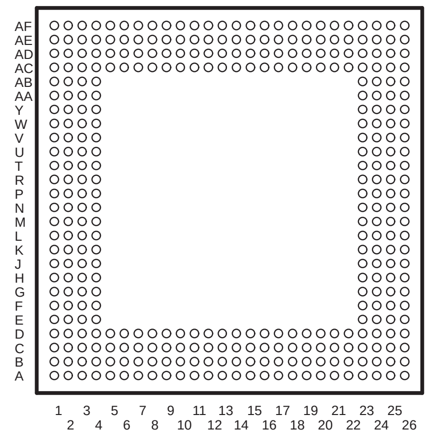
- **Highest Performance Fixed-Point Digital Signal Processor (DSP) TMS320C6201**
  - **5-ns Instruction Cycle Time**
  - **200-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **1600 MIPS**
- **Highest Performance Fixed-Point Digital Signal Processor (DSP) TMS320C6201B**
  - **4.3-ns Instruction Cycle Time**
  - **167-, 200-, and 233-MHz Clock Rates**
  - **Eight 32-Bit Instructions/Cycle**
  - **Over 1860 MIPS**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) ’C6200 CPU Core**
  - **Eight Independent Functional Units:**
    - **Six 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 a Single Block (’6201)**
  - **512K-Bit Dual-Access Internal Data (64K Bytes) Organized as Two Blocks for Improved Concurrency (’6201B)**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Synchronous Memories: SDRAM and SBSRAM**
  - **Glueless Interface to Asynchronous Memories: SRAM and EPROM**
- **Four-Channel Bootloading Direct-Memory-Access (DMA) Controller with an Auxiliary Channel**

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



- **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 (GGP Suffix) (’6201)**
- **352-Pin BGA Package (GJC Suffix) (’6201B)**
- **352-Pin BGA Package (GJL Suffix) (’6201B)**
- **CMOS Technology**
  - **0.25-µm/5-Level Metal Process (’6201)**
  - **0.18-µm/5-Level Metal Process (’6201B)**
- **3.3-V I/Os, 2.5-V Internal (’6201)**
- **3.3-V I/Os, 1.8-V Internal (’6201B)**

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.

Signal Descriptions

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
CLOCK/PLL				
CLKIN	C10	B9	I	Clock Input
CLKOUT1	AF22	AC18	O	Clock output at full device speed
CLKOUT2	AF20	AC16	O	Clock output at half of device speed
CLKMODE1	C6	D8	I	Clock-mode select • Selects whether the CPU clock frequency = input clock frequency x4 or x1
CLKMODE0	C5	C7		
PLLFREQ3	A9	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	D11		
PLLFREQ1	B10	B10		
PLLV‡	D12	B11	A§	PLL analog V_{CC} connection for the low-pass filter
PLLG‡	C12	C12	A§	PLL analog GND connection for the low-pass filter
PLLF	A11	D12	A§	PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION				
TMS	L3	L3	I	JTAG test port mode select (features an internal pullup)
TDO	W2	U4	O/Z	JTAG test port data out
TDI	R4	T2	I	JTAG test port data in (features an internal pullup)
TCK	R3	R3	I	JTAG test port clock
$\overline{TRST}$	T1	R4	I	JTAG test port reset (features an internal pulldown)
EMU1	Y1	V3	I/O/Z	Emulation pin 1, pullup with a dedicated 20-kΩ resistor
EMU0	W3	W2	I/O/Z	Emulation pin 0, pullup with a dedicated 20-kΩ resistor
RESET AND INTERRUPTS				
$\overline{RESET}$	K2	K2	I	Device reset
NMI	L2	L2	I	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	U3	U2	I	External interrupts • Edge-driven (rising edge)
EXT_INT6	V2	T4		
EXT_INT5	W1	V1		
EXT_INT4	U4	V2		
IACK	Y2	Y1	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	AA1	V4	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	Y2		
INUM1	AA2	AA1		
INUM0	AB1	W4		
LITTLE ENDIAN/BIG ENDIAN				
LENDIAN	H3	G2	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	E2	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)

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
HOST-PORT INTERFACE (HPI)				
$\overline{\text{HINT}}$	H26	J26	O	Host interrupt (from DSP to host)
HCNTL1	F23	G24	I	Host control – selects between control, address, or data registers
HCNTL0	D25	F25	I	Host control – selects between control, address, or data registers
HHWIL	C26	E26	I	Host half-word select – first or second half-word (not necessarily high or low order)
$\overline{\text{HBE1}}$	E23	F24	I	Host byte select within word or half-word
$\overline{\text{HBE0}}$	D24	E25	I	Host byte select within word or half-word
HR/$\overline{\text{W}}$	C23	B22	I	Host read or write select
HD15	B13	A12		
HD14	B14	D13		
HD13	C14	C13		
HD12	B15	D14		
HD11	D15	B15		
HD10	B16	C15		
HD9	A17	D15		
HD8	B17	B16	I/O/Z	Host-port data (used for transfer of data, address, and control)
HD7	D16	C16		
HD6	B18	B17		
HD5	A19	D16		
HD4	C18	A18		
HD3	B19	B18		
HD2	C19	D17		
HD1	B20	C18		
HD0	B21	A20		
$\overline{\text{HAS}}$	C22	C20	I	Host address strobe
$\overline{\text{HCS}}$	B23	B21	I	Host chip select
$\overline{\text{HDS1}}$	D22	C21	I	Host data strobe 1
$\overline{\text{HDS2}}$	A24	D20	I	Host data strobe 2
$\overline{\text{HRDY}}$	J24	J25	O	Host ready (from DSP to host)
BOOT MODE				
BOOTMODE4	D8	C8		
BOOTMODE3	B4	B6		
BOOTMODE2	A3	D7	I	Boot mode
BOOTMODE1	D5	C6		
BOOTMODE0	C4	B5		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY				
$\overline{CE3}$	AE22	AD20	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	AA24		
$\overline{CE1}$	AB24	AB26		
$\overline{CE0}$	AC26	AA25		
$\overline{BE3}$	AB25	Y24	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	W23		
$\overline{BE1}$	Y23	AA26		
$\overline{BE0}$	AA26	W25		
EMIF – ADDRESS				
EA21	J26	K25	O/Z	External address (word address)
EA20	K25	L24		
EA19	L24	L25		
EA18	K26	M23		
EA17	M26	M25		
EA16	M25	M24		
EA15	P25	N23		
EA14	P24	P24		
EA13	R25	P23		
EA12	T26	R25		
EA11	R23	R24		
EA10	U26	R23		
EA9	U25	T25		
EA8	T23	T24		
EA7	V26	U25		
EA6	V25	T23		
EA5	W26	V26		
EA4	V24	V25		
EA3	W25	U23		
EA2	Y26	V24		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
EMIF – DATA				
ED31	AB2	Y3	I/O/Z	External data
ED30	AC1	AA2		
ED29	AA4	AB1		
ED28	AD1	AA3		
ED27	AC3	AB2		
ED26	AD4	AE5		
ED25	AF3	AD6		
ED24	AE4	AC7		
ED23	AD5	AE6		
ED22	AF4	AD7		
ED21	AE5	AC8		
ED20	AD6	AD8		
ED19	AE6	AC9		
ED18	AD7	AF7		
ED17	AC8	AD9		
ED16	AF7	AC10		
ED15	AD9	AE9		
ED14	AD10	AF9		
ED13	AF9	AC11		
ED12	AC11	AE10		
ED11	AE10	AD11		
ED10	AE11	AE11		
ED9	AF11	AC12		
ED8	AE14	AD12		
ED7	AF15	AE12		
ED6	AE15	AC13		
ED5	AF16	AD14		
ED4	AC15	AC14		
ED3	AE17	AE15		
ED2	AF18	AD15		
ED1	AF19	AE16		
ED0	AC17	AD16		
EMIF – ASYNCHRONOUS MEMORY CONTROL				
$\overline{ARE}$	Y24	V23	O/Z	Asynchronous memory read enable
$\overline{AOE}$	AC24	AB25	O/Z	Asynchronous memory output enable
$\overline{AWE}$	AD23	AE22	O/Z	Asynchronous memory write enable
ARDY	W23	Y26	I	Asynchronous memory ready input

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
EMIF – SYNCHRONOUS BURST SRAM (SBSRAM) CONTROL				
$\overline{\text{SSADS}}$	AC20	AD19	O/Z	SBSRAM address strobe
$\overline{\text{SSOE}}$	AF21	AD18	O/Z	SBSRAM output enable
$\overline{\text{SSWE}}$	AD19	AF18	O/Z	SBSRAM write enable
SSCLK	AD17	AC15	O	SBSRAM clock
EMIF – SYNCHRONOUS DRAM (SDRAM) CONTROL				
SDA10	AD21	AC19	O/Z	SDRAM address 10 (separate for deactivate command)
$\overline{\text{SDRAS}}$	AF24	AD21	O/Z	SDRAM row-address strobe
$\overline{\text{SDCAS}}$	AD22	AC20	O/Z	SDRAM column-address strobe
$\overline{\text{SDWE}}$	AF23	AE21	O/Z	SDRAM write enable
SDCLK	AE20	AC17	O	SDRAM clock
EMIF – BUS ARBITRATION				
$\overline{\text{HOLD}}$	AA25	Y25	I	Hold request from the host
$\overline{\text{HOLDA}}$	A7	C9	O	Hold-request acknowledge to the host
TIMERS				
TOUT1	H24	K23	O	Timer 1 or general-purpose output
TINP1	K24	L23	I	Timer 1 or general-purpose input
TOUT0	M4	M4	O	Timer 0 or general-purpose output
TINP0	K4	H2	I	Timer 0 or general-purpose input
DMA ACTION COMPLETE STATUS				
DMAC3	D2	E1	O	DMA action complete
DMAC2	F4	F2		
DMAC1	D1	G3		
DMAC0	E2	H4		
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)				
CLKS1	E25	F26	I	External clock source (as opposed to internal)
CLKR1	H23	H25	I/O/Z	Receive clock
CLKX1	F26	J24	I/O/Z	Transmit clock
DR1	D26	H23	I	Receive data
DX1	G23	G25	O/Z	Transmit data
FSR1	E26	J23	I/O/Z	Receive frame sync
FSX1	F25	G26	I/O/Z	Transmit frame sync

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)				
CLKS0	L4	L4	I	External clock source (as opposed to internal)
CLKR0	M2	M2	I/O/Z	Receive clock
CLKX0	L1	M3	I/O/Z	Transmit clock
DR0	J1	J1	I	Receive data
DX0	R1	P4	O/Z	Transmit data
FSR0	P4	N3	I/O/Z	Receive frame sync
FSX0	P3	N4	I/O/Z	Transmit frame sync
RESERVED FOR TEST				
RSV0	T2	T3	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV1	G2	F1	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV2	C11	C11	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV3	B9	D10	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV4	A6	D9	I	Reserved for testing, ***pulldown*** with a dedicated 20-kΩ resistor
RSV5	C8	A7	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV6	C21	D18	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV7	B22	C19	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV8	A23	D19	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV9	E4	F3	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
UNCONNECTED PINS				
NC	A8	AF20		Unconnected pins
	B8	AE18		
	C9	AE17		
	D10	–		
	D21	–		
	G1	J4		
	H1	J3		
	H2	G1		
	J2	K4		
	K3	J2		
	R2	R2		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
3.3-V SUPPLY VOLTAGE PINS				
DV_{DD}	A10	A5	S	3.3-V supply voltage
	A15	A11		
	A18	A16		
	A21	A22		
	A22	B7		
	B7	B8		
	C1	B19		
	D17	B20		
	F3	C10		
	G24	C14		
	G25	C17		
	H25	G4		
	J25	G23		
	L25	H3		
	M3	H24		
	N3	K3		
	N23	K24		
	R26	L1		
	T24	L26		
	U24	N24		
	W24	P3		
	Y4	T1		
	AB3	T26		
	AB4	U3		
	AB26	U24		
	AC6	W3		
	AC10	W24		
	AC19	Y4		
	AC21	Y23		
	AC22	AD10		
	AC25	AD13		
	AD11	AD17		
	AD13	AE7		
	AD15	AE8		
	AD18	AE19		
	AE18	AE20		
	AE21	AF5		
	AF5	AF11		
	AF6	AF16		
	AF17	AF22		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
2.5-V SUPPLY VOLTAGE PINS FOR ’C6201 1.8-V SUPPLY VOLTAGE PINS FOR ’C6201B				
CV_{DD}	A5	A1	S	2.5-V supply voltage for ’C6201 1.8-V supply voltage for ’C6201B
	A12	A2		
	A16	A3		
	A20	A24		
	B2	A25		
	B6	A26		
	B11	B1		
	B12	B2		
	B25	B3		
	C3	B24		
	C15	B25		
	C20	B26		
	C24	C1		
	D4	C2		
	D6	C3		
	D7	C4		
	D9	C23		
	D14	C24		
	D18	C25		
	D20	C26		
	D23	D3		
	E1	D4		
	F1	D5		
	H4	D22		
	J4	D23		
	J23	D24		
	K1	E4		
	K23	E23		
	M1	AB4		
	M24	AB23		
	N4	AC3		
	N25	AC4		
	P2	AC5		
	P23	AC22		
	T3	AC23		
	T4	AC24		
	U1	AD1		
	V4	AD2		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
2.5-V SUPPLY VOLTAGE PINS FOR ’C6201 1.8-V SUPPLY VOLTAGE PINS FOR ’C6201B (CONTINUED)				
CV_{DD}	V23	AD3	S	2.5-V supply voltage for ’C6201 1.8-V supply voltage for ’C6201B
	AC4	AD4		
	AC9	AD23		
	AC12	AD24		
	AC13	AD25		
	AC18	AD26		
	AC23	AE1		
	AD3	AE2		
	AD8	AE3		
	AD14	AE24		
	AD24	AE25		
	AE2	AE26		
	AE8	AF1		
	AE12	AF2		
	AE25	AF3		
	AF12	AF24		
	–	AF25		
	–	AF26		
GROUND PINS				
V_{SS}	A1	A4	GND	Ground pins
	A2	A6		
	A4	A8		
	A13	A10		
	A14	A13		
	A25	A14		
	A26	A15		
	B1	A17		
	B3	A19		
	B5	A21		
	B24	A23		
	B26	B4		
	C2	B12		
	C7	B13		
	C13	B14		
	C16	B23		
	C17	C5		
	C25	C22		
	D13	D1		

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

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)				
V_{SS}	D19	D2	GND	Ground pins
	E3	D6		
	E24	D21		
	F2	D25		
	F24	D26		
	G3	E3		
	G4	E24		
	G26	F4		
	J3	F23		
	L23	H1		
	L26	H26		
	M23	K1		
	N1	K26		
	N2	M1		
	N24	M26		
	N26	N1		
	P1	N2		
	P26	N25		
	R24	N26		
	T25	P1		
	U2	P2		
	U23	P25		
	V1	P26		
	V3	R1		
	Y3	R26		
	Y25	U1		
	AA3	U26		
	AA23	W1		
	AB23	W26		
	AC2	AA4		
	AC5	AA23		
	AC7	AB3		
	AC14	AB24		
	AC16	AC1		
	AD2	AC2		
	AD12	AC6		
	AD16	AC21		
	AD20	AC25		

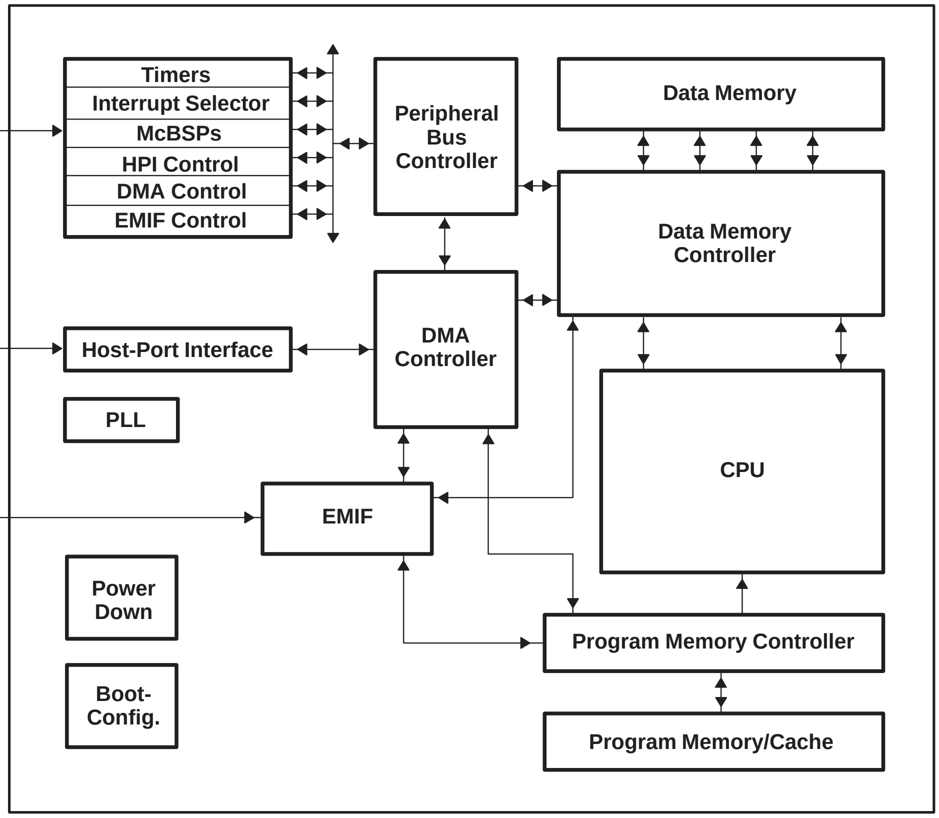
† I = Input, O = Output, Z = High Impedance, S = Supply Voltage, GND = Ground

Signal Descriptions (Continued)

SIGNAL NAME	GGP, GJC PIN NO.	GJL PIN NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)				
V_{SS}	AD25	AC26	GND	Ground pins
	AE1	AD5		
	AE3	AD22		
	AE7	AE4		
	AE9	AE13		
	AE13	AE14		
	AE16	AE23		
	AE19	AF4		
	AE23	AF6		
	AE24	AF8		
	AE26	AF10		
	AF1	AF12		
	AF2	AF13		
	AF8	AF14		
	AF10	AF15		
	AF13	AF17		
	AF14	AF19		
	AF25	AF21		
	AF26	AF23		

