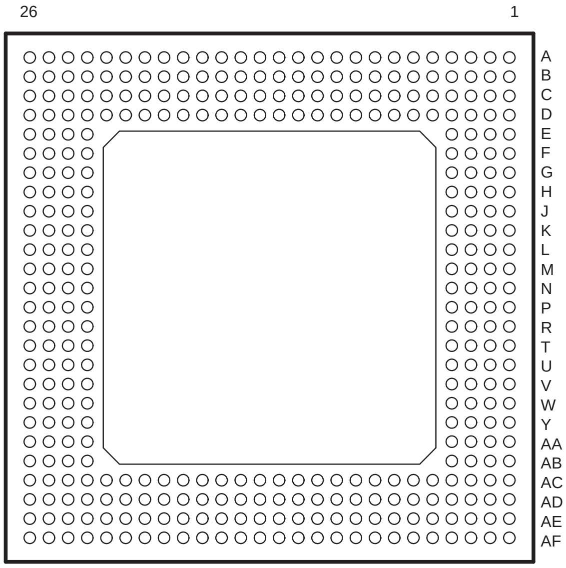
TMS320C6701
FLOATING-POINT DIGITAL SIGNAL PROCESSOR

- **Highest Performance Floating-Point Digital Signal Processor (DSP) TMS320C6701**
  - **6.7-, 6-ns Instruction Cycle Time**
  - **150-, 167-MHz Clock Rate**
  - **Eight 32-Bit Instructions/Cycle**
  - **1 GFLOPS**
  - **Pin-Compatible With ’C6201 Fixed-Point DSP**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) ’C67x CPU Core**
  - **Eight Highly Independent Functional Units:**
    - **Four ALUs (Floating- and Fixed-Point)**
    - **Two ALUs (Fixed-Point)**
    - **Two Multipliers (Floating- and Fixed-Point)**
  - **Load-Store Architecture With 32 32-Bit General-Purpose Registers**
  - **Instruction Packing Reduces Code Size**
  - **All Instructions Conditional**
- **Instruction Set Features**
  - **Hardware Support for IEEE Single-Precision Instructions**
  - **Hardware Support for IEEE Double-Precision Instructions**
  - **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)**
- **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 (352-PIN BGA) PACKAGE
(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 Ball Grid Array (BGA) Package (GJC Suffix)**
- **0.18-μm/5-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1.8-V Internal**

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.

---

description

The TMS320C67x DSPs are the floating-point DSP family in the TMS320C6000 platform. The TMS320C6701 (’C6701) device is based on the high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI™), making this DSP an excellent choice for multichannel and multifunction applications. With performance of up to 1 giga floating-point operations per second (GFLOPS) at a clock rate of 167 MHz, the ’C6701 offers cost-effective solutions to high-performance DSP programming challenges. The ’C6701 DSP possesses the operational flexibility of high-speed controllers and the numerical capability of array processors. This processor has 32 general-purpose registers of 32-bit word length and eight highly independent functional units. The eight functional units provide four floating-/fixed-point ALUs, two fixed-point ALUs, and two floating-/fixed-point multipliers. The ’C6701 can produce two multiply-accumulates (MACs) per cycle for a total of 334 million MACs per second (MMACS). The ’C6701 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The ’C6701 includes a large bank of on-chip memory and has a powerful and diverse set of peripherals. Program memory consists of a 64K-byte block that is user-configurable as cache or memory-mapped program space. Data memory consists of two 32K-byte blocks of RAM. The peripheral set includes two multichannel buffered serial ports (McBSPs), two general-purpose timers, a host-port interface (HPI), and a glueless external memory interface (EMIF) capable of interfacing to SDRAM or SBSRAM and asynchronous peripherals.

The ’C6701 has a complete set of development tools which includes: a new C compiler, an assembly optimizer to simplify programming and scheduling, and a Windows™ debugger interface for visibility into source code execution.

device characteristics

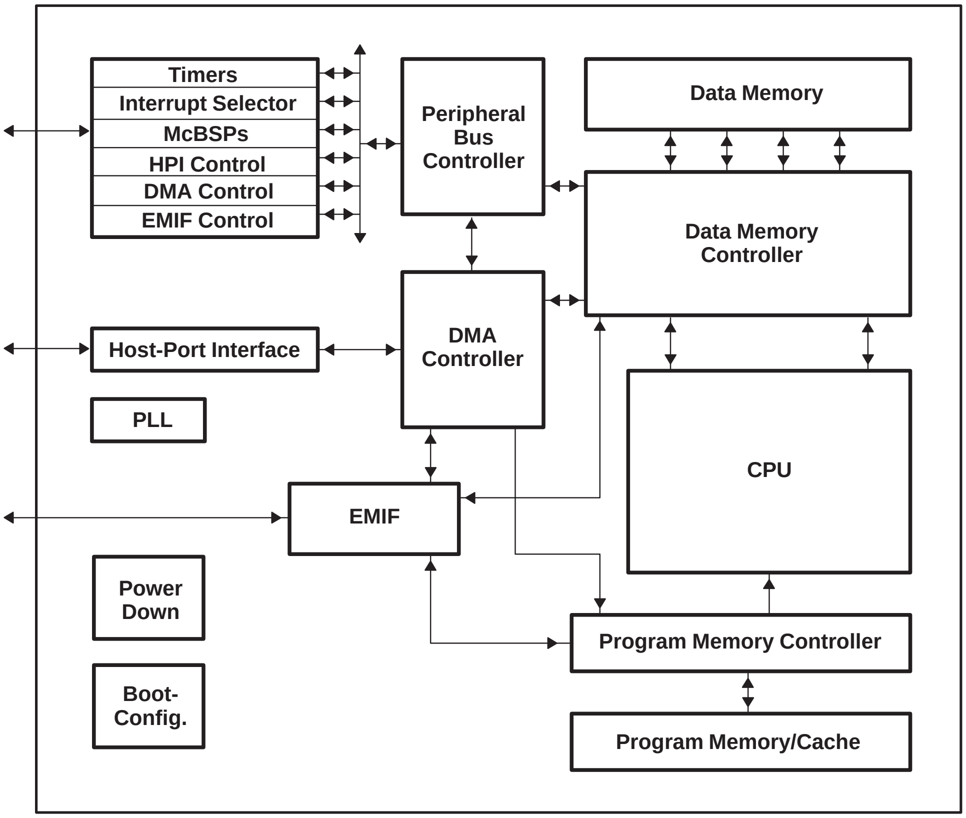
Table 1 provides an overview of the ’C6701 DSP. The table shows significant features of each device, including the capacity of on-chip RAM, the peripherals, the execution time, and the package type with pin count.

