- **Controlled Baseline**
  - **One Assembly/Test Site, One Fabrication Site**
- **Extended Temperature Performance of –40°C to 105°C**
- **Enhanced Diminishing Manufacturing Sources (DMS) Support**
- **Enhanced Product-Change Notification**
- **Qualification Pedigree†**
- **High-Performance Fixed-Point Digital Signal Processor (DSP) SM320C6201**
  - **5-ns Instruction Cycle Time**
  - **200-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **1600 MIPS**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) TMS320C62x™ DSP CPU Core**
  - **Eight Independent Functional Units:**
    - **Six Arithmetic Logic Units (ALUs) (32-/40-Bit)**
    - **Two 16-Bit Multipliers (32-Bit Results)**
  - **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)**
  - **32-Bit Address Range**
  - **8-Bit Overflow Protection**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit-Counting**
  - **Normalization**
- **1M-Bit On-Chip SRAM**
  - **512K-Bit Internal Program/Cache (16K 32-Bit Instructions)**
  - **512K-Bit Dual-Access Internal Data (64K Bytes) Organized as Two Blocks for Improved Concurrency**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Asynchronous Memories: SRAM and EPROM**
  - **Glueless Interface to Synchronous Memories: SDRAM and SBSRAM**
- **Four-Channel Bootloading Direct-Memory-Access (DMA) Controller with an Auxiliary Channel**
- **16-Bit Host-Port Interface (HPI)**
  - **Access to Entire Memory Map**
- **Two Multichannel Buffered Serial Ports (McBSPs)**
  - **Direct Interface to T1/E1, MVIP, SCSA Framers**
  - **ST-Bus-Switching Compatible**
  - **Up to 256 Channels Each**
  - **AC97-Compatible**
  - **Serial Peripheral Interface (SPI) Compatible (Motorola™)**
- **Two 32-Bit General-Purpose Timers**
- **Flexible Phase-Locked Loop (PLL) Clock Generator**
- **IEEE-1149.1 (JTAG‡) Boundary-Scan Compatible**
- **352-Pin BGA Package (GJC Suffix)**
- **CMOS Technology**
  - **0.18-µm/5-Level Metal Process**
- **3.3-V I/Os, 1.8-V Internal**

† Component qualification in accordance with JEDEC and industry standards to ensure reliable operation over an extended temperature range. This includes, but is not limited to, Highly Accelerated Stress Test (HAST) or biased 85/85, temperature cycle, autoclave or unbiased HAST, electromigration, bond intermetallic life, and mold compound life. Such qualification testing should not be viewed as justifying use of this component beyond specified performance and environmental limits.
‡ IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

GJC/GJL
352-PIN BALL GRID ARRAY (BGA) PACKAGES
(BOTTOM VIEW)

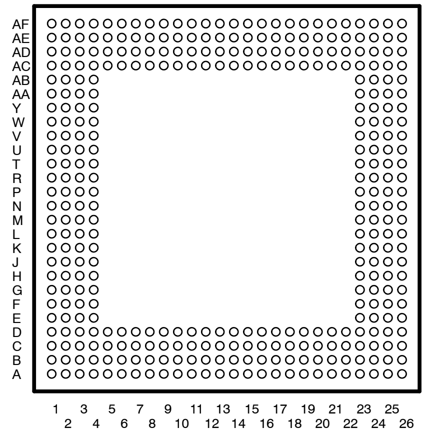


Table of Contents

description

The TMS320C62x™ DSPs (including the SM320C6201-EP†) are the fixed-point DSP family in the TMS320C6000™ DSP platform. The C6201 device is 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. With performance of up to 1600 MIPS at a clock rate of 200 MHz, the C6201 offers cost-effective solutions to high-performance DSP programming challenges. The C6201 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. The processor has 32 general-purpose registers of 32-bit word length and eight highly independent functional units. The eight functional units provide six arithmetic logic units (ALUs) for a high degree of parallelism and two 16-bit multipliers for a 32-bit result. The C6201 can produce two multiply-accumulates (MACs) per cycle—for a total of 466 million MACs per second (MMACS). The C62x™ DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

TMS320C6000, C6000, and C62x are trademarks of Texas Instruments.
Windows is a registered trademark of the Microsoft Corporation.
† The SM320C6201-EP device shall be referred to as C6201 throughout the remainder of this document.

description (continued)

The C6201 includes a large bank of on-chip memory and has a powerful and diverse set of peripherals. Program memory consists of a 64K-byte block that is user-configurable as cache or memory-mapped program space. Data memory of the C6201 consists of two 32K-byte blocks of RAM for improved concurrency. The peripheral set includes two multichannel buffered serial ports (McBSPs), two general-purpose timers, a host-port interface (HPI), and a glueless external memory interface (EMIF) capable of interfacing to SDRAM or SBSRAM and asynchronous peripherals.

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

device characteristics

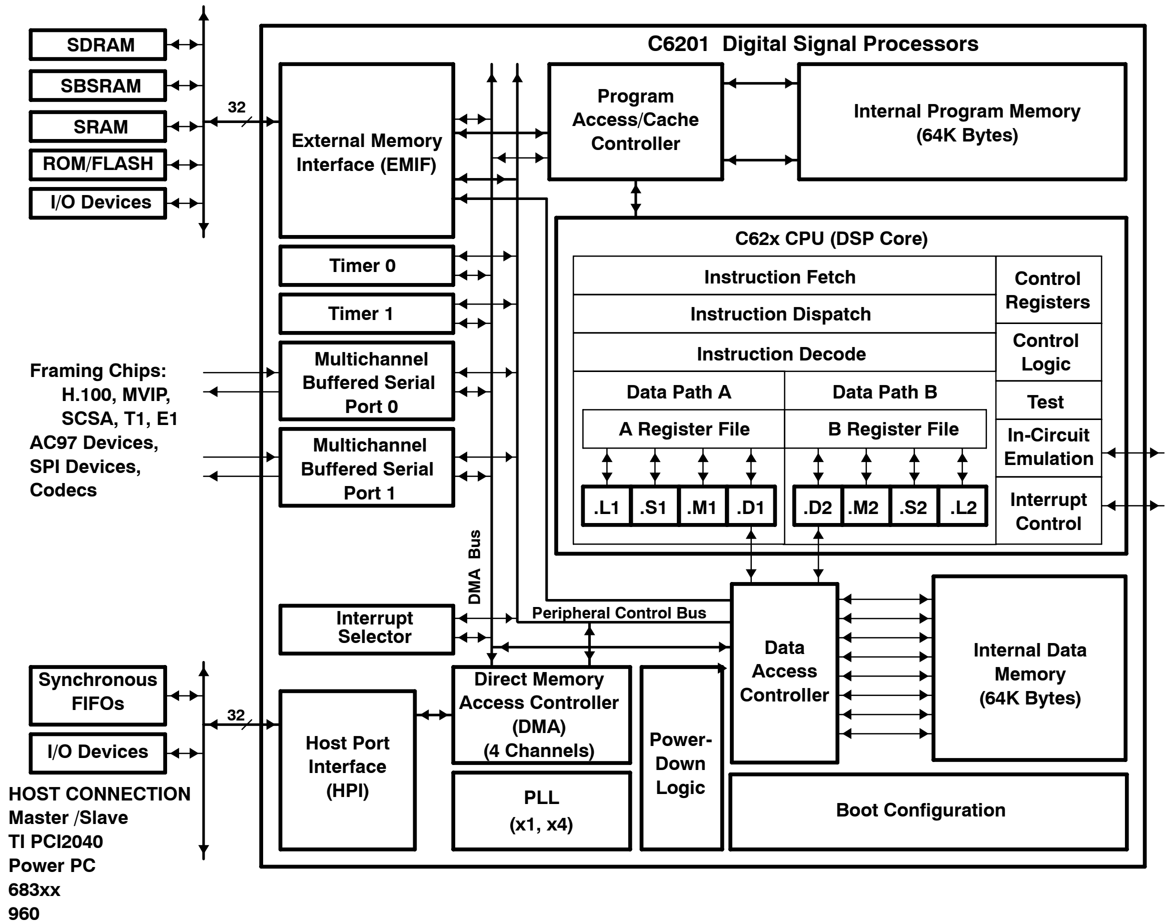
Table 1 provides an overview of the C6201 DSP. The table shows significant features of each device, including the capacity of on-chip RAM, the peripherals, the execution time, and the package type with pin count.

Table 1. Characteristics of the C6201 Processor

HARDWARE FEATURES		C6201 (FIXED-POINT DSP)
Peripherals	EMIF	1
	DMA	1
	HPI	1
	McBSPs	2
	32-Bit Timers	2
On-Chip Memory	Size (Bytes)	72K
	Organization	512-Kbit Program Memory 512-Kbit Data Memory (organized as two blocks)
CPU ID+Rev ID	Control Status Register (CSR.[31:16])	0x0002
Frequency	MHz	200
Cycle Time	ns	5 ns (C6201-200)
Voltage	Core (V)	1.8
	I/O (V)	3.3
PLL Options	CLKIN frequency multiplier	Bypass (x1), x4
BGA Packages	27 x 27 mm	352-Pin BGA (GJL)
	35 x 35 mm	352-Pin BGA (GJC)
Process Technology	μm	0.18 μm
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	PD
Device Part Numbers	(For more details on the C6000™ DSP part numbering, see Figure 4)	SM320C6201GJCA20EP

C6000 is a trademark of Texas Instruments.

functional and CPU (DSP core) block diagram



CPU (DSP core) description

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

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

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

The processing flow begins when a 256-bit-wide instruction fetch packet is fetched from a program memory. The 32-bit instructions destined for the individual functional units are “linked” together by “1” bits in the least significant bit (LSB) position of the instructions. The instructions that are “chained” together for simultaneous execution (up to eight in total) compose an execute packet. A “0” in the LSB of an instruction breaks the chain, effectively placing the instructions that follow it in the next execute packet. If an execute packet crosses the 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 (DSP core) description (continued)

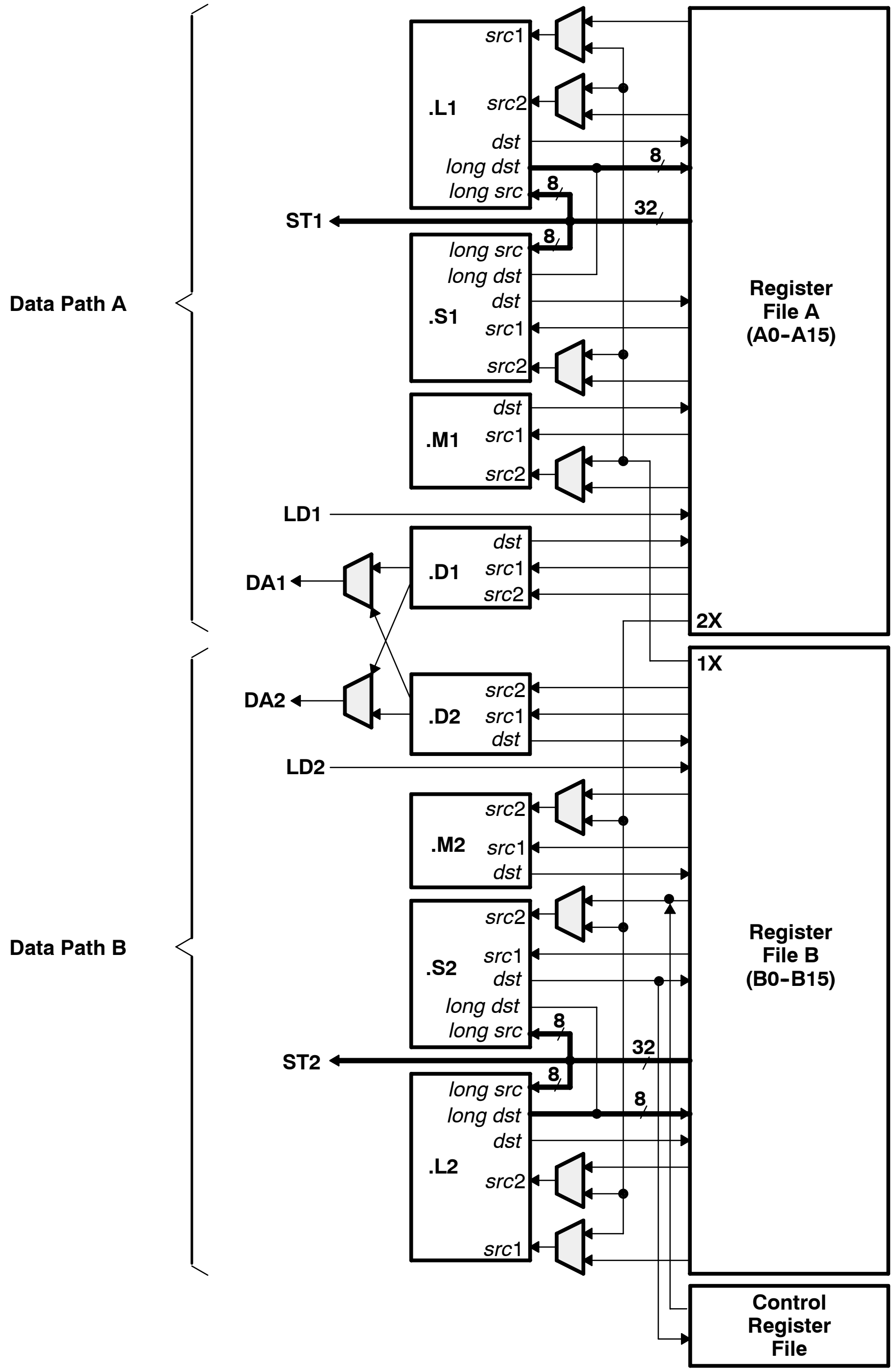


Figure 1. TMS320C62x CPU (DSP Core) Data Paths

signal groups description

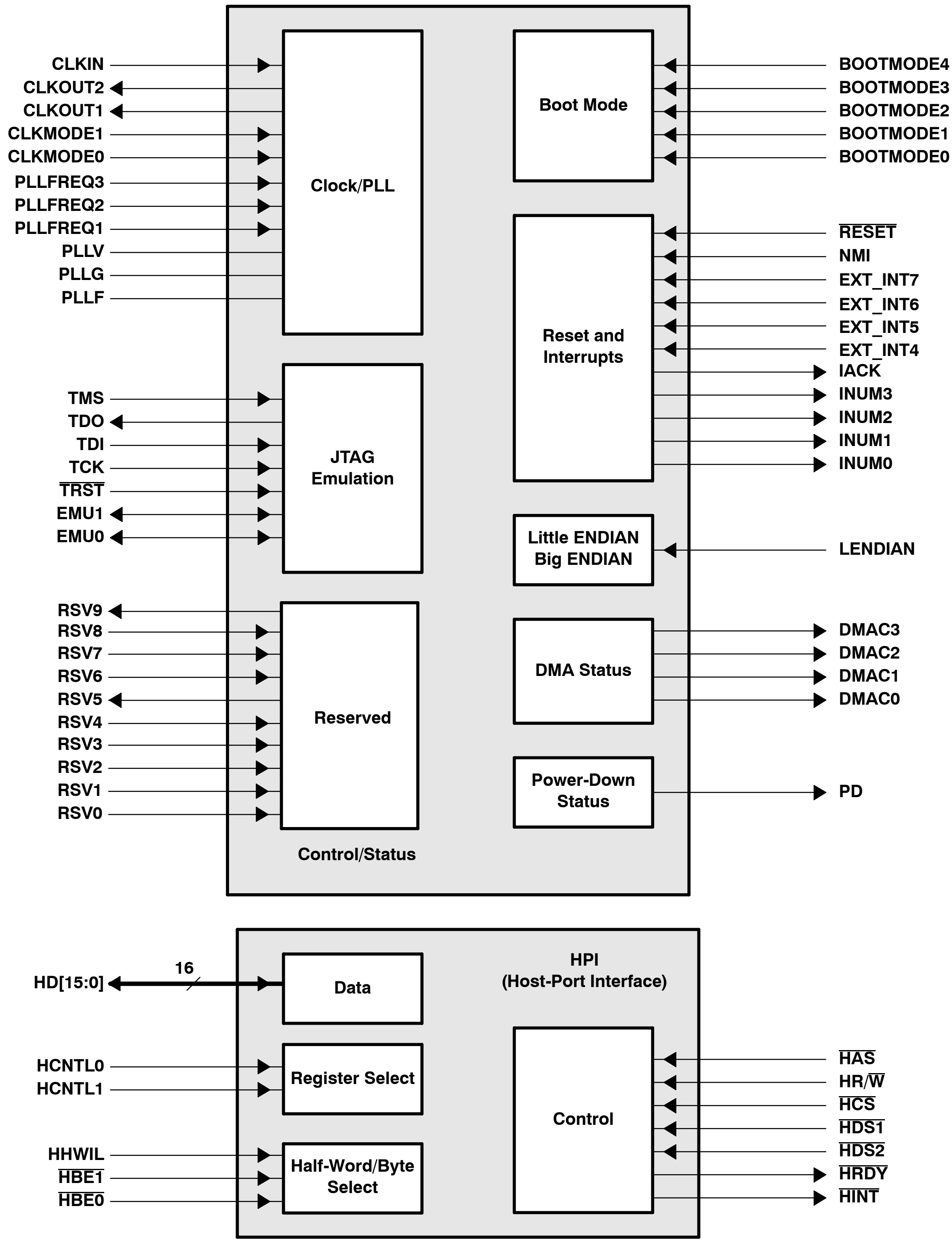


Figure 2. CPU (DSP Core) and Peripheral Signals

signal groups description (continued)