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

functional block diagram



signal groups

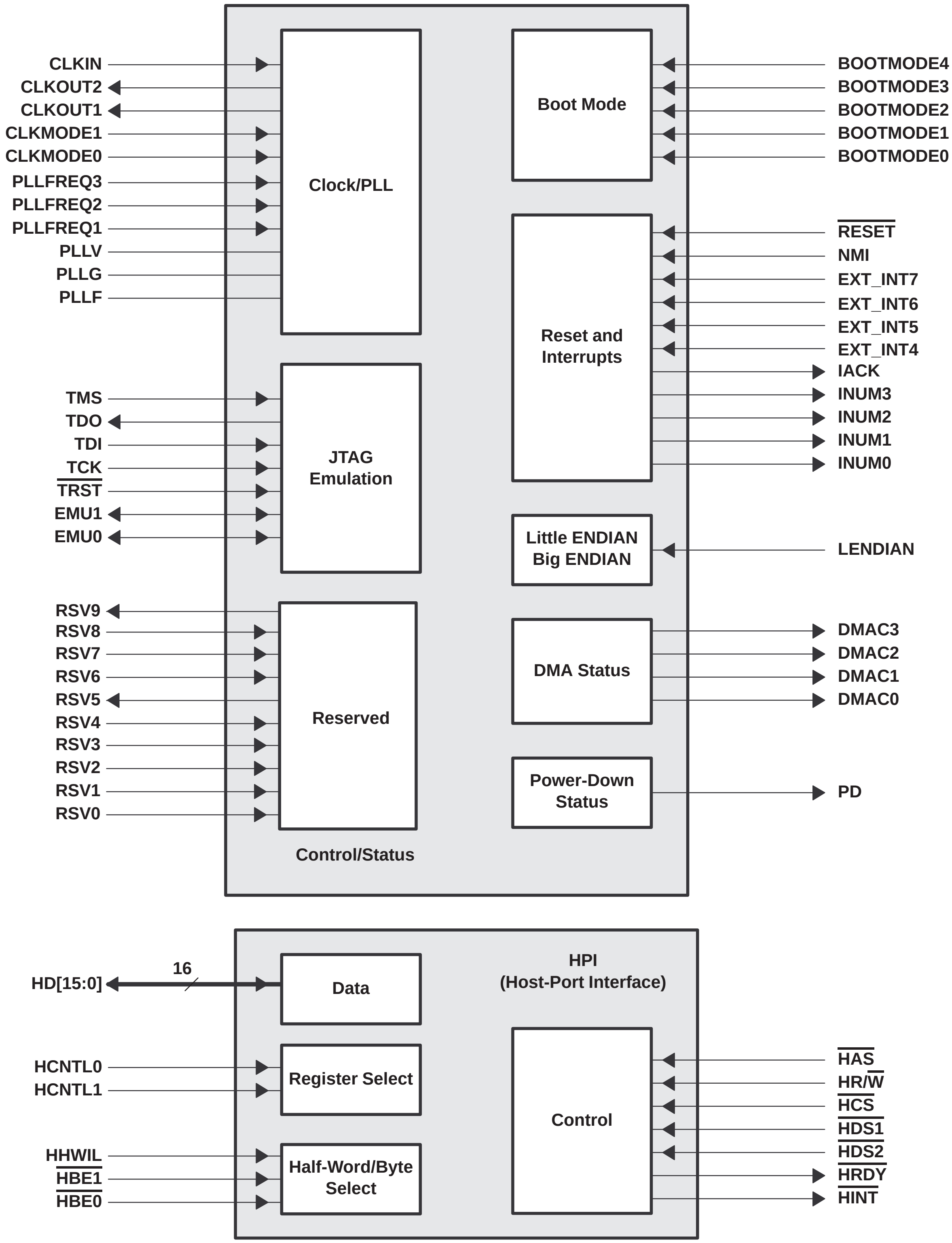


Figure 1. CPU and Peripheral Signals

signal groups (continued)

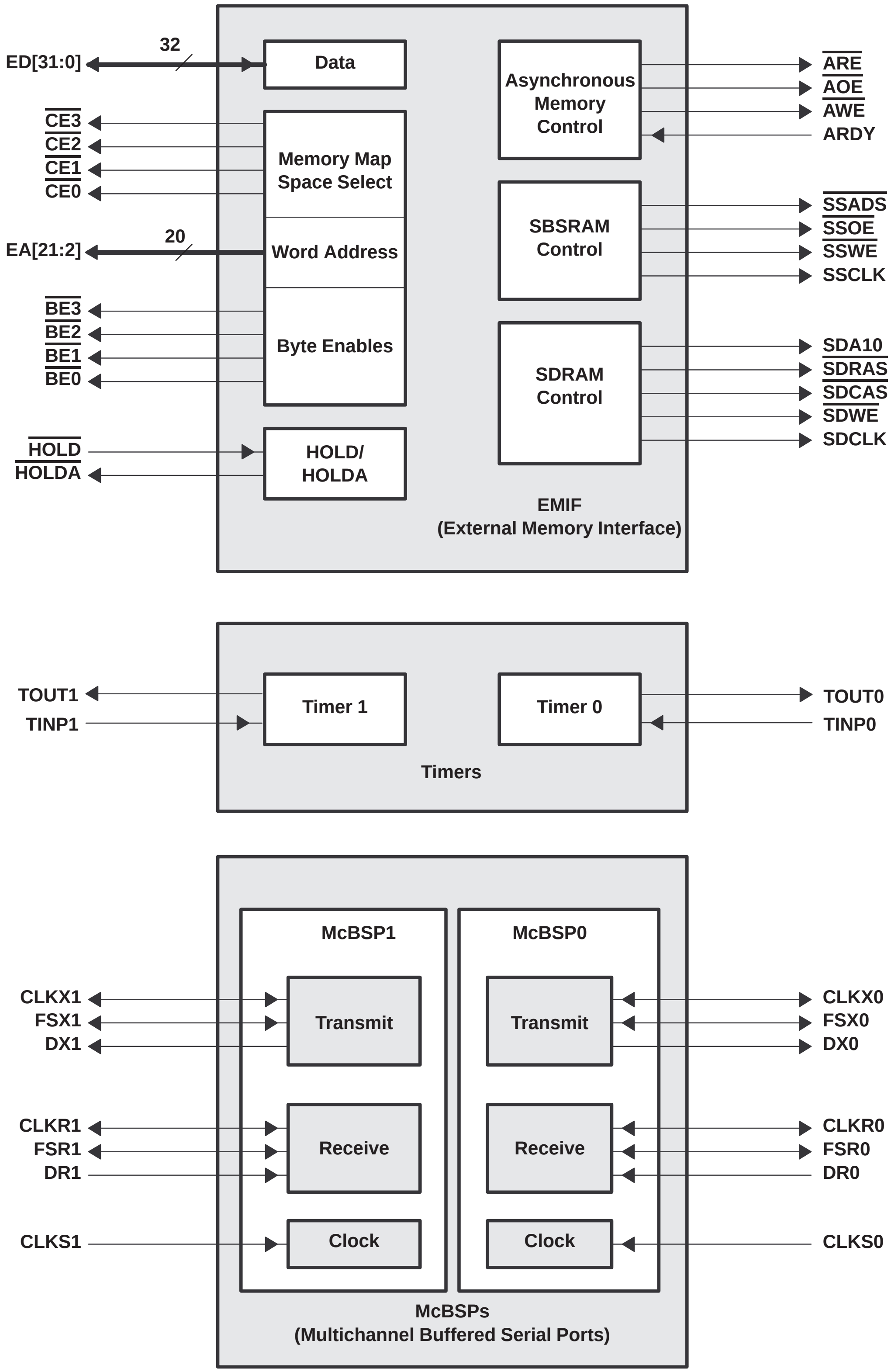


Figure 2. Peripheral Signals

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 ’C6200† 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 Figure 3 and Figure 4). 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 ’C6200 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 ’C6200 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.

† Where unique device characteristics are specified, TMS320C6201 and TMS320C6201B identifiers are used. For generic characteristics, no identifiers are needed, ’C62xx is used, or ’C6200 is used.

CPU description (continued)

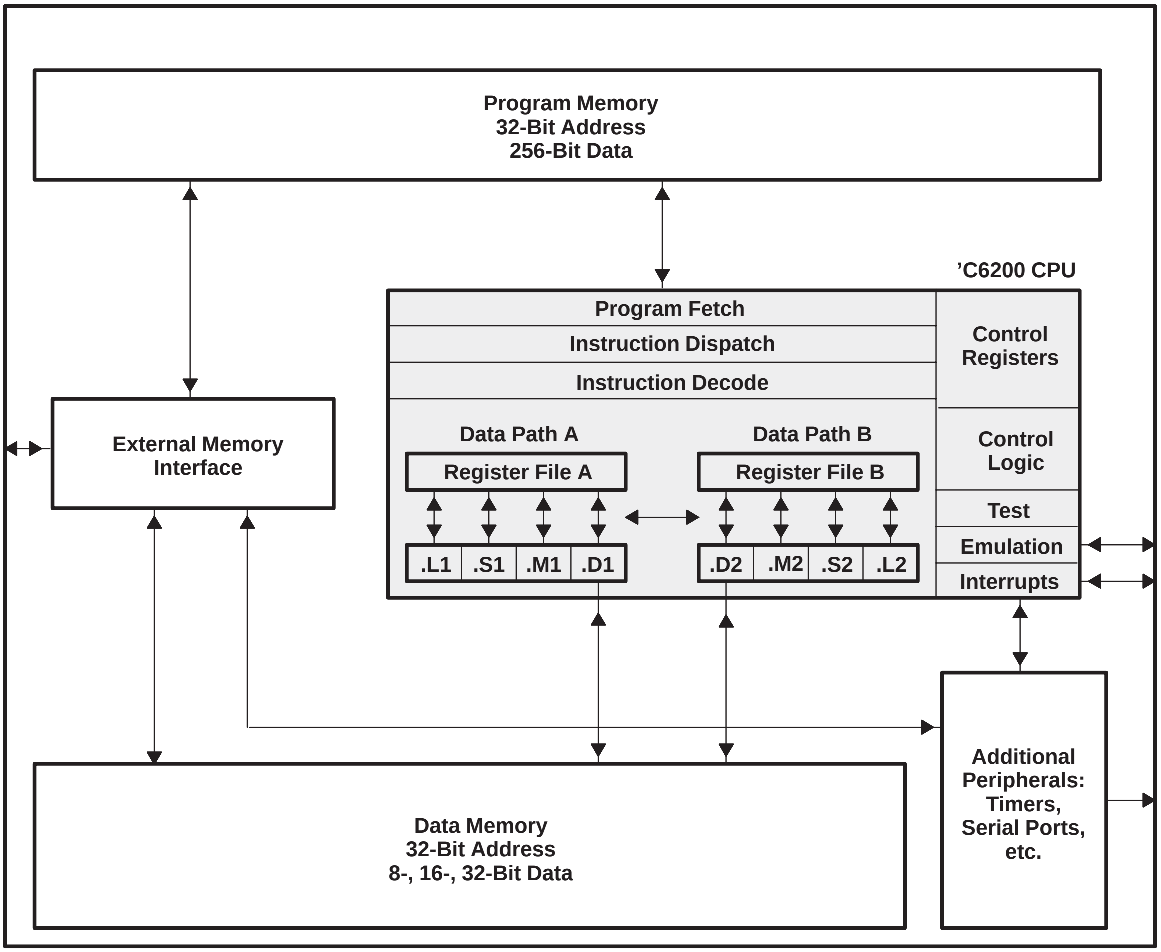


Figure 3. TMS320C6200 CPU Block Diagram

CPU description (continued)

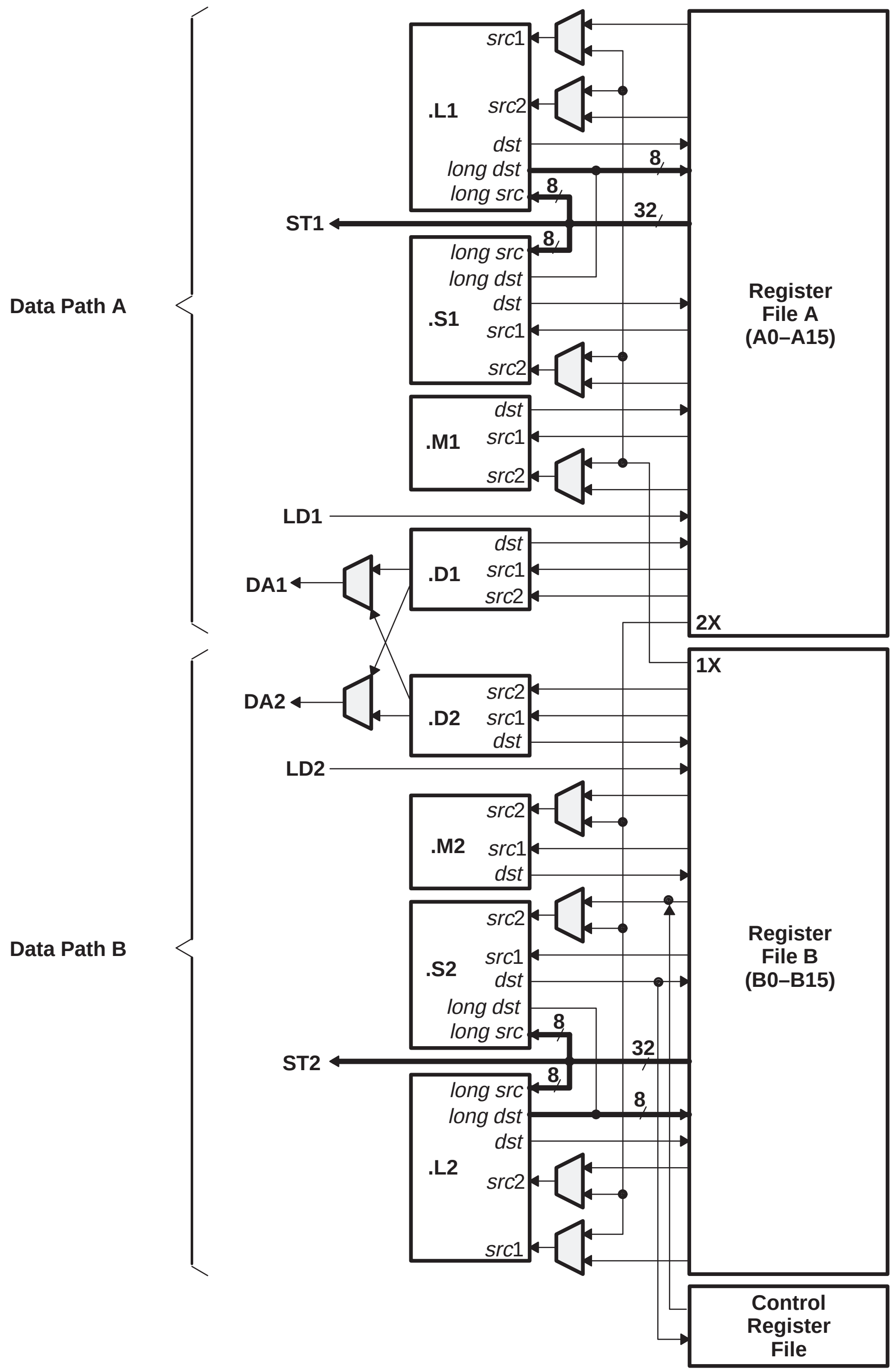


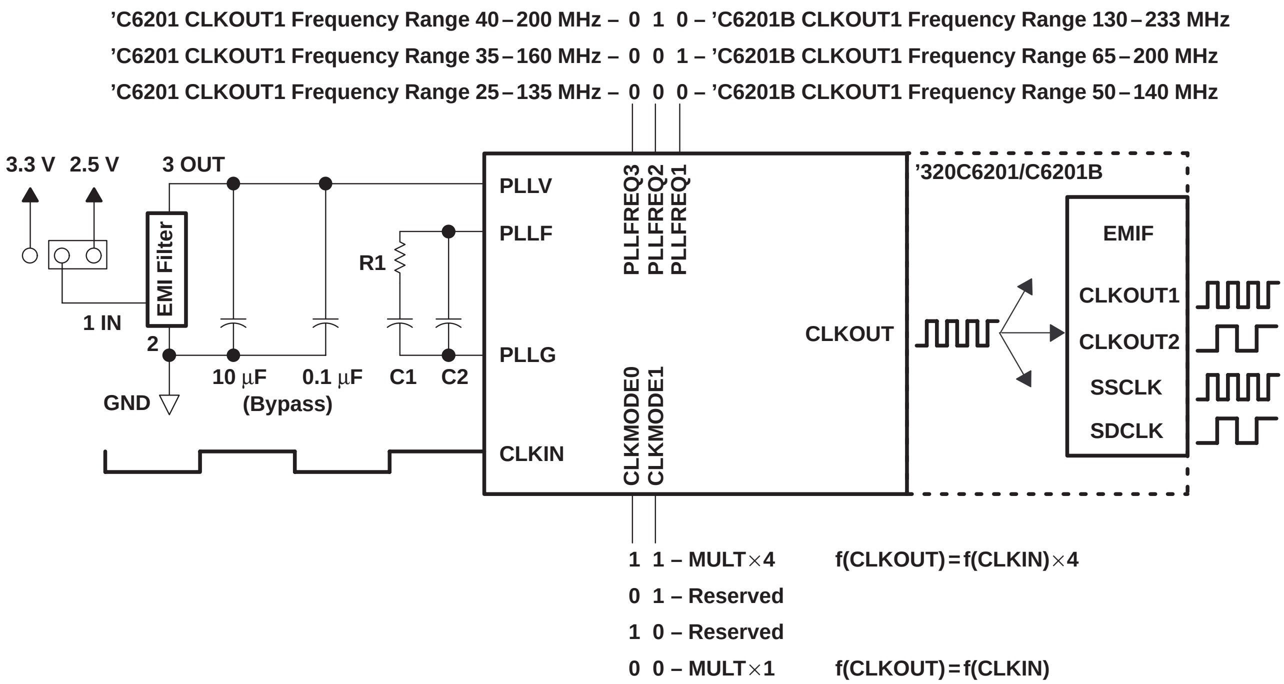
Figure 4. TMS320C6200 CPU Data Paths

clock PLL

All of the ’C62xx 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 core voltage (2.5 V), and for ’C6201B, it must be powered by the I/O voltage (3.3 V).

To configure the ’C62xx PLL clock for proper operation, see Figure 5 and Table 1. To minimize the clock jitter, a single clean power supply should power both the ’C62x 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. For the ’C6201 CLKMODE x4, values for C1, C2, and R1 depend on CLKIN and CLKOUT frequencies.
For the ’C6201B 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, CLKOUT1 = 133 MHz, a PLLFREQ value of 000b should be used for both the ’C6201 and the ’C6201B. For CLKOUT1 = 200 MHz, PLLFREQ should be set to 010b for the ’C6201 or 001b for the ’C6201B. PLLFREQ values other than 000b, 001b, and 010b are reserved.
D. EMI filter manufacturer TDK part number ACF451832-153-T
E. For the ’C6201B, 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}.

Figure 5. PLL Block Diagram

clock PLL (continued)

Table 1. TMS320C6201 PLL Component Selection Table†

CYCLE TIME (ns)	CLKMODE	CLKIN (MHz)	CLKOUT1 (MHz)	R1 (Ω)	C1 (μF)	C2 (pF)	EMI FILTER PART NO.‡	TYPICAL LOCK TIME (μs)§
5	x4	50	200	16.9	0.15	2700	TDK #153	59
5.5	x4	45.5	181.8	13.7	0.18	3900	TDK #153	49
6	x4	41.6	166.7	17.4	0.15	3300	TDK #153	68
6.5	x4	38.5	153.8	16.2	0.18	3900	TDK #153	70
7	x4	35.7	142.9	15	0.22	3900	TDK #153	72
7.5	x4	33.3	133.3	16.2	0.22	3900	TDK #153	84
8	x4	31.3	125	14	0.27	4700	TDK #153	77
8.5	x4	29.4	117.7	11.8	0.33	6800	TDK #153	67
9	x4	27.7	111.1	11	0.39	6800	TDK #153	68
9.5	x4	26.3	105.3	10.5	0.39	8200	TDK #153	65
10	x4	25	100	10	0.47	8200	TDK #153	68

† 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.
‡ Full EMI filter part number : ACF 451832-153-T
§ 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.