Table 1. Characteristics of the ’C6701 Processors

CHARACTERISTICS	DESCRIPTION
Device Number	TMS320C6701
On-Chip Memory	512-Kbit Program Memory 512-Kbit Data Memory (organized as 2 blocks)
Peripherals	2 Mutichannel Buffered Serial Ports (McBSP) 2 General-Purpose Timers Host-Port Interface (HPI) External Memory Interface (EMIF)
Cycle Time	6 ns (TMS320C6701-167), 6.7 ns (TMS320C6701-150)
Package Type	35 mm × 35 mm, 352-Pin BGA (GJC)
Nominal Voltage	1.8 V Core 3.3 V I/O

TI is a trademark of Texas Instruments Incorporated.
Windows is a registered trademark of the Microsoft Corporation.

functional block diagram



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 ’C67x CPU from other VLIW architectures.

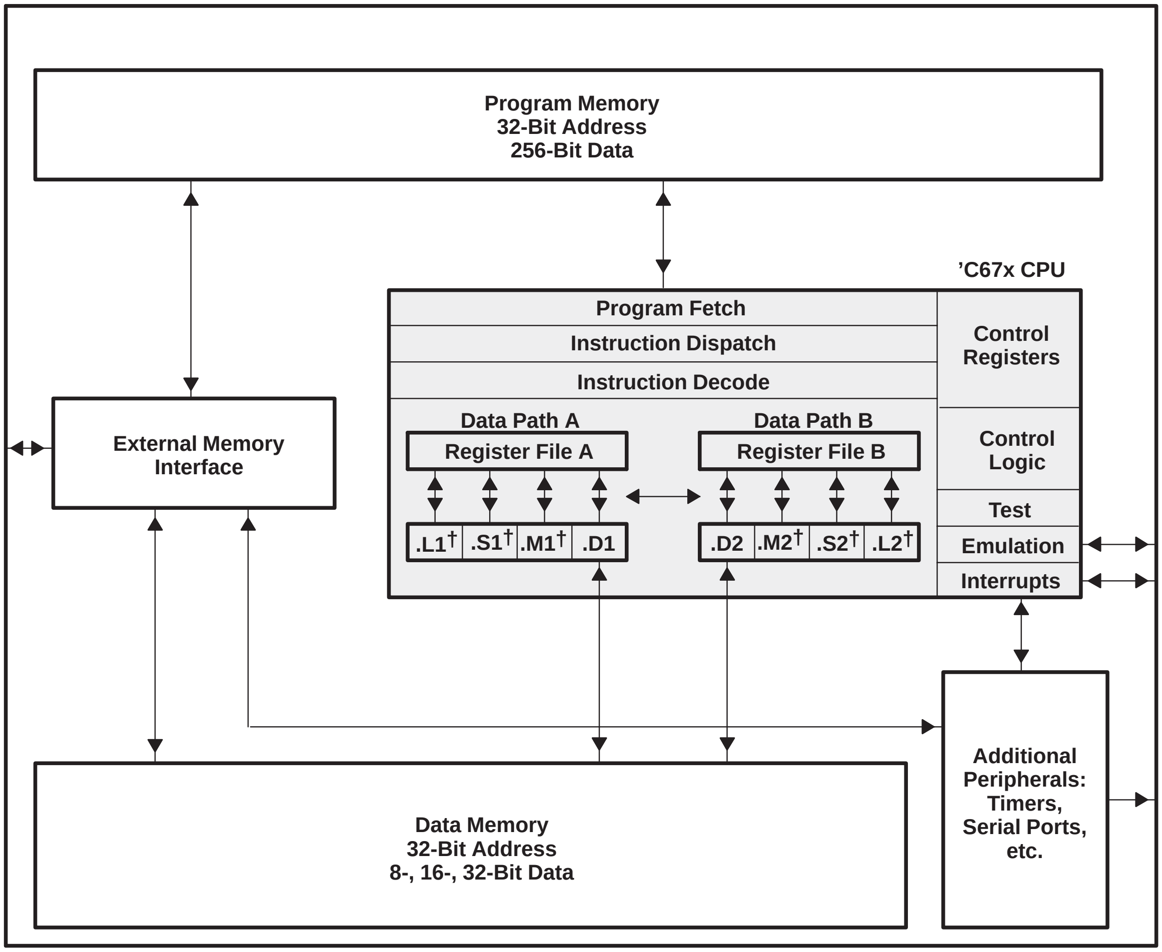
The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files contain 16 32-bit registers each for the total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU (see Figure 1 and Figure 2). 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 registers on the other side, by which the two sets of functional units can access data from the register files on opposite sides. 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.

