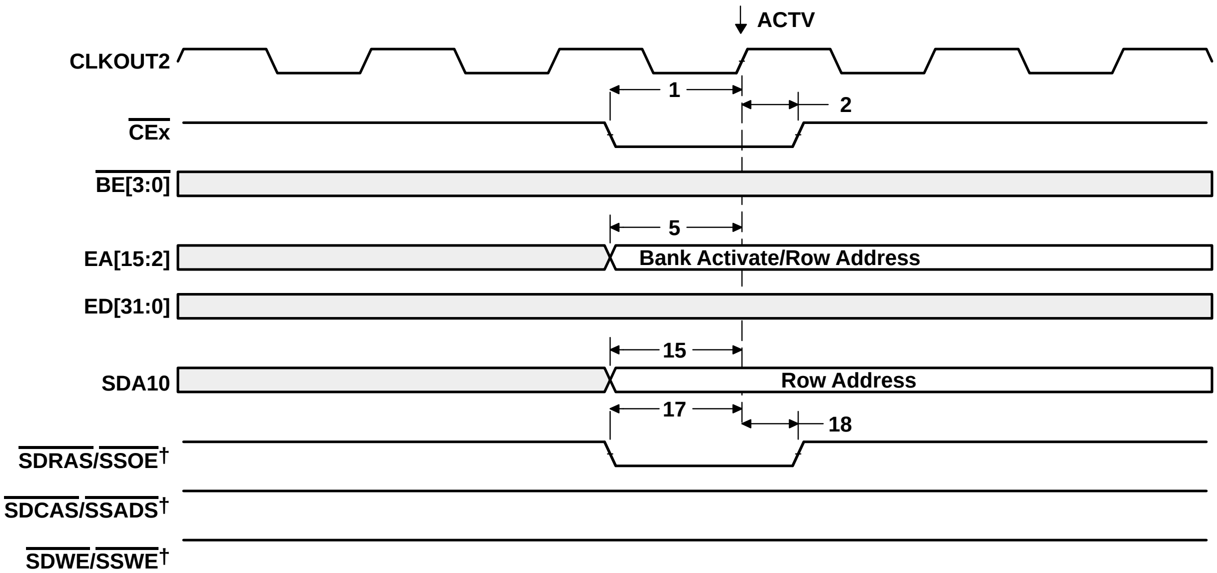
Figure 21. Three SDRAM READ Commands



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

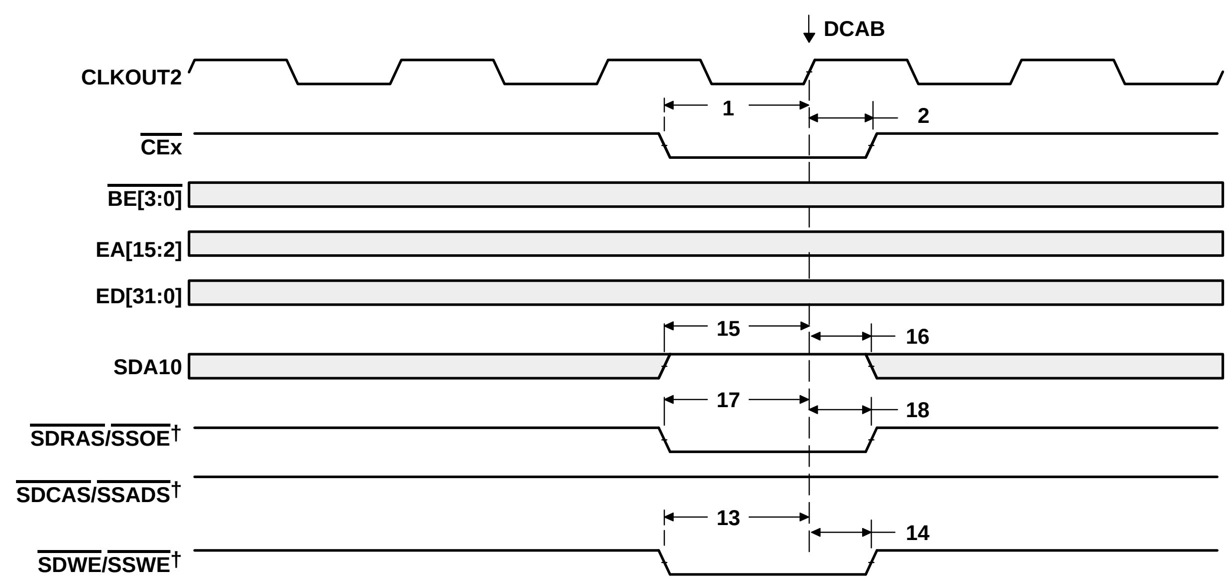
Figure 22. Three SDRAM WRT Commands

SYNCHRONOUS DRAM TIMING (CONTINUED)



† $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

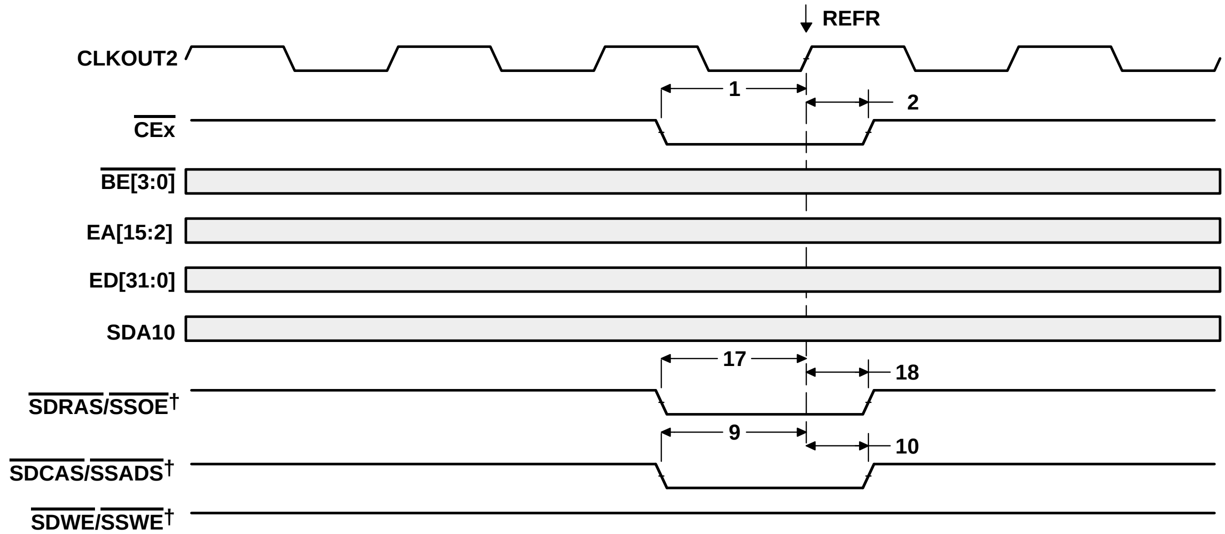
Figure 23. SDRAM ACTV Command



† $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

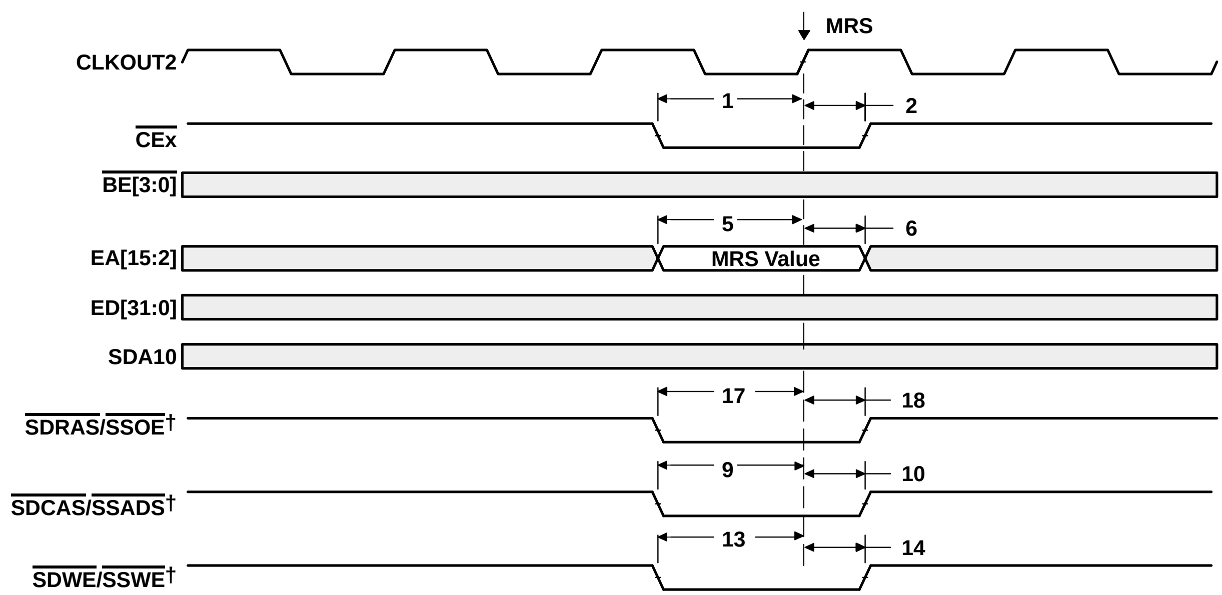
Figure 24. SDRAM DCAB Command

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

Figure 25. SDRAM REFR Command



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

Figure 26. SDRAM MRS Command

HOLD/HOLDA TIMING

timing requirements for the HOLD/HOLDA cycles† (see Figure 27)

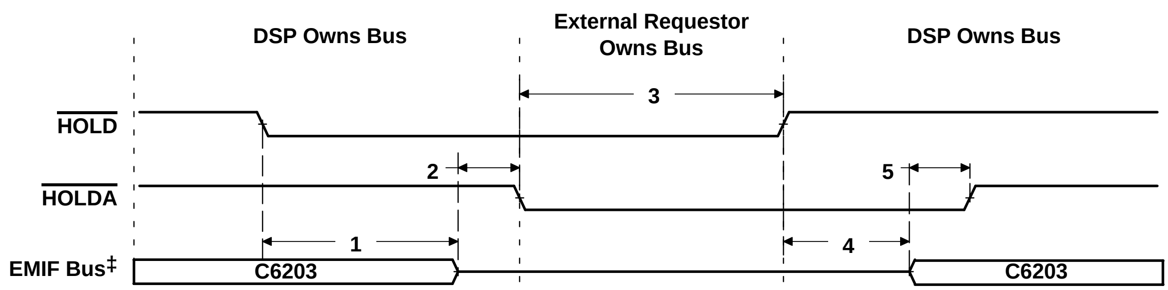
NO.			MIN	MAX	UNIT
3	$t_{oh(HOLDAL-HOLDL)}$	Output hold time, HOLD low after HOLDA low	*P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

switching characteristics over recommended operating conditions for the HOLD/HOLDA cycles†‡ (see Figure 27)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(HOLDL-EMHZ)}$	Delay time, HOLD low to EMIF Bus high impedance	*3P	§	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	*0	*2P	ns
4	$t_{d(HOLDH-EMLZ)}$	Delay time, HOLD high to EMIF Bus low impedance	*3P	*7P	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	*0	*2P	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ EMIF Bus consists of CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE, AOE, AWE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, and SDA10.
§ All pending EMIF transactions are allowed to complete before HOLDA is asserted. The worst case for this is an asynchronous read or write with external ARDY used or a minimum of eight consecutive SDRAM reads or writes when RBTR8 = 1. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.