Table 2. TMS320C6201B PLL Component Selection Table†

CLKMODE	R1 (Ω)	C1 (nF)	C2 (pF)	EMI FILTER PART NO.‡	TYPICAL LOCK TIME (μs)§
x4	60.4	27	560	TDK #153	75

† 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.
‡ Full EMI filter part number : ACF 451832-153-T
§ 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 the ’C6201 device, the 2.5-V supply powers the core and the 3.3-V supply powers the I/O buffers. For the ’C6201B device, the 1.8-V supply powers the core and the 3.3-V supply powers the I/O buffers. 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.

development support

Texas Instruments (TI™) offers an extensive line of development tools for the ’C6200 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 ’C6200-based applications:

Software Development Tools:
Assembly optimizer
Assembler/Linker
Simulator
Optimizing ANSI C compiler
Application algorithms
C/Assembly debugger and code profiler

Hardware Development Tools:
Extended development system (XDS™) emulator (supports ’C6200 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 3 for a complete listing of development-support tools for the ’C6200. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Table 3. TMS320C6xx Development-Support Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Software		
C Compiler/Assembler/Linker/Assembly Optimizer	Win32™	TMDX3246855-07
C Compiler/Assembler/Linker/Assembly Optimizer	SPARC™ Solaris™	TMDX324655-07
Simulator	Win32	TMDS3246851-07
Simulator	SPARC Solaris	TMDS3246551-07
XDS510™ Debugger/Emulation Software	Win32, Windows NT™	TMDX324016X-07
Hardware		
XDS510 Emulator†	PC	TMDS00510
XDS510WS™ Emulator‡	SCSI	TMDS00510WS
Software/Hardware		
EVM Evaluation Kit	PC/Win95/Windows NT	TMDX3260A6201
EVM Evaluation Kit (including TMDX3246855–07)	PC/Win95/Windows NT	TMDX326006201

† Includes XDS510 board and JTAG emulation cable. TMDX324016X-07 C-source Debugger/Emulation software is not included.
‡ Includes XDS510WS box, SCSI cable, power supply, and JTAG emulation cable.

TI, XDS, XDS510, and XDS510WS are trademarks of Texas Instruments Incorporated.
Win32 and Windows NT are trademarks of Microsoft Corporation.
SPARC is a trademark of SPARC International, Inc.
Solaris is a trademark of Sun Microsystems, Inc.

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). This development flow follows.

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, GGP, GJC, or GJL) and the device speed range in megahertz (for example, -200 is 200 MHz). Figure 6 provides a legend for reading the complete device name for any TMS320 family member.

device and development-support tool nomenclature (continued)

TMS 320 C 6201 GGP –200

PREFIX

- TMX = Experimental device
- TMP = Prototype device
- TMS = Qualified device
- SMJ = MIL-STD-883C
- SM = High Rel (non-883C)

DEVICE FAMILY

- 320 = TMS320 family

TECHNOLOGY

- C = CMOS
- E = CMOS EPROM
- F = CMOS Flash EEPROM

DEVICE SPEED RANGE

- –150 MHz
- –167 MHz
- –200 MHz
- –233 MHz
- –250 MHz

PACKAGE TYPE†

- N = Plastic DIP
- J = Ceramic DIP
- JD = Ceramic DIP side-brazed
- GB = Ceramic PGA
- FZ = Ceramic CC
- FN = Plastic leaded CC
- FD = Ceramic leadless CC
- PJ = 100-pin plastic EIAJ QFP
- PQ = 132-pin plastic bumpered QFP
- PZ = 100-pin plastic TQFP
- PBK = 128-pin plastic TQFP
- PGE = 144-pin plastic TQFP
- GFN = 272-pin plastic BGA
- GGU = 144-pin plastic BGA
- GGP = 352-pin plastic BGA
- GJC = 352-pin plastic BGA
- GJL = 352-pin plastic BGA
- GJL = 452-pin plastic BGA

DEVICE

’1x DSP:
10 16
14 17
15

’2x DSP:
25
26

’2xx DSP:
203 206 240
204 209

’3x DSP:
30
31
32

’4x DSP:
40
44

’5x DSP:
50 53
51 56
52 57

’54x DSP:
541 545
542 546
543 548

’6x DSP:
6201
6201B
6202
6211
6701

† DIP = Dual-In-Line Package
PGA = Pin Grid Array
CC = Chip Carrier
QFP = Quad Flat Package
TQFP = Thin Quad Flat Package
BGA = Ball Grid Array

Figure 6. TMS320 Device Nomenclature (Including TMS320C6201/TMS320C6201B)

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; technical briefs; development-support tools; and hardware and software applications. The following is a brief, descriptive list of support documentation specific to the 'C6x devices:

The *TMS320C62x/C67x CPU and Instruction Set Reference Guide* (literature number SPRU189) describes the 'C62x/C67x CPU architecture, instruction set, pipeline, and associated interrupts.

The *TMS320C6201/C6701 Peripherals Reference Guide* (literature number SPRU190) describes functionally 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) controller, clocking and phase-locked loop (PLL); and power-down modes. This guide also includes information on internal data and program memories.

The *TMS320C62x/C67x Programmer's Guide* (literature number SPRU198) describes ways to optimize C and assembly code for 'C6x devices and includes application program examples.

The *TMS320C6x Optimizing C Compiler User's Guide* (literature number SPRU187) describes the 'C6x C compiler and the assembly optimizer, explaining that the C compiler accepts ANSI standard C source code, and produces assembly language source code for the 'C6x generation devices, and that the assembly optimizer helps to optimize the programmer's assembly code.

The *TMS320C6x C Source Debugger User's Guide* (literature number SPRU188) describes how to invoke the 'C6x simulator and emulator versions of the C source debugger interface and discusses various aspects of the debugger, including: command entry, code execution, data management, breakpoints, profiling, and analysis.

The *TMS320C6x Peripheral Support Library Programmer's Reference* (literature number SPRU273) describes the contents of the 'C6x peripheral support library of functions and macros. It lists functions and macros both by header file and alphabetically, provides a complete description of each, and gives code examples to show how they are used.

The *TMS320C6x Evaluation Module Reference Guide* (literature number SPRU269) provides instructions for installing and operating the 'C6x evaluation module. It also includes support software documentation, application programming interfaces, and technical reference material.

The *TMS320C62x/C67x Technical Brief* (literature number SPRU197) gives an introduction to the 'C62x/C67x devices, associated development tools, and third-party support.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support DSP research and education. The TMS320 newsletter, *Details on Signal Processing*, is published quarterly and distributed to update TMS320 customers on product information. The TMS320 DSP bulletin board service (BBS) provides access to information pertaining to the TMS320 family, including documentation, source code, and object code for many DSP algorithms and utilities. The BBS can be reached at 281/274-2323.

Information regarding TI DSP products is also available on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

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

Supply voltage range, CV_{DD} (see Note 1) for ’C6201 . . . . –0.3 V to 3 V
Supply voltage range, CV_{DD} (see Note 1) for ’C6201B . . . . –0.3 V to 2.3 V
Supply voltage range, DV_{DD} (see Note 1) . . . . –0.3 V to 4 V
Input voltage range . . . . –0.3 V to 4 V
Output voltage range . . . . –0.3 V to 4 V
Operating case temperature range, T_C . . . . 0°C to 90°C
Storage temperature range, T_{stg} . . . . –55°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

		’C6201			’C6201B			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
CV_{DD}	Supply voltage	2.38	2.50	2.62	1.71	1.8	1.89	V
DV_{DD}	Supply voltage	3.14	3.30	3.46	3.14	3.30	3.46	V
V_{SS}	Supply ground	0	0	0	0	0	0	V
V_{IH}	High-level input voltage	2.0			2.0			V
V_{IL}	Low-level input voltage			0.8			0.8	V
I_{OH}	High-level output current			–12			–12	mA
I_{OL}	Low-level output current			12			12	mA
T_C	Operating case temperature	0		90	0		90	°C

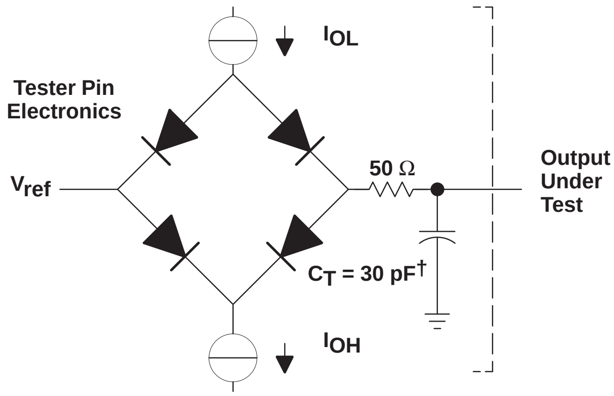
PRODUCT PREVIEW

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

PARAMETER		TEST CONDITIONS	’C6201			’C6201B			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
V_{OH}	High-level output voltage	DV_{DD} = MIN, I_{OH} = MAX	2.4			2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN, I_{OL} = MAX			0.6			0.6	V
I_I	Input current†	V_I = V_{SS} to DV_{DD}			±10			±10	uA
I_{OZ}	Off-state output current	V_O = DV_{DD} or 0 V			±10			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access‡	CV_{DD} = NOM, CPU clock = 167 MHz		1860			780		mA
I_{DD2V}	Supply current, peripherals§	CV_{DD} = NOM, CPU clock = 167 MHz		200			140		mA
I_{DD3V}	Supply current, I/O pins¶	DV_{DD} = NOM, CPU clock = 167 MHz		100			100		mA
C_i	Input capacitance				10			10	pF
C_o	Output capacitance				10			10	pF

† TMS and TDI are not included due to internal pullups.
TRST is not included due to internal pulldown.
‡ Measured with average CPU activity:
50% of time: 8 instructions per cycle, 32-bit DMEM access per cycle
50% of time: 2 instructions per cycle, 16-bit DMEM access per cycle
§ Measured with average peripheral activity:
50% of time: Timers at max rate
McBSPs at E1 rate
DMA burst transfer between DMEM and SDRAM
50% of time: Timers at max rate
McBSPs at E1 rate
DMA servicing McBSPs
¶ Measured with average I/O activity (30-pF load, SDCLK on):
25% of time: Reads from external SDRAM
25% of time: Writes to external SDRAM
50% of time: No activity

PARAMETER MEASUREMENT INFORMATION



† Typical distributed load circuit capacitance

Figure 7. 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.

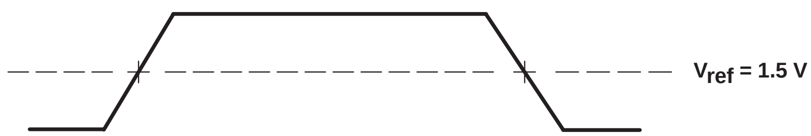


Figure 8. Input and Output Voltage Reference Levels for AC Timing Measurements

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN† (see Figure 9) (’C6201)

NO.			’C6201-167				’C6201-200				UNIT
			CLKMODE = x4		CLKMODE = x1		CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	24		6		20		5		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	9.8		2.7		8		2.25		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	9.8		2.7		8		2.25		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		0.6		5		0.6	ns

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of V_{IH}.

timing requirements for CLKIN (see Figure 9) (’C6201B)

NO.			’C6201B-167				’C6201B-200				’C6201B-233				UNIT
			CLKMODE = x4		CLKMODE = x1		CLKMODE = x4		CLKMODE = x1		CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	24		6		20		5		17.2		4.3		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	9.8		2.7		8		2.25		6.9		1.9		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	9.8		2.7		8		2.25		6.9		1.9		ns
4	$t_{t(CLKIN)}$	Transition time, CLKIN		5		0.6		5		0.6		5		0.6	ns

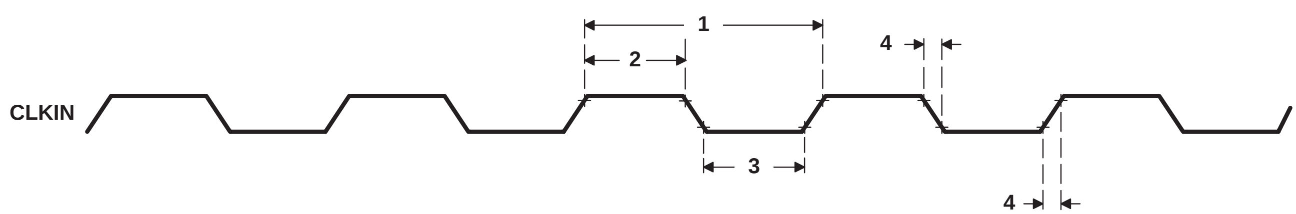


Figure 9. CLKIN Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT1†‡ (see Figure 10) (’C6201)

NO.	PARAMETER		’C6201-167 ’C6201-200				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

† PH is the high period of CLKOUT1 in ns and PL is the low period of CLKOUT1 in ns.
‡ P = 1/CPU clock frequency in nanoseconds (ns).

switching characteristics for CLKOUT1†‡ (see Figure 10) (’C6201B)

NO.	PARAMETER		’C6201B-167 ’C6201B-200 ’C6201B-233				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

† PH is the high period of CLKOUT1 in ns and PL is the low period of CLKOUT1 in ns.
‡ P = 1/CPU clock frequency in nanoseconds (ns).

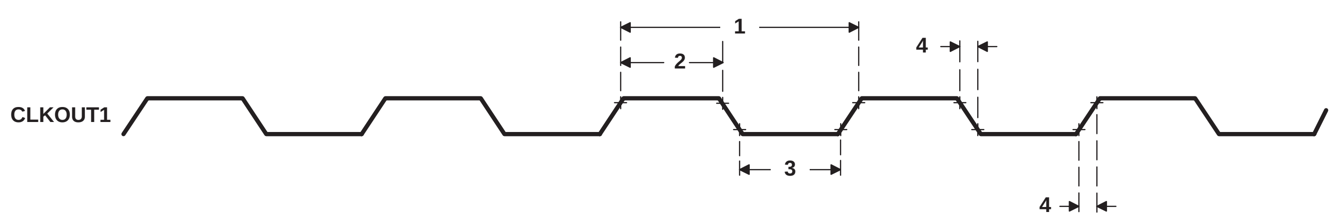


Figure 10. CLKOUT1 Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT2† (see Figure 11)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	2P – 0.7	2P + 0.7	2P – 0.7	2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	P – 0.7	P + 0.7	P – 0.7	P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	P – 0.7	P + 0.7	P – 0.7	P + 0.7	ns
4	$t_{t(CKO2)}$	Transition time, CLKOUT2		0.6		0.6	ns

† P = 1/CPU clock frequency in ns.

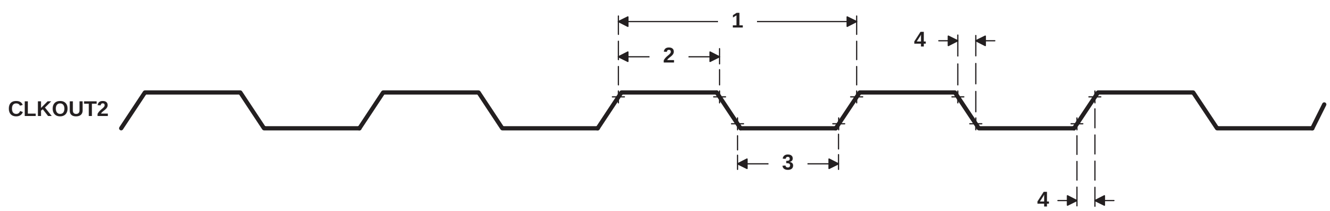


Figure 11. CLKOUT2 Timings

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 for the relation of SSCLK, SDCLK, and CLKOUT2 to CLKOUT1 (see Figure 12)†

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(CKO1-SSCLK)}$	Delay time, CLKOUT1 edge to SSCLK edge	–1.2	1.6	(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)	–1.0	2.4	(P/2) – 1	(P/2) + 2.4	ns
3	$t_{d(CKO1-CKO2)}$	Delay time, CLKOUT1 edge to CLKOUT2 edge	–1.0	2.4	(P/2) – 1	(P/2) + 2.4	ns
4	$t_{d(CKO1-SDCLK)}$	Delay time, CLKOUT1 edge to SDCLK edge	–1.0	2.4	(P/2) – 1	(P/2) + 2.4	ns

† P = 1/CPU clock frequency in ns.