The ’C67x CPU executes all ’C62x instructions. In addition to ’C62x fixed-point instructions, the six out of eight functional units (.L1, .M1, .D1, .D2, .M2, and .L2) also execute floating-point instructions. The remaining two functional units (.S1 and .S2) also execute the new LDDW instruction which loads 64 bits per CPU side for a total of 128 bits per cycle.

Another key feature of the ’C67x 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 ’C67x CPU supports a variety of indirect-addressing modes using either linear- or circular-addressing modes with 5- or 15-bit offsets. All instructions are conditional, and most can access any one of the 32 registers. Some registers, however, are singled out to support specific addressing or to hold the condition for conditional instructions (if the condition is not automatically “true”). The two .M functional units are dedicated for multiplies. The two .S and .L functional units perform a general set of arithmetic, logical, and branch functions with results available every clock cycle.

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

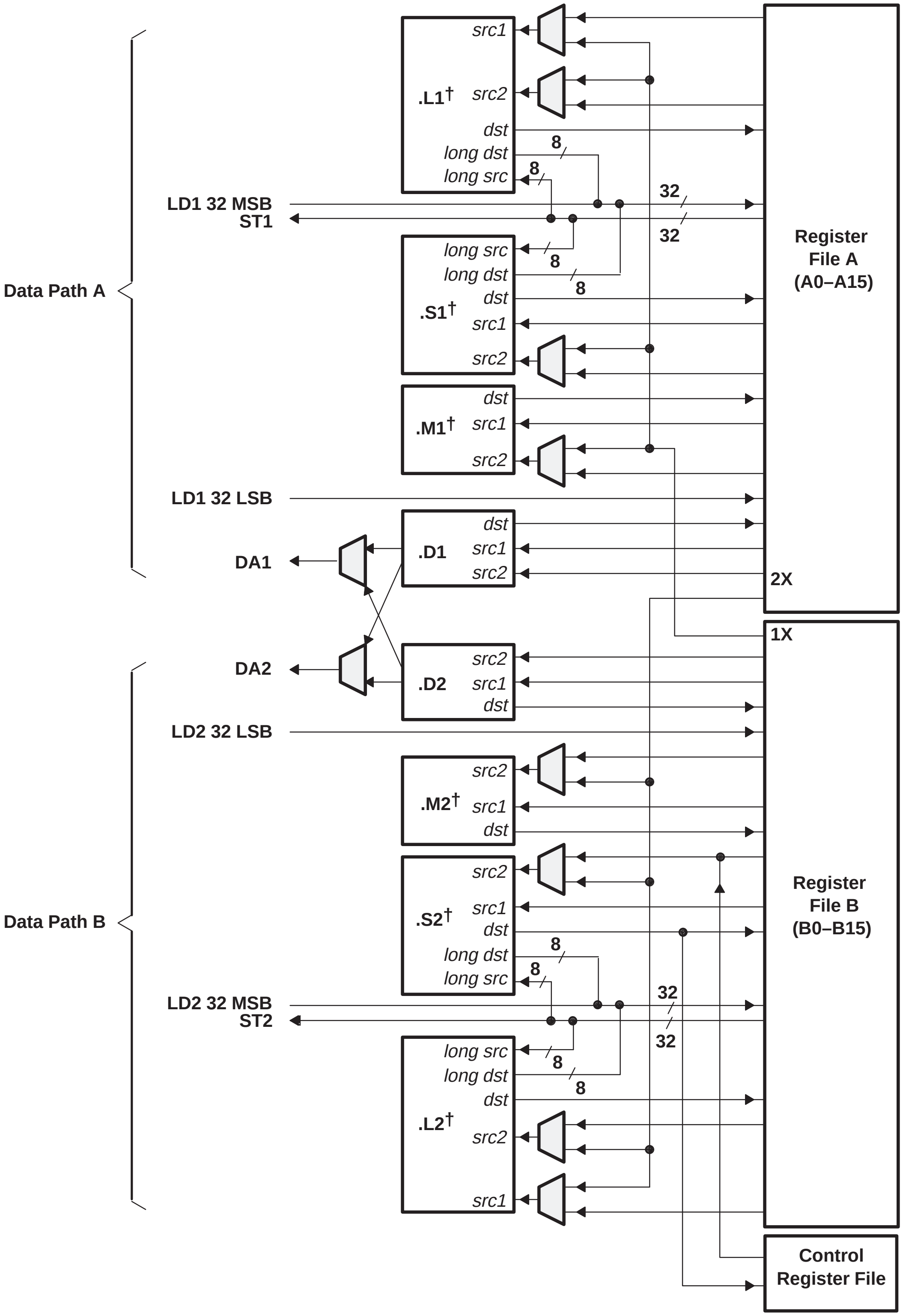
CPU description (continued)



† These functional units execute floating-point instructions.

Figure 1. CPU Block Diagram

CPU description (continued)



† These functional units execute floating-point instructions.