‡ EMIF Bus consists of CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE, AOE, AWE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, and SDA10.

Figure 27. HOLD/HOLDA Timing

RESET TIMING

timing requirements for reset† (see Figure 28)

NO.			MIN	MAX	UNIT
1	$t_{w(RST)}$	Width of the RESET pulse (PLL stable)¶	*10P		ns
		Width of the RESET pulse (PLL needs to sync up)#	*250		µs
10	$t_{su(XD)}$	Setup time, XD configuration bits valid before RESET high||	*5P		ns
11	$t_{h(XD)}$	Hold time, XD configuration bits valid after RESET high||	*5P		ns

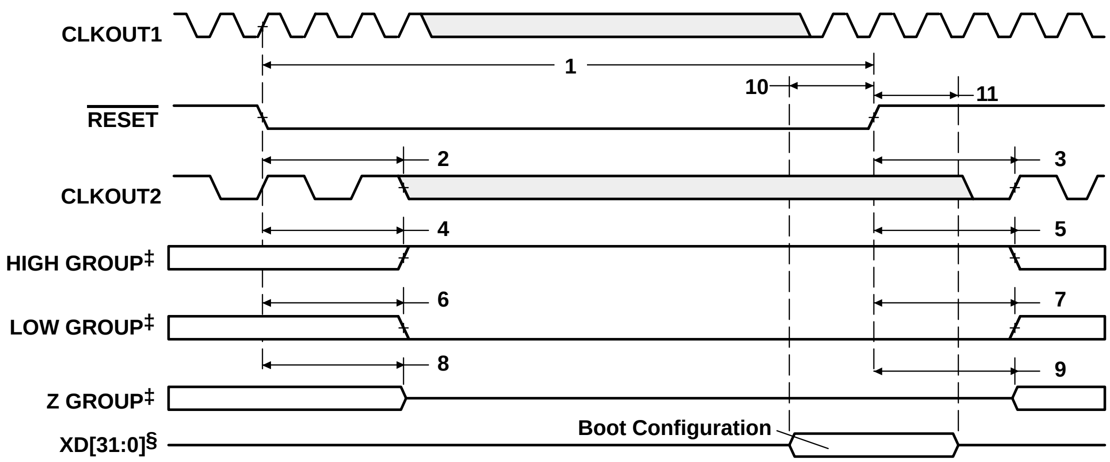
*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
¶ This parameter applies to CLKMODE x1 when CLKIN is stable, and applies to CLKMODE x4, x6, x7, x8, x9, x10, and x11 when CLKIN and PLL are stable.
# This parameter applies to CLKMODE x4, x6, x7, x8, x9, x10, and x11 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.
|| XD[31:0] are the boot configuration pins during device reset.

RESET TIMING (CONTINUED)

switching characteristics over recommended operating conditions during reset†‡ (see Figure 28)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(RSTL-CKO2IV)}$	Delay time, RESET low to CLKOUT2 invalid	*P		ns
3	$t_{d(RSTH-CKO2V)}$	Delay time, RESET high to CLKOUT2 valid		*4P	ns
4	$t_{d(RSTL-HIGHIV)}$	Delay time, RESET low to high group invalid	*P		ns
5	$t_{d(RSTH-HIGHV)}$	Delay time, RESET high to high group valid		*4P	ns
6	$t_{d(RSTL-LOWIV)}$	Delay time, RESET low to low group invalid	*P		ns
7	$t_{d(RSTH-LOWV)}$	Delay time, RESET high to low group valid		*4P	ns
8	$t_{d(RSTL-ZHZ)}$	Delay time, RESET low to Z group high impedance	*P		ns
9	$t_{d(RSTH-ZV)}$	Delay time, RESET high to Z group valid		*4P	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ High group consists of: XFCLK, HOLDA
Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE, AWE, AOE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, SDA10, CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, XCE[3:0], XBE[3:0]/XA[5:2], XOE, XRE, XWE/XWAIT, XAS, XW/R, XRDY, XBLAST, XHOLD, and XHOLDA



‡ High group consists of: XFCLK, HOLDA
Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE, AWE, AOE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, SDA10, CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, XCE[3:0], XBE[3:0]/XA[5:2], XOE, XRE, XWE/XWAIT, XAS, XW/R, XRDY, XBLAST, XHOLD, and XHOLDA
§ XD[31:0] are the boot configuration pins during device reset.

Figure 28. Reset Timing

EXTERNAL INTERRUPT TIMING

timing requirements for interrupt response cycles† (see Figure 29)

NO.			MIN	MAX	UNIT
2	$t_{w(ILOW)}$	Width of the interrupt pulse low	*2P		ns
3	$t_{w(IHIGH)}$	Width of the interrupt pulse high	*2P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

switching characteristics over recommended operating conditions during interrupt response cycles†‡ (see Figure 29)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{R(EINTH – IACKH)}$	Response time, EXT_INTx high to IACK high	*9P		ns
4	$t_{d(CKO2L-IACKV)}$	Delay time, CLKOUT2 low to IACK valid	*–1.5	*10	ns
5	$t_{d(CKO2L-INUMV)}$	Delay time, CLKOUT2 low to INUMx valid	*–2.0	*10	ns
6	$t_{d(CKO2L-INUMIV)}$	Delay time, CLKOUT2 low to INUMx invalid	*–2.0	*10	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ When CLKOUT2 is in half (1/2) mode (see CLKOUT2 in *Signal Descriptions* table), timings are based on falling edges

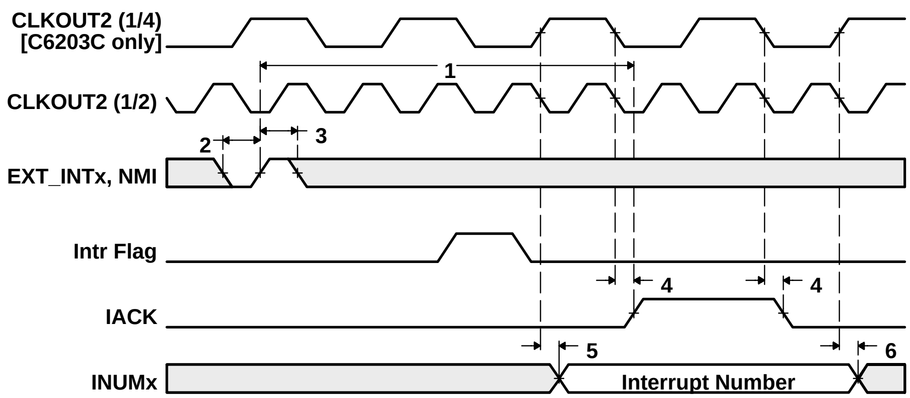


Figure 29. Interrupt Timing

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 ● HOUSTON, TEXAS 77251–1443

EXPANSION BUS SYNCHRONOUS FIFO TIMING

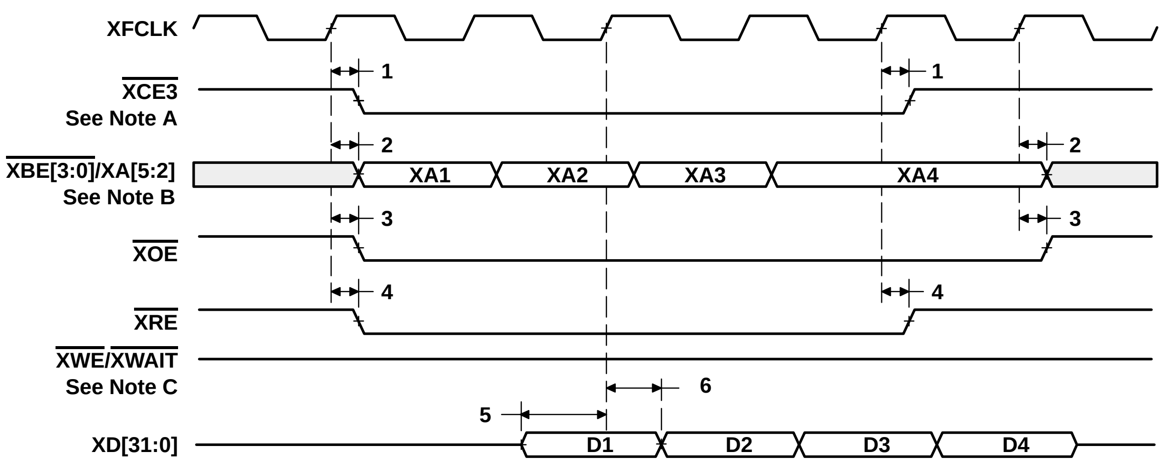
timing requirements for synchronous FIFO interface (see Figure 30, Figure 31, and Figure 32)

NO.			MIN	MAX	UNIT
5	$t_{su(XDV\text{-}XFCKH)}$	Setup time, read XDx valid before XFCLK high	3		ns
6	$t_{h(XFCKH\text{-}XDV)}$	Hold time, read XDx valid after XFCLK high	2.5		ns

switching characteristics over recommended operating conditions for synchronous FIFO interface (see Figure 30, Figure 31, and Figure 32)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(XFCKH\text{-}XCEV)}$	Delay time, XFCLK high to $\overline{XCE}$x valid	*–1.5	4.5	ns
2	$t_{d(XFCKH\text{-}XAV)}$	Delay time, XFCLK high to $\overline{XBE[3:0]}$/XA[5:2] valid†	*–1.5	4.5	ns
3	$t_{d(XFCKH\text{-}XOEV)}$	Delay time, XFCLK high to $\overline{XOE}$ valid	*–1.5	4.5	ns
4	$t_{d(XFCKH\text{-}XREV)}$	Delay time, XFCLK high to $\overline{XRE}$ valid	*–1.5	4.5	ns
7	$t_{d(XFCKH\text{-}XWEV)}$	Delay time, XFCLK high to $\overline{XWE}/\overline{XWAIT}$‡ valid	*–1.5	4.5	ns
8	$t_{d(XFCKH\text{-}XDV)}$	Delay time, XFCLK high to XDx valid		4.5	ns
9	$t_{d(XFCKH\text{-}XDIV)}$	Delay time, XFCLK high to XDx invalid	*–1.5		ns

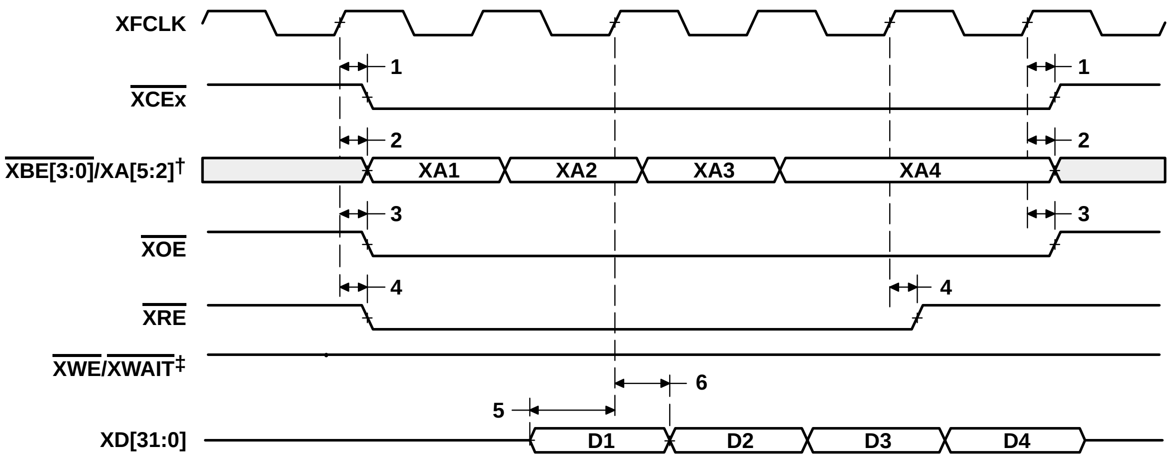
*This parameter is not production tested.
† $\overline{XBE[3:0]}$/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
‡ $\overline{XWE}/\overline{XWAIT}$ operates as the write-enable signal $\overline{XWE}$ during synchronous FIFO accesses.



