# SMJ320C6203 Fixed-Point Digital Signal Processor

## 1 Features

- 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
- 7Mb On-Chip SRAM
  - 3Mb Internal Program/Cache (96K 32-Bit Instructions)
  - 4Mb Dual-Access Internal Data (512KB)
  - Organized as Two 256KB 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
  - 52MB 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[2]) Boundary-Scan-Compatible
- 0.15-µm/5-Level Metal Process
  - CMOS Technology
- 3.3-V I/Os, 1.5-V Internal

## 2 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 VLIW architecture developed by 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.

**Device Information**[1]

| PART NUMBER | PACKAGE | BODY SIZE (NOM) |
|---|---|---|
| SMJ320C6203 | CFCBGA (429) | 27.00 mm × 27.00 mm × 2.26 mm |

[1] For all available packages, see the orderable addendum at the end of the data sheet.

[2] IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

# Table of Contents

# 3 Revision History

**Changes from Original (February 2002) to Revision A** **Page**

- Added *Feature Description* section, *Application and Implementation* section, *Power Supply Recommendations* section, *Device and Documentation Support* section, and *Mechanical, Packaging, and Orderable Information* section .......... 1
- Updated minimum values in *Timing Requirements for Synchronous-Burst SRAM Cycles* .......... 14
- Updated minimum values in *Switching Characteristics for Synchronous-Burst SRAM Cycles* .......... 21
- Updated minimum values in *Switching Characteristics for Synchronous DRAM Cycles* .......... 21
- Updated maximum values in *Switching Characteristics With External Device as Bus Master* .......... 23
- Updated maximum values in *Switching Characteristics With C62x as Bus Master* .......... 24

## 4 Description (continued)

The eight functional units provide six 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 256KB block configured as memory-mapped program space, and the other 128KB block user-configurable as cache or memory-mapped program space. Data memory for the C6203 consists of two 256KB blocks of RAM.

The C6203 device has a powerful and diverse set of peripherals. The peripheral set includes three 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 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.

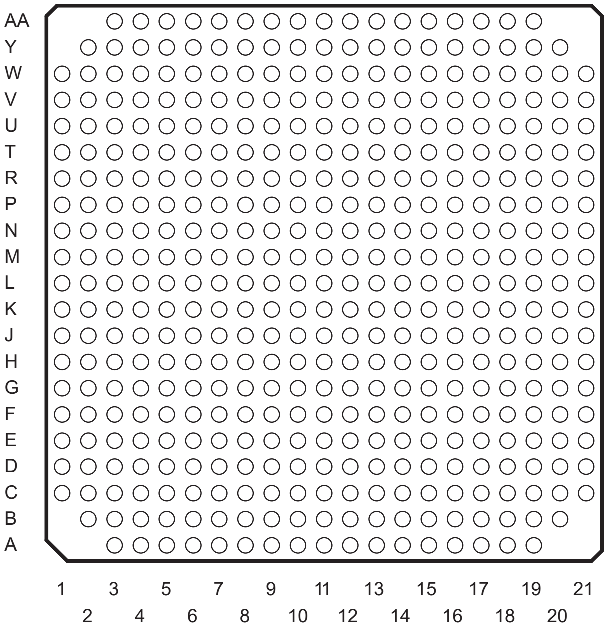
## 5 Characteristics of the C6203 DSP

This table shows significant features of the device, including the capacity of on-chip RAM, the peripherals, execution time, and 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 56.

| 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: 256KB mapped program<br>Block 1: 128KB cache/mapped program |
| 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 |

# 6 Pin Configuration and Functions

**GLP Package
429-Pin CFCBGA
Bottom View**



## Signal Descriptions

| SIGNAL NAME | PIN NO. | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| **CLOCK/PLL** | | | |
| CLKIN | D10 | I | Clock input |
| CLKOUT1 | Y17 | O | Clock output at full device speed |
| CLKOUT2 | Y16 | O | Clock output at half 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.<br>For more details on the CLKMODE pins and the PLL multiply factors for the C6203 device, see *Clock PLL* |
| CLKMODE1 | G10 | I | |
| CLKMODE2 | G12 | I | |
| PLLV[2] | B11 | A[3] | PLL analog $V_{CC}$ connection for the low-pass filter |
| PLLG[2] | A11 | A[3] | PLL analog GND connection for the low-pass filter |
| PLLF[2] | G11 | A[3] | 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 |

[1] I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

[2] PLLV, PLLG, and PLLF are not part of external voltage supply or ground. See *Clock PLL* for information on how to connect these pins.

[3] 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. | TYPE(1) | DESCRIPTION |
|---|---|---|---|
| **RESET AND INTERRUPTS** | | | |
| $\overline{\text{RESET}}$ | J4 | I | Device reset |
| NMI | K2 | I | Nonmaskable interrupt<br>Edge-driven (rising edge) |
| EXT_INT7 | R4 | I | External interrupts<br>• Edge-driven<br>• 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<br>• Valid during IACK for all active interrupts (not just external)<br>• Encoding order follows the interrupt-service fetch-packet ordering |
| INUM2 | P1 | | |
| INUM1 | P2 | | |
| INUM0 | N6 | | |
| **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 |

## Signal Descriptions (continued)

| SIGNAL NAME | PIN NO. | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| XD31 | D11 | I/O/Z | Expansion bus data<br>• Used for transfer of data, address, and control<br>• Also controls initialization of DSP modes and expansion bus at reset<br>Note: For more information on pin control and boot configuration fields, see *TMS320C6000 Peripherals Reference Guide* (SPRU190)<br>XD[30:16] − XCE[3:0] memory type<br>XD13 − XBLAST polarity<br>XD12 − XW/R polarity<br>XD11 − Asynchronous or synchronous host operation<br>XD10 − Arbitration mode (internal or external) XD9 − FIFO mode<br>XD8 − Little endian/big endian<br>XD7 − SCRT select<br>XD[4:0] − Boot mode<br>All other expansion bus data pins not listed should be pulled down.<br>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 | | |
| $\overline{\text{XCE3}}$ | D3 | O/Z | Expansion bus I/O port memory space enables<br>• Enabled by bits 28, 29, and 30 of the word address<br>• Only one asserted during any I/O port data access |
| $\overline{\text{XCE2}}$ | G6 | | |
| $\overline{\text{XCE1}}$ | D4 | | |
| $\overline{\text{XCE0}}$ | E4 | | |
| $\overline{\text{XBE3}}$/XA5 | F6 | I/O/Z | Expansion bus multiplexed byte-enable control/address signals<br>• Act as byte-enable for host-port operation<br>• Act as address for I/O port operation |
| $\overline{\text{XBE2}}$/XA4 | F7 | | |
| $\overline{\text{XBE1}}$/XA3 | B5 | | |
| $\overline{\text{XBE0}}$/XA2 | C7 | | |
| $\overline{\text{XOE}}$ | B7 | O/Z | Expansion bus I/O port output-enable |
| $\overline{\text{XRE}}$ | B8 | O/Z | Expansion bus I/O port read-enable |
| $\overline{\text{XWE}}$/$\overline{\text{XWAIT}}$ | D7 | O/Z | Expansion bus I/O port write-enable and host-port wait signals |
| $\overline{\text{XCS}}$ | D8 | I | Expansion bus host-port chip-select input |
| $\overline{\text{XAS}}$ | 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. |

## Signal Descriptions (continued)

| SIGNAL NAME | PIN NO. | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| 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** | | | |
| $\overline{\text{CE3}}$ | V18 | O/Z | Memory space enables<br>• Enabled by bits 24 and 25 of the word address<br>• Only one asserted during any external data access |
| $\overline{\text{CE2}}$ | W18 | | |
| $\overline{\text{CE1}}$ | T15 | | |
| $\overline{\text{CE0}}$ | U18 | | |
| $\overline{\text{BE3}}$ | R15 | O/Z | Byte-enable control<br>• Decoded from the two lowest bits of the internal address<br>• Byte-write enables for most types of memory<br>• Can be directly connected to SDRAM read and write mask signal (SDQM) |
| $\overline{\text{BE2}}$ | V19 | | |
| $\overline{\text{BE1}}$ | U20 | | |
| $\overline{\text{BE0}}$ | V16 | | |
| **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 | | |

## Signal Descriptions (continued)

| SIGNAL NAME | PIN NO. | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| **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 | | |
| ED13 | W10 | | |
| ED12 | W12 | | |
| ED11 | Y11 | | |
| ED10 | Y12 | | |
| ED9 | T12 | | |
| ED8 | AA13 | | |
| ED7 | R12 | | |
| ED6 | V13 | | |
| ED5 | Y13 | | |
| ED4 | Y14 | I/O/Z | External data |
| 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 |
| $\overline{\text{ARDY}}$ | 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 |

## Signal Descriptions (continued)

| SIGNAL NAME | PIN NO. | TYPE[1] | DESCRIPTION |
|---|---|---|---|
| **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 | | |
| **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 | — | |

## Signal Descriptions (continued)

| SIGNAL NAME | PIN NO. | TYPE[1] | 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, A3, A5, A7, A12, A13, A16, A18, B2, B4, B6, B10, B12, B19, C1, C3, C20, D2, D15, D16, D21, E1, E6, E8, E10, E12, E14, E16 | S | 3.3-V supply voltage (I/O) |
| $CV_{DD}$ - 1.5 V | F5, F8, F10, F14, F17, F20, F21, G1, G7, G8, G13, G18, H5, H16, 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, Y18, Y20, AA4, AA6, AA11, AA12, AA15, AA17, AA19 | S | 1.5-V supply voltage (core) |
| **GROUND PINS** | | | |
| VSS | A4, 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, G19, 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, P10, 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 | GND | Ground pins |

# 7 Specifications

## 7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted) [1]

| | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| | Supply voltage | $CV_{DD}$ [2] | –0.3 | 1.8 | V |
| | | $DV_{DD}$ [2] | –0.3 | 4 | |
| | Input voltage | | –0.3 | 4 | V |
| | Output voltage | | –0.3 | 4 | V |
| $T_C$ | Operating case temperature | | –55 | 125 | °C |
| | Temperature cycle (1000-cycle performance) | | –55 | 125 | °C |
| $T_{stg}$ | Storage temperature | | –65 | 150 | °C |

[1] Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

[2] All voltage values are with respect to $V_{SS}$.

## 7.2 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

| | | 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 [1] | | | 2 | V |
| $V_{IL}$ | Low-level input voltage [2] | | | 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 |

[1] $V_{IH}$ is not production tested for: CLKMODE [2:0], CLKIN, XCLKIN, XCS.

[2] $V_{IL}$ is not production tested for: CLKIN, $\overline{\text{TRST}}$.

## 7.3 Thermal Information

| THERMAL METRIC [1] | | | SMJ320C6203 | UNIT |
|---|---|---|---|---|
| | | | GLP (CFCBGA) | |
| | | | 529 PINS | |
| $R_{\theta JA}$ | Junction-to-ambient thermal resistance | | 14.5 | °C/W |
| $R_{\theta JC(top)}$ | Junction-to-case (top) thermal resistance, measured to top of the package lid | | 7.3 | °C/W |
| $R_{\theta JB}$ | Junction-to-board thermal resistance, measured by soldering a thermocouple to one of the middle traces on the board at the edge of the package | | 6.2 | °C/W |
| $R_{\theta JC(bot)}$ | Junction-to-case (bottom) thermal resistance, measured to bottom of solder ball | | 3.0 | °C/W |
| $R_{\theta JMA}$ | Junction-to-moving air thermal resistance | 150 fpm | 11.8 | °C/W |
| | | 250 fpm | 11.1 | |
| | | 500 fpm | 10.2 | |

[1] For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.

## 7.4 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[1] | $DV_{DD}$ = MIN, $I_{OH}$ = MAX | | 2.4 | | V |
| $V_{OL}$ | Low-level output voltage[1] | $DV_{DD}$ = MIN, $I_{OL}$ = MAX | | 0.6 | | V |
| $I_I$ | Input current[2] | $V_I$ = $V_{SS}$ to $DV_{DD}$ | | ±10 | | µA |
| $I_{OZ}$ | Off-state output current[3] | $V_O$ = $DV_{DD}$ or 0 V | | ±10 | | µA |
| $I_{DD2V}$ | Supply current, CPU + CPU memory access[4] | $CV_{DD}$ = NOM, CPU clock = 200 MHz | | 340 | | mA |
| $I_{DD2V}$ | Supply current, peripherals[4] | $CV_{DD}$ = NOM, CPU clock = 200 MHz | | 235 | | mA |
| $I_{DD3V}$ | Supply current, I/O pins[4] | $CV_{DD}$ = NOM, CPU clock = 200 MHz | | 45 | | mA |
| $C_i$ | Input capacitance | | | | 12 | pF |
| $C_o$ | Output capacitance | | | | 15 | pF |

[1] $V_{OH}$ and $V_{OL}$ are not production tested for: CLKOUT1, $\overline{\text{EMU}}0$, and EMU1.
[2] TMS and TDI are not included due to internal pullups. $\overline{\text{TRST}}$ is not included due to internal pulldown.
[3] TDO is not production tested.
[4] Measured with average activity (50% high power/ 50% low power). For more details on CPU, peripheral, and I/O activity, see the *TMS320C6000 Power Consumption Summary* application report (SPRA486).

## 7.5 Timing Requirements for CLKIN (PLL Used)

see Figure 5[1][2][3]

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

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

## 7.6 Timing Requirements for CLKIN [PLL Bypassed (x1)]

see Figure 5[1][2]

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

[1] The reference points for the rise and fall transitions are measured at $V_{IL}$ MAX and $V_{IH}$ MIN.
[2] 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.
[3] This parameter is not production tested.

## 7.7 Timing Requirements for XCLKIN

see Figure 6 [1]

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

[1] P = 1 / CPU clock frequency in ns.
[2] This parameter is not production tested.

## 7.8 Timing Requirements for Asynchronous Memory Cycles

see Figure 9 through Figure 12 [1][2][3][4]

| 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) × P − 6] | | ns |
| 7 | $t_{h(AREL-ARDYH)}$ | Hold time, ARDY high after $\overline{\text{ARE}}$ low | (RST − 3) × P + 2 | | ns |
| 9 | $t_{su(ARDYL-AREL)}$ | Setup time, ARDY low before $\overline{\text{ARE}}$ low | −[(RST − 3) × P − 6] | | ns |
| 10 | $t_{h(AREL-ARDYL)}$ | Hold time, ARDY low after $\overline{\text{ARE}}$ low | (RST − 3) × P + 2 | | ns |
| 11 | $t_{w(ARDYH)}$ | Pulse duration, ARDY high | [5]2P | | ns |
| 15 | $t_{su(ARDYH-AWEL)}$ | Setup time, ARDY high before $\overline{\text{AWE}}$ low | −[(WST − 3) × P − 6] | | ns |
| 16 | $t_{h(AWEL-ARDYH)}$ | Hold time, ARDY high after $\overline{\text{AWE}}$ low | (WST − 3) × P + 2 | | ns |
| 18 | $t_{su(ARDYL-AWEL)}$ | Setup time, ARDY low before $\overline{\text{AWE}}$ low | −[(WST − 3) × P − 6] | | ns |
| 19 | $t_{h(AWEL-ARDYL)}$ | Hold time, ARDY low after $\overline{\text{AWE}}$ low | (WST − 3) × P + 2 | | ns |

[1] 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.

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

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

[4] The sum of RS and RST (or WS and WST) must be a minimum of 4 to use ARDY input to extend strobe width.

[5] This parameter is not production tested.

## 7.9 Timing Requirements for Synchronous-Burst SRAM Cycles

see Figure 13

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 7 | $t_{su(EDV-CKO2H)}$ | Setup time, read EDx valid before CLKOUT2 high | 2.9 | | ns |
| 8 | $t_{h(CKO2H-EDV)}$ | Hold time, read EDx valid after CLKOUT2 high | 2.3 | | ns |

## 7.10 Timing Requirements for Synchronous DRAM Cycles

see Figure 15

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 7 | $t_{su(EDV-CKO2H)}$ | Setup time, read EDx valid before CLKOUT2 high | 1.3 | | ns |
| 8 | $t_{h(CKO2H-EDV)}$ | Hold time, read EDx valid after CLKOUT2 high | 2.9 | | ns |

## 7.11 Timing Requirements for the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ Cycles

see Figure 21 [1]

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{oh(HOLDAL-HOLDL)}$ | Output hold time, $\overline{\text{HOLD}}$ low after $\overline{\text{HOLDA}}$ low | [2]P | | ns |

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

[2] This parameter is not production tested.

## 7.12 Timing Requirements for Reset

see Figure 22 [1]

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(RST)}$ | Duration of the $\overline{\text{RESET}}$ pulse (PLL stable)[2] | [3]10P | | ns |
| | | Duration of the $\overline{\text{RESET}}$ pulse (PLL needs to sync up)[4] | [3]250 | | µs |
| 10 | $t_{su(XD)}$ | Setup time, XD configuration bits valid before $\overline{\text{RESET}}$ high[5] | [3]5P | | ns |
| 11 | $t_{h(XD)}$ | Hold time, XD configuration bits valid after $\overline{\text{RESET}}$ high[5] | [3]5P | | ns |

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] 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.
[3] This parameter is not production tested.
[4] This parameter applies to CLKMODE x4, x6, x7, x8, x9, x10, and x11 only. (It does not apply to CLKMODE x1.) The $\overline{\text{RESET}}$ signal is not connected internally to the clock PLL circuit. However, $\overline{\text{the PLL}}$ may need up to 250 µs to stabilize following device power-up or after the PLL configuration has been changed. During that time, $\overline{\text{RESET}}$ must be asserted to ensure proper device operation. See *Clock PLL* for PLL lock times.
[5] XD[31:0] are the boot configuration pins during device reset.

## 7.13 Timing Requirements for Interrupt Response Cycles

see Figure 23[1]

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 2 | $t_{w(ILOW)}$ | Duration of the interrupt pulse low | [2]2P | | ns |
| 3 | $t_{w(IHIGH)}$ | Duration of the interrupt pulse high | [2]2P | | ns |

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

## 7.14 Timing Requirements for Synchronous FIFO Interface

see Figure 24 through Figure 26

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

## 7.15 Timing Requirements for Asynchronous Peripheral Cycles

see Figure 27 through Figure 30[1][2][3][4]

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{su(XDV-XREH)}$ | Setup time, XDx valid before $\overline{\text{XRE}}$ high | 4.5 | | ns |
| 4 | $t_{h(XREH-XDV)}$ | Hold time, XDx valid after $\overline{\text{XRE}}$ high | 2.5 | | ns |
| 6 | $t_{su(XRDYH-XREL)}$ | Setup time, XRDY high before $\overline{\text{XRE}}$ low | $-[(RST - 3) \times P - 6]$ | | ns |
| 7 | $t_{h(XREL-XRDYH)}$ | Hold time, XRDY high after $\overline{\text{XRE}}$ low | $(RST - 3) \times P + 2$ | | ns |
| 9 | $t_{su(XRDYL-XREL)}$ | Setup time, XRDY low before $\overline{\text{XRE}}$ low | $-[(RST - 3) \times P - 6]$ | | ns |
| 10 | $t_{h(XREL-XRDYL)}$ | Hold time, XRDY low after $\overline{\text{XRE}}$ low | $(RST - 3) \times P + 2$ | | ns |
| 11 | $t_{w(XRDYH)}$ | Pulse duration, XRDY high | [5]2P | | ns |
| 15 | $t_{su(XRDYH-XWEL)}$ | Setup time, XRDY high before $\overline{\text{XWE}}$ low | $-[(WST - 3) \times P - 6]$ | | ns |
| 16 | $t_{h(XWEL-XRDYH)}$ | Hold time, XRDY high after $\overline{\text{XWE}}$ low | $(WST - 3) \times P + 2$ | | ns |
| 18 | $t_{su(XRDYL-XWEL)}$ | Setup time, XRDY low before $\overline{\text{XWE}}$ low | $-[(WST - 3) \times P - 6]$ | | ns |
| 19 | $t_{h(XWEL-XRDYL)}$ | Hold time, XRDY low after $\overline{\text{XWE}}$ low | $(WST - 3) \times P + 2$ | | ns |

[1] 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.
[2] RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed by the expansion bus XCE space control registers.
[3] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[4] The sum of RS and RST (or WS and WST) must be a minimum of 4 to use XRDY input to extend strobe width.
[5] This parameter is not production tested.

## 7.16 Timing Requirements With External Device as Bus Master

see Figure 31 and Figure 32

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

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

## 7.17 Timing Requirements With C62x as Bus Master

see Figure 33 through Figure 35

| 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[1] | 3.5 | | ns |
| 12 | $t_{h(XCKIH-XRY)}$ | Hold time, XRDY valid after XCLKIN high[1] | 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 |

[1] XRDY operates as active-low ready input/output during host-port accesses.

## 7.18 Timing Requirements With External Device as Asynchronous Bus Master

see Figure 36 and Figure 37 [1]

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

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

## 7.19 Timing Requirements for Expansion Bus Arbitration (Internal Arbiter Enabled)

see Figure 38[1]

| NO. | | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{oh(XHDAH-XHDH)}$ | Output hold time, XHOLD high after XHOLDA high | [2]P | | ns |

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

## 7.20 Timing Requirements for McBSP

see Figure 40[1][2]

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

[1] 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.
[2] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[3] The maximum bit rate for the C6203 device is 100 Mbps or CPU / 2 (the slower of the two). Take care 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.
[4] This parameter is not production tested.
[5] 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.