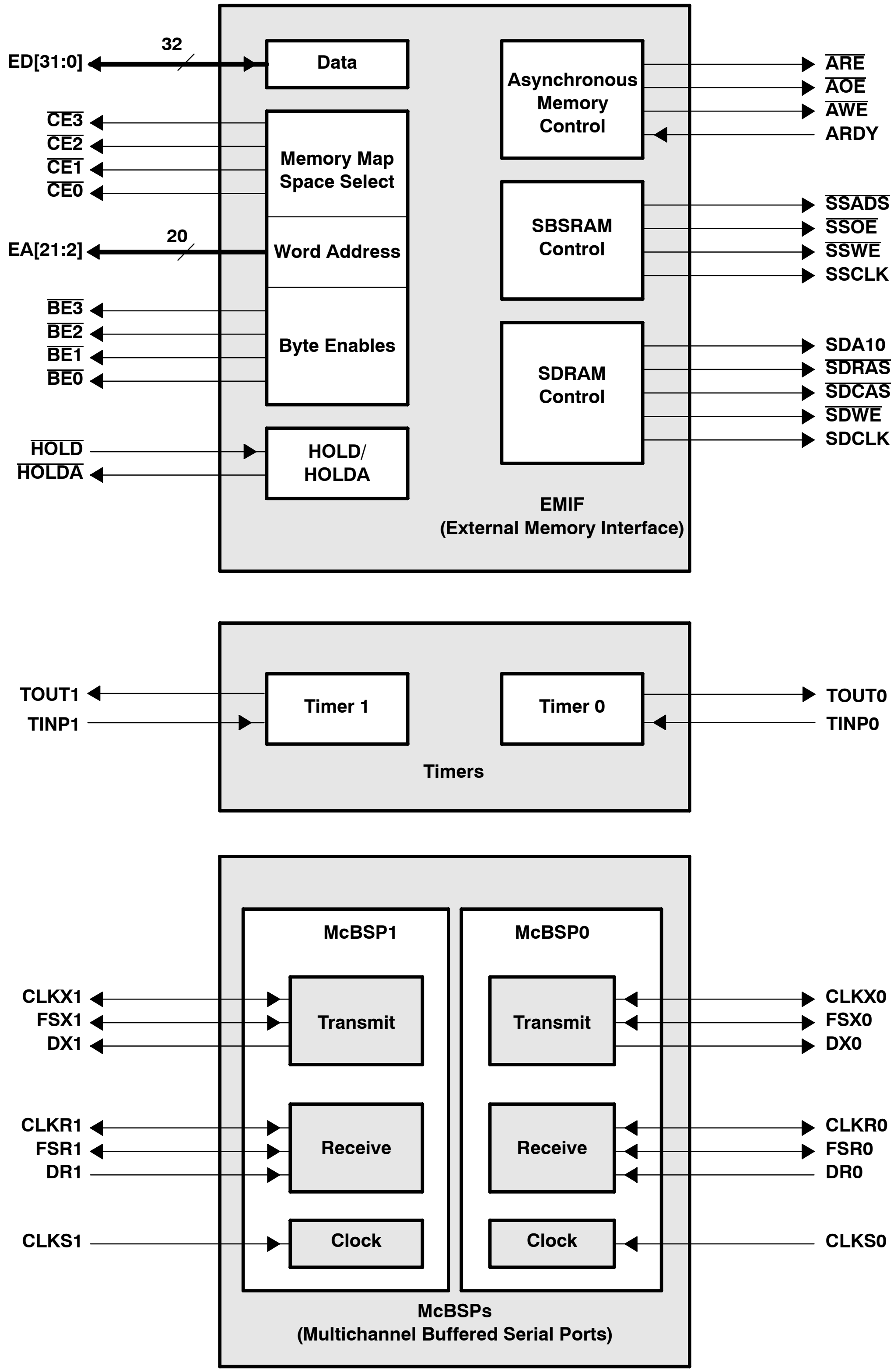


Figure 3. Peripheral Signals

Signal Descriptions

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
CLOCK/PLL			
CLKIN	C10	I	Clock Input
CLKOUT1	AF22	O	Clock output at full device speed
CLKOUT2	AF20	O	Clock output at half of device speed
CLKMODE1	C6	I	Clock mode selects • Selects whether the CPU clock frequency = input clock frequency x4 or x1 For more details on the GJC and GJL CLKMODE pins and the PLL multiply factors, see the *Clock PLL* section of this data sheet.
CLKMODE0	C5		
PLLFREQ3	A9	I	PLL frequency range (3, 2, and 1) • The target range for CLKOUT1 frequency is determined by the 3-bit value of the PLLFREQ pins.
PLLFREQ2	D11		
PLLFREQ1	B10		
PLLV‡	D12	A§	PLL analog V_{CC} connection for the low-pass filter
PLLG‡	C12	A§	PLL analog GND connection for the low-pass filter
PLLF	A11	A§	PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION			
TMS	L3	I	JTAG test port mode select (features an internal pullup)
TDO	W2	O/Z	JTAG test port data out
TDI	R4	I	JTAG test port data in (features an internal pullup)
TCK	R3	I	JTAG test port clock
$\overline{\text{TRST}}$	T1	I	JTAG test port reset (features an internal pulldown)
EMU1	Y1	I/O/Z	Emulation pin 1, pullup with a dedicated 20-kΩ resistor¶
EMU0	W3	I/O/Z	Emulation pin 0, pullup with a dedicated 20-kΩ resistor¶
RESET AND INTERRUPTS			
$\overline{\text{RESET}}$	K2	I	Device reset
NMI	L2	I	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	U3	I	External interrupts • Edge-driven • Polarity independently selected via the external interrupt polarity register bits (EXTPOL.[3:0])
EXT_INT6	V2		
EXT_INT5	W1		
EXT_INT4	U4		
IACK	Y2	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	AA1	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	W4		
INUM1	AA2		
INUM0	AB1		
LITTLE ENDIAN/BIG ENDIAN			
LENDIAN	H3	I	If high, LENDIAN selects little-endian byte/half-word addressing order within a word If low, LENDIAN selects big-endian addressing
POWER-DOWN STATUS			
PD	D3	O	Power-down mode 2 or 3 (active if high)

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
HOST-PORT INTERFACE (HPI)			
$\overline{\text{HINT}}$	H26	O	Host interrupt (from DSP to host)
HCNTL1	F23	I	Host control – selects between control, address, or data registers
HCNTL0	D25	I	Host control – selects between control, address, or data registers
HHWIL	C26	I	Host half-word select – first or second half-word (not necessarily high or low order)
$\overline{\text{HBE1}}$	E23	I	Host byte select within word or half-word
$\overline{\text{HBE0}}$	D24	I	Host byte select within word or half-word
HR/$\overline{\text{W}}$	C23	I	Host read or write select
HD15	B13	I/O/Z	Host-port data (used for transfer of data, address, and control)
HD14	B14		
HD13	C14		
HD12	B15		
HD11	D15		
HD10	B16		
HD9	A17		
HD8	B17		
HD7	D16		
HD6	B18		
HD5	A19		
HD4	C18		
HD3	B19		
HD2	C19		
HD1	B20		
HD0	B21		
$\overline{\text{HAS}}$	C22	I	Host address strobe
$\overline{\text{HCS}}$	B23	I	Host chip select
$\overline{\text{HDS1}}$	D22	I	Host data strobe 1
$\overline{\text{HDS2}}$	A24	I	Host data strobe 2
$\overline{\text{HRDY}}$	J24	O	Host ready (from DSP to host)
BOOT MODE			
BOOTMODE4	D8	I	Boot mode
BOOTMODE3	B4		
BOOTMODE2	A3		
BOOTMODE1	D5		
BOOTMODE0	C4		

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY			
$\overline{CE3}$	AE22	O/Z	Memory space enables • Enabled by bits 24 and 25 of the word address • Only one asserted during any external data access
$\overline{CE2}$	AD26		
$\overline{CE1}$	AB24		
$\overline{CE0}$	AC26		
$\overline{BE3}$	AB25	O/Z	Byte-enable control • Decoded from the two lowest bits of the internal address • Byte-write enables for most types of memory • Can be directly connected to SDRAM read and write mask signal (SDQM)
$\overline{BE2}$	AA24		
$\overline{BE1}$	Y23		
$\overline{BE0}$	AA26		
EMIF – ADDRESS			
EA21	J26	O/Z	External address (word address)
EA20	K25		
EA19	L24		
EA18	K26		
EA17	M26		
EA16	M25		
EA15	P25		
EA14	P24		
EA13	R25		
EA12	T26		
EA11	R23		
EA10	U26		
EA9	U25		
EA8	T23		
EA7	V26		
EA6	V25		
EA5	W26		
EA4	V24		
EA3	W25		
EA2	Y26		

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
EMIF – DATA			
ED31	AB2	I/O/Z	External data
ED30	AC1		
ED29	AA4		
ED28	AD1		
ED27	AC3		
ED26	AD4		
ED25	AF3		
ED24	AE4		
ED23	AD5		
ED22	AF4		
ED21	AE5		
ED20	AD6		
ED19	AE6		
ED18	AD7		
ED17	AC8		
ED16	AF7		
ED15	AD9		
ED14	AD10		
ED13	AF9		
ED12	AC11		
ED11	AE10		
ED10	AE11		
ED9	AF11		
ED8	AE14		
ED7	AF15		
ED6	AE15		
ED5	AF16		
ED4	AC15		
ED3	AE17		
ED2	AF18		
ED1	AF19		
ED0	AC17		
EMIF – ASYNCHRONOUS MEMORY CONTROL			
$\overline{ARE}$	Y24	O/Z	Asynchronous memory read enable
$\overline{AOE}$	AC24	O/Z	Asynchronous memory output enable
$\overline{AWE}$	AD23	O/Z	Asynchronous memory write enable
ARDY	W23	I	Asynchronous memory ready input

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
EMIF – SYNCHRONOUS BURST SRAM (SBSRAM) CONTROL			
$\overline{\text{SSADS}}$	AC20	O/Z	SBSRAM address strobe
$\overline{\text{SSOE}}$	AF21	O/Z	SBSRAM output enable
$\overline{\text{SSWE}}$	AD19	O/Z	SBSRAM write enable
SSCLK	AD17	O	SBSRAM clock
EMIF – SYNCHRONOUS DRAM (SDRAM) CONTROL			
SDA10	AD21	O/Z	SDRAM address 10 (separate for deactivate command)
$\overline{\text{SDRAS}}$	AF24	O/Z	SDRAM row-address strobe
$\overline{\text{SDCAS}}$	AD22	O/Z	SDRAM column-address strobe
$\overline{\text{SDWE}}$	AF23	O/Z	SDRAM write enable
SDCLK	AE20	O	SDRAM clock
EMIF – BUS ARBITRATION			
$\overline{\text{HOLD}}$	AA25	I	Hold request from the host
$\overline{\text{HOLDA}}$	A7	O	Hold-request acknowledge to the host
TIMER1			
TOUT1	H24	O	Timer 1 or general-purpose output
TINP1	K24	I	Timer 1 or general-purpose input
TIMER0			
TOUT0	M4	O	Timer 0 or general-purpose output
TINP0	K4	I	Timer 0 or general-purpose input
DMA ACTION COMPLETE STATUS			
DMAC3	D2	O	DMA action complete
DMAC2	F4		
DMAC1	D1		
DMAC0	E2		
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)			
CLKS1	E25	I	External clock source (as opposed to internal)
CLKR1	H23	I/O/Z	Receive clock
CLKX1	F26	I/O/Z	Transmit clock
DR1	D26	I	Receive data
DX1	G23	O/Z	Transmit data
FSR1	E26	I/O/Z	Receive frame sync
FSX1	F25	I/O/Z	Transmit frame sync

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)			
CLKS0	L4	I	External clock source (as opposed to internal)
CLKR0	M2	I/O/Z	Receive clock
CLKX0	L1	I/O/Z	Transmit clock
DR0	J1	I	Receive data
DX0	R1	O/Z	Transmit data
FSR0	P4	I/O/Z	Receive frame sync
FSX0	P3	I/O/Z	Transmit frame sync
RESERVED FOR TEST			
RSV0	T2	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV1	G2	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV2	C11	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV3	B9	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV4	A6	I	Reserved for testing, ***pulldown*** with a dedicated 20-kΩ resistor
RSV5	C8	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV6	C21	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV7	B22	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV8	A23	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV9	E4	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
UNCONNECTED PINS			
NC	A8 B8 C9 D10 D21 G1 H1 H2 J2 K3 R2		Unconnected pins

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
3.3-V SUPPLY VOLTAGE PINS			
DV_{DD}	A10, A15, A18, A21, A22, B7, C1, D17, F3, G24, G25, H25, J25, L25, M3, N3, N23, R26, T24, U24, W24, Y4, AB3, AB4, AB26, AC6, AC10, AC19, AC21, AC22, AC25, AD11, AD13, AD15, AD18, AE18, AE21, AF5, AF6, AF17	S	3.3-V supply voltage

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
1.8-V SUPPLY VOLTAGE PINS			
CV_{DD}	A5	S	1.8-V supply voltage
	A12		
	A16		
	A20		
	B2		
	B6		
	B11		
	B12		
	B25		
	C3		
	C15		
	C20		
	C24		
	D4		
	D6		
	D7		
	D9		
	D14		
	D18		
	D20		
	D23		
	E1		
	F1		
	H4		
	J4		
	J23		
	K1		
	K23		
	M1		
	M24		
	N4		
	N25		
	P2		
	P23		
	T3		
	T4		
	U1		
	V4		

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
1.8-V SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD}	V23	S	1.8-V supply voltage
	AC4		
	AC9		
	AC12		
	AC13		
	AC18		
	AC23		
	AD3		
	AD8		
	AD14		
	AD24		
	AE2		
	AE8		
	AE12		
	AE25		
	AF12		
	–		
	–		
GROUND PINS			
V_{SS}	A1	GND	Ground pins
	A2		
	A4		
	A13		
	A14		
	A25		
	A26		
	B1		
	B3		
	B5		
	B24		
	B26		
	C2		
	C7		
	C13		
	C16		
	C17		
	C25		
	D13		

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
	D19		
	E3		
	E24		
	F2		
	F24		
	G3		
	G4		
	G26		
	J3		
	L23		
	L26		
	M23		
	N1		
	N2		
	N24		
	N26		
	P1		
	P26		
	R24		
V_{SS}	T25	GND	Ground pins
	U2		
	U23		
	V1		
	V3		
	Y3		
	Y25		
	AA3		
	AA23		
	AB23		
	AC2		
	AC5		
	AC7		
	AC14		
	AC16		
	AD2		
	AD12		
	AD16		
	AD20		

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

Signal Descriptions (Continued)

SIGNAL NAME	PIN NO. GJC	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	AD25	GND	Ground pins
	AE1		
	AE3		
	AE7		
	AE9		
	AE13		
	AE16		
	AE19		
	AE23		
	AE24		
	AE26		
	AF1		
	AF2		
	AF8		
	AF10		
	AF13		
	AF14		
	AF25		
	AF26		

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

development support

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

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

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

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

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

For a complete listing of development-support tools for the TMS320C6000™ DSP platform, visit the Texas Instruments web site on the Worldwide Web at http://www.ti.com uniform resource locator (URL) and under “Development Tools”, select “Digital Signal Processors”. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

device and development support tool nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all TMS320™ DSP family devices and support tools. Each TMS320™ DSP member has one of three prefixes: TMX, TMP, or TMS, and each SMJ320™ DSP member has one of three prefixes: SMX, SM, or SMJ. Texas Instruments recommends two of three possible prefix designators for its 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). This development flow is defined below.