NOTES: A. FIFO read (glueless) mode only available in $\overline{XCE3}$.
B. $\overline{XBE[3:0]}$/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
C. $\overline{XWE}/\overline{XWAIT}$ operates as the write-enable signal $\overline{XWE}$ during synchronous FIFO accesses.

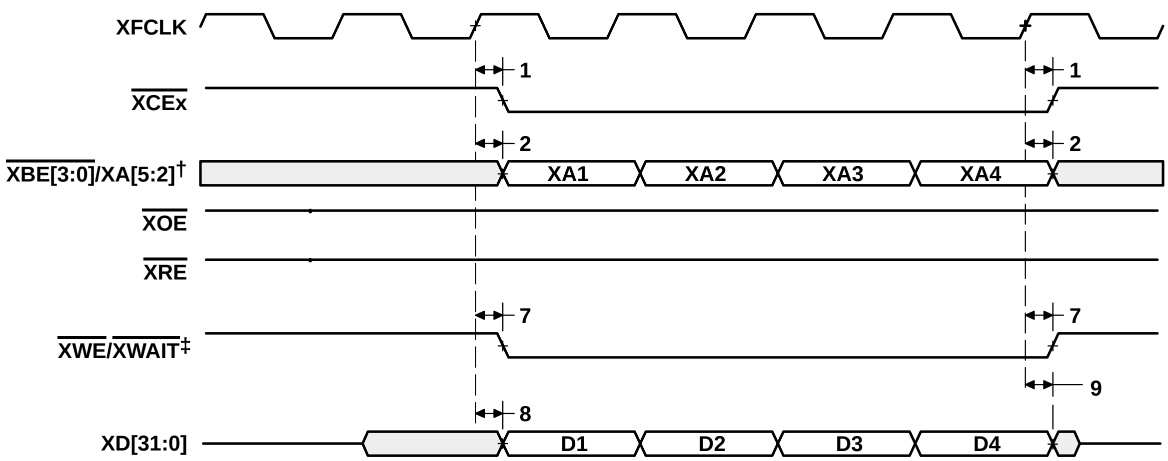
Figure 30. FIFO Read Timing (Glueless Read Mode)

EXPANSION BUS SYNCHRONOUS FIFO TIMING (CONTINUED)



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during synchronous FIFO accesses.

Figure 31. FIFO Read Timing



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during synchronous FIFO accesses.

Figure 32. FIFO Write Timing

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING

timing requirements for asynchronous peripheral cycles†‡§¶ (see Figure 33–Figure 36)

NO.			MIN	MAX	UNIT
3	$t_{su(XDV-XREH)}$	Setup time, XDx valid before $\overline{XRE}$ high	4.5		ns
4	$t_{h(XREH-XDV)}$	Hold time, XDx valid after $\overline{XRE}$ high	2.5		ns
6	$t_{su(XRDYH-XREL)}$	Setup time, XRDY high before $\overline{XRE}$ low	–[(RST – 3) x P – 6]		ns
7	$t_{h(XREL-XRDYH)}$	Hold time, XRDY high after $\overline{XRE}$ low	(RST – 3) x P + 2		ns
9	$t_{su(XRDYL-XREL)}$	Setup time, XRDY low before $\overline{XRE}$ low	–[(RST – 3) x P – 6]		ns
10	$t_{h(XREL-XRDYL)}$	Hold time, XRDY low after $\overline{XRE}$ low	(RST – 3) x P + 2		ns
11	$t_{w(XRDYH)}$	Pulse width, XRDY high	*2P		ns
15	$t_{su(XRDYH-XWEL)}$	Setup time, XRDY high before $\overline{XWE}$ low	–[(WST – 3) x P – 6]		ns
16	$t_{h(XWEL-XRDYH)}$	Hold time, XRDY high after $\overline{XWE}$ low	(WST – 3) x P + 2		ns
18	$t_{su(XRDYL-XWEL)}$	Setup time, XRDY low before $\overline{XWE}$ low	–[(WST – 3) x P – 6]		ns
19	$t_{h(XWEL-XRDYL)}$	Hold time, XRDY low after $\overline{XWE}$ low	(WST – 3) x P + 2		ns

*This parameter is not production tested.
† To ensure data setup time, simply program the strobe width wide enough. XRDY is internally synchronized. If XRDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, XRDY can be an asynchronous input.
‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the expansion bus XCE space control registers.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
¶ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use XRDY input to extend strobe width.

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)

switching characteristics over recommended operating conditions for asynchronous peripheral cycles†‡§¶ (see Figure 33–Figure 36)

NO.	PARAMETER		MIN	TYP	MAX	UNIT
1	$t_{osu(SELV-XREL)}$	Output setup time, select signals valid to $\overline{XRE}$ low	RS x P – 2			ns
2	$t_{oh(XREH-SELIV)}$	Output hold time, $\overline{XRE}$ low to select signals invalid	*RH x P – 2			ns
5	$t_{w(XREL)}$	Pulse width, $\overline{XRE}$ low		RST x P		ns
8	$t_{d(XRDYH-XREH)}$	Delay time, XRDY high to $\overline{XRE}$ high	*3P		*4P + 5	ns
12	$t_{osu(SELV-XWEL)}$	Output setup time, select signals valid to $\overline{XWE}$ low	WS x P – 3			ns
13	$t_{oh(XWEH-SELIV)}$	Output hold time, $\overline{XWE}$ low to select signals invalid	*WH x P – 2			ns
14	$t_{w(XWEL)}$	Pulse width, $\overline{XWE}$ low		WST x P		ns
17	$t_{d(XRDYH-XWEH)}$	Delay time, XRDY high to $\overline{XWE}$ high	*3P		*4P + 5	ns

*This parameter is not production tested.

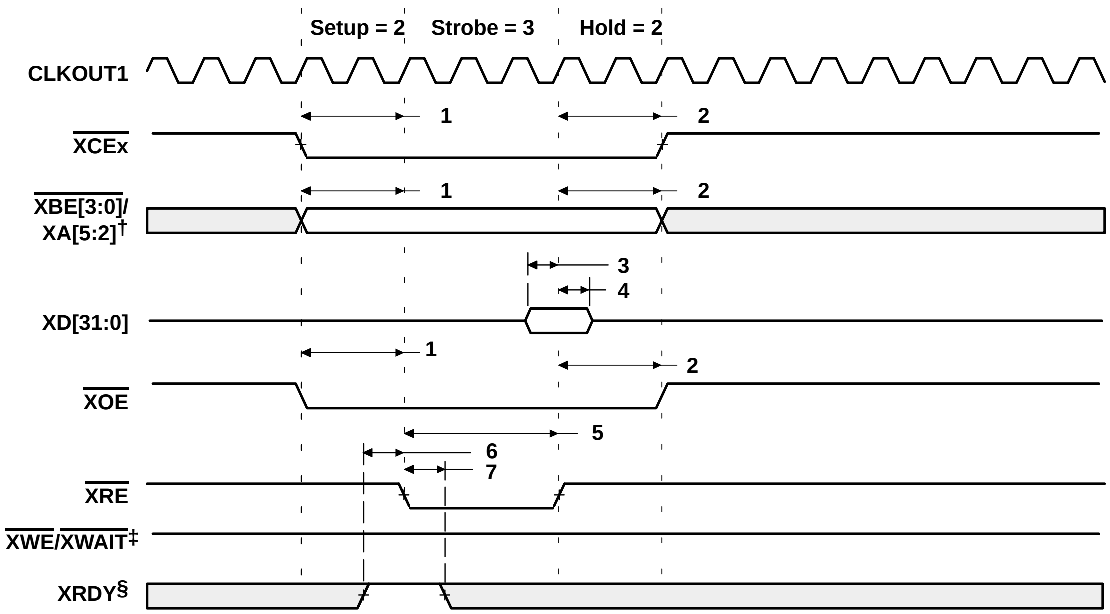
† RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the expansion bus XCE space control registers.

‡ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

§ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use XRDY input to extend strobe width.

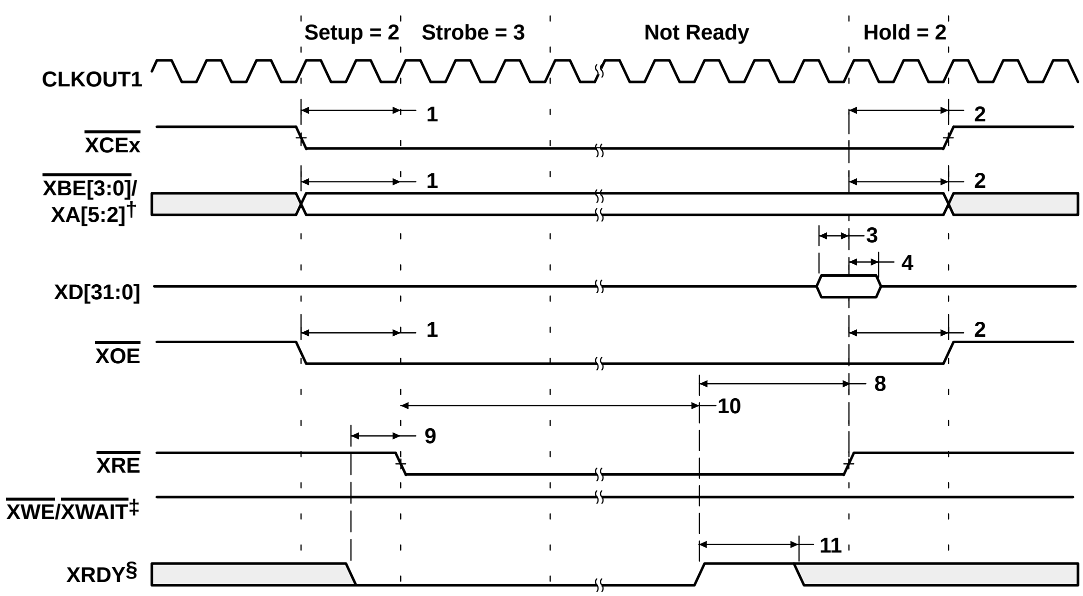
¶ Select signals include: $\overline{XCEx}$, $\overline{XBE[3:0]}$/XA[5:2], $\overline{XOE}$; and for writes, include XD[31:0], with the exception that $\overline{XCEx}$ can stay active for an additional 7P ns following the end of the cycle.