Figure 2. TMS320C67x CPU Data Paths

signal groups description

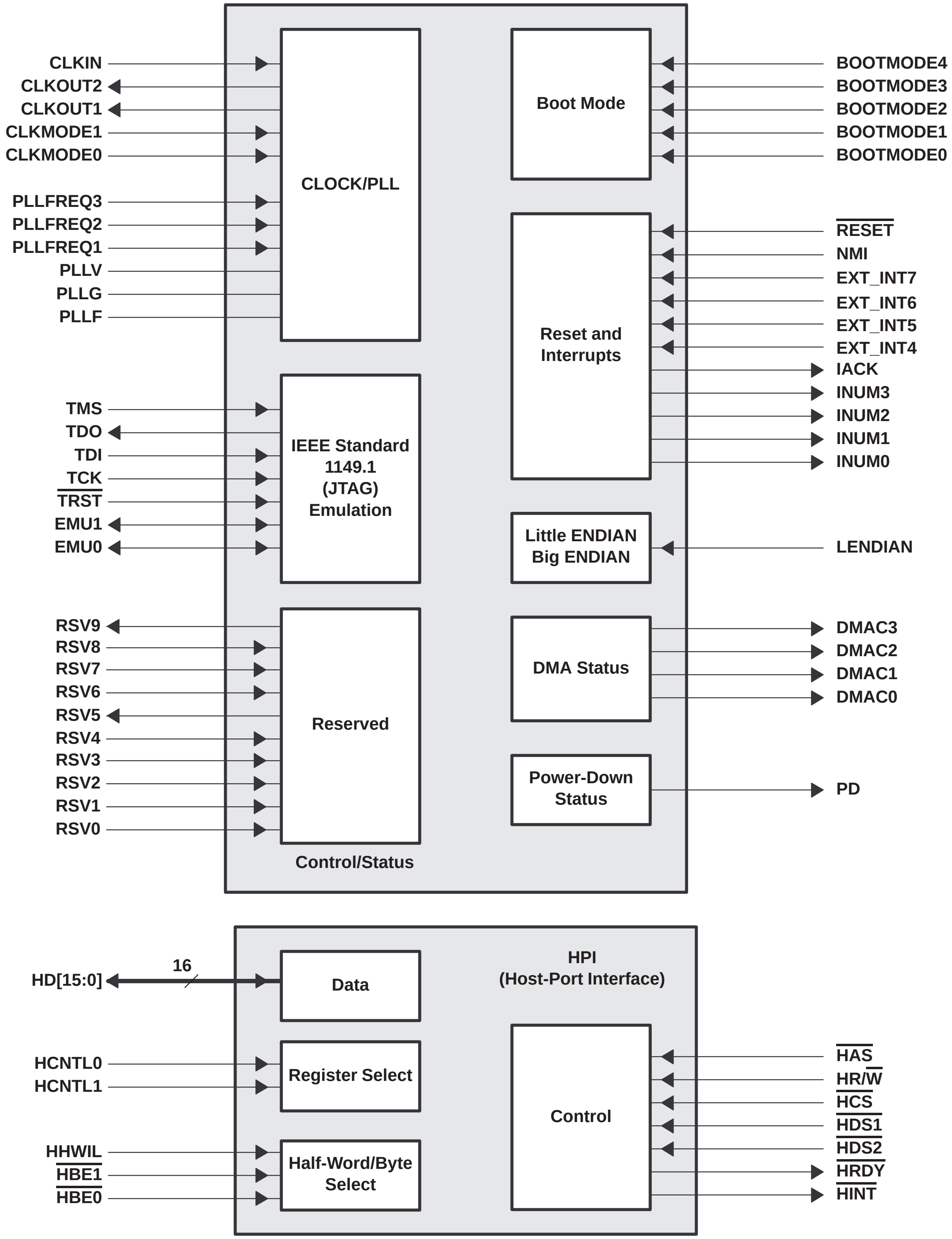


Figure 3. CPU and Peripheral Signals

signal groups description (continued)