Device development evolutionary flow:

SMX 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

SM/SMJ 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 as well as SM/SMJ devices and TMDS development support tools have been characterized fully, and the quality and reliability of the device has 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, GNM) and temperature range (for example, M). Figure 4 provides a legend for reading the complete device name for many TMS320™ DSP family members.

device and development-support tool nomenclature (continued)

SM 320 C 6201 GJC A 20 EP

PREFIX
- SMX = experimental device
- TMP = prototype device
- TMS = qualified device
- SMJ = MIL-PRF-38535 (QML)
- SMQ = QML Plastic device
- SM = commercial processing

DEVICE FAMILY
- 320 = TMS320 Family

TECHNOLOGY
- C = CMOS
- LC = Low-Voltage CMOS (3.3 V)
- VC = Low-Voltage CMOS [3.3 V (2.5 V or 1.8 V core)]

ENHANCED PLASTIC

SPEED
- 20 = 200 MHz

TEMPERATURE RANGE
- A = Extended temperature (–40°C to 105°C)
- M = Military

PACKAGE TYPE†
- GJC = 352-pin plastic BGA
- GJL = 352-pin plastic BGA
- GFN = 256-pin plastic BGA

DEVICE

6000 DSP:
6201 6203
6701 6711

NOTE: Not all speed, package, process, and temperature combinations are available.

† BGA = Ball Grid Array

Figure 4. TMS320C6000™ Device Nomenclature (Including SM320C6201-EP)

MicroStar BGA is a trademark of Texas Instruments.

documentation support

Extensive documentation supports all TMS320™ DSP family devices from product announcement through applications development. The types of documentation available include: data sheets, such as this document, with design specifications; complete user's reference guides for all devices and tools; technical briefs; development-support tools; on-line help; and hardware and software applications. The following is a brief, descriptive list of support documentation specific to the C6000™ DSP devices:

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

The *TMS320C6000 Peripherals Reference Guide* (literature number SPRU190) describes the functionality of the peripherals available on the C6000™ DSP platform of devices, such as the 64-/32-/16-bit external memory interfaces (EMIFs), 32-/16-bit host-port interfaces (HPIs), multichannel buffered serial ports (McBSPs), direct memory access (DMA), enhanced direct-memory-access (EDMA) controller, expansion bus (XB), peripheral component interconnect (PCI), clocking and phase-locked loop (PLL); and power-down modes. This guide also includes information on internal data and program memories.

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

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

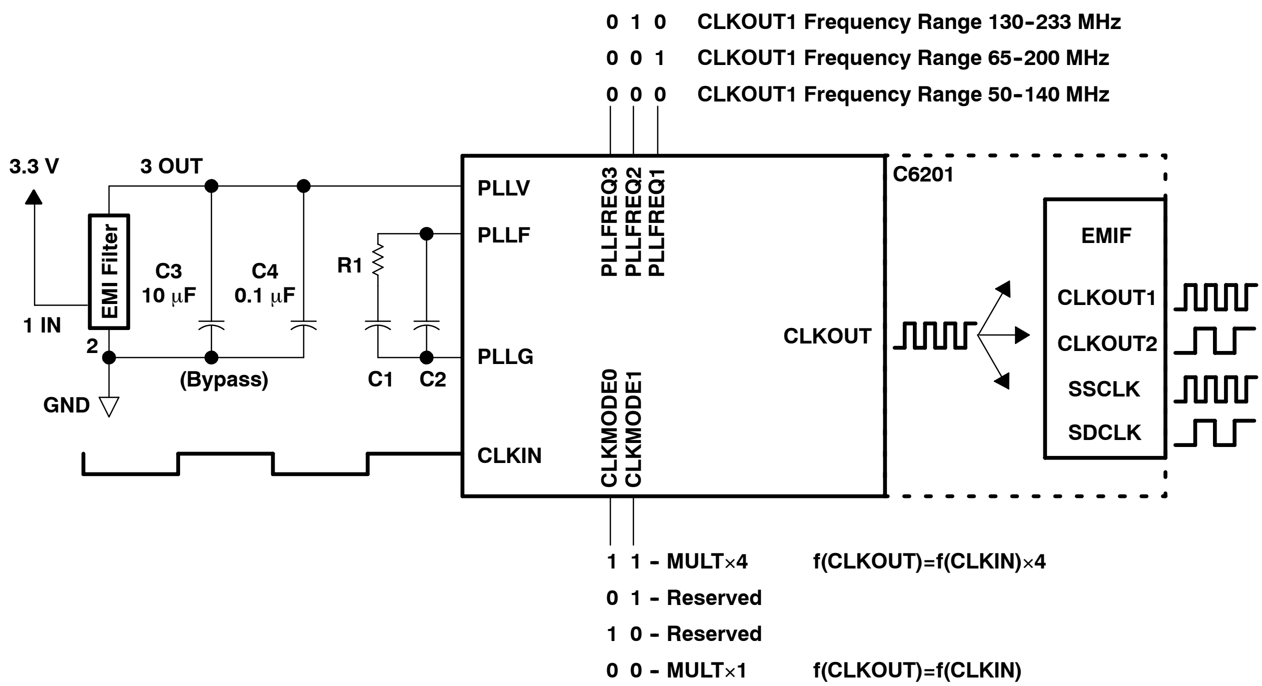
C67x is a trademark of Texas Instruments.

clock PLL

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

To use the PLL to generate the CPU clock, the filter circuit shown in Figure 5 must be properly designed. Note that for C6201, the EMI filter must be powered by the I/O voltage (3.3 V).

To configure the C62x™ PLL clock for proper operation, see Figure 5 and Table 2. To minimize the clock jitter, a single clean power supply should power both the C62x™ DSP device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. See the *input and output clocks* section for input clock timing requirements.



NOTES: A. Keep the lead length and the number of vias between pin PLLF, pin PLLG, R1, C1, and C2 to a minimum. In addition, place all PLL components (R1, C1, C2, C3, C4, and EMI Filter) as close to the C6000™ DSP device as possible. Best performance is achieved with the PLL components on a single side of the board without jumpers, switches, or components other than the ones shown. For CLKMODE x4, values for C1, C2, and R1 are fixed and apply to all valid frequency ranges of CLKIN and CLKOUT.

B. For CLKMODE x1, the PLL is bypassed and all six external PLL components can be removed. For this case, the PLLV terminal has to be connected to a clean supply and the PLLG and PLLF terminals should be tied together.

C. Due to overlap of frequency ranges when choosing the PLLFREQ, more than one frequency range can contain the CLKOUT1 frequency. Choose the lowest frequency range that includes the desired frequency. For example, for CLKOUT1 = 133 MHz, a PLLFREQ value of 000b should be used. For CLKOUT1 = 200 MHz, PLLFREQ should be set to 001b. PLLFREQ values other than 000b, 001b, and 010b are reserved.

D. The 3.3-V supply for the EMI filter (and PLLV) must be from the same 3.3-V power plane supplying the I/O voltage, DV_{DD}.

E. EMI filter manufacturer TDK part number ACF451832-153-T

Figure 5. PLL Block Diagram

clock PLL (continued)

Table 2. 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	12.5–50	50–200	25–100	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

TI DSPs do not require specific power sequencing between the core supply and the I/O supply. However, systems should be designed to ensure that neither supply is powered up for extended periods of time if the other supply is below the proper operating voltage.

system-level design considerations

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

power-supply design considerations

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

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

Core and I/O supply voltage regulators should be located close to the DSP (or DSP array) to minimize inductance and resistance in the power delivery path. Additionally, when designing for high-performance applications utilizing the C6000™ platform of DSPs, the PC board should include separate power planes for core, I/O, and ground, all bypassed with high-quality low-ESL/ESR capacitors.

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

Supply voltage range, CV_{DD} (see Note 1) . . . . −0.3 V to 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 ranges T_C: (A version) . . . . −40°C to 105°C
Storage temperature range, T_{stg} . . . . −65°C to 150°C

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

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

recommended operating conditions

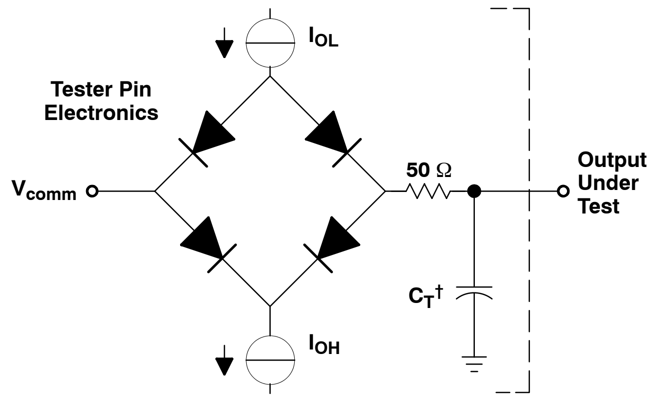
			MIN	NOM	MAX	UNIT
CV_{DD}	Supply voltage		1.71	1.8	1.89	V
DV_{DD}	Supply voltage		3.14	3.30	3.46	V
V_{SS}	Supply ground		0	0	0	V
V_{IH}	High-level input voltage		2			V
V_{IL}	Low-level input voltage				0.8	V
I_{OH}	High-level output current				−12	mA
I_{OL}	Low-level output current				12	mA
T_C	Operating case temperature	A version	−40		105	°C

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

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	DV_{DD} = MIN, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN, I_{OL} = MAX			0.6	V
I_I	Input current‡	$V_I = V_{SS}$ to DV_{DD}			±10	uA
I_{OZ}	Off-state output current	$V_O = DV_{DD}$ or 0 V			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access§	CV_{DD} = NOM, CPU clock = 167 MHz		380		mA
I_{DD2V}	Supply current, peripherals§	CV_{DD} = NOM, CPU clock = 167 MHz		240		mA
I_{DD3V}	Supply current, I/O pins§	DV_{DD} = NOM, CPU clock = 167 MHz		90		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 details on CPU, peripheral, and I/O activity, see the *TMS320C6000 Power Consumption Summary* application report (literature number SPRA486).

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 2 mA
I_{OH} = 2 mA
V_{comm} = 0.8 V
C_T = 15–30-pF typical load-circuit capacitance

† Typical distributed load circuit capacitance

Figure 6. TTL-Level Outputs

signal transition levels

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

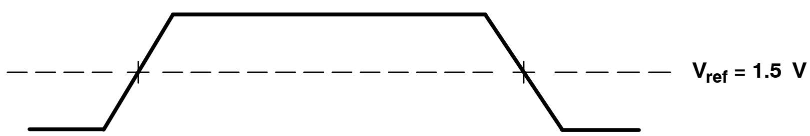


Figure 7. Input and Output Voltage Reference Levels for AC Timing Measurements

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

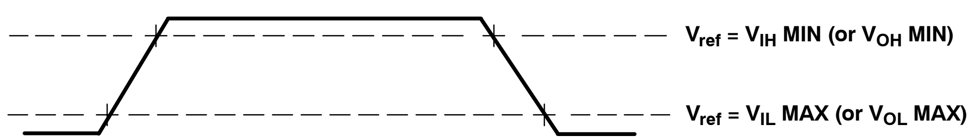


Figure 8. Rise and Fall Transition Time Voltage Reference Levels

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN†‡ (see Figure 9)

NO.			-200				UNIT
			CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	20		5		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.45C		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		0.6	ns

† The reference points for the rise and fall transitions are measured at V_{IL} MAX and V_{IH} MIN.
‡ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 50 MHz, use C = 20 ns.

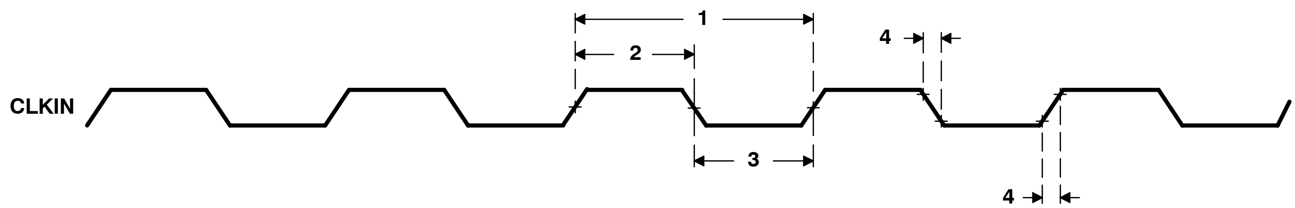


Figure 9. CLKIN Timing Diagram

switching characteristics over recommended operating conditions for CLKOUT1§¶# (see Figure 10)

NO.	PARAMETER		-200				UNIT
			CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	
1	$t_{c(CKO1)}$	Cycle time, CLKOUT1	P – 0.7	P + 0.7	P – 0.7	P + 0.7	ns
2	$t_{w(CKO1H)}$	Pulse duration, CLKOUT1 high	(P/2) – 0.5	(P/2) + 0.5	PH – 0.5	PH + 0.5	ns
3	$t_{w(CKO1L)}$	Pulse duration, CLKOUT1 low	(P/2) – 0.5	(P/2) + 0.5	PL – 0.5	PL + 0.5	ns
4	$t_{t(CKO1)}$	Transition time, CLKOUT1		0.6		0.6	ns

§ P = 1/CPU clock frequency in ns.
¶ The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.
# PH is the high period of CLKIN in ns and PL is the low period of CLKIN in ns.

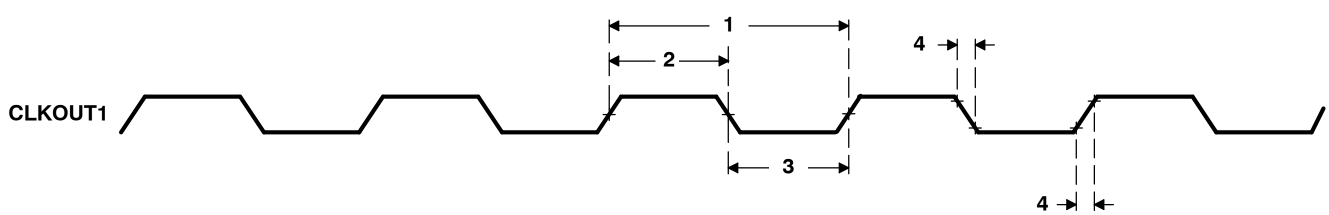


Figure 10. CLKOUT1 Timing Diagram

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics over recommended operating conditions for CLKOUT2†‡ (see Figure 11)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	2P – 0.7	2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	P – 0.7	P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	P – 0.7	P + 0.7	ns
4	$t_{t(CKO2)}$	Transition time, CLKOUT2		0.6	ns

† P = 1/CPU clock frequency in ns.
‡ The reference points for the rise and fall transitions are measured at V_{OL} MAX and V_{OH} MIN.

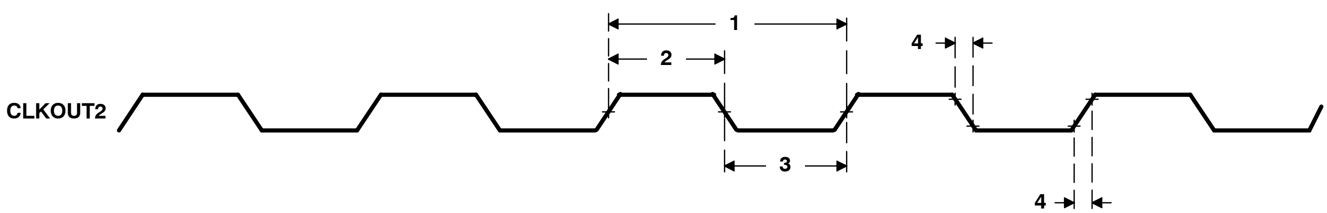


Figure 11. CLKOUT2 Timings

SDCLK, SSCLK timing parameters

SDCLK timing parameters are the same as CLKOUT2 parameters.