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

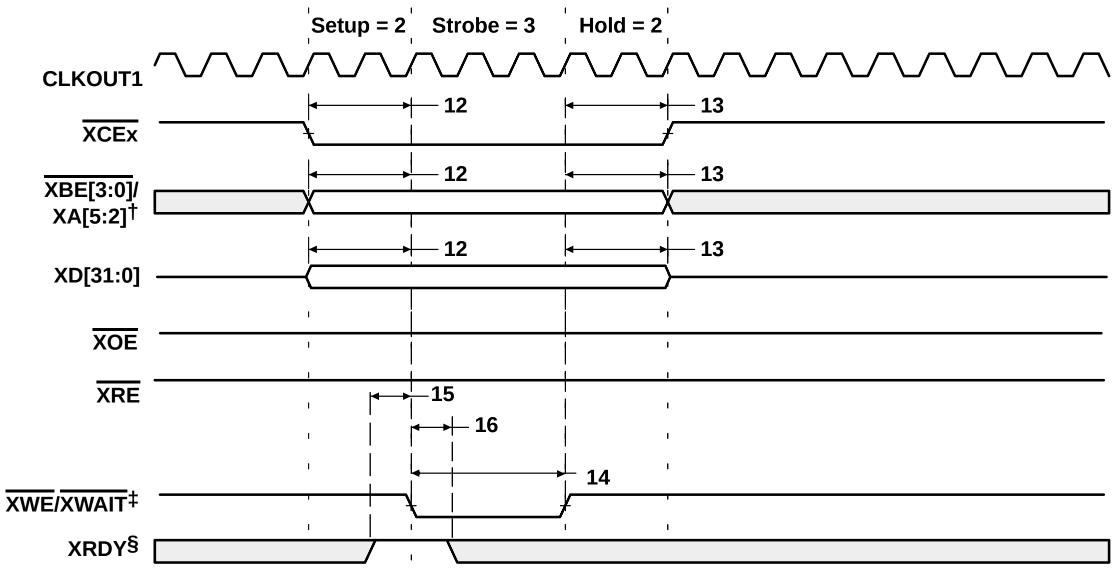
Figure 33. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Not Used)



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

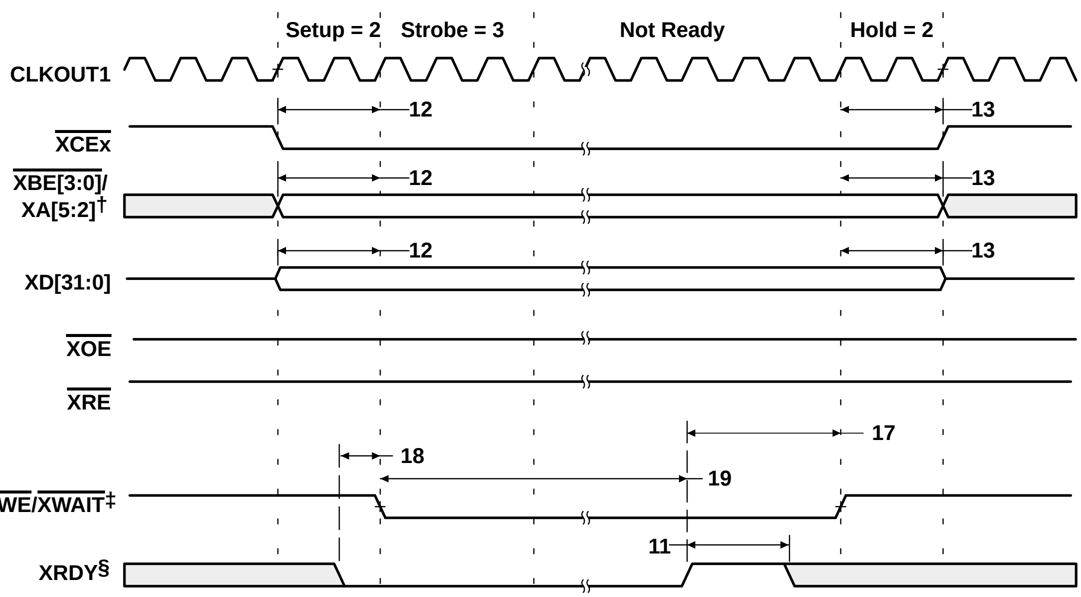
Figure 34. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Used)

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

Figure 35. Expansion Bus Asynchronous Peripheral Write Timing (XRDY Not Used)



† XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
‡ XWE/XWAIT operates as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
§ XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

Figure 36. Expansion Bus Asynchronous Peripheral Write Timing (XRDY Used)

EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING

timing requirements with external device as bus master (see Figure 37 and Figure 38)

NO.			MIN	MAX	UNIT
1	$t_{su(XCSV-XCKIH)}$	Setup time, $\overline{XCS}$ valid before XCLKIN high	3.5		ns
2	$t_{h(XCKIH-XCS)}$	Hold time, $\overline{XCS}$ valid after XCLKIN high	2.8		ns
3	$t_{su(XAS-XCKIH)}$	Setup time, $\overline{XAS}$ valid before XCLKIN high	3.5		ns
4	$t_{h(XCKIH-XAS)}$	Hold time, $\overline{XAS}$ valid after XCLKIN high	2.8		ns
5	$t_{su(XCTL-XCKIH)}$	Setup time, XCNTL valid before XCLKIN high	3.5		ns
6	$t_{h(XCKIH-XCTL)}$	Hold time, XCNTL valid after XCLKIN high	2.8		ns
7	$t_{su(XWR-XCKIH)}$	Setup time, XW/R valid before XCLKIN high†	3.5		ns
8	$t_{h(XCKIH-XWR)}$	Hold time, XW/R valid after XCLKIN high†	2.8		ns
9	$t_{su(XBLTV-XCKIH)}$	Setup time, XBLAST valid before XCLKIN high‡	3.5		ns
10	$t_{h(XCKIH-XBLTV)}$	Hold time, XBLAST valid after XCLKIN high‡	2.8		ns
16	$t_{su(XBEV-XCKIH)}$	Setup time, $\overline{XBE[3:0]}$/XA[5:2] valid before XCLKIN high§	3.5		ns
17	$t_{h(XCKIH-XBEV)}$	Hold time, $\overline{XBE[3:0]}$/XA[5:2] valid after XCLKIN high§	2.8		ns
18	$t_{su(XD-XCKIH)}$	Setup time, XDx valid before XCLKIN high	3.5		ns
19	$t_{h(XCKIH-XD)}$	Hold time, XDx valid after XCLKIN high	2.8		ns

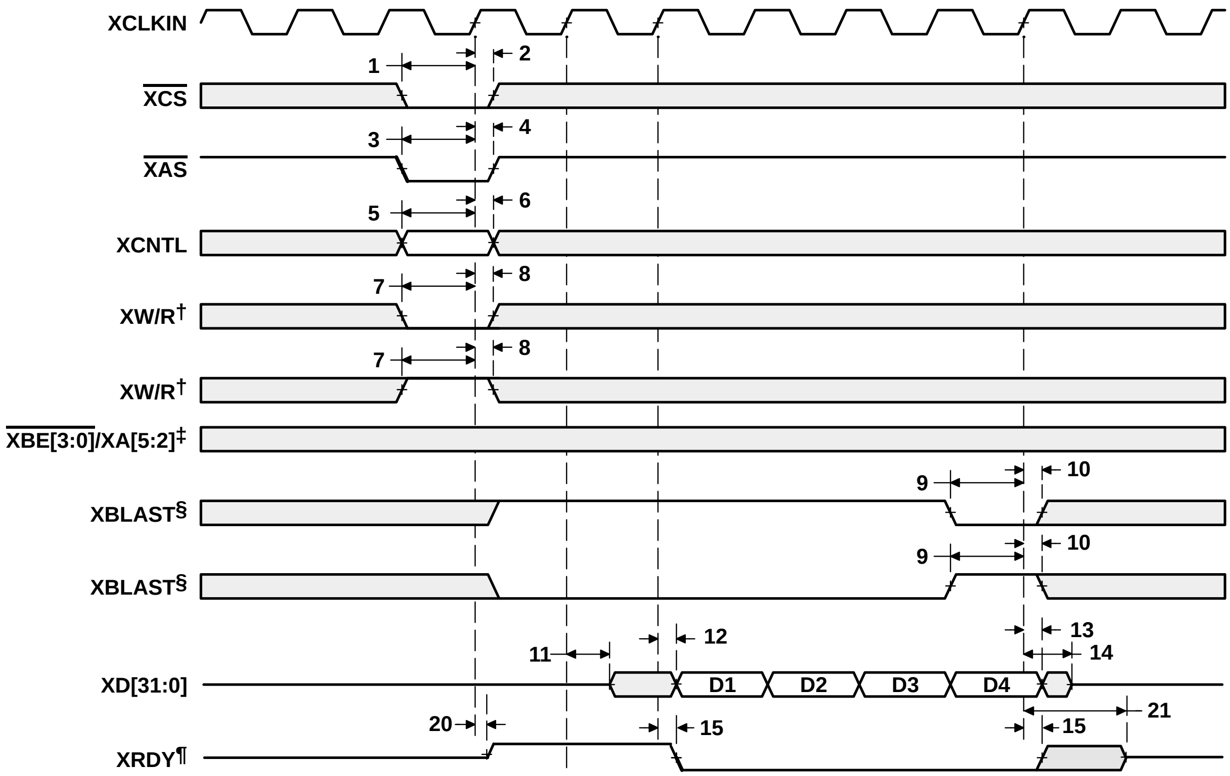
† XW/R input/output polarity selected at boot.
‡ XBLAST input polarity selected at boot
§ $\overline{XBE[3:0]}$/XA[5:2] operate as byte-enables $\overline{XBE[3:0]}$ during host-port accesses.

switching characteristics over recommended operating conditions with external device as bus master¶ (see Figure 37 and Figure 38)

NO.	PARAMETER		MIN	MAX	UNIT
11	$t_{d(XCKIH-XDLZ)}$	Delay time, XCLKIN high to XDx low impedance	*0		ns
12	$t_{d(XCKIH-XDV)}$	Delay time, XCLKIN high to XDx valid		17	ns
13	$t_{d(XCKIH-XDIV)}$	Delay time, XCLKIN high to XDx invalid	*5		ns
14	$t_{d(XCKIH-XDHZ)}$	Delay time, XCLKIN high to XDx high impedance		*4P	ns
15	$t_{d(XCKIH-XRY)}$	Delay time, XCLKIN high to XRDY invalid#	*5	*17	ns
20	$t_{d(XCKIH-XRYLZ)}$	Delay time, XCLKIN high to XRDY low impedance	*5	*17	ns
21	$t_{d(XCKIH-XRYHZ)}$	Delay time, XCLKIN high to XRDY high impedance#	*2P + 5	*3P + 17	ns

*This parameter is not production tested.
¶ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
# XRDY operates as active-low ready input/output during host-port accesses.