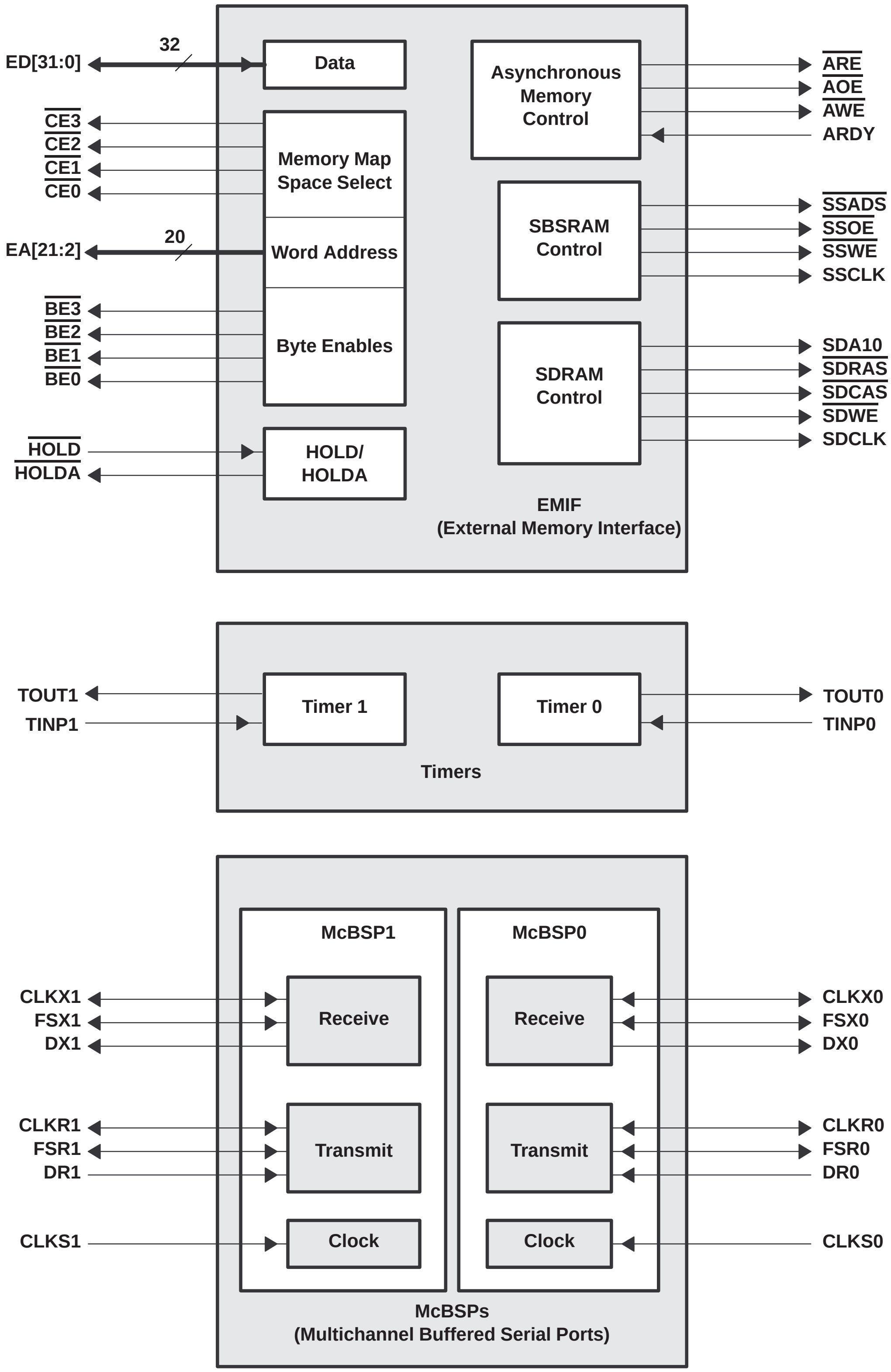


Figure 4. Peripheral Signals

Signal Descriptions

SIGNAL		TYPE[†]	DESCRIPTION
NAME	NO.		
CLOCK/PLL			
CLKIN	C10	I	Clock Input
CLKOUT1	AF22	O	Clock output at full device speed
CLKOUT2	AF20	O	Clock output at half of device speed
CLKMODE1	C6	I	Clock mode select • Selects whether the output clock frequency = input clock frequency x4 or x1
CLKMODE0	C5		
PLLFREQ3	A9	I	PLL frequency range (3, 2, and 1) • The target range for CLKOUT1 frequency is determined by the 3-bit value of the PLLFREQ pins.
PLLFREQ2	D11		
PLLFREQ1	B10		
PLLV[‡]	D12	A[§]	PLL analog V_{CC} connection for the low-pass filter
PLLG[‡]	C12	A[§]	PLL analog GND connection for the low-pass filter
PLLF	A11	A[§]	PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION			
TMS	L3	I	JTAG test-port mode select (features an internal pullup)
TDO	W2	O/Z	JTAG test-port data out
TDI	R4	I	JTAG test-port data in (features an internal pullup)
TCK	R3	I	JTAG test-port clock
$\overline{\text{TRST}}$	T1	I	JTAG test-port reset (features an internal pulldown)
EMU1	Y1	I/O/Z	Emulation pin 1, pullup with a dedicated 20-kΩ resistor[¶]
EMU0	W3	I/O/Z	Emulation pin 0, pullup with a dedicated 20-kΩ resistor[¶]
CONTROL			
$\overline{\text{RESET}}$	K2	I	Device reset
NMI	L2	I	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	U3	I	External interrupts • Edge-driven (rising edge)
EXT_INT6	V2		
EXT_INT5	W1		
EXT_INT4	U4		
IACK	Y2	O	Interrupt acknowledge for all active interrupts serviced by the CPU
INUM3	AA1	O	Active interrupt identification number • Valid during IACK for all active interrupts (not just external) • Encoding order follows the interrupt-service fetch-packet ordering
INUM2	W4		
INUM1	AA2		
INUM0	AB1		
LENDIAN	H3	I	If high, LENDIAN selects little-endian byte/half-word addressing order within a word If low, LENDIAN selects big-endian addressing
PD	D3	O	Power-down mode 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 documentation for information on how to connect these pins.

[§] A = Analog Signal (PLL Filter)

[¶] For emulation and normal operation, pull up EMU1 and EMU0 with a dedicated 20-kΩ resistor. For boundary scan, pull down EMU1 and EMU0 with a dedicated 20-kΩ resistor.

Signal Descriptions (Continued)

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

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

Signal Descriptions (Continued)

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

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

Signal Descriptions (Continued)

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

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

Signal Descriptions (Continued)

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

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

Signal Descriptions (Continued)

SIGNAL		TYPE[†]	DESCRIPTION
NAME	NO.		
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)			
CLKS0	L4	I	External clock source (as opposed to internal)
CLKR0	M2	I/O/Z	Receive clock
CLKX0	L1	I/O/Z	Transmit clock
DR0	J1	I	Receive data
DX0	R1	O/Z	Transmit data
FSR0	P4	I/O/Z	Receive frame sync
FSX0	P3	I/O/Z	Transmit frame sync
RESERVED FOR TEST			
RSV0	T2	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV1	G2	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV2	C11	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV3	B9	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV4	A6	I	Reserved for testing, ***pulldown*** with a dedicated 20-kΩ resistor
RSV5	C8	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV6	C21	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV7	B22	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV8	A23	I	Reserved for testing, pullup with a dedicated 20-kΩ resistor
RSV9	E4	O	Reserved (leave unconnected, ***do not*** connect to power or ground)
SUPPLY VOLTAGE PINS			
DV_{DD}	A10	S	3.3-V supply voltage
	A15		
	A18		
	A21		
	A22		
	B7		
	C1		
	D17		
	F3		
	G24		
	G25		
	H25		
	J25		
	L25		
	M3		
	N3		
	N23		
	R26		
	T24		
	U24		
	W24		

