SPRS104A – OCTOBER 1999 – REVISED MARCH 2000

- Highest Performance Fixed-Point Digital Signal Processors (DSPs) TMS320C62x
  - 5-, 4-, 3.33-ns Instruction Cycle Time
  - 200-, 250-, 300-MHz Clock Rate
  - Eight 32-Bit Instructions/Cycle
  - 1600, 2000, 2400 MIPS
- VelociTI™ Advanced Very Long Instruction Word (VLIW) ’C62x CPU Core
  - Eight Highly Independent Functional Units:
    - Six 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
- On-Chip SRAM
  - 1M-Bit (’C6204)
  - 3M-Bit (’C6202/’C6202B)
  - 7M-Bit (’C6203)
- 32-Bit External Memory Interface (EMIF)
  - Glueless Interface to Synchronous Memories: SDRAM or SBSRAM
  - Glueless Interface to Asynchronous Memories: SRAM and EPROM
  - 52M-Byte Addressable External Memory Space
- Four-Channel Bootloading Direct-Memory-Access (DMA) Controller With an Auxiliary Channel
- Flexible Phase-Locked-Loop (PLL) Clock Generator
- 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
- Multichannel Buffered Serial Ports (McBSPs)
  - Direct Interface to T1/E1, MVIP, SCSA Framers
  - ST-Bus-Switching Compatible
  - Up to 256 Channels Each
  - AC97-Compatible
  - Serial-Peripheral Interface (SPI) Compatible (Motorola™)
- Two 32-Bit General-Purpose Timers
- IEEE-1149.1 (JTAG†) Boundary-Scan-Compatible
- 352-Pin BGA Package (GJL) (’02/02B/03)
- 384-Pin BGA Package (GLS) (’02/02B/03)
- 340-Pin BGA Package (GLW) (’C6204 only)
  - Pin-Compatible With the GLS Package Except Inner Row of Balls (Additional Power and Ground Pins) are Removed‡
- 0.18-μm/5-Level Metal Process (’6202 only)
  0.15-μm/5-Level Metal Process (’02B/03/04)
  - CMOS Technology
- 3.3-V I/Os, 1.8-V Internal (’C6202 only)
  3.3-V I/Os, 1.5-V Internal (’C6202B/03/04)

PRODUCT PREVIEW

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

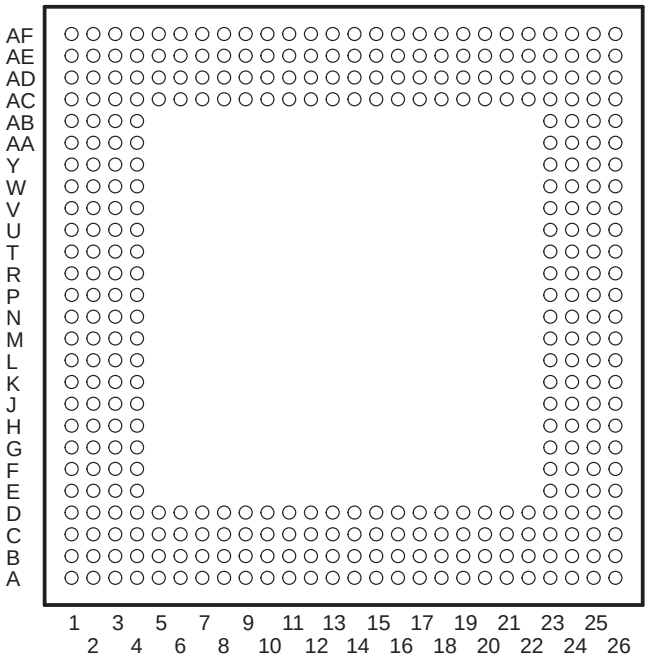
VelociTI is a trademark of Texas Instruments Incorporated.
Motorola is a trademark of Motorola, Inc.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.
‡ For more details, see the GLS/GLW BGA package bottom view.

Table of Contents

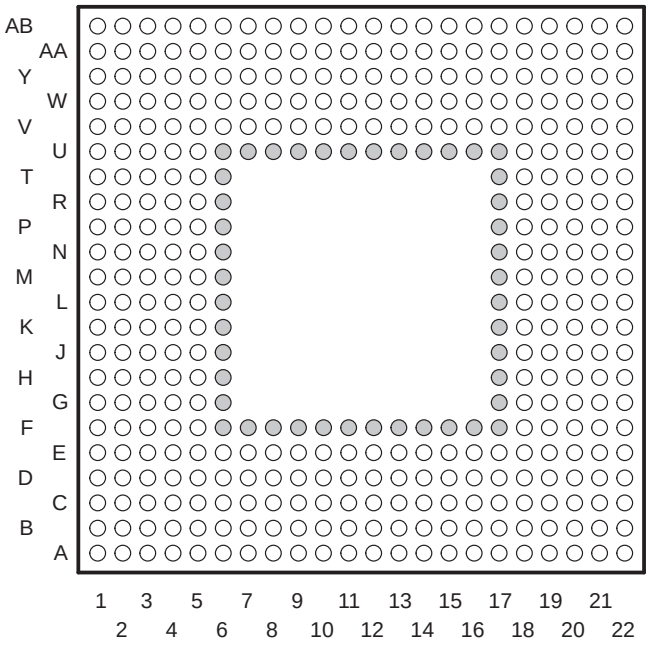
PRODUCT PREVIEW

GJL/GLS/GLW BGA packages (bottom view)

GJL 352-PIN BALL GRID ARRAY (BGA) PACKAGE ('C6202/02B/03 ONLY)
(BOTTOM VIEW)



GLS 384-PIN BGA PACKAGE ('C6202/02B/03 ONLY)
GLW 340-PIN BGA PACKAGE ('C6204 ONLY)
(BOTTOM VIEW)



○ **These balls are *NOT* applicable for the 'C6204 devices GLW 340-pin BGA package.**

PRODUCT PREVIEW

device selection guide

Table 1 provides an overview of the TMS320C6202/02B/03/04 pin-compatible DSPs. The table shows significant features of each device, including the capacity of on-chip RAM, the peripherals, the execution time, and the package type with pin count, etc.

Table 1. TMS320C6202/02B/03/04 DSP Selection Guide

HARDWARE FEATURES		’C6202	’C6202B	’C6203	’C6204
Peripherals	EMIF	√	√	√	√
	DMA	4-Channel	4-Channel With Throughput Enhancements	4-Channel With Throughput Enhancements	4-Channel With Throughput Enhancements
	Expansion Bus	√	√	√	√
	McBSPs	3	3	3	2
	32-Bit Timers	2	2	2	2
Internal Program Memory	Size (Bytes)	256K	256K	384K	64K
	Organization	Block 0: 128K Bytes Mapped Program Block 1: 128K Bytes Cache/Mapped Program	Block 0: 128K Bytes Mapped Program Block 1: 128K Bytes Cache/Mapped Program	Block 0: 256K Bytes Mapped Program Block 1: 128K Bytes Cache/Mapped Program	1 Block: 64K Bytes Cache/Mapped Program
Internal Data Memory	Size (Bytes)	128K	128K	512K	64K
	Organization	2 Blocks: Four 16-Bit Banks per Block 50/50 Split	2 Blocks: Four 16-Bit Banks per Block 50/50 Split	2 Blocks: Four 16-Bit Banks per Block 50/50 Split	2 Blocks: Four 16-Bit Banks per Block 50/50 Split
Frequency	MHz	200, 250	250	250, 300	200
Cycle Time	ns	4 ns (’6202-250) 5 ns (’6202-200)	4 ns (’6202B-250)	3.33 ns (’6203-300) 4 ns (’6203-250)	5 ns (’6204-200)
Voltage	Core (V)	1.8	1.5	1.5	1.5
	I/O (V)	3.3	3.3	3.3	3.3
PLL Options: In Both Packages	Bypass (x1)	√	√	√	√
	x4	√	√	√	√
	x8	–	√	√	–
	x10	–	√	√	–
Additional PLL Options: 18 x 18 mm Packages (GLS/GLW only)	x6	–	√	√	–
	x7	–	√	√	–
	x9	–	√	√	–
	x11	–	√	√	–
BGA Package	27 x 27 mm	352-pin GJL	352-pin GJL	352-pin GJL	–
	18 x 18 mm	384-pin GLS	384-pin GLS	384-pin GLS	340-pin GLW
Process Technology	µm	0.18 µm (18C05)	0.15 µm (15C05)	0.15 µm (15C05)	0.15 µm (15C05)
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	PD	PP	AI	PP

description

The TMS320C6202, TMS320C6202B, TMS320C6203, and TMS320C6204 devices are part of the TMS320C62x fixed-point DSP family in the TMS320C6000 platform. The ’C62x devices are based on the high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunction applications.

The TMS320C62x DSP offers cost-effective solutions to high-performance DSP programming challenges. The TMS320C6202B/’03 has a performance of up to 2400 million instructions per second (MIPS) at 300 MHz, while the TMS320C6202 has a performance of up to 2000 MIPS at 250 MHz, and the TMS320C6204 has a performance of up to 1600 MIPS at 200 MHz. The ’C6202/’02B/’03/’04 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. These processors have 32 general-purpose registers of 32-bit word length and eight highly independent functional units. The eight functional units provide six arithmetic logic units (ALUs) for a high degree of parallelism and two 16-bit multipliers for a 32-bit result. The ’C6202/’02B/’03/’04 can produce two multiply-accumulates (MACs) per cycle. This gives a total of 600 million MACs per second (MMACS) for the ’C6202B/’03 device, a total of 500 MMACS for the ’C6202 device, and a total of 400 MMACS for the ’C6204 device. The ’C6202/’02B/’03/’04 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The TMS320C62x DSPs include an on-chip memory, with the ’C6203 device offering the most memory at 7 Mbits. For the ’C6202/’02B device, program memory consists of two blocks, with a 128K-byte block configured as memory-mapped program space, and the other 128K-byte block user-configurable as cache or memory-mapped program space. Data memory consists of two 64K-byte blocks of RAM. Similarly, the ’C6203 device program memory consists of two blocks, with a 256K-byte block configured as memory-mapped program space, and the other 128K-byte block user-configurable as cache or memory-mapped program space. Data memory consists of two 256K-byte blocks of RAM. For the ’C6204 device, program memory consists of a single 64K-byte block that is user-configured as cache or memory-mapped program space. Data memory consists of two 32K-byte blocks of RAM.

The ’C6202/’02B/’03/’04 device has a powerful and diverse set of peripherals. The peripheral set includes multichannel buffered serial ports (McBSPs), general-purpose timers, a 32-bit expansion bus (XB) that offers ease of interface to synchronous or asynchronous industry-standard host bus protocols, and a glueless 32-bit external memory interface (EMIF) capable of interfacing to SDRAM or SBSRAM and asynchronous peripherals.

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

PRODUCT PREVIEW

Windows is a registered trademark of the Microsoft Corporation.

’C62x device compatibility

The TMS320C6202, ’C6202B, ’C6203, and ’C6204 devices are pin-compatible; thus, making new system designs easier and providing faster time to market. The following list summarizes the ’C62x device characteristic differences:

- Core Supply Voltage (1.8 V versus 1.5 V)
- PLL Options Availability

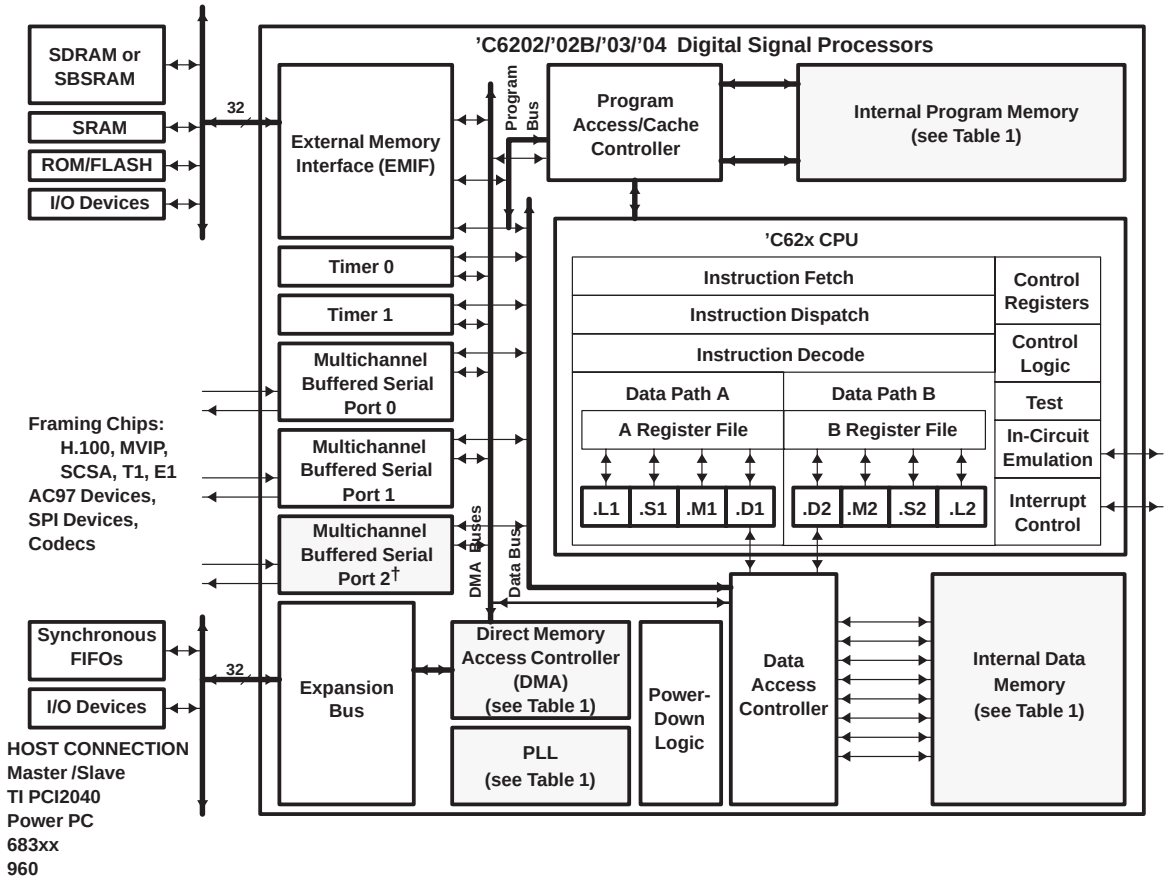
  Table 1 identifies the available PLL multiply factors [e.g., CLKIN x1 (PLL bypassed), x4] for each of the ’C62x devices. For additional details on the PLL clock module, see the Clock PLL section of this data sheet.
- On-Chip Memory Size

  The ’C6202/’02B, ’C6203, and ’C6204 devices have different on-chip program memory and data memory sizes (see Table 1).
- McBSPs

  The ’C6204 device has two McBSPs while the ’C6202/’02B/’03 devices have three McBSPs on-chip.

For a more detailed discussion on migration concerns, and similarities/differences between the ’C6202, ’C6202B, ’C6203, and ’C6204 devices, see the *How to Begin Development and Migrate Across the TMS320C6202/6202B/6203/6204 DSPs* application report (literature number SPRA603) document.

functional and CPU block diagram (’C62x devices)



† McBSP2 is *not* applicable for the ’C6204 device.

PRODUCT PREVIEW

CPU description

The CPU fetches VelociTI advanced very-long instruction words (VLIW) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VelociTI VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the ’C62x CPU from other VLIW architectures.

The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files each contain 16 32-bit registers for a total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU (see the Functional and CPU Block Diagram and Figure 1). The four functional units on each side of the CPU can freely share the 16 registers belonging to that side. Additionally, each side features a single data bus connected to all the registers on the other side, by which the two sets of functional units can access data from the register files on the opposite side. While register access by functional units on the same side of the CPU as the register file can service all the units in a single clock cycle, register access using the register file across the CPU supports one read and one write per cycle.

Another key feature of the ’C62x CPU is the load/store architecture, where all instructions operate on registers (as opposed to data in memory). Two sets of data-addressing units (.D1 and .D2) are responsible for all data transfers between the register files and the memory. The data address driven by the .D units allows data addresses generated from one register file to be used to load or store data to or from the other register file. The ’C62x CPU supports a variety of indirect addressing modes using either linear- or circular-addressing modes with 5- or 15-bit offsets. All instructions are conditional, and most can access any one of the 32 registers. Some registers, however, are singled out to support specific addressing or to hold the condition for conditional instructions (if the condition is not automatically “true”). The two .M functional units are dedicated for multiplies. The two .S and .L functional units perform a general set of arithmetic, logical, and branch functions with results available every clock cycle.

The processing flow begins when a 256-bit-wide instruction fetch packet is fetched from a program memory. The 32-bit instructions destined for the individual functional units are “linked” together by “1” bits in the least significant bit (LSB) position of the instructions. The instructions that are “chained” together for simultaneous execution (up to eight in total) compose an execute packet. A “0” in the LSB of an instruction breaks the chain, effectively placing the instructions that follow it in the next execute packet. If an execute packet crosses the fetch-packet boundary (256 bits wide), the assembler places it in the next fetch packet, while the remainder of the current fetch packet is padded with NOP instructions. The number of execute packets within a fetch packet can vary from one to eight. Execute packets are dispatched to their respective functional units at the rate of one per clock cycle and the next 256-bit fetch packet is not fetched until all the execute packets from the current fetch packet have been dispatched. After decoding, the instructions simultaneously drive all active functional units for a maximum execution rate of eight instructions every clock cycle. While most results are stored in 32-bit registers, they can be subsequently moved to memory as bytes or half-words as well. All load and store instructions are byte-, half-word, or word-addressable.

CPU description (continued)

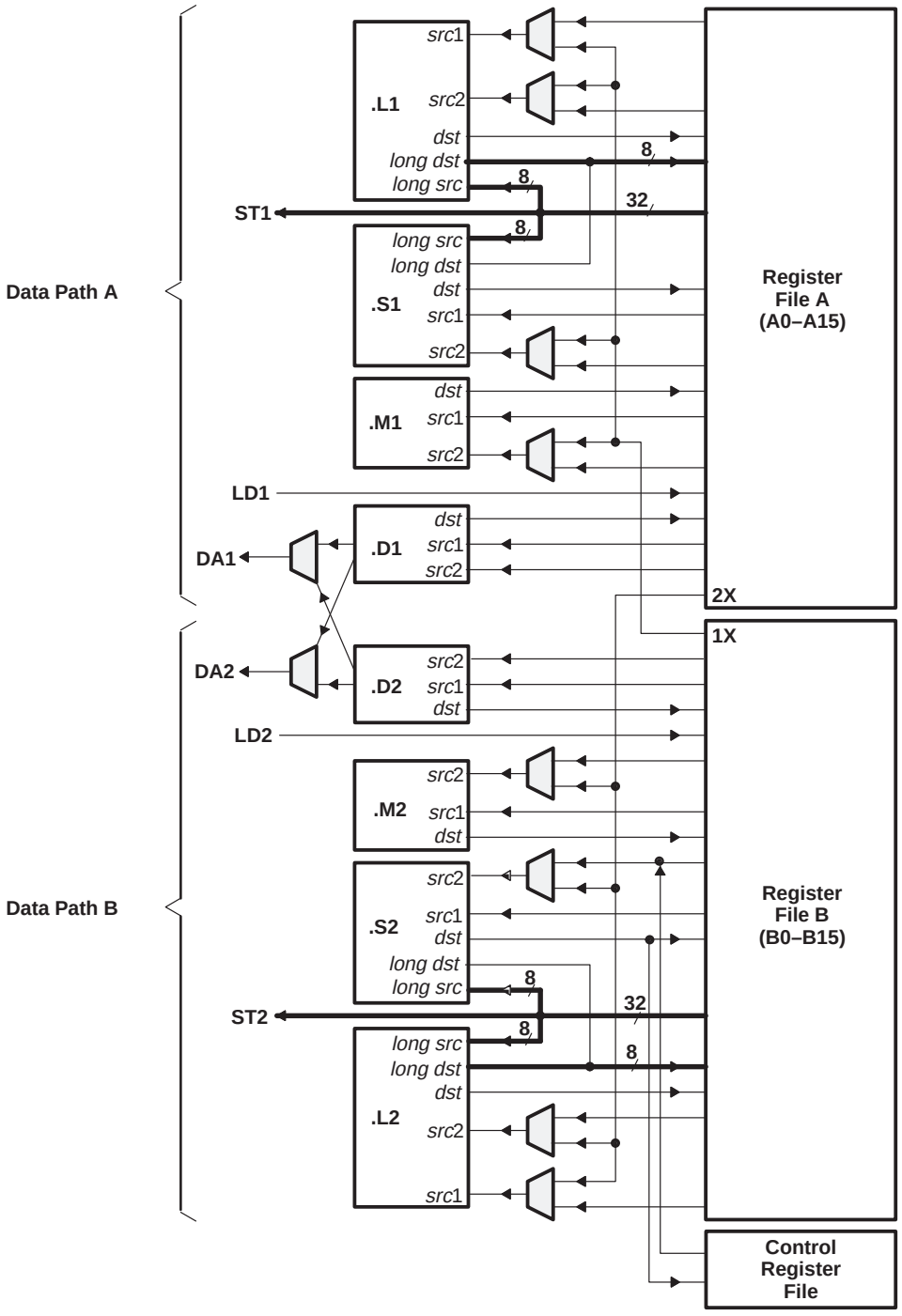
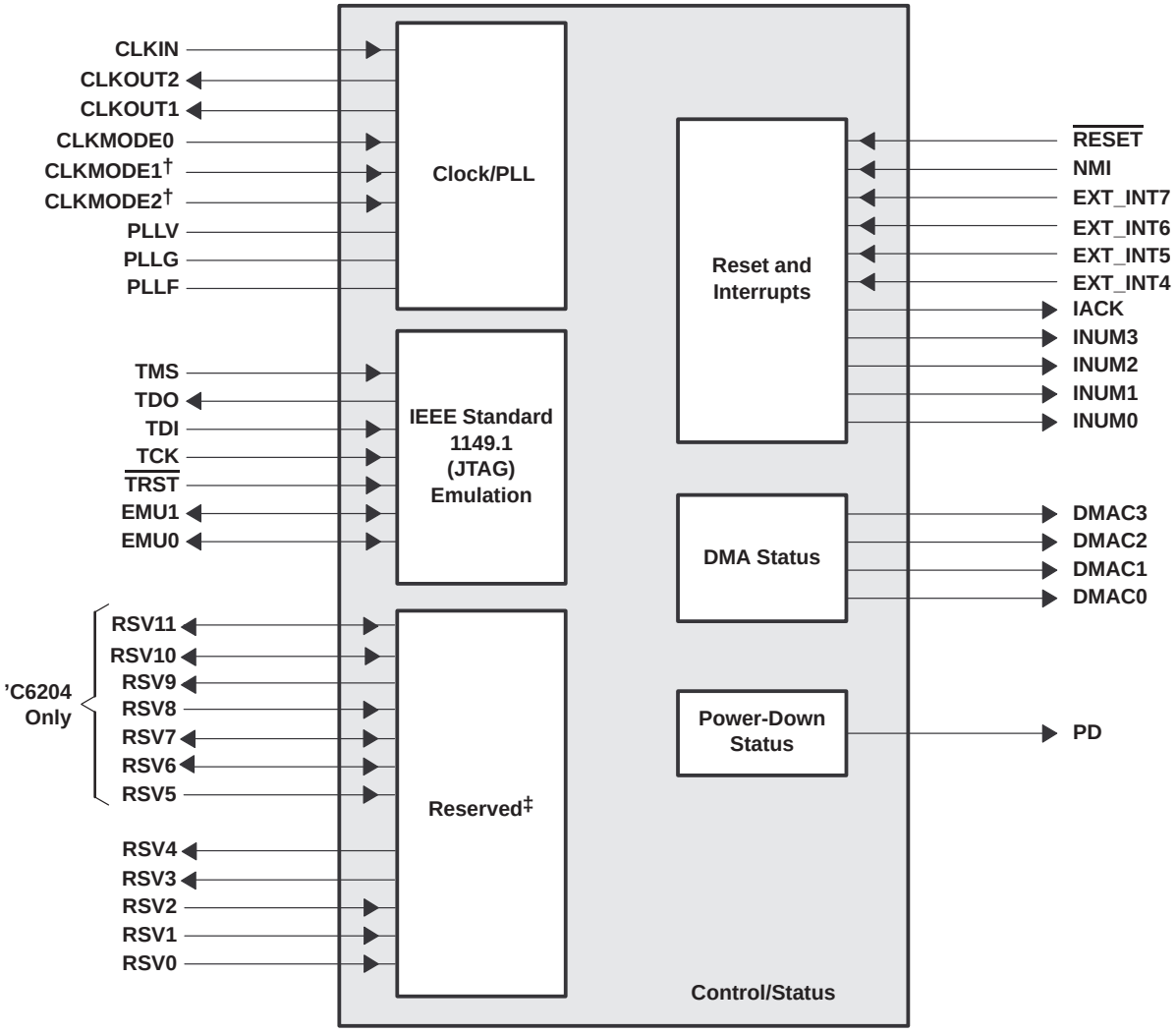


Figure 1. TMS320C62x CPU Data Paths

signal groups description



† CLKMODE1 is NOT available on the 'C6202 device GJL package.
CLKMODE2 is NOT available on the GJL packages for the 'C6202/'02B/'03 devices.
‡ RSV5 through RSV11 pins are used on the 'C6204 device only.