INPUT AND OUTPUT CLOCKS (CONTINUED)

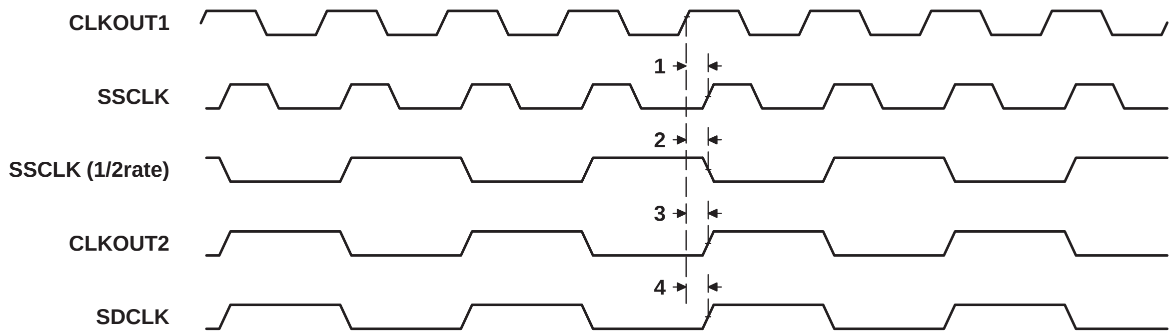


Figure 12. Relation of CLKOUT2, SDCLK, and SSCLK to CLKOUT1

ASYNCHRONOUS MEMORY TIMING

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

NO.			’C6201-167 ’C6201-200		’C6201B-167		’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
6	$t_{su(EDV-CKO1H)}$	Setup time, read EDx valid before CLKOUT1 high	5.0		5.0		4.0		ns
7	$t_{h(CKO1H-EDV)}$	Hold time, read EDx valid after CLKOUT1 high	0		0		0.8		ns
10	$t_{su(ARDY-CKO1H)}$	Setup time, ARDY valid before CLKOUT1 high	5.0		5.0		4.0		ns
11	$t_{h(CKO1H-ARDY)}$	Hold time, ARDY valid after CLKOUT1 high	0		0		0.8		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 for asynchronous memory cycles‡ (see Figure 13 and Figure 14)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167		’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{d(CKO1H-CEV)}$	Delay time, CLKOUT1 high to $\overline{\text{CEx}}$ valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
2	$t_{d(CKO1H-BEV)}$	Delay time, CLKOUT1 high to $\overline{\text{BEx}}$ valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
3	$t_{d(CKO1H-BEIV)}$	Delay time, CLKOUT1 high to $\overline{\text{BEx}}$ invalid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
4	$t_{d(CKO1H-EAV)}$	Delay time, CLKOUT1 high to EAx valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
5	$t_{d(CKO1H-EAIV)}$	Delay time, CLKOUT1 high to EAx invalid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
8	$t_{d(CKO1H-AOEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AOE}}$ valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
9	$t_{d(CKO1H-AREV)}$	Delay time, CLKOUT1 high to $\overline{\text{ARE}}$ valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns
12	$t_{d(CKO1H-EDV)}$	Delay time, CLKOUT1 high to EDx valid		5.0		5.0		4.0	ns
13	$t_{d(CKO1H-EDIV)}$	Delay time, CLKOUT1 high to EDx invalid	–1.0		–1.0		–0.2		ns
14	$t_{d(CKO1H-AWEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AWE}}$ valid	–1.0	5.0	–1.0	5.0	–0.2	4.0	ns

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

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

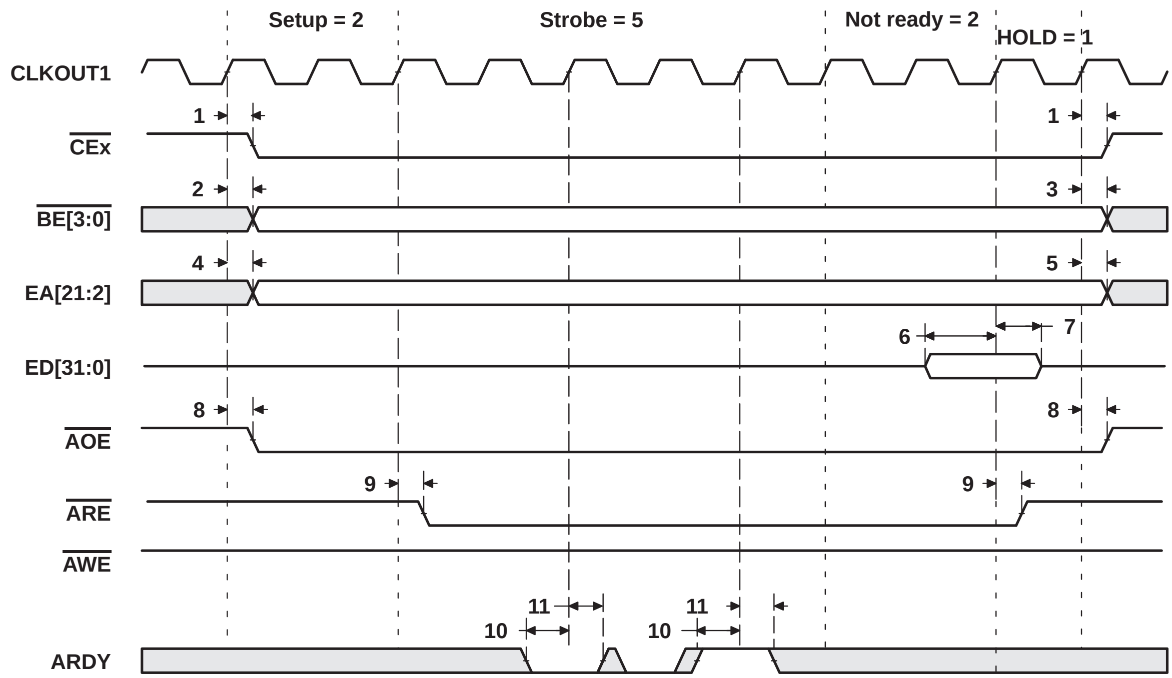


Figure 13. Asynchronous Memory Read Timing

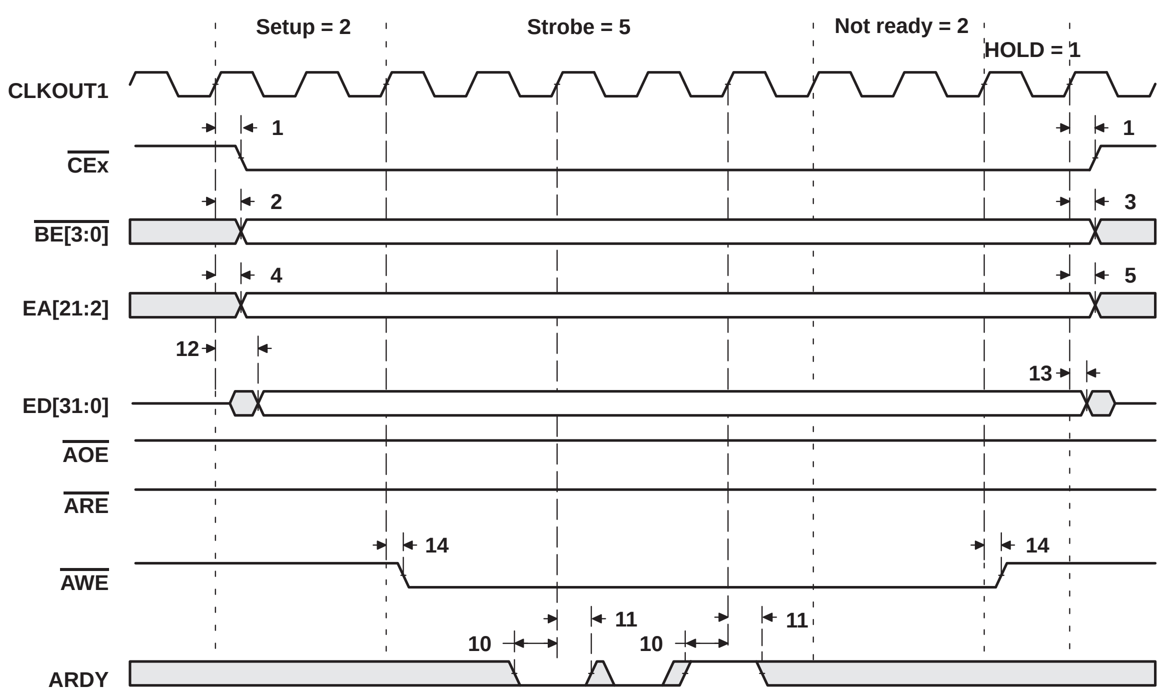


Figure 14. Asynchronous Memory Write Timing

SYNCHRONOUS-BURST MEMORY TIMING

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

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
7	$t_{su(EDV-SSCLKH)}$	Setup time, read EDx valid before SSCLK high	1.5		1.5		ns
8	$t_{h(SSCLKH-EDV)}$	Hold time, read EDx valid after SSCLK high	1.2		1.5		ns

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

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(CEV-SSCLKH)}$	Setup time, $\overline{CEx}$ valid before SSCLK high	P – 4		0.5P – 1.3		ns
2	$t_{oh(SSCLKH-CEV)}$	Output hold time, $\overline{CEx}$ valid after SSCLK high	0		0.5P – 2.3		ns
3	$t_{su(BEV-SSCLKH)}$	Setup time, $\overline{BEx}$ valid before SSCLK high	P – 4		0.5P – 1.3		ns
4	$t_{oh(SSCLKH-BEIV)}$	Output hold time, $\overline{BEx}$ invalid after SSCLK high	1		0.5P – 2.3		ns
5	$t_{su(EAV-SSCLKH)}$	Setup time, EAx valid before SSCLK high	P – 4		0.5P – 1.3		ns
6	$t_{oh(SSCLKH-EAIV)}$	Output hold time, EAx invalid after SSCLK high	1		0.5P – 2.3		ns
9	$t_{su(ADSV-SSCLKH)}$	Setup time, $\overline{SSADS}$ valid before SSCLK high	P – 3		0.5P – 1.3		ns
10	$t_{oh(SSCLKH-ADSV)}$	Output hold time, $\overline{SSADS}$ valid after SSCLK high	0		0.5P – 2.3		ns
11	$t_{su(OEV-SSCLKH)}$	Setup time, $\overline{SSOE}$ valid before SSCLK high	P – 4		0.5P – 1.3		ns
12	$t_{oh(SSCLKH-OEV)}$	Output hold time, $\overline{SSOE}$ valid after SSCLK high	0		0.5P – 2.3		ns
13	$t_{su(EDV-SSCLKH)}$	Setup time, EDx valid before SSCLK high	P – 4		0.5P – 1.3		ns
14	$t_{oh(SSCLKH-EDIV)}$	Output hold time, EDx invalid after SSCLK high	1		0.5P – 2.3		ns
15	$t_{su(WEV-SSCLKH)}$	Setup time, $\overline{SSWE}$ valid before SSCLK high	P – 3		0.5P – 1.3		ns
16	$t_{oh(SSCLKH-WEV)}$	Output hold time, $\overline{SSWE}$ valid after SSCLK high	0		0.5P – 2.3		ns

† The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SSCLK duty cycle.
P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

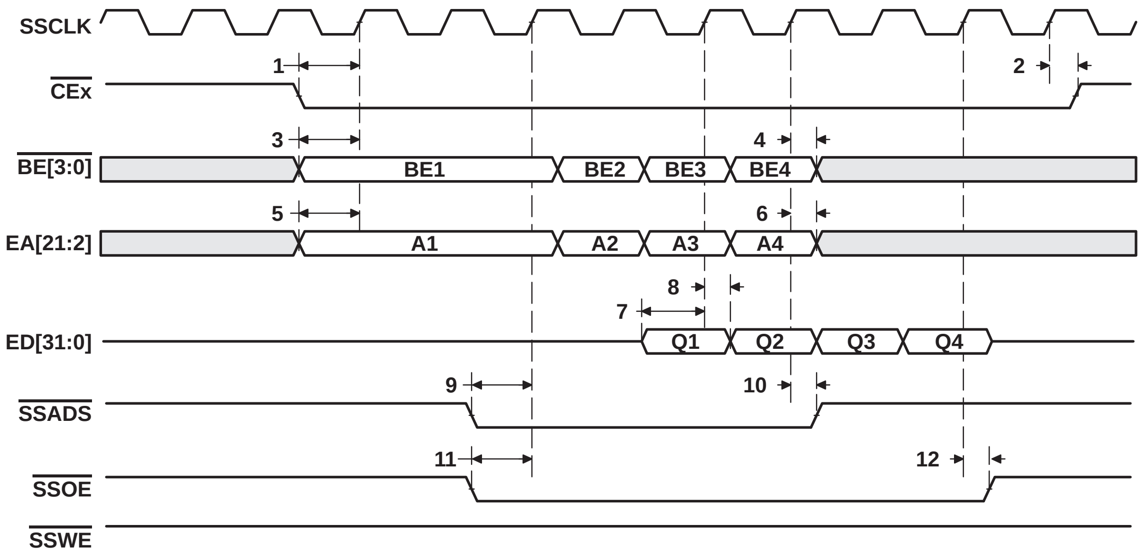


Figure 15. SBSRAM Read Timing (Full-Rate SSCLK)

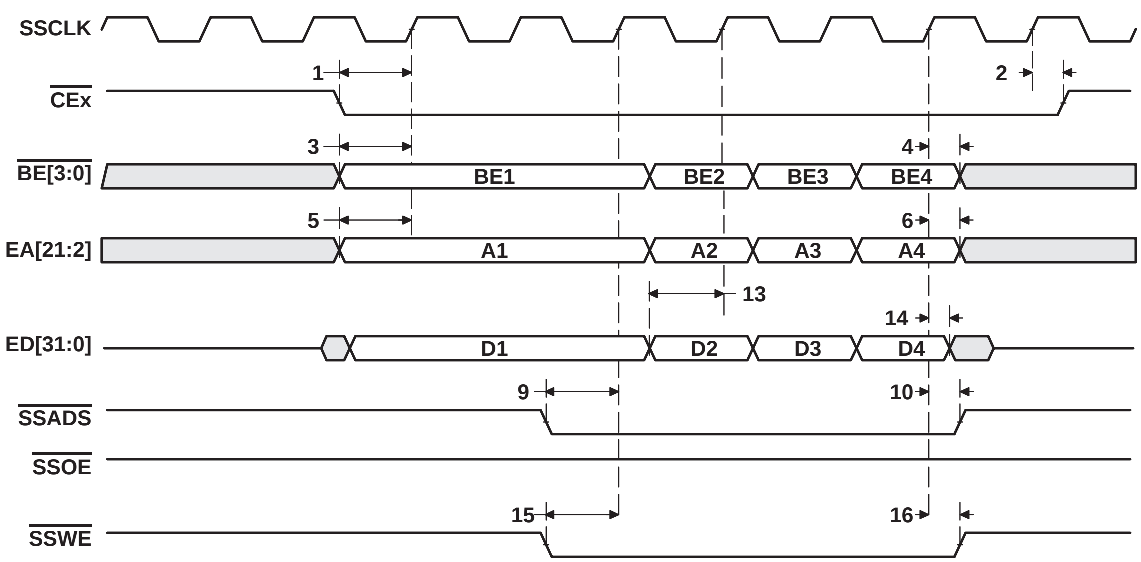


Figure 16. SBSRAM Write Timing (Full-Rate SSCLK)

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

timing requirements for synchronous-burst SRAM cycles (half-rate SSCLK) (see Figure 17) (’C6201)

NO.			’C6201-167		’C6201-200		UNIT
			MIN	MAX	MIN	MAX	
7	$t_{su(EDV-SSCLKH)}$	Setup time, read EDx valid before SSCLK high	3.6		3.6		ns
8	$t_{h(SSCLKH-EDV)}$	Hold time, read EDx valid after SSCLK high	1.2		1.2		ns

switching characteristics for synchronous-burst SRAM cycles† (half-rate SSCLK) (see Figure 17 and Figure 18) (’C6201)

NO.	PARAMETER		’C6201-167		’C6201-200		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(CEV-SSCLKH)}$	Setup time, $\overline{CEx}$ valid before SSCLK high	P – 3.4		P – 3.4		ns
2	$t_{oh(SSCLKH-CEV)}$	Output hold time, $\overline{CEx}$ valid after SSCLK high	P – 5		P – 4		ns
3	$t_{su(BEV-SSCLKH)}$	Setup time, $\overline{BEx}$ valid before SSCLK high	P – 3.3		P – 2.3		ns
4	$t_{oh(SSCLKH-BEIV)}$	Output hold time, $\overline{BEx}$ invalid after SSCLK high	P – 5		P – 4		ns
5	$t_{su(EAV-SSCLKH)}$	Setup time, EAx valid before SSCLK high	P – 3.3		P – 2.3		ns
6	$t_{oh(SSCLKH-EAIV)}$	Output hold time, EAx invalid after SSCLK high	P – 5		P – 4		ns
9	$t_{su(ADSV-SSCLKH)}$	Setup time, $\overline{SSADS}$ valid before SSCLK high	P – 3.3		P – 2.3		ns
10	$t_{oh(SSCLKH-ADSV)}$	Output hold time, $\overline{SSADS}$ valid after SSCLK high	P – 5		P – 4		ns
11	$t_{su(OEV-SSCLKH)}$	Setup time, $\overline{SSOE}$ valid before SSCLK high	P – 3.3		P – 3.1		ns
12	$t_{oh(SSCLKH-OEV)}$	Output hold time, $\overline{SSOE}$ valid after SSCLK high	P – 5		P – 4		ns
13	$t_{su(EDV-SSCLKH)}$	Setup time, EDx valid before SSCLK high	P – 3.3		P – 2.3		ns
14	$t_{oh(SSCLKH-EDIV)}$	Output hold time, EDx invalid after SSCLK high	P – 5		P – 4		ns
15	$t_{su(WEV-SSCLKH)}$	Setup time, $\overline{SSWE}$ valid before SSCLK high	P – 3.3		P – 2.3		ns
16	$t_{oh(SSCLKH-WEV)}$	Output hold time, $\overline{SSWE}$ valid after SSCLK high	P – 5		P – 4		ns

† The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SSCLK duty cycle. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

timing requirements for synchronous-burst SRAM cycles (half-rate SSCLK) (see Figure 17) (’C6201B)

NO.			’C6201B-167		’C6201B-200		’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
7	$t_{su(EDV-SSCLKH)}$	Setup time, read EDx valid before SSCLK high	3.5		2.5		1.1		ns
8	$t_{h(SSCLKH-EDV)}$	Hold time, read EDx valid after SSCLK high	1.5		1.5		1.5		ns

switching characteristics for synchronous-burst SRAM cycles† (half-rate SSCLK) (see Figure 17 and Figure 18) (’C6201B)

NO.	PARAMETER		’C6201B-167		’C6201B-200		’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{su(CEV-SSCLKH)}$	Setup time, $\overline{CEx}$ valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
2	$t_{oh(SSCLKH-CEV)}$	Output hold time, $\overline{CEx}$ valid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
3	$t_{su(BEV-SSCLKH)}$	Setup time, $\overline{BEx}$ valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
4	$t_{oh(SSCLKH-BEIV)}$	Output hold time, $\overline{BEx}$ invalid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
5	$t_{su(EAV-SSCLKH)}$	Setup time, EAx valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
6	$t_{oh(SSCLKH-EAIV)}$	Output hold time, EAx invalid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
9	$t_{su(ADSV-SSCLKH)}$	Setup time, $\overline{SSADS}$ valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
10	$t_{oh(SSCLKH-ADSV)}$	Output hold time, $\overline{SSADS}$ valid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
11	$t_{su(OEV-SSCLKH)}$	Setup time, $\overline{SSOE}$ valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
12	$t_{oh(SSCLKH-OEV)}$	Output hold time, $\overline{SSOE}$ valid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
13	$t_{su(EDV-SSCLKH)}$	Setup time, EDx valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
14	$t_{oh(SSCLKH-EDIV)}$	Output hold time, EDx invalid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns
15	$t_{su(WEV-SSCLKH)}$	Setup time, $\overline{SSWE}$ valid before SSCLK high	1.5P – 4.5		1.5P – 3		1.5P – 2.2		ns
16	$t_{oh(SSCLKH-WEV)}$	Output hold time, $\overline{SSWE}$ valid after SSCLK high	0.5P – 2		0.5P – 1.5		0.5P – 1.1		ns

† The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SSCLK duty cycle. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