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

Signal Descriptions (Continued)

SIGNAL NAME	NO.	TYPE[†]	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
DV_{DD}	Y4 AB3 AB4 AB26 AC6 AC10 AC19 AC21 AC22 AC25 AD11 AD13 AD15 AD18 AE18 AE21 AF5 AF6 AF17	S	3.3-V supply voltage
CV_{DD}	A5 A12 A16 A20 B2 B6 B11 B12 B25 C3 C15 C20 C24 D4 D6 D7 D9 D14 D18 D20	S	1.8-V supply voltage

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

Signal Descriptions (Continued)

SIGNAL NAME	NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD}	D23 E1 F1 H4 J4 J23 K1 K23 M1 M24 N4 N25 P2 P23 T3 T4 U1 V4 V23 AC4 AC9 AC12 AC13 AC18 AC23 AD3 AD8 AD14 AD24 AE2 AE8 AE12 AE25 AF12	S	1.8-V supply voltage

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

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE[†]	DESCRIPTION
GROUND PINS			
V_{SS}	A1	GND	Ground pins
	A2		
	A4		
	A13		
	A14		
	A25		
	A26		
	B1		
	B3		
	B5		
	B24		
	B26		
	C2		
	C7		
	C13		
	C16		
	C17		
	C25		
	D13		
	D19		
	E3		
	E24		
	F2		
	F24		
	G3		
	G4		
	G26		
	J3		
	L23		
	L26		
	M23		
	N1		
	N2		
	N24		
	N26		
	P1		
	P26		
	R24		
	T25		

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

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE[†]	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	U2 U23 V1 V3 Y3 Y25 AA3 AA23 AB23 AC2 AC5 AC7 AC14 AC16 AD2 AD12 AD16 AD20 AD25 AE1 AE3 AE7 AE9 AE13 AE16 AE19 AE23 AE24 AE26 AF1 AF2 AF8 AF10 AF13 AF14 AF25 AF26	GND	Ground pins

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

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE[†]	DESCRIPTION
REMAINING UNCONNECTED PINS			
NC	A8 B8 C9 D10 D21 G1 H1 H2 J2 K3 R2		Unconnected pins

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

development support

Texas Instruments (TI) offers an extensive line of development tools for the ’C6x 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 ’C6x-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 ’C6x multiprocessor system debug)
EVM (Evaluation Module)

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

See Table 2 for a complete listing of development-support tools for the ’C6x. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Table 2. TMS320C6x 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™	TMDX3246555-07
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.

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).

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, GJC), the temperature range (for example, blank is the default commercial temperature range), and the device speed range in megahertz (for example, -167 is 167 MHz). Figure 5 provides a legend for reading the complete device name for any TMS320 family member.

device and development-support tool nomenclature (continued)

TMS 320 C 6701 GJC (A) –167

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
–100 MHz
–150 MHz
–167 MHz
–200 MHz
–233 MHz
–250 MHz
–300 MHz

TEMPERATURE RANGE (DEFAULT: 0°C TO 90°C)
Blank = 0°C to 90°C, commercial temperature
A = –40°C to 105°C, extended temperature

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 = 256-pin plastic BGA
GGU = 144-pin plastic BGA
GGP = 352-pin plastic BGA
GJC = 352-pin plastic BGA
GJL = 352-pin plastic BGA
GLS = 384-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
6203
6211
6701
6711

---

† 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 5. TMS320 Device Nomenclature (Including TMS320C6701)

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

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

The *TMS320C6000 Programmer's Guide* (literature number SPRU198) describes ways to optimize C and assembly code for 'C6x devices and includes application program examples.

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.

TMS320C6000 Assembly Language Tools User's Guide (literature number SPRU186) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the 'C6000 generation of devices.

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.

TMS320C6000 DSP/BIOS User's Guide (literature number SPRU303) describes how to use DSP/BIOS tools and APIs to analyze embedded real-time DSP applications.

Code Composer User's Guide (literature number SPRU296) explains how to use the Code Composer development environment to build and debug embedded real-time DSP applications.

Code Composer Studio Tutorial (literature number SPRU301) introduces the Code Composer Studio integrated development environment and software tools.

The *TMS320C6000 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.