## 7.21 Timing Requirements for FSR when GSYNC = 1

see Figure 41

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

[1] This parameter is not production tested.

## 7.22 Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

see Figure 42[1][2]

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] This parameter is not production tested.

## 7.23 Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

see Figure 43[1][2]

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] This parameter is not production tested.

## 7.24 Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

see Figure 44[1][2]

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] This parameter is not production tested.

## 7.25 Timing Requirements for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

see Figure 45[1][2]

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] This parameter is not production tested.

## 7.26 Timing Requirements for Timer Inputs

see Figure 47[1]

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

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

## 7.27 Timing Requirements for JTAG Test Port

see Figure 49

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

[1] This parameter is not production tested.

## 7.28 Switching Characteristics for CLKOUT2

over recommended operating conditions for CLKOUT2[1][2] (see Figure 7)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{\text{c(CKO2)}}$ | Cycle time, CLKOUT2 | [3]2P − 0.7 | [3]2P + 0.7 | ns |
| 2 | $t_{\text{w(CKO2H)}}$ | Pulse duration, CLKOUT2 high | [3]P − 0.7 | [3]P + 0.7 | ns |
| 3 | $t_{\text{w(CKO2L)}}$ | Pulse duration, CLKOUT2 low | [3]P − 0.7 | [3]P + 0.7 | ns |

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

## 7.29 Switching Characteristics for XFCLK

over recommended operating conditions for XFCLK[1][2] (see Figure 8)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{\text{c(XFCK)}}$ | Cycle time, XFCLK | [3]D × P − 0.7 | [3]D × P + 0.7 | ns |
| 2 | $t_{\text{w(XFCKH)}}$ | Pulse duration, XFCLK high | [3](D/2) × P − 0.7 | [3](D/2) × P + 0.7 | ns |
| 3 | $t_{\text{w(XFCKL)}}$ | Pulse duration, XFCLK low | [3](D/2) × P − 0.7 | [3](D/2) × P + 0.7 | ns |

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

## 7.30 Asynchronous Memory Timing Switching Characteristics

over recommended operating conditions for asynchronous memory cycles[1][2][3][4] (see Figure 9 through Figure 12)

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

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

## 7.31 Switching Characteristics for Synchronous-Burst SRAM Cycles

over recommended operating conditions for synchronous-burst SRAM cycles[1][2] (see Figure 13 and Figure 14)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{osu(CEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{CEx}}$ valid before CLKOUT2 high | P − 1.7 | | ns |
| 2 | $t_{oh(CKO2H\text{-}CEV)}$ | Output hold time, $\overline{\mathrm{CEx}}$ valid after CLKOUT2 high | [3]P − 4 | | ns |
| 3 | $t_{osu(BEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{BEx}}$ valid before CLKOUT2 high | P − 1.7 | | ns |
| 4 | $t_{oh(CKO2H\text{-}BEIV)}$ | Output hold time, $\overline{\mathrm{BEx}}$ invalid after CLKOUT2 high | [3]P − 4 | | ns |
| 5 | $t_{osu(EAV\text{-}CKO2H)}$ | Output setup time, EAx valid before CLKOUT2 high | P − 1.7 | | ns |
| 6 | $t_{oh(CKO2H\text{-}EAIV)}$ | Output hold time, EAx invalid after CLKOUT2 high | [3]P − 4 | | ns |
| 9 | $t_{osu(ADSV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDCAS}}$/$\overline{\mathrm{SSADS}}$ valid before CLKOUT2 high | P − 1.7 | | ns |
| 10 | $t_{oh(CKO2H\text{-}ADSV)}$ | Output hold time, $\overline{\mathrm{SDCAS}}$/$\overline{\mathrm{SSADS}}$ valid after CLKOUT2 high | [3]P − 4 | | ns |
| 11 | $t_{osu(OEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDRAS}}$/$\overline{\mathrm{SSOE}}$ valid before CLKOUT2 high | P − 1.7 | | ns |
| 12 | $t_{oh(CKO2H\text{-}OEV)}$ | Output hold time, $\overline{\mathrm{SDRAS}}$/$\overline{\mathrm{SSOE}}$ valid after CLKOUT2 high | [3]P − 4 | | ns |
| 13 | $t_{osu(EDV\text{-}CKO2H)}$ | Output setup time, EDx valid before CLKOUT2 high[4] | P − 2.3 | | ns |
| 14 | $t_{oh(CKO2H\text{-}EDIV)}$ | Output hold time, EDx invalid after CLKOUT2 high | [3]P − 4 | | ns |
| 15 | $t_{osu(WEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDWE}}$/$\overline{\mathrm{SSWE}}$ valid before CLKOUT2 high | P − 1.7 | | ns |
| 16 | $t_{oh(CKO2H\text{-}WEV)}$ | Output hold time, $\overline{\mathrm{SDWE}}$/$\overline{\mathrm{SSWE}}$ valid after CLKOUT2 high | [3]P − 4 | | ns |

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] $\overline{\mathrm{SDCAS/SSADS}}$, $\overline{\mathrm{SDRAS/SSOE}}$, and $\overline{\mathrm{SDWE/SSWE}}$ operate as $\overline{\mathrm{SSADS}}$, $\overline{\mathrm{SSOE}}$, and $\overline{\mathrm{SSWE}}$, respectively, during SBSRAM accesses.
[3] This parameter is not production tested.
[4] 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.

## 7.32 Switching Characteristics for Synchronous DRAM Cycles

over recommended operating conditions for synchronous DRAM cycles for C6203B Rev. 2[1][2] (see Figure 15 through Figure 20)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{osu(CEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{CEx}}$ valid before CLKOUT2 high | P − 0.9 | | ns |
| 2 | $t_{oh(CKO2H\text{-}CEV)}$ | Output hold time, $\overline{\mathrm{CEx}}$ valid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 3 | $t_{osu(BEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{BEx}}$ valid before CLKOUT2 high | P − 0.9 | | ns |
| 4 | $t_{oh(CKO2H\text{-}BEIV)}$ | Output hold time, $\overline{\mathrm{BEx}}$ invalid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 5 | $t_{osu(EAV\text{-}CKO2H)}$ | Output setup time, EAx valid before CLKOUT2 high | P − 0.9 | | ns |
| 6 | $t_{oh(CKO2H\text{-}EAIV)}$ | Output hold time, EAx invalid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 9 | $t_{osu(CASV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDCAS}}$/$\overline{\mathrm{SSADS}}$ valid before CLKOUT2 high | P − 0.9 | | ns |
| 10 | $t_{oh(CKO2H\text{-}CASV)}$ | Output hold time, $\overline{\mathrm{SDCAS}}$/$\overline{\mathrm{SSADS}}$ valid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 11 | $t_{osu(EDV\text{-}CKO2H)}$ | Output setup time, EDx valid before CLKOUT2 high[4] | P − 1.5 | | ns |
| 12 | $t_{oh(CKO2H\text{-}EDIV)}$ | Output hold time, EDx invalid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 13 | $t_{osu(WEV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDWE}}$/$\overline{\mathrm{SSWE}}$ valid before CLKOUT2 high | P − 0.9 | | ns |
| 14 | $t_{oh(CKO2H\text{-}WEV)}$ | Output hold time, $\overline{\mathrm{SDWE}}$/$\overline{\mathrm{SSWE}}$ valid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 15 | $t_{osu(SDA10V\text{-}CKO2H)}$ | Output setup time, SDA10 valid before CLKOUT2 high | P − 0.9 | | ns |
| 16 | $t_{oh(CKO2H\text{-}SDA10IV)}$ | Output hold time, SDA10 invalid after CLKOUT2 high | [3]P − 4.1 | | ns |
| 17 | $t_{osu(RASV\text{-}CKO2H)}$ | Output setup time, $\overline{\mathrm{SDRAS}}$/$\overline{\mathrm{SSOE}}$ valid before CLKOUT2 high | P − 0.9 | | ns |
| 18 | $t_{oh(CKO2H\text{-}RASV)}$ | Output hold time, $\overline{\mathrm{SDRAS}}$/$\overline{\mathrm{SSOE}}$ valid after CLKOUT2 high | [3]P − 4.1 | | ns |

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] $\overline{\mathrm{SDCAS/SSADS}}$, $\overline{\mathrm{SDRAS/SSOE}}$, and $\overline{\mathrm{SDWE/SSWE}}$ operate as $\overline{\mathrm{SDCAS}}$, $\overline{\mathrm{SDRAS}}$, and $\overline{\mathrm{SDWE}}$, respectively, during SDRAM accesses.
[3] This parameter is not production tested.
[4] 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.

## 7.33 Switching Characteristics for the HOLD/HOLDA Cycles

over recommended operating conditions for the HOLD/HOLDA cycles[1][2] (see Figure 21)

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] 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.
[3] This parameter is not production tested.
[4] 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.

## 7.34 Switching Characteristics for Reset

over recommended operating conditions during reset[1][2] (see Figure 22)

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] 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
[3] This parameter is not production tested.

## 7.35 Switching Characteristics for Interrupt Response Cycles

over recommended operating conditions during interrupt response cycles[1][2] (see Figure 23)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{R(EINTH - IACKH)}$ | Response time, EXT_INTx high to IACK high | [3]9P | | ns |
| 4 | $t_{d(CKO2L-IACKV)}$ | Delay time, CLKOUT2 low to IACK valid | [3]−1.5 | [3]10 | ns |
| 5 | $t_{d(CKO2L-INUMV)}$ | Delay time, CLKOUT2 low to INUMx valid | [3]−2.0 | [3]10 | ns |
| 6 | $t_{d(CKO2L-INUMIV)}$ | Delay time, CLKOUT2 low to INUMx invalid | [3]−2.0 | [3]10 | ns |

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

## 7.36 Switching Characteristics for Synchronous FIFO Interface

over recommended operating conditions for synchronous FIFO interface (see Figure 24 through Figure 26)

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

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

## 7.37 Switching Characteristics for Asynchronous Peripheral Cycles

over recommended operating conditions for asynchronous peripheral cycles[1][2][3][4] (see Figure 27 through Figure 30)

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

[1] RS = Read setup, RST = Read strobe, RH = Read hold, WS = Write setup, WST = Write strobe, WH = Write hold. These parameters are programmed by the expansion bus XCE space control registers.
[2] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[3] The sum of RS and RST (or WS and WST) must be a minimum of 4 to use XRDY input to extend strobe width.
[4] Select signals include: $\overline{\text{XCE}}$x, $\overline{\text{XBE[3:0]}}$/XA[5:2], $\overline{\text{XOE}}$; and for writes, include XD[31:0], with the exception that $\overline{\text{XCE}}$x can stay active for an additional 7P ns following the end of the cycle.
[5] This parameter is not production tested.

## 7.38 Switching Characteristics With External Device as Bus Master

over recommended operating conditions with external device as bus master[1] (see Figure 31 and Figure 32)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 11 | $t_{d(XCKIH-XDLZ)}$ | Delay time, XCLKIN high to XDx low impedance | [2]0 | | ns |
| 12 | $t_{d(XCKIH-XDV)}$ | Delay time, XCLKIN high to XDx valid | | 4P | ns |
| 13 | $t_{d(XCKIH-XDIV)}$ | Delay time, XCLKIN high to XDx invalid | [2]5 | | ns |
| 14 | $t_{d(XCKIH-XDHZ)}$ | Delay time, XCLKIN high to XDx high impedance | | [2]4P | ns |
| 15 | $t_{d(XCKIH-XRY)}$ | Delay time, XCLKIN high to XRDY invalid[3] | [2]5 | [2]4P | ns |
| 20 | $t_{d(XCKIH-XRYLZ)}$ | Delay time, XCLKIN high to XRDY low impedance | [2]5 | [2]4P | ns |
| 21 | $t_{d(XCKIH-XRYHZ)}$ | Delay time, XCLKIN high to XRDY high impedance[3] | [2]2P + 5 | [2]7P | ns |

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

## 7.39 Switching Characteristics With C62x as Bus Master

over recommended operating conditions with C62x as bus master[1] (see Figure 33 through Figure 35)

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

(1) P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
(2) This parameter is not production tested.
(3) XW/R input/output polarity selected at boot.
(4) XBLAST output polarity is always active low.
(5) XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
(6) XWE/XWAIT operates as XWAIT output signal during host-port accesses.

## 7.40 Switching Characteristics With External Device as Asynchronous Bus Master

over recommended operating conditions with external device as asynchronous bus master[1] (see Figure 36 and Figure 37)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 5 | $t_{d(XCSL-XDLZ)}$ | Delay time, XCS low to XDx low impedance | [2]0 | | ns |
| 6 | $t_{d(XCSH-XDIV)}$ | Delay time, XCS high to XDx invalid | [2]0 | [2]12 | ns |
| 7 | $t_{d(XCSH-XDHZ)}$ | Delay time, XCS high to XDx high impedance | | [2]4P | ns |
| 8 | $t_{d(XRYL-XDV)}$ | Delay time, XRDY low to XDx valid | [2]–4 | [2]1 | ns |
| 9 | $t_{d(XCSH-XRYH)}$ | Delay time, XCS high to XRDY high | [2]0 | 12 | ns |

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

## 7.41 Switching Characteristics for Expansion Bus Arbitration (Internal Arbiter Enabled)

over recommended operating conditions for expansion bus arbitration (internal arbiter enabled)[1][2] (see Figure 38)

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

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

## 7.42 Switching Characteristics for Expansion Bus Arbitration (Internal Arbiter Disabled)

over recommended operating conditions for expansion bus arbitration (internal arbiter disabled)[1] (see Figure 39)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{d(XHDAH-XBLZ)}$ | Delay time, XHOLDA high to expansion bus low impedance[2] | [3]2P | [3]2P + 10 | ns |
| 2 | $t_{d(XBHZ-XHDL)}$ | Delay time, expansion bus high impedance to XHOLD low[2] | [3]0 | [3]2P | ns |

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

## 7.43 Switching Characteristics for McBSP

over recommended operating conditions for McBSP[1][2] (see Figure 40)

| 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 | | [3]4 | [3]16 | ns |
| 2 | $t_{c(CKRX)}$ | Cycle time, CLKR/X | CLKR/X int | [3]2P[4][5] | | ns |
| 3 | $t_{w(CKRX)}$ | Pulse duration, CLKR/X high or CLKR/X low | CLKR/X int | [3]C − 2[6] | [3]C + 2[6] | ns |
| 4 | $t_{d(CKRH-FRV)}$ | Delay time, CLKR high to internal FSR valid | CLKR int | [3]−3 | [3]3 | ns |
| | | | CLKX int | [3]−3 | 3 | |
| 9 | $t_{d(CKXH-FXV)}$ | Delay time, CLKX high to internal FSX valid | CLKX ext | [3]−3 | 9 | ns |
| | | | CLKX int | [3]−1 | [3]5 | |
| 12 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | CLKX ext | [3]2 | [3]9 | ns |
| | | | CLKX int | [3]−1 | [3]4 | |
| 13 | $t_{d(CKXH-DXV)}$ | Delay time, CLKX high to DX valid | CLKX ext | [3]2 | [3]11 | ns |
| 14 | $t_{d(FXH-DXV)}$ | Delay time, FSX high to DX valid *only* applies when in data delay 0 (XDATDLY = 00b) mode. | FSX int | [3]−1 | [3]5 | ns |
| | | | FSX ext | [3]0 | [3]10 | |

[1] 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.

[2] Minimum delay times also represent minimum output hold times.

[3] This parameter is not production tested.

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

[5] The maximum bit rate for the C6203 device is 100 Mbps or CPU / 2 (the slower of the two). Take care 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.

[6] 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 duration = (CLKGDV/2 + 1) × S if CLKGDV is even
= (CLKGDV + 1) / 2 × S if CLKGDV is odd or zero
L = CLKX low pulse duration = (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.

## 7.44 Switching Characteristics for McBSP as SPI Master or Slave

over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0[1][2] (see Figure 42)

| NO. | PARAMETER | | MASTER[3] | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXL-FXL)}$ | Hold time, FSX low after CLKX low[4] | [5]T − 2 | [5]T + 3 | | | ns |
| 2 | td(FXL-CKXH) | Delay time, FSX low to CLKX high[6] | [5]L − 2 | [5]L + 3 | | | ns |
| 3 | td(CKXH-DXV) | Delay time, CLKX high to DX valid | [5]−4 | [5]4 | [5]3P + 4 | [5]5P + 17 | ns |
| 6 | tdis(CKXL-DXHZ) | Disable time, DX high impedance following last data bit from CLKX low | [5]L − 2 | [5]L + 3 | | | ns |
| 7 | tdis(FXH-DXHZ) | Disable time, DX high impedance following last data bit from FSX high | | | [5]P + 3 | [5]3P + 17 | ns |
| 8 | td(FXL-DXV) | Delay time, FSX low to DX valid | | | [5]2P + 2 | [5]4P + 17 | ns |

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] 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 duration = (CLKGDV / 2 + 1) × S if CLKGDV is even
= (CLKGDV + 1) / 2 × S if CLKGDV is odd or zero
L = CLKX low pulse duration = (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.
[4] 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
[5] This parameter is not production tested.
[6] 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).

## 7.45 Switching Characteristics for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0[1][2] (see Figure 43)

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] 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 duration = (CLKGDV / 2 + 1) × S if CLKGDV is even
= (CLKGDV + 1) / 2 × S if CLKGDV is odd or zero
L = CLKX low pulse duration = (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.
[4] 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
[5] This parameter is not production tested.
[6] 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).

## 7.46 Switching Characteristics for McBSP as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1[1][2] (see Figure 44)

| NO. | PARAMETER | | MASTER[3] | | SLAVE | | UNIT |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| 1 | $t_{h(CKXH-FXL)}$ | Hold time, FSX low after CLKX high[4] | [5]T − 2 | [5]T + 3 | | | ns |
| 2 | $t_{d(FXL-CKXL)}$ | Delay time, FSX low to CLKX low[6] | [5]H − 2 | [5]H + 3 | | | ns |
| 3 | $t_{d(CKXL-DXV)}$ | Delay time, CLKX low to DX valid | [5]−4 | [5]4 | [5]3P + 4 | [5]5P + 17 | ns |
| 6 | $t_{dis(CKXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from CLKX high | [5]H − 2 | [5]H + 3 | | | ns |
| 7 | $t_{dis(FXH-DXHZ)}$ | Disable time, DX high impedance following last data bit from FSX high | | | [5]P + 3 | [5]3P + 17 | ns |
| 8 | $t_{d(FXL-DXV)}$ | Delay time, FSX low to DX valid | | | [5]2P + 2 | [5]4P + 17 | ns |

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] 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 duration = (CLKGDV / 2 + 1) × S if CLKGDV is even = (CLKGDV + 1) / 2 × S if CLKGDV is odd or zero
L = CLKX low pulse duration = (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.
[4] 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
[5] This parameter is not production tested.
[6] 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).

## 7.47 Switching Characteristics for McBSP as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1[1][2] (see Figure 45)

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

[1] P = 1 / CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
[2] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[3] 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 duration = (CLKGDV / 2 + 1) × S if CLKGDV is even
= (CLKGDV + 1) / 2 × S if CLKGDV is odd or zero
L = CLKX low pulse duration = (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.
[4] 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
[5] This parameter is not production tested.
[6] 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).

## 7.48 Switching Characteristics for DMAC Outputs

over recommended operating conditions for DMAC outputs[1] (see Figure 46)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(DMACH)}$ | Pulse duration, DMAC high | [2]2P − 3 | | ns |

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

## 7.49 Switching Characteristics for Timer Outputs

over recommended operating conditions for timer outputs[1] (see Figure 47)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 3 | $t_{w(TOUTH)}$ | Pulse duration, TOUT high | [2]2P − 3 | | ns |
| 4 | $t_{w(TOUTL)}$ | Pulse duration, TOUT low | [2]2P − 3 | | ns |

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

## 7.50 Switching Characteristics for Power-Down Outputs

over recommended operating conditions for power-down outputs[1] (see Figure 48)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 1 | $t_{w(PDH)}$ | Pulse duration, PD high | [2]2P−3 | | ns |

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

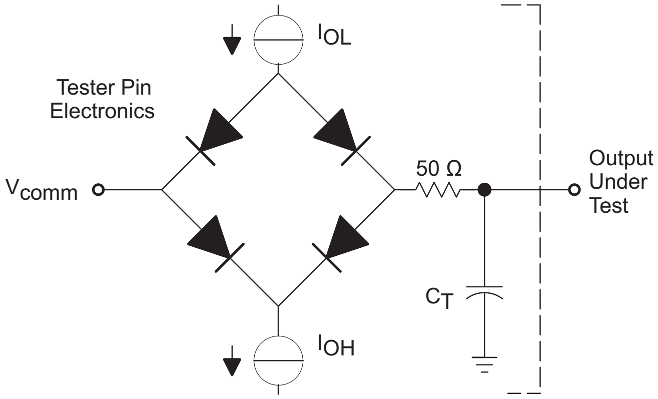
## 7.51 Switching Characteristics for JTAG Test Port

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

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| 2 | $t_{d(TCKL-TDOV)}$ | Delay time, TCK low to TDO valid | [1]−4.5 | [1]13.5 | ns |

[1] This parameter is not production tested.