Figure 2. CPU Signals

signal groups description (continued)

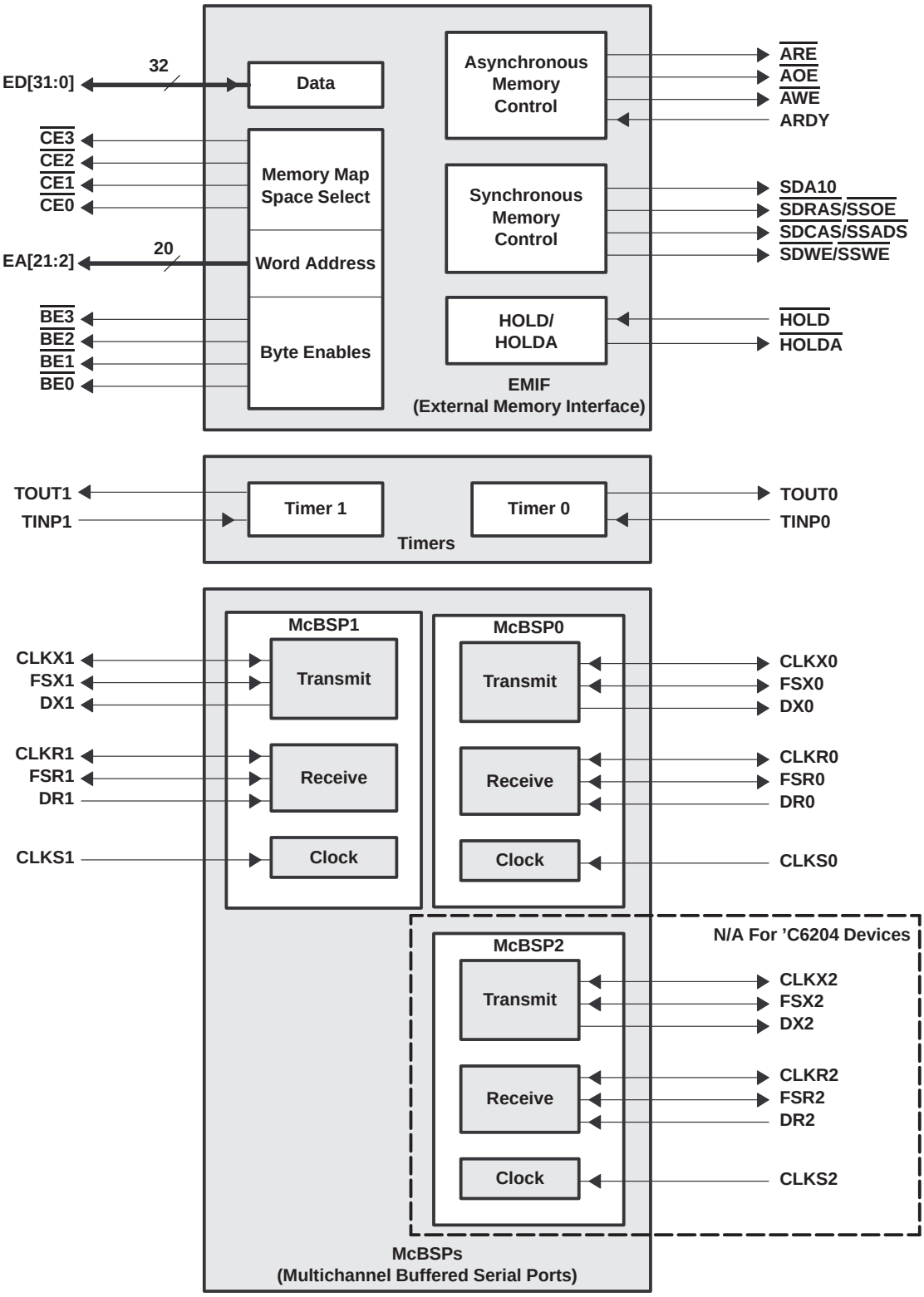


Figure 3. Peripheral Signals

signal groups description (continued)

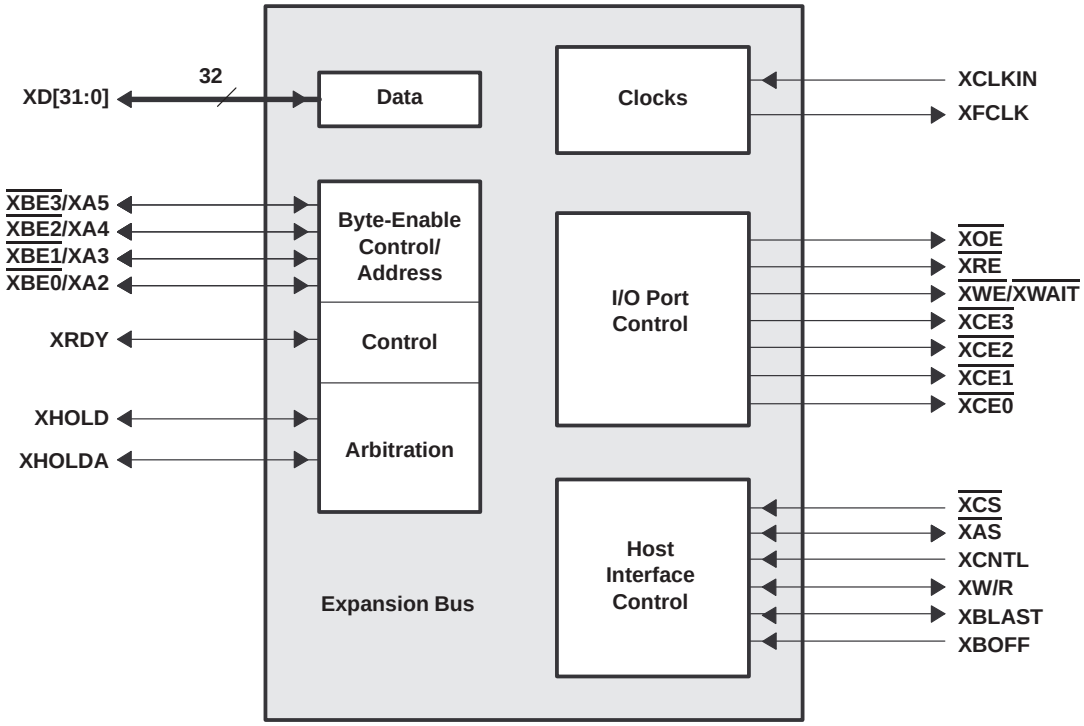


Figure 3. Peripheral Signals (Continued)

Signal Descriptions

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
CLOCK/PLL					
CLKIN	C12	B10	B10	I	Clock Input
CLKOUT1	AD20	Y18	Y18	O	Clock output at full device speed
CLKOUT2	AC19	AB19	AB19	O	Clock output at half of device speed • Used for synchronous memory interface
CLKMODE0	B15	B12	B12	I	Clock mode selects • Selects what multiply factors of the input clock frequency the CPU frequency equals. For more detail on CLKMODE pins and the PLL multiply factors, see the Clock PLL section of this data sheet.
CLKMODE1	C11§	A9¶	A9¶	I	
CLKMODE2	–	A14¶	A14¶	I	
PLLV#	D13	C11	C11	A‖	PLL analog V_{CC} connection for the low-pass filter
PLLG#	D14	C12	C12	A‖	PLL analog GND connection for the low-pass filter
PLLF#	C13	A11	A11	A‖	PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION					
TMS	AD7	Y5	Y5	I	JTAG test-port mode select (features an internal pullup)
TDO	AE6	AA4	AA4	O/Z	JTAG test-port data out
TDI	AF5	Y4	Y4	I	JTAG test-port data in (features an internal pullup)
TCK	AE5	AB2	AB2	I	JTAG test-port clock
$\overline{\text{TRST}}$	AC7	AA3	AA3	I	JTAG test-port reset (features an internal pulldown)
EMU1	AF6	AA5	AA5	I/O/Z	Emulation pin 1, pullup with a dedicated 20-kΩ resistor☆
EMU0	AC8	AB4	AB4	I/O/Z	Emulation pin 0, pullup with a dedicated 20-kΩ resistor☆
RESET AND INTERRUPTS					
$\overline{\text{RESET}}$	K2	J3	J3	I	Device reset
NMI	L2	K2	K2	I	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	V4	U2	U2	I	External interrupts • Edge-driven (rising edge)
EXT_INT6	Y2	U3	U3		
EXT_INT5	AA1	W1	W1		
EXT_INT4	W4	V2	V2		
IACK	Y1	V1	V1	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	V2	R3	R3	O	Active interrupt identification number • Valid during IACK for all active interrupts (not just external) • Encoding order follows the interrupt-service fetch-packet ordering
INUM2	U4	T1	T1		
INUM1	V3	T2	T2		
INUM0	W2	T3	T3		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground
§ For the 'C6202 GJL package only, the C11 pin is ground (V_{SS}). For all other 'C62x GJL packages, the C11 pin is CLKMODE1.
¶ For the 'C6202 GLS and 'C6204 GLW packages, the CLKMODE2 (A14) and CLKMODE1 (A9) pins are internally unconnected.
# PLLV, PLLG, and PLLF are not part of external voltage supply or ground. See the *clock PLL* section for information on how to connect these pins.
‖ A = Analog Signal (PLL Filter)
☆ For emulation and normal operation, pull up EMU1 and EMU0 with a dedicated 20-kΩ resistor. For boundary scan, pull down EMU1 and EMU0 with a dedicated 20-kΩ resistor.

PRODUCT PREVIEW

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
POWER-DOWN STATUS					
PD	AB2	Y2	Y2	O	Power-down modes 2 or 3 (active if high)
EXPANSION BUS					
XCLKIN	A9	C8	C8	I	Expansion bus synchronous host interface clock input
XFCLK	B9	A8	A8	O	Expansion bus FIFO interface clock output
XD31	D15	C13	C13	I/O/Z	Expansion bus data • Used for transfer of data, address, and control • Also controls initialization of DSP modes and expansion bus at reset via pullup/ pulldown resistors (Note: Reserved boot configuration fields should be pulled down.) – XCE[3:0] memory type – XBLAST polarity – XW/R polarity – Asynchronous or synchronous host operation – Arbitration mode (internal or external) – FIFO mode – Little endian/big endian – Boot mode
XD30	B16	A13	A13		
XD29	A17	C14	C14		
XD28	B17	B14	B14		
XD27	D16	B15	B15		
XD26	A18	C15	C15		
XD25	B18	A15	A15		
XD24	D17	B16	B16		
XD23	C18	C16	C16		
XD22	A20	A17	A17		
XD21	D18	B17	B17		
XD20	C19	C17	C17		
XD19	A21	B18	B18		
XD18	D19	A19	A19		
XD17	C20	C18	C18		
XD16	B21	B19	B19		
XD15	A22	C19	C19		
XD14	D20	B20	B20		
XD13	B22	A21	A21		
XD12	E25	C21	C21		
XD11	F24	D20	D20		
XD10	E26	B22	B22		
XD9	F25	D21	D21		
XD8	G24	E20	E20		
XD7	H23	E21	E21		
XD6	F26	D22	D22		
XD5	G25	F20	F20		
XD4	J23	F21	F21		
XD3	G26	E22	E22		
XD2	H25	G20	G20		
XD1	J24	G21	G21		
XD0	K23	G22	G22		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
EXPANSION BUS (CONTINUED)					
$\overline{\text{XCE3}}$	F2	D2	D2	O/Z	Expansion bus I/O port memory space enables • Enabled by bits 28, 29, and 30 of the word address • Only one asserted during any I/O port data access
$\overline{\text{XCE2}}$	E1	B1	B1		
$\overline{\text{XCE1}}$	F3	D3	D3		
$\overline{\text{XCE0}}$	E2	C2	C2		
$\overline{\text{XBE3}}$/XA5	C7	C5	C5	I/O/Z	Expansion bus multiplexed byte-enable control/address signals • Act as byte enable for host port operation • Act as address for I/O port operation
$\overline{\text{XBE2}}$/XA4	D8	A4	A4		
$\overline{\text{XBE1}}$/XA3	A6	B5	B5		
$\overline{\text{XBE0}}$/XA2	C8	C6	C6		
$\overline{\text{XOE}}$	A7	A6	A6	O/Z	Expansion bus I/O port output enable
$\overline{\text{XRE}}$	C9	C7	C7	O/Z	Expansion bus I/O port read enable
$\overline{\text{XWE}}$/$\overline{\text{XWAIT}}$	D10	B7	B7	O/Z	Expansion bus I/O port write enable and host port wait signals
$\overline{\text{XCS}}$	A10	C9	C9	I	Expansion bus host port chip-select input
$\overline{\text{XAS}}$	D9	B6	B6	I/O/Z	Expansion bus host port address strobe
XCNTL	B10	B9	B9	I	Expansion bus host control. XCNTL selects between expansion bus address or data register
XW/R	D11	B8	B8	I/O/Z	Expansion bus host port write/read enable. XW/R polarity selected at reset
XRDY	A5	C4	C4	I/O/Z	Expansion bus host port ready (active low) and I/O port ready (active high)
XBLAST	B6	B4	B4	I/O/Z	Expansion bus host port burst last–polarity selected at reset
XBOFF	B11	A10	A10	I	Expansion bus back off
XHOLD	B5	A2	A2	I/O/Z	Expansion bus hold request
XHOLDA	D7	B3	B3	I/O/Z	Expansion bus hold acknowledge
EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY					
$\overline{\text{CE3}}$	AB25	Y21	Y21	O/Z	Memory space enables • Enabled by bits 24 and 25 of the word address • Only one asserted during any external data access
$\overline{\text{CE2}}$	AA24	W20	W20		
$\overline{\text{CE1}}$	AB26	AA22	AA22		
$\overline{\text{CE0}}$	AA25	W21	W21		
$\overline{\text{BE3}}$	Y24	V20	V20	O/Z	Byte-enable control • Decoded from the two lowest bits of the internal address • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{\text{BE2}}$	W23	V21	V21		
$\overline{\text{BE1}}$	AA26	W22	W22		
$\overline{\text{BE0}}$	Y25	U20	U20		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
EMIF – ADDRESS					
EA21	J25	H20	H20	O/Z	External address (word address)
EA20	J26	H21	H21		
EA19	L23	H22	H22		
EA18	K25	J20	J20		
EA17	L24	J21	J21		
EA16	L25	K21	K21		
EA15	M23	K20	K20		
EA14	M24	K22	K22		
EA13	M25	L21	L21		
EA12	N23	L20	L20		
EA11	P24	L22	L22		
EA10	P23	M20	M20		
EA9	R25	M21	M21		
EA8	R24	N22	N22		
EA7	R23	N20	N20		
EA6	T25	N21	N21		
EA5	T24	P21	P21		
EA4	U25	P20	P20		
EA3	T23	R22	R22		
EA2	V26	R21	R21		
EMIF – DATA					
ED31	AD8	Y6	Y6	I/O/Z	External data
ED30	AC9	AA6	AA6		
ED29	AF7	AB6	AB6		
ED28	AD9	Y7	Y7		
ED27	AC10	AA7	AA7		
ED26	AE9	AB8	AB8		
ED25	AF9	Y8	Y8		
ED24	AC11	AA8	AA8		
ED23	AE10	AA9	AA9		
ED22	AD11	Y9	Y9		
ED21	AE11	AB10	AB10		
ED20	AC12	Y10	Y10		
ED19	AD12	AA10	AA10		
ED18	AE12	AA11	AA11		
ED17	AC13	Y11	Y11		
ED16	AD14	AB12	AB12		
ED15	AC14	Y12	Y12		
ED14	AE15	AA12	AA12		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
EMIF – DATA (CONTINUED)					
ED13	AD15	AA13	AA13	I/O/Z	External data
ED12	AC15	Y13	Y13		
ED11	AE16	AB13	AB13		
ED10	AD16	Y14	Y14		
ED9	AE17	AA14	AA14		
ED8	AC16	AA15	AA15		
ED7	AF18	Y15	Y15		
ED6	AE18	AB15	AB15		
ED5	AC17	AA16	AA16		
ED4	AD18	Y16	Y16		
ED3	AF20	AB17	AB17		
ED2	AC18	AA17	AA17		
ED1	AD19	Y17	Y17		
ED0	AF21	AA18	AA18		
EMIF – ASYNCHRONOUS MEMORY CONTROL					
$\overline{\text{ARE}}$	V24	T21	T21	O/Z	Asynchronous memory read enable
$\overline{\text{AOE}}$	V25	R20	R20	O/Z	Asynchronous memory output enable
$\overline{\text{AWE}}$	U23	T22	T22	O/Z	Asynchronous memory write enable
ARDY	W25	T20	T20	I	Asynchronous memory ready input
EMIF – SYNCHRONOUS DRAM (SDRAM)/SYNCHRONOUS BURST SRAM (SBSRAM) CONTROL					
SDA10	AE21	AA19	AA19	O/Z	SDRAM address 10 (separate for deactivate command)
$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$	AE22	AB21	AB21	O/Z	SDRAM column-address strobe/SBSRAM address strobe
$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$	AF22	Y19	Y19	O/Z	SDRAM row-address strobe/SBSRAM output enable
$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$	AC20	AA20	AA20	O/Z	SDRAM write enable/SBSRAM write enable
EMIF – BUS ARBITRATION					
$\overline{\text{HOLD}}$	Y26	V22	V22	I	Hold request from the host
$\overline{\text{HOLDA}}$	V23	U21	U21	O	Hold-request-acknowledge to the host
TIMERS					
TOUT1	J4	F2	F2	O	Timer 1 or general-purpose output
TINP1	G2	F3	F3	I	Timer 1 or general-purpose input
TOUT0	F1	D1	D1	O	Timer 0 or general-purpose output
TINP0	H4	E2	E2	I	Timer 0 or general-purpose input
DMA ACTION COMPLETE STATUS					
DMAC3	Y3	V3	V3	O	DMA action complete
DMAC2	AA2	W2	W2		
DMAC1	AB1	AA1	AA1		
DMAC0	AA3	W3	W3		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)					
CLKS0	M4	K3	K3	I	External clock source (as opposed to internal)
CLKR0	M2	L2	L2	I/O/Z	Receive clock
CLKX0	M3	K1	K1	I/O/Z	Transmit clock
DR0	R2	M2	M2	I	Receive data
DX0	P4	M3	M3	O/Z	Transmit data
FSR0	N3	M1	M1	I/O/Z	Receive frame sync
FSX0	N4	L3	L3	I/O/Z	Transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)					
CLKS1	G1	E1	E1	I	External clock source (as opposed to internal)
CLKR1	J3	G2	G2	I/O/Z	Receive clock
CLKX1	H2	G3	G3	I/O/Z	Transmit clock
DR1	L4	H1	H1	I	Receive data
DX1	J1	H2	H2	O/Z	Transmit data
FSR1	J2	H3	H3	I/O/Z	Receive frame sync
FSX1	K4	G1	G1	I/O/Z	Transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 2 (McBSP2) ('C6202/'C6202B/'C6203 ONLY)					
CLKS2	R3	N1	–	I	External clock source (as opposed to internal)
CLKR2	T2	N2	–	I/O/Z	Receive clock
CLKX2	R4	N3	–	I/O/Z	Transmit clock
DR2	V1	R2	–	I	Receive data
DX2	T4	R1	–	O/Z	Transmit data
FSR2	U2	P3	–	I/O/Z	Receive frame sync
FSX2	T3	P2	–	I/O/Z	Transmit frame sync
RESERVED FOR TEST					
RSV0	L3	J2	J2	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV1	G3	E3	E3	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV2	A12	B11	B11	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV3	C15	B13	B13	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV4	D12	C10	C10	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
ADDITIONAL RESERVED FOR TEST ('C6204 ONLY)					
RSV5	–	–	N1	I	Reserved (leave unconnected)
RSV6	–	–	N2	I/O	Reserved (leave unconnected)
RSV7	–	–	N3	I/O	Reserved (leave unconnected)
RSV8	–	–	R2	I	Reserved (leave unconnected)
RSV9	–	–	R1	O	Reserved (leave unconnected)
RSV10	–	–	P3	I/O	Reserved (leave unconnected)
RSV11	–	–	P2	I/O	Reserved (leave unconnected)

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
SUPPLY VOLTAGE PINS					
DV_{DD}	A11	A3	A3	S	3.3-V supply voltage (I/O)
	A16	A7	A7		
	B7	A16	A16		
	B8	A20	A20		
	B19	D4	D4		
	B20	D6	D6		
	C6	D7	D7		
	C10	D9	D9		
	C14	D10	D10		
	C17	D13	D13		
	C21	D14	D14		
	G4	D16	D16		
	G23	D17	D17		
	H3	D19	D19		
	H24	F1	F1		
	K3	F4	F4		
	K24	F19	F19		
	L1	F22	F22		
	L26	G4	G4		
	N24	G19	G19		
	P3	J4	J4		
	T1	J19	J19		
	T26	K4	K4		
	U3	K19	K19		
	U24	L1	L1		
	W3	M22	M22		
	W24	N4	N4		
	Y4	N19	N19		
	Y23	P4	P4		
	AD6	P19	P19		
	AD10	T4	T4		
	AD13	T19	T19		
	AD17	U1	U1		
	AD21	U4	U4		
	AE7	U19	U19		
	AE8	U22	U22		
	AE19	W4	W4		
	AE20	W6	W6		
	AF11	W7	W7		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
SUPPLY VOLTAGE PINS (CONTINUED)					
DV_{DD}	AF16	W9	W9	S	3.3-V supply voltage (I/O)
	–	W10	W10		
	–	W13	W13		
	–	W14	W14		
	–	W16	W16		
	–	W17	W17		
	–	W19	W19		
	–	AB5	AB5		
	–	AB9	AB9		
	–	AB14	AB14		
	–	AB18	AB18		
CV_{DD}	A1	E7	E7	S	1.5-V supply voltage (core) (’C6202B, ’C6203, and ’C6204 only) 1.8-V supply voltage (core) (’C6202 only)
	A2	E8	E8		
	A3	E10	E10		
	A24	E11	E11		
	A25	E12	E12		
	A26	E13	E13		
	B1	E15	E15		
	B2	E16	E16		
	B3	F7	–		
	B24	F8	–		
	B25	F9	–		
	B26	F11	–		
	C1	F12	–		
	C2	F14	–		
	C3	F15	–		
	C4	F16	–		
	C23	G5	G5		
	C24	G6	–		
	C25	G17	–		
	C26	G18	G18		
	D3	H5	H5		
	D4	H6	–		
	D5	H17	–		
	D22	H18	H18		
	D23	J6	–		
	D24	J17	–		
	E4	K5	K5		
	E23	K18	K18		
	AB4	L5	L5		

† The GLW BGA package (’C6204 only) is a subset of the GLS package (’C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
SUPPLY VOLTAGE PINS (CONTINUED)					
CV_{DD}	AB23	L6	–	S	1.5-V supply voltage (core) ('C6202B, 'C6203, and 'C6204 only) 1.8-V supply voltage (core) ('C6202 only)
	AC3	L17	–		
	AC4	L18	L18		
	AC5	M5	M5		
	AC22	M6	–		
	AC23	M17	–		
	AC24	M18	M18		
	AD1	N5	N5		
	AD2	N18	N18		
	AD3	P6	–		
	AD4	P17	–		
	AD23	R5	R5		
	AD24	R6	–		
	AD25	R17	–		
	AD26	R18	R18		
	AE1	T5	T5		
	AE2	T6	–		
	AE3	T17	–		
	AE24	T18	T18		
	AE25	U7	–		
	AE26	U8	–		
	AF1	U9	–		
	AF2	U11	–		
	AF3	U12	–		
	AF24	U14	–		
	AF25	U15	–		
	AF26	U16	–		
	–	V7	V7		
	–	V8	V8		
	–	V10	V10		
	–	V11	V11		
	–	V12	V12		
	–	V13	V13		
	–	V15	V15		
	–	V16	V16		
GROUND PINS					
V_{SS}	A4	A1	A1	GND	Ground pins
	A8	A5	A5		
	A13	A12	A12		
	A14	A18	A18		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
GROUND PINS (CONTINUED)					
V_{SS}	A15	A22	A22	GND	Ground pins
	A19	B2	B2		
	A23	B21	B21		
	B4	C1	C1		
	B12	C3	C3		
	B13	C20	C20		
	B14	C22	C22		
	B23	D5	D5		
	C5	D8	D8		
	C11§	D11	D11		
	C16	D12	D12		
	C22	D15	D15		
	D1	D18	D18		
	D2	E4	E4		
	D6	E5	E5		
	D21	E6	E6		
	D25	E9	E9		
	D26	E14	E14		
	E3	E17	E17		
	E24	E18	E18		
	F4	E19	E19		
	F23	F5	F5		
	H1	F6	–		
	H26	F10	–		
	K1	F13	–		
	K26	F17	–		
	M1	F18	F18		
	M26	H4	H4		
	N1	H19	H19		
	N2	J1	J1		
	N25	J5	J5		
	N26	J18	J18		
	P1	J22	J22		
	P2	K6	–		
	P25	K17	–		
	P26	L4	L4		
	R1	L19	L19		
	R26	M4	M4		
	U1	M19	M19		
	U26	N6	–		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground
§ For the 'C6202 GJL package only, the C11 pin is ground (V_{SS}). For all other 'C62x GJL packages, the C11 pin is CLKMODE1.