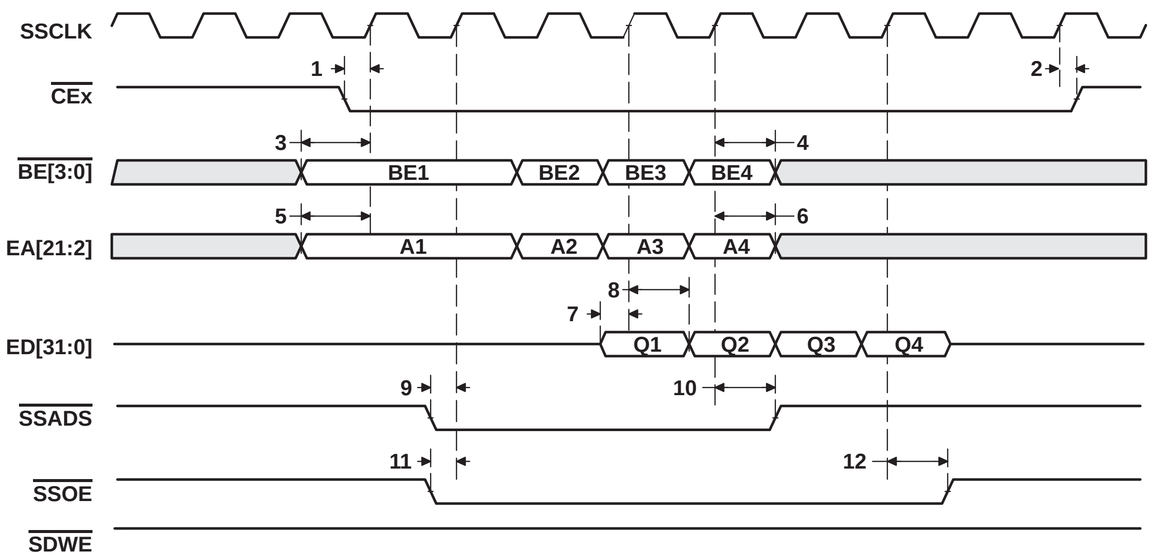


Figure 17. SBSRAM Read Timing (1/2 Rate SSCLK)

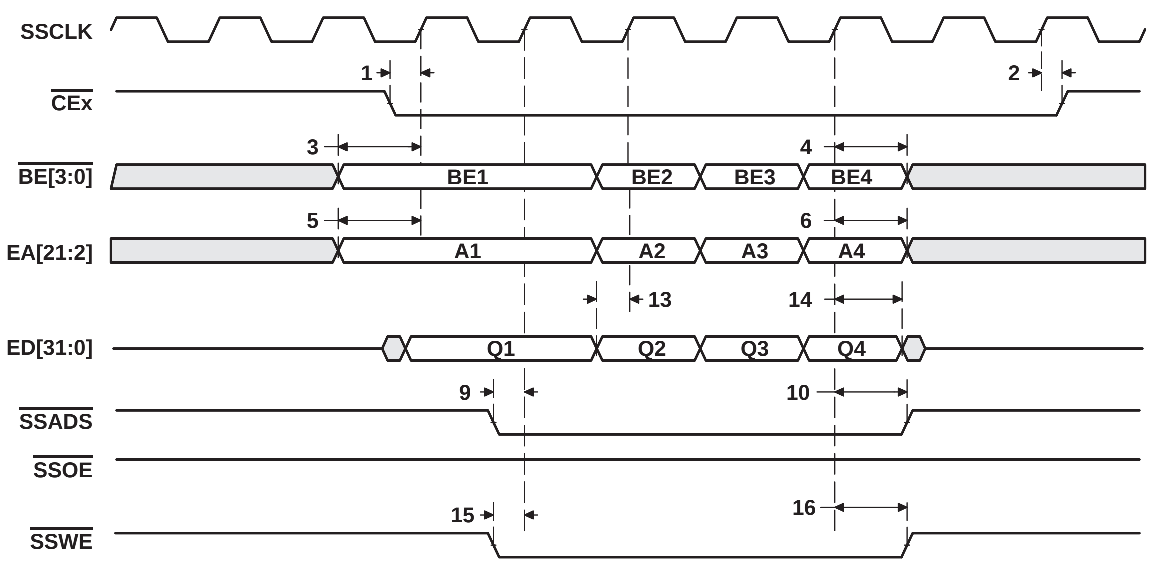


Figure 18. SBSRAM Write Timing (1/2 Rate SSCLK)

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles (see Figure 19) (’C6201)

NO.			’C6201-167		’C6201-200		UNIT
			MIN	MAX	MIN	MAX	
7	$t_{su(EDV\text{-}SDCLKH)}$	Setup time, read EDx valid before SDCLK high	3.5		1.5		ns
8	$t_{h(SDCLKH\text{-}EDV)}$	Hold time, read EDx valid after SDCLK high	1.2		1.2		ns

switching characteristics for synchronous DRAM cycles† (see Figure 19–Figure 24) (’C6201)

NO.	PARAMETER		’C6201-167		’C6201-200		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(CEV\text{-}SDCLKH)}$	Setup time, $\overline{CE}$x valid before SDCLK high	P – 3.5		P – 2.5		ns
2	$t_{oh(SDCLKH\text{-}CEV)}$	Output hold time, $\overline{CE}$x valid after SDCLK high	P – 4.5		P – 3.5		ns
3	$t_{su(BEV\text{-}SDCLKH)}$	Setup time, $\overline{BE}$x valid before SDCLK high	P – 3.5		P – 2.5		ns
4	$t_{oh(SDCLKH\text{-}BEIV)}$	Output hold time, $\overline{BE}$x invalid after SDCLK high	P – 4.5		P – 3.5		ns
5	$t_{su(EAV\text{-}SDCLKH)}$	Setup time, EAx valid before SDCLK high	P – 3.5		P – 2.5		ns
6	$t_{oh(SDCLKH\text{-}EAIV)}$	Output hold time, EAx invalid after SDCLK high	P – 4.5		P – 3.5		ns
9	$t_{su(SDCAS\text{-}SDCLKH)}$	Setup time, $\overline{SDCAS}$ valid before SDCLK high	P – 3.5		P – 2.5		ns
10	$t_{oh(SDCLKH\text{-}SDCAS)}$	Output hold time, $\overline{SDCAS}$ valid after SDCLK high	P – 4.5		P – 3.5		ns
11	$t_{su(EDV\text{-}SDCLKH)}$	Setup time, EDx valid before SDCLK high	P – 3.5		P – 2.5		ns
12	$t_{oh(SDCLKH\text{-}EDIV)}$	Output hold time, EDx invalid after SDCLK high	P – 4.5		P – 3.5		ns
13	$t_{su(SDWE\text{-}SDCLKH)}$	Setup time, $\overline{SDWE}$ valid before SDCLK high	P – 3.5		P – 2.5		ns
14	$t_{oh(SDCLKH\text{-}SDWE)}$	Output hold time, $\overline{SDWE}$ valid after SDCLK high	P – 4.5		P – 3.5		ns
15	$t_{su(SDA10V\text{-}SDCLKH)}$	Setup time, SDA10 valid before SDCLK high	P – 3.5		P – 2.5		ns
16	$t_{oh(SDCLKH\text{-}SDA10IV)}$	Output hold time, SDA10 invalid after SDCLK high	P – 4.5		P – 3.5		ns
17	$t_{su(SDRAS\text{-}SDCLKH)}$	Setup time, $\overline{SDRAS}$ valid before SDCLK high	P – 3.5		P – 2.5		ns
18	$t_{oh(SDCLKH\text{-}SDRAS)}$	Output hold time, $\overline{SDRAS}$ valid after SDCLK high	P – 4.5		P – 3.5		ns

† The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SDCLK duty cycle. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

SYNCHRONOUS DRAM TIMING (CONTINUED)

timing requirements for synchronous DRAM cycles (see Figure 19) (’C6201B)

NO.			’C6201B-167		’C6201B-200		’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
7	$t_{su(EDV-SDCLKH)}$	Setup time, read EDx valid before SDCLK high	1.5		1		1		ns
8	$t_{h(SDCLKH-EDV)}$	Hold time, read EDx valid after SDCLK high	3		3		3		ns

switching characteristics for synchronous DRAM cycles† (see Figure 19–Figure 24) (’C6201B)

NO.	PARAMETER		’C6201B-167		’C6201B-200		’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{su(CEV-SDCLKH)}$	Setup time, $\overline{CE}$x valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
2	$t_{oh(SDCLKH-CEV)}$	Output hold time, $\overline{CE}$x valid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
3	$t_{su(BEV-SDCLKH)}$	Setup time, $\overline{BE}$x valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
4	$t_{oh(SDCLKH-BEIV)}$	Output hold time, $\overline{BE}$x invalid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
5	$t_{su(EAV-SDCLKH)}$	Setup time, EAx valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
6	$t_{oh(SDCLKH-EAIV)}$	Output hold time, EAx invalid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
9	$t_{su(SDCAS-SDCLKH)}$	Setup time, $\overline{SDCAS}$ valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
10	$t_{oh(SDCLKH-SDCAS)}$	Output hold time, $\overline{SDCAS}$ valid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
11	$t_{su(EDV-SDCLKH)}$	Setup time, EDx valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
12	$t_{oh(SDCLKH-EDIV)}$	Output hold time, EDx invalid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
13	$t_{su(SDWE-SDCLKH)}$	Setup time, $\overline{SDWE}$ valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
14	$t_{oh(SDCLKH-SDWE)}$	Output hold time, $\overline{SDWE}$ valid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
15	$t_{su(SDA10V-SDCLKH)}$	Setup time, SDA10 valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
16	$t_{oh(SDCLKH-SDA10IV)}$	Output hold time, SDA10 invalid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns
17	$t_{su(SDRAS-SDCLKH)}$	Setup time, $\overline{SDRAS}$ valid before SDCLK high	1.5P – 4		1.5P – 3.5		1.5P – 2.4		ns
18	$t_{oh(SDCLKH-SDRAS)}$	Output hold time, $\overline{SDRAS}$ valid after SDCLK high	0.5P – 1.5		0.5P – 1		0.5P – 0.6		ns

† The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SDCLK duty cycle. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

SYNCHRONOUS DRAM TIMING (CONTINUED)

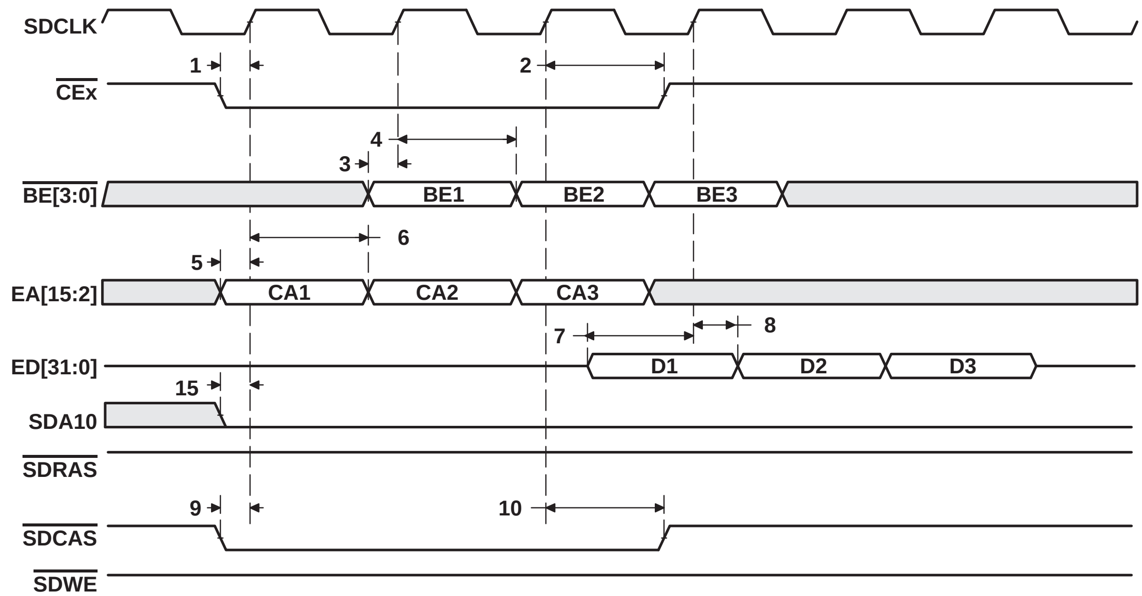


Figure 19. Three SDRAM Read Commands

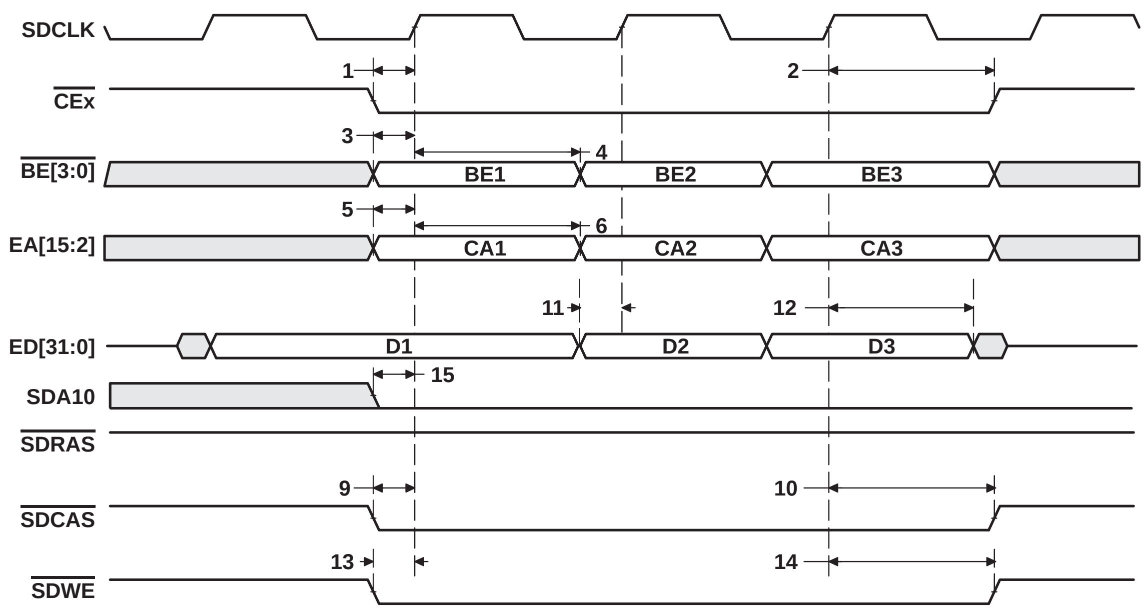


Figure 20. Three SDRAM WRT Commands

SYNCHRONOUS DRAM TIMING (CONTINUED)

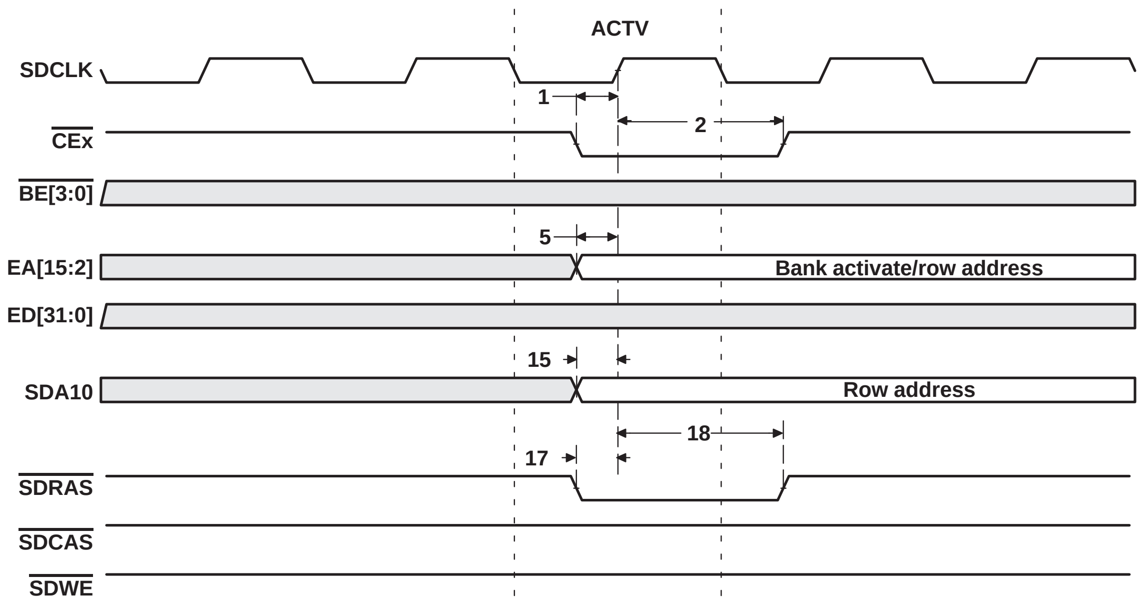


Figure 21. SDRAM ACTV Command

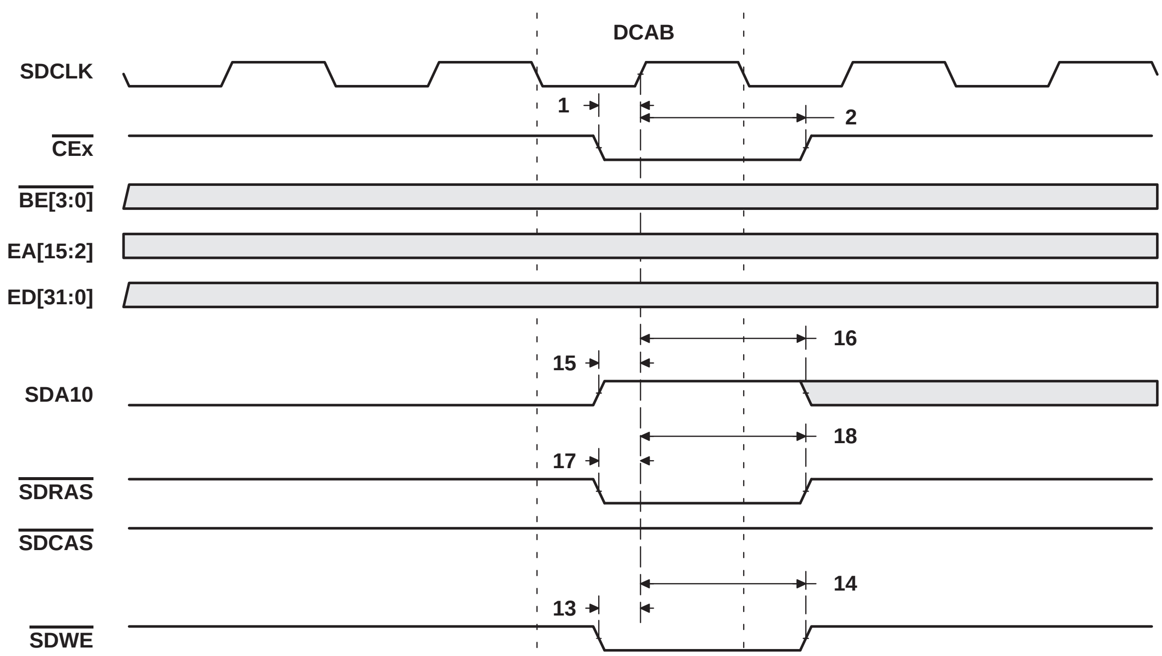


Figure 22. SDRAM DCAB Command

SYNCHRONOUS DRAM TIMING (CONTINUED)