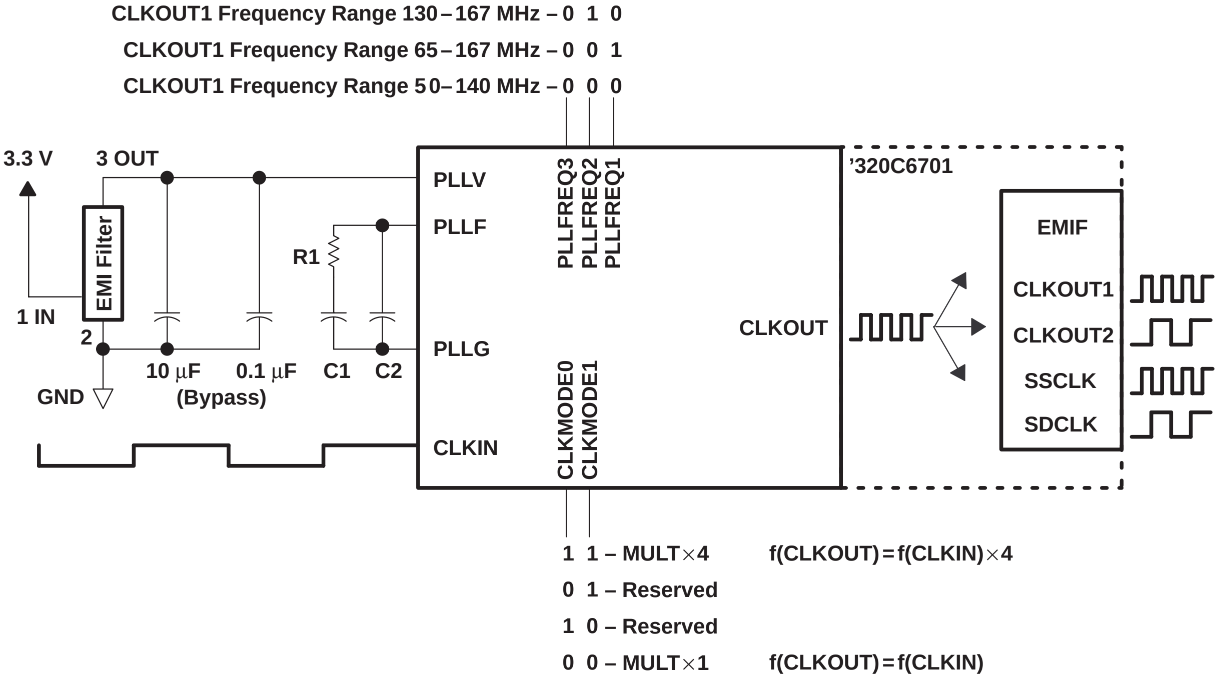
Information regarding TI DSP products is also available on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

clock PLL

All of the ’C67x 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 6 must be properly designed.

To configure the ’C67x PLL clock for proper operation, see Figure 6 and Table 3. To minimize the clock jitter, a single clean power supply should power both the ’C67x device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. See the section titled “input and output clocks” for input clock timing requirements.



NOTES:
A. For CLKMODE x4, values for C1, C2, and R1 are fixed and apply to all valid frequency ranges of CLKIN and CLKOUT.
B. For CLKMODE x1, the PLL is bypassed and all six external PLL components can be removed. For this case, the PLLV terminal has to be connected to a clean 3.3-V supply and the PLLG and PLLF terminals should be tied together.
C. Due to overlap of frequency ranges when choosing the PLLFREQ, more than one frequency range can contain the CLKOUT1 frequency. Choose the lowest frequency range that includes the desired frequency. For example, for CLKOUT1 = 133 MHz, choose PLLFREQ value of 000b. For CLKOUT1 = 167 MHz, choose PLLFREQ value of 001b. PLLFREQ values other than 000b, 001b, and 010b are reserved.
D. EMI filter manufacturer TDK part number ACF451832-153-T
E. 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 6. PLL Block Diagram

clock PLL (continued)

Table 3. ’C6701 PLL Component Selection Table

CLKMODE	CLKIN RANGE (MHz)	CPU CLOCK FREQUENCY (CLKOUT1) RANGE (MHz)	CLKOUT2 RANGE (MHz)	R1 (Ω)	C1 (nF)	C2 (pF)	TYPICAL LOCK TIME (μs)†
x4	12.5–41.7	50–167	25–83.5	60.4	27	560	75

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

power-supply sequencing

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 supply. This is to ensure that the I/O buffers have valid inputs from the core before the output buffers are powered up, thus preventing bus contention with other chips on the board.

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

Supply voltage range, CV_{DD} (see Note 1)	–0.3 V to 2.3 V
Supply voltage range, DV_{DD} (see Note 1)	–0.3 V to 4 V
Input voltage range	–0.3 V to 4 V
Output voltage range	–0.3 V to 4 V
Operating case temperature range, T_C	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

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

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

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	DV_{DD} = MIN, I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN, I_{OL} = MAX			0.6	V
I_I	Input current†	$V_I = V_{SS}$ to DV_{DD}			±10	uA
I_{OZ}	Off-state output current	$V_O = DV_{DD}$ or 0 V			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access‡	CV_{DD} = NOM, CPU clock = 150 MHz		470		mA
I_{DD2V}	Supply current, peripherals§	CV_{DD} = NOM, CPU clock = 150 MHz		250		mA
I_{DD3V}	Supply current, I/O pins¶	DV_{DD} = NOM, CPU clock = 150 MHz		85		mA
C_i	Input capacitance				10	pF
C_o	Output capacitance				10	pF

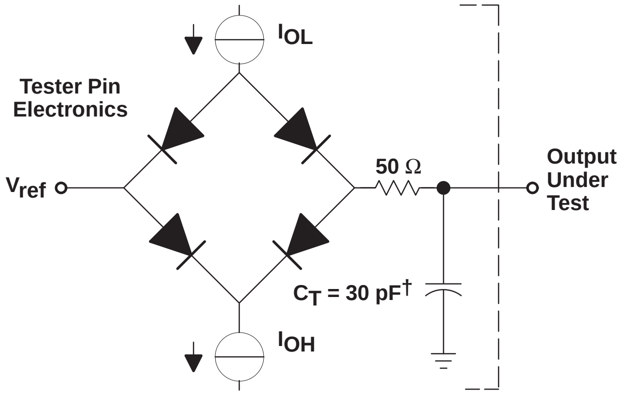
† TMS and TDI are not included due to internal pullups.
$\overline{\text{TRST}}$ is not included due to internal pulldown.