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO.			TYPE‡	DESCRIPTION
	GJL	GLS	GLW†		
GROUND PINS (CONTINUED)					
V_{SS}	W1	N17	–	GND	Ground pins
	W26	P1	P1		
	AA4	P5	P5		
	AA23	P18	P18		
	AB3	P22	P22		
	AB24	R4	R4		
	AC1	R19	R19		
	AC2	U5	U5		
	AC6	U6	–		
	AC21	U10	–		
	AC25	U13	–		
	AC26	U17	–		
	AD5	U18	U18		
	AD22	V4	V4		
	AE4	V5	V5		
	AE13	V6	V6		
	AE14	V9	V9		
	AE23	V14	V14		
	AF4	V17	V17		
	AF8	V18	V18		
	AF10	V19	V19		
	AF12	W5	W5		
	AF13	W8	W8		
	AF14	W11	W11		
	AF15	W12	W12		
	AF17	W15	W15		
	AF19	W18	W18		
	AF23	Y1	Y1		
	–	Y3	Y3		
	–	Y20	Y20		
	–	Y22	Y22		
	–	AA2	AA2		
	–	AA21	AA21		
	–	AB1	AB1		
	–	AB3	AB3		
	–	AB7	AB7		
	–	AB11	AB11		
	–	AB16	AB16		
	–	AB20	AB20		
	–	AB22	AB22		

† The GLW BGA package ('C6204 only) is a subset of the GLS package ('C6202/02B/03), with the inner row of core supply voltage (CV_{DD}) and ground (V_{SS}) pins removed (see the GLS/GLW BGA package bottom view).
‡ I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

development support

TI offers an extensive line of development tools for the TMS320C6000™ generation of DSPs, 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-based applications:

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

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

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

See Table 2 for a complete listing of development-support tools for the TMS320C6000 DSP family. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

development support (continued)

Table 2. TMS320C6000 Development-Support Tools

TOOL PART NUMBER	DESCRIPTION	DSP/ BIOS	CODE COMPOSER STUDIO™ IDE	CODE GENERATION TOOLS	EMULATION DRIVERS	RTDX	SIMULATOR	TARGET HARDWARE
TMDX320DAIS-07	TMS320™ DSP Algorithm Standard Developer's Kit							
SOFTWARE TOOLS								
6CCSFreeTool	TMS320C6000™ Code Composer Studio™ Free Evaluation Tools **(FREE 30-Day Trial)**†	√	√	√		√	√	
TMDX324685C-07 (Windows™ 95/98 Windows NT™)	TMS320C6000 DSP Code Composer Studio™ IDE	√	√	√	√	√	√	
TMDX3246855-07 (Windows 95/98/NT)	TMS320C6000 DSP Code Composer Studio™ IDE Compile Tools	√	√	√			√	
TMDX3240160-07 (Windows 95/98/NT)	TMS320C6000 DSP Code Composer Studio™ IDE Debug Tools	√	√		√	√		
HARDWARE TOOLS								
TMDX320006211 (DSK)	TMS320C6211 DSP Starter Kit (DSK) **256KB Code Memory Limit**	√	√	√	DSK-Specific	√		C6211 DSP
TMDS3260A6201	TMS320C62x™ DSP Evaluation Module (EVM)	√	√		EVM-Specific	√		C6201 DSP
TMDS326006201	TMS320C62x DSP EVM Bundle	√	√	√	EVM-Specific	√	√	C6201 DSP
TMDX3260A6701	TMS320C67x™ DSP EVM	√	√		EVM-Specific	√		C6701 DSP
TMDX326006701	TMS320C67x DSP EVM Bundle	√	√	√	EVM-Specific	√	√	C6701 DSP
TMDS00510	XDS510™ DSP Emulation Hardware							Any C6000 DSP via JTAG

† The TMS320C6000 Code Composer Studio Free Evaluation Tools can be downloaded for a free 30-day trial from the Texas Instruments web site at http://www.ti.com. A CD-ROM version of the TMS320C6000 Code Composer Studio Free Evaluation Tools (literature number SPRC020) is also available. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

PRODUCT PREVIEW

device and development-support tool nomenclature

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

Device development evolutionary flow:

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

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

TMS Fully qualified production device

Support tool development evolutionary flow:

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

TMDS Fully qualified development-support product

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

"Developmental product is intended for internal evaluation purposes."

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

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

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, GJL), the temperature range (for example, blank is the default commercial temperature range), and the device speed range in megahertz (for example, -300 is 300 MHz). Figure 4 provides a legend for reading the complete device name for any TMS320 family member.

device and development-support tool nomenclature (continued)

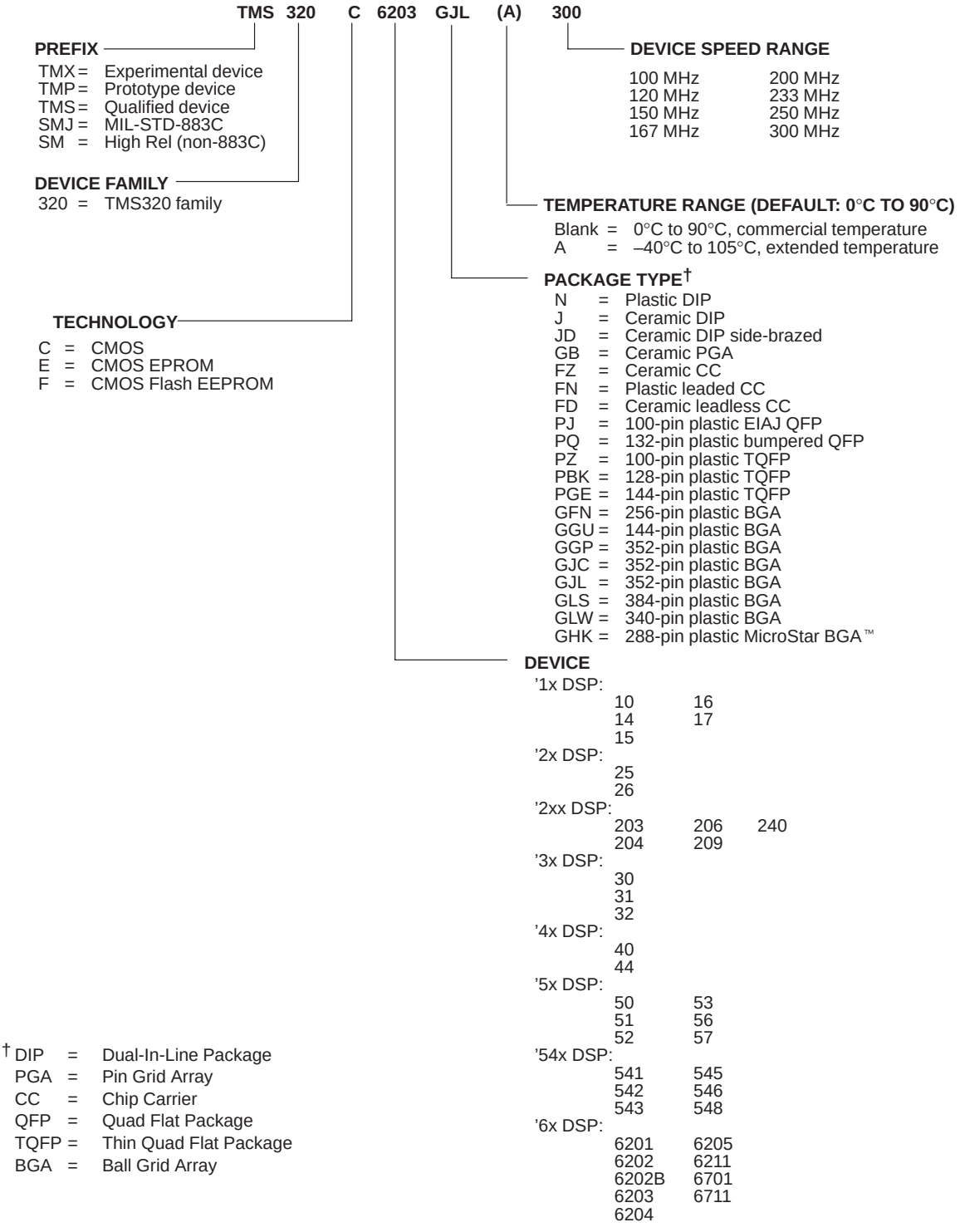


Figure 4. TMS320 Device Nomenclature (Including TMS320C6202, 'C6202B, 'C6203, and 'C6204)

PRODUCT PREVIEW

MicroStar BGA is a trademark of Texas Instruments.

documentation support

Extensive documentation supports all TMS320 family generations of devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's reference guides for all devices and tools; technical briefs; development-support tools; on-line help; and hardware and software applications. The following is a brief, descriptive list of support documentation specific to the 'C6x devices:

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

The *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190) describes the functionality of the peripherals available on 'C6x devices, such as the external memory interface (EMIF), host-port interface (HPI), multichannel buffered serial ports (McBSPs), direct-memory-access (DMA), enhanced direct-memory-access (EDMA) controller, expansion bus (XB), clocking and phase-locked loop (PLL); and power-down modes. This guide also includes information on internal data and program memories.

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

The *How to Begin Development and Migrate Across the TMS320C6202/6202B/6203/6204 DSPs* application report (literature number SPRA603) describes the migration concerns and identifies the similarites and differences between the 'C6202, 'C6202B, 'C6203, and 'C6204 'C6000 DSP devices.

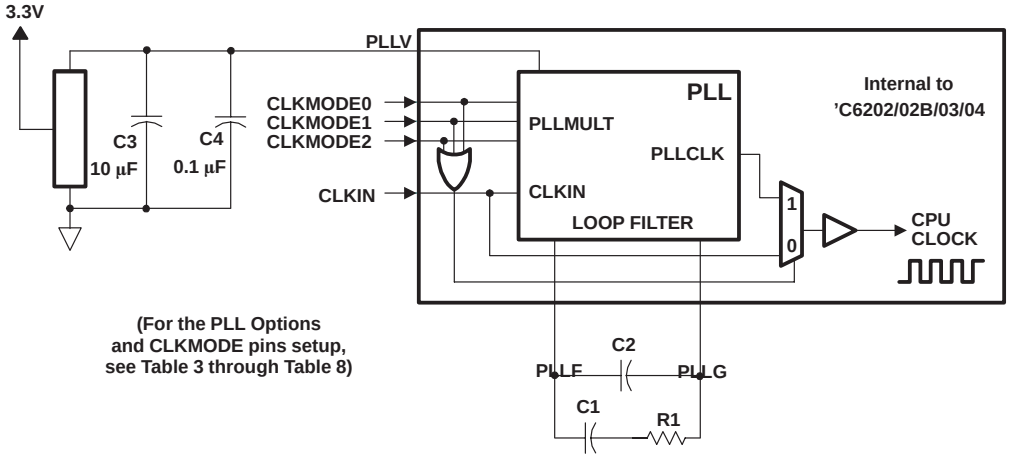
The tools support documentation is electronically available within the Code Composer Studio™ IDE. For a complete listing of 'C6000 latest documentation, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

clock PLL

All of the internal ’C62x clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock in frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock.

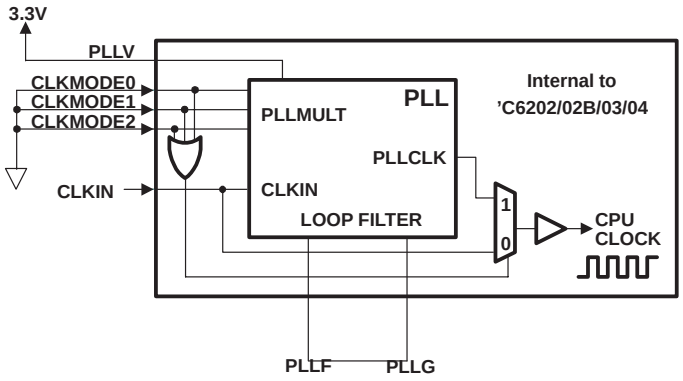
To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 5, and Table 3 through Table 8 show the external PLL circuitry for either x1 (PLL bypass) or x4 PLL multiply modes. Figure 6 shows the external PLL circuitry for a system with ONLY x1 (PLL bypass) mode.

To minimize the clock jitter, a single clean power supply should power both the ’C62x device and the external clock oscillator circuit. Noise coupling into PLLF directly impacts PLL clock jitter. The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see the *input and output clocks* electricals section.



NOTES: A. Keep the lead length and the number of vias between pin PLLF, pin PLLG, R1, C1, and C2 to a minimum. In addition, place all PLL components (R1, C1, C2, C3, C4, and EMI Filter) as close to the ’C6000 device as possible. Best performance is achieved with PLL components on single side of the board without jumpers, switches, or components other than the ones shown.
B. For reduced PLL jitter, maximize the spacing between switching signals and the PLL external components (R1, C1, C2, C3, C4, and the EMI Filter).
C. The 3.3-V supply for the EMI filter must be from the same 3.3-V power plane supplying the I/O voltage, DV_{DD}.
D. EMI filter manufacturer: TDK part number ACF451832-333, 223, 153, 103. Panasonic part number EXCCET103U.

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



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

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

PRODUCT PREVIEW

clock PLL (continued)

Table 3. TMS320C6202/'02B/'03/'04 GLS/GLW Packages PLL Multiply and Bypass (x1) Options†

GLS PACKAGE – 18 x 18 mm BGA ('C6202/'02B/'03 only) GLW PACKAGE – 18 x 18 mm BGA ('C6204 only)					
BIT (PIN NO.)	CLKMODE2 (A14)	CLKMODE1 (A9)	CLKMODE0 (B12)	DEVICES AND PLL CLOCK OPTIONS	
				'C6202, 'C6204‡	'C6202B, 'C6203
Value	0	0	0	Bypass (x1)	Bypass (x1)
	0	0	1	x4	x4
	0	1	0	Bypass (x1)	x8
	0	1	1	x4	x10
	1	0	0	Bypass (x1)	x6
	1	0	1	x4	x9
	1	1	0	Bypass (x1)	x7
	1	1	1	x4	x11

† f(CPU Clock) = f(CLKIN) x (PLL mode)
‡ For the 'C6202 GLS and 'C6204 GLW packages, the CLKMODE2 (A14) and CLKMODE1 (A9) pins are internally unconnected.

Table 4. TMS320C6202/'02B/'03 GJL Package PLL Multiply and Bypass (x1) Options†§

GJL PACKAGE 27 x 27 mm BGA					
BIT (PIN NO.)	CLKMODE2 (N/A)¶#	CLKMODE1 (C11)§¶	CLKMODE0 (B15)	DEVICES AND PLL CLOCK OPTIONS	
				'C6202¶	'C6202B, 'C6203¶
Value	N/A#	0	0	Bypass (x1)	Bypass (x1)
		0	1	x4	x4
		1	0	N/A CLKMODE1 pin (C11) Must Be Grounded§#	x8
		1	1		x10

† f(CPU Clock) = f(CLKIN) x (PLL mode)
§ Note: The C11 pin is CLKMODE1 on the 'C6202B/'03 GJL package and a ground pin (V_{SS}) for the 'C6202 GJL package. If a 'C6202 GJL package is placed in a 'C6202B/'03 GJL board with the CLKMODE1 pin pulled to the non-default state (default is GND), current is drawn through the pullup (3.3 V/ 20 kΩ or 165 μA). **If a 'C6202 GJL package is placed in a 'C6202B/'03 board with the C11 pin directly connected to the V_{CC} plane for the PLL mode, a ground/power is shorted through the package.** For more detailed information on device compatibility, see the *How to Begin Development and Migrate Across the TMS320C6202/6202B/6203/6204 DSPs* application report (literature number SPRA603).
¶ CLKMODE2 and CLKMODE1 pins are *not* available on the 'C6202 GJL package.
The CLKMODE2 pin is *not* available on the 'C6202B/'C6203 GJL package.
N/A = Not Applicable

Table 5. TMS320C6202 PLL Component Selection Table‖

CLKMODE	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY (CLKOUT1) RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 (Ω)	C1 (nF)	C2 (pF)	TYPICAL LOCK TIME (μs)
x4	32.5–62.5	130–250	65–125	60.4	27	560	75

‖ Under some operating conditions, the maximum PLL lock time may vary 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.

clock PLL (continued)

Table 6. TMS320C6202B PLL Component Selection Table†

CLKMODE‡	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 (Ω)	C1 (nF)	C2 (pF)	TYPICAL LOCK TIME (μs)
x4	32.5–62.5	130–250	65–125	60.4	27	560	75
x6	21.7–41.7						
x7	18.6–35.7						
x8	16.3–31.3						
x9	14.4–27.8						
x10	13–25						
x11	11.8–22.7						

† Under some operating conditions, the maximum PLL lock time may vary 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.
‡ CLKMODE x1, x4, x6, x7, x8, x9, x10, and x11 apply to the GLS device. The GJL device is restricted to x1, x4, x8, and x10 multiply factors.

Table 7. TMS320C6203 PLL Component Selection Table†

CLKMODE‡	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 (Ω)	C1 (nF)	C2 (pF)	TYPICAL LOCK TIME (μs)
x4	32.5–75	130–300	65–150	60.4	27	560	75
x6	21.7–50						
x7	18.6–42.9						
x8	16.3–37.5						
x9	14.4–33.3						
x10	13–30						
x11	11.8–27.3						

† Under some operating conditions, the maximum PLL lock time may vary 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.
‡ CLKMODE x1, x4, x6, x7, x8, x9, x10, and x11 apply to the GLS device. The GJL device is restricted to x1, x4, x8, and x10 multiply factors.

Table 8. TMS320C6204 PLL Component Selection Table†

CLKMODE	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 (Ω)	C1 (nF)	C2 (pF)	TYPICAL LOCK TIME (μs)
x4	32.5–50	130–200	65–100	60.4	27	560	75

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

power-supply sequencing

For ’C6202B, ’C6203, and ’C6204 devices only, the 1.5-V supply powers the core and the 3.3-V supply powers the I/O buffers. For the ’C6202 device only, the 1.8-V supply powers the core and the 3.3-V supply powers the I/O buffers. For internal device reliability, there are no specific sequencing requirements between the core supply and the I/O supply. The only constraint is that neither supply should be powered on for extended periods of time if the other supply is below the valid operating voltage.

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

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

Supply voltage range, CV_{DD} (see Note 1) . . . . . – 0.3 V to 2.3 V
Supply voltage range, DV_{DD} (see Note 1) . . . . . –0.3 V to 4 V
Input voltage range . . . . . –0.3 V to 4 V
Output voltage range . . . . . –0.3 V to 4 V
Operating case temperature range, T_C: (default) . . . . . 0°C to 90°C
(A version) . . . . . –40°C to105°C
Storage temperature range, T_{stg} . . . . . –55°C to 150°C
Temperature cycle range, (1000-cycle performance) . . . . . –40°C to 125°C

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

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

recommended operating conditions

			MIN	NOM	MAX	UNIT
CV_{DD}	Supply voltage (CORE)	’C6202B, ’C6203, and ’C6204 only	1.425	1.5	1.575	V
		’C6202 only	1.71	1.8	1.89	
DV_{DD}	Supply voltage (I/O)		3.14	3.30	3.46	V
V_{SS}	Supply ground		0	0	0	V
V_{IH}	High-level input voltage		2.0			V
V_{IL}	Low-level input voltage				0.8	V
I_{OH}	High-level output current				–8	mA
I_{OL}	Low-level output current				8	mA
T_C	Operating case temperature		0		90	°C

PRODUCT PREVIEW

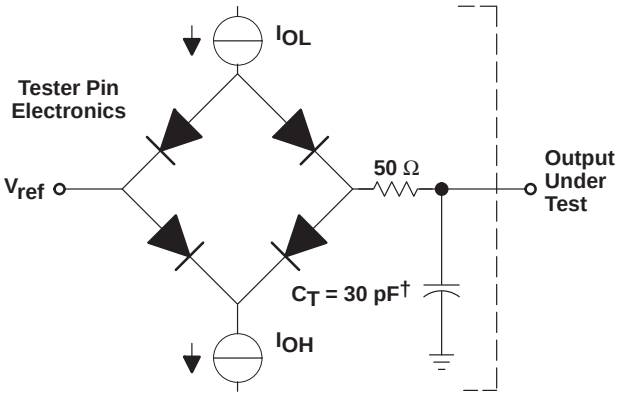
electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	DV_{DD} = MIN, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN, I_{OL} = MAX			0.6	V
I_I	Input current†	$V_I = V_{SS}$ to DV_{DD}			±10	uA
I_{OZ}	Off-state output current	$V_O = DV_{DD}$ or 0 V			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access‡	’C6202, CV_{DD} = NOM, CPU clock = 200 MHz		520		mA
		’C6202B, CV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
		’C6203, CV_{DD} = NOM, CPU clock = 200 MHz				mA
		’C6204, CV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
I_{DD2V}	Supply current, peripherals‡	’C6202, CV_{DD} = NOM, CPU clock = 200 MHz		390		mA
		’C6202B, CV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
		’C6203, CV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
		’C6204, CV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
I_{DD3V}	Supply current, I/O pins‡	’C6202, DV_{DD} = NOM, CPU clock = 200 MHz		70		mA
		’C6202B, DV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
		’C6203, DV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
		’C6204, DV_{DD} = NOM, CPU clock = 200 MHz		TBD		mA
C_i	Input capacitance				10	pF
C_o	Output capacitance				10	pF

† TMS and TDI are not included due to internal pullups. $\overline{TRST}$ is not included due to internal pulldown.

‡ Measured with average activity (50% high / 50% low power). For more detailed information on CPU/peripheral/I/O activity, see the *TMS320C6000 Power Consumption Summary* application report (literature number SPRA486).

PARAMETER MEASUREMENT INFORMATION



† Typical distributed load circuit capacitance

Figure 7. Test Load Circuit

signal transition levels

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

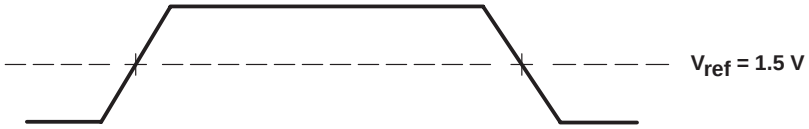


Figure 8. Input and Output Voltage Reference Levels for ac Timing Measurements

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN (PLL used)†‡§ (see Figure 9)

NO.			-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	5 * M		4 * M		3.33 * M		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.4C		0.4C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.4C		0.4C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		5		5	ns

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of V_{IH}.
‡ M = the PLL multiplier factor (x4, x6, x7, x8, x9, x10, or x11) For more detail, see the clock PLL section.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 10 MHz, use C = 100 ns.

timing requirements for CLKIN [PLL bypassed (x1)]†§ (see Figure 9)

NO.			-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	5		4		3.33		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.45C		0.45C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.45C		0.45C		0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		0.6		0.6		0.6	ns

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of V_{IH}.
§ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 10 MHz, use C = 100 ns.

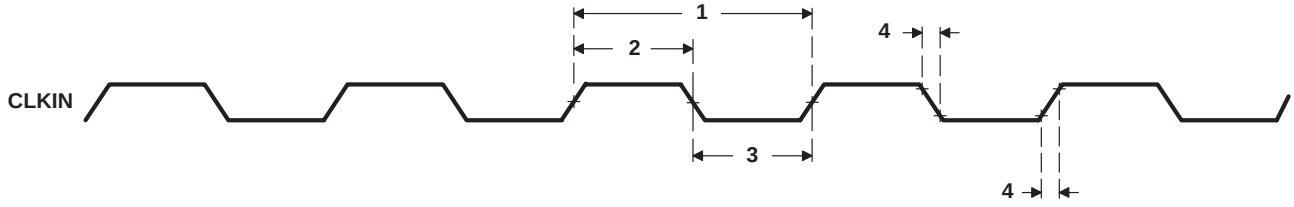


Figure 9. CLKIN Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

timing requirements for XCLKIN† (see Figure 10)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{c(XCLKIN)}$	Cycle time, XCLKIN	4P		ns
2	$t_{w(XCLKINH)}$	Pulse duration, XCLKIN high	1.8P		ns
3	$t_{w(XCLKINL)}$	Pulse duration, XCLKIN low	1.8P		ns

† P = 1/CPU clock frequency in ns.

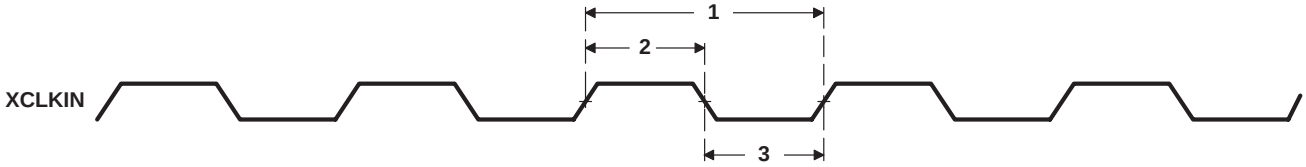


Figure 10. XCLKIN Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT2† (see Figure 11)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	2P – 0.7	2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	P – 0.7	P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	P – 0.7	P + 0.7	ns

† P = 1/CPU clock frequency in ns.