SSCLK timing parameters are the same as CLKOUT1 or CLKOUT2 parameters, depending on SSCLK configuration.

switching characteristics over recommended operating conditions for the relation of SSCLK, SDCLK, and CLKOUT2 to CLKOUT1 (see Figure 12)†

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
1	$t_{d(CKO1-SSCLK)}$	Delay time, CLKOUT1 edge to SSCLK edge	(P/2) + 0.2	(P/2) + 4.2	ns
2	$t_{d(CKO1-SSCLK1/2)}$	Delay time, CLKOUT1 edge to SSCLK edge (1/2 clock rate)	(P/2) – 1	(P/2) + 2.4	ns
3	$t_{d(CKO1-CKO2)}$	Delay time, CLKOUT1 edge to CLKOUT2 edge	(P/2) – 1	(P/2) + 2.4	ns
4	$t_{d(CKO1-SDCLK)}$	Delay time, CLKOUT1 edge to SDCLK edge	(P/2) – 1	(P/2) + 2.4	ns

† P = 1/CPU clock frequency in ns.

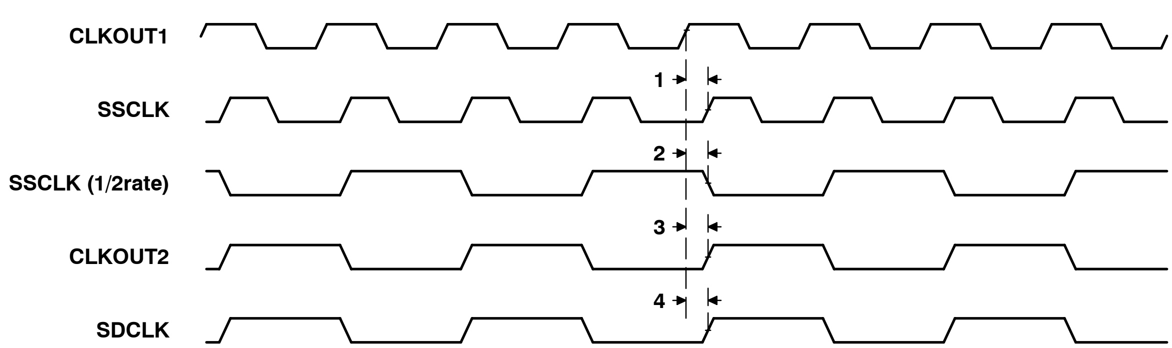


Figure 12. Relation of CLKOUT2, SDCLK, and SSCLK to CLKOUT1

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles† (see Figure 13 and Figure 14)

NO.			-200 MIN	-200 MAX	UNIT
6	$t_{su(EDV\text{-}CKO1H)}$	Setup time, read EDx valid before CLKOUT1 high	4		ns
7	$t_{h(CKO1H\text{-}EDV)}$	Hold time, read EDx valid after CLKOUT1 high	0.8		ns
10	$t_{su(ARDY\text{-}CKO1H)}$	Setup time, ARDY valid before CLKOUT1 high	3		ns
11	$t_{h(CKO1H\text{-}ARDY)}$	Hold time, ARDY valid after CLKOUT1 high	1.8		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.

switching characteristics over recommended operating conditions for asynchronous memory cycles‡ (see Figure 13 and Figure 14)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
1	$t_{d(CKO1H\text{-}CEV)}$	Delay time, CLKOUT1 high to $\overline{\text{CEx}}$ valid	–0.2	4	ns
2	$t_{d(CKO1H\text{-}BEV)}$	Delay time, CLKOUT1 high to $\overline{\text{BEx}}$ valid		4	ns
3	$t_{d(CKO1H\text{-}BEIV)}$	Delay time, CLKOUT1 high to $\overline{\text{BEx}}$ invalid	–0.2		ns
4	$t_{d(CKO1H\text{-}EAV)}$	Delay time, CLKOUT1 high to EAx valid		4	ns
5	$t_{d(CKO1H\text{-}EAIV)}$	Delay time, CLKOUT1 high to EAx invalid	–0.2		ns
8	$t_{d(CKO1H\text{-}AOEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AOE}}$ valid	–0.2	4	ns
9	$t_{d(CKO1H\text{-}AREV)}$	Delay time, CLKOUT1 high to $\overline{\text{ARE}}$ valid	–0.2	4	ns
12	$t_{d(CKO1H\text{-}EDV)}$	Delay time, CLKOUT1 high to EDx valid		4	ns
13	$t_{d(CKO1H\text{-}EDIV)}$	Delay time, CLKOUT1 high to EDx invalid	–0.2		ns
14	$t_{d(CKO1H\text{-}AWEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AWE}}$ valid	–0.2	4	ns

‡ The minimum delay is also the minimum output hold after CLKOUT1 high.

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

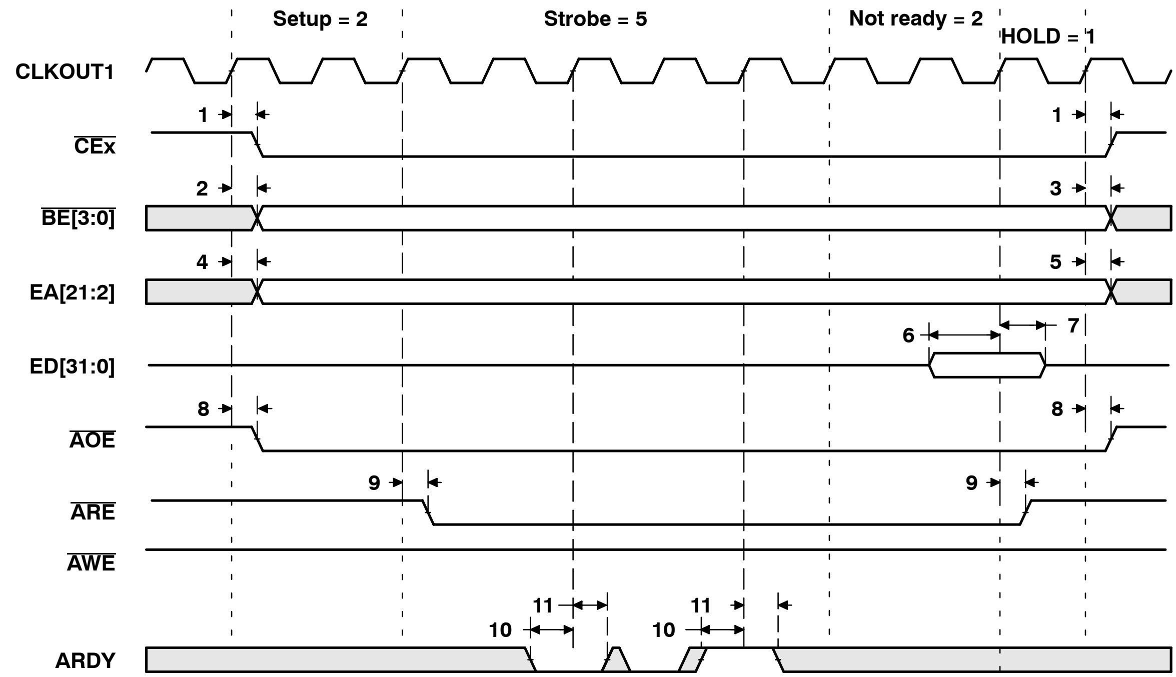


Figure 13. Asynchronous Memory Read Timing

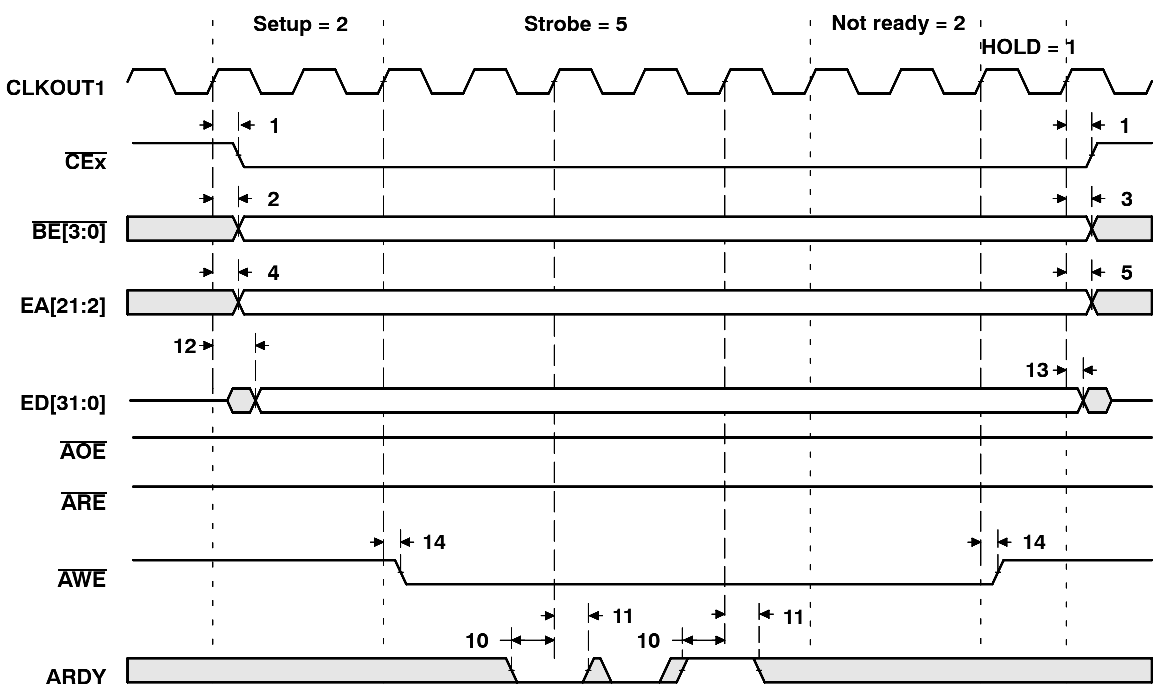


Figure 14. Asynchronous Memory Write Timing

SYNCHRONOUS-BURST MEMORY TIMING

timing requirements for synchronous-burst SRAM cycles (full-rate SSCLK) (see Figure 15)

NO.			-200 MIN	-200 MAX	UNIT
7	$t_{su(EDV-SSCLKH)}$	Setup time, read EDx valid before SSCLK high	1.5		ns
8	$t_{h(SSCLKH-EDV)}$	Hold time, read EDx valid after SSCLK high	1.5		ns

switching characteristics over recommended operating conditions for synchronous-burst SRAM cycles† (full-rate SSCLK) (see Figure 15 and Figure 16)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
1	$t_{osu(CEV-SSCLKH)}$	Output setup time, $\overline{\text{CEx}}$ valid before SSCLK high	0.5P – 1.3		ns
2	$t_{oh(SSCLKH-CEV)}$	Output hold time, $\overline{\text{CEx}}$ valid after SSCLK high	0.5P – 2.3		ns
3	$t_{osu(BEV-SSCLKH)}$	Output setup time, $\overline{\text{BEx}}$ valid before SSCLK high	0.5P – 1.3		ns
4	$t_{oh(SSCLKH-BEIV)}$	Output hold time, $\overline{\text{BEx}}$ invalid after SSCLK high	0.5P – 2.3		ns
5	$t_{osu(EAV-SSCLKH)}$	Output setup time, EAx valid before SSCLK high	0.5P – 1.3		ns
6	$t_{oh(SSCLKH-EAIV)}$	Output hold time, EAx invalid after SSCLK high	0.5P – 2.3		ns
9	$t_{osu(ADSV-SSCLKH)}$	Output setup time, $\overline{\text{SSADS}}$ valid before SSCLK high	0.5P – 1.3		ns
10	$t_{oh(SSCLKH-ADSV)}$	Output hold time, $\overline{\text{SSADS}}$ valid after SSCLK high	0.5P – 2.3		ns
11	$t_{osu(OEV-SSCLKH)}$	Output setup time, $\overline{\text{SSOE}}$ valid before SSCLK high	0.5P – 1.3		ns
12	$t_{oh(SSCLKH-OEV)}$	Output hold time, $\overline{\text{SSOE}}$ valid after SSCLK high	0.5P – 2.3		ns
13	$t_{osu(EDV-SSCLKH)}$	Output setup time, EDx valid before SSCLK high	0.5P – 1.3		ns
14	$t_{oh(SSCLKH-EDIV)}$	Output hold time, EDx invalid after SSCLK high	0.5P – 2.3		ns
15	$t_{osu(WEV-SSCLKH)}$	Output setup time, $\overline{\text{SSWE}}$ valid before SSCLK high	0.5P – 1.3		ns
16	$t_{oh(SSCLKH-WEV)}$	Output hold time, $\overline{\text{SSWE}}$ valid after SSCLK high	0.5P – 2.3		ns

† When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
For CLKMODE x1, 0.5P is defined as PH (pulse duration of CLKIN high) for all output setup times; 0.5P is defined as PL (pulse duration of CLKIN low) for all output hold times.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

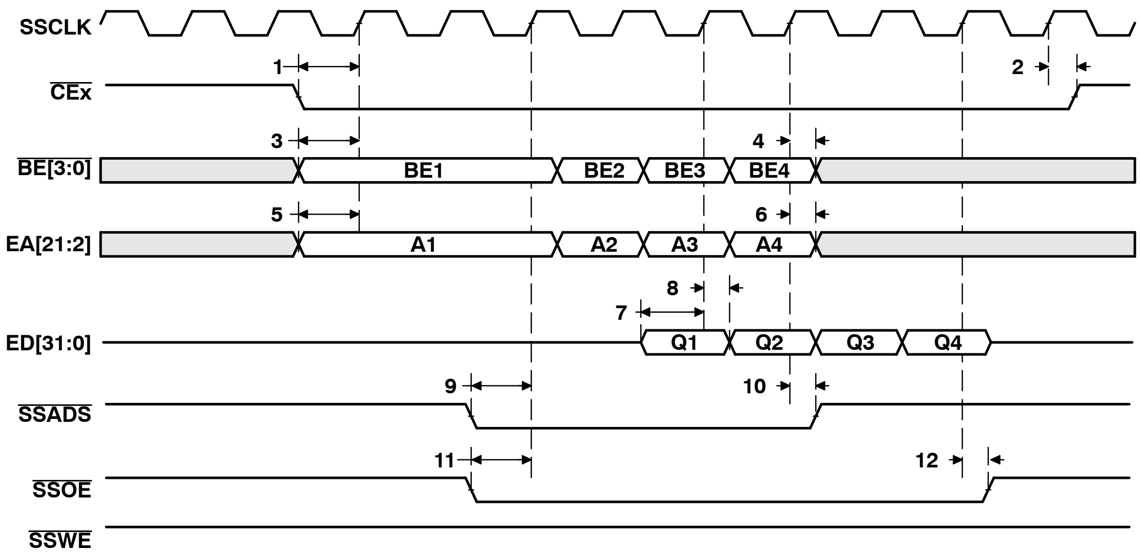


Figure 15. SBSRAM Read Timing (Full-Rate SSCLK)

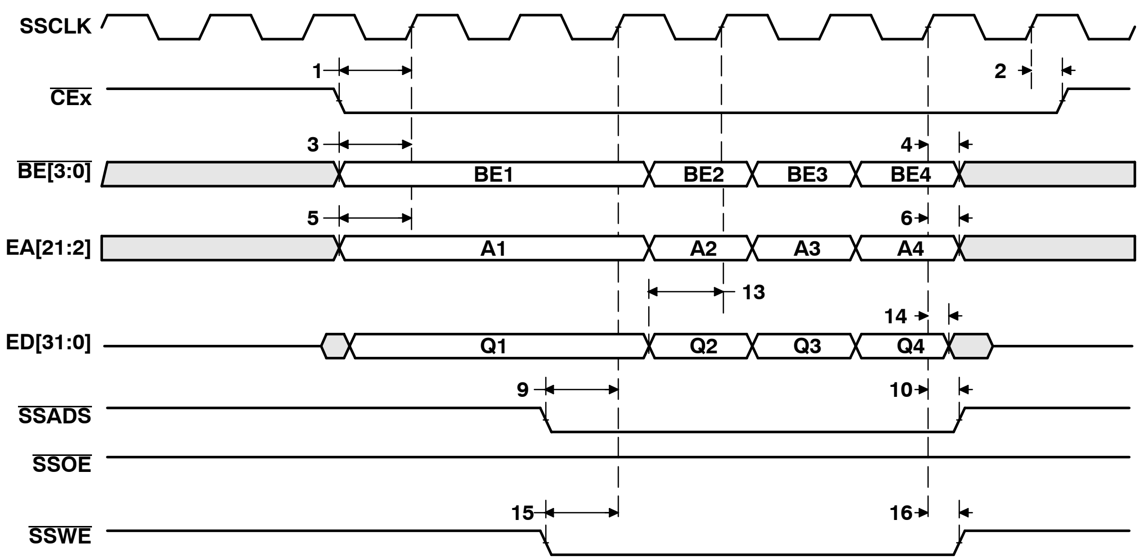


Figure 16. SBSRAM Write Timing (Full-Rate SSCLK)