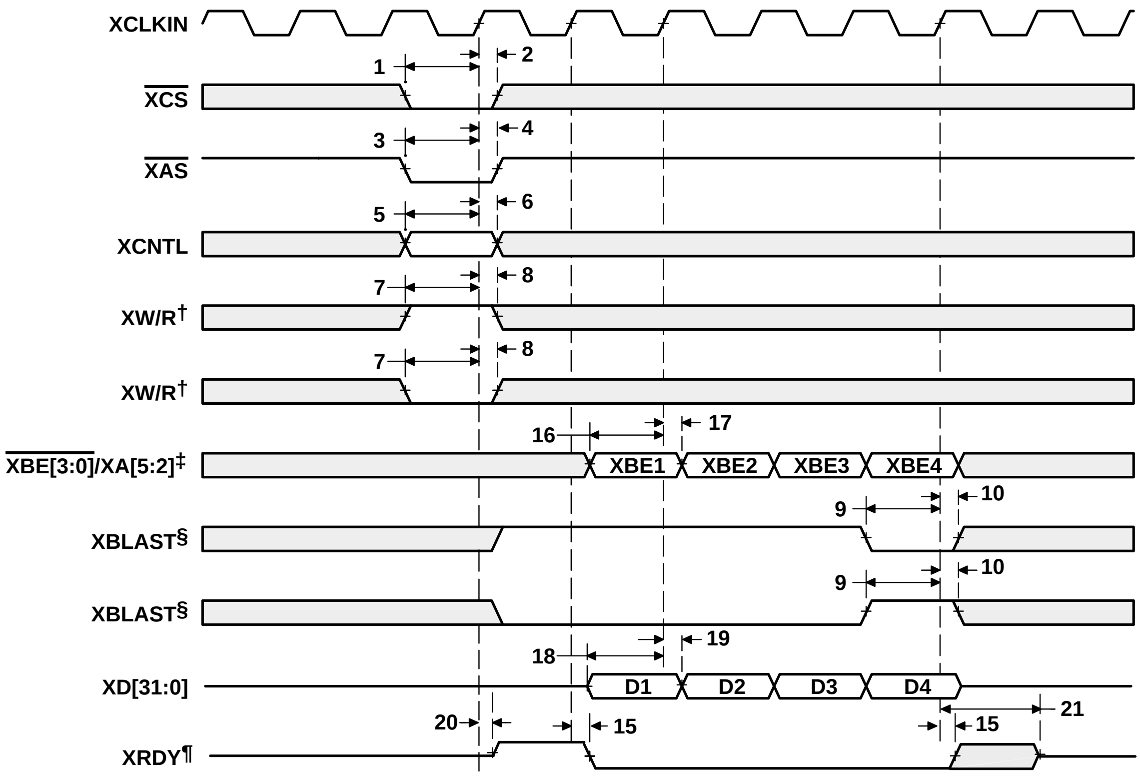
EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING (CONTINUED)



† XW/R input/output polarity selected at boot
‡ XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
§ XBLAST input polarity selected at boot
¶ XRDY operates as active-low ready input/output during host-port accesses.

Figure 37. External Host as Bus Master—Read

EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING (CONTINUED)



† XW/R input/output polarity selected at boot
‡ XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
§ XBLAST input polarity selected at boot
¶ XRDY operates as active-low ready input/output during host-port accesses.

Figure 38. External Host as Bus Master—Write

EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING (CONTINUED)

timing requirements with C62x™ as bus master (see Figure 39, Figure 40, and Figure 41)

NO.			MIN	MAX	UNIT
9	$t_{su(XDV-XCKIH)}$	Setup time, XDx valid before XCLKIN high	3.5		ns
10	$t_{h(XCKIH-XDV)}$	Hold time, XDx valid after XCLKIN high	2.8		ns
11	$t_{su(XRY-XCKIH)}$	Setup time, XRDY valid before XCLKIN high†	3.5		ns
12	$t_{h(XCKIH-XRY)}$	Hold time, XRDY valid after XCLKIN high†	2.8		ns
14	$t_{su(XBFF-XCKIH)}$	Setup time, XBOFF valid before XCLKIN high	3.5		ns
15	$t_{h(XCKIH-XBFF)}$	Hold time, XBOFF valid after XCLKIN high	2.8		ns

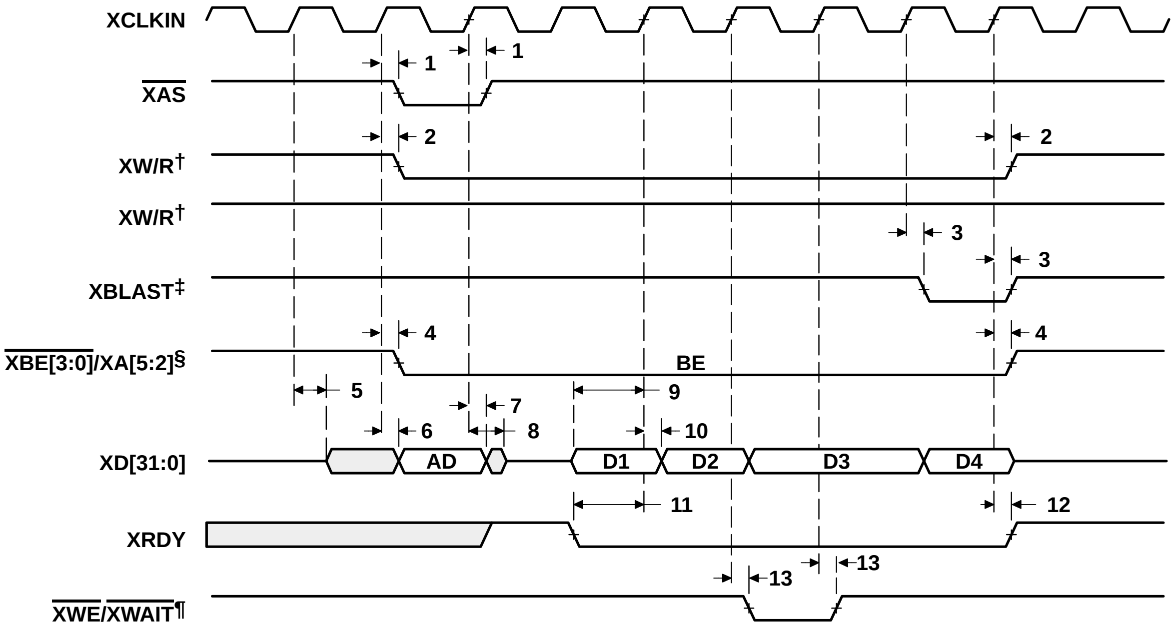
† XRDY operates as active-low ready input/output during host-port accesses.

switching characteristics over recommended operating conditions with C62x™ as bus master‡ (see Figure 39, Figure 40, and Figure 41)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(XCKIH-XASV)}$	Delay time, XCLKIN high to $\overline{XAS}$ valid	*5	17	ns
2	$t_{d(XCKIH-XWRV)}$	Delay time, XCLKIN high to XW/R valid§	*5	17	ns
3	$t_{d(XCKIH-XBLTV)}$	Delay time, XCLKIN high to XBLAST valid¶	*5	17	ns
4	$t_{d(XCKIH-XBEV)}$	Delay time, XCLKIN high to $\overline{XBE[3:0]}$/XA[5:2] valid#	*5	17	ns
5	$t_{d(XCKIH-XDLZ)}$	Delay time, XCLKIN high to XDx low impedance	*0		ns
6	$t_{d(XCKIH-XDV)}$	Delay time, XCLKIN high to XDx valid		17	ns
7	$t_{d(XCKIH-XDIV)}$	Delay time, XCLKIN high to XDx invalid	*5		ns
8	$t_{d(XCKIH-XDHZ)}$	Delay time, XCLKIN high to XDx high impedance		*4P	ns
13	$t_{d(XCKIH-XWTV)}$	Delay time, XCLKIN high to $\overline{XWE}$/$\overline{XWAIT}$ valid‖	*5	17	ns

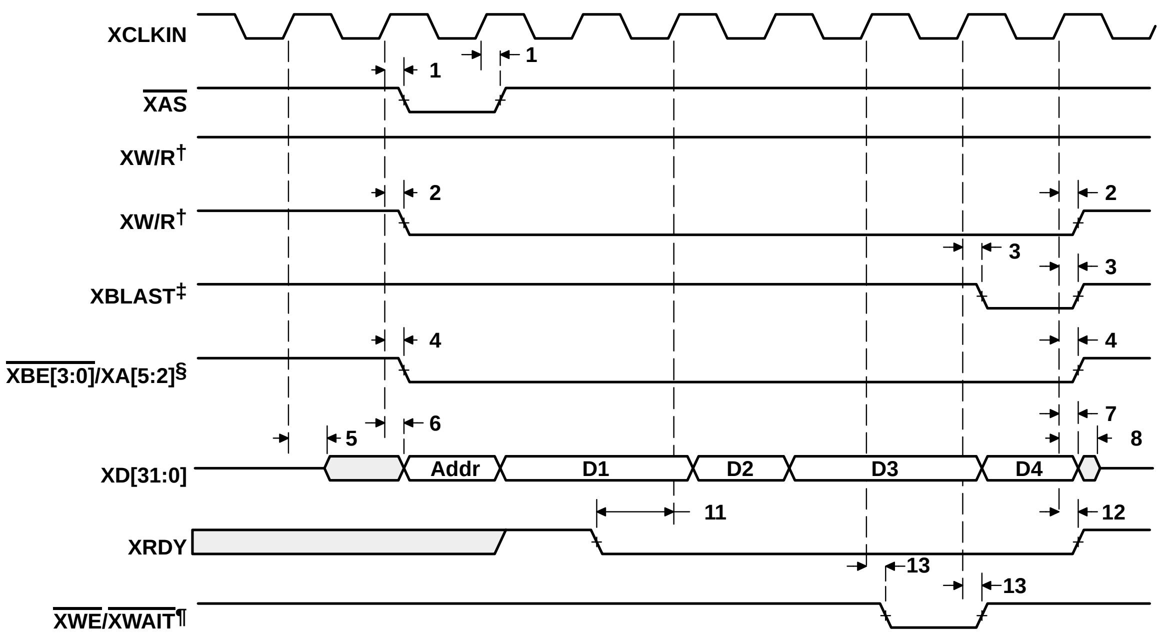
*This parameter is not production tested.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
§ XW/R input/output polarity selected at boot.
¶ $\overline{XBLAST}$ output polarity is always active low.
# $\overline{XBE[3:0]}$/$\overline{XA}$[5:2] operate as byte-enables $\overline{XBE[3:0]}$ during host-port accesses.
‖ $\overline{XWE}$/$\overline{XWAIT}$ operates as $\overline{XWAIT}$ output signal during host-port accesses.

EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING (CONTINUED)



† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
¶ XWE/XWAIT operates as XWAIT output signal during host-port accesses.