# 8 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 1. Test Load Circuit for AC Timing Measurements**

## 8.1 Signal Transition Levels

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

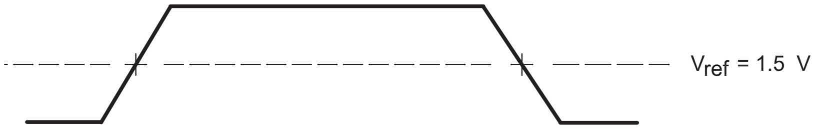


**Figure 2. Input and Output Voltage Reference Levels for AC Timing Measurements**

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.

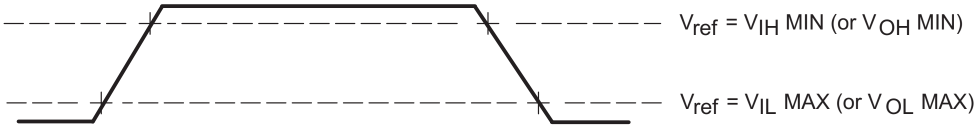

$V_{ref}$ = $V_{IH}$ MIN (or $V_{OH}$ MIN)
$V_{ref}$ = $V_{IL}$ MAX (or $V_{OL}$ MAX)

**Figure 3. Rise and Fall Transition Time Voltage Reference Levels**

## 8.2 Timing Parameters and Board Routing Analysis

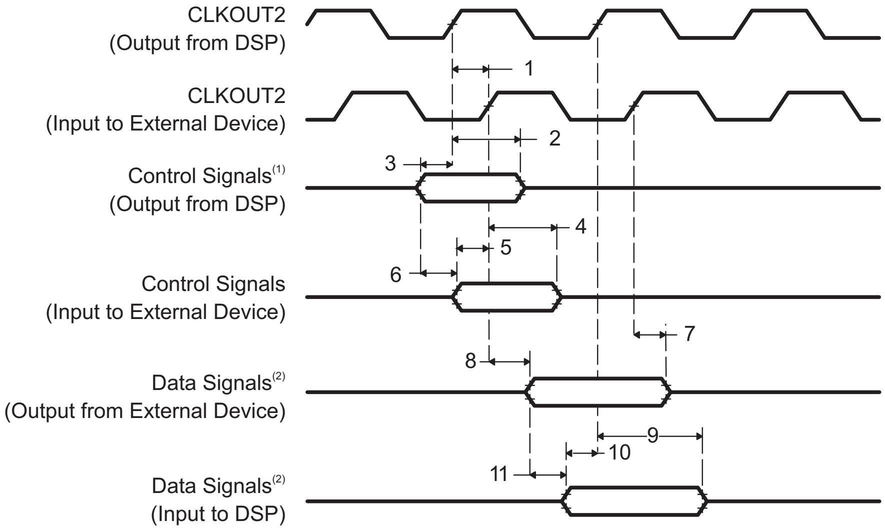
The timing parameter values specified in this data sheet do not include delays by board routings. As a good board design practice, always account for such delays. Timing values may be adjusted by increasing/decreasing such delays. TI recommends using the available I/O buffer information specification (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 1 and Figure 4).

Figure 4 represents a general transfer between the DSP and an external device. Figure 4 also represents board route delays and how they are perceived by the DSP and the external device.

**Table 1. IBIS Timing Parameters Example (See Figure 4)**

| 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 |



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

**Figure 4. IBIS Input/Output Timings**

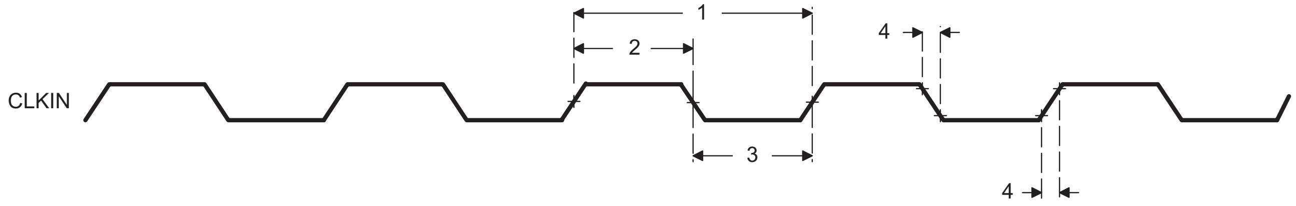


**Figure 5. CLKIN Timings**

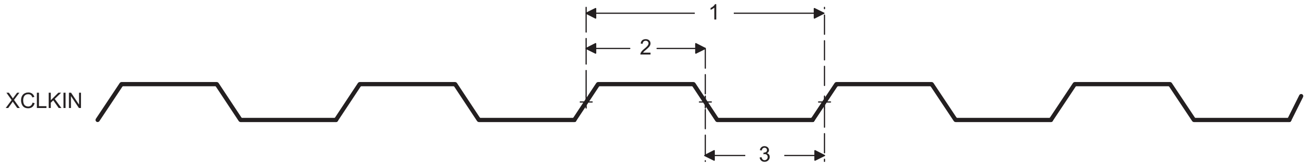


**Figure 6. XCLKIN Timings**

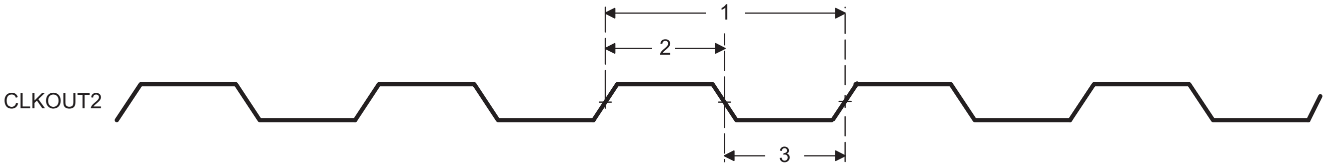


**Figure 7. CLKOUT2 Timings**

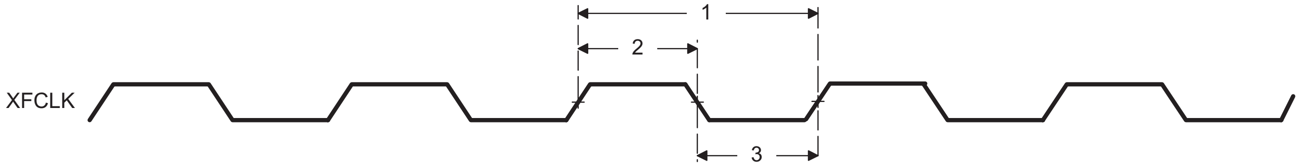
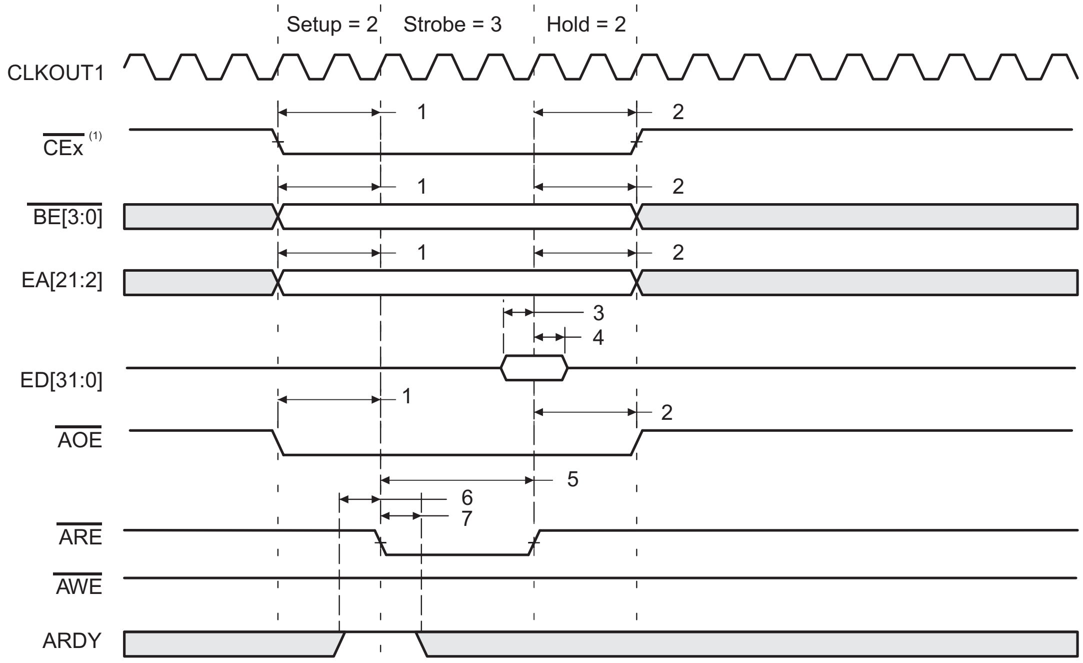
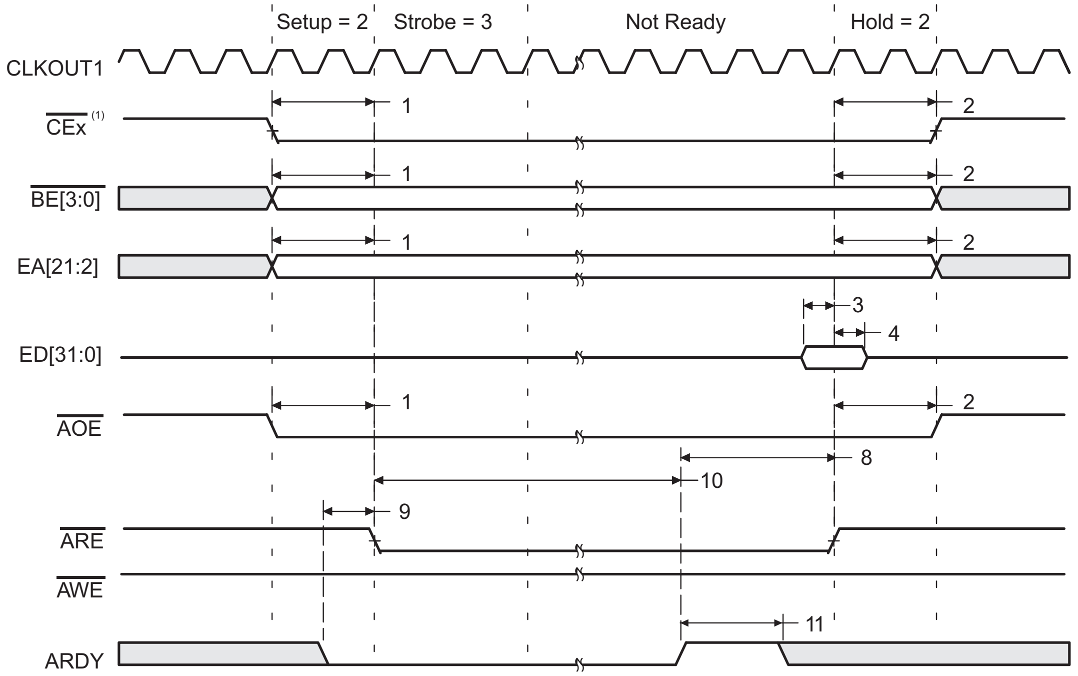


**Figure 8. XFCLK Timings**



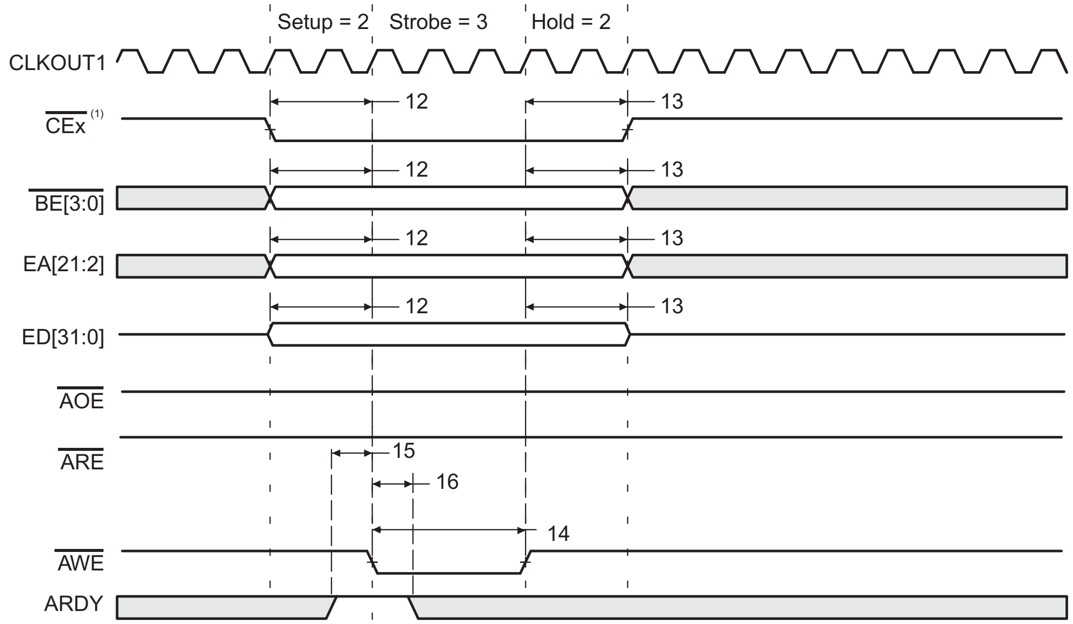
1. CEx stays active for 7 – the value of Read Hold cycles after the last access (DMA transfer or CPU access). For example, if read HOLD = 1, then CEx stays active for six more cycles. This does not affect performance, it merely reflects the overhead of the EMIF.

**Figure 9. Asynchronous Memory Read Timing (ARDY Not Used)**



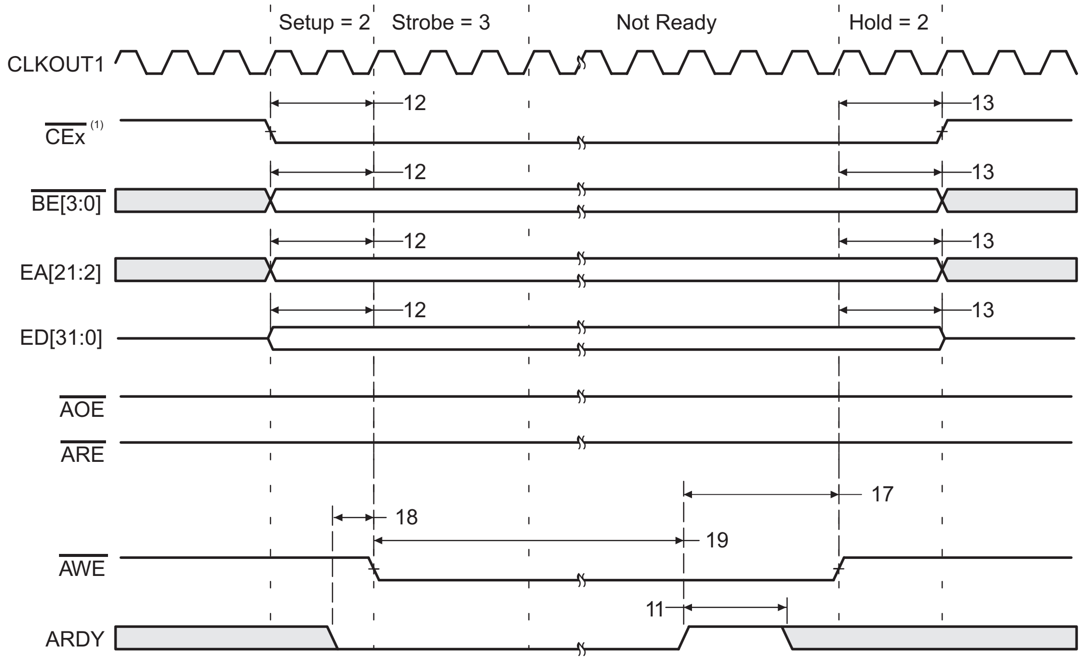
1. CEx stays active for 7 – the value of Read Hold cycles after the last access (DMA transfer or CPU access). For example, if read HOLD = 1, then CEx stays active for six more cycles. This does not affect performance, it merely reflects the overhead of the EMIF.

**Figure 10. Asynchronous Memory Read Timing (ARDY Used)**



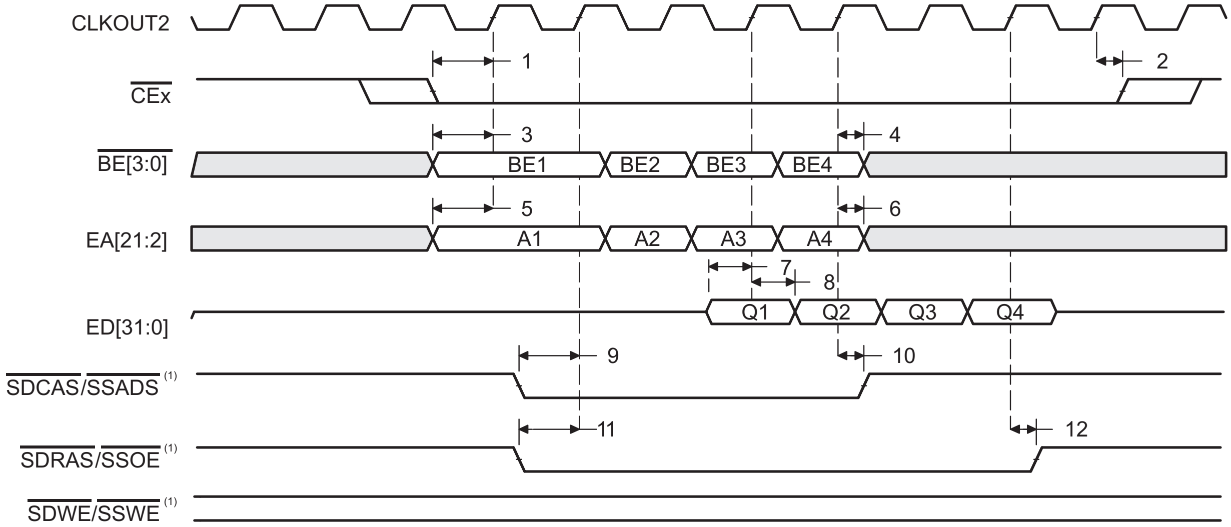
1. If no write accesses are scheduled for the next cycle and write hold is set to 1 or greater, then CEx stays active for three cycles after the value of the programmed hold period. If write hold is set to 0, then CEx stays active for four more cycles. This does not affect performance, it merely reflects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

**Figure 11. Asynchronous Memory Write Timing (ARDY Not Used)**



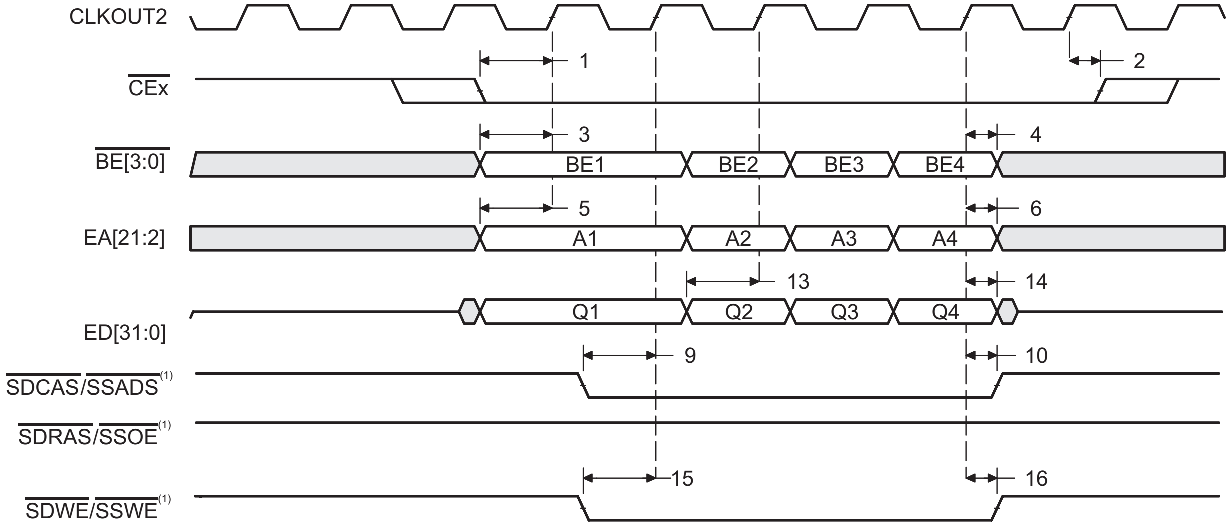
1. If no write accesses are scheduled for the next cycle and write hold is set to 1 or greater, then CEx stays active for three cycles after the value of the programmed hold period. If write hold is set to 0, then CEx stays active for four more cycles. This does not affect performance, it merely reflects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