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

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

NO.			-200		UNIT
			MIN	MAX	
7	$t_{su(EDV-SSCLKH)}$	Setup time, read EDx valid before SSCLK high	2.5		ns
8	$t_{h(SSCLKH-EDV)}$	Hold time, read EDx valid after SSCLK high	1.5		ns

switching characteristics over recommended operating conditions for synchronous-burst SRAM cycles† (half-rate SSCLK) (see Figure 17 and Figure 18)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
1	$t_{osu(CEV-SSCLKH)}$	Output setup time, $\overline{\text{CEx}}$ valid before SSCLK high	1.5P – 3		ns
2	$t_{oh(SSCLKH-CEV)}$	Output hold time, $\overline{\text{CEx}}$ valid after SSCLK high	0.5P – 1.5		ns
3	$t_{osu(BEV-SSCLKH)}$	Output setup time, $\overline{\text{BEx}}$ valid before SSCLK high	1.5P – 3		ns
4	$t_{oh(SSCLKH-BEIV)}$	Output hold time, $\overline{\text{BEx}}$ invalid after SSCLK high	0.5P – 1.5		ns
5	$t_{osu(EAV-SSCLKH)}$	Output setup time, EAx valid before SSCLK high	1.5P – 3		ns
6	$t_{oh(SSCLKH-EAIV)}$	Output hold time, EAx invalid after SSCLK high	0.5P – 1.5		ns
9	$t_{osu(ADSV-SSCLKH)}$	Output setup time, $\overline{\text{SSADS}}$ valid before SSCLK high	1.5P – 3		ns
10	$t_{oh(SSCLKH-ADSV)}$	Output hold time, $\overline{\text{SSADS}}$ valid after SSCLK high	0.5P – 1.5		ns
11	$t_{osu(OEV-SSCLKH)}$	Output setup time, $\overline{\text{SSOE}}$ valid before SSCLK high	1.5P – 3		ns
12	$t_{oh(SSCLKH-OEV)}$	Output hold time, $\overline{\text{SSOE}}$ valid after SSCLK high	0.5P – 1.5		ns
13	$t_{osu(EDV-SSCLKH)}$	Output setup time, EDx valid before SSCLK high	1.5P – 3		ns
14	$t_{oh(SSCLKH-EDIV)}$	Output hold time, EDx invalid after SSCLK high	0.5P – 1.5		ns
15	$t_{osu(WEV-SSCLKH)}$	Output setup time, $\overline{\text{SSWE}}$ valid before SSCLK high	1.5P – 3		ns
16	$t_{oh(SSCLKH-WEV)}$	Output hold time, $\overline{\text{SSWE}}$ valid after SSCLK high	0.5P – 1.5		ns

† When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
For CLKMODE x1:
1.5P = P + PH, where P = 1/CPU clock frequency, and PH = pulse duration of CLKIN high.
0.5P = PL, where PL = pulse duration of CLKIN low.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

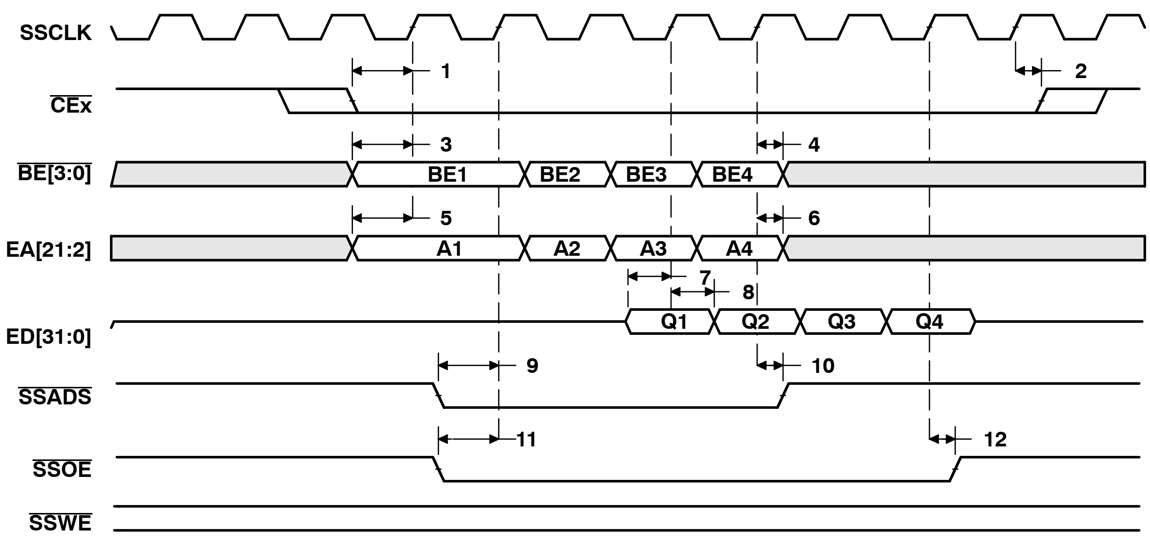


Figure 17. SBSRAM Read Timing (1/2 Rate SSCLK)

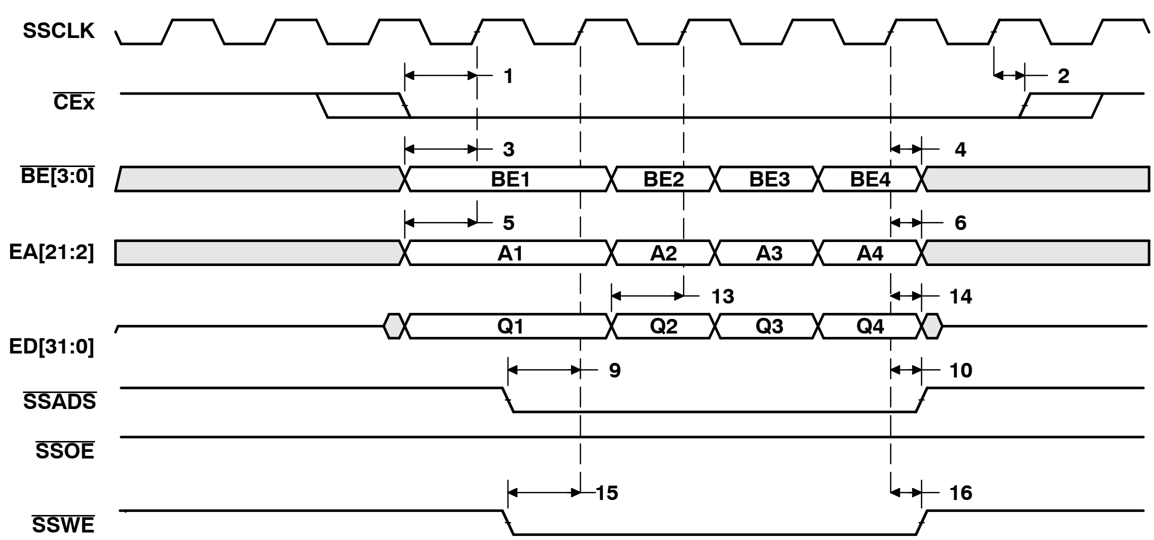


Figure 18. SBSRAM Write Timing (1/2 Rate SSCLK)

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles (see Figure 19)

NO.			-200 MIN	-200 MAX	UNIT
7	$t_{su(EDV-SDCLKH)}$	Setup time, read EDx valid before SDCLK high	0.5		ns
8	$t_{h(SDCLKH-EDV)}$	Hold time, read EDx valid after SDCLK high	3		ns

switching characteristics over recommended operating conditions for synchronous DRAM cycles† (see Figure 19–Figure 24)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
1	$t_{osu(CEV-SDCLKH)}$	Output setup time, $\overline{\text{CEx}}$ valid before SDCLK high	1.5P – 3.5		ns
2	$t_{oh(SDCLKH-CEV)}$	Output hold time, $\overline{\text{CEx}}$ valid after SDCLK high	0.5P – 1		ns
3	$t_{osu(BEV-SDCLKH)}$	Output setup time, $\overline{\text{BEx}}$ valid before SDCLK high	1.5P – 3.5		ns
4	$t_{oh(SDCLKH-BEIV)}$	Output hold time, $\overline{\text{BEx}}$ invalid after SDCLK high	0.5P – 1		ns
5	$t_{osu(EAV-SDCLKH)}$	Output setup time, EAx valid before SDCLK high	1.5P – 3.5		ns
6	$t_{oh(SDCLKH-EAIV)}$	Output hold time, EAx invalid after SDCLK high	0.5P – 1		ns
9	$t_{osu(SDCAS-SDCLKH)}$	Output setup time, $\overline{\text{SDCAS}}$ valid before SDCLK high	1.5P – 3.5		ns
10	$t_{oh(SDCLKH-SDCAS)}$	Output hold time, $\overline{\text{SDCAS}}$ valid after SDCLK high	0.5P – 1		ns
11	$t_{osu(EDV-SDCLKH)}$	Output setup time, EDx valid before SDCLK high	1.5P – 3.5		ns
12	$t_{oh(SDCLKH-EDIV)}$	Output hold time, EDx invalid after SDCLK high	0.5P – 1		ns
13	$t_{osu(SDWE-SDCLKH)}$	Output setup time, $\overline{\text{SDWE}}$ valid before SDCLK high	1.5P – 3.5		ns
14	$t_{oh(SDCLKH-SDWE)}$	Output hold time, $\overline{\text{SDWE}}$ valid after SDCLK high	0.5P – 1		ns
15	$t_{osu(SDA10V-SDCLKH)}$	Output setup time, SDA10 valid before SDCLK high	1.5P – 3.5		ns
16	$t_{oh(SDCLKH-SDA10IV)}$	Output hold time, SDA10 invalid after SDCLK high	0.5P – 1		ns
17	$t_{osu(SDRAS-SDCLKH)}$	Output setup time, $\overline{\text{SDRAS}}$ valid before SDCLK high	1.5P – 3.5		ns
18	$t_{oh(SDCLKH-SDRAS)}$	Output hold time, $\overline{\text{SDRAS}}$ valid after SDCLK high	0.5P – 1		ns

† When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
For CLKMODE x1:
1.5P = P + PH, where P = 1/CPU clock frequency, and PH = pulse duration of CLKIN high.
0.5P = PL, where PL = pulse duration of CLKIN low.

SYNCHRONOUS DRAM TIMING (CONTINUED)

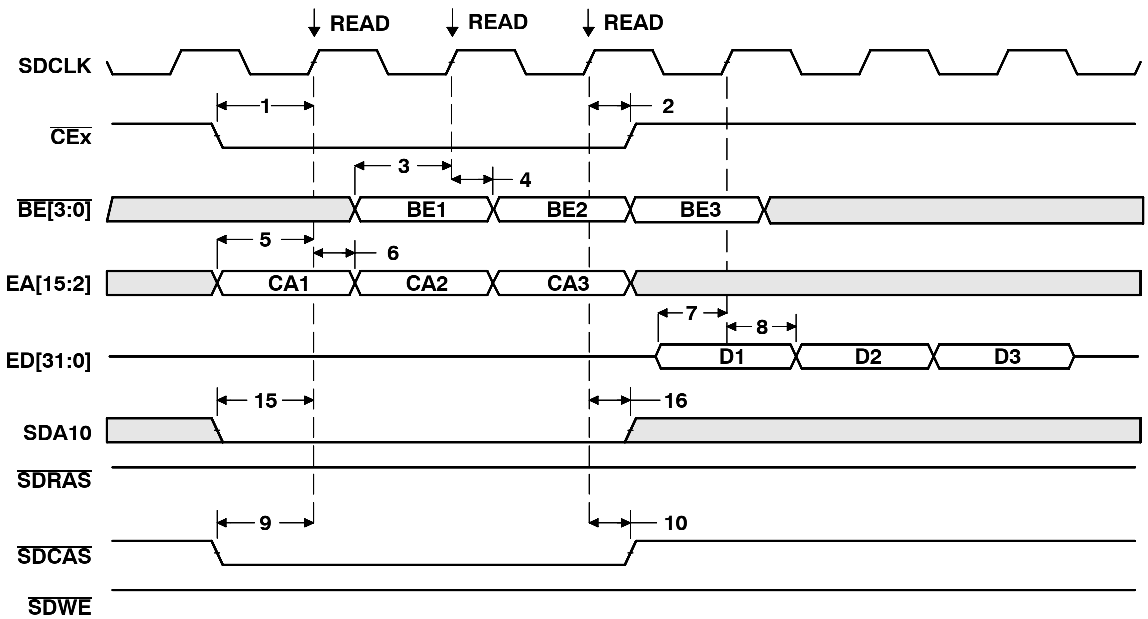


Figure 19. Three SDRAM Read Commands

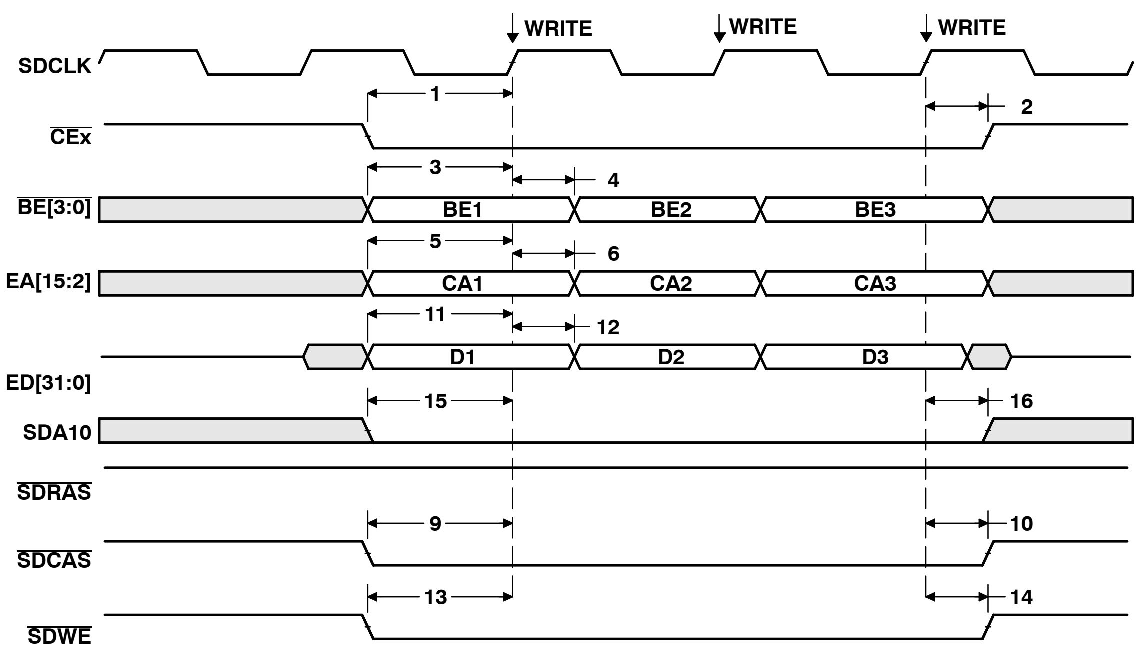


Figure 20. Three SDRAM WRT Commands

SYNCHRONOUS DRAM TIMING (CONTINUED)