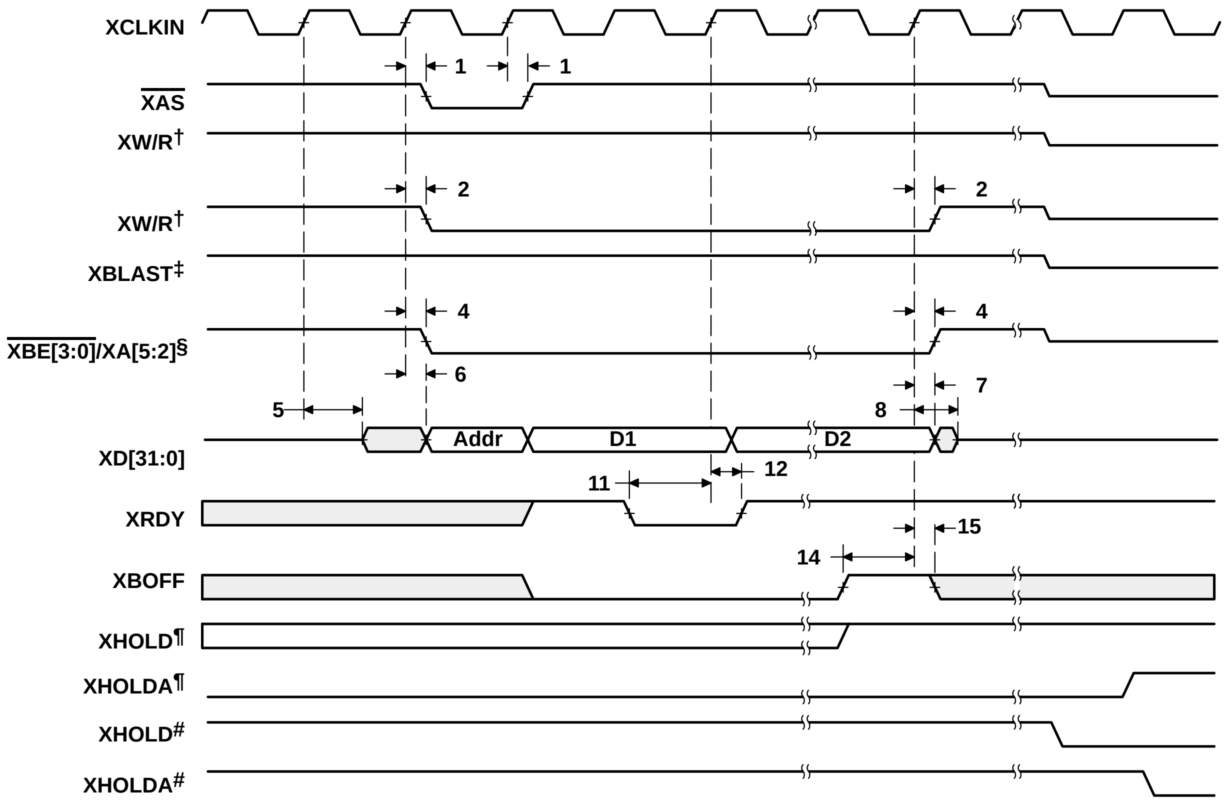
Figure 39. C62x™ as Bus Master—Read



† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
¶ XWE/XWAIT operates as XWAIT output signal during host-port accesses.

Figure 40. C62x™ as Bus Master—Write

EXPANSION BUS SYNCHRONOUS HOST-PORT TIMING (CONTINUED)



† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
¶ Internal arbiter enabled
# External arbiter enabled
|| This diagram illustrates XBOFF timing. Bus arbitration timing is shown in Figure 44 and Figure 45.

Figure 41. C62x™ as Bus Master—BOFF Operation||

EXPANSION BUS ASYNCHRONOUS HOST-PORT TIMING

timing requirements with external device as asynchronous bus master† (see Figure 42 and Figure 43)

NO.			MIN	MAX	UNIT
1	$t_{w(XCSL)}$	Pulse duration, $\overline{XCS}$ low	4P		ns
2	$t_{w(XCSH)}$	Pulse duration, $\overline{XCS}$ high	4P		ns
3	$t_{su(XSEL-XCSL)}$	Setup time, expansion bus select signals‡ valid before $\overline{XCS}$ low	1		ns
4	$t_{h(XCSL-XSEL)}$	Hold time, expansion bus select signals‡ valid after $\overline{XCS}$ low	3.4		ns
10	$t_{h(XRYL-XCSL)}$	Hold time, $\overline{XCS}$ low after XRDY low	*P + 1.5		ns
11	$t_{su(XBEV-XCSH)}$	Setup time, $\overline{XBE[3:0]}$/XA[5:2] valid before $\overline{XCS}$ high§	1		ns
12	$t_{h(XCSH-XBEV)}$	Hold time, $\overline{XBE[3:0]}$/XA[5:2] valid after $\overline{XCS}$ high§	3		ns
13	$t_{su(XDV-XCSH)}$	Setup time, XDx valid before $\overline{XCS}$ high	1		ns
14	$t_{h(XCSH-XDV)}$	Hold time, XDx valid after $\overline{XCS}$ high	3		ns

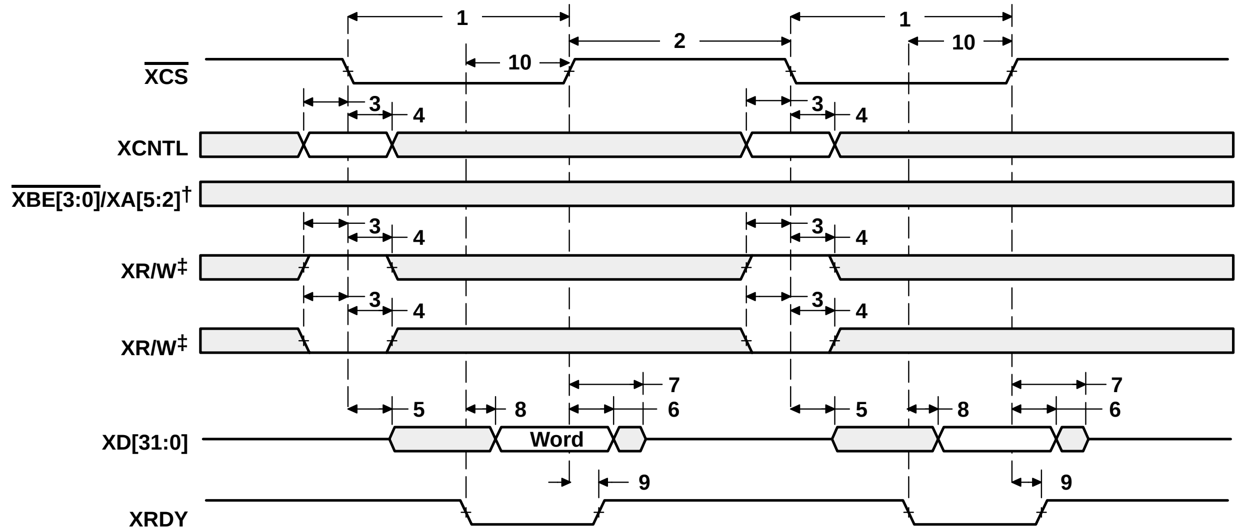
*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ Expansion bus select signals include XCNTL and XR/W.
§ $\overline{XBE[3:0]}$/XA[5:2] operate as byte-enables $\overline{XBE[3:0]}$ during host-port accesses.

switching characteristics over recommended operating conditions with external device as asynchronous bus master† (see Figure 42 and Figure 43)

NO.	PARAMETER		MIN	MAX	UNIT
5	$t_{d(XCSL-XDLZ)}$	Delay time, $\overline{XCS}$ low to XDx low impedance	*0		ns
6	$t_{d(XCSH-XDIV)}$	Delay time, $\overline{XCS}$ high to XDx invalid	*0	*12	ns
7	$t_{d(XCSH-XDHZ)}$	Delay time, $\overline{XCS}$ high to XDx high impedance		*4P	ns
8	$t_{d(XRYL-XDV)}$	Delay time, XRDY low to XDx valid	*–4	*1	ns
9	$t_{d(XCSH-XRYH)}$	Delay time, $\overline{XCS}$ high to XRDY high	*0	12	ns

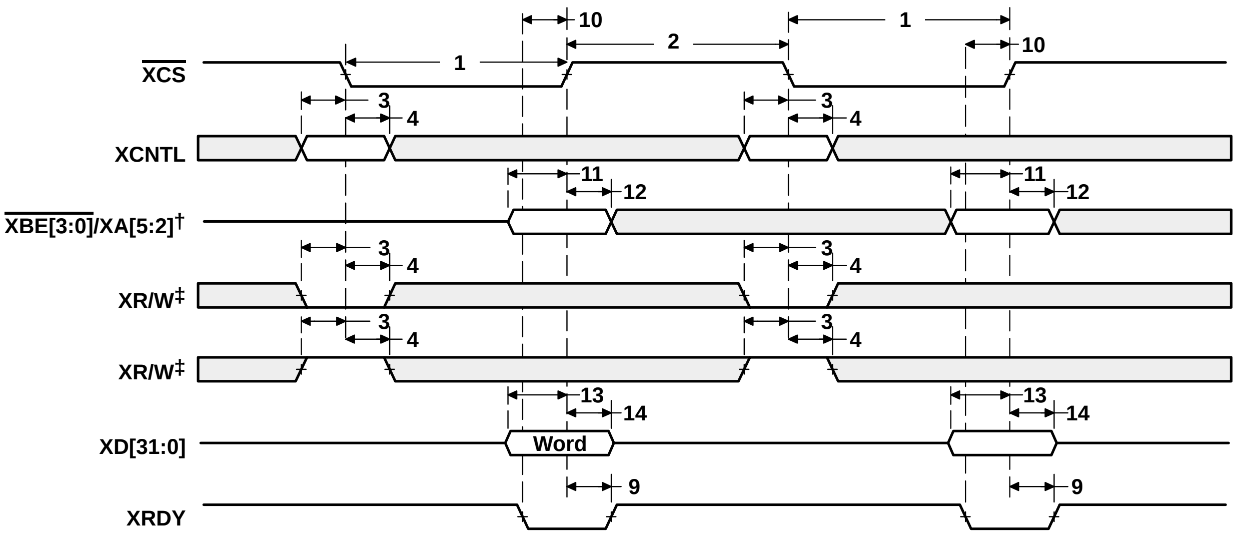
*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

EXPANSION BUS ASYNCHRONOUS HOST-PORT TIMING (CONTINUED)



† XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
‡ XW/R input/output polarity selected at boot

Figure 42. External Device as Asynchronous Master—Read



† XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
‡ XW/R input/output polarity selected at boot

Figure 43. External Device as Asynchronous Master—Write

XHOLD/XHOLDA TIMING

timing requirements for expansion bus arbitration (internal arbiter enabled)† (see Figure 44)

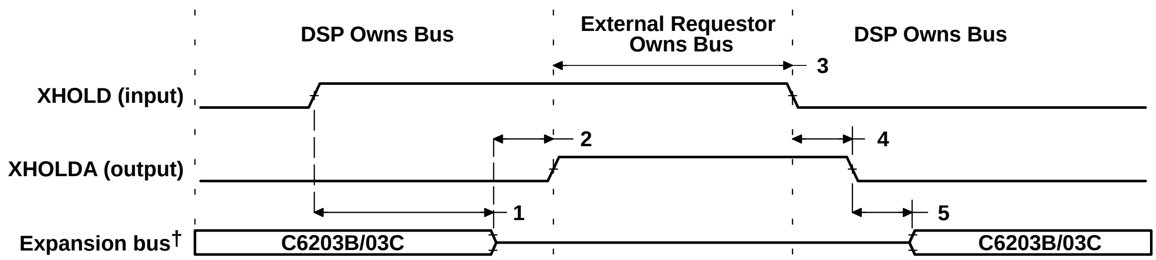
NO.			MIN	MAX	UNIT
3	$t_{oh(XHDAH-XHDH)}$	Output hold time, XHOLD high after XHOLDA high	*P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

switching characteristics over recommended operating conditions for expansion bus arbitration (internal arbiter enabled)†‡ (see Figure 44)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(XHDH-XBHZ)}$	Delay time, XHOLD high to expansion bus high impedance	*3P	§	ns
2	$t_{d(XBHZ-XHDAH)}$	Delay time, expansion bus high impedance to XHOLDA high	*0	*2P	ns
4	$t_{d(XHDL-XHDAL)}$	Delay time, XHOLD low to XHOLDA low	*3P		ns
5	$t_{d(XHDAL-XBLZ)}$	Delay time, XHOLDA low to expansion bus low impedance	*0	*2P	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ Expansion bus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.
§ All pending expansion bus transactions are allowed to complete before XHOLDA is asserted.