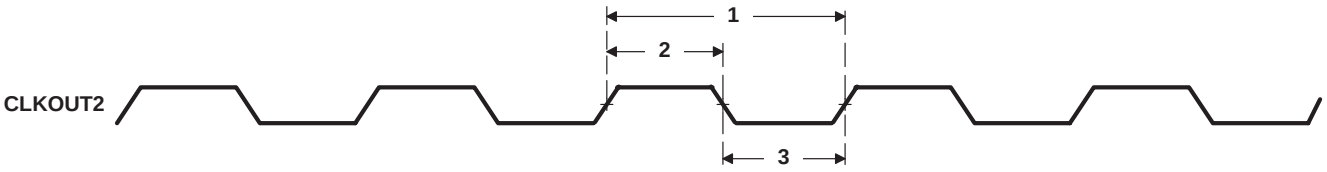


Figure 11. CLKOUT2 Timings

switching characteristics for XFCLK†‡ (see Figure 12)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{c(XFCK)}$	Cycle time, XFCLK	D * P – 0.7	D * P + 0.7	ns
2	$t_{w(XFCKH)}$	Pulse duration, XFCLK high	(D/2) * P – 0.7	(D/2) * P + 0.7	ns
3	$t_{w(XFCKL)}$	Pulse duration, XFCLK low	(D/2) * P – 0.7	(D/2) * P + 0.7	ns

† P = 1/CPU clock frequency in ns.
‡ D = 8, 6, 4, or 2; FIFO clock divide ratio, user-programmable

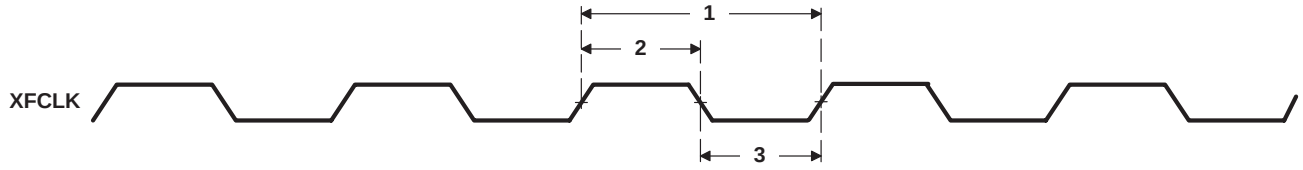


Figure 12. XFCLK Timings

PRODUCT PREVIEW

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles[†‡§¶] (see Figure 13 – Figure 16)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{ARE}$ high	1		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{ARE}$ high	3.5		ns
6	$t_{su(ARDYH-AREL)}$	Setup time, ARDY high before $\overline{ARE}$ low	–[(RST – 3) * P – 6]		ns
7	$t_{h(AREL-ARDYH)}$	Hold time, ARDY high after $\overline{ARE}$ low	(RST – 3) * P + 2		ns
9	$t_{su(ARDYL-AREL)}$	Setup time, ARDY low before $\overline{ARE}$ low	–[(RST – 3) * P – 6]		ns
10	$t_{h(AREL-ARDYL)}$	Hold time, ARDY low after $\overline{ARE}$ low	(RST – 3) * P + 2		ns
11	$t_{w(ARDYH)}$	Pulse width, ARDY high	2P		ns
15	$t_{su(ARDYH-AWEL)}$	Setup time, ARDY high before $\overline{AWE}$ low	–[(WST – 3) * P – 6]		ns
16	$t_{h(AWEL-ARDYH)}$	Hold time, ARDY high after $\overline{AWE}$ low	(WST – 3) * P + 2		ns
18	$t_{su(ARDYL-AWEL)}$	Setup time, ARDY low before $\overline{AWE}$ low	–[(WST – 3) * P – 6]		ns
19	$t_{h(AWEL-ARDYL)}$	Hold time, ARDY low after $\overline{AWE}$ low	(WST – 3) * P + 2		ns

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. If ARDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, ARDY can be an asynchronous input.
‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the EMIF CE space control registers.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
¶ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use ARDY input to extend strobe width.

switching characteristics for asynchronous memory cycles[‡§¶#] (see Figure 13 – Figure 16)

NO.	PARAMETER		-200 -250 -300			UNIT
			MIN	TYP	MAX	
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{ARE}$ low	RS * P – 2			ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{ARE}$ high to select signals invalid	RH * P – 2			ns
5	$t_{w(AREL)}$	Pulse width, $\overline{ARE}$ low		RST * P		ns
8	$t_{d(ARDYH-AREH)}$	Delay time, ARDY high to $\overline{ARE}$ high	3P		4P + 5	ns
12	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{AWE}$ low	WS * P – 3			ns
13	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{AWE}$ high to select signals invalid	WH * P – 2			ns
14	$t_{w(AWEL)}$	Pulse width, $\overline{AWE}$ low		WST * P		ns
17	$t_{d(ARDYH-AWEH)}$	Delay time, ARDY high to $\overline{AWE}$ high	3P		4P + 5	ns

‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the EMIF CE space control registers.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
¶ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use ARDY input to extend strobe width.
# Select signals include: $\overline{CEx}$, $\overline{BE[3:0]}$, EA[21:2], $\overline{AOE}$; and for writes, include ED[31:0], with the exception that $\overline{CEx}$ can stay active for an additional 7P ns following the end of the cycle.

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

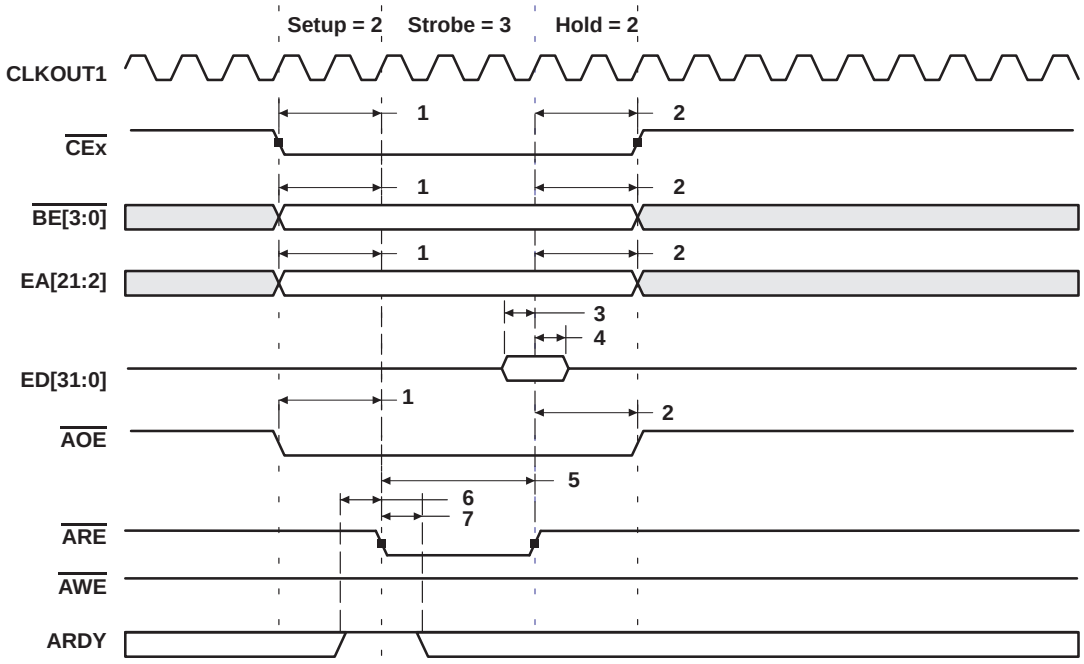


Figure 13. Asynchronous Memory Read Timing (ARDY Not Used)

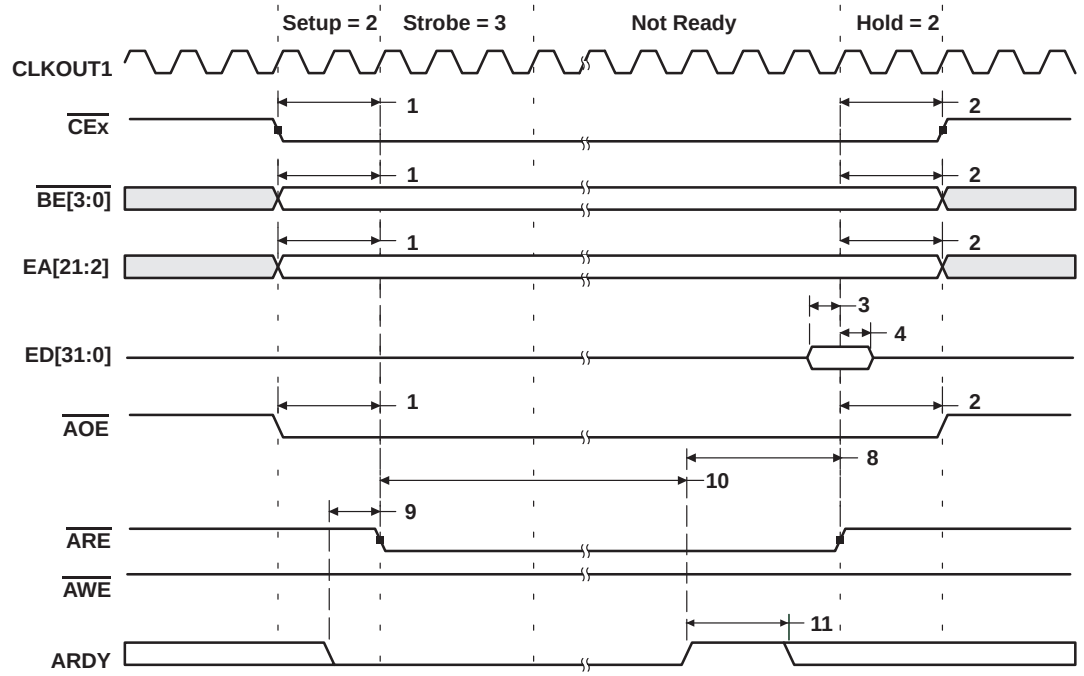


Figure 14. Asynchronous Memory Read Timing (ARDY Used)

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

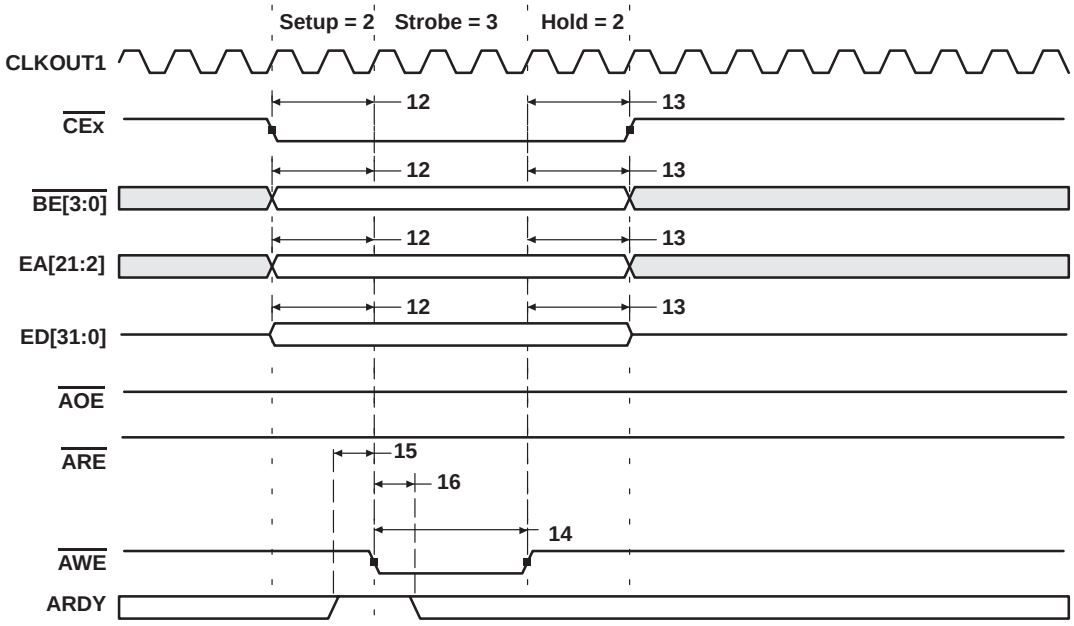


Figure 15. Asynchronous Memory Write Timing (ARDY Not Used)

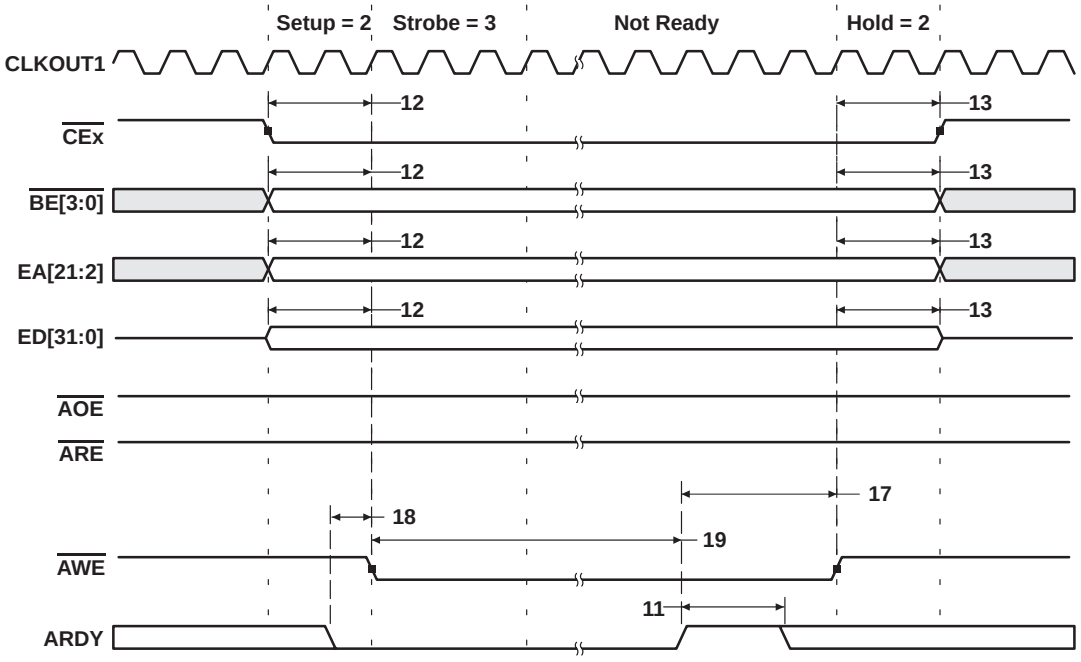


Figure 16. Asynchronous Memory Write Timing (ARDY Used)

SYNCHRONOUS-BURST MEMORY TIMING

timing requirements for synchronous-burst SRAM cycles (see Figure 17)

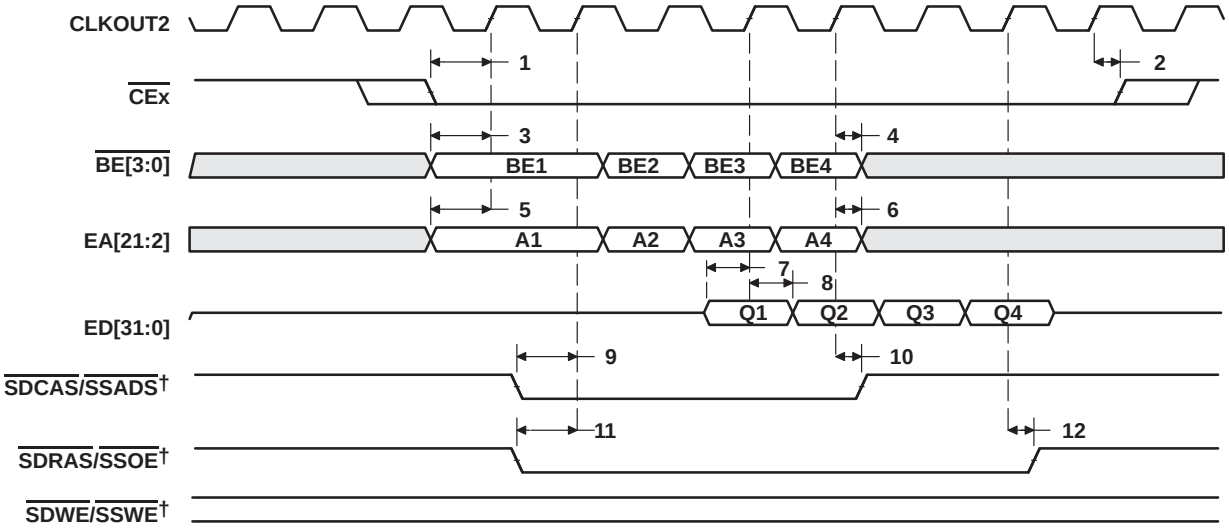
NO.			-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
7	$t_{su(EDV-CKO2H)}$	Setup time, read EDx valid before CLKOUT2 high	2.5		2.0		1.7		ns
8	$t_{h(CKO2H-EDV)}$	Hold time, read EDx valid after CLKOUT2 high	2.0		2.0		1.5		ns

switching characteristics for synchronous-burst SRAM cycles†‡ (see Figure 17 and Figure 18)

NO.	PARAMETER		-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{osu(CEV-CKO2H)}$	Output setup time, $\overline{CEx}$ valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
2	$t_{oh(CKO2H-CEV)}$	Output hold time, $\overline{CEx}$ valid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
3	$t_{osu(BEV-CKO2H)}$	Output setup time, $\overline{BEx}$ valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
4	$t_{oh(CKO2H-BEIV)}$	Output hold time, $\overline{BEx}$ invalid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
5	$t_{osu(EAV-CKO2H)}$	Output setup time, EAx valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
6	$t_{oh(CKO2H-EAIV)}$	Output hold time, EAx invalid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
9	$t_{osu(ADSV-CKO2H)}$	Output setup time, $\overline{SDCAS}/\overline{SSADS}$ valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
10	$t_{oh(CKO2H-ADSV)}$	Output hold time, $\overline{SDCAS}/\overline{SSADS}$ valid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
11	$t_{osu(OEV-CKO2H)}$	Output setup time, $\overline{SDRAS}/\overline{SSOE}$ valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
12	$t_{oh(CKO2H-OEV)}$	Output hold time, $\overline{SDRAS}/\overline{SSOE}$ valid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
13	$t_{osu(EDV-CKO2H)}$	Output setup time, EDx valid before CLKOUT2 high§	P – 1.2		P – 1.2		P + 0.1		ns
14	$t_{oh(CKO2H-EDIV)}$	Output hold time, EDx invalid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns
15	$t_{osu(WEV-CKO2H)}$	Output setup time, $\overline{SDWE}/\overline{SSWE}$ valid before CLKOUT2 high	P – 0.8		P – 0.8		P + 0.1		ns
16	$t_{oh(CKO2H-WEV)}$	Output hold time, $\overline{SDWE}/\overline{SSWE}$ valid after CLKOUT2 high	P – 4		P – 3		P – 2.3		ns

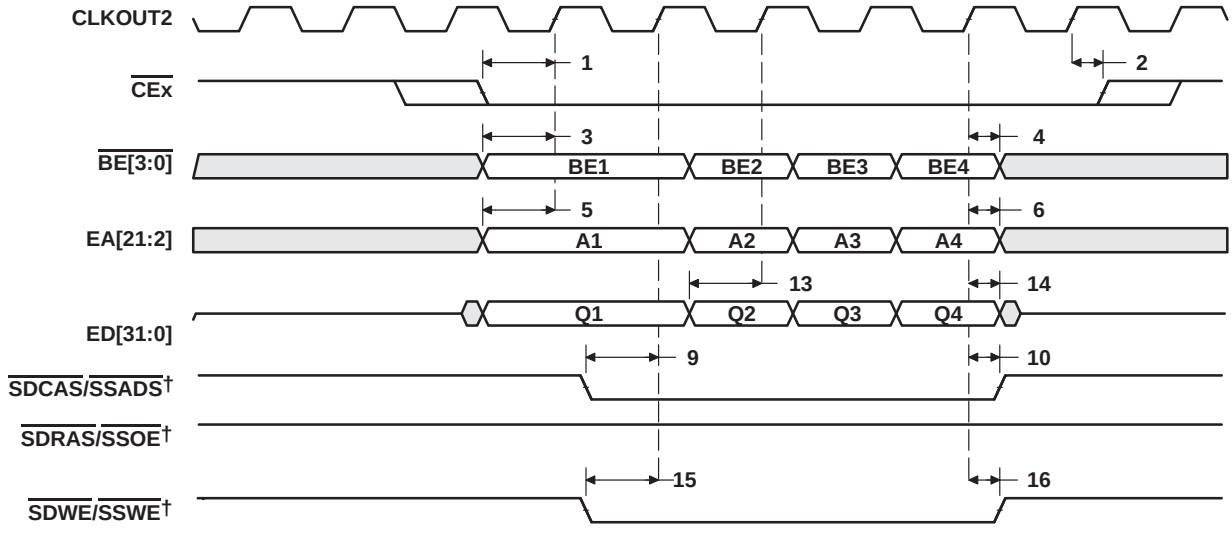
† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ $\overline{SDCAS}/\overline{SSADS}$, $\overline{SDRAS}/\overline{SSOE}$, and $\overline{SDWE}/\overline{SSWE}$ operate as $\overline{SSADS}$, $\overline{SSOE}$, and $\overline{SSWE}$, respectively, during SBSRAM accesses.
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)



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

Figure 17. SBSRAM Read Timing



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

Figure 18. SBSRAM Write Timing

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles (see Figure 19)

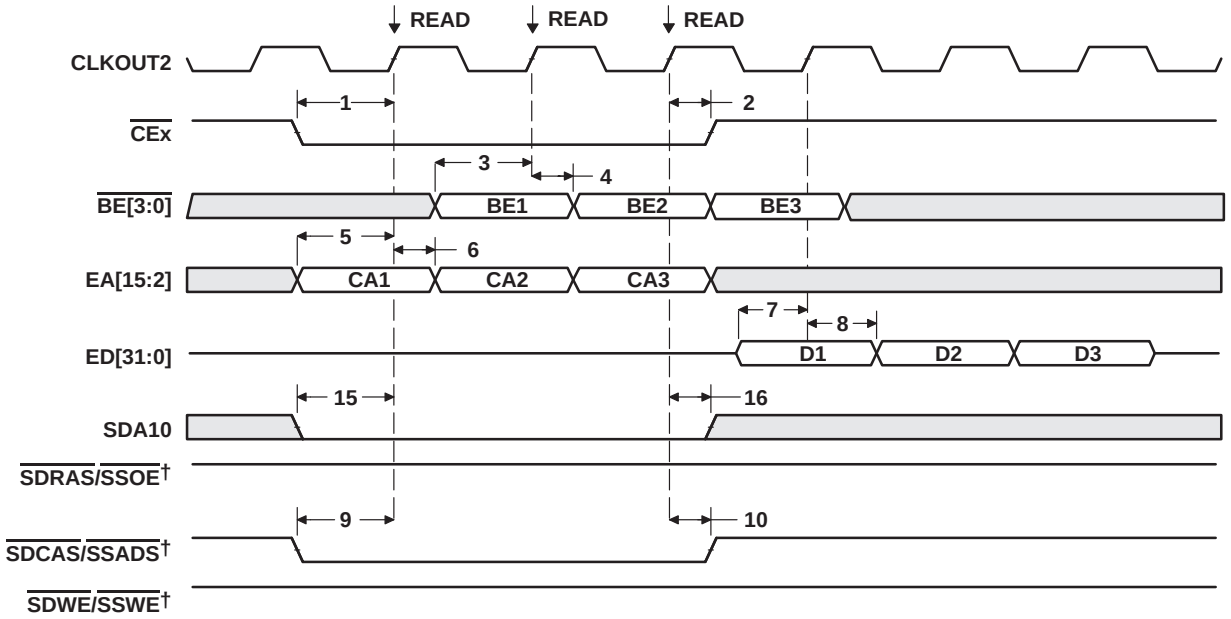
NO.			-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
7	$t_{su(EDV-CKO2H)}$	Setup time, read EDx valid before CLKOUT2 high	1.2		1.2		0.5		ns
8	$t_{h(CKO2H-EDV)}$	Hold time, read EDx valid after CLKOUT2 high	3		2.7		2		ns

switching characteristics for synchronous DRAM cycles†‡ (see Figure 19–Figure 24)