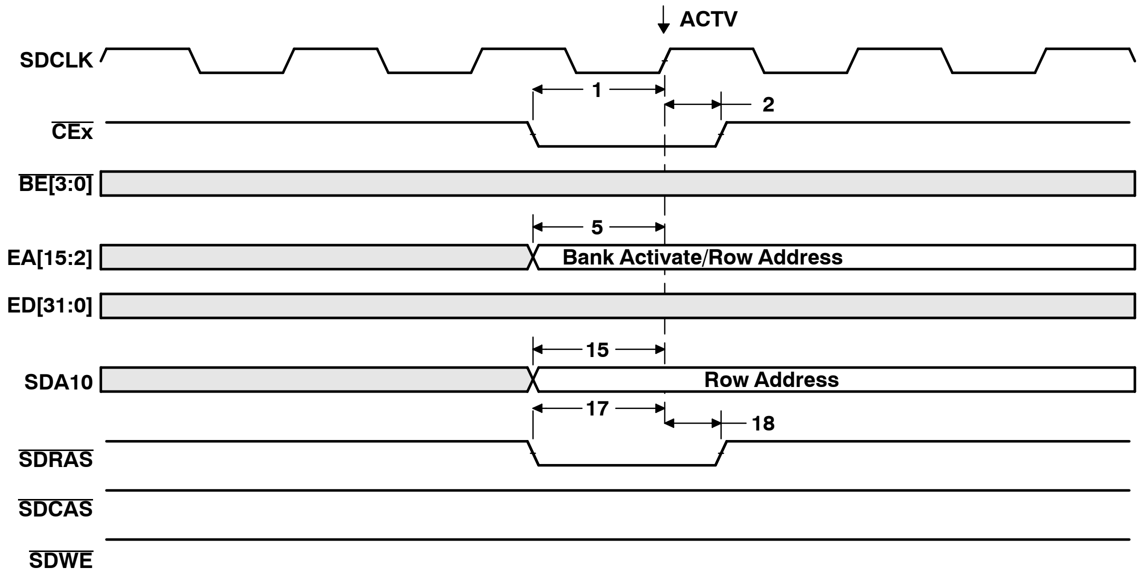


Figure 21. SDRAM ACTV Command

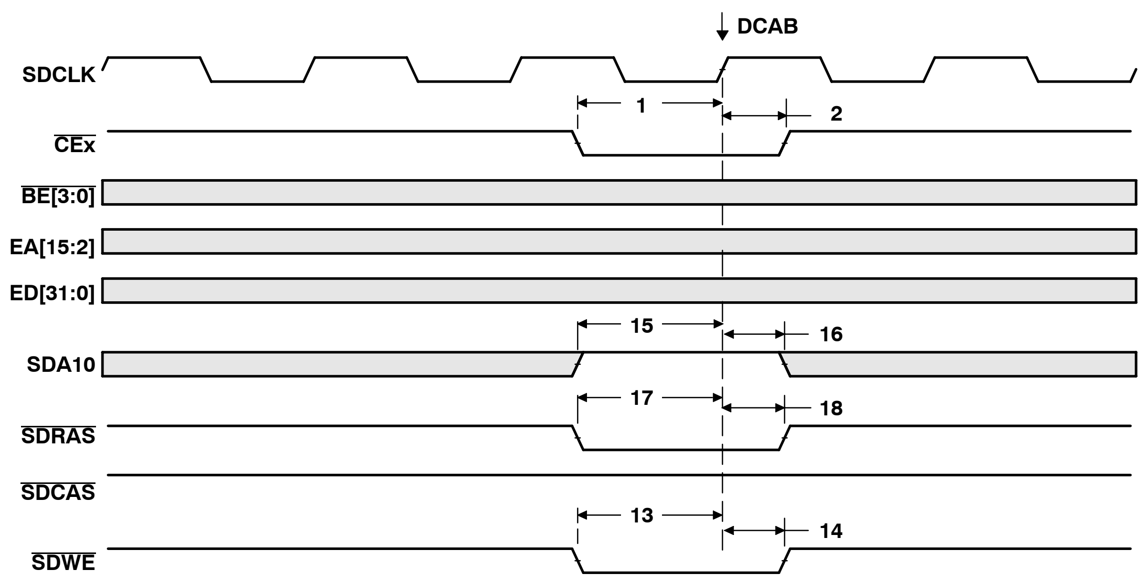


Figure 22. SDRAM DCAB Command

SYNCHRONOUS DRAM TIMING (CONTINUED)

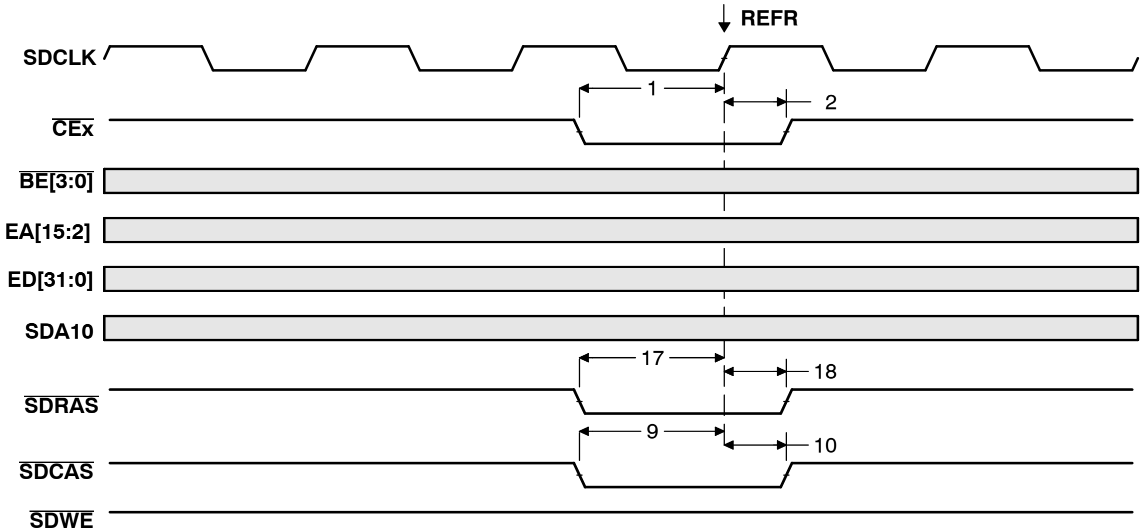


Figure 23. SDRAM REFR Command

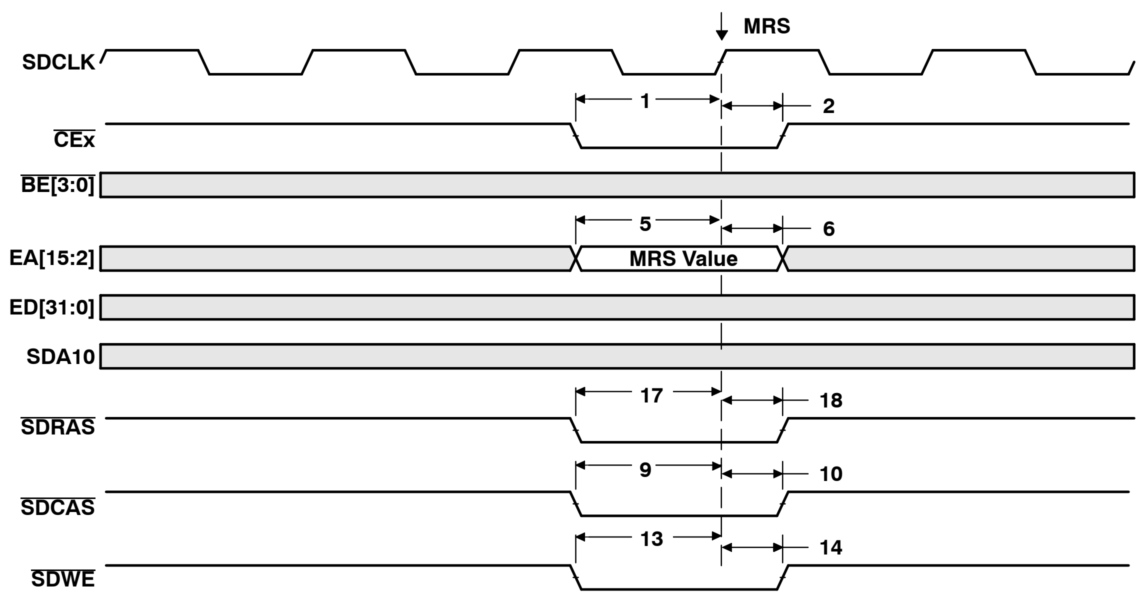


Figure 24. SDRAM MRS Command

$\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ TIMING

timing requirements for the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ cycles† (see Figure 25)

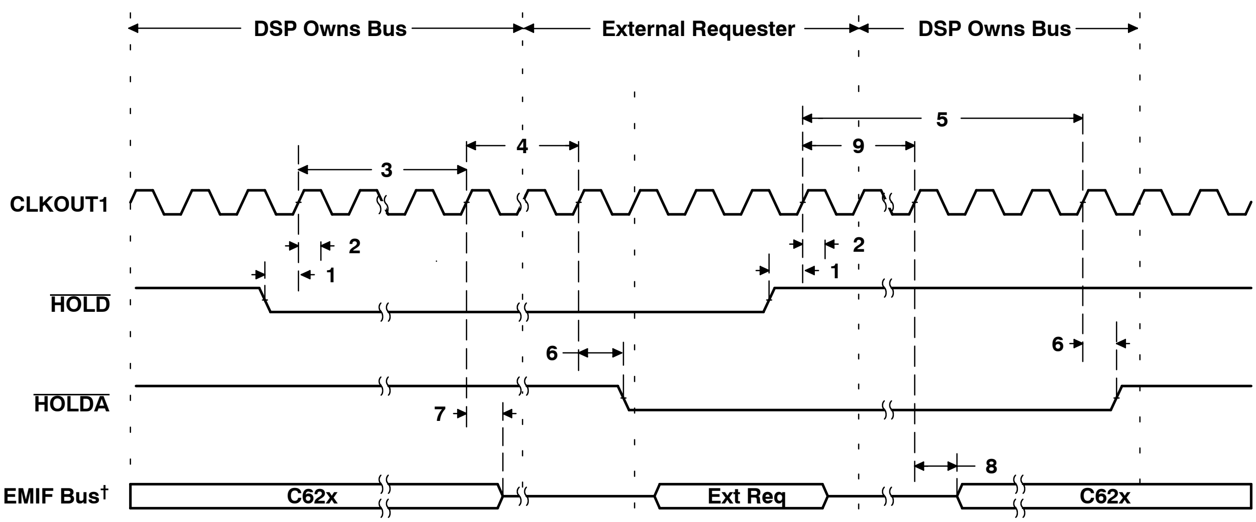
NO.			-200 MIN	-200 MAX	UNIT
1	$t_{su(HOLDH-CKO1H)}$	Setup time, $\overline{\text{HOLD}}$ high before CLKOUT1 high	1		ns
2	$t_{h(CKO1H-HOLDL)}$	Hold time, $\overline{\text{HOLD}}$ low after CLKOUT1 high	4		ns

† $\overline{\text{HOLD}}$ is synchronized internally. Therefore, if setup and hold times are not met, it will either be recognized in the current cycle or in the next cycle. Thus, $\overline{\text{HOLD}}$ can be an asynchronous input.

switching characteristics over recommended operating conditions for the $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ cycles‡ (see Figure 25)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
3	$t_{d(HOLDL-BHZ)}$	Delay time, $\overline{\text{HOLD}}$ low to EMIF Bus high impedance	4P	§	ns
4	$t_{d(BHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to $\overline{\text{HOLDA}}$ low	P	2P	ns
5	$t_{d(HOLDH-HOLDAH)}$	Delay time, $\overline{\text{HOLD}}$ high to $\overline{\text{HOLDA}}$ high	4P	7P	ns
6	$t_{d(CKO1H-HOLDAL)}$	Delay time, CLKOUT1 high to $\overline{\text{HOLDA}}$ valid	1	8	ns
7	$t_{d(CKO1H-BHZ)}$	Delay time, CLKOUT1 high to EMIF Bus high impedance¶	3	11	ns
8	$t_{d(CKO1H-BLZ)}$	Delay time, CLKOUT1 high to EMIF Bus low impedance¶	3	11	ns
9	$t_{d(HOLDH-BLZ)}$	Delay time, HOLD high to EMIF Bus low impedance	3P	6P	ns

‡ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
§ All pending EMIF transactions are allowed to complete before $\overline{\text{HOLDA}}$ is asserted. The worst cases 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 $\overline{\text{CE[3:0]}}$, $\overline{\text{BE[3:0]}}$, ED[31:0], EA[21:2], $\overline{\text{ARE}}$, $\overline{\text{AOE}}$, $\overline{\text{AWE}}$, $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, $\overline{\text{SSWE}}$, SDA10, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, and $\overline{\text{SDWE}}$.



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

Figure 25. $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ Timing

RESET TIMING

timing requirements for reset (see Figure 26)

NO.			-200 MIN	-200 MAX	UNIT
1	$t_{w(RST)}$	Width of the $\overline{RESET}$ pulse (PLL stable)†	10		CLKOUT1 cycles
		Width of the $\overline{RESET}$ pulse (PLL needs to sync up)‡	250		μs

† This parameter applies to CLKMODE x1 when CLKIN is stable and applies to CLKMODE x4 when CLKIN and PLL are stable.
‡ This parameter only applies to CLKMODE x4. The $\overline{RESET}$ 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, $\overline{RESET}$ must be asserted to ensure proper device operation. See the *Clock PLL* section for PLL lock times.

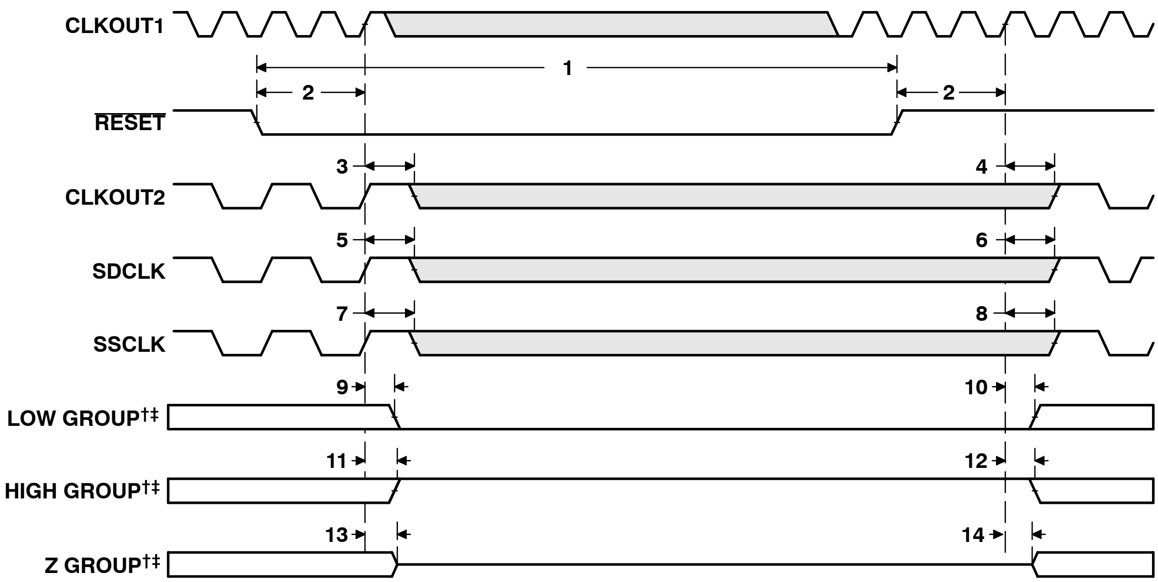
switching characteristics over recommended operating conditions during reset§¶ (see Figure 26)

NO.	PARAMETER		-200 MIN	-200 MAX	UNIT
2	$t_{R(RST)}$	Response time to change of value in $\overline{RESET}$ signal	2		CLKOUT1 cycles
3	$t_{d(CKO1H-CKO2IV)}$	Delay time, CLKOUT1 high to CLKOUT2 invalid	–1		ns
4	$t_{d(CKO1H-CKO2V)}$	Delay time, CLKOUT1 high to CLKOUT2 valid		10	ns
5	$t_{d(CKO1H-SDCLKIV)}$	Delay time, CLKOUT1 high to SDCLK invalid	–1		ns
6	$t_{d(CKO1H-SDCLKV)}$	Delay time, CLKOUT1 high to SDCLK valid		10	ns
7	$t_{d(CKO1H-SSCKIV)}$	Delay time, CLKOUT1 high to SSCLK invalid	–1		ns
8	$t_{d(CKO1H-SSCKV)}$	Delay time, CLKOUT1 high to SSCLK valid		10	ns
9	$t_{d(CKO1H-LOWIV)}$	Delay time, CLKOUT1 high to low group invalid	–1		ns
10	$t_{d(CKO1H-LOWV)}$	Delay time, CLKOUT1 high to low group valid		10	ns
11	$t_{d(CKO1H-HIGHIV)}$	Delay time, CLKOUT1 high to high group invalid	–1		ns
12	$t_{d(CKO1H-HIGHV)}$	Delay time, CLKOUT1 high to high group valid		10	ns
13	$t_{d(CKO1H-ZHZ)}$	Delay time, CLKOUT1 high to Z group high impedance	–1		ns
14	$t_{d(CKO1H-ZV)}$	Delay time, CLKOUT1 high to Z group valid		10	ns

§ Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
High group consists of: $\overline{HINT}$
Z group consists of: EA[21:2], ED[31:0], $\overline{CE[3:0]}$, $\overline{BE[3:0]}$, $\overline{ARE}$, $\overline{AWE}$, $\overline{AOE}$, $\overline{SSADS}$, $\overline{SSOE}$, $\overline{SSWE}$, SDA10, $\overline{SDRAS}$, $\overline{SDCAS}$, $\overline{SDWE}$, HD[15:0], CLKX0, CLKX1, FSX0, FSX1, DX0, DX1, CLKR0, CLKR1, FSR0, and FSR1.