**Figure 12. Asynchronous Memory Write Timing (ARDY Used)**



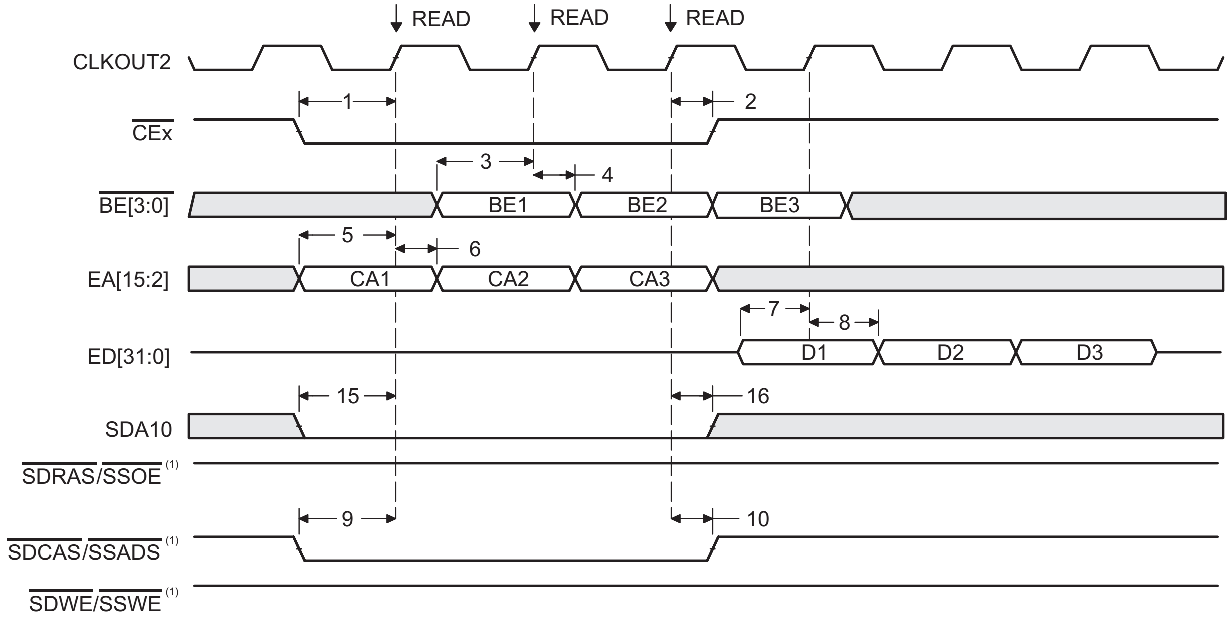
1. $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

**Figure 13. SBSRAM Read Timing**



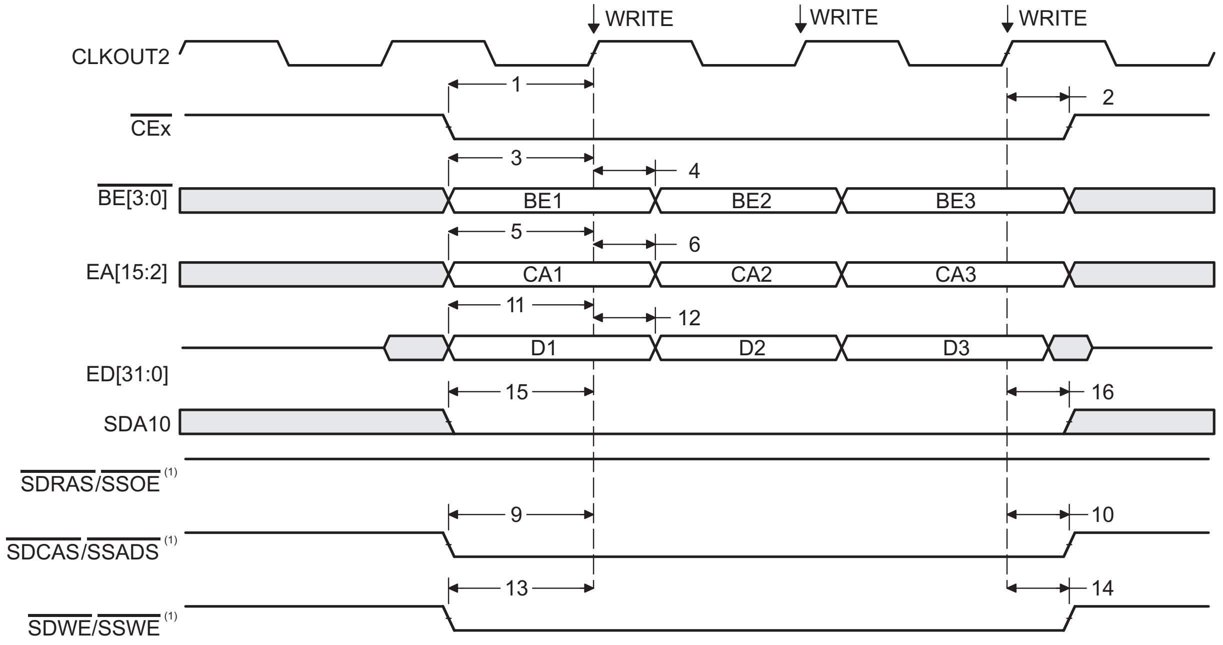
1. $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

**Figure 14. SBSRAM Write Timing**



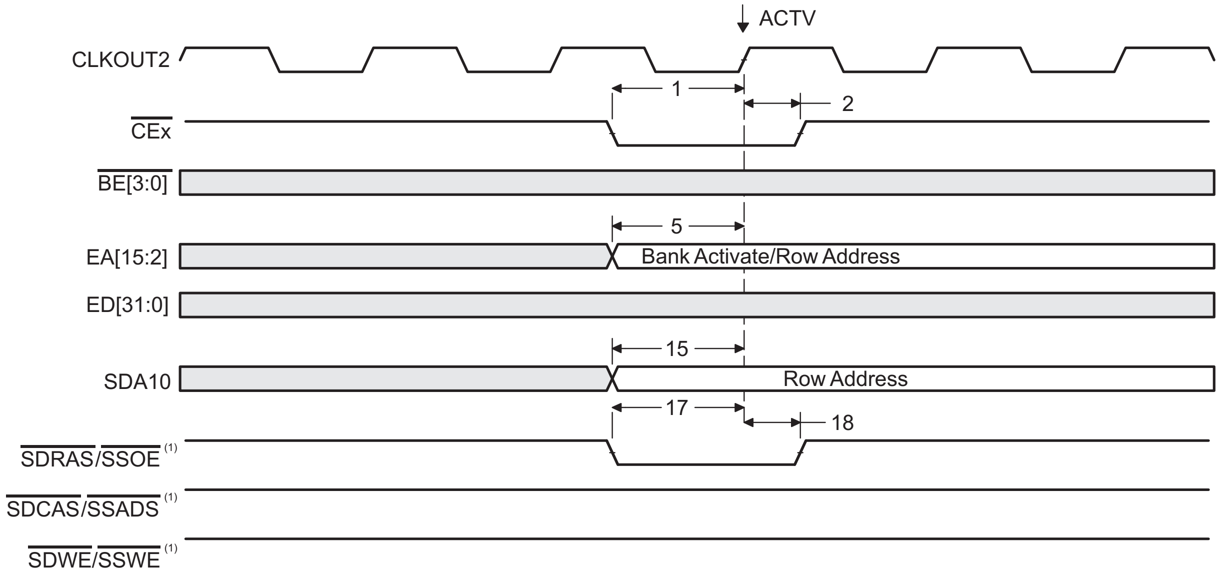
1. $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

**Figure 15. Three SDRAM READ Commands**



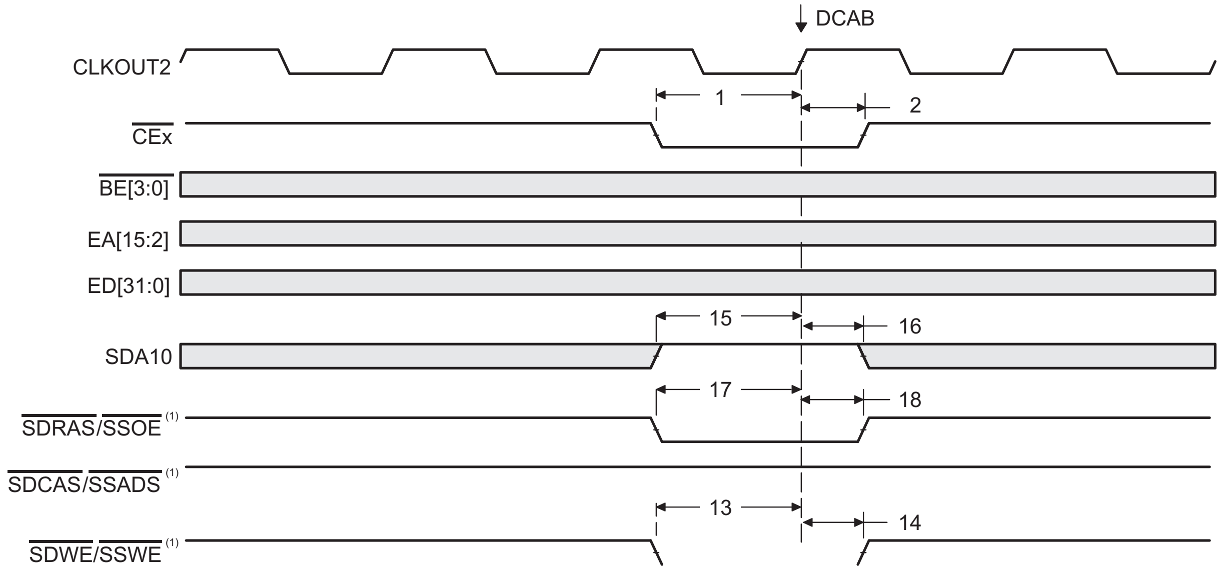
1. $\overline{\text{SDCAS}}/\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}/\overline{\text{SSOE}}$, and $\overline{\text{SDWE}}/\overline{\text{SSWE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$, and $\overline{\text{SDWE}}$, respectively, during SDRAM accesses.

**Figure 16. Three SDRAM WRT Commands**



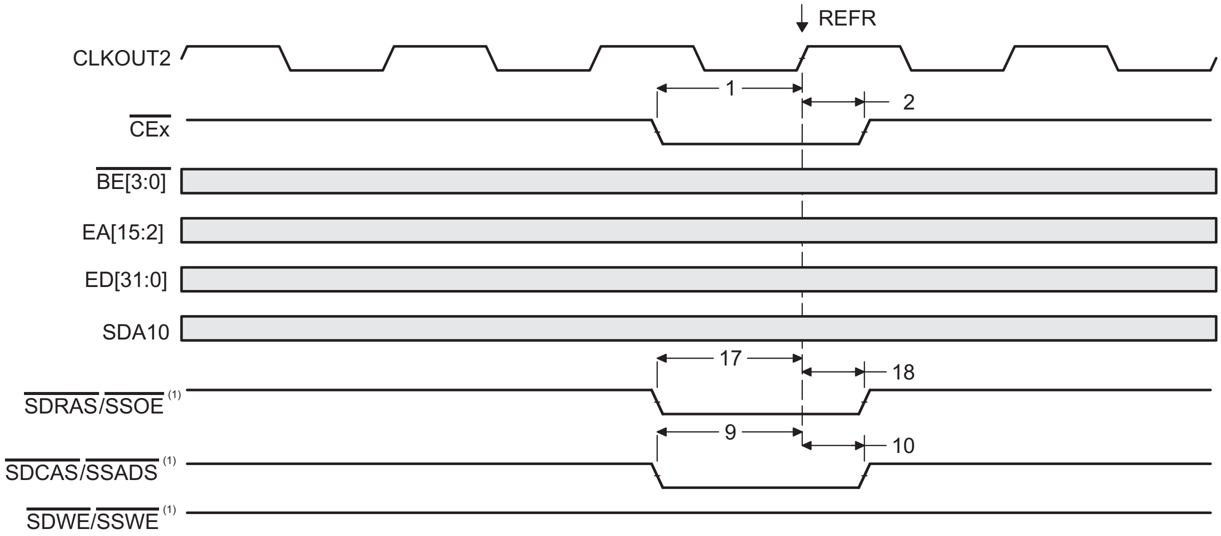
1. SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 17. SDRAM ACTV Command**



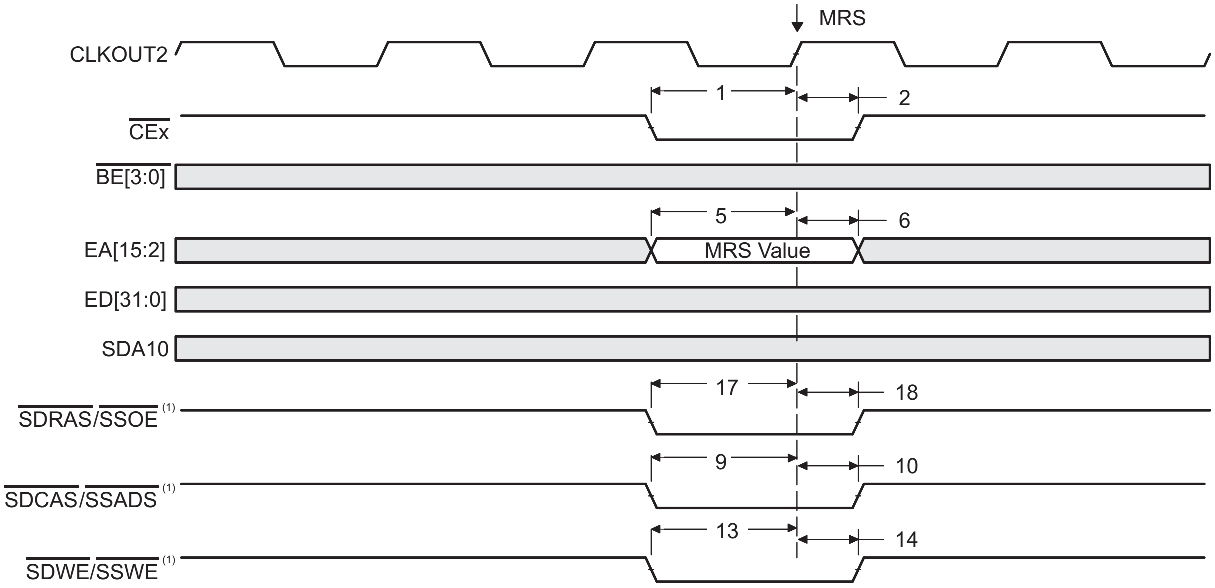
1. SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 18. SDRAM DCAB Command**



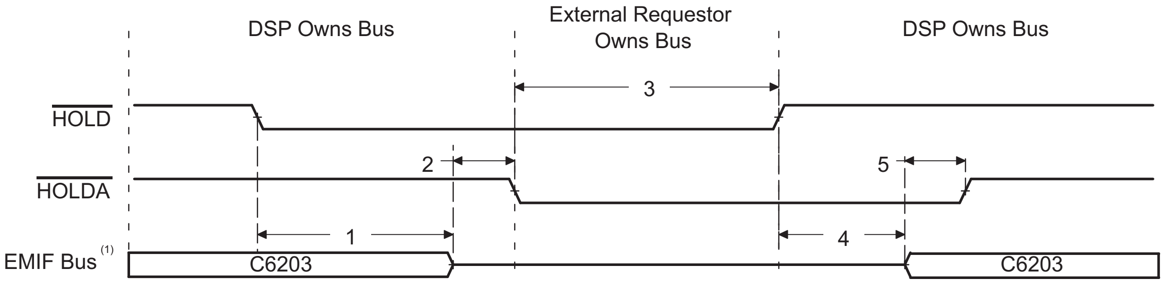
1. SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 19. SDRAM REFR Command**



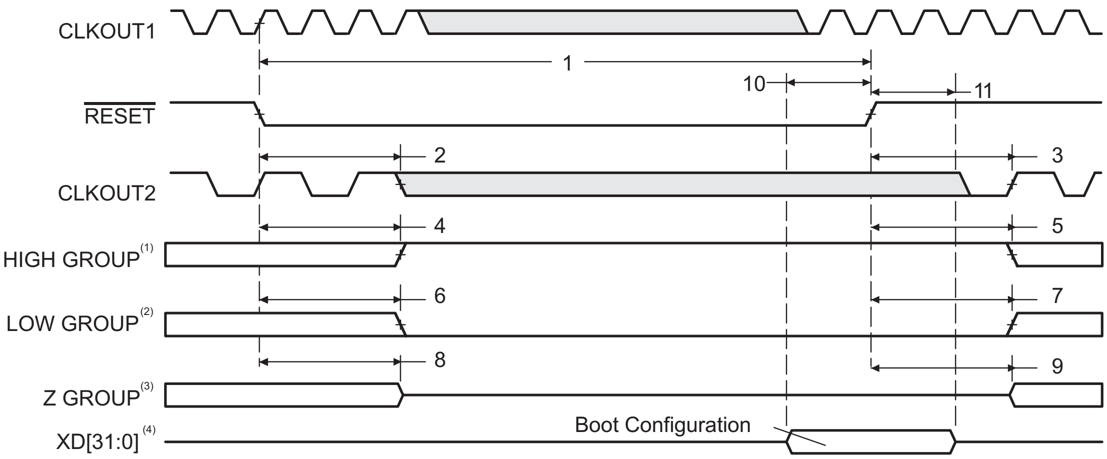
1. SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

**Figure 20. SDRAM MRS Command**



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 21. HOLD/HOLDA Timing**



1. High group consists of: XFCLK, HOLDA
2. Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
3. 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
4. XD[31:0] are the boot configuration pins during device reset.

**Figure 22. Reset Timing**

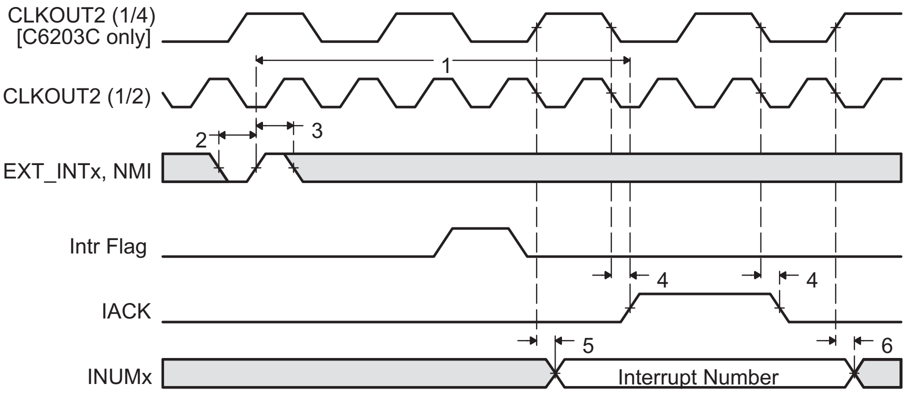
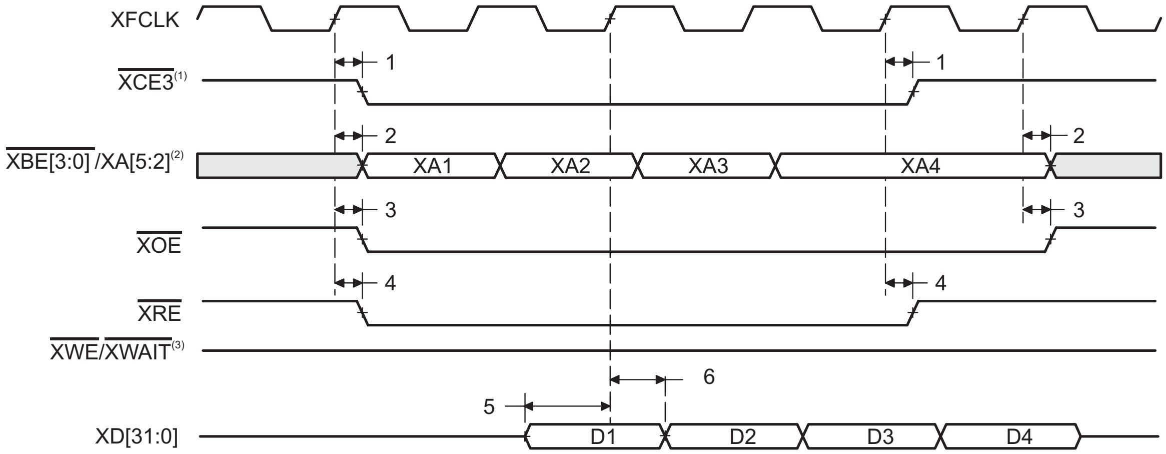
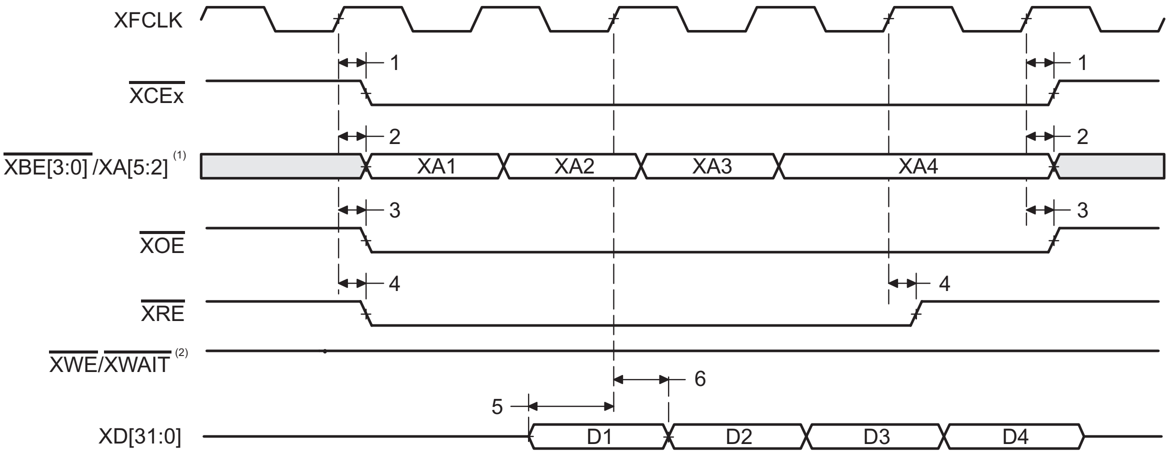