NO.	PARAMETER		-200		-250		-300		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{osu(CEV-CKO2H)}$	Output setup time, $\overline{CEx}$ valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
2	$t_{oh(CKO2H-CEV)}$	Output hold time, $\overline{CEx}$ valid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
3	$t_{osu(BEV-CKO2H)}$	Output setup time, $\overline{BEx}$ valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
4	$t_{oh(CKO2H-BEIV)}$	Output hold time, $\overline{BEx}$ invalid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
5	$t_{osu(EAV-CKO2H)}$	Output setup time, EAx valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
6	$t_{oh(CKO2H-EAIV)}$	Output hold time, EAx invalid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
9	$t_{osu(CASV-CKO2H)}$	Output setup time, $\overline{SDCAS}$/$\overline{SSADS}$ valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
10	$t_{oh(CKO2H-CASV)}$	Output hold time, $\overline{SDCAS}$/$\overline{SSADS}$ valid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
11	$t_{osu(EDV-CKO2H)}$	Output setup time, EDx valid before CLKOUT2 high§	P – 1		P – 1.5		P + 0.6		ns
12	$t_{oh(CKO2H-EDIV)}$	Output hold time, EDx invalid after CLKOUT2 high	P – 3.5		P – 2.8		P – 1.8		ns
13	$t_{osu(WEV-CKO2H)}$	Output setup time, $\overline{SDWE}$/$\overline{SSWE}$ valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
14	$t_{oh(CKO2H-WEV)}$	Output hold time, $\overline{SDWE}$/$\overline{SSWE}$ valid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
15	$t_{osu(SDA10V-CKO2H)}$	Output setup time, SDA10 valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
16	$t_{oh(CKO2H-SDA10IV)}$	Output hold time, SDA10 invalid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns
17	$t_{osu(RASV-CKO2H)}$	Output setup time, $\overline{SDRAS}$/$\overline{SSOE}$ valid before CLKOUT2 high	P – 1		P – 0.9		P + 0.6		ns
18	$t_{oh(CKO2H-RASV)}$	Output hold time, $\overline{SDRAS}$/$\overline{SSOE}$ valid after CLKOUT2 high	P – 3.5		P – 2.9		P – 1.8		ns

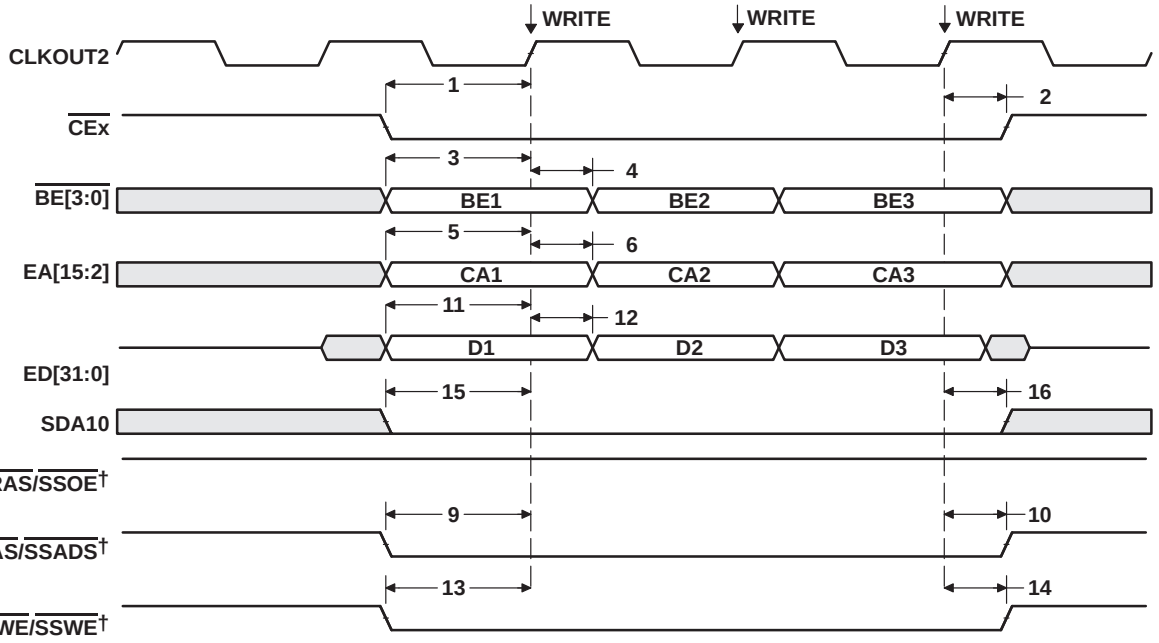
† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ $\overline{SDCAS}$/$\overline{SSADS}$, $\overline{SDRAS}$/$\overline{SSOE}$, and $\overline{SDWE}$/$\overline{SSWE}$ operate as $\overline{SDCAS}$, $\overline{SDRAS}$, and $\overline{SDWE}$, respectively, during SDRAM accesses.
§ For the first write in a series of one or more consecutive adjacent writes, the write data is generated one CLKOUT2 cycle early to accommodate the ED enable time.

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

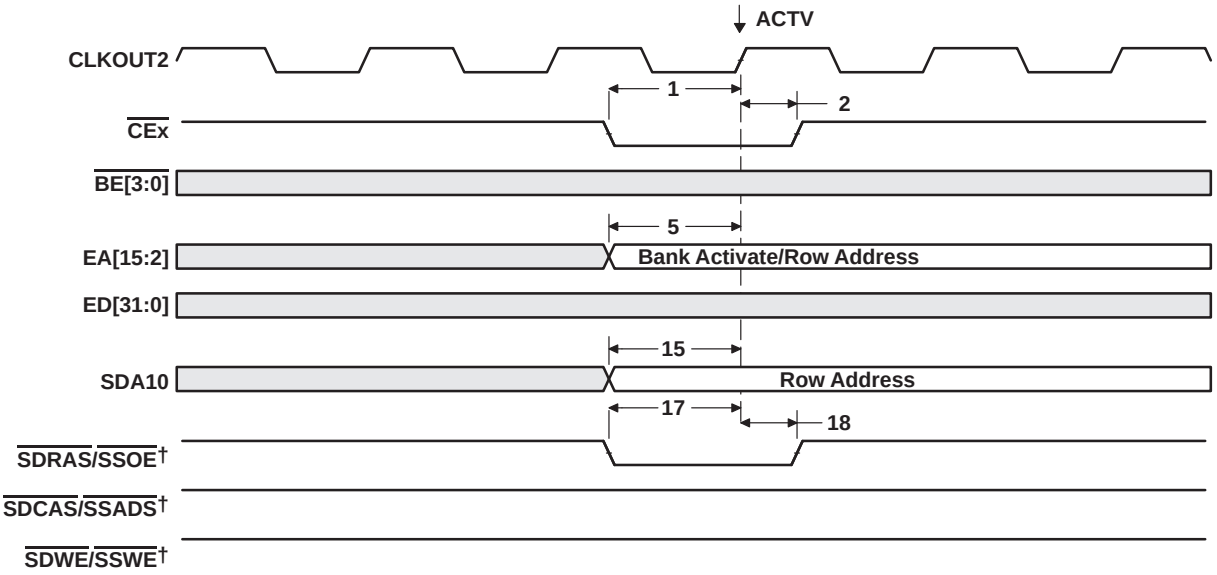
Figure 19. Three SDRAM READ Commands



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

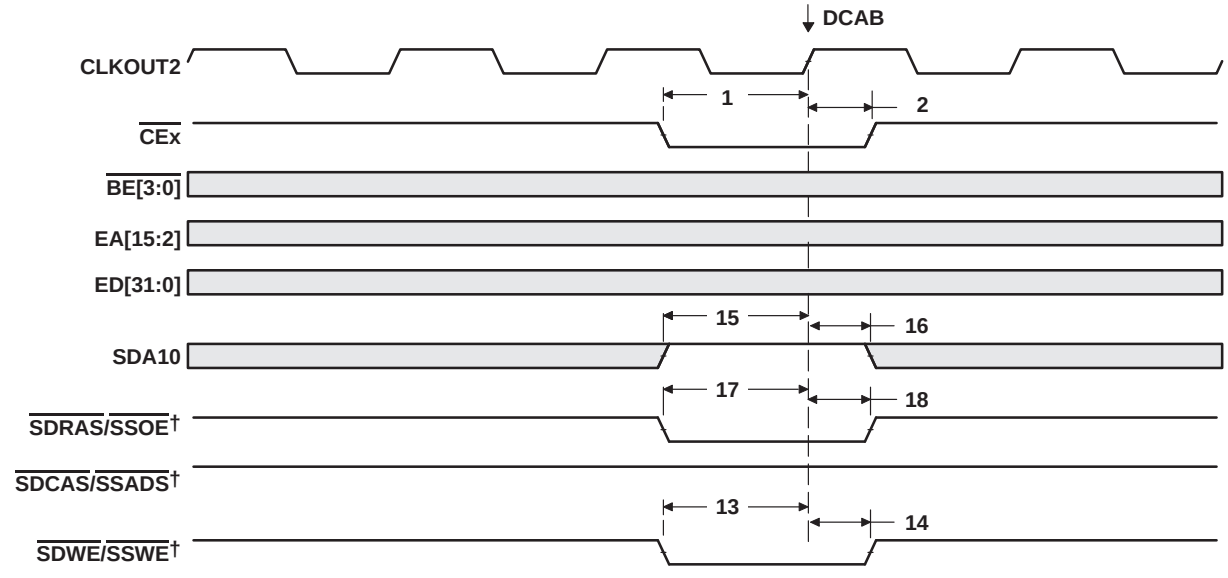
Figure 20. Three SDRAM WRT Commands

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

Figure 21. SDRAM ACTV Command

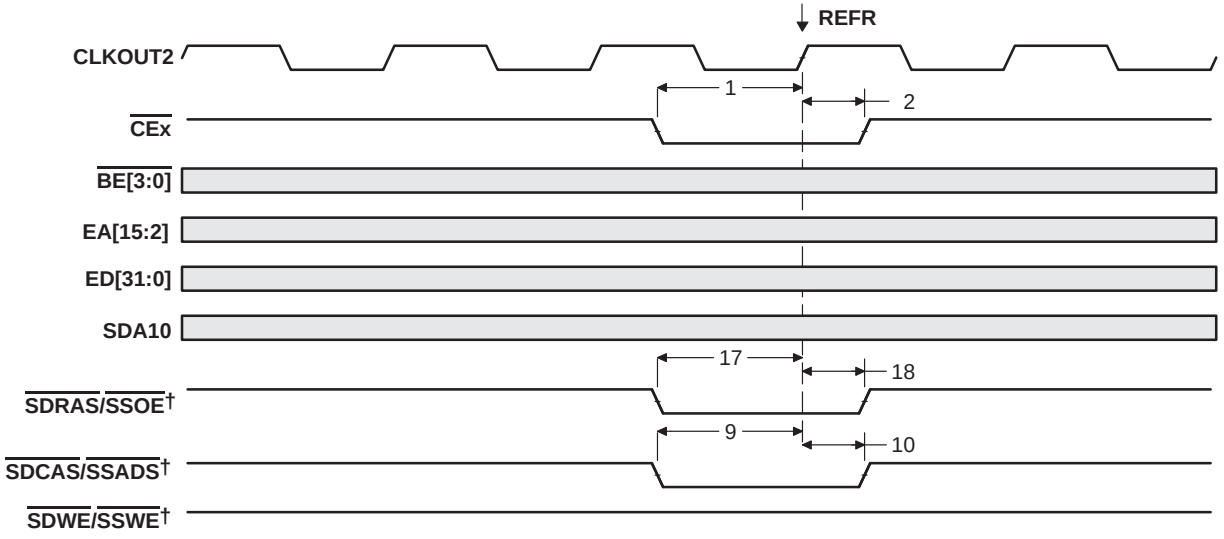


† SDCAS/SSADS, SDRAS/SSOE, and SDWE/SSWE operate as SDCAS, SDRAS, and SDWE, respectively, during SDRAM accesses.

Figure 22. SDRAM DCAB Command

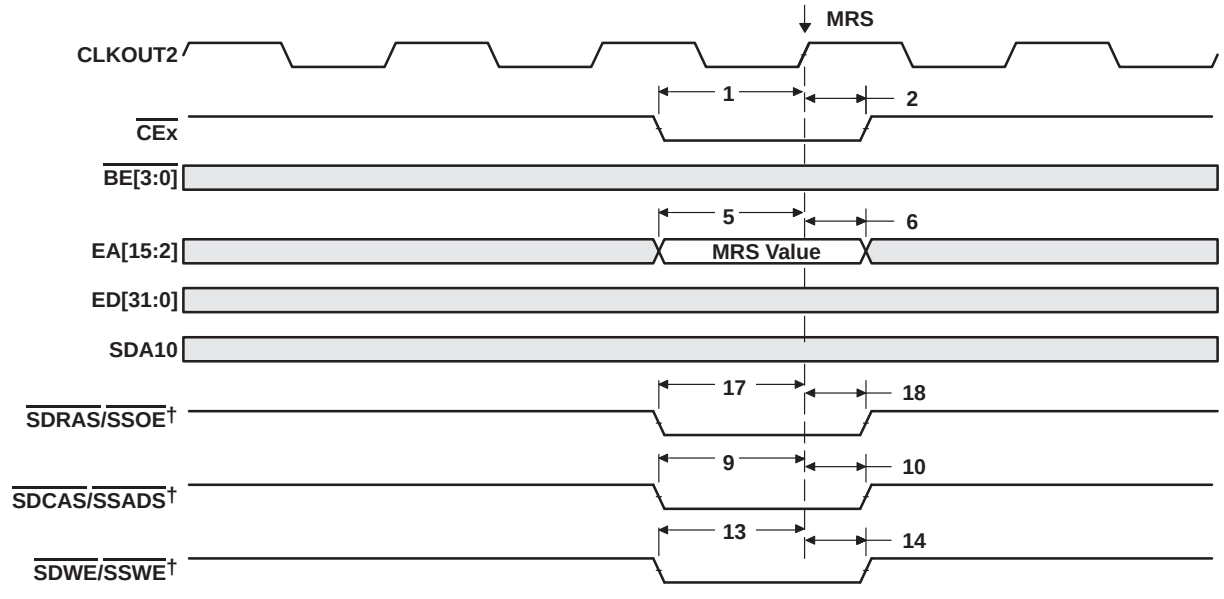
PRODUCT PREVIEW

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

Figure 23. SDRAM REFR Command



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

Figure 24. SDRAM MRS Command

HOLD/HOLDA TIMING

timing requirements for the HOLD/HOLDA cycles† (see Figure 25)

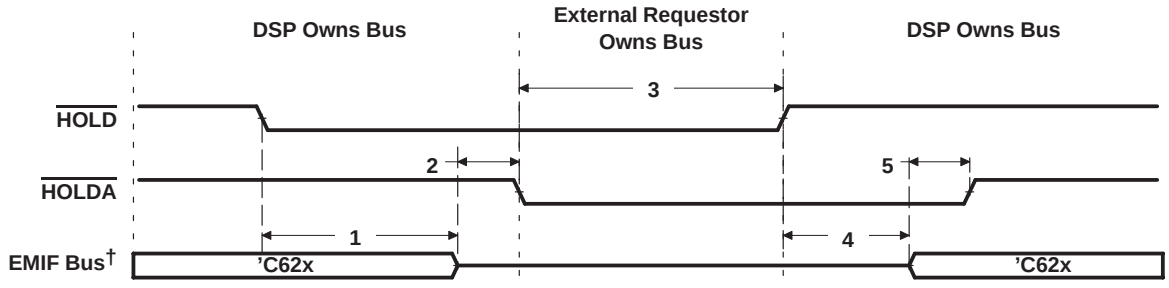
NO.		-200 -250 -300		UNIT
		MIN	MAX	
3	$t_{oh(HOLDAL-HOLDL)}$ Hold time, HOLD low after HOLDA low	P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

switching characteristics for the HOLD/HOLDA cycles†‡ (see Figure 25)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{R(HOLDL-EMHZ)}$	Response time, HOLD low to EMIF Bus high impedance	3P	§	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	0	2P	ns
4	$t_{R(HOLDH-EMLZ)}$	Response time, HOLD high to EMIF Bus low impedance	3P	7P	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	0	2P	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ EMIF Bus consists of CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE, AOE, AWE, SDCAS/SSADS, SDRAS/SSOE, SDWE/SSWE, and SDA10.
§ All pending EMIF transactions are allowed to complete before HOLDA is asserted. The worst case for this is an asynchronous read or write with external ARDY used or a minimum of eight consecutive SDRAM reads or writes when RBTR8 = 1. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.



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

Figure 25. HOLD/HOLDA Timing

RESET TIMING

timing requirements for reset† (see Figure 26)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{w(RST)}$	Width of the $\overline{\text{RESET}}$ pulse (PLL stable)‡	10P		ns
		Width of the $\overline{\text{RESET}}$ pulse (PLL needs to sync up)§	250		µs
10	$t_{su(XD)}$	Setup time, XD configuration bits valid before $\overline{\text{RESET}}$ high¶	5P		ns
11	$t_{h(XD)}$	Hold time, XD configuration bits valid after $\overline{\text{RESET}}$ high¶	5P		ns

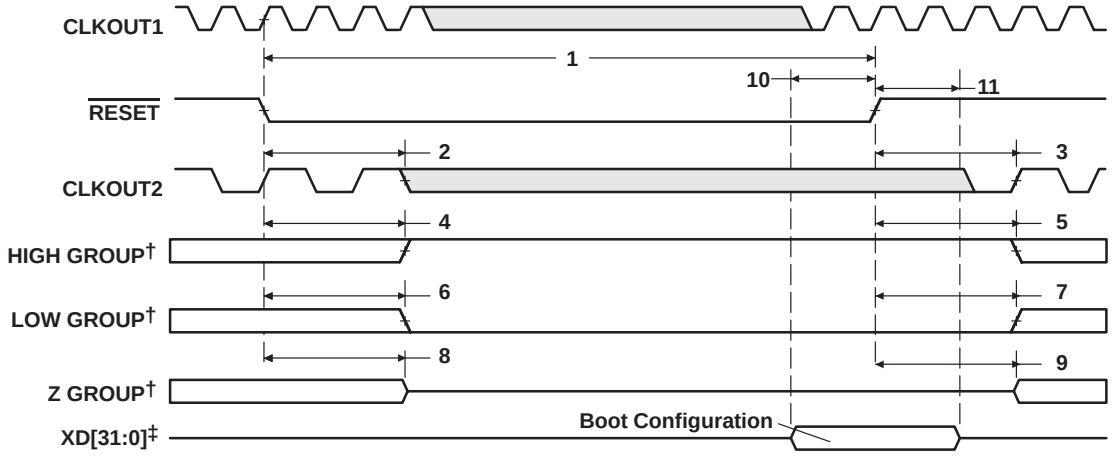
† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ This parameter applies to CLKMODE x1 when CLKIN is stable, and applies to CLKMODE x4, x6, x7, x8, x9, x10, and x11 when CLKIN and PLL are stable.
§ This parameter applies to CLKMODE x4, x6, x7, x8, x9, x10,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

switching characteristics during reset†# (see Figure 26)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
2	$t_{d(RSTL-CKO2IV)}$	Delay time, $\overline{\text{RESET}}$ low to CLKOUT2 invalid	P		ns
3	$t_{d(RSTH-CKO2V)}$	Delay time, $\overline{\text{RESET}}$ high to CLKOUT2 valid		4P	ns
4	$t_{d(RSTL-HIGHIV)}$	Delay time, $\overline{\text{RESET}}$ low to high group invalid	P		ns
5	$t_{d(RSTH-HIGHV)}$	Delay time, $\overline{\text{RESET}}$ high to high group valid		4P	ns
6	$t_{d(RSTL-LOWIV)}$	Delay time, $\overline{\text{RESET}}$ low to low group invalid	P		ns
7	$t_{d(RSTH-LOWV)}$	Delay time, $\overline{\text{RESET}}$ high to low group valid		4P	ns
8	$t_{d(RSTL-ZHZ)}$	Delay time, $\overline{\text{RESET}}$ low to Z group high impedance	P		ns
9	$t_{d(RSTH-ZV)}$	Delay time, $\overline{\text{RESET}}$ high to Z group valid		4P	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
# High group consists of: XFCLK, $\overline{\text{HOLDA}}$
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], $\overline{\text{CE}}$[3:0], $\overline{\text{BE}}$[3:0], $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{AOE}}$, $\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, $\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$, SDA10, CLKX0, CLKX1, CLKX2, FSX0, FSX1, FSX2, DX0, DX1, DX2, CLKR0, CLKR1, CLKR2, FSR0, FSR1, FSR2, $\overline{\text{XCE}}$[3:0], $\overline{\text{XBE}}$[3:0]/XA[5:2], $\overline{\text{XOE}}$, $\overline{\text{XRE}}$, $\overline{\text{XWE}}$/XWAIT, $\overline{\text{XAS}}$, XW/R, XRDY, XBLAST, XHOLD, and XHOLDA

RESET TIMING (CONTINUED)



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

‡ XD[31:0] are the boot configuration pins during device reset.

Figure 26. Reset Timing

EXTERNAL INTERRUPT TIMING

timing requirements for interrupt response cycles† (see Figure 27)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
2	$t_{w(ILOW)}$	Width of the interrupt pulse low	2P		ns
3	$t_{w(IHIGH)}$	Width of the interrupt pulse high	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

switching characteristics during interrupt response cycles† (see Figure 27)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{R(EINTH – IACKH)}$	Response time, EXT_INTx high to IACK high	9P		ns
4	$t_{d(CKO2L-IACKV)}$	Delay time, CLKOUT2 low to IACK valid	0	10	ns
5	$t_{d(CKO2L-INUMV)}$	Delay time, CLKOUT2 low to INUMx valid	0	10	ns
6	$t_{d(CKO2L-INUMIV)}$	Delay time, CLKOUT2 low to INUMx invalid	0	10	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

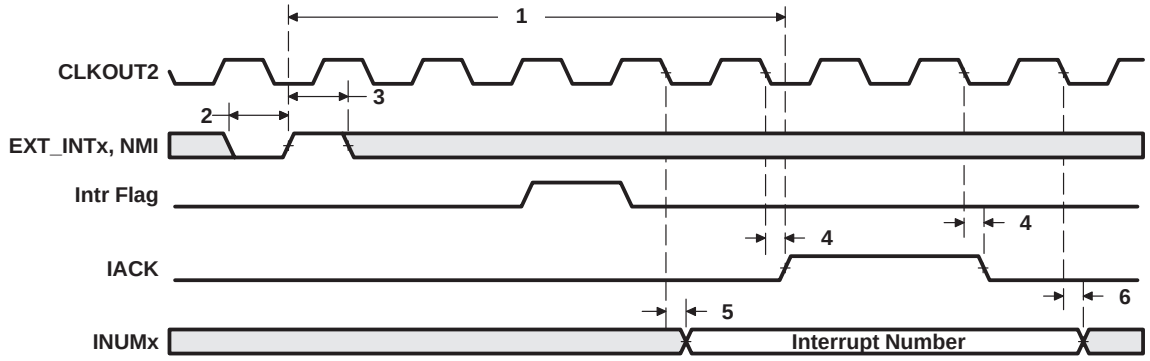


Figure 27. Interrupt Timing

EXPANSION BUS SYNCHRONOUS FIFO TIMING

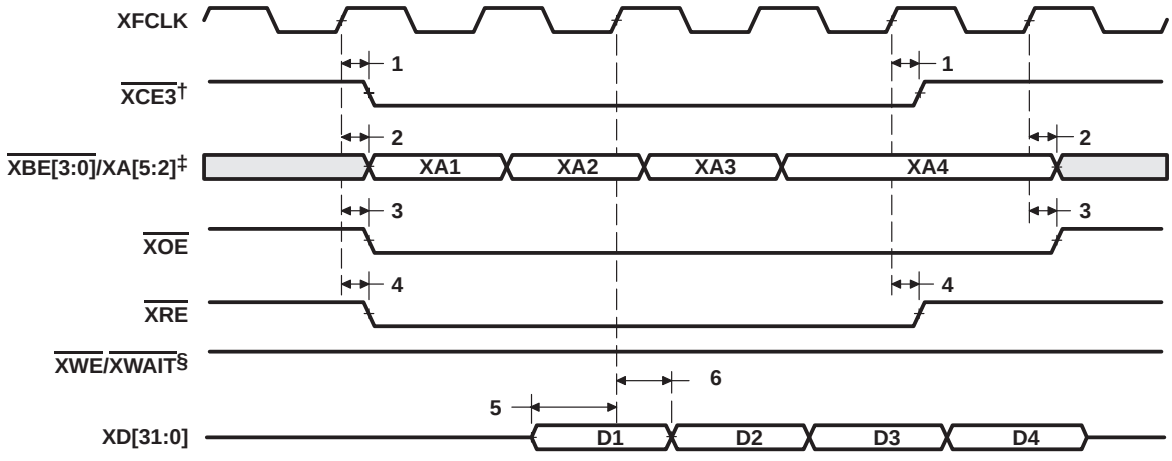
timing requirements for synchronous FIFO interface (see Figure 28, Figure 29, and Figure 30)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
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

switching characteristics for synchronous FIFO interface (see Figure 28, Figure 29, and Figure 30)

NO.	PARAMETER		’C6202-200, ’C6202-250		’C6202B-250 ’C6202B-300 ’C6203-250 ’C6203-300 ’C6204-200		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(XFCKH-XCEV)}$	Delay time, XFCLK high to $\overline{XCEx}$ valid	1.5	5.2	1.5	4.5	ns
2	$t_{d(XFCKH-XAV)}$	Delay time, XFCLK high to $\overline{XBE[3:0]}$/XA[5:2] valid†	1.5	5.2	1.5	4.5	ns
3	$t_{d(XFCKH-XOEV)}$	Delay time, XFCLK high to $\overline{XOE}$ valid	1.5	5.2	1.5	4.5	ns
4	$t_{d(XFCKH-XREV)}$	Delay time, XFCLK high to $\overline{XRE}$ valid	1.5	5.2	1.5	4.5	ns
7	$t_{d(XFCKH-XWEV)}$	Delay time, XFCLK high to $\overline{XWE}$/$\overline{XWAIT}$‡ valid	1.5	5.2	1.5	4.5	ns
8	$t_{d(XFCKH-XDV)}$	Delay time, XFCLK high to XDx valid		5.2		4.5	ns
9	$t_{d(XFCKH-XDIV)}$	Delay time, XFCLK high to XDx invalid	1.5		1.5		ns

† $\overline{XBE[3:0]}$/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
‡ XWE/XWAIT operates as the write enable signal XWE during synchronous FIFO accesses.