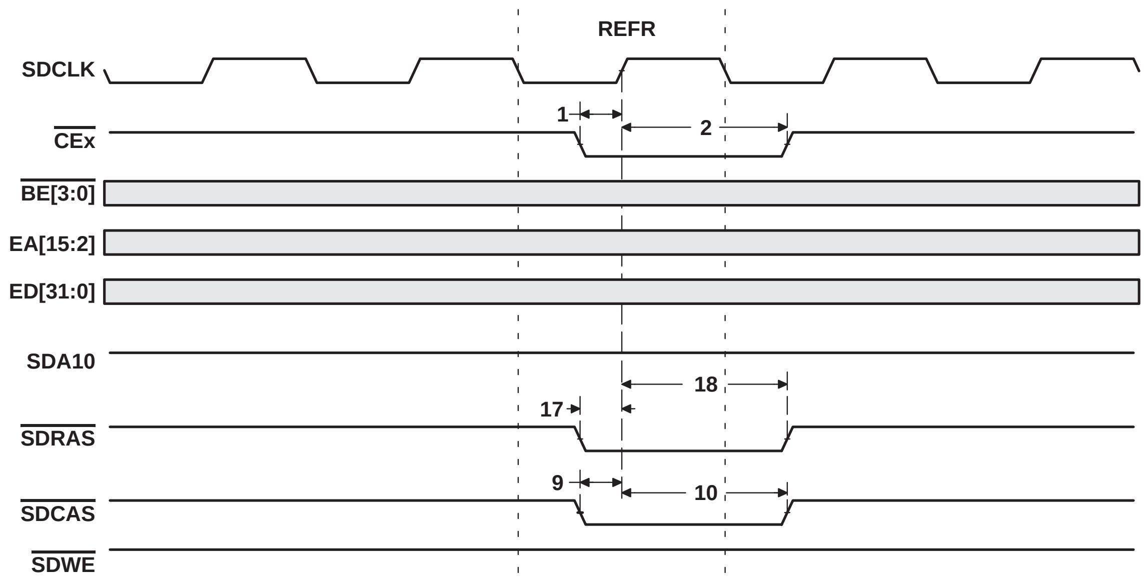


Figure 23. SDRAM REFR Command

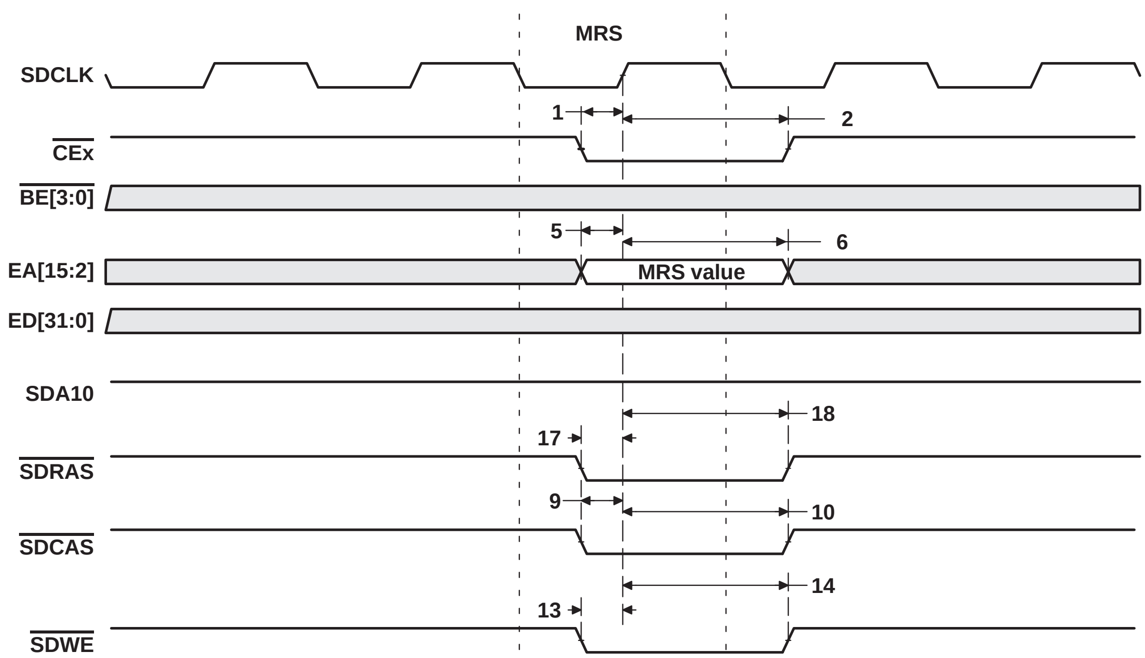


Figure 24. SDRAM MRS Command

HOLD/HOLDA TIMING

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

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(HOLDH-CKO1H)}$	Setup time, HOLD high before CLKOUT1 high	5		1		ns
2	$t_{h(CKO1H-HOLDL)}$	Hold time, HOLD low after CLKOUT1 high	2		4		ns

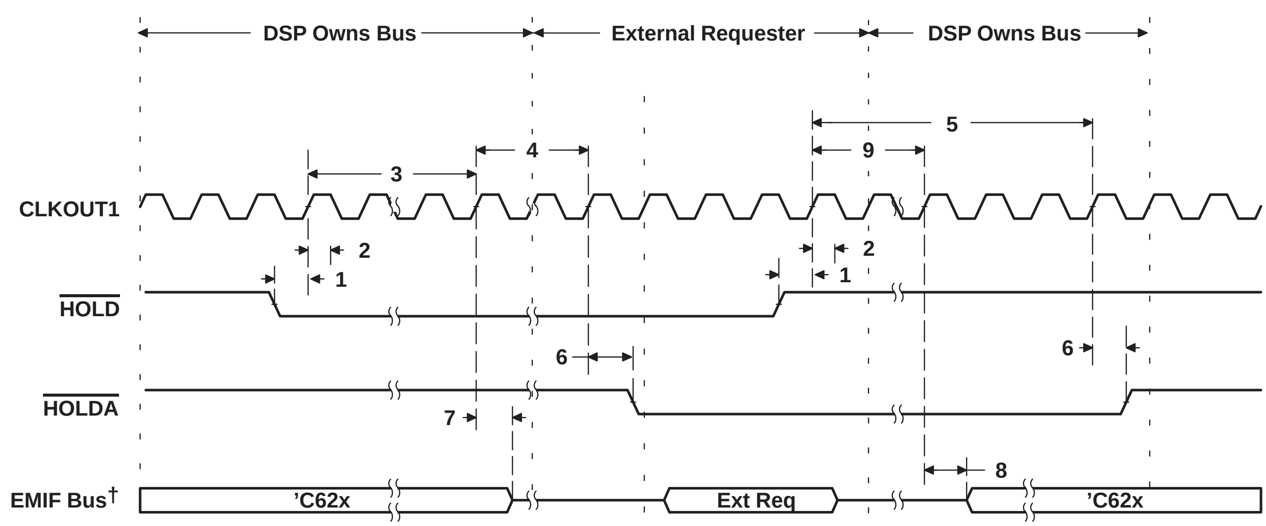
† HOLD is synchronized internally. Therefore, if setup and hold times are not met, it will either be recognized in the current cycle or in the next cycle. Thus, HOLD can be an asynchronous input.

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

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
3	$t_{R(HOLDL-BHZ)}$	Response time, HOLD low to EMIF Bus high impedance	4	‡	4	‡	CLKOUT1 cycles
4	$t_{R(BHZ-HOLDAL)}$	Response time, EMIF Bus high impedance to HOLDA low	1	2	1	2	CLKOUT1 cycles
5	$t_{R(HOLDH-HOLDAH)}$	Response time, HOLD high to HOLDA high	4	6	4	7	CLKOUT1 cycles
6	$t_{d(CKO1H-HOLDAL)}$	Delay time, CLKOUT1 high to HOLDA valid	–1	5	1	8	ns
7	$t_{d(CKO1H-BHZ)}$	Delay time, CLKOUT1 high to EMIF Bus high impedance§	–1	5	3	11	ns
8	$t_{d(CKO1H-BLZ)}$	Delay time, CLKOUT1 high to EMIF Bus low impedance§	–1	5	3	11	ns
9	$t_{R(HOLDH-BLZ)}$	Response time, HOLD high to EMIF Bus low impedance	3	5	3	6	CLKOUT1 cycles

‡ All pending EMIF transactions are allowed to complete before 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 CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE, AOE, AWE, SSADS, SSOE, SSWE, SDA10, SDRAS, SDCAS, and SDWE.



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

Figure 25. HOLD/HOLDA Timing

RESET TIMING

timing requirements for reset (see Figure 26)

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{w(RST)}$	Width of the RESET pulse (PLL stable)	10		10		CLKOUT1 cycles
		Width of the RESET pulse (PLL needs to sync up)†	250		250		µs

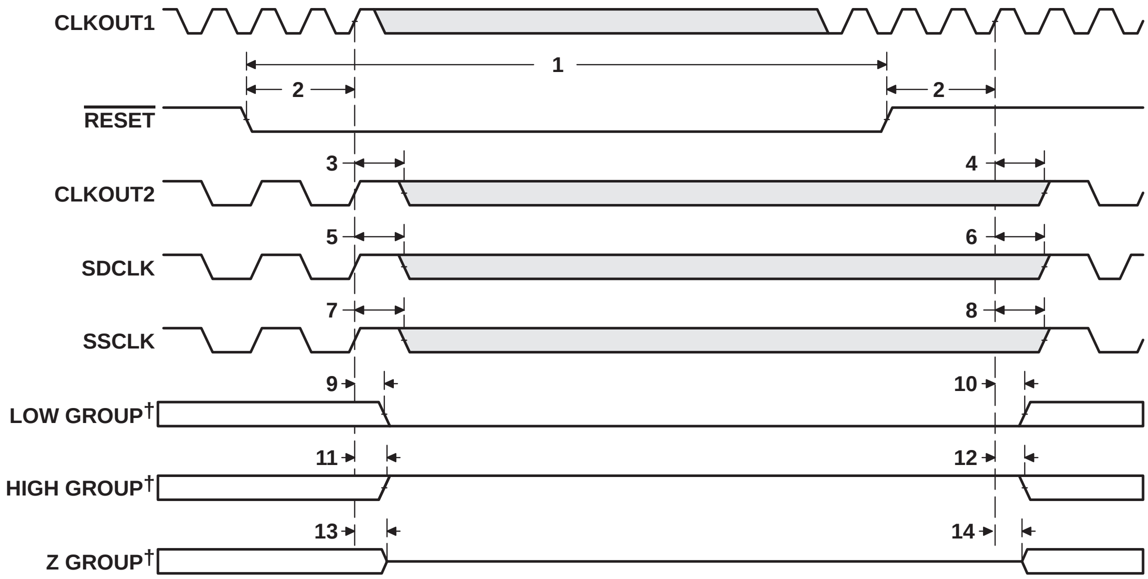
† The RESET signal is not connected internally to the clock PLL circuit. The PLL, however, may need up to 250 µs to stabilize following device powerup or after PLL configuration has been changed. During that time, RESET must be asserted to ensure proper device operation. See the *clock PLL* section for PLL lock times.

switching characteristics during reset‡ (see Figure 26)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
2	$t_{R(RST)}$	Response time to change of value in RESET signal	2		2		CLKOUT1 cycles
3	$t_{d(CKO1H-CKO2IV)}$	Delay time, CLKOUT1 high to CLKOUT2 invalid	–1	10	–1	10	ns
4	$t_{d(CKO1H-CKO2V)}$	Delay time, CLKOUT1 high to CLKOUT2 valid	–1	10	–1	10	ns
5	$t_{d(CKO1H-SDCLKIV)}$	Delay time, CLKOUT1 high to SDCLK invalid	–1	10	–1	10	ns
6	$t_{d(CKO1H-SDCLKV)}$	Delay time, CLKOUT1 high to SDCLK valid	–1	10	–1	10	ns
7	$t_{d(CKO1H-SSCKIV)}$	Delay time, CLKOUT1 high to SSCLK invalid	–1	10	–1	10	ns
8	$t_{d(CKO1H-SSCKV)}$	Delay time, CLKOUT1 high to SSCLK valid	–1	10	–1	10	ns
9	$t_{d(CKO1H-LOWIV)}$	Delay time, CLKOUT1 high to low group invalid	–1	10	–1	10	ns
10	$t_{d(CKO1H-LOWV)}$	Delay time, CLKOUT1 high to low group valid	–1		–1		ns
11	$t_{d(CKO1H-HIGHIV)}$	Delay time, CLKOUT1 high to high group invalid	–1	10	–1	10	ns
12	$t_{d(CKO1H-HIGHV)}$	Delay time, CLKOUT1 high to high group valid	–1		–1		ns
13	$t_{d(CKO1H-ZHZ)}$	Delay time, CLKOUT1 high to Z group high impedance	–1	10	–1	10	ns
14	$t_{d(CKO1H-ZV)}$	Delay time, CLKOUT1 high to Z group valid	–1		–1		ns

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

RESET TIMING (CONTINUED)



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

Figure 26. Reset Timing

EXTERNAL INTERRUPT/RESET TIMING

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

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
3	$t_{w(ILOW)}$	Width of the interrupt pulse low	2		2		CLKOUT1 cycles
4	$t_{w(IHIGH)}$	Width of the interrupt pulse high	2		2		CLKOUT1 cycles

† 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.

switching characteristics during interrupt response cycles (see Figure 27)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{R(EINTH-IACKH)}$	Response time, EXT_INTx high to IACK high	9‡		9‡		CLKOUT1 cycles
2	$t_{R(ISFP)}$	Response time, interrupt service fetch packet execution after EXT_INTx high	11‡		11‡		CLKOUT1 cycles
5	$t_{d(CKO2L-IACKV)}$	Delay time, CLKOUT2 low to IACK valid	0	10	0	10	ns
6	$t_{d(CKO2L-INUMV)}$	Delay time, CLKOUT2 low to INUMx valid	0	10	0	10	ns
7	$t_{d(CKO2L-INUMIV)}$	Delay time, CLKOUT2 low to INUMx invalid	0	10	0	10	ns

‡ Add two CLKOUT1 cycles to this parameter if the interrupt is recognized during the high half of CLKOUT2

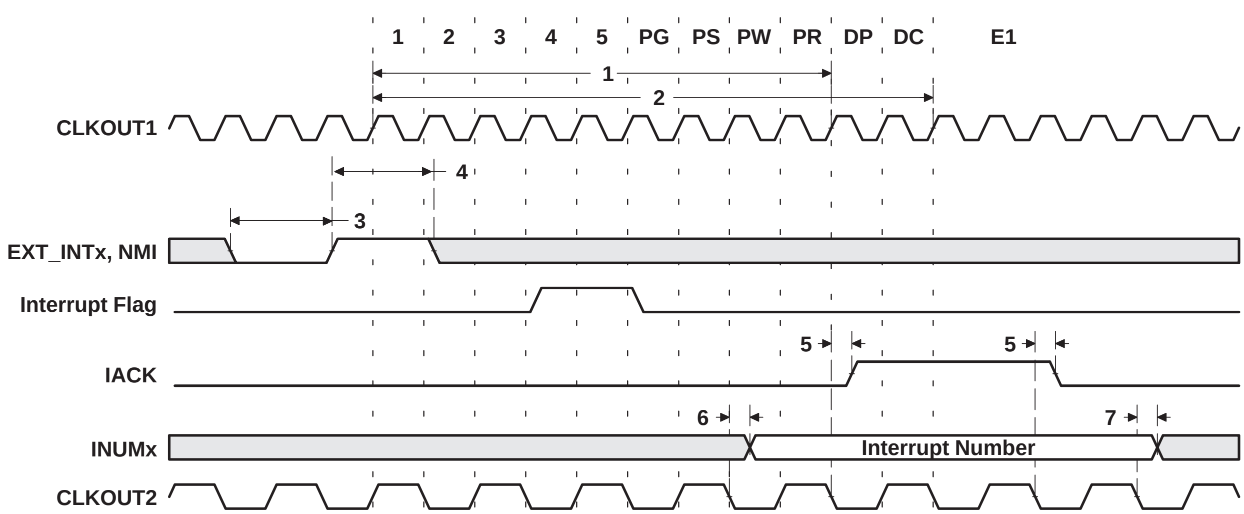


Figure 27. Interrupt Timing

HOST-PORT INTERFACE TIMING

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

NO.			'C6201-167 'C6201-200		'C6201B-167 'C6201B-200 'C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(SEL-HSTBL)}$	Setup time, select signals‡ valid before HSTROBE low	1		1		ns
2	$t_{h(HSTBL-SEL)}$	Hold time, select signals‡ valid after HSTROBE low	2		2		ns
3	$t_{w(HSTBL)}$	Pulse duration, HSTROBE low	2		2		CLKOUT1 cycles
4	$t_{w(HSTBH)}$	Pulse duration, HSTROBE high between consecutive accesses	2		2		CLKOUT1 cycles
10	$t_{su(SEL-HASL)}$	Setup time, select signals‡ valid before HAS low	1		1		ns
11	$t_{h(HASL-SEL)}$	Hold time, select signals‡ valid after HAS low	2		2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before HSTROBE high	1		1		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after HSTROBE high	1		1		ns
14	$t_{h(HRDYL-HSTBL)}$	Hold time, HSTROBE low after HRDY low. HSTROBE should not be inactivated until HRDY is active (low); otherwise, HPI writes will not complete properly.	1		1		ns

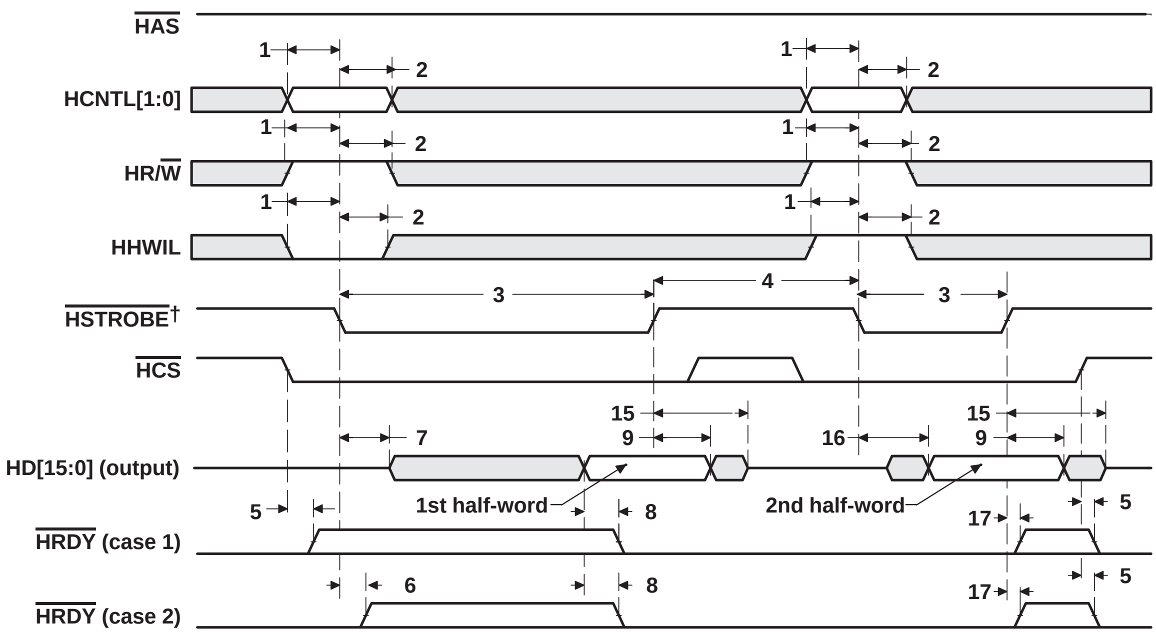
† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.
‡ Select signals include: HCNTRL[1:0], HR/W, and HHWIL.