**Figure 23. Interrupt Timing**



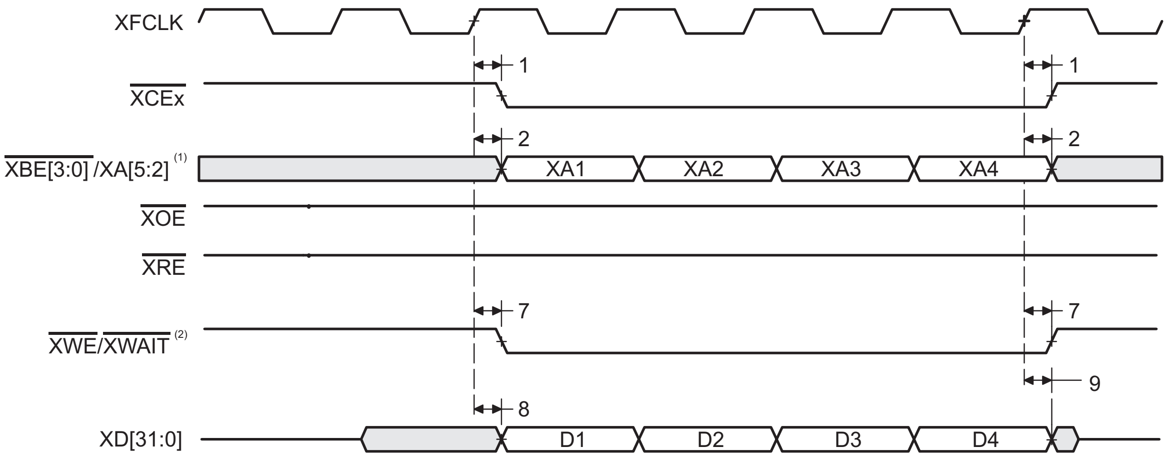
1. FIFO read (glueless) mode only available in XCE3.
2. XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
3. XWE/XWAIT operate as the write-enable signal XWE during synchronous FIFO accesses.

**Figure 24. FIFO Read Timing (Glueless Read Mode)**



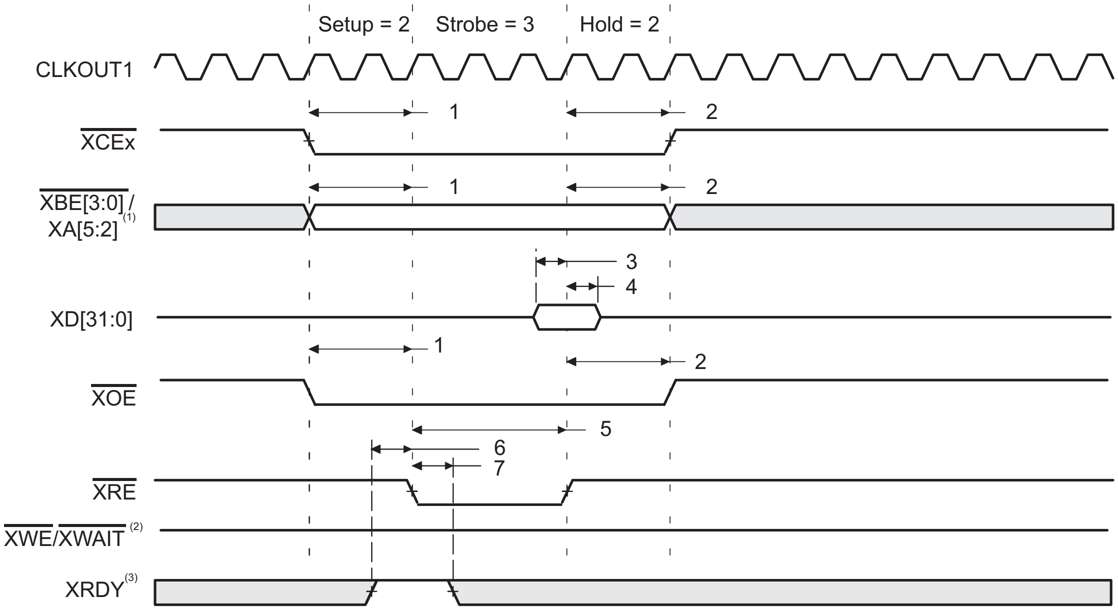
1. XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
2. XWE/XWAIT operate as the write-enable signal XWE during synchronous FIFO accesses.

**Figure 25. FIFO Read Timing**



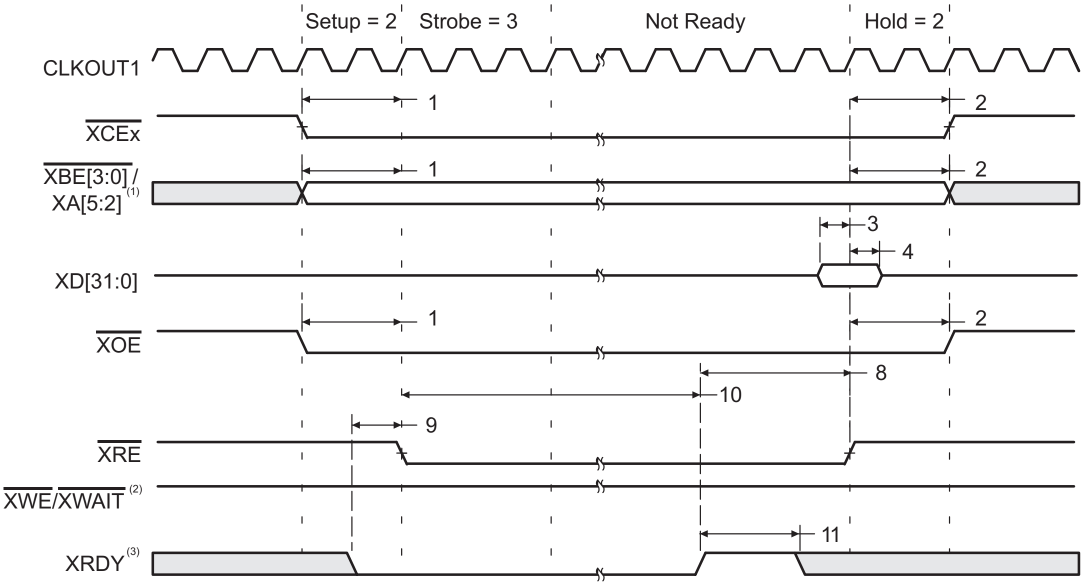
1. XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during synchronous FIFO accesses.
2. XWE/XWAIT operate as the write-enable signal XWE during synchronous FIFO accesses.

**Figure 26. FIFO Write Timing**



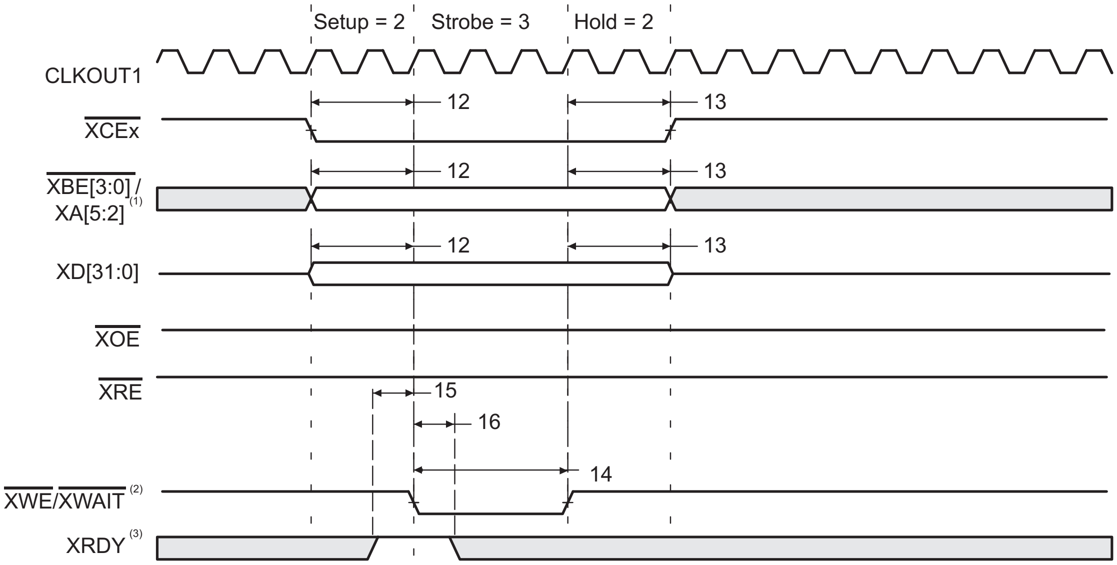
1. XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
2. XWE/XWAIT operate as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
3. XRDY operates as active-high ready input during expansion bus asynchronous peripheral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

**Figure 27. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Not Used)**



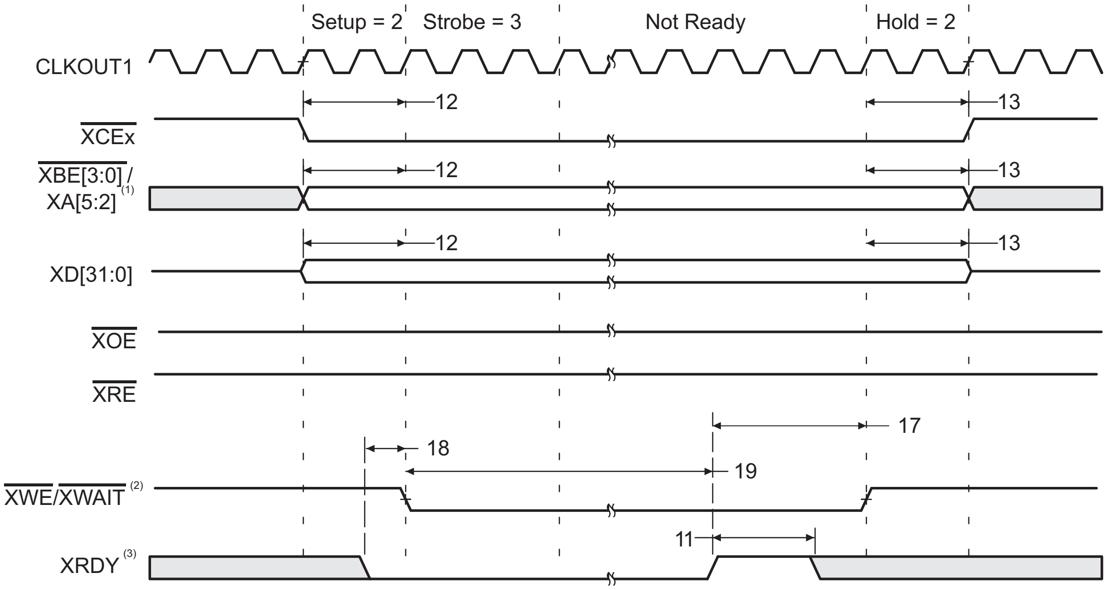
1. $\overline{XBE[3:0]}$/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
2. $\overline{XWE}$/$\overline{XWAIT}$ operate as the write-enable signal $\overline{XWE}$ during expansion bus asynchronous peripheral accesses.
3. XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

**Figure 28. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Used)**



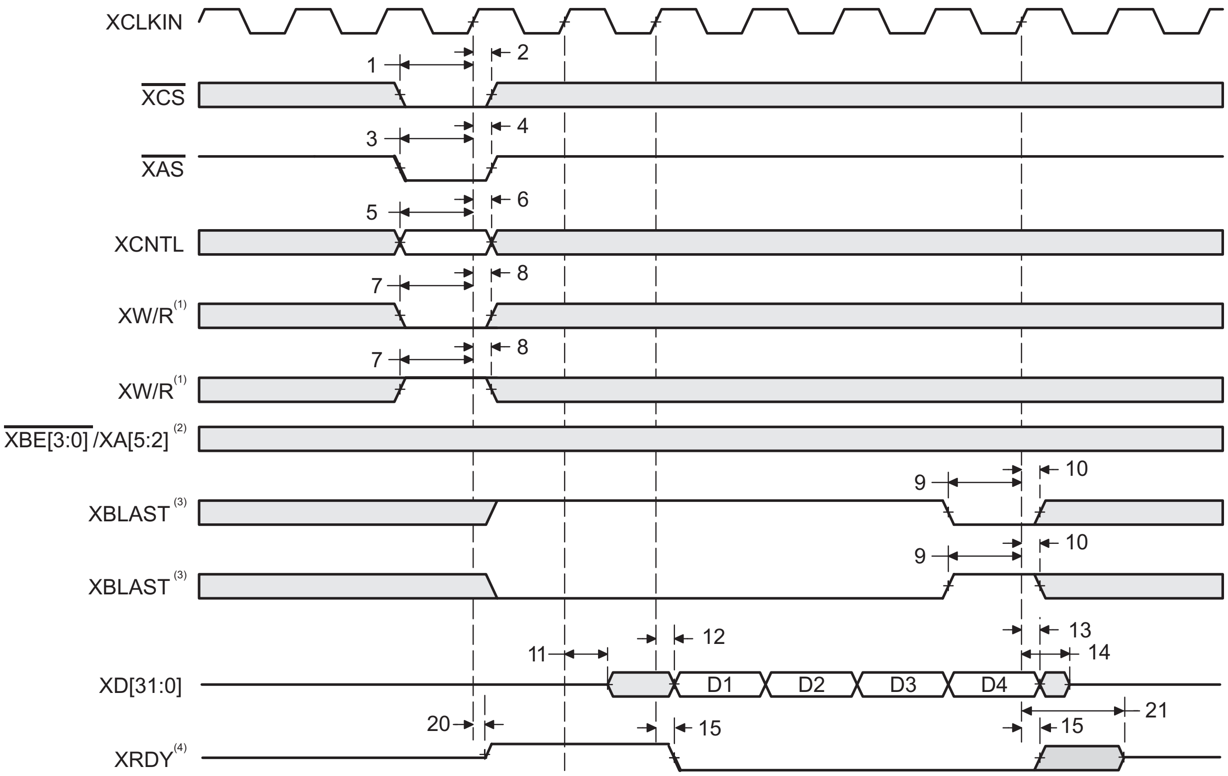
1. $\overline{XBE[3:0]}$/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
2. $\overline{XWE}$/$\overline{XWAIT}$ operate as the write-enable signal $\overline{XWE}$ during expansion bus asynchronous peripheral accesses.
3. XRDY operates as active-high ready input during expansion bus asynchronous peripheral accesses.

**Figure 29. Expansion Bus Asynchronous Peripheral Write Timing (XRDY Not Used)**



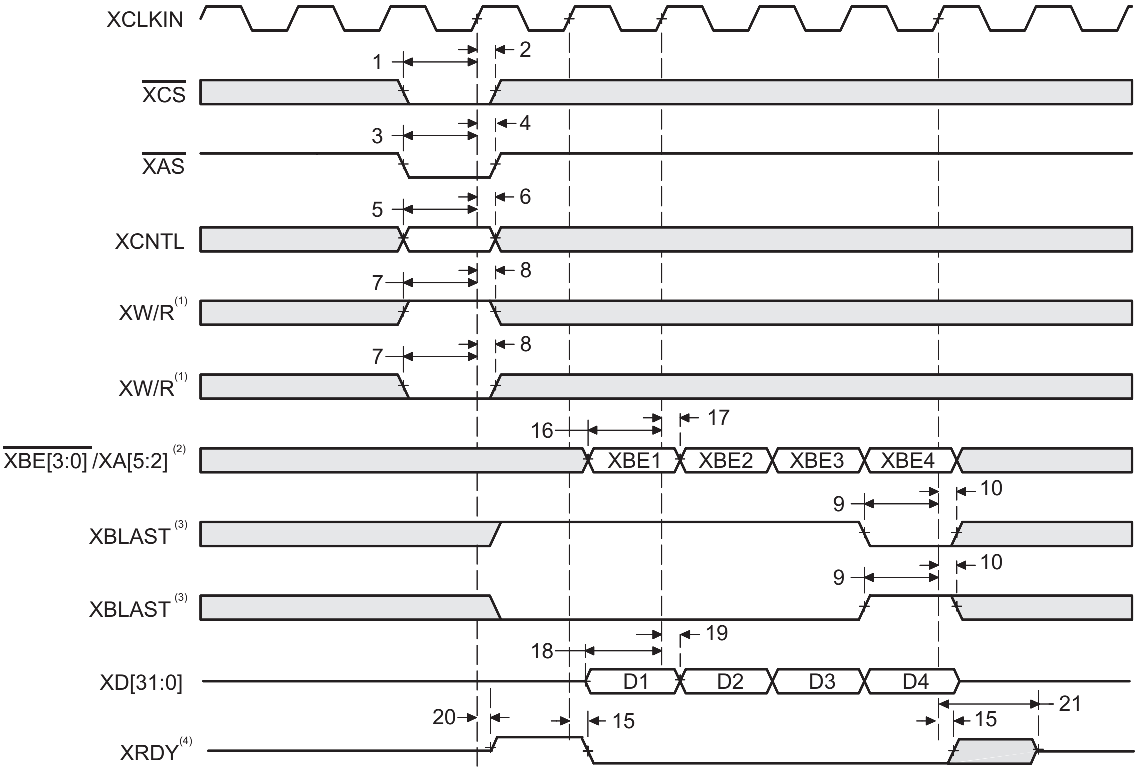
1. XBE[3:0]/XA[5:2] operate as address signals XA[5:2] during expansion bus asynchronous peripheral accesses.
2. XWE/XWAIT operate as the write-enable signal XWE during expansion bus asynchronous peripheral accesses.
3. XRDY operates as active-high ready input during expansion bus asynchronous peripheral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

**Figure 30. Expansion Bus Asynchronous Peripheral Write Timing (XRDY Used)**



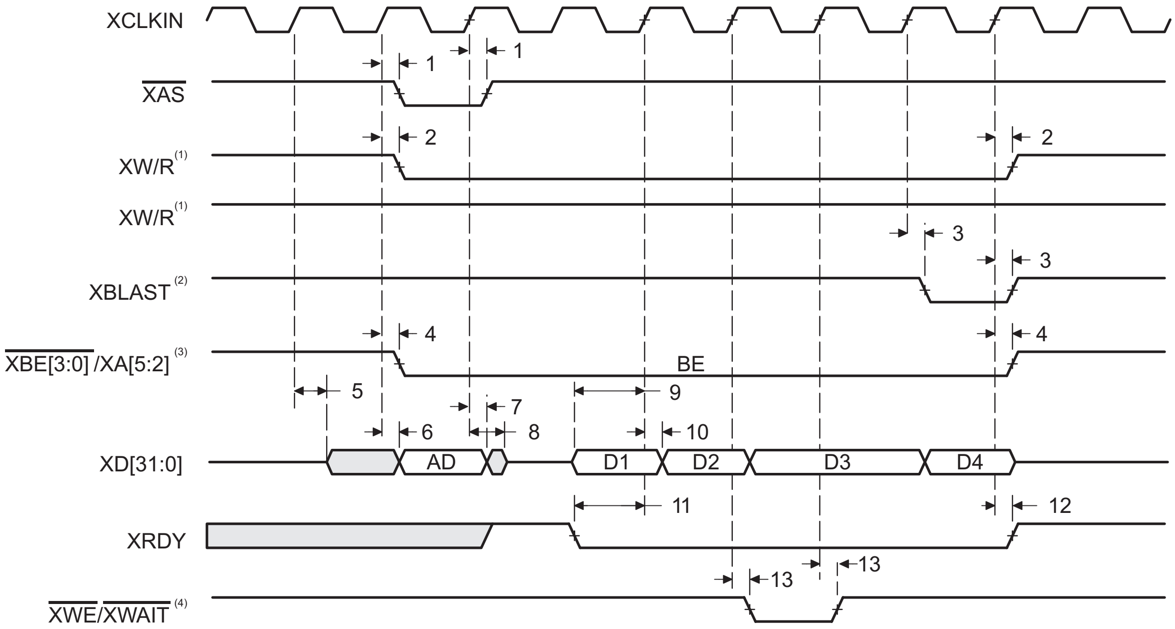
1. XW/R input/output polarity selected at boot
2. XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
3. XBLAST input polarity selected at boot
4. XRDY operates as active-low ready input/output during host-port accesses.