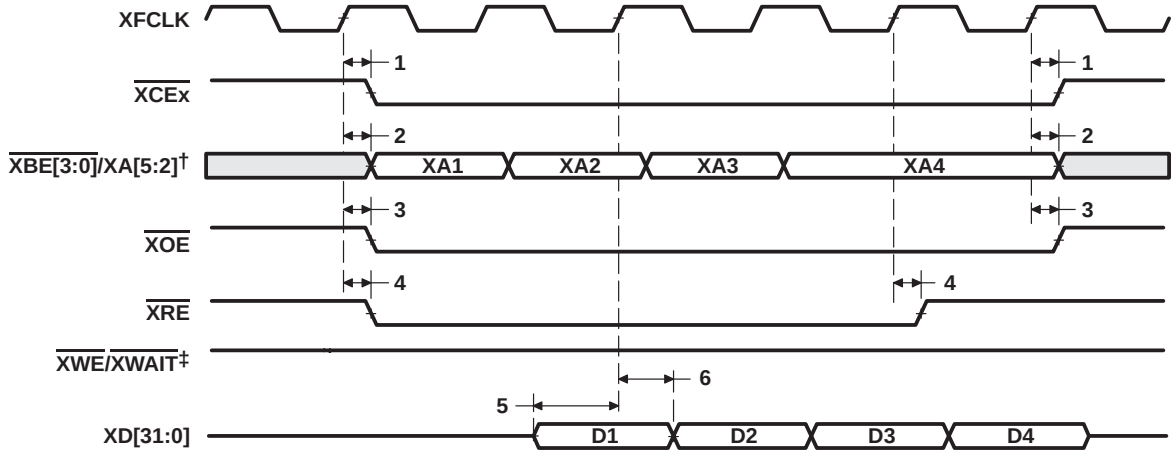


† FIFO read (glueless) mode only available in $\overline{XCE3}$.
‡ $\overline{XBE[3:0]}$/XA[5:2] operates as address signals XA[5:2] during synchronous FIFO accesses.
§ XWE/XWAIT operates as the write enable signal XWE during synchronous FIFO accesses.

Figure 28. FIFO Read Timing (Glueless Read Mode)

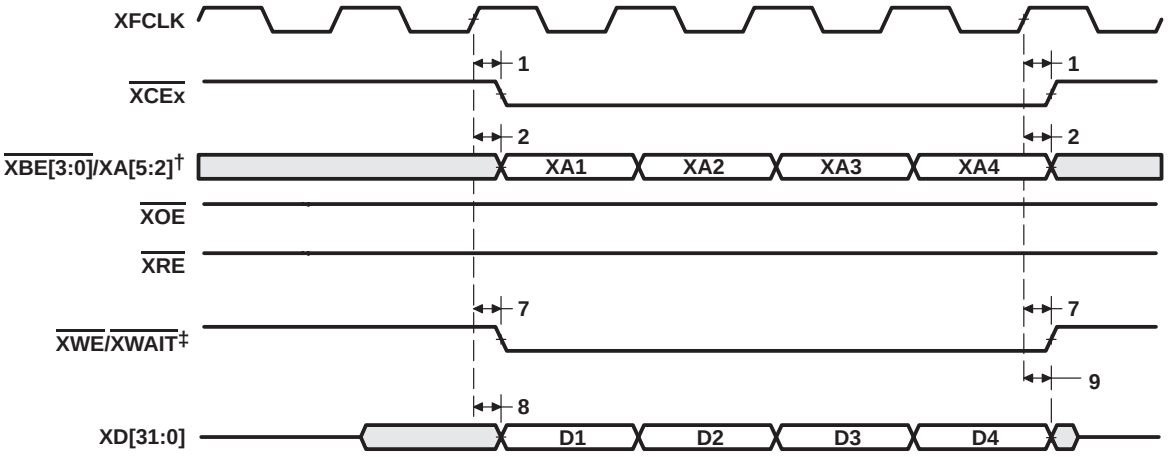
PRODUCT PREVIEW

EXPANSION BUS SYNCHRONOUS FIFO TIMING (CONTINUED)



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

Figure 29. FIFO Read Timing



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

Figure 30. FIFO Write Timing

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING

timing requirements for asynchronous peripheral cycles†‡§¶ (see Figure 31–Figure 34)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
3	$t_{su(XDV-XREH)}$	Setup time, XDx valid before $\overline{XRE}$ high	4.5		ns
4	$t_{h(XREH-XDV)}$	Hold time, XDx valid after $\overline{XRE}$ high	1		ns
6	$t_{su(XRDYH-XREL)}$	Setup time, XRDY high before $\overline{XRE}$ low	–[(RST – 3) * P – 6]		ns
7	$t_{h(XREL-XRDYH)}$	Hold time, XRDY high after $\overline{XRE}$ low	(RST – 3) * P + 2		ns
9	$t_{su(XRDYL-XREL)}$	Setup time, XRDY low before $\overline{XRE}$ low	–[(RST – 3) * P – 6]		ns
10	$t_{h(XREL-XRDYL)}$	Hold time, XRDY low after $\overline{XRE}$ low	(RST – 3) * P + 2		ns
11	$t_{w(XRDYH)}$	Pulse width, XRDY high	2P		ns
15	$t_{su(XRDYH-XWEL)}$	Setup time, XRDY high before $\overline{XWE}$ low	–[(WST – 3) * P – 6]		ns
16	$t_{h(XWEL-XRDYH)}$	Hold time, XRDY high after $\overline{XWE}$ low	(WST – 3) * P + 2		ns
18	$t_{su(XRDYL-XWEL)}$	Setup time, XRDY low before $\overline{XWE}$ low	–[(WST – 3) * P – 6]		ns
19	$t_{h(XWEL-XRDYL)}$	Hold time, XRDY low after $\overline{XWE}$ low	(WST – 3) * P + 2		ns

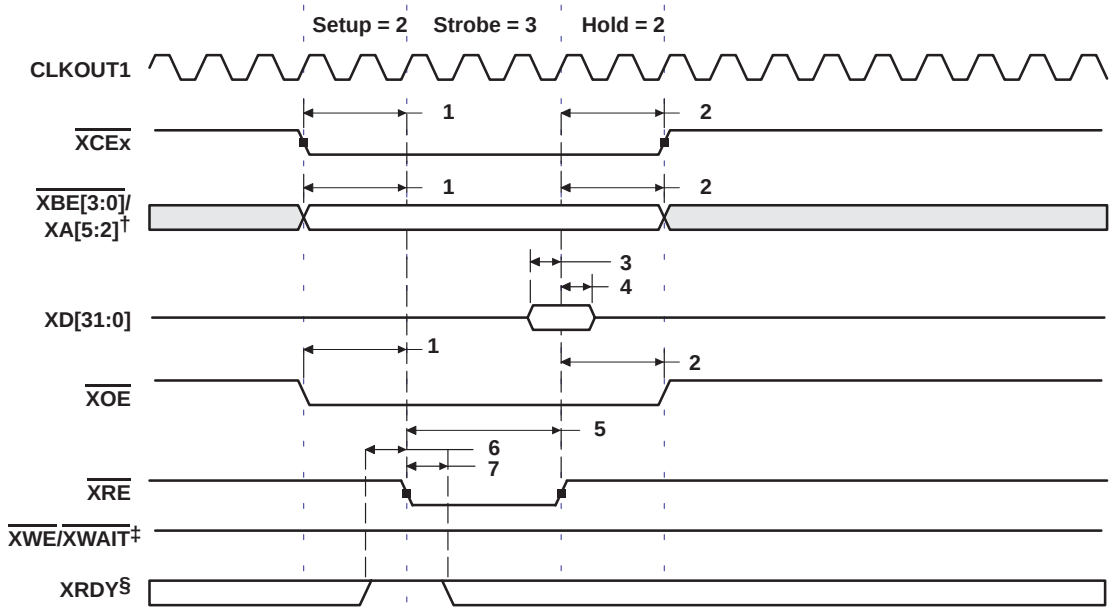
† To ensure data setup time, simply program the strobe width wide enough. XRDY is internally synchronized. If XRDY does meet setup or hold time, it may be recognized in the current cycle or the next cycle. Thus, XRDY can be an asynchronous input.
‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the XBUS XCE space control registers.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
¶ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use XRDY input to extend strobe width.

switching characteristics for asynchronous peripheral cycles‡§¶# (see Figure 31–Figure 34)

NO.	PARAMETER		-200 -250 -300			UNIT
			MIN	TYP	MAX	
1	$t_{osu(SELV-XREL)}$	Output setup time, select signals valid to $\overline{XRE}$ low	RS * P – 2			ns
2	$t_{oh(XREH-SELIV)}$	Output hold time, $\overline{XRE}$ low to select signals invalid	RH * P – 2			ns
5	$t_{w(XREL)}$	Pulse width, $\overline{XRE}$ low		RST * P		ns
8	$t_{d(XRDYH-XREH)}$	Delay time, XRDY high to $\overline{XRE}$ high	3P		4P + 5	ns
12	$t_{osu(SELV-XWEL)}$	Output setup time, select signals valid to $\overline{XWE}$ low	WS * P – 2			ns
13	$t_{oh(XWEH-SELIV)}$	Output hold time, $\overline{XWE}$ low to select signals invalid	WH * P – 2			ns
14	$t_{w(XWEL)}$	Pulse width, $\overline{XWE}$ low		WST * P		ns
17	$t_{d(XRDYH-XWEH)}$	Delay time, XRDY high to $\overline{XWE}$ high	3P		4P + 5	ns

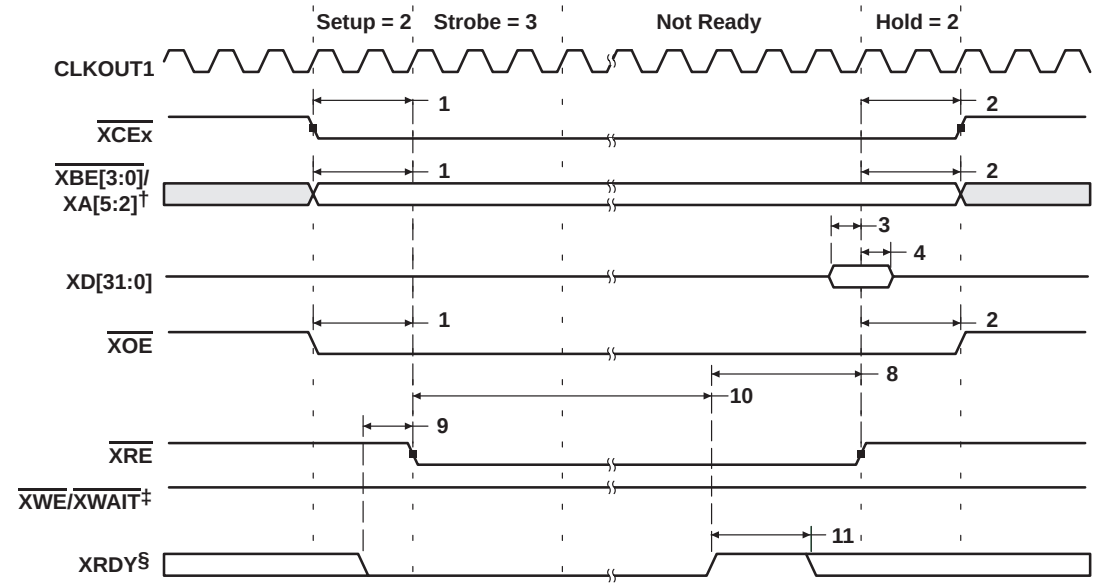
‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the XBUS XCE space control registers.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
¶ The sum of RS and RST (or WS and WST) must be a minimum of 4 in order to use XRDY input to extend strobe width.
# Select signals include: $\overline{XCE}$x, $\overline{XBE}$[3:0], XA[5:2], $\overline{XOE}$; and for writes, include XD[31:0], with the exception that $\overline{XCE}$x can stay active for an additional 7P ns following the end of the cycle.

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)



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

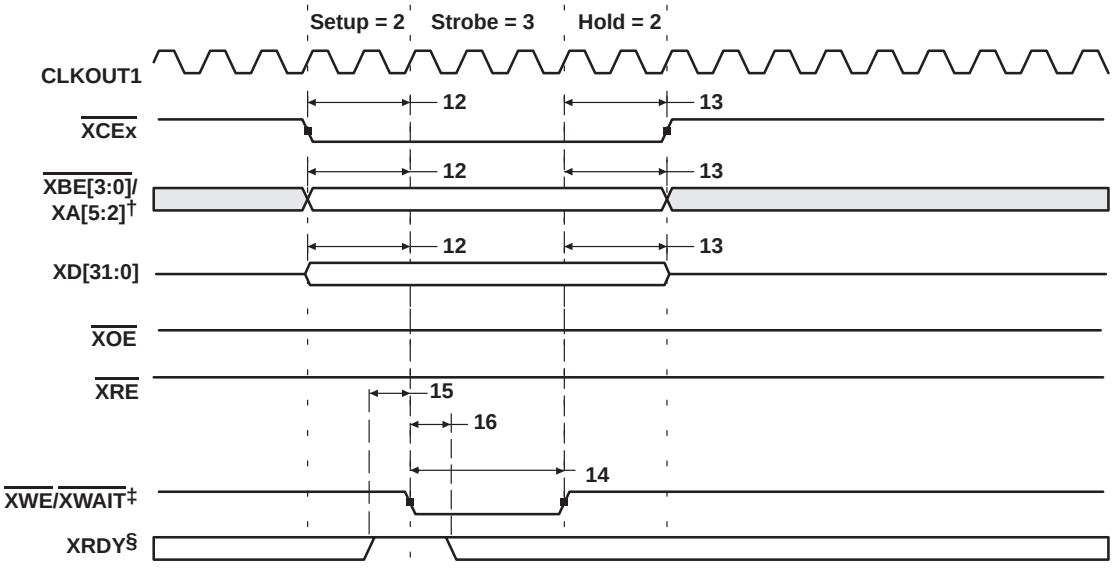
Figure 31. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Not Used)



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

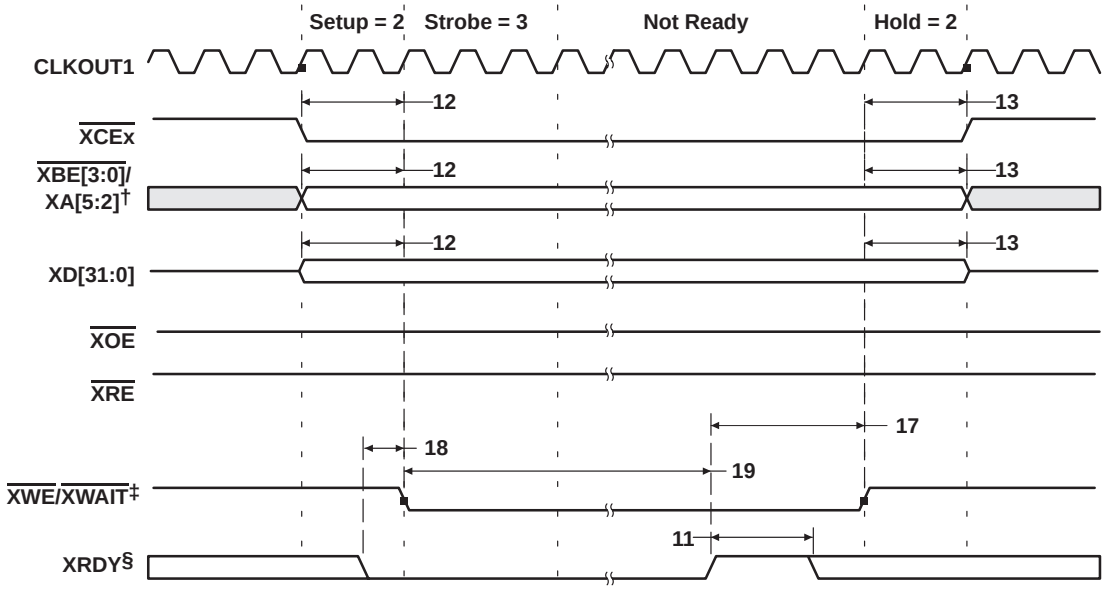
Figure 32. Expansion Bus Asynchronous Peripheral Read Timing (XRDY Used)

EXPANSION BUS ASYNCHRONOUS PERIPHERAL TIMING (CONTINUED)



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

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



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

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

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING

timing requirements with external device as bus master (see Figure 35 and Figure 36)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{su(XCSV-XCKIH)}$	Setup time, $\overline{XCS}$ valid before XCLKIN high	3.5		ns
2	$t_{h(XCKIH-XCS)}$	Hold time, $\overline{XCS}$ valid after XCLKIN high	2.8		ns
3	$t_{su(XAS-XCKIH)}$	Setup time, $\overline{XAS}$ valid before XCLKIN high	3.5		ns
4	$t_{h(XCKIH-XAS)}$	Hold time, $\overline{XAS}$ valid after XCLKIN high	2.8		ns
5	$t_{su(XCTL-XCKIH)}$	Setup time, XCNTL valid before XCLKIN high	3.5		ns
6	$t_{h(XCKIH-XCTL)}$	Hold time, XCNTL valid after XCLKIN high	2.8		ns
7	$t_{su(XWR-XCKIH)}$	Setup time, XW/R valid before XCLKIN high†	3.5		ns
8	$t_{h(XCKIH-XWR)}$	Hold time, XW/R valid after XCLKIN high†	2.8		ns
9	$t_{su(XBLTV-XCKIH)}$	Setup time, XBLAST valid before XCLKIN high‡	3.5		ns
10	$t_{h(XCKIH-XBLTV)}$	Hold time, XBLAST valid after XCLKIN high‡	2.8		ns
16	$t_{su(XBEV-XCKIH)}$	Setup time, $\overline{XBE[3:0]}$/XA[5:2] valid before XCLKIN high§	3.5		ns
17	$t_{h(XCKIH-XBEV)}$	Hold time, $\overline{XBE[3:0]}$/XA[5:2] valid after XCLKIN high§	2.8		ns
18	$t_{su(XD-XCKIH)}$	Setup time, XDx valid before XCLKIN high	3.5		ns
19	$t_{h(XCKIH-XD)}$	Hold time, XDx valid after XCLKIN high	2.8		ns

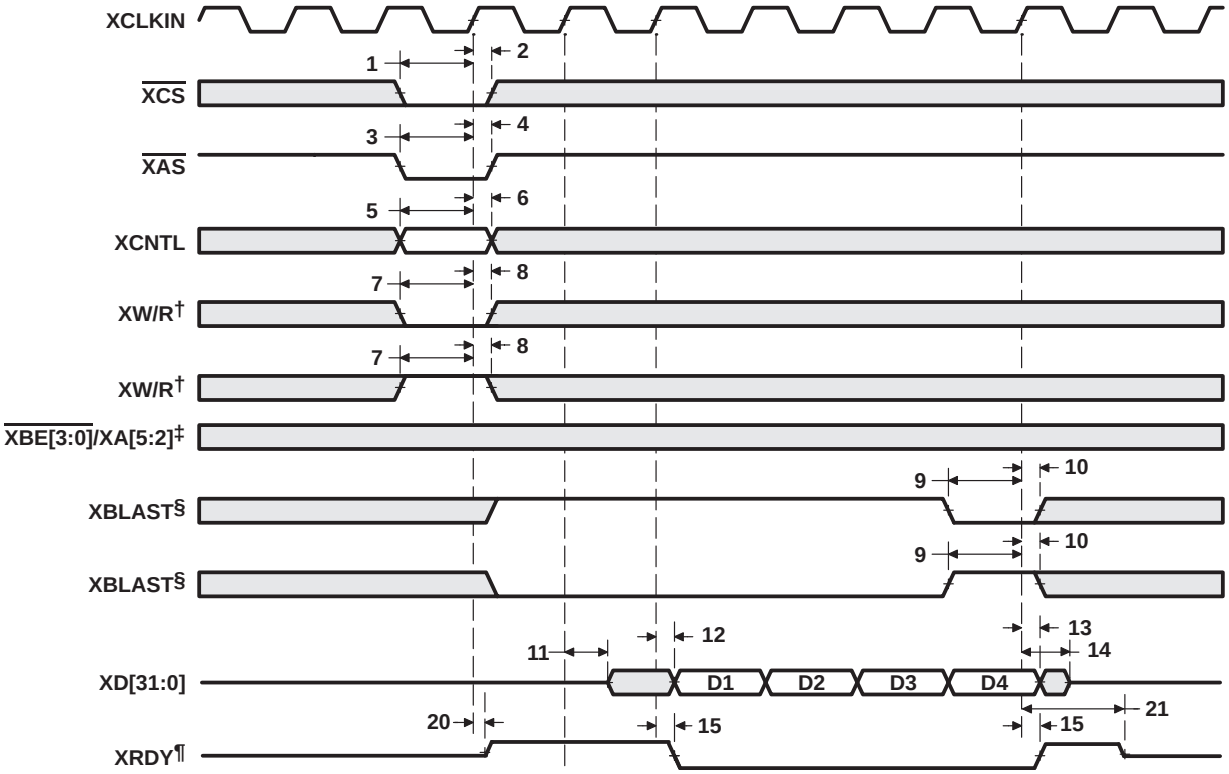
† XW/R input/output polarity selected at boot.
‡ $\overline{XBLAST}$ input polarity selected at boot.
§ $\overline{XBE[3:0]}$/XA[5:2] operates as byte enables $\overline{XBE[3:0]}$ during host-port accesses.

switching characteristics with external device as bus master¶ (see Figure 35 and Figure 36)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
11	$t_{d(XCKIH-XDLZ)}$	Delay time, XCLKIN high to XDx low impedance	0		ns
12	$t_{d(XCKIH-XDV)}$	Delay time, XCLKIN high to XDx valid		16.5	ns
13	$t_{d(XCKIH-XDIV)}$	Delay time, XCLKIN high to XDx invalid	5		ns
14	$t_{d(XCKIH-XDHZ)}$	Delay time, XCLKIN high to XDx high impedance		4P	ns
15	$t_{d(XCKIH-XRY)}$	Delay time, XCLKIN high to XRDY valid#	5	16.5	ns
20	$t_{d(XCKIH-XRYLZ)}$	Delay time, XCLKIN high to XRDY low impedance	5	16.5	ns
21	$t_{d(XCKIH-XRYHZ)}$	Delay time, XCLKIN high to XRDY high impedance#	2P + 5	3P + 16.5	ns

¶ P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
# XRDY operates as active-low ready input/output during host-port accesses.