¶ $\overline{HRDY}$ is gated by input $\overline{HCS}$.
If $\overline{HCS}$ = 0 at device reset, $\overline{HRDY}$ belongs to the high group.
If $\overline{HCS}$ = 1 at device reset, $\overline{HRDY}$ belongs to the low group.

RESET TIMING (CONTINUED)



† Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1
High group consists of: $\overline{\text{HINT}}$
Z group consists of: EA[21:2], ED[31:0], $\overline{\text{CE}}$[3:0], $\overline{\text{BE}}$[3:0], $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, $\overline{\text{AOE}}$, $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, $\overline{\text{SSWE}}$, SDA10, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, HD[15:0], CLKX0, CLKX1, FSX0, FSX1, DX0, DX1, CLKR0, CLKR1, FSR0, and FSR1.

‡ $\overline{\text{HRDY}}$ is gated by input $\overline{\text{HCS}}$.
If $\overline{\text{HCS}}$ = 0 at device reset, $\overline{\text{HRDY}}$ belongs to the high group.
If $\overline{\text{HCS}}$ = 1 at device reset, $\overline{\text{HRDY}}$ belongs to the low group.

Figure 26. Reset Timing

EXTERNAL INTERRUPT TIMING

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

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

† Interrupt signals are synchronized internally and are potentially recognized one cycle later if setup and hold times are violated. Thus, they can be connected to asynchronous inputs.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

switching characteristics over recommended operating conditions during interrupt response cycles§ (see Figure 27)

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

§ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
When the PLL is used (CLKMODE x4), 0.5P = 1/(2 × CPU clock frequency).
For CLKMODE x1: 0.5P = PH, where PH is the high period of CLKIN.

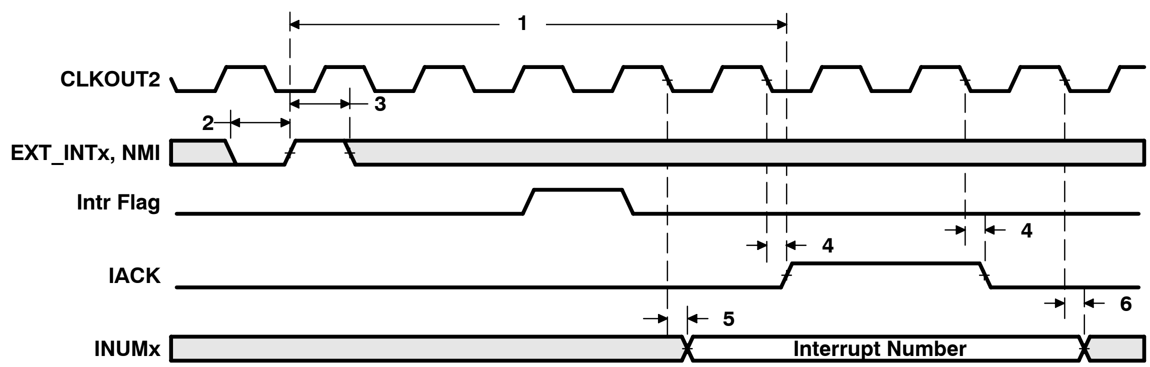


Figure 27. Interrupt Timing

HOST-PORT INTERFACE TIMING

timing requirements for host-port interface cycles†‡ (see Figure 28, Figure 29, Figure 30, and Figure 31)

NO.			-200		UNIT
			MIN	MAX	
1	$t_{su(SEL-HSTBL)}$	Setup time, select signals§ valid before $\overline{\text{HSTROBE}}$ low	4		ns
2	$t_{h(HSTBL-SEL)}$	Hold time, select signals§ valid after $\overline{\text{HSTROBE}}$ low	2		ns
3	$t_{w(HSTBL)}$	Pulse duration, $\overline{\text{HSTROBE}}$ low	2P		ns
4	$t_{w(HSTBH)}$	Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses	2P		ns
10	$t_{su(SEL-HASL)}$	Setup time, select signals§ valid before $\overline{\text{HAS}}$ low	4		ns
11	$t_{h(HASL-SEL)}$	Hold time, select signals§ valid after $\overline{\text{HAS}}$ low	2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before $\overline{\text{HSTROBE}}$ high	3		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after $\overline{\text{HSTROBE}}$ high	2		ns
14	$t_{h(HRDYL-HSTBL)}$	Hold time, $\overline{\text{HSTROBE}}$ low after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly.	1		ns
18	$t_{su(HASL-HSTBL)}$	Setup time, $\overline{\text{HAS}}$ low before $\overline{\text{HSTROBE}}$ low	2		ns
19	$t_{h(HSTBL-HASL)}$	Hold time, $\overline{\text{HAS}}$ low after $\overline{\text{HSTROBE}}$ low	2		ns

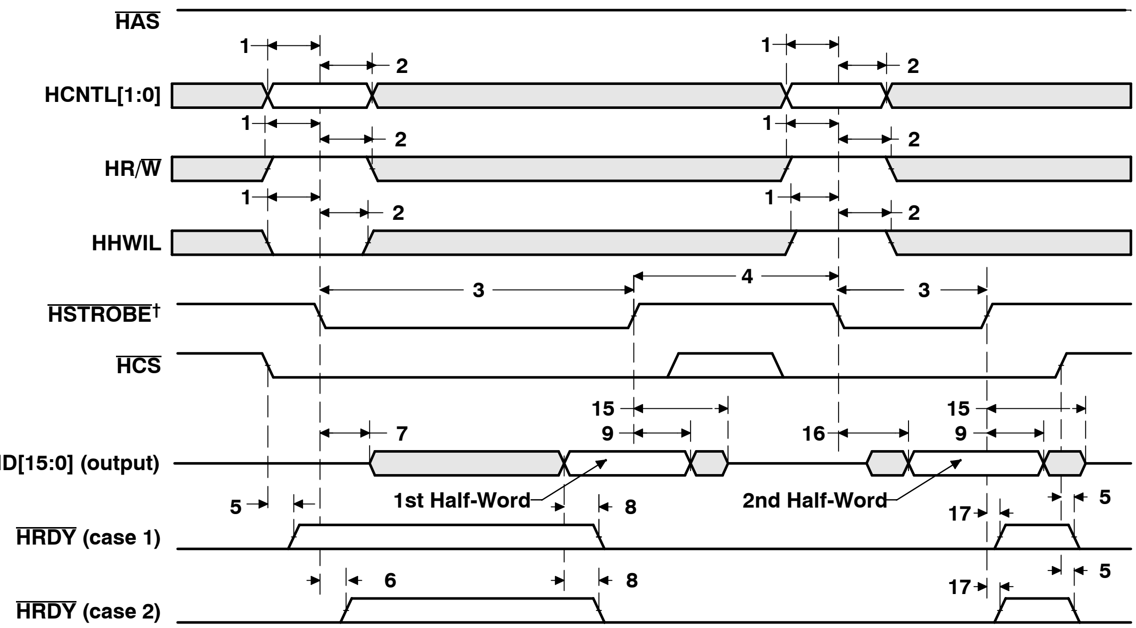
† $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
‡ The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
§ Select signals include: HCNTRL[1:0], HR/$\overline{\text{W}}$, and HHWIL.

switching characteristics over recommended operating conditions during host-port interface cycles†‡ (see Figure 28, Figure 29, Figure 30, and Figure 31)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
5	$t_{d(HCS-HRDY)}$	Delay time, $\overline{\text{HCS}}$ to $\overline{\text{HRDY}}$¶	1	9	ns
6	$t_{d(HSTBL-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high#	3	12	ns
7	$t_{oh(HSTBL-HDLZ)}$	Output hold time, HD low impedance after $\overline{\text{HSTROBE}}$ low for an HPI read	4		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to $\overline{\text{HRDY}}$ low	P – 3	P + 3	ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high	2	12	ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance	3	12	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD valid	2	12	ns
17	$t_{d(HSTBH-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ high to $\overline{\text{HRDY}}$ high‖	3	12	ns
20	$t_{d(HASL-HRDYH)}$	Delay time, $\overline{\text{HAS}}$ low to $\overline{\text{HRDY}}$ high	3	12	ns

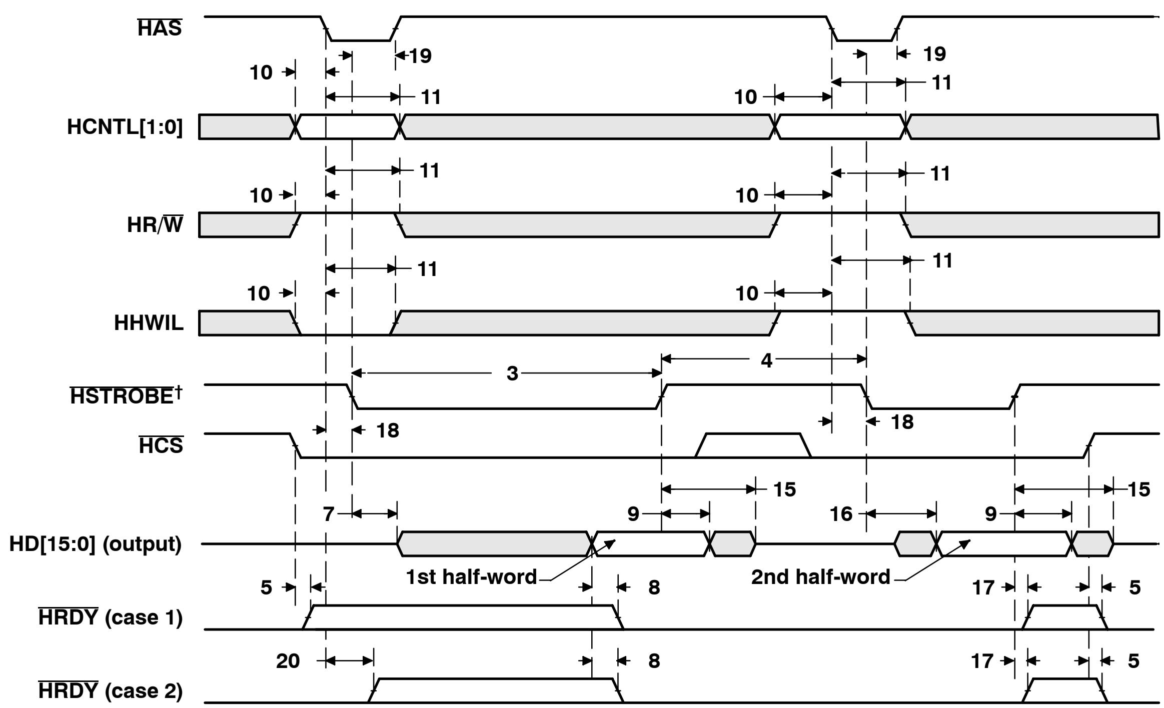
† $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
‡ The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
¶ $\overline{\text{HCS}}$ enables $\overline{\text{HRDY}}$, and $\overline{\text{HRDY}}$ is always low when $\overline{\text{HCS}}$ is high. The case where $\overline{\text{HRDY}}$ goes high when $\overline{\text{HCS}}$ falls indicates that HPI is busy completing a previous HPID write or READ with autoincrement.
# This parameter is used during an HPID read. At the beginning of the first half-word transfer on the falling edge of $\overline{\text{HSTROBE}}$, the HPI sends the request to the DMA auxiliary channel, and $\overline{\text{HRDY}}$ remains high until the DMA auxiliary channel loads the requested data into HPID.
‖ This parameter is used after the second half-word of an HPID write or autoincrement read. $\overline{\text{HRDY}}$ remains low if the access is not an HPID write or autoincrement read. Reading or writing to HPIC or HPIA does not affect the $\overline{\text{HRDY}}$ signal.

HOST-PORT INTERFACE TIMING (CONTINUED)



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

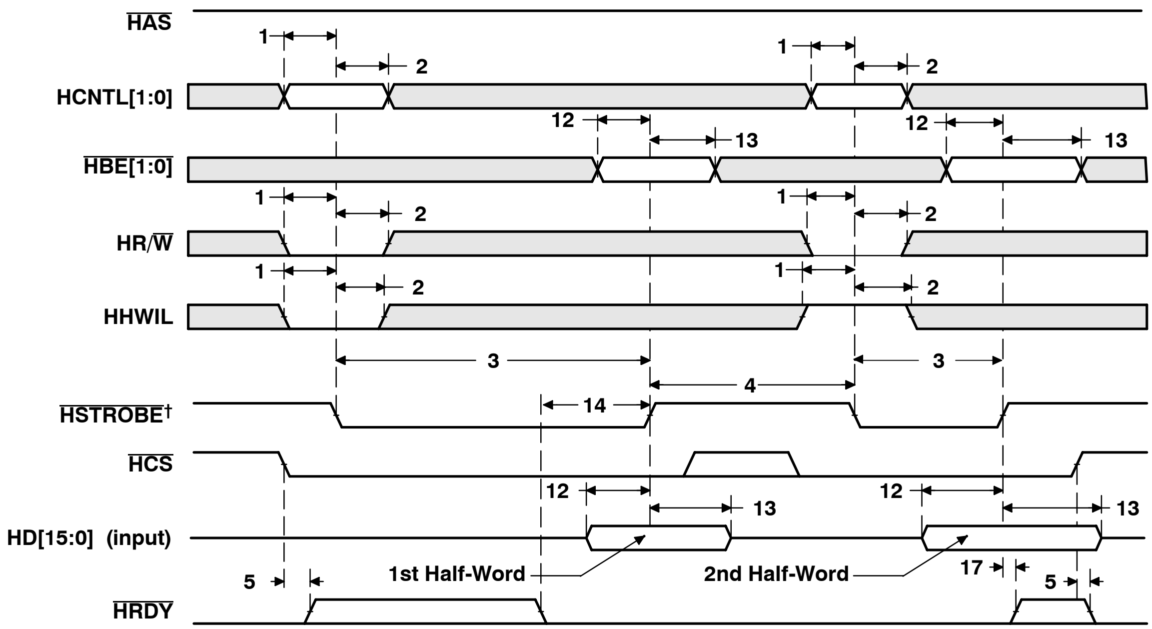
Figure 28. HPI Read Timing (HAS Not Used, Tied High)



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

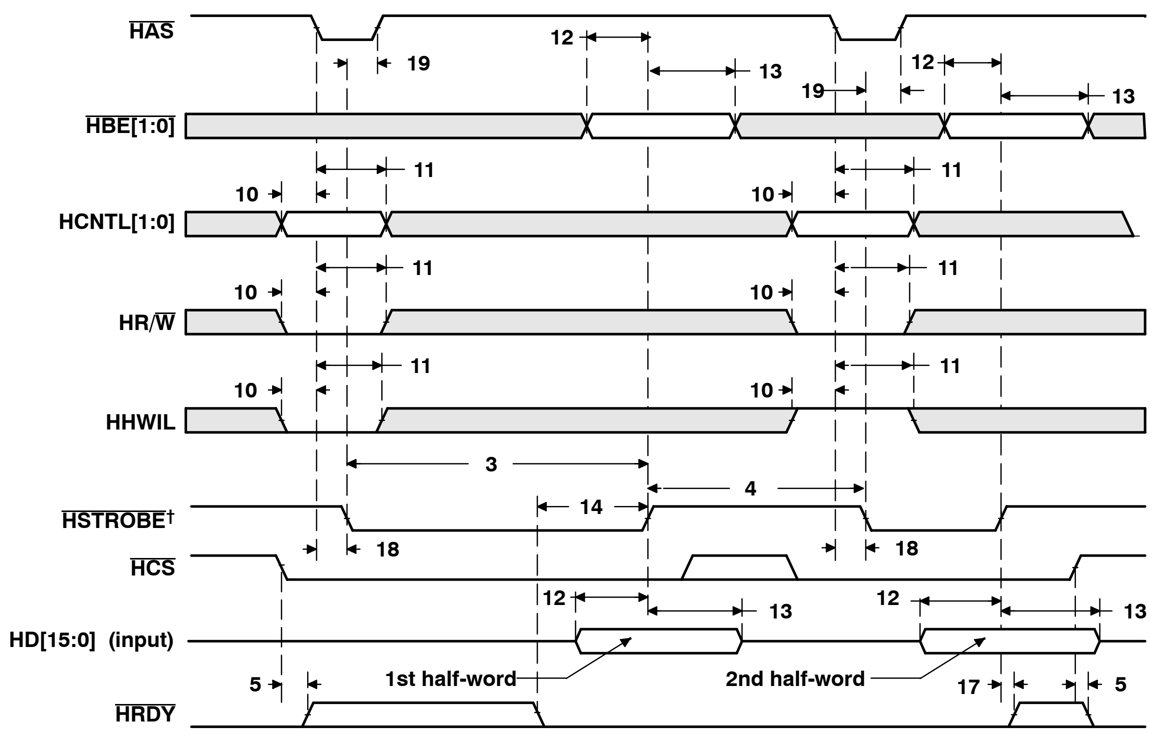
Figure 29. HPI Read Timing (HAS Used)