**Figure 31. External Host as Bus Master—Read**



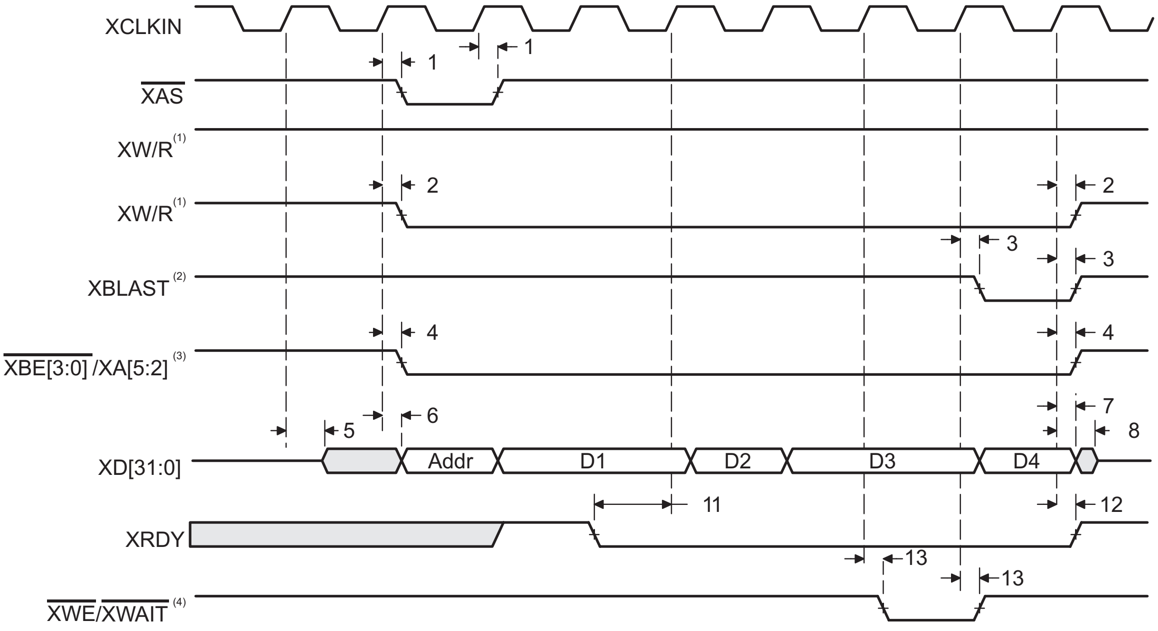
1. XW/R input/output polarity selected at boot
2. $\overline{\text{XBE[3:0]}}$/XA[5:2] operate as byte-enables $\overline{\text{XBE[3:0]}}$ during host-port accesses.
3. XBLAST input polarity selected at boot
4. XRDY operates as active-low ready input/output during host-port accesses.

**Figure 32. External Host as Bus Master—Write**



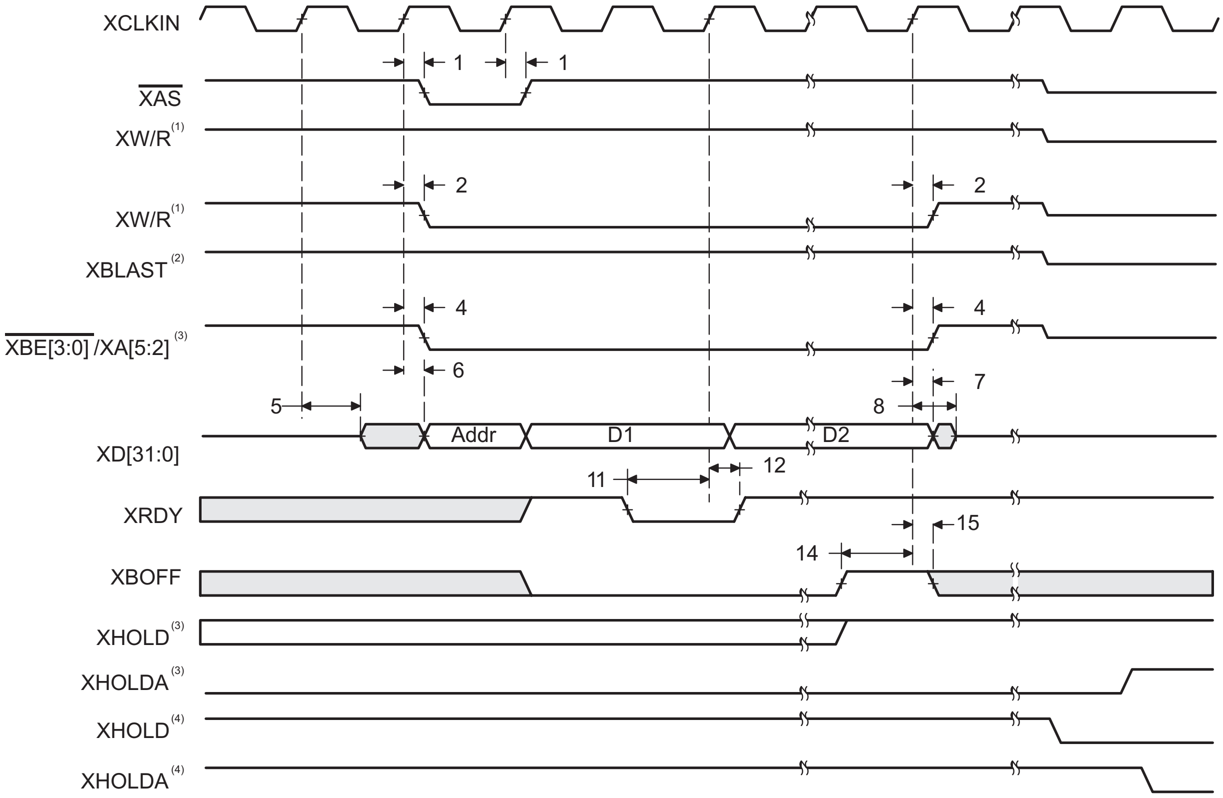
1. XW/R input/output polarity selected at boot
2. XBLAST output polarity is always active low.
3. XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
4. XWE/XWAIT operate as XWAIT output signal during host-port accesses.

**Figure 33. C62x as Bus Master—Read**



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

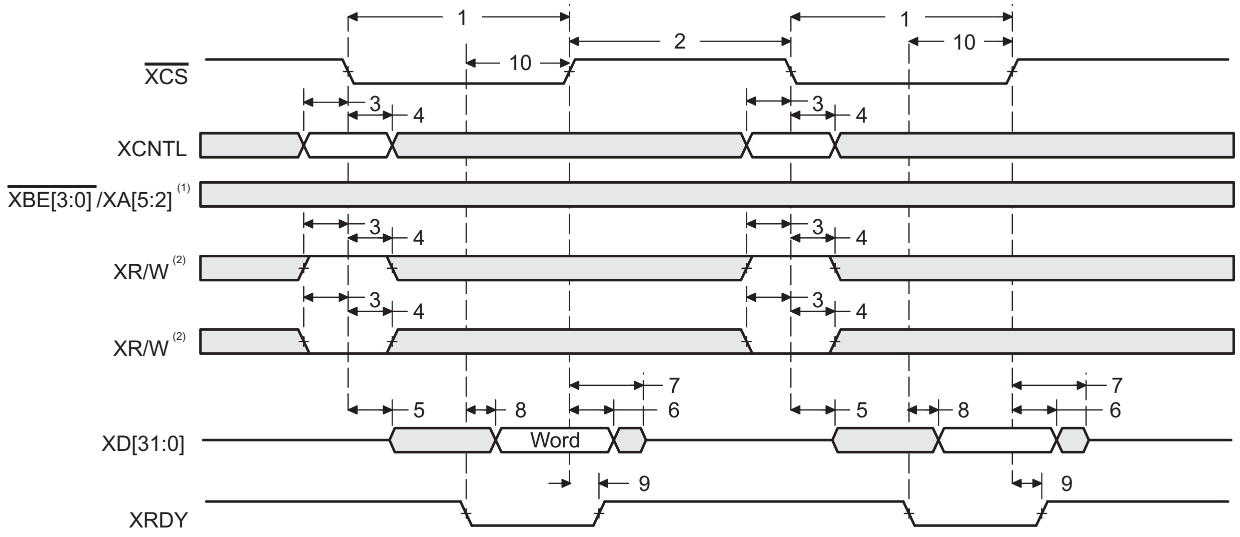
**Figure 34. C62x as Bus Master—Write**



1. XW/R input/output polarity selected at boot
2. XBLAST output polarity is always active low.
3. XBE[3:0]/XA[5:2] operate as byte-enables XBE[3:0] during host-port accesses.
4. Internal arbiter enabled
5. External arbiter enabled

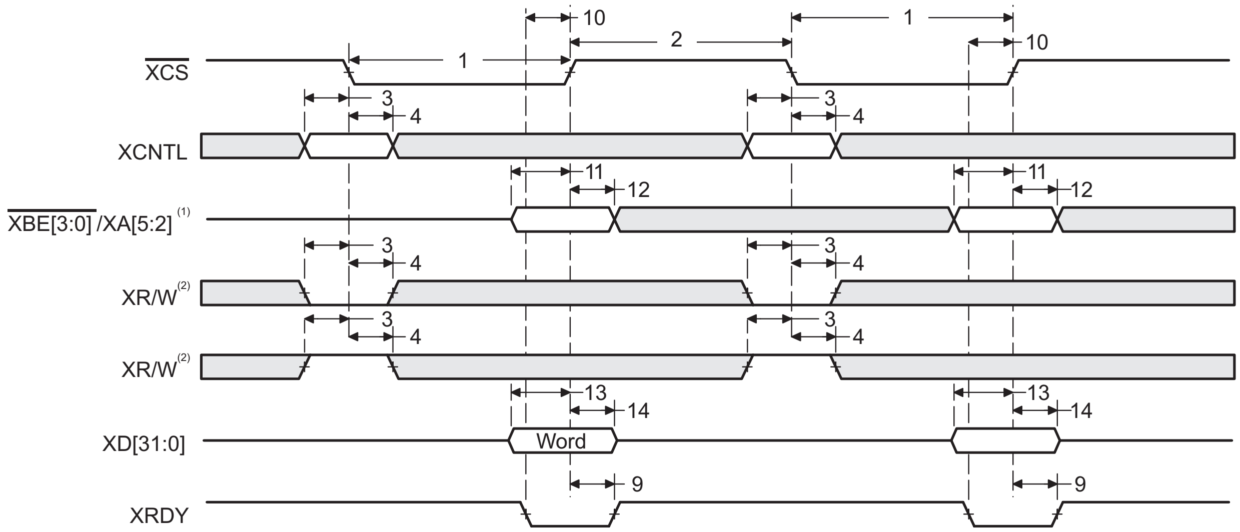
NOTE: This diagram illustrates XBOFF timing. Figure 38 and Figure 39 show bus arbitration timing.

**Figure 35. C62x as Bus Master—BOFF Operation**



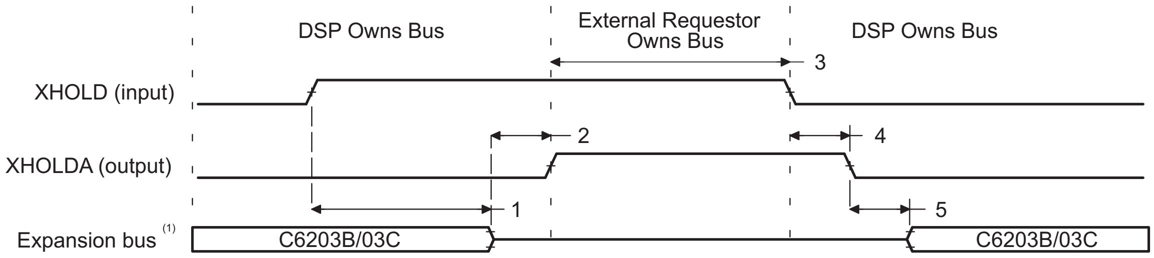
1. $\overline{XBE[3:0]}$/XA[5:2] operate as byte-enables $\overline{XBE[3:0]}$ during host-port accesses.
2. XW/R input/output polarity selected at boot

**Figure 36. External Device as Asynchronous Master—Read**



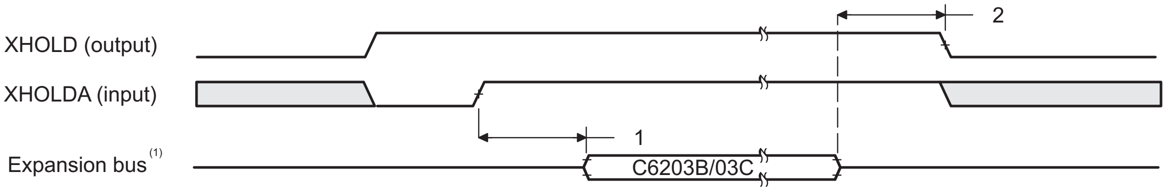
1. $\overline{XBE[3:0]}$/XA[5:2] operate as byte-enables $\overline{XBE[3:0]}$ during host-port accesses.
2. XW/R input/output polarity selected at boot