switching characteristics during host-port interface cycles†§ (see Figure 28, Figure 29, Figure 30, and Figure 31)

NO.	PARAMETER		'C6201-167 'C6201-200		'C6201B-167 'C6201B-200 'C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
5	$t_{d(HCS-HRDY)}$	Delay time, HCS to HRDY¶	1	7	1	7	ns
6	$t_{d(HSTBL-HRDYH)}$	Delay time, HSTROBE low to HRDY high#	3	12	3	12	ns
7	$t_{oh(HSTBL-HDLZ)}$	Output hold time, HD low impedance after HSTROBE low for an HPI read	4		4		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to HRDY low	P – 2	P	P – 2	P	ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after HSTROBE high	3	12	3	12	ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, HSTROBE high to HD high impedance	3	12	3	12	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, HSTROBE low to HD valid	3	12	3	12	ns
17	$t_{d(HSTBH-HRDYH)}$	Delay time, HSTROBE high to HRDY high‖	3	12	3	12	ns

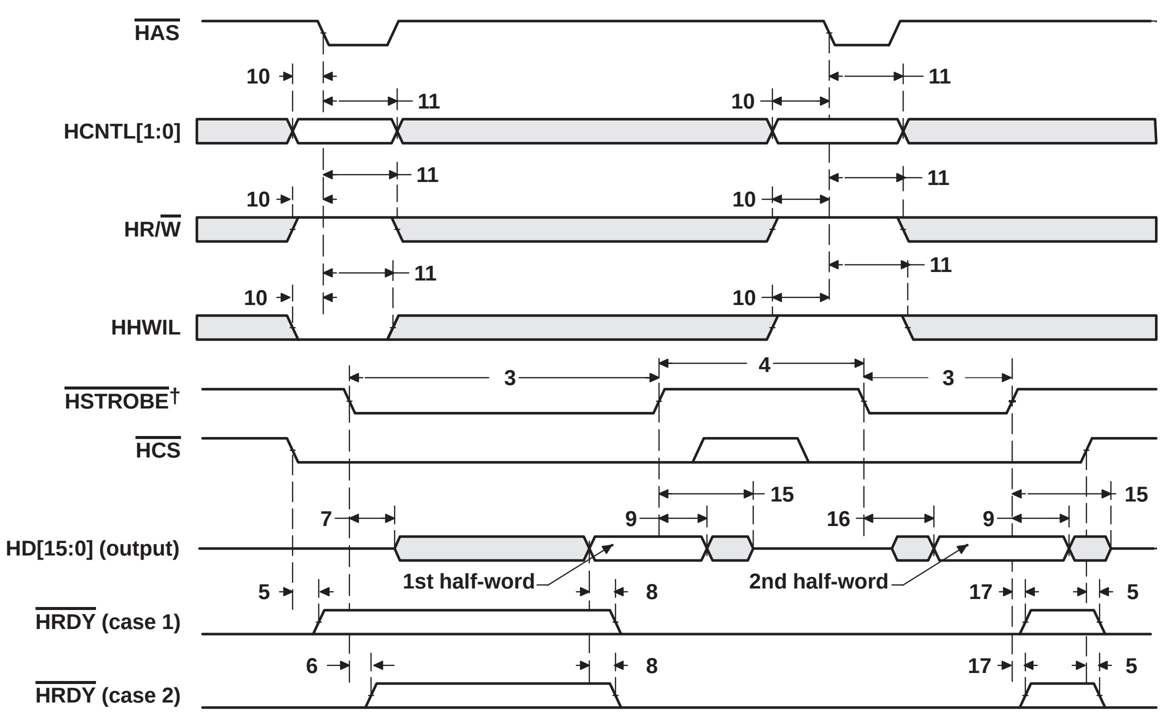
† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.
§ The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.
¶ HCS enables HRDY, and HRDY is always low when HCS is high. The case where HRDY goes high when 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 HSTROBE, the HPI sends the request to the DMA auxiliary channel, and 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. 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 HRDY signal.

HOST-PORT INTERFACE TIMING (CONTINUED)



† $\overline{\text{HSTROBE}}$ 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 $\overline{\text{HCS}}$.

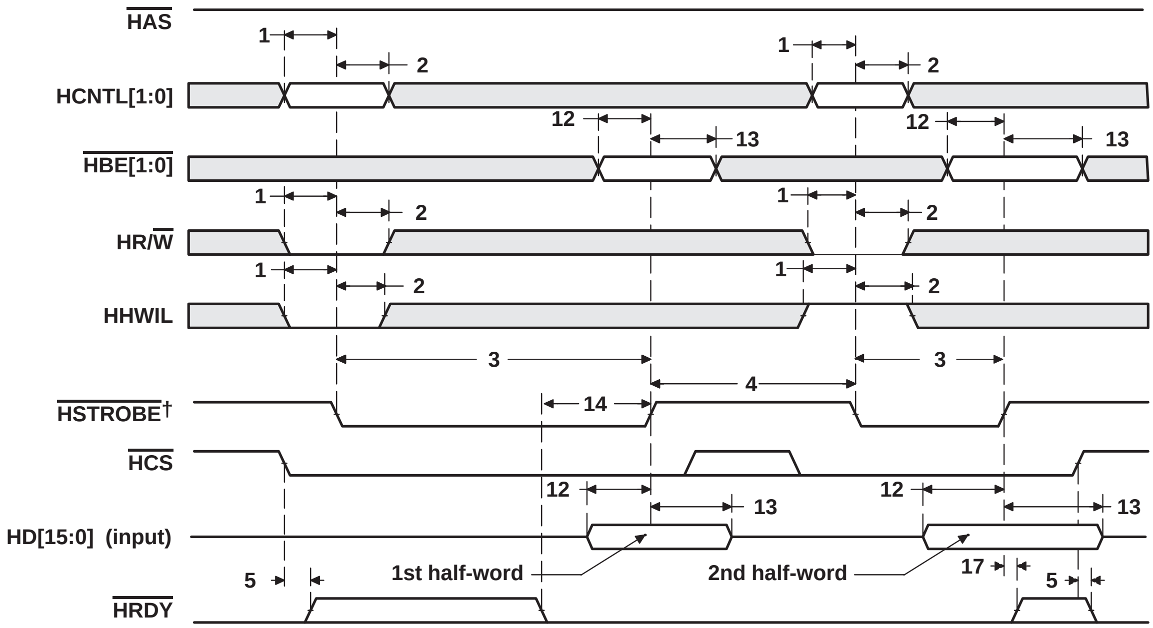
Figure 28. HPI Read Timing ($\overline{\text{HAS}}$ Not Used, Tied High)



† $\overline{\text{HSTROBE}}$ 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 $\overline{\text{HCS}}$.

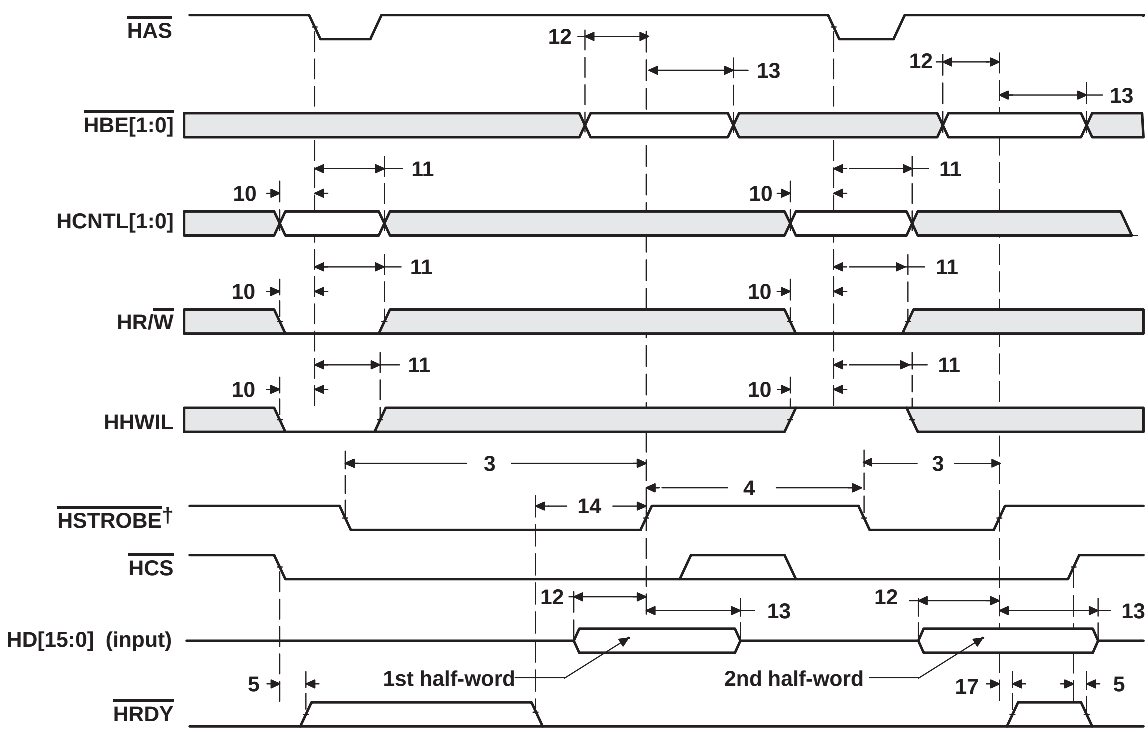
Figure 29. HPI Read Timing ($\overline{\text{HAS}}$ Used)

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.				’C6201-167 ’C6201-200 MIN	’C6201-167 ’C6201-200 MAX	’C6201B-167 ’C6201B-200 ’C6201B-233 MIN	’C6201B-167 ’C6201B-200 ’C6201B-233 MAX	UNIT
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2		2		CLKOUT1 cycles
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1		P – 1		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	13		9		ns
			CLKR ext	4		1		
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	7		6		ns
			CLKR ext	3		3		
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	10		8		ns
			CLKR ext	1		0		
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	4		3		ns
			CLKR ext	4		3		
10	$t_{su(FXH-CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	13		9		ns
			CLKX ext	4		1		
11	$t_{h(CKXL-FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	7		6		ns
			CLKX ext	3		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.
‡ The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter or SSCLK duty cycle. P = 1/CPU clock frequency in ns. For example, when running parts at 200 MHz, use P = 5 ns.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics for McBSP†‡ (see Figure 32)

NO.	PARAMETER			’C6201-167 ’C6201-200 MIN	’C6201-167 ’C6201-200 MAX	’C6201B-167 ’C6201B-200 ’C6201B-233 MIN	’C6201B-167 ’C6201B-200 ’C6201B-233 MAX	UNIT
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		4	15	4	10	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	2		2		CLKOUT1 cycles
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C – 1¶	C + 1¶	C – 1¶	C + 1¶	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	–2	4	–2	3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	0	4	–2	3	ns
			CLKX ext	3	16	3	9	
12	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	0	4	–1	4	ns
			CLKX ext	3	16	3	9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit.	CLKX int	0	4	–1	4	ns
			CLKX ext	3	16	3	9	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit. ONLY applies when in data delay 0 (XDATDLY = 00b) mode	FSX int	–2	4	–1	3	ns
			FSX ext	3	16	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.
¶ C = H or L
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

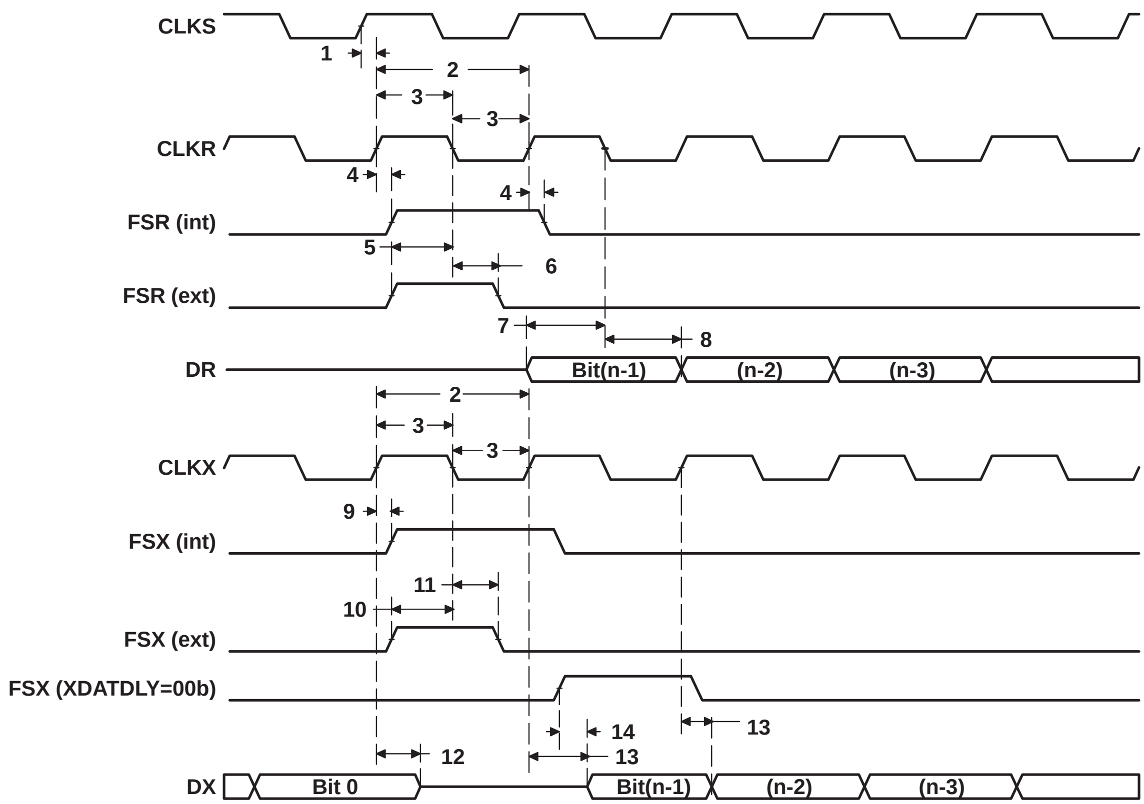


Figure 32. McBSP Timings

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{su(FRH\text{-}CKSH)}$	Setup time, FSR high before CLKS high	4		4		ns
2	$t_{h(CKSH\text{-}FRH)}$	Hold time, FSR high after CLKS high	4		4		ns

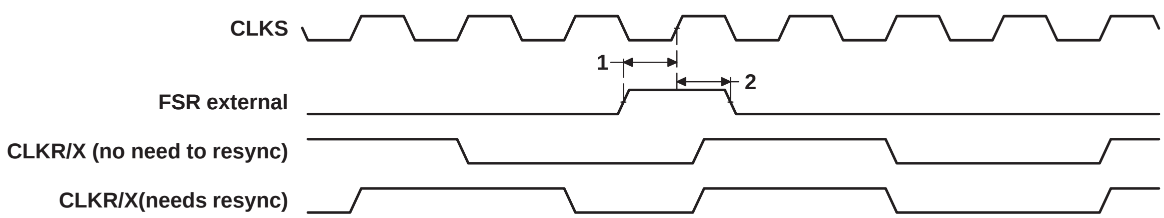


Figure 33. FSR Timing When GSYNC = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 34) (’C6201)

NO.			’C6201-167 ’C6201-200				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

† 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.
‡ 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 34) (’C6201)

NO.	PARAMETER		’C6201-167 ’C6201-200				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 This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit.	–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 + 4	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 4	4P + 17	ns

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
  = CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ 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)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 34) (’C6201B)

NO.			’C6201B-167 ’C6201B-200 ’C6201B-233				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

† 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.
‡ 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 34) (’C6201B)

NO.	PARAMETER		’C6201B-167 ’C6201B-200 ’C6201B-233				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 This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit.	–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

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
  = CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
 H = CLKX high pulse width = (CLKGDV/2 + 1) * T
 L = CLKX low pulse width = (CLKGDV/2) * T
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
 CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
 CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

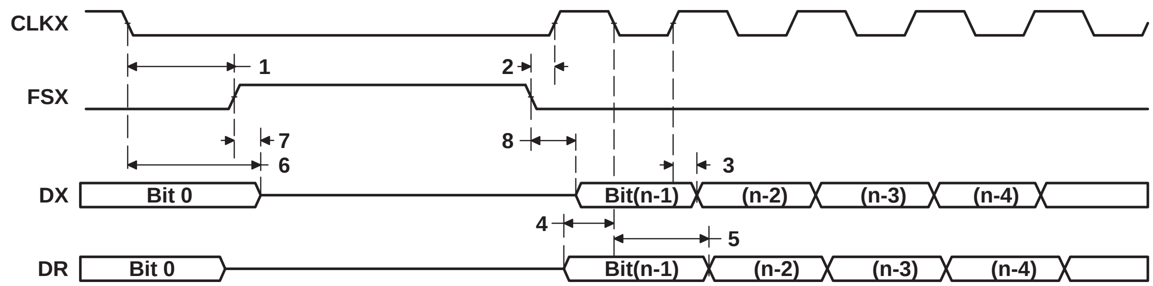


Figure 34. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 35) (’C6201)

NO.			’C6201-167 ’C6201-200				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

† 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.
‡ 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 35) (’C6201)

NO.	PARAMETER		’C6201-167 ’C6201-200				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 + 4	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H – 2	H + 4	2P + 4	4P + 17	ns

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ 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)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 35) (’C6201B)

NO.			’C6201B-167 ’C6201B-200 ’C6201B-233				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

† 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.
‡ 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 35) (’C6201B)

NO.	PARAMETER		’C6201B-167 ’C6201B-200 ’C6201B-233				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

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

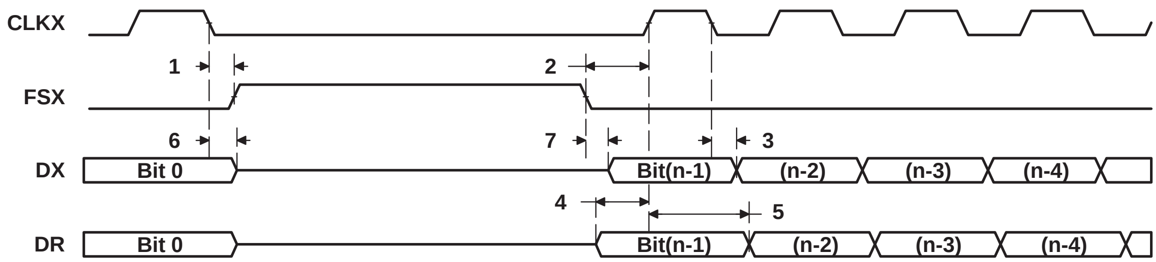


Figure 35. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 36) (’C6201)

NO.			’C6201-167 ’C6201-200				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

† 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.
‡ 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 36) (’C6201)

NO.	PARAMETER		’C6201-167 ’C6201-200				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 This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit.	–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 + 4	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 4	4P + 17	ns

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P ; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 36) ('C6201B)