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)

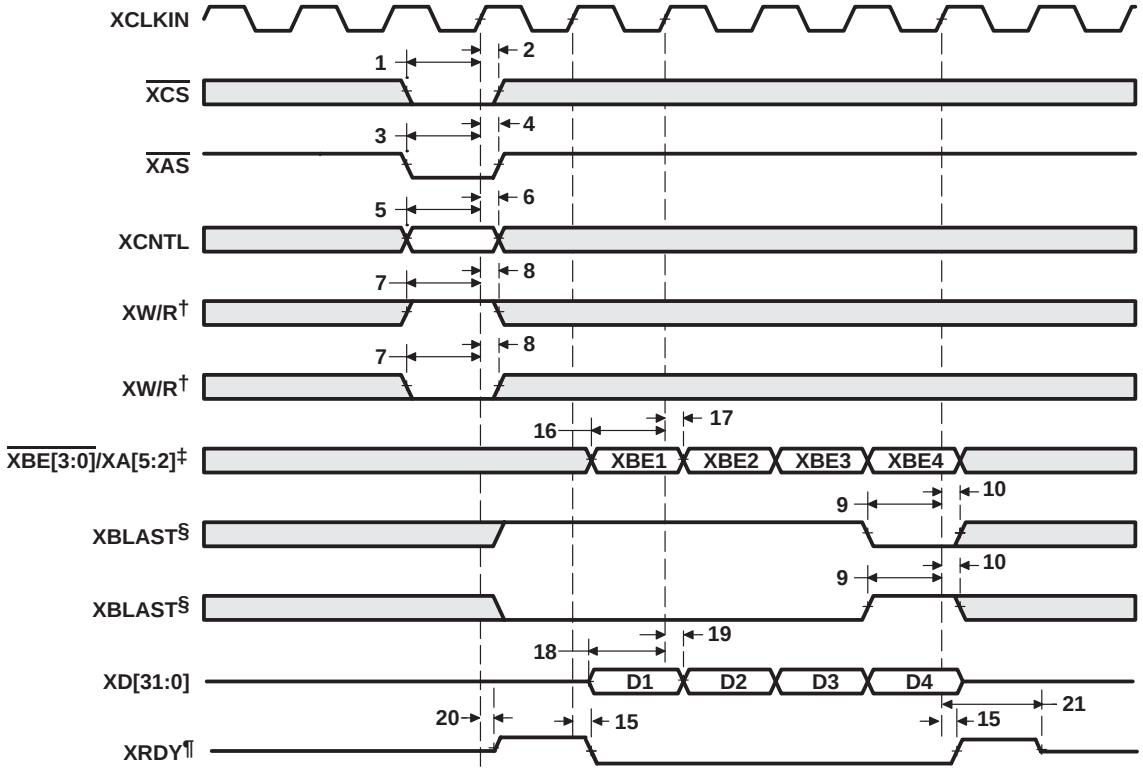


† XW/R input/output polarity selected at boot
‡ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
§ XBLAST input polarity selected at boot
¶ XRDY operates as active-low ready input/output during host-port accesses.

Figure 35. External Host as Bus Master—Read

PRODUCT PREVIEW

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)



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

Figure 36. External Host as Bus Master—Write

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)

timing requirements with ’C62x as bus master (see Figure 37, Figure 38, and Figure 39)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
9	$t_{su(XDV-XCKIH)}$	Setup time, XDx valid before XCLKIN high	3.5		ns
10	$t_{h(XCKIH-XDV)}$	Hold time, XDx valid after XCLKIN high	2.8		ns
11	$t_{su(XRY-XCKIH)}$	Setup time, XRDY valid before XCLKIN high†	3.5		ns
12	$t_{h(XCKIH-XRY)}$	Hold time, XRDY valid after XCLKIN high†	2.8		ns
14	$t_{su(XBFF-XCKIH)}$	Setup time, XBOFF valid before XCLKIN high	3.5		ns
15	$t_{h(XCKIH-XBFF)}$	Hold time, XBOFF valid after XCLKIN high	2.8		ns

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

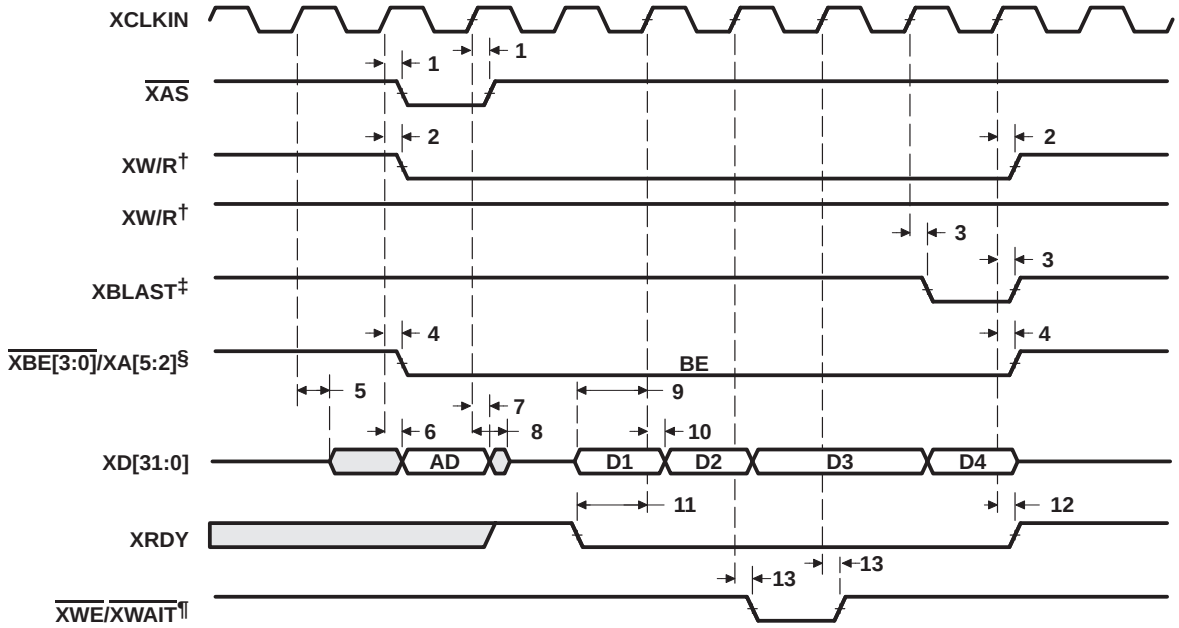
switching characteristics with ’C62x as bus master (see Figure 37, Figure 38, and Figure 39)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{d(XCKIH-XASV)}$	Delay time, XCLKIN high to $\overline{\text{XAS}}$ valid	5	16.5	ns
2	$t_{d(XCKIH-XWRV)}$	Delay time, XCLKIN high to XW/R valid‡	5	16.5	ns
3	$t_{d(XCKIH-XBLTV)}$	Delay time, XCLKIN high to XBLAST valid§	5	16.5	ns
4	$t_{d(XCKIH-XBEV)}$	Delay time, XCLKIN high to $\overline{\text{XBE[3:0]}}$/XA[5:2] valid¶	5	16.5	ns
5	$t_{d(XCKIH-XDLZ)}$	Delay time, XCLKIN high to XDx low impedance	0		ns
6	$t_{d(XCKIH-XDV)}$	Delay time, XCLKIN high to XDx valid		16.5	ns
7	$t_{d(XCKIH-XDIV)}$	Delay time, XCLKIN high to XDx invalid	5		ns
8	$t_{d(XCKIH-XDHZ)}$	Delay time, XCLKIN high to XDx high impedance		4P	ns
13	$t_{d(XCKIH-XWTV)}$	Delay time, XCLKIN high to $\overline{\text{XWE}}$/$\overline{\text{XWAIT}}$ valid#	5	16.5	ns

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

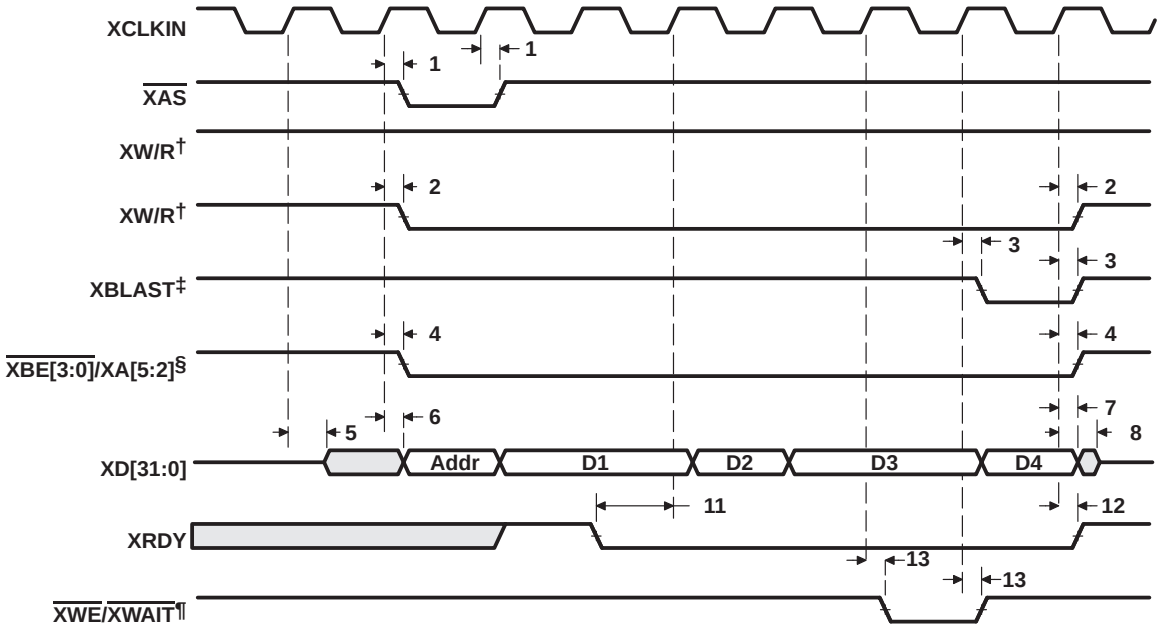
PRODUCT PREVIEW

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)



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

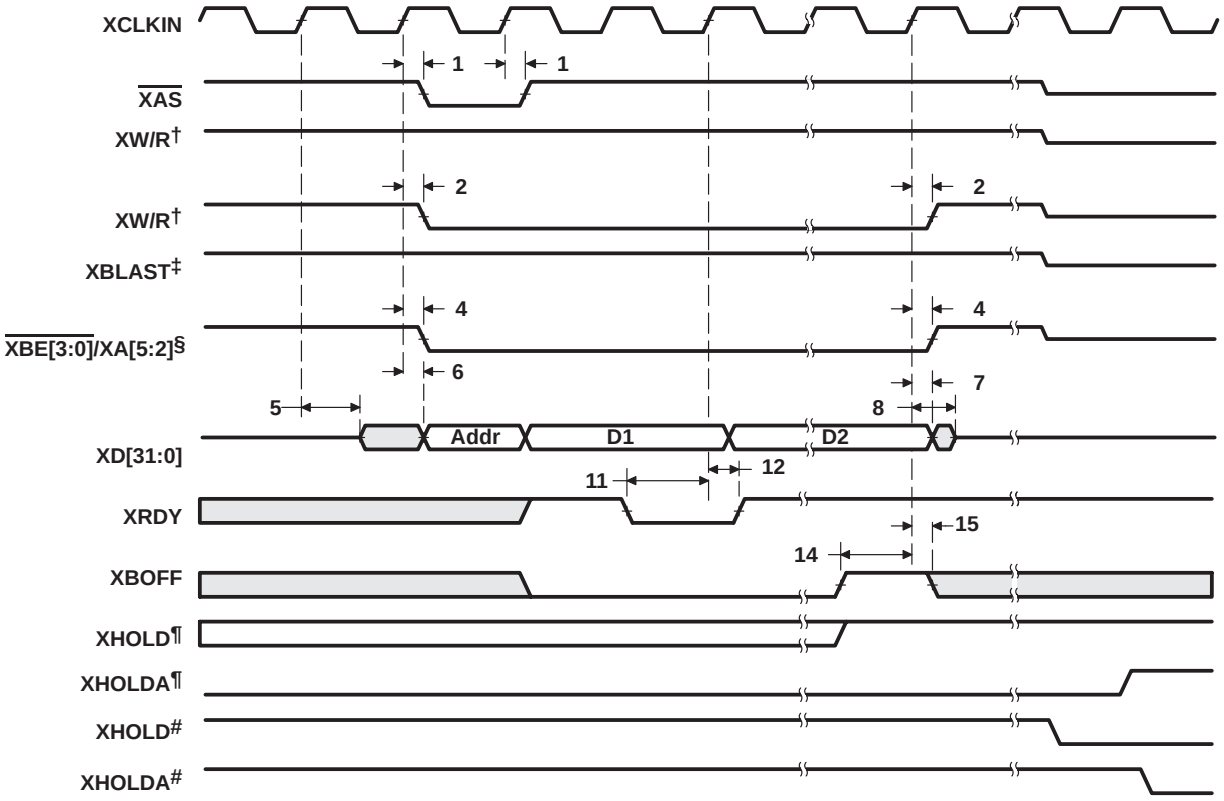
Figure 37. ’C62x as Bus Master—Read



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

Figure 38. ’C62x as Bus Master—Write

EXPANSION BUS SYNCHRONOUS HOST PORT TIMING (CONTINUED)



† XW/R input/output polarity selected at boot
‡ XBLAST output polarity is always active low.
§ XBE[3:0]/XA[5:2] operates as byte enables XBE[3:0] during host-port accesses.
¶ Internal arbiter enabled
# External arbiter enabled
‖ This diagram illustrates XBOFF timing. Bus arbitration timing is shown in Figure 42 and Figure 43.

Figure 39. ’C62x as Bus Master—BOFF Operation‖

EXPANSION BUS ASYNCHRONOUS HOST PORT TIMING

timing requirements with external device as asynchronous bus master† (see Figure 40 and Figure 41)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{w(XCSL)}$	Pulse duration, $\overline{XCS}$ low	4P		ns
2	$t_{w(XCSH)}$	Pulse duration, $\overline{XCS}$ high	4P		ns
3	$t_{su(XSEL-XCSL)}$	Setup time, expansion bus select signals‡ valid before $\overline{XCS}$ low	1		ns
4	$t_{h(XCSL-XSEL)}$	Hold time, expansion bus select signals‡ valid after $\overline{XCS}$ low	3		ns
10	$t_{h(XRYL-XCSL)}$	Hold time, $\overline{XCS}$ low after XRDY low	P + 1.5		ns
11	$t_{su(XBEV-XCSH)}$	Setup time, $\overline{XBE[3:0]}$/XA[5:2] valid before $\overline{XCS}$ high§	1		ns
12	$t_{h(XCSH-XBEV)}$	Hold time, $\overline{XBE[3:0]}$/XA[5:2] valid after $\overline{XCS}$ high§	3		ns
13	$t_{su(XDV-XCSH)}$	Setup time, XDx valid before $\overline{XCS}$ high	1		ns
14	$t_{h(XCSH-XDV)}$	Hold time, XDx valid after $\overline{XCS}$ high	3		ns

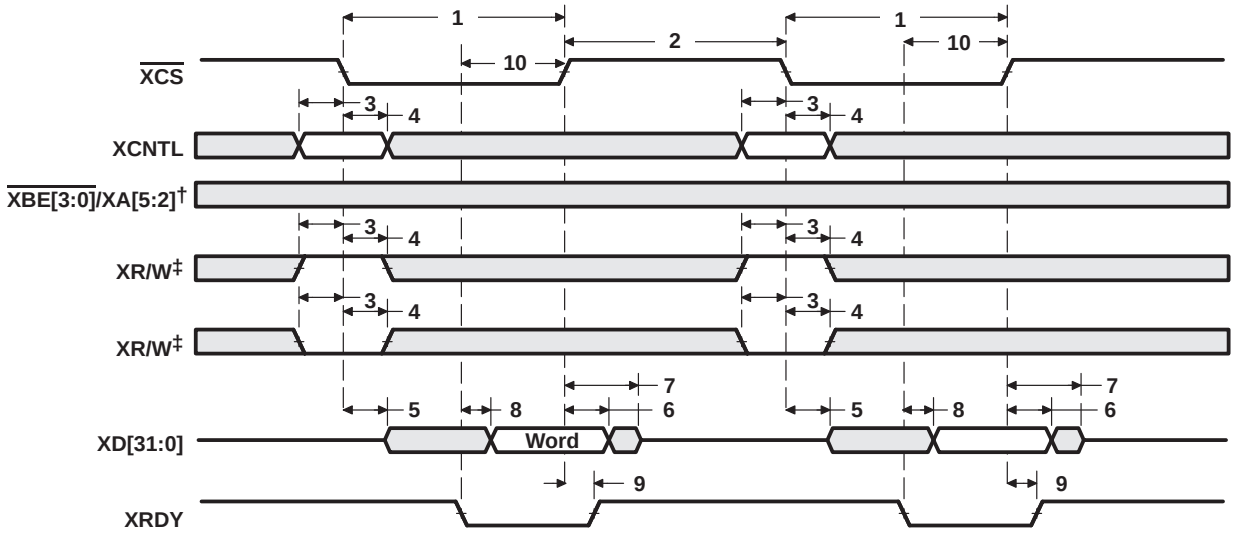
† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ Expansion bus select signals include XCNTL and XR/W.
§ $\overline{XBE[3:0]}$/XA[5:2] operates as byte enables $\overline{XBE[3:0]}$ during host-port accesses.

switching characteristics with external device as asynchronous bus master† (see Figure 40 and Figure 41)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
5	$t_{d(XCSL-XDLZ)}$	Delay time, $\overline{XCS}$ low to XDx low impedance	0		ns
6	$t_{d(XCSH-XDIV)}$	Delay time, $\overline{XCS}$ high to XDx invalid	0	12	ns
7	$t_{d(XCSH-XDHZ)}$	Delay time, $\overline{XCS}$ high to XDx high impedance		4P	ns
8	$t_{d(XRYL-XDV)}$	Delay time, XRDY low to XDx valid	–4	1	ns
9	$t_{d(XCSH-XRYH)}$	Delay time, $\overline{XCS}$ high to XRDY high	0	12	ns

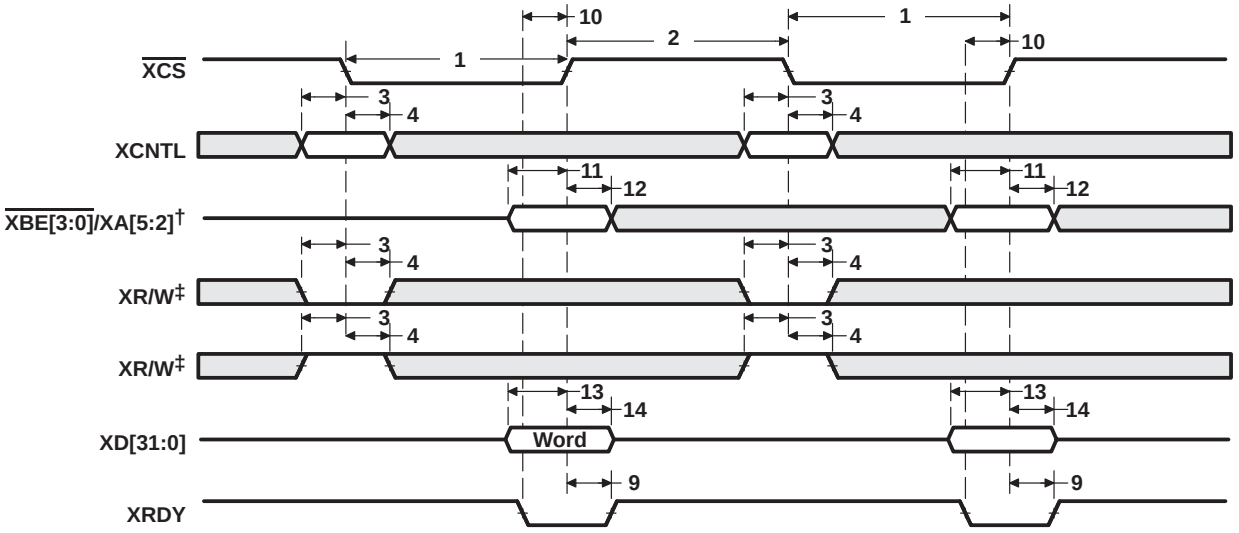
† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

EXPANSION BUS ASYNCHRONOUS HOST PORT TIMING (CONTINUED)



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

Figure 40. External Device as Asynchronous Master—Read



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

Figure 41. External Device as Asynchronous Master—Write

PRODUCT PREVIEW

XHOLD/XHOLDA TIMING

timing requirements for expansion bus arbitration (internal arbiter enabled)† (see Figure 42)

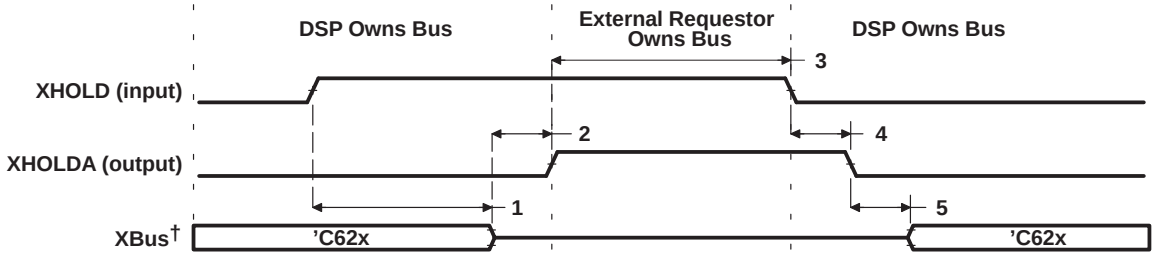
NO.			-200 -250 -300		UNIT
			MIN	MAX	
3	$t_{oh(XHDAH-XHDH)}$	Output hold time, XHOLD high after XHOLDA high	P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

switching characteristics for expansion bus arbitration (internal arbiter enabled)†‡ (see Figure 42)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{R(XHDH-XBHZ)}$	Response time, XHOLD high to XBus high impedance	3P	§	ns
2	$t_{d(XBHZ-XHDAH)}$	Delay time, XBus high impedance to XHOLDA high	0	2P	ns
4	$t_{R(XHDL-XHDAL)}$	Response time, XHOLD low to XHOLDA low	3P		ns
5	$t_{d(XHDAL-XBLZ)}$	Delay time, XHOLDA low to XBus low impedance	0	2P	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ XBus consists of $\overline{XBE[3:0]}$/XA[5:2], $\overline{XAS}$, XW/R, and XBLAST.
§ All pending XBus transactions are allowed to complete before XHOLDA is asserted.



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

Figure 42. Expansion Bus Arbitration—Internal Arbiter Enabled

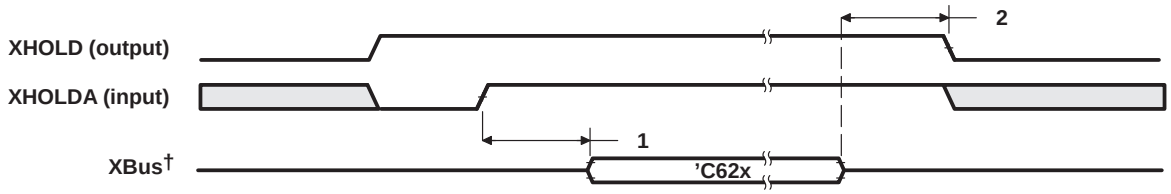
PRODUCT PREVIEW

XHOLD/XHOLDA TIMING (CONTINUED)

switching characteristics for expansion bus arbitration (internal arbiter disabled)† (see Figure 43)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{d(XHDAH-XBLZ)}$	Delay time, XHOLDA high to XBus low impedance‡	2P	2P + 10	ns
2	$t_{d(XBHZ-XHDL)}$	Delay time, XBus high impedance to XHOLD low‡	0	2P	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ XBus consists of $\overline{\text{XBE}}$[3:0]/XA[5:2], $\overline{\text{XAS}}$, XW/R, and XBLAST.