**Figure 37. External Device as Asynchronous Master—Write**



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

**Figure 38. Expansion Bus Arbitration—Internal Arbiter Enabled**



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

**Figure 39. Expansion Bus Arbitration—Internal Arbiter Disabled**

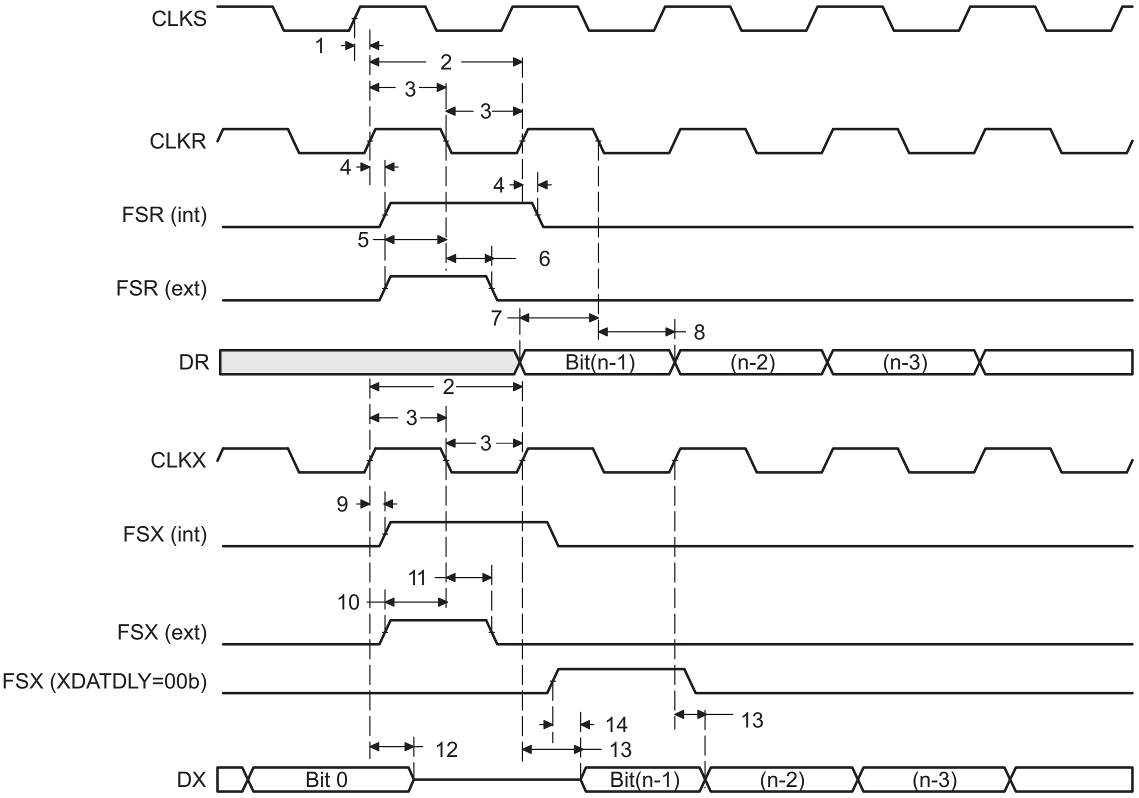


**Figure 40. McBSP Timings**

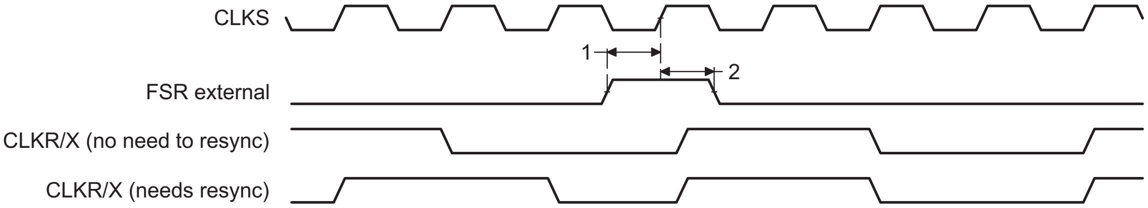


**Figure 41. FSR Timing When GSYNC = 1**

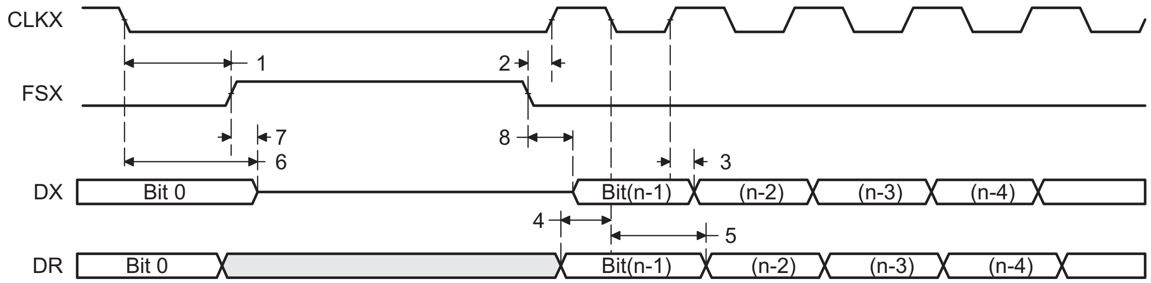


Figure 42. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

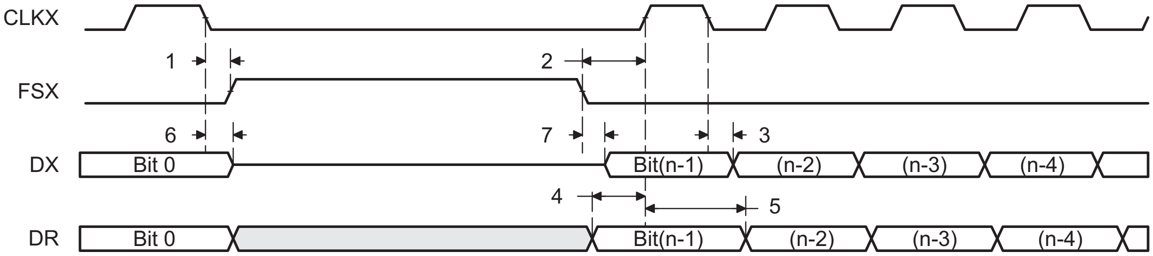


Figure 43. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

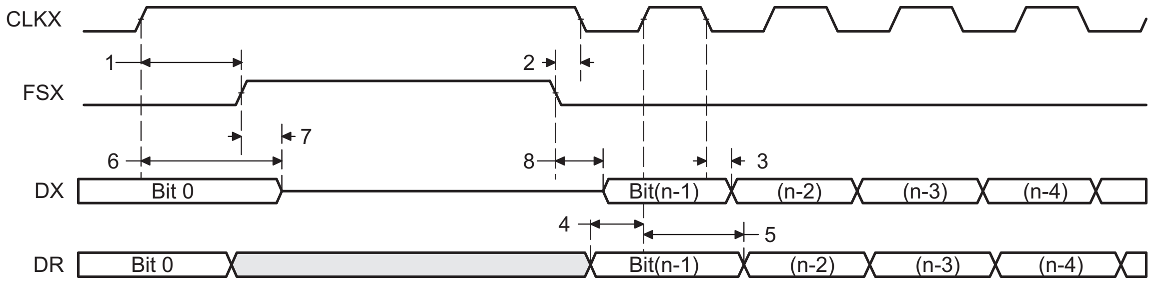


Figure 44. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

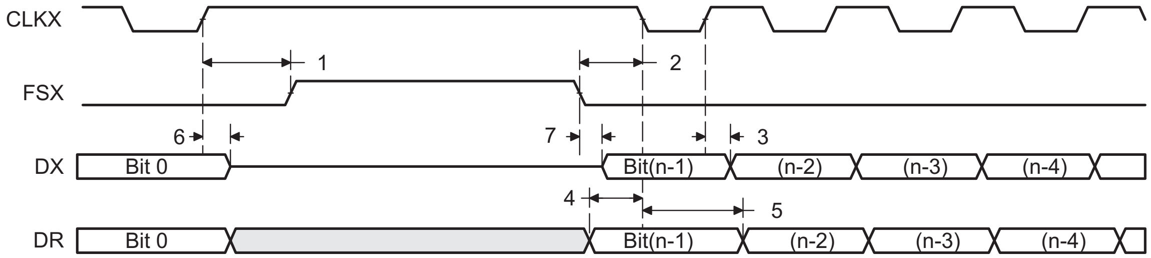


Figure 45. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

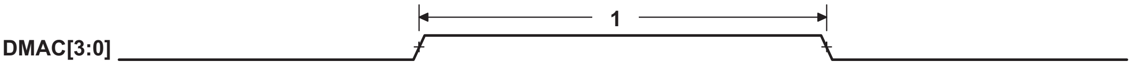


Figure 46. DMAC Timing

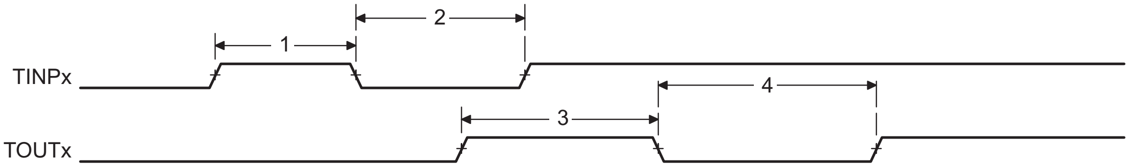


**Figure 47. Timer Timing**

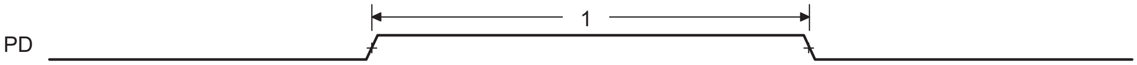


**Figure 48. Power-Down Timing**

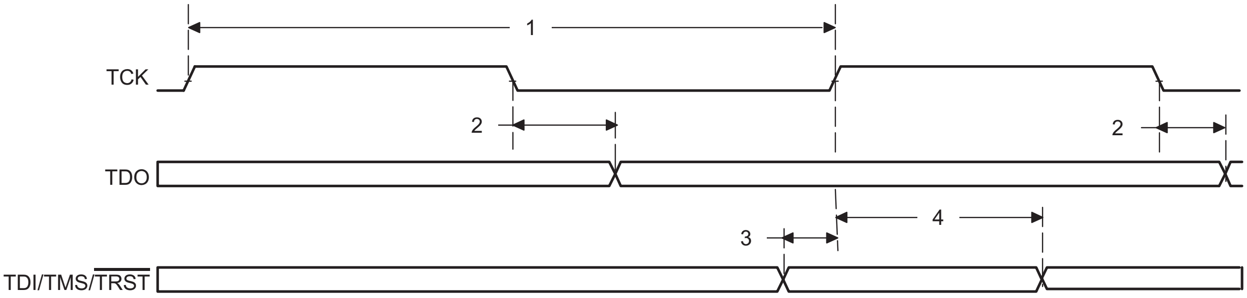
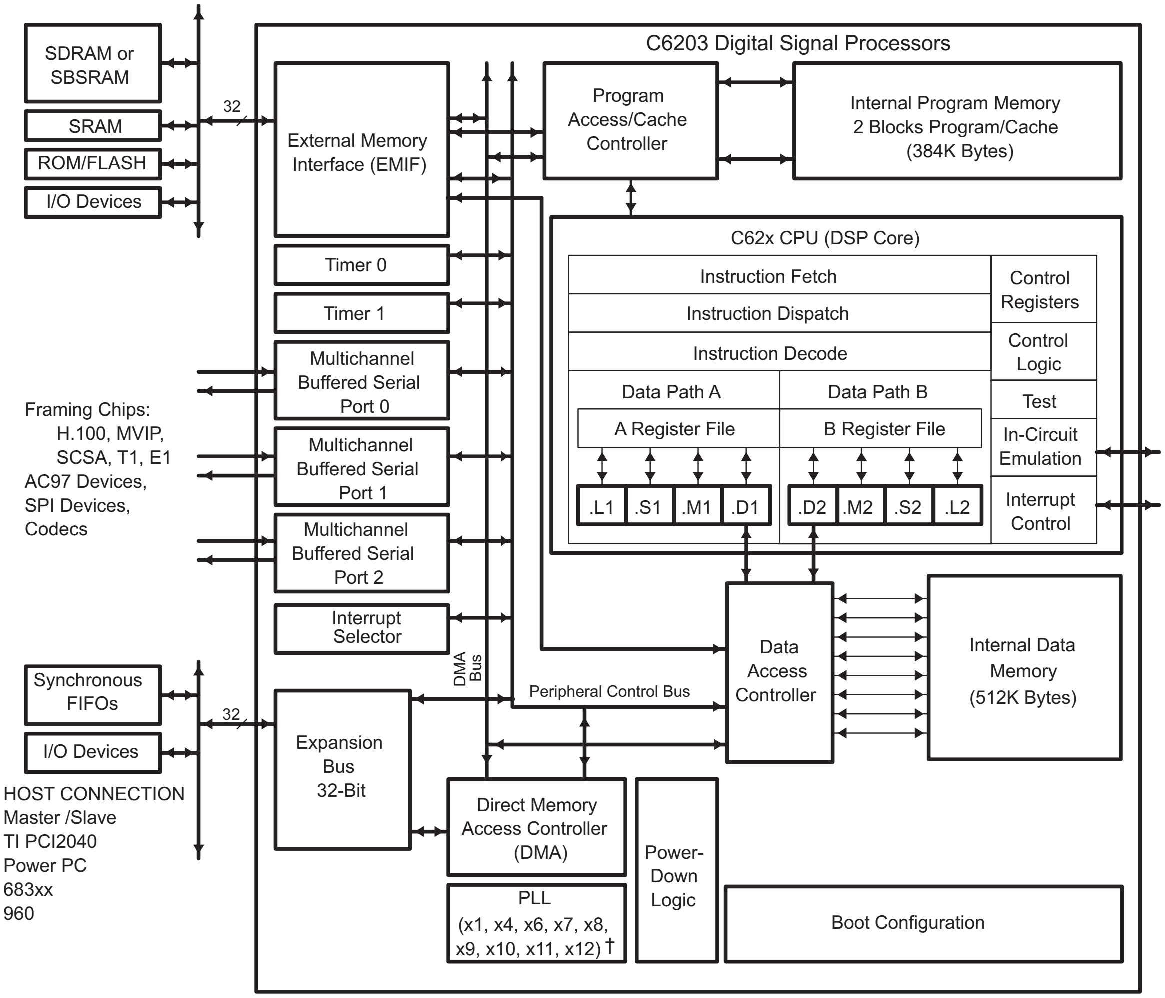


**Figure 49. JTAG Test-Port Timing**

# 9 Detailed Description

## 9.1 Functional Block Diagram



A. For additional details on the PLL clock module and specific options for the C6203 device, see *Characteristics of the C6203 DSP* and *Clock PLL*.

## 9.2 Feature Description

### 9.2.1 Signal Groups Description

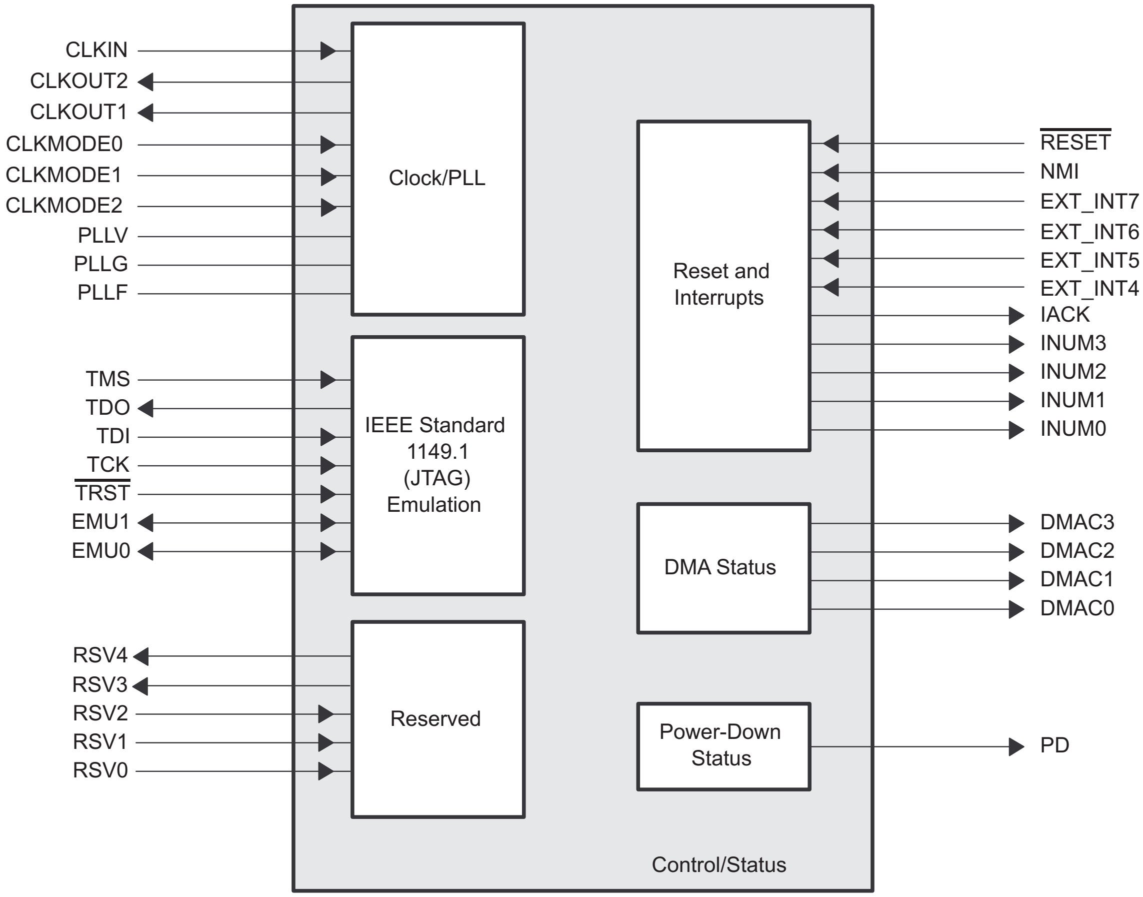


**Figure 50. CPU (DSP Core) Signals**

## Feature Description (continued)

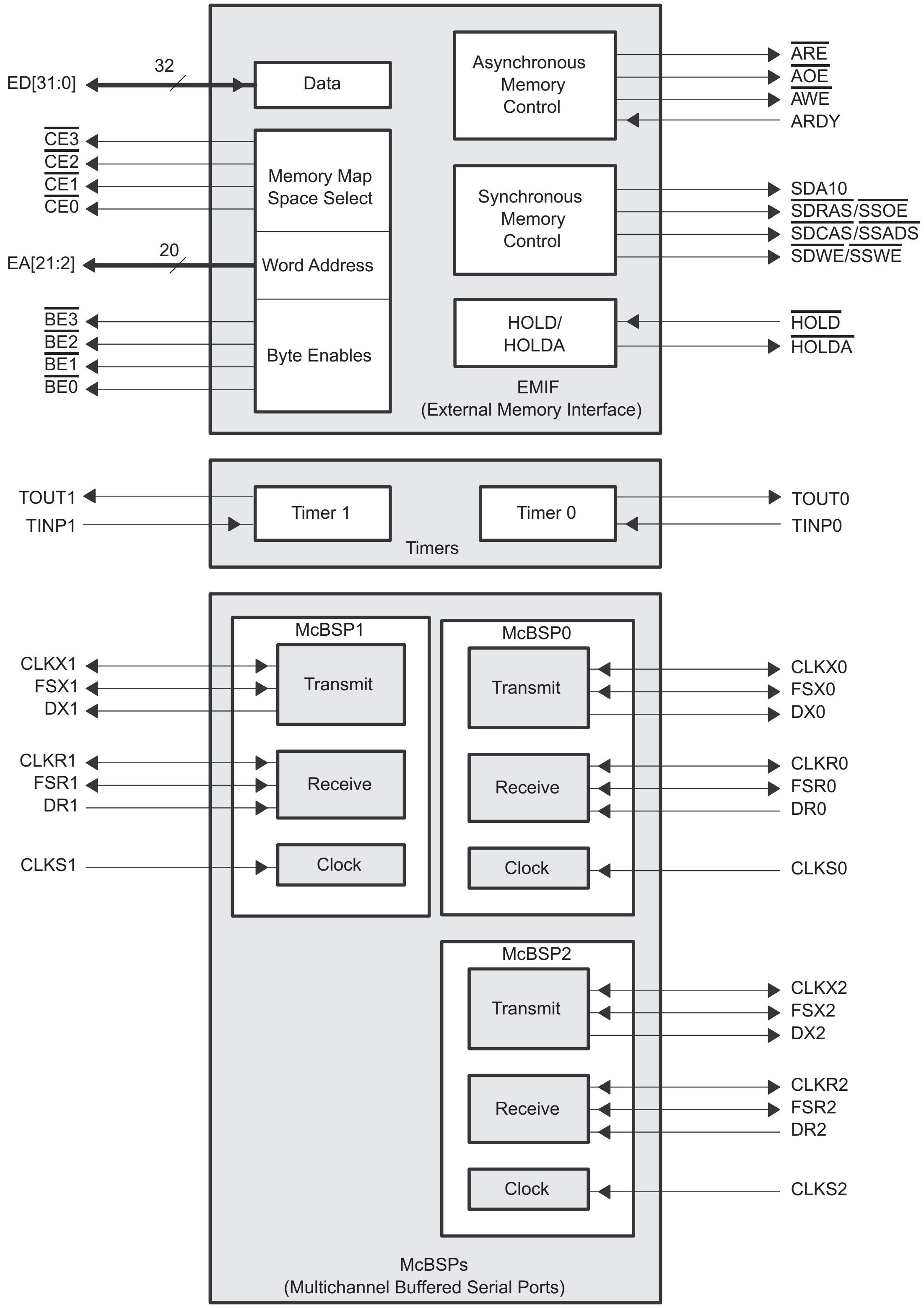


**Figure 51. Peripheral Signals**

## Feature Description (continued)

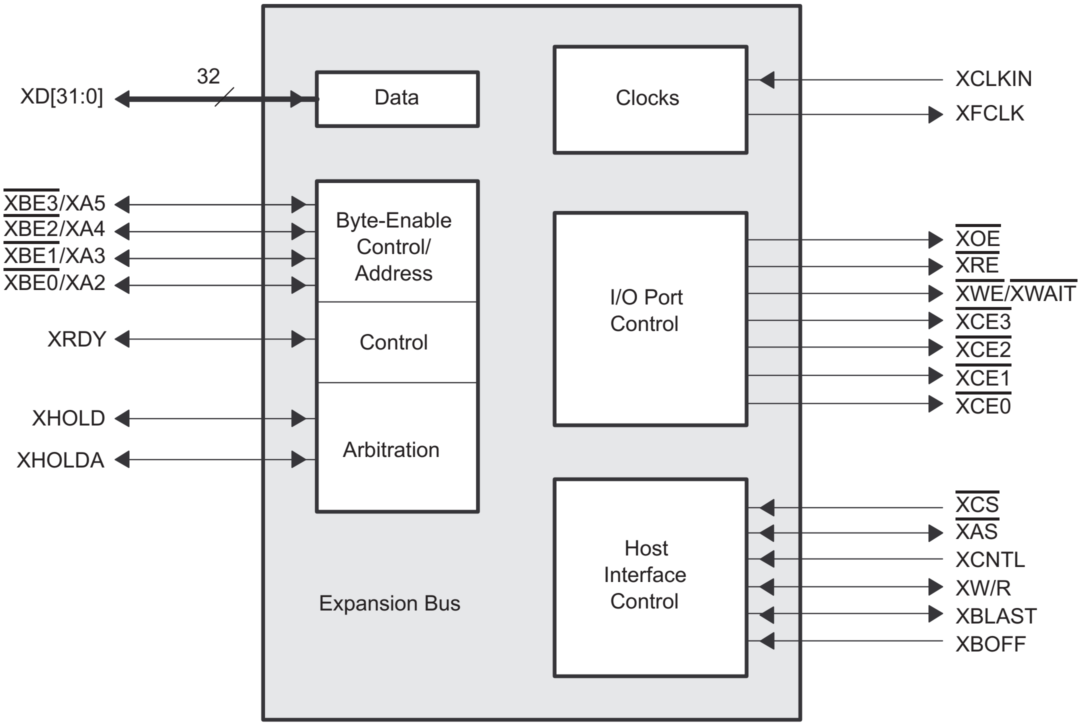


**Figure 52. Peripheral Signals (continued)**

### 9.2.2 CPU (DSP Core) Description

The CPU fetches VelociTI advanced 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 *Functional Block Diagram* and Figure 53). 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. 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 instructions can access any of the 32 registers. However, some registers 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.

## Feature Description (continued)

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.

## Feature Description (continued)

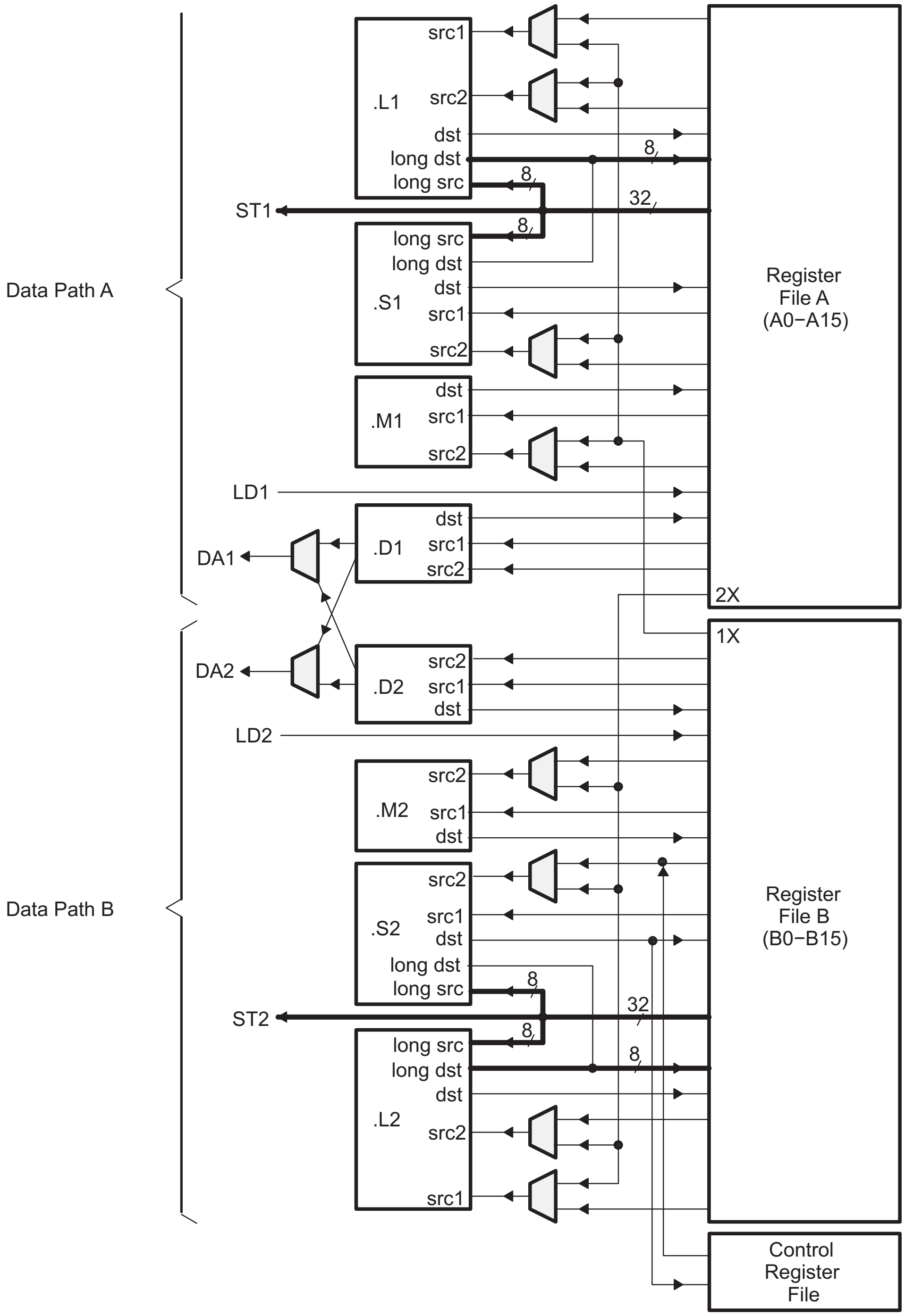


**Figure 53. SMJ320C62x CPU (DSP Core) Data Paths**

## Feature Description (continued)

### 9.2.3 Clock PLL

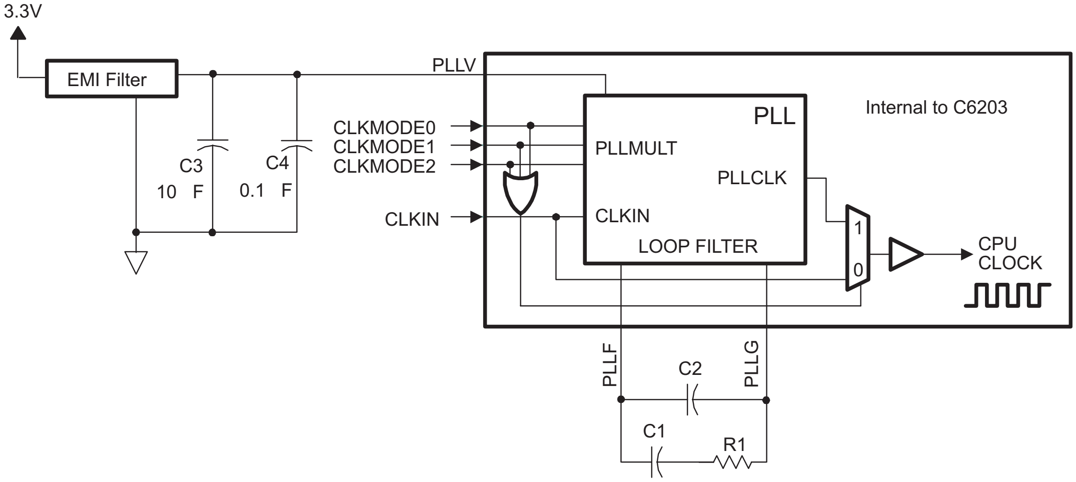
Most of the internal C6203 clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock in frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock.