HOST-PORT INTERFACE TIMING (CONTINUED)



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

Figure 30. HPI Write Timing (HAS Not Used, Tied High)



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

Figure 31. HPI Write Timing (HAS Used)

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP†‡ (see Figure 32)

NO.				-200 MIN	-200 MAX	UNIT
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P§		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1¶		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9		ns
			CLKR ext	2		
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	6		ns
			CLKR ext	3		
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	8		ns
			CLKR ext	0		
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	3		ns
			CLKR ext	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. For example, when running parts at 200 MHz, use P = 5 ns.
§ The maximum bit rate for the C6202/02B/03 device is 100 Mbps or CPU/2 (the slower of the two). Care must be taken to ensure that the AC timings specified in this data sheet are met. The maximum bit rate for McBSP-to-McBSP communications is 100 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 200 MHz (P = 5 ns), use 10 ns as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 100 MHz (P = 10 ns), use 2P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies when the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
¶ The minimum CLKR/X pulse duration is either (P–1) or 4 ns, whichever is larger. For example, when running parts at 200 MHz (P = 5 ns), use 4 ns as the minimum CLKR/X pulse duration. When running parts at 100 MHz (P = 10 ns), use (P–1) = 9 ns as the minimum CLKR/X pulse duration.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics over recommended operating conditions for McBSP†‡§ (see Figure 32)

NO.	PARAMETER			–200 MIN	–200 MAX	UNIT
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		3	10	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 – 1.3#	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	4	ns
			CLKX ext	3	9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	–1	4	ns
			CLKX ext	3	9	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode	FSX int	–1	3	ns
			FSX ext	3	9	

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum delay times also represent minimum output hold times.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
¶ The maximum bit rate for the C6202/02B/03 device is 100 Mbps or CPU/2 (the slower of the two). Care must be taken to ensure that the AC timings specified in this data sheet are met. The maximum bit rate for McBSP-to-McBSP communications is 100 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 10 ns (100 MHz), whichever value is larger. For example, when running parts at 200 MHz (P = 5 ns), use 10 ns as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 100 MHz (P = 10 ns), use 2P = 20 ns (50 MHz) as the minimum CLKR/X clock cycle. The maximum bit rate for McBSP-to-McBSP communications applies when the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
\# C = H or L
S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

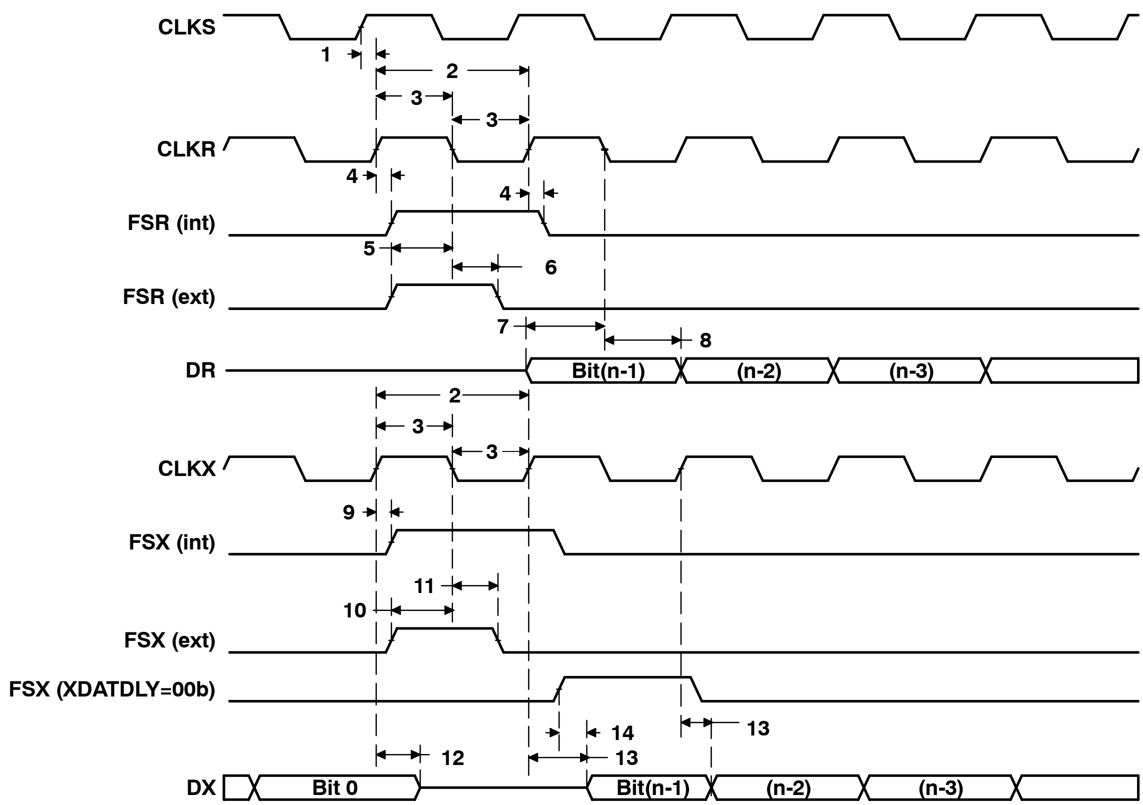


Figure 32. McBSP Timing Diagram

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 33)

NO.			-200		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

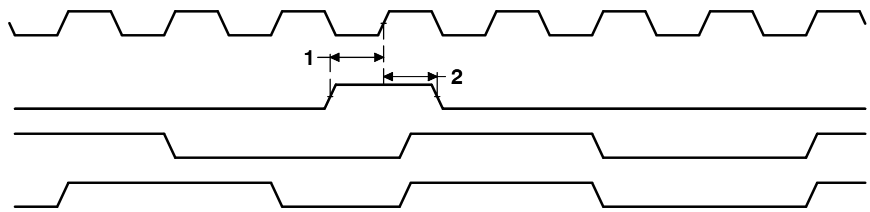


Figure 33. FSR Timing When GSYNC = 1

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 34)

NO.			-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 34)

NO.	PARAMETER		-200				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L – 2	L + 3			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ 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)

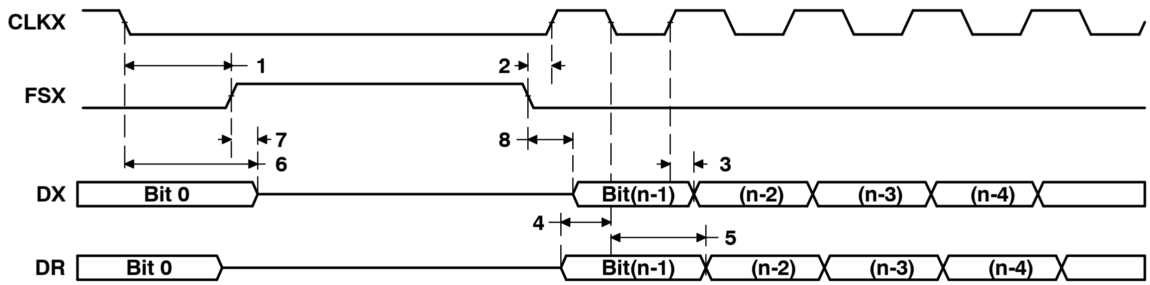


Figure 34. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 35)

NO.			-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 35)

NO.	PARAMETER		-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ 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)

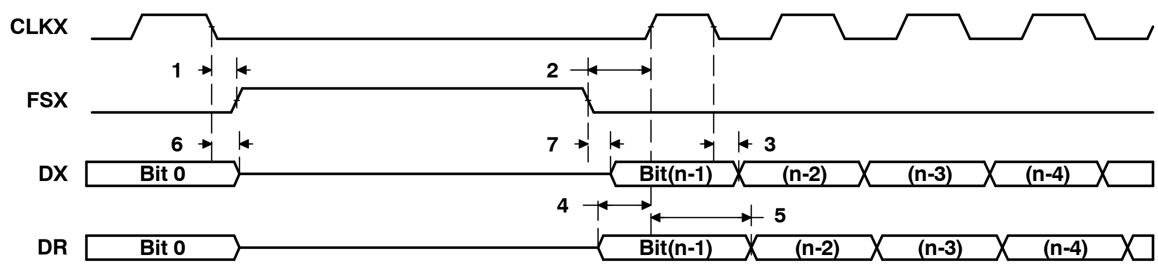


Figure 35. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 36)

NO.			-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 36)

NO.	PARAMETER		-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ 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)

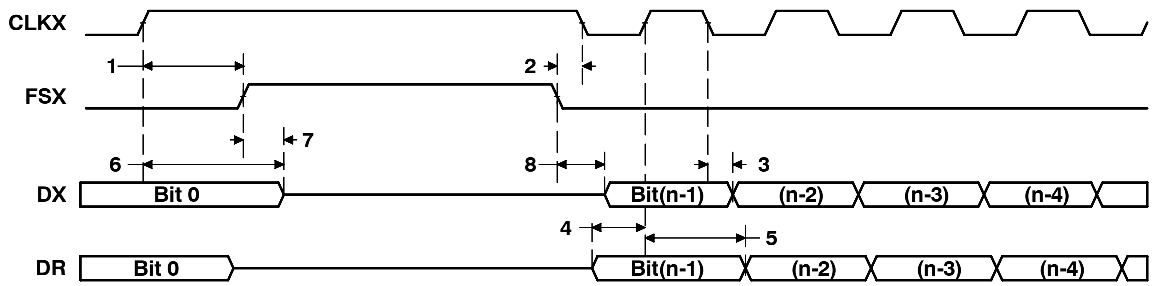


Figure 36. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 37)

NO.			-200				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 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

switching characteristics over recommended operating conditions for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 37)

NO.	PARAMETER		-200				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	H – 2	H + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	T – 2	T + 1			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	−2	4	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 + 4	2P + 2	4P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ 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)

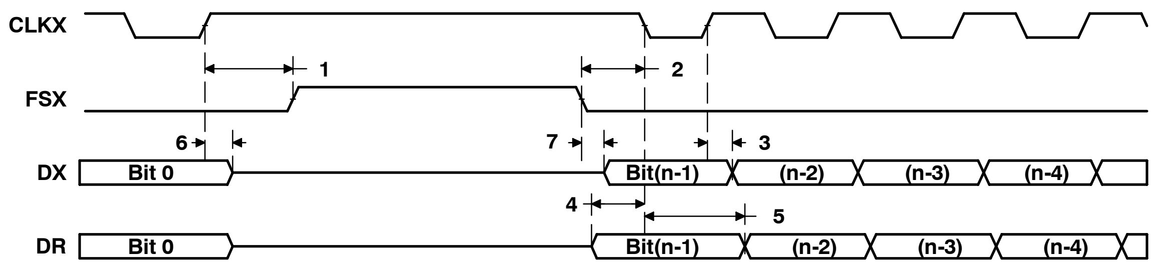


Figure 37. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

DMAC, TIMER, POWER-DOWN TIMING

switching characteristics over recommended operating conditions for DMAC outputs (see Figure 38)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
1	$t_{d(CKO1H-DMACV)}$	Delay time, CLKOUT1 high to DMAC valid	2	10	ns

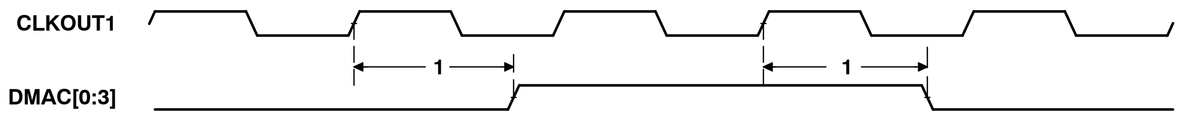


Figure 38. DMAC Timing Diagram

timing requirements for timer inputs† (see Figure 39)

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

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

switching characteristics over recommended operating conditions for timer outputs (see Figure 39)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
2	$t_{d(CKO1H-TOUTV)}$	Delay time, CLKOUT1 high to TOUT valid	2	9	ns

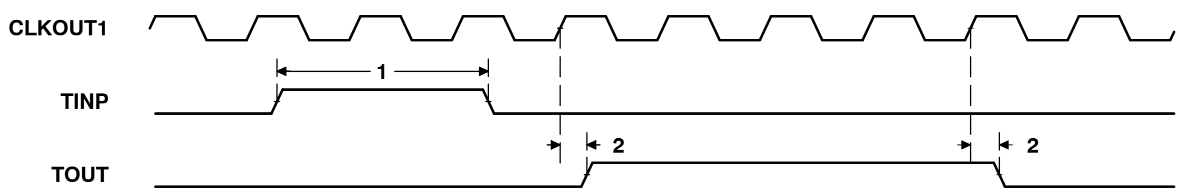


Figure 39. Timer Timing Diagram

switching characteristics over recommended operating conditions for power-down outputs (see Figure 40)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
1	$t_{d(CKO1H-PDV)}$	Delay time, CLKOUT1 high to PD valid	2	9	ns

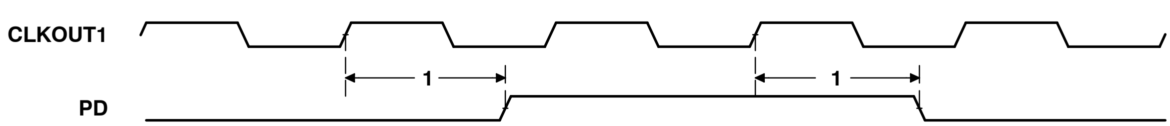


Figure 40. Power-Down Timing

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 41)

NO.			-200		UNIT
			MIN	MAX	
1	$t_{c(TCK)}$	Cycle time, TCK	35		ns
3	$t_{su(TDIV\text{-}TCKH)}$	Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high	10		ns
4	$t_{h(TCKH\text{-}TDIV)}$	Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high	9		ns

switching characteristics over recommended operating conditions for JTAG test port (see Figure 41)

NO.	PARAMETER		-200		UNIT
			MIN	MAX	
2	$t_{d(TCKL\text{-}TDOV)}$	Delay time, TCK low to TDO valid	−3	12	ns

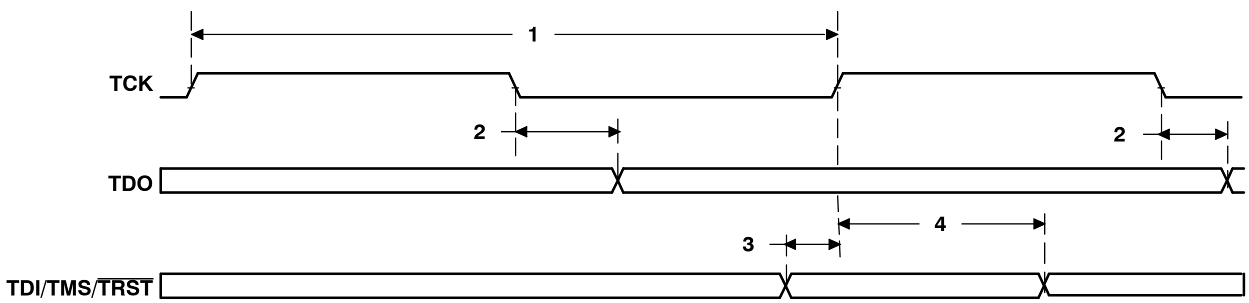


Figure 41. JTAG Test-Port Timing Diagram

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