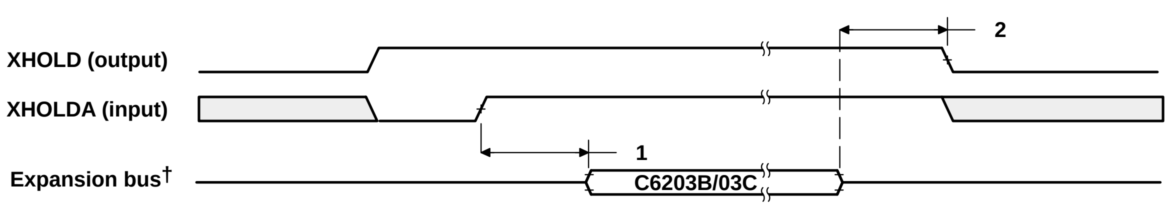
† Expansion bus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.

Figure 44. Expansion Bus Arbitration—Internal Arbiter Enabled

switching characteristics over recommended operating conditions for expansion bus arbitration (internal arbiter disabled)† (see Figure 45)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{d(XHDAH-XBLZ)}$	Delay time, XHOLDA high to Expansion bus low impedance‡	*2P	*2P + 10	ns
2	$t_{d(XBHZ-XHDL)}$	Delay time, expansion bus high impedance to XHOLD low‡	*0	*2P	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ Expansion bus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.



† Expansion bus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.

Figure 45. Expansion Bus Arbitration—Internal Arbiter Disabled

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP†‡ (see Figure 46)

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

*This parameter is not production tested.

† CLKRP = CLKXP = FSRP = FSXP = 0. If the 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 200 MHz, use P = 5 ns.

§ The maximum bit rate for the C6203 device is 100 Mbps or CPU/2 (the slower of the two). Care must be taken to ensure that the AC timings specified in this data sheet are met. The maximum bit rate for McBSP-to-McBSP communications is 100 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 200 MHz (P = 5 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 100 MHz (P = 10 ns), use 2P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies when 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.

¶ The minimum CLKR/X pulse duration is either (P–1) or 4 ns, whichever is larger. For example, when running parts at 200 MHz (P = 5 ns), use 4 ns as the minimum CLKR/X pulse duration. When running parts at 100 MHz (P = 10 ns), use (P–1) = 9 ns as the minimum CLKR/X pulse duration.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics over recommended operating conditions for McBSP†‡ (see Figure 46)

NO.	PARAMETER			MIN	MAX	UNIT
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		*4	*16	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	*2P§¶		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	*C – 2#	*C + 2#	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	*–3	*3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	*–3	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	*5	ns
			CLKX ext	*2	*9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	*–1	*4	ns
			CLKX ext	*2	*11	
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	*5	ns
			FSX ext	*0	*10	

*This parameter is not production tested.

† CLKRP = CLKXP = FSRP = FSXP = 0. If the 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 200 MHz, use P = 5 ns.

¶ The maximum bit rate for the C6203 device is 100 Mbps or CPU/2 (the slower of the two). Care must be taken to ensure that the AC timings specified in this data sheet are met. The maximum bit rate for McBSP-to-McBSP communications is 100 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 200 MHz (P = 5 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 100 MHz (P = 10 ns), use 2P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies when 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 = P 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 100-MHz limit.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

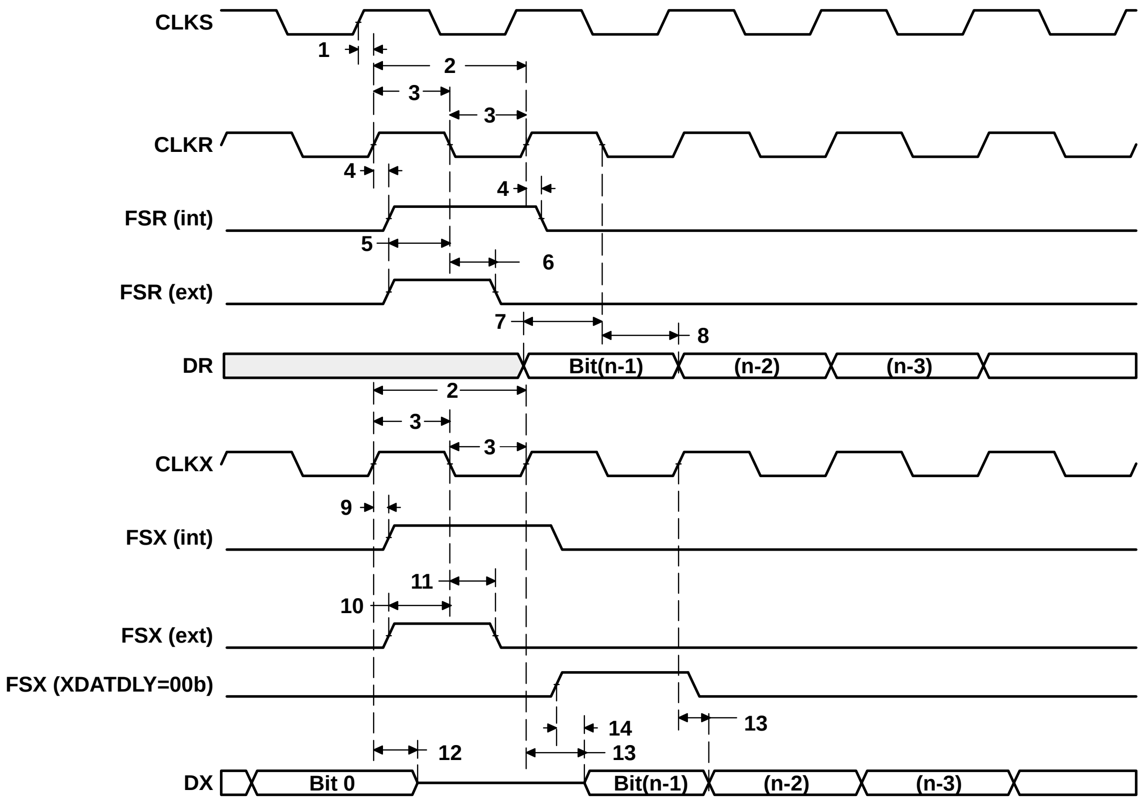


Figure 46. McBSP Timings

timing requirements for FSR when GSYNC = 1 (see Figure 47)

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

*This parameter is not production tested.

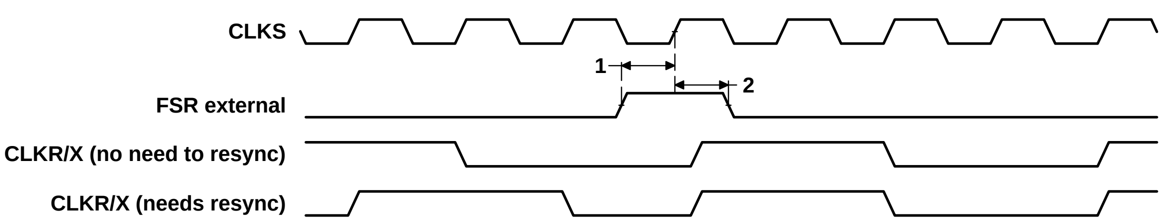


Figure 47. FSR Timing When GSYNC = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 48)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	*12		*2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	*4		*5 + 6P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 48)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	*T – 2	*T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	*L – 2	*L + 3			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	*–4	*4	*3P + 4	*5P + 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			*P + 3	*3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			*2P + 2	*4P + 17	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 = P 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
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100-MHz limit.
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

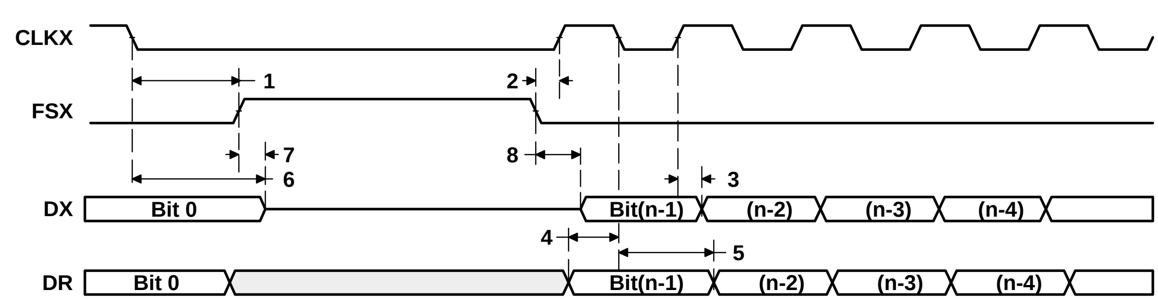


Figure 48. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 49)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	*12		*2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	*4		*5 + 6P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 49)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	*L – 2	*L + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	*T – 2	*T + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	*–4	*4	*3P + 4	*5P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	*–2	*4	*3P + 3	*5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	*H – 2	*H + 4	*2P + 2	*4P + 17	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 = P 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
  The maximum transfer rate for SPI mode is limited to the above AC timing constraints.
¶ 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).

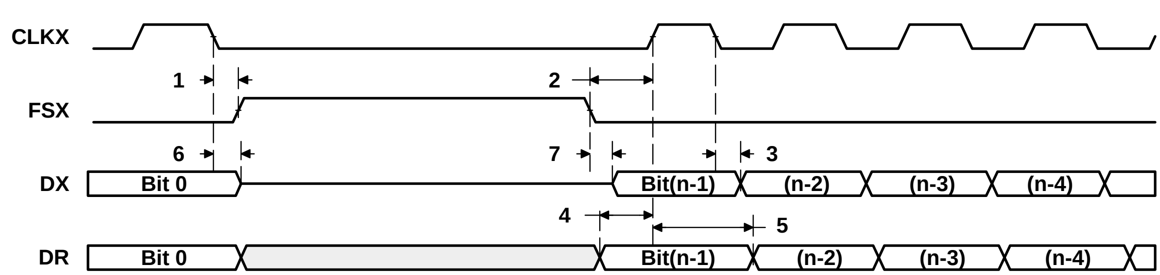


Figure 49. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 50)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	*12		*2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	*4		*5 + 6P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 50)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	*T – 2	*T + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	*H – 2	*H + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	*–4	*4	*3P + 4	*5P + 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			*P + 3	*3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			*2P + 2	*4P + 17	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 = P 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
The maximum transfer rate for SPI mode is limited to the above AC timing constraints.
¶ 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).