To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 54, and Table 3 through Table 17 show the external PLL circuitry for either x1 (PLL bypass) or x4 PLL multiply modes. Figure 55 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. Observe the minimum CLKIN rise and fall times. For the input clock timing requirements, see the input and output clocks in *Specifications*. Table 2 lists some examples of compatible CLKIN external clock sources:

**Table 2. 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 |



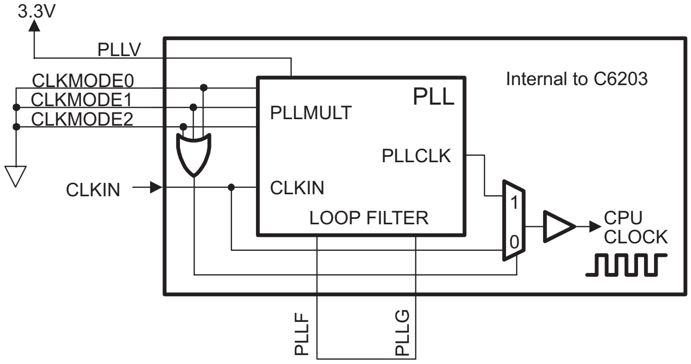
(1) For the PLL options and CLKMODE pins setup, see Table 3 and Table 17.

(2) 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.

(3) For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (R1, C1, C2, C3, C4, and the EMI Filter).

(4) The 3.3-V supply for the EMI filter must be from the same 3.3-V power plane supplying the I/O voltage, DVDD.

**Figure 54. External PLL Circuitry for Either PLL Multiply Modes or x1 (Bypass) Mode**



(1) For a system with **only** PLL x1 (bypass) mode, short the PLLF to PLLG.

(2) The 3.3-V supply for PLLV must be from the same 3.3-V power plane supplying the I/O voltage, DVDD.

**Figure 55. External PLL Circuitry for x1 (Bypass) PLL Mode Only**

**Table 3. PLL Multiply and Bypass (x1) Options[1]**

| 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 |

[1] f(CPU Clock) = f(CLKIN) × (PLL mode)

## 9.3 Register Maps

### 9.3.1 Memory Map Summary

Table 4 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 *TMS320C6000 Peripherals Reference Guide* (SPRU190) for information regarding boot configuration.

**Table 4. 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 |

### 9.3.2 Peripheral Register Descriptions

Table 5 through Table 14 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* (SPRU190).

### Table 5. 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 | |

### Table 6. 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 |

### Table 7. 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 8. Interrupt Selector Registers

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 019C 0000 | MUXH | Interrupt multiplexer high | Selects which interrupts drive CPU interrupts 10 to 15 (INT10 to INT15) |
| 019C 0004 | MUXL | Interrupt multiplexer low | Selects which interrupts drive CPU interrupts 4 to 9 (INT04 to 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 9. Peripheral Power-Down Control Register

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

### Table 10. 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 | |

### Table 10. McBSP 0 Registers (continued)

| HEX ADDRESS RANGE | ACRONYM | REGISTER NAME | COMMENTS |
|---|---|---|---|
| 018C 0028 − 018F FFFF | – | Reserved | |

### Table 11. 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 | |

### Table 12. 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 13. Timer 0 Registers

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

### Table 14. Timer 1 Registers

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

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 15. Selection of these events is done by 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 *TMS320C6000 Peripherals Reference Guide* (SPRU190).

### Table 15. 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. |

### 9.3.3 Interrupt Sources and Interrupt Selector

The C62x DSP core supports 16 prioritized interrupts, which are listed in Table 16. The highest-priority interrupt is INT_00 (dedicated to RESET) while the lowest-priority interrupt is INT_15. The first four interrupts (INT_00 to INT_03) are non-maskable and fixed. The remaining interrupts (INT_04 to INT_15) are maskable and default to the interrupt source specified in Table 16. The interrupt source for interrupts 4 to 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 16. C6203 DSP Interrupts

| CPU INTERRUPT NUMBER | INTERRUPT SELECTOR CONTROL REGISTER | SELECTOR VALUE (BINARY) | INTERRUPT EVENT | INTERRUPT SOURCE |
|---|---|---|---|---|
| INT_00[1] | – | – | RESET | |
| INT_01[1] | – | – | NMI | |
| INT_02[1] | – | – | Reserved | Reserved. Do not use. |
| INT_03[1] | – | – | Reserved | Reserved. Do not use. |
| INT_04[2] | MUXL[4:0] | 00100 | EXT_INT4 | External interrupt pin 4 |
| INT_05[2] | MUXL[9:5] | 00101 | EXT_INT5 | External interrupt pin 5 |
| INT_06[2] | MUXL[14:10] | 00110 | EXT_INT6 | External interrupt pin 6 |
| INT_07[2] | MUXL[20:16] | 00111 | EXT_INT7 | External interrupt pin 7 |
| INT_08[2] | MUXL[25:21] | 01000 | DMA_INT0 | DMA channel 0 interrupt |
| INT_09[2] | MUXL[30:26] | 01001 | DMA_INT1 | DMA channel 1 interrupt |
| INT_10[2] | MUXH[4:0] | 00011 | SD_INT | EMIF SDRAM timer interrupt |
| INT_11[2] | MUXH[9:5] | 01010 | DMA_INT2 | DMA channel 2 interrupt |
| INT_12[2] | MUXH[14:10] | 01011 | DMA_INT3 | DMA channel 3 interrupt |
| INT_13[2] | MUXH[20:16] | 00000 | DSP_INT | Host-processor-to-DSP interrupt |
| INT_14[2] | MUXH[25:21] | 00001 | TINT0 | Timer 0 interrupt |
| INT_15[2] | 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. |

[1] Interrupts INT_00 through INT_03 are non-maskable and fixed.

[2] Interrupts INT_04 through INT_15 are programmable by modifying the binary selector values in the Interrupt Selector Control registers fields. Table 16 shows the default interrupt sources for Interrupts INT_04 through INT_15. For more detailed information on interrupt sources and selection, see the *TMS320C6000 Peripherals Reference Guide* (SPRU190).

# 10 Application and Implementation

**NOTE**

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

## 10.1 Typical Application

### 10.1.1 Detailed Design Procedure

See the component selection in Table 17.

**Table 17. SMJ320C6203 PLL Component Selection Table[1]**

| 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 to 75 | 130 to 300 | 65 to 150 | 45.3 Ω | 47 nF | 10 pF | 75 |
| x6 | 21.7 to 50 | | | | | | |
| x7 | 18.6 to 42.9 | | | | | | |
| x8 | 16.3 to 37.5 | | | | | | |
| x9 | 14.4 to 33.3 | | | | | | |
| x10 | 13 to 30 | | | | | | |
| x11 | 11.8 to 27.3 | | | | | | |

[1] 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.

# 11 Power Supply Recommendations

## 11.1 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.

## 11.2 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.

## 11.3 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 and is corrected after the CPU detects 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 after the I/O power supply is turned on and the CPU detects 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 (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 using the C6000 platform of DSPs, the PCB should include separate power planes for core, I/O, and ground, all bypassed with high-quality low-ESL/ESR capacitors.

# 12 Device and Documentation Support

## 12.1 Device Support

### 12.1.1 Third-Party Products Disclaimer

TI'S PUBLICATION OF INFORMATION REGARDING THIRD-PARTY PRODUCTS OR SERVICES DOES NOT CONSTITUTE AN ENDORSEMENT REGARDING THE SUITABILITY OF SUCH PRODUCTS OR SERVICES OR A WARRANTY, REPRESENTATION OR ENDORSEMENT OF SUCH PRODUCTS OR SERVICES, EITHER ALONE OR IN COMBINATION WITH ANY TI PRODUCT OR SERVICE.

### 12.1.2 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:

#### *12.1.2.1 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.

#### *12.1.2.2 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 at www.ti.com 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.

### 12.1.3 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.

## Device Support (continued)

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

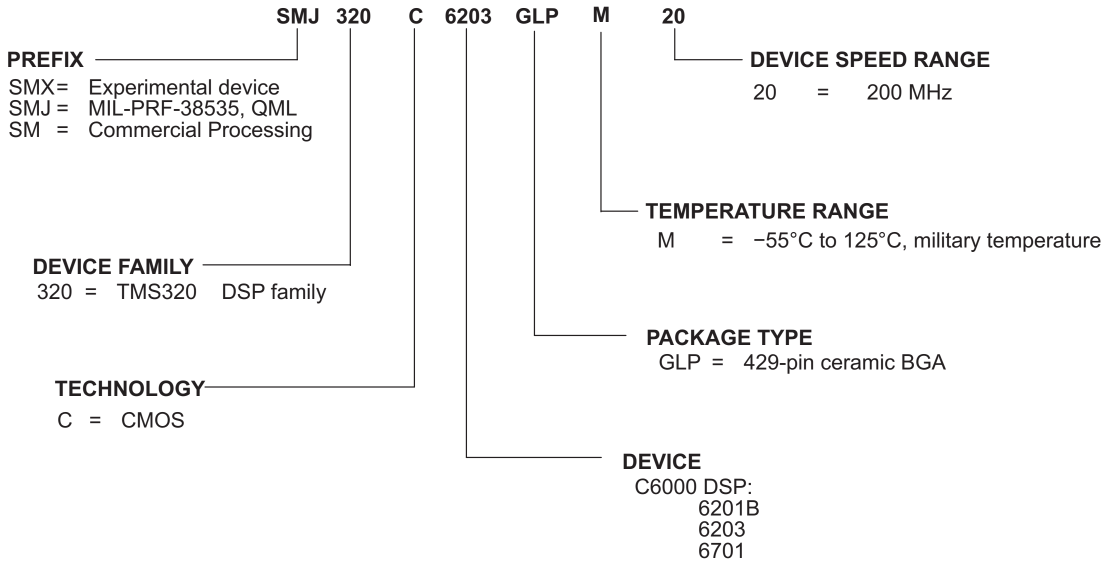


**Figure 56. SMJ320C6000 DSP Platform Device Nomenclature**

## 12.2 Documentation Support

### 12.2.1 Related Documentation

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* (SPRU189) describes the C6000 CPU (DSP core) architecture, instruction set, pipeline, and associated interrupts.

The *TMS320C6000 Peripherals Reference Guide* (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.

### Documentation Support (continued)

The *TMS320C6000 Technical Brief* (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 website at www.ti.com.

### 12.3 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

**TI E2E™ Online Community** ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

**Design Support** ***TI's Design Support*** Quickly find helpful E2E forums along with design support tools and contact information for technical support.

### 12.4 Trademarks

SMJ320C62x, VelociTI, C62x, C6000, Code Composer Studio, DSP/BIOS, XDS, E2E are trademarks of Texas Instruments.
Windows is a registered trademark of Microsoft Corporation.
Motorola is a registered trademark of Motorola Trademark Holdings, LLC.
All other trademarks are the property of their respective owners.

### 12.5 Electrostatic Discharge Caution

These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

### 12.6 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

## 13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead finish/ Ball material (6) | MSL Peak Temp (3) | Op Temp (°C) | Device Marking (4/5) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5962-0051001QXA | ACTIVE | CFCBGA | GLP | 429 | 1 | Non-RoHS & Green | SNPB | N / A for Pkg Type | -55 to 125 | 5962-0051001QX<br>A<br>SMJ320C6203GLP<br>M20 | Samples |
| SM320C6203GLPM20 | ACTIVE | CFCBGA | GLP | 429 | 1 | Non-RoHS & Green | SNPB | N / A for Pkg Type | -55 to 125 | SM320C6203GLPM<br>20 | Samples |
| SMJ320C6203GLPM20 | ACTIVE | CFCBGA | GLP | 429 | 1 | Non-RoHS & Green | SNPB | N / A for Pkg Type | -55 to 125 | 5962-0051001QX<br>A<br>SMJ320C6203GLP<br>M20 | Samples |

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

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

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

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

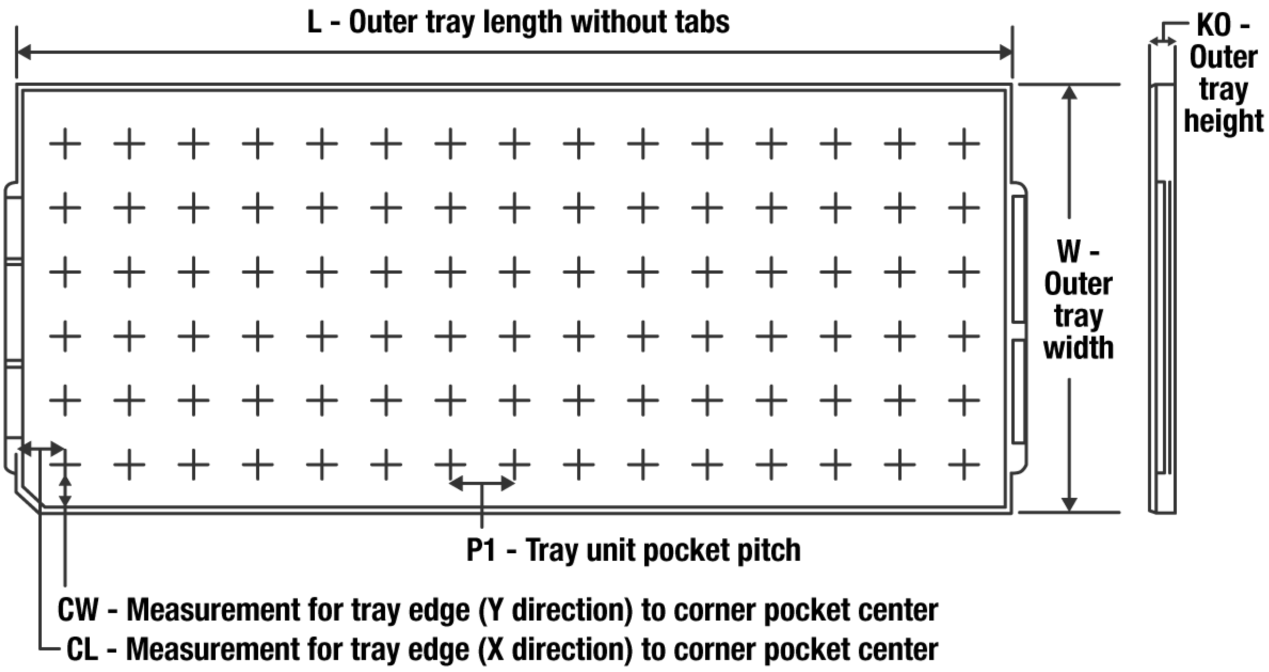
[5] Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

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

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

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

## TRAY

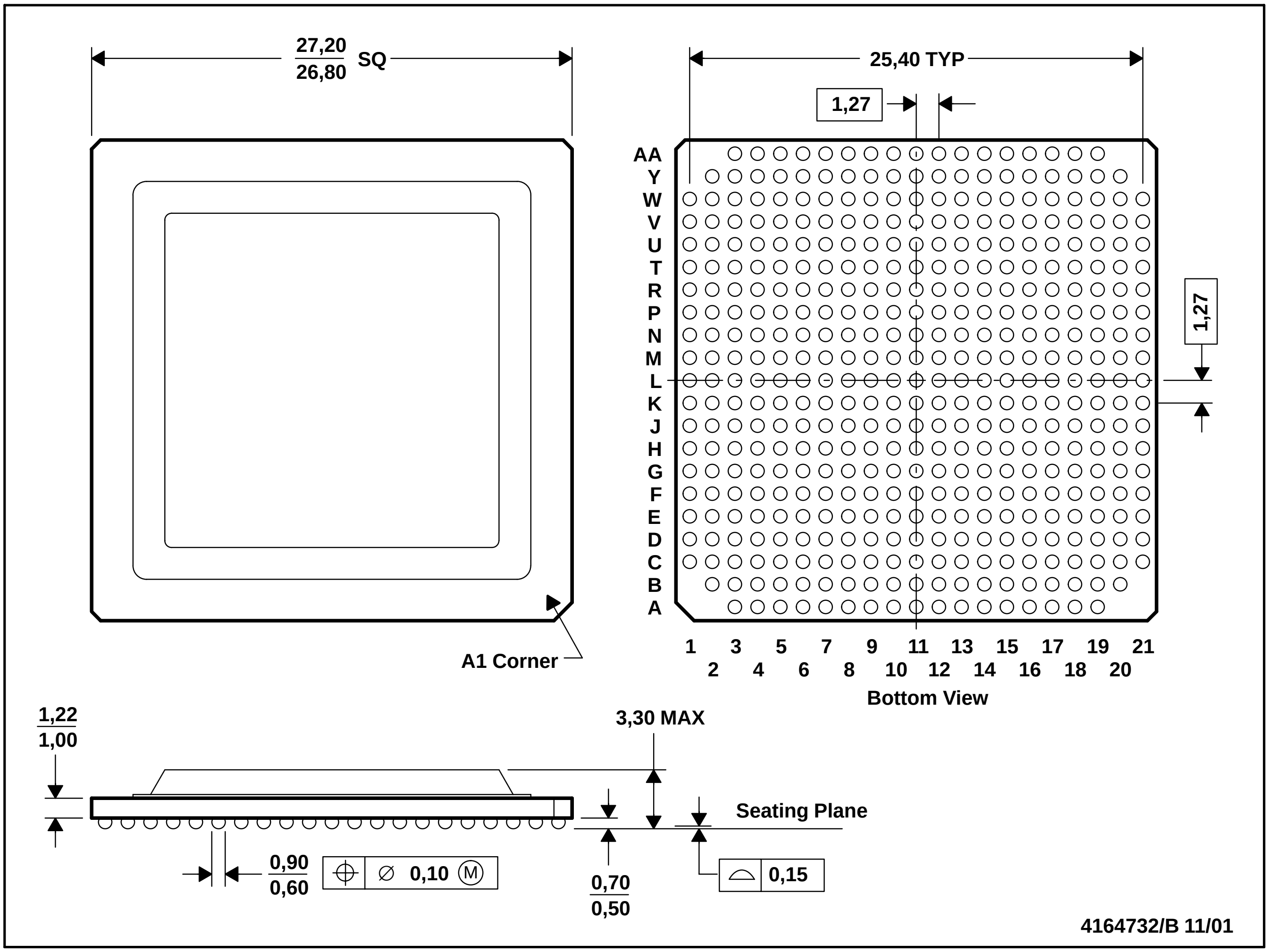

L - Outer tray length without tabs
KO - Outer tray height
W - Outer tray width
P1 - Tray unit pocket pitch
CW - Measurement for tray edge (Y direction) to corner pocket center
CL - Measurement for tray edge (X direction) to corner pocket center

Chamfer on Tray corner indicates Pin 1 orientation of packed units.

*All dimensions are nominal

| Device | Package Name | Package Type | Pins | SPQ | Unit array matrix | Max temperature (°C) | L (mm) | W (mm) | K0 (µm) | P1 (mm) | CL (mm) | CW (mm) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5962-0051001QXA | GLP | CFCBGA | 429 | 1 | 4x10 | 150 | 315 | 135.9 | 7620 | 29.2 | 26.1 | 24.15 |
| SM320C6203GLPM20 | GLP | CFCBGA | 429 | 1 | 4x10 | 150 | 315 | 135.9 | 7620 | 29.2 | 26.1 | 24.15 |
| SMJ320C6203GLPM20 | GLP | CFCBGA | 429 | 1 | 4x10 | 150 | 315 | 135.9 | 7620 | 29.2 | 26.1 | 24.15 |

## 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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265