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

Figure 43. Expansion Bus Arbitration—Internal Arbiter Disabled

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP†‡ (see Figure 44)

NO.				-200 -250 -300		UNIT
				MIN	MAX	
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P§		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1¶		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9		ns
			CLKR ext	2		
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	6		ns
			CLKR ext	3		
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	8		ns
			CLKR ext	0.5		
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	3		ns
			CLKR ext	4		
10	$t_{su(FXH-CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	9		ns
			CLKX ext	2		
11	$t_{h(CKXL-FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	6		ns
			CLKX ext	3		

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ P = 1/CPU clock frequency in ns.
§ The maximum McBSP bit rate 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 250 MHz (P = 4 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 McBSP bit rate applies to the following hardware configuration: the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
¶ The minimum CLKR/X pulse duration is either (P – 1) or 4 ns, whichever is larger. For example, when running parts at 250 MHz (P = 4 ns), use 4 ns as the minimum CLKR/X pulse duration. When running parts at 100 MHz (P = 10 ns), use (P – 1) = 9 ns as the minimum CLKR/X pulse duration.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics for McBSP†‡ (see Figure 44)

NO.	PARAMETER			-200 -250 -300		UNIT
				MIN	MAX	
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		4	16	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	2P§¶		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C – 1#	C + 1#	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	–2	3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	–2	3	ns
			CLKX ext	3	9	
12	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	–1	5	ns
			CLKX ext	2	9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	–1	4	ns
			CLKX ext	2	11	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode.	FSX int	–1	5	ns
			FSX ext	0	10	

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum delay times also represent minimum output hold times.
§ P = 1/CPU clock frequency in ns.
¶ The maximum McBSP bit rate 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 250 MHz (P = 4 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 McBSP bit rate applies to the following hardware configuration: the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
# C = H or L
S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100 MHz limit.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

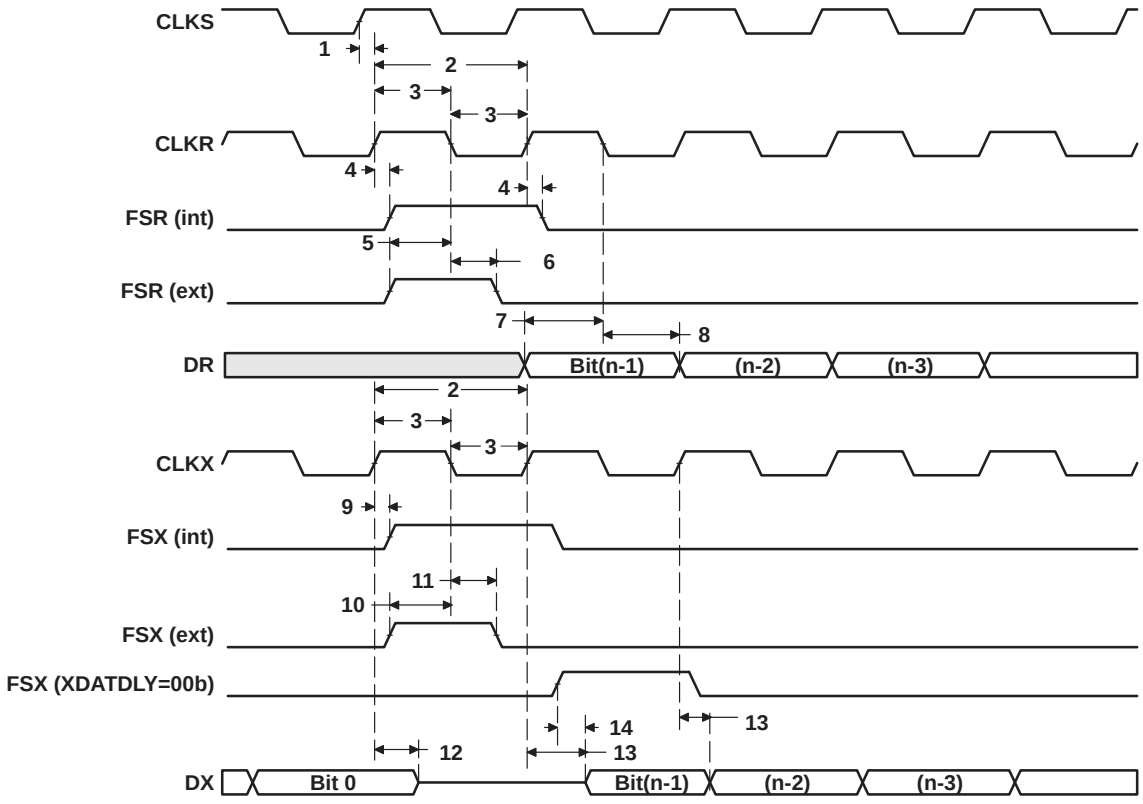


Figure 44. McBSP Timings

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 45)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{su(FRH-CKSH)}$	Setup time, FSR high before CLKS high	4		ns
2	$t_{h(CKSH-FRH)}$	Hold time, FSR high after CLKS high	4		ns

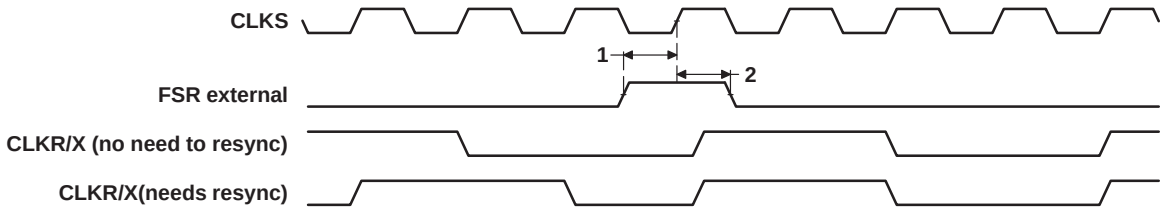


Figure 45. FSR Timing When GSYNC = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 46)

NO.			-200 -250 -300				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 6P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 46)

NO.	PARAMETER		-200 -250 -300				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L – 2	L + 3			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–3	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	L – 2	L + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			P + 3	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 2	4P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100 MHz limit.
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

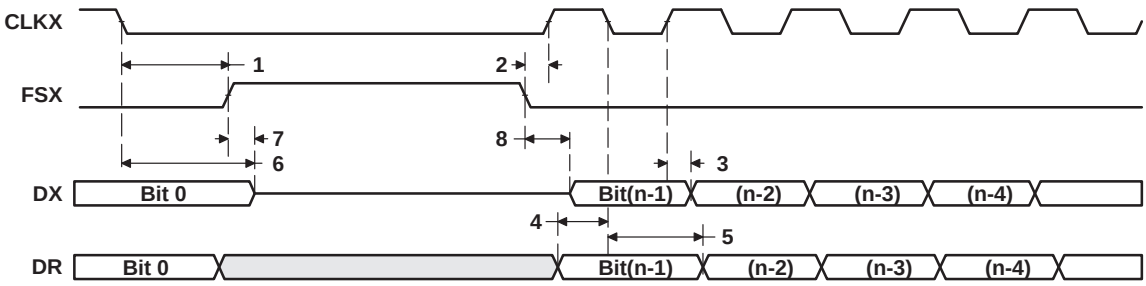


Figure 46. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 47)

NO.			-200 -250 -300				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 6P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 47)

NO.	PARAMETER		-200 -250 -300				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	L – 2	L + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	T – 2	T + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–2	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	–2	4	3P + 3	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H – 2	H + 4	2P + 2	4P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100 MHz limit.
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

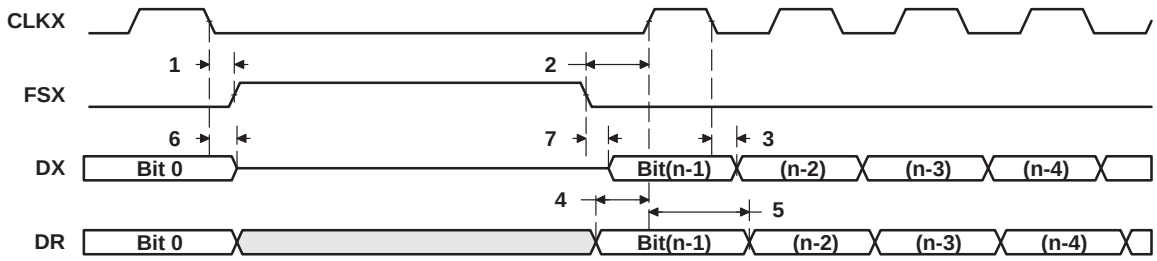


Figure 47. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 48)

NO.			-200 -250 -300				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 6P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 48)

NO.	PARAMETER		-200 -250 -300				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	H – 2	H + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–2	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	H – 2	H + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			P + 3	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 2	4P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100 MHz limit.
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

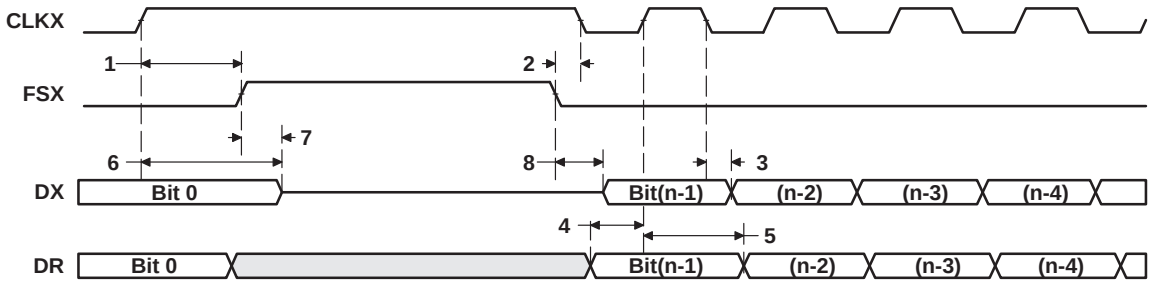


Figure 48. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 49)

NO.			-200 -250 -300				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 6P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 49)

NO.	PARAMETER		-200 -250 -300				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	H – 2	H + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	T – 2	T + 2			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–3	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	4	3P + 3	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L – 2	L + 5	2P + 2	4P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 100 MHz limit.
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

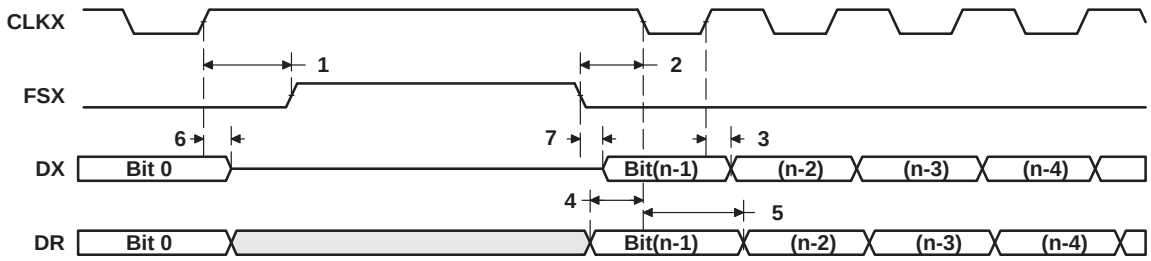


Figure 49. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

DMAC, TIMER, POWER-DOWN TIMING

switching characteristics for DMAC outputs† (see Figure 50)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{w(DMACH)}$	Pulse duration, DMAC high	2P – 3		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

Figure 50. DMAC Timing

timing requirements for timer inputs† (see Figure 51)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{w(TINPH)}$	Pulse duration, TINP high	2P		ns
2	$t_{w(TINPL)}$	Pulse duration, TINP low	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

switching characteristics for timer outputs† (see Figure 51)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
3	$t_{w(TOUTH)}$	Pulse duration, TOUT high	2P – 3		ns
4	$t_{w(TOUTL)}$	Pulse duration, TOUT low	2P – 3		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

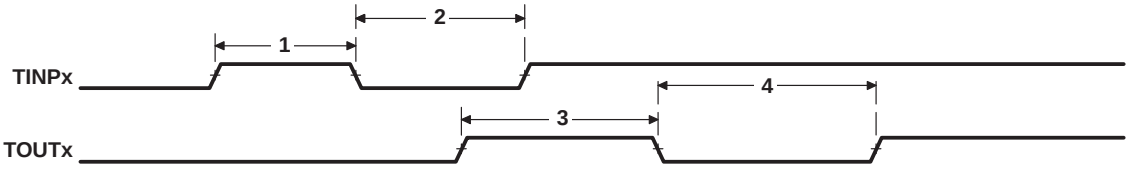


Figure 51. Timer Timing

DMAC, TIMER, POWER-DOWN TIMING (CONTINUED)

switching characteristics for power-down outputs† (see Figure 52)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{w(PDH)}$	Pulse duration, PD high	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 250 MHz, use P = 4 ns.

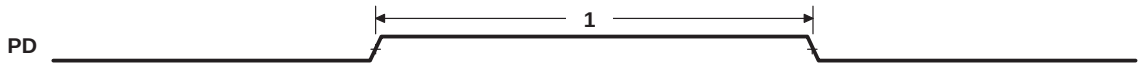


Figure 52. Power-Down Timing

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 53)

NO.			-200 -250 -300		UNIT
			MIN	MAX	
1	$t_{c(TCK)}$	Cycle time, TCK	50		ns
3	$t_{su(TDIV-TCKH)}$	Setup time, TDI/TMS/$\overline{TRST}$ valid before TCK high	11		ns
4	$t_{h(TCKH-TDIV)}$	Hold time, TDI/TMS/$\overline{TRST}$ valid after TCK high	9		ns

switching characteristics for JTAG test port (see Figure 53)

NO.	PARAMETER		-200 -250 -300		UNIT
			MIN	MAX	
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid	–4.5	12	ns

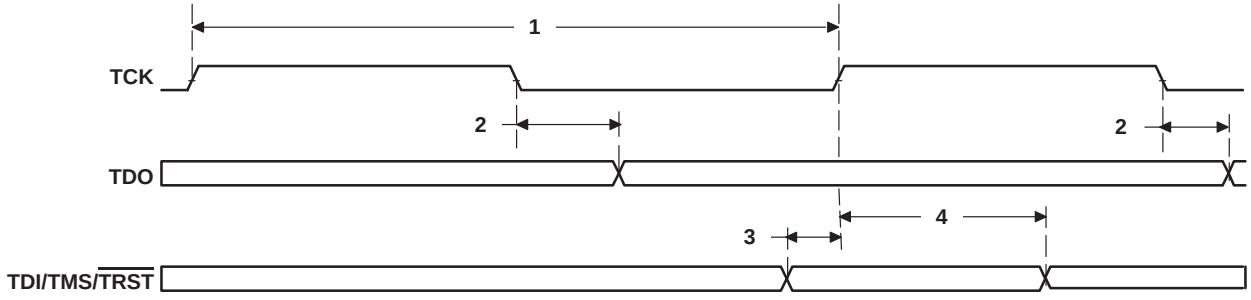
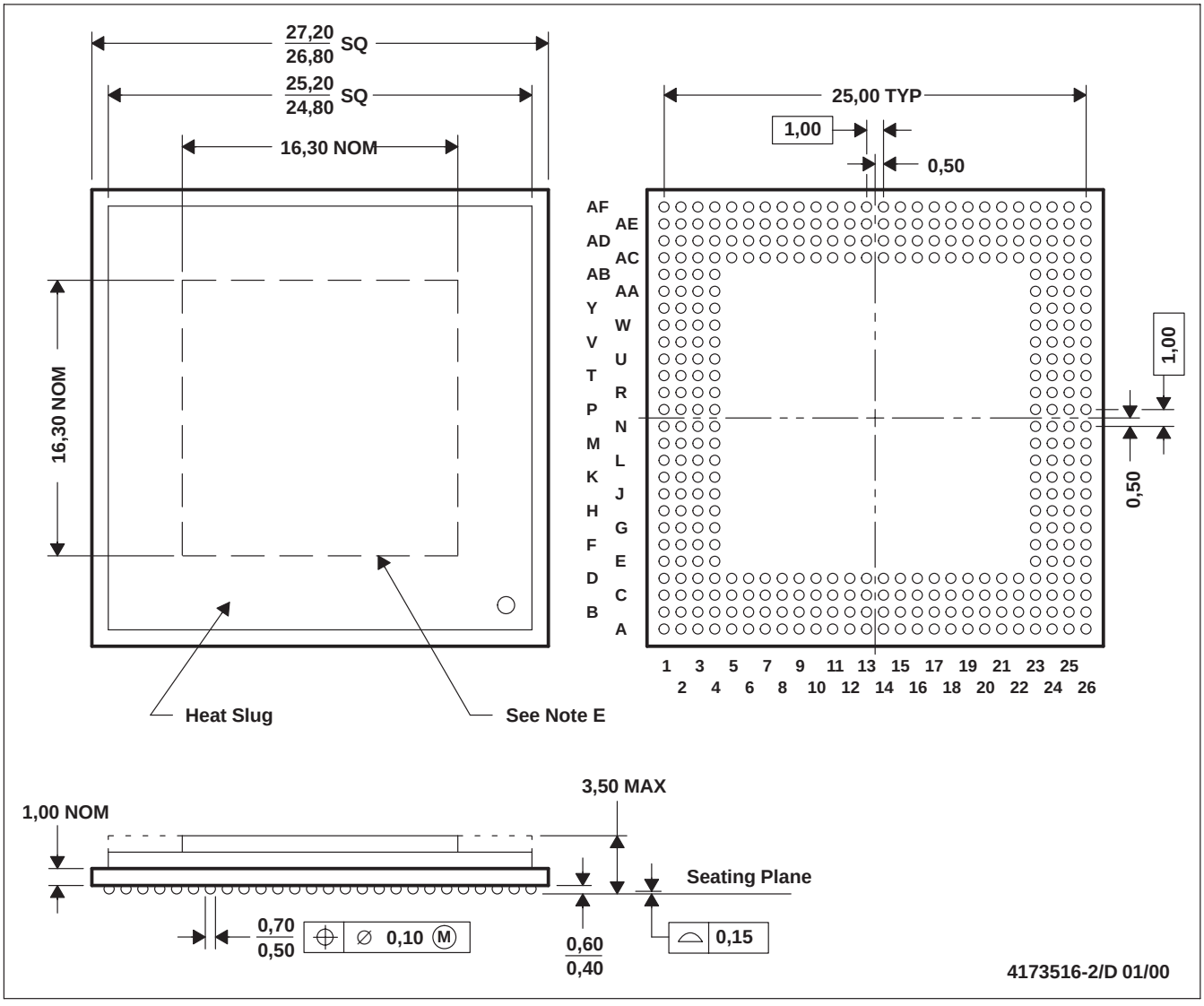


Figure 53. JTAG Test-Port Timing

MECHANICAL DATA

GJL (S-PBGA-N352) **PLASTIC BALL GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced plastic package with heat slug (HSL)
D. Flip chip application only
E. Possible protrusion in this area, but within 3,50 max package height specification
F. Falls within JEDEC MO-151/AAL-1

thermal resistance characteristics (S-PBGA package)

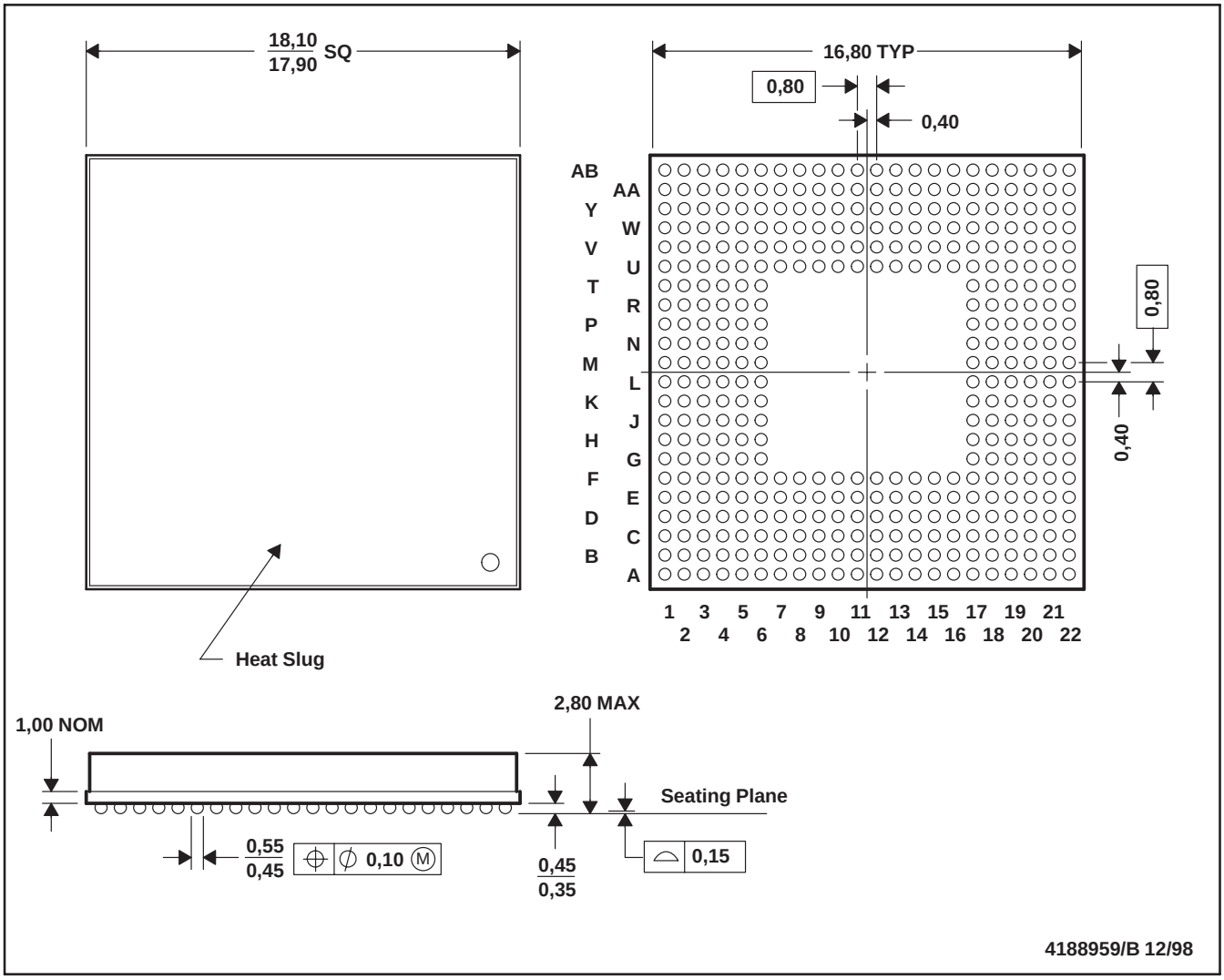
NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	0.47	N/A
2	$R\Theta_{JA}$	Junction-to-free air	14.2	0
3	$R\Theta_{JA}$	Junction-to-free air	12.3	100
4	$R\Theta_{JA}$	Junction-to-free air	10.2	250
5	$R\Theta_{JA}$	Junction-to-free air	8.6	500

† LFPM = Linear Feet Per Minute

PRODUCT PREVIEW

MECHANICAL DATA

GLS (S-PBGA-N384) **PLASTIC BALL GRID ARRAY**



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

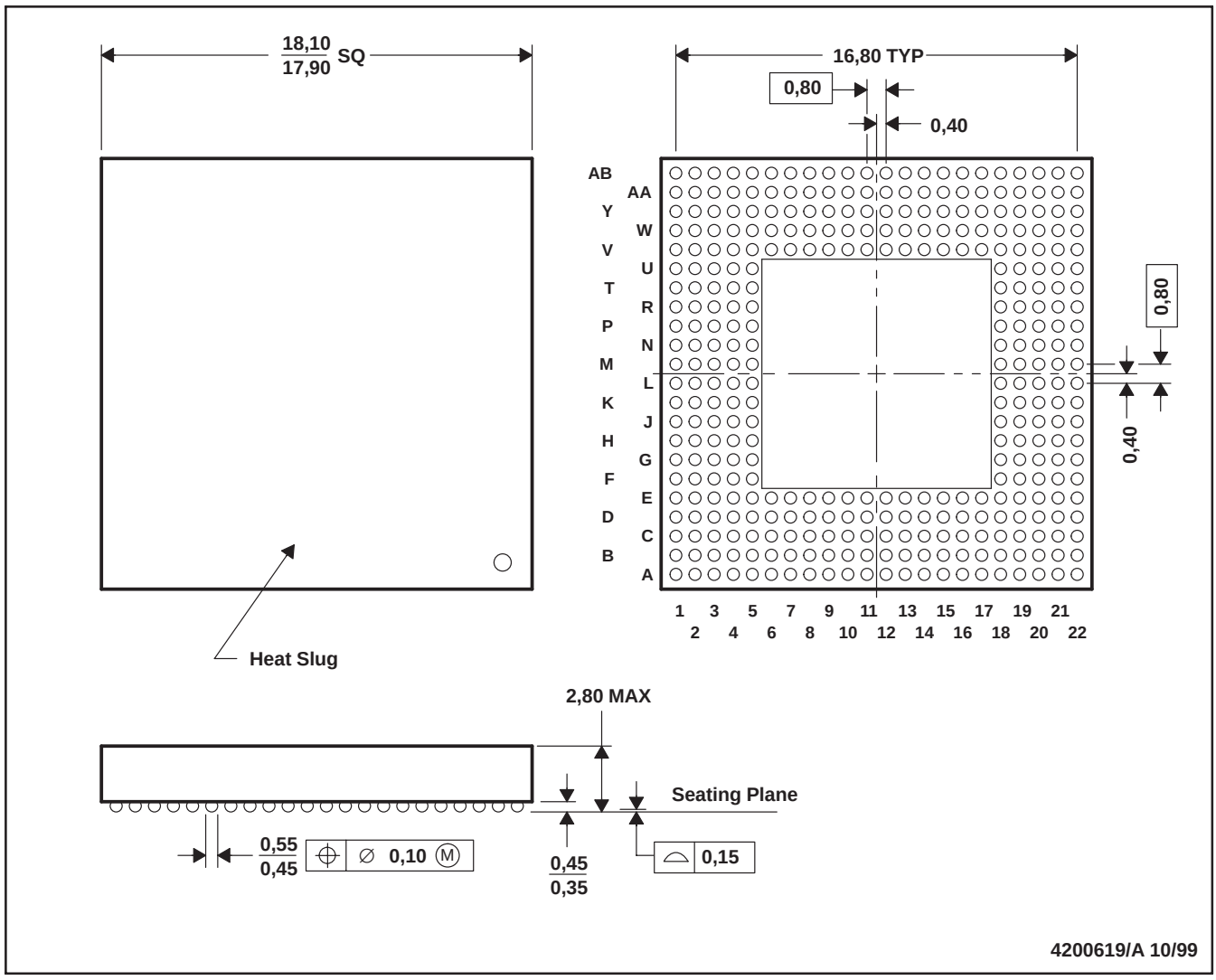
thermal resistance characteristics (S-PBGA package)

NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	0.85	N/A
2	$R\Theta_{JA}$	Junction-to-free air	21.6	0
3	$R\Theta_{JA}$	Junction-to-free air	17.9	100
4	$R\Theta_{JA}$	Junction-to-free air	14.2	250
5	$R\Theta_{JA}$	Junction-to-free air	11.8	500

† LFPM = Linear Feet Per Minute

MECHANICAL DATA

GLW (S-PBGA-N340) **PLASTIC BALL GRID ARRAY (CAVITY DOWN)**



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

PRODUCT PREVIEW

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

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

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.