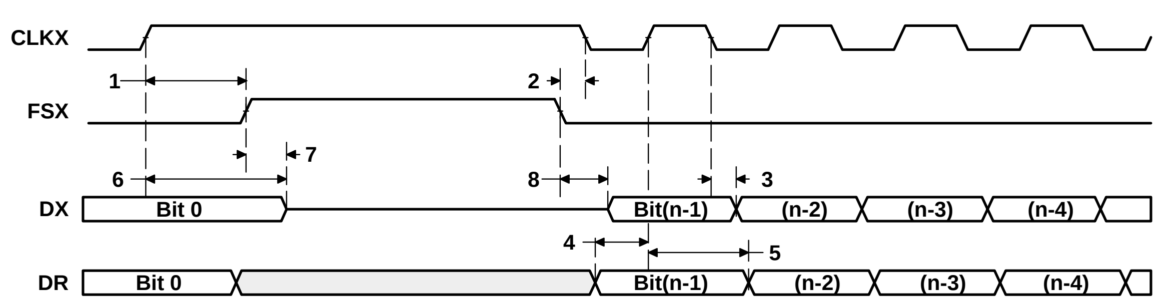


Figure 50. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 51)

NO.			MASTER		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	*12		*2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	*4		*5 + 6P		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 51)

NO.	PARAMETER		MASTER§		SLAVE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	*H – 2	*H + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	*T – 2	*T + 2			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	*–4	*4	*3P + 4	*5P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	*–2	*4	*3P + 3	*5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	*L – 2	*L + 5	*2P + 2	*4P + 17	ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 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 = P 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
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100-MHz limit.
¶ 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).

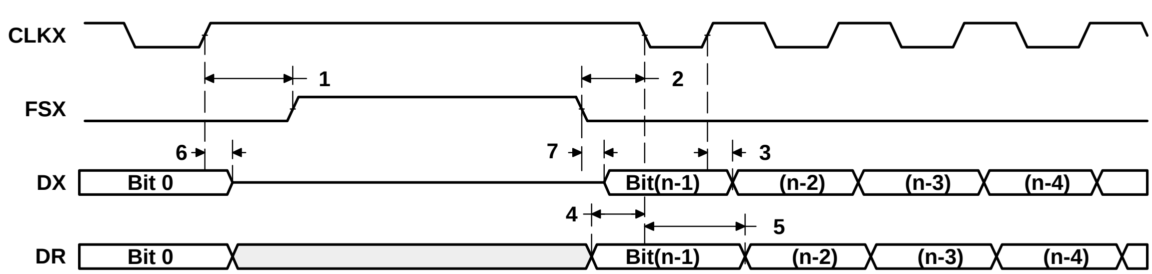


Figure 51. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

DMAC, TIMER, POWER-DOWN TIMING

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

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{w(DMACH)}$	Pulse duration, DMAC high	*2P–3		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

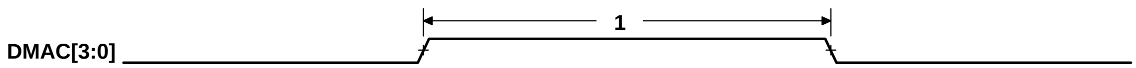


Figure 52. DMAC Timing

timing requirements for timer inputs† (see Figure 53)

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

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

switching characteristics over recommended operating conditions for timer outputs† (see Figure 53)

NO.	PARAMETER		MIN	MAX	UNIT
3	$t_{w(TOUTH)}$	Pulse duration, TOUT high	*2P–3		ns
4	$t_{w(TOUTL)}$	Pulse duration, TOUT low	*2P–3		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

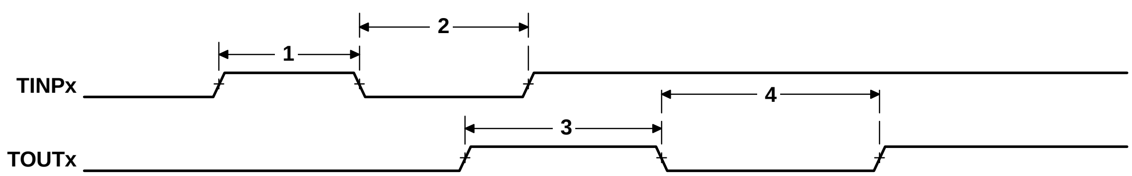


Figure 53. Timer Timing

DMAC, TIMER, POWER-DOWN TIMING (CONTINUED)

switching characteristics over recommended operating conditions for power-down outputs† (see Figure 54)

NO.	PARAMETER		MIN	MAX	UNIT
1	$t_{w(PDH)}$	Pulse duration, PD high	*2P–3		ns

*This parameter is not production tested.
† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

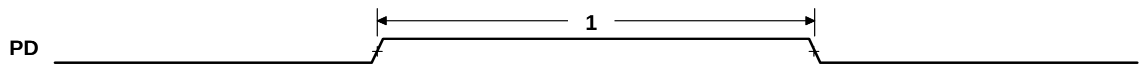


Figure 54. Power-Down Timing

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 55)

NO.			MIN	MAX	UNIT
1	$t_{c(TCK)}$	Cycle time, TCK	*35		ns
3	$t_{su(TDIV-TCKH)}$	Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high	*11		ns
4	$t_{h(TCKH-TDIV)}$	Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high	*9		ns

*This parameter is not production tested.

switching characteristics over recommended operating conditions for JTAG test port (see Figure 55)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid	*–4.5	*13.5	ns

*This parameter is not production tested.

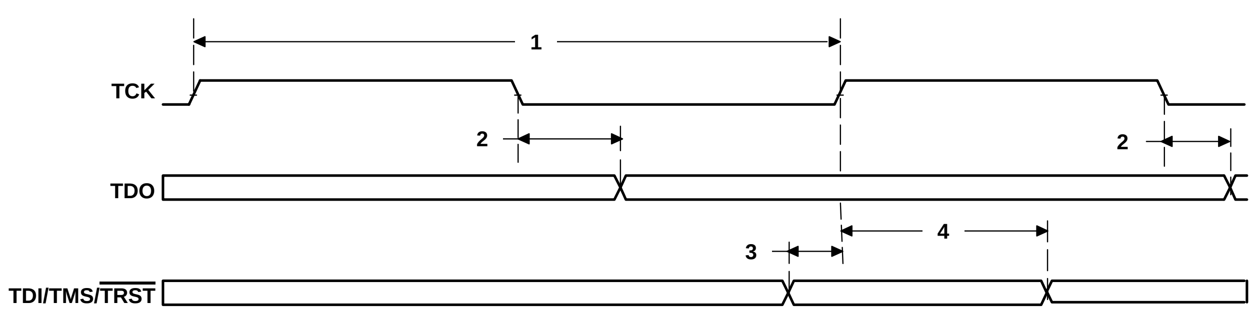
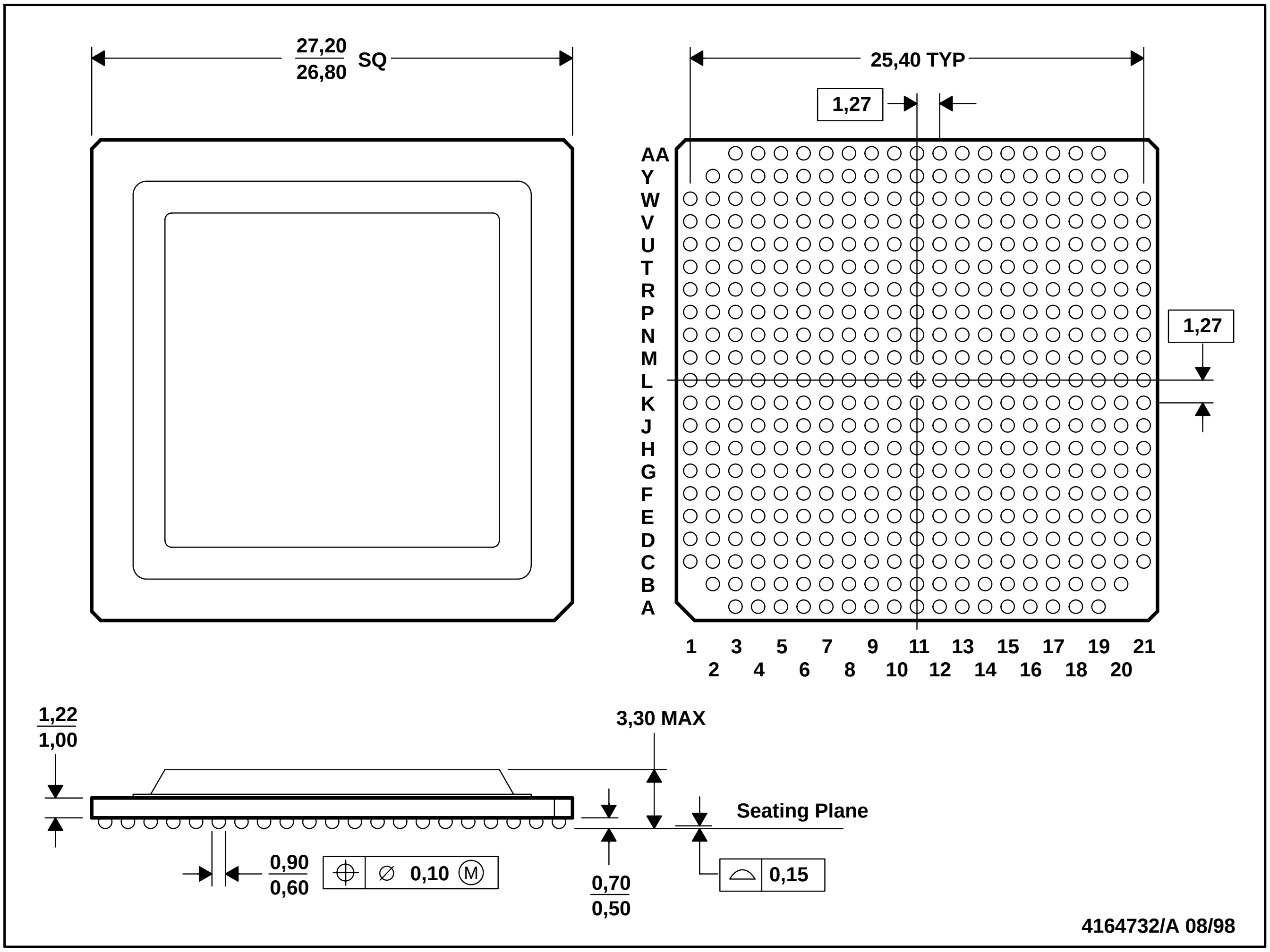


Figure 55. JTAG Test-Port Timing

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 • HOUSTON, TEXAS 77251–1443

MECHANICAL DATA

GLP (S-CBGA-N429) **CERAMIC BALL GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MO-156
D. Flip chip application only

thermal resistance characteristics (S-CBGA package)

NO			°C/W	Air Flow
1	$R\Theta_{JC}$	Junction-to-Case, measured to the bottom of solder ball	3.0	N/A
2	$R\Theta_{JC}$	Junction-to-Case, measured to the top of the package lid	7.3	N/A
3	$R\Theta_{JA}$	Junction-to-Ambient	14.5	0
4	$R\Theta_{JMA}$	Junction-to-Moving-Air	11.8	150 fpm
5			11.1	250 fpm
6			10.2	500 fpm
7	$R\Theta_{JB}$	Junction-to-Board, measured by soldering a thermocouple to one of the middle traces on the board at the edge of the package	6.2	N/A

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265