NO.			'C6201B-167 'C6201B-200 'C6201B-233				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

† 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.
‡ 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 36) ('C6201B)

NO.	PARAMETER		'C6201B-167 'C6201B-200 'C6201B-233				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 This is also specified by design but not tested to be the delay time for data to be low impedance on the first data bit.	–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

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ 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)

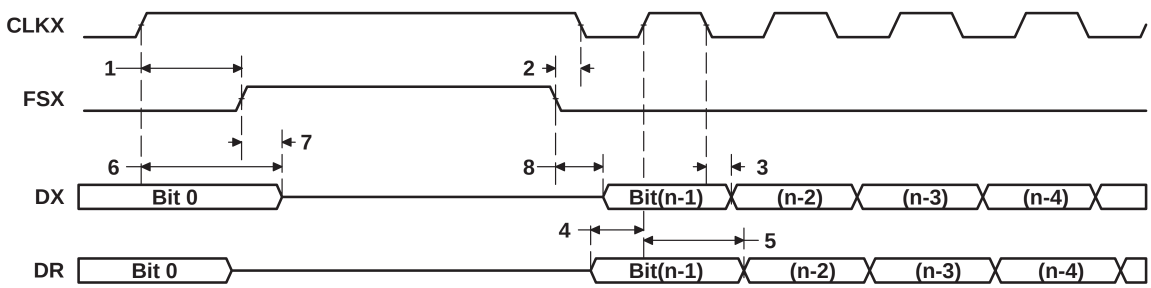


Figure 36. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 37) (’C6201)

NO.			’C6201-167 ’C6201-200				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

† 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.
‡ 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 37) (’C6201)

NO.	PARAMETER		’C6201-167 ’C6201-200				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 + 4	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L – 2	L + 4	2P + 4	4P + 17	ns

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ 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)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 37) (’C6201B)

NO.			’C6201B-167 ’C6201B-200 ’C6201B-233				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

† 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.
‡ 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 37) (’C6201B)

NO.	PARAMETER		’C6201B-167 ’C6201B-200 ’C6201B-233				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

† 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.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ T = CLKX period = (1 + CLKGDV) * P; if CLKSM = 1, then P = 1/CPU clock frequency
= CLKX period = (1 + CLKGDV) * P_clks; if CLKSM = 0, then P_clks = CLKS period.
H = CLKX high pulse width = (CLKGDV/2 + 1) * T
L = CLKX low pulse width = (CLKGDV/2) * T
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

PRODUCT PREVIEW

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

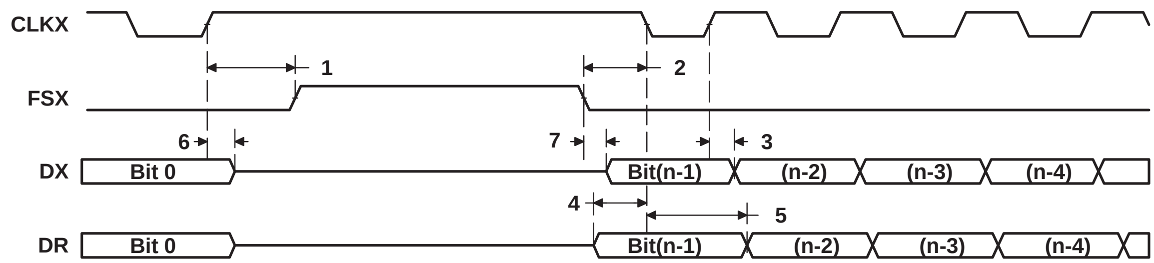


Figure 37. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

DMAC, TIMER, POWER-DOWN TIMING

switching characteristics for DMAC outputs (see Figure 38)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(CKO1H-DMACV)}$	Delay time, CLKOUT1 high to DMAC valid	2	7	2	7	ns

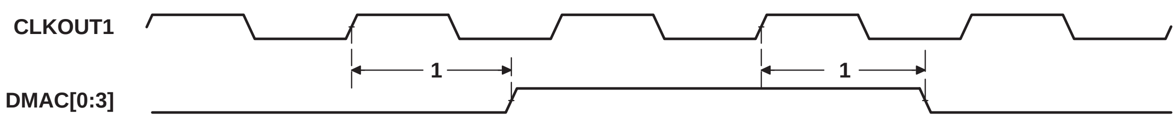


Figure 38. DMAC Timing

timing requirements for timer inputs (see Figure 39)

NO.			’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{w(TINP)}$	Pulse duration, TINP high or low	2		2		CLKOUT1 cycles

switching characteristics for timer outputs (see Figure 39)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
2	$t_{d(CKO1H-TOUTV)}$	Delay time, CLKOUT1 high to TOUT valid	3	9	3	9	ns

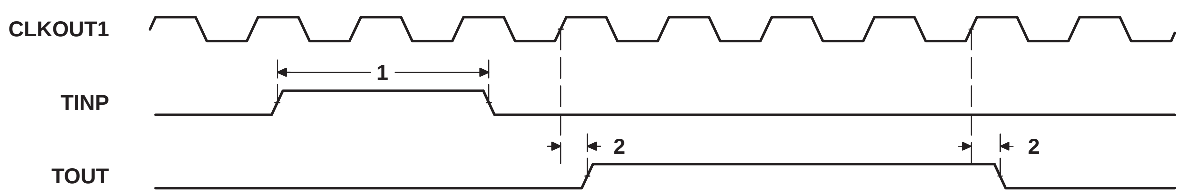


Figure 39. Timer Timing

DMAC, TIMER, POWER-DOWN TIMING (CONTINUED)

switching characteristics for power-down outputs (see Figure 40)

NO.	PARAMETER		’C6201-167 ’C6201-200		’C6201B-167 ’C6201B-200 ’C6201B-233		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(CKO1H-PDV)}$	Delay time, CLKOUT1 high to PD valid	3	5	3	5	ns

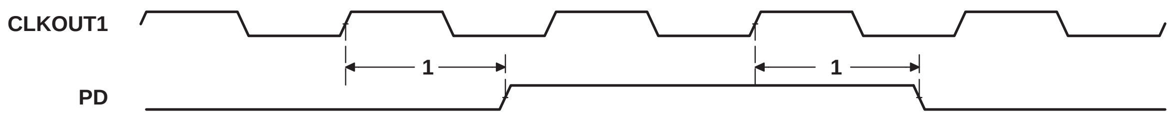
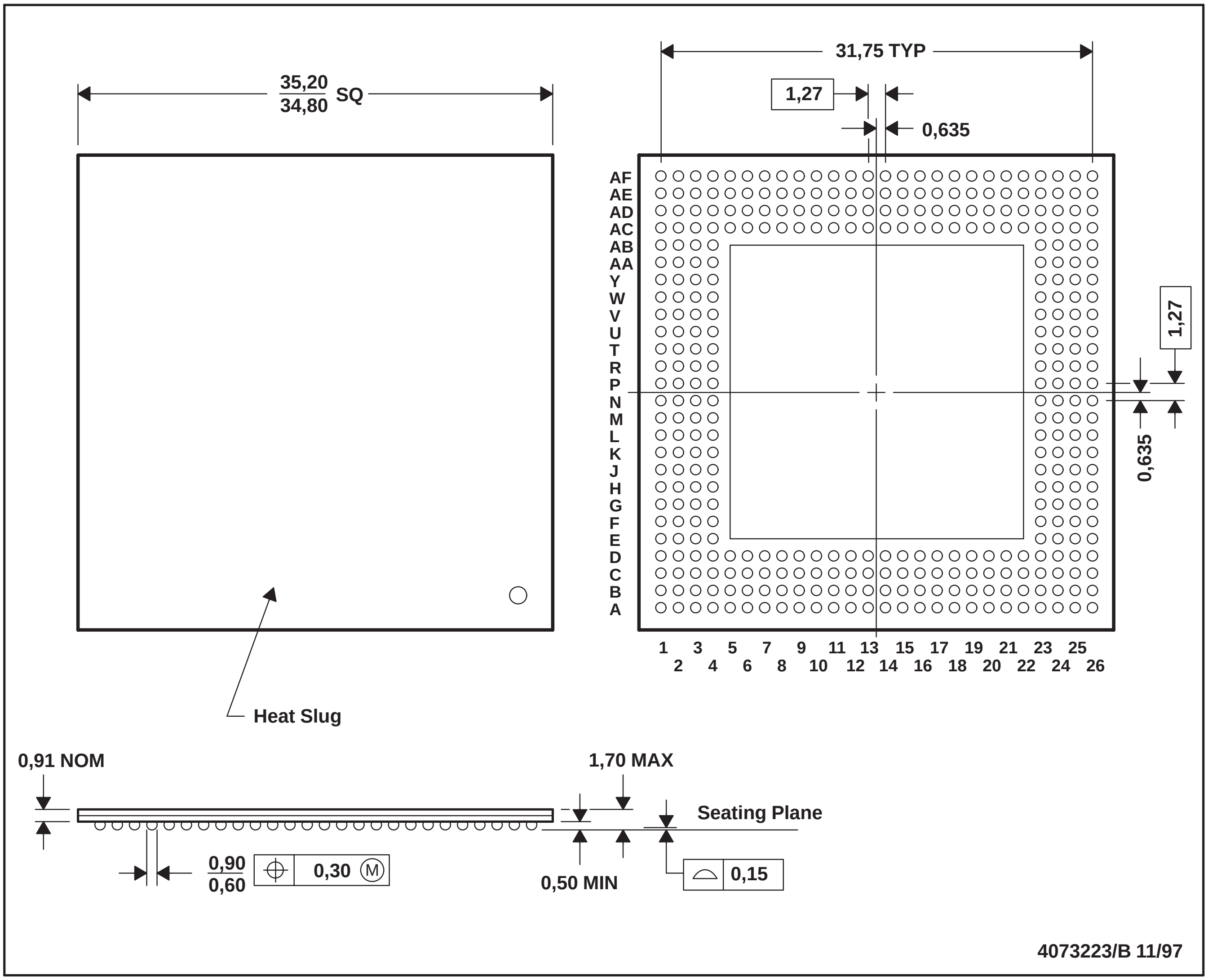


Figure 40. Power-Down Timing

MECHANICAL DATA

GGP (S-PBGA-N352) **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 metal heat slug (HSL).

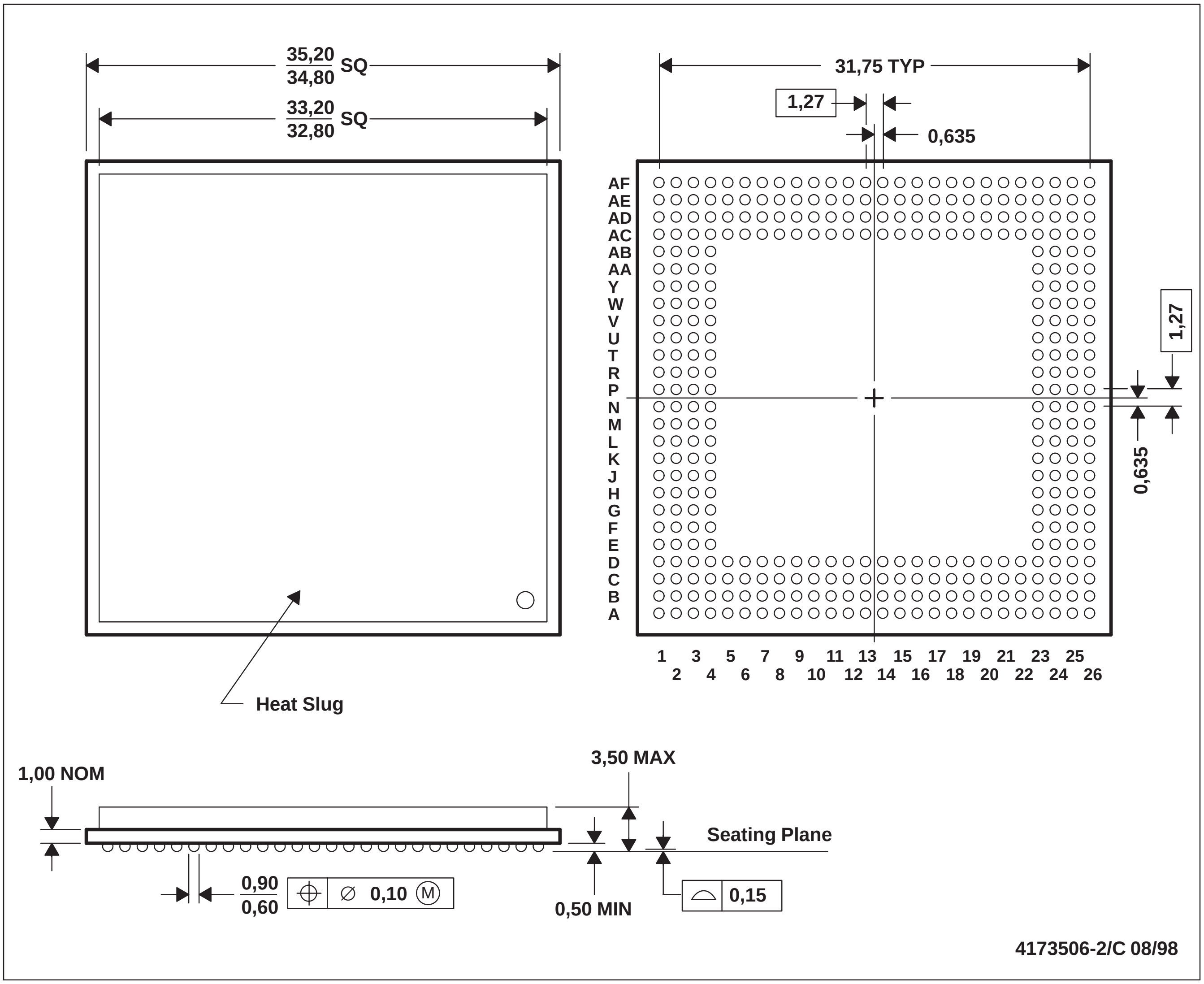
thermal resistance characteristics (S-PBGA package)

NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	0.94	N/A
2	$R\Theta_{JA}$	Junction-to-free air	11.11	0
3	$R\Theta_{JA}$	Junction-to-free air	9.61	100
4	$R\Theta_{JA}$	Junction-to-free air	8.24	250
5	$R\Theta_{JA}$	Junction-to-free air	7.10	500

† LFPM = Linear Feet Per Minute

MECHANICAL DATA

GJC (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. Falls within JEDEC MO-151/BAR-2
E. Flip chip application only.

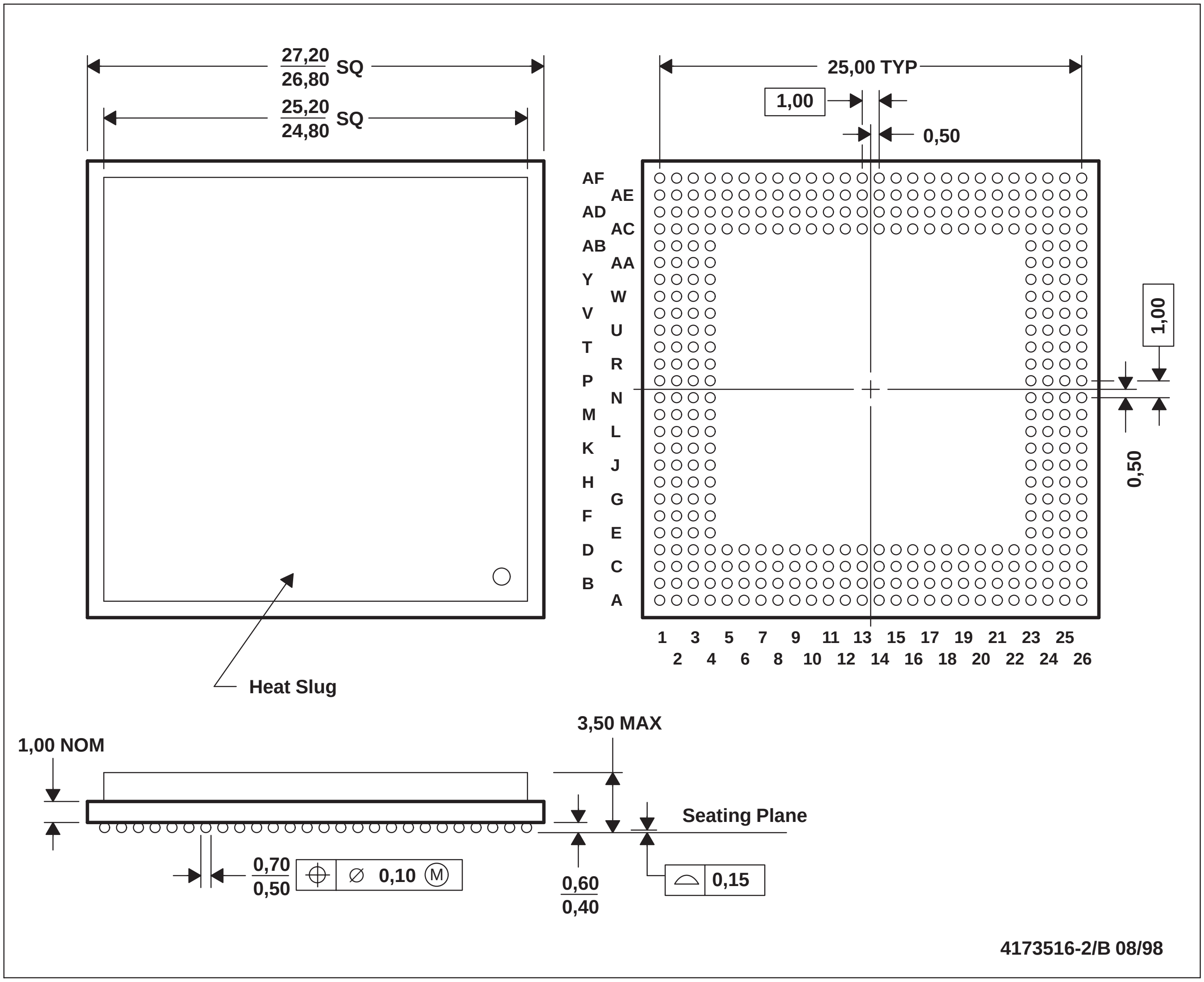
thermal resistance characteristics (S-PBGA package)

NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	0.74	N/A
2	$R\Theta_{JA}$	Junction-to-free air	11.31	0
3	$R\Theta_{JA}$	Junction-to-free air	9.60	100
4	$R\Theta_{JA}$	Junction-to-free air	8.34	250
5	$R\Theta_{JA}$	Junction-to-free air	7.30	500

† LFPM = Linear Feet Per Minute

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. Falls within JEDEC MO-151/AAL-1
E. Flip chip application only.

thermal resistance characteristics (S-PBGA package)

NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	1.0	N/A
2	$R\Theta_{JA}$	Junction-to-free air	16.0	0
3	$R\Theta_{JA}$	Junction-to-free air	13.6	100
4	$R\Theta_{JA}$	Junction-to-free air	11.8	250
5	$R\Theta_{JA}$	Junction-to-free air	10.3	500

† LFPM = Linear Feet Per Minute