‡ 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, and DMA burst transfer between DMEM and SDRAM
50% of time: Timers at max rate, McBSPs at E1 rate, and 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.

signal-transition levels

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

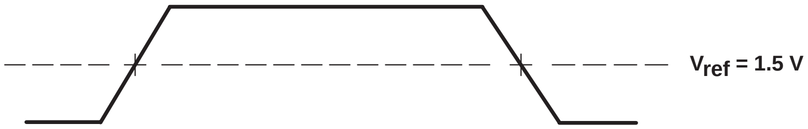


Figure 7. Input and Output Voltage Reference Levels for ac Timing Measurements

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN[†] (see Figure 8)

NO.			’C6701-150				’C6701-167				UNIT
			CLKMODE = x4		CLKMODE = x1		CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	26.7		6.7		24		6		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	10.9		3		9.8		2.7		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	10.9		3		9.8		2.7		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}.

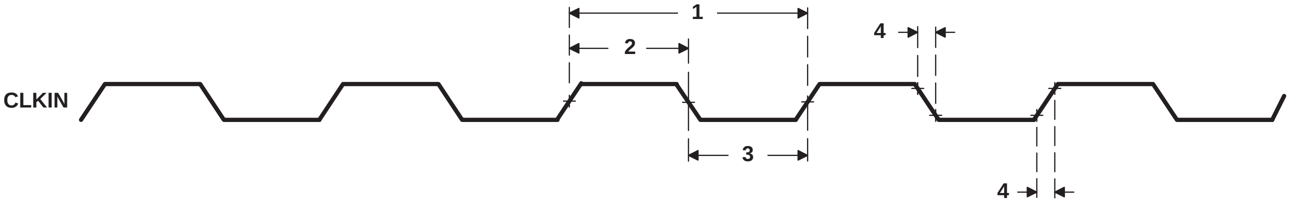


Figure 8. CLKIN Timings

switching characteristics for CLKOUT1[‡§] (see Figure 9)

NO.	PARAMETER		’C6701-150 ’C6701-167				UNIT
			CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	
1	$t_{c(CKO1)}$	Cycle time, CLKOUT1	P – 0.7	P + 0.7	P – 0.7	P + 0.7	ns
2	$t_{w(CKO1H)}$	Pulse duration, CLKOUT1 high	(P/2) – 0.5	(P/2) + 0.5	PH – 0.5	PH + 0.5	ns
3	$t_{w(CKO1L)}$	Pulse duration, CLKOUT1 low	(P/2) – 0.5	(P/2) + 0.5	PL – 0.5	PL + 0.5	ns
4	$t_{t(CKO1)}$	Transition time, CLKOUT1		0.6		0.6	ns

[‡] P = 1/CPU clock frequency in nanoseconds (ns).
[§] PH is the high period of CLKIN in ns and PL is the low period of CLKIN in ns.

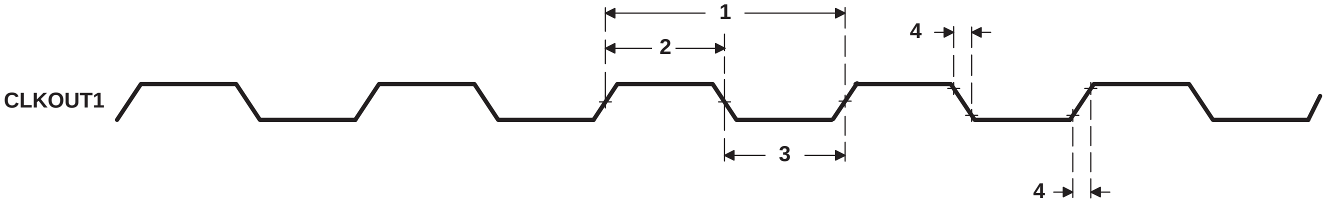


Figure 9. CLKOUT1 Timings

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT2[†] (see Figure 10)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	2P – 0.7	2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	P – 0.7	P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	P – 0.7	P + 0.7	ns
4	$t_{t(CKO2)}$	Transition time, CLKOUT2		0.6	ns

[†] P = 1/CPU clock frequency in ns.

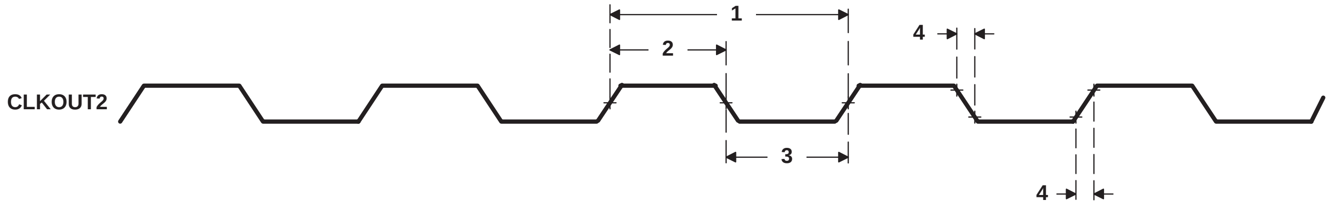


Figure 10. CLKOUT2 Timings

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 11)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{d(CKO1-SSCLK)}$	Delay time, CLKOUT1 edge to SSCLK edge	–0.8	3.4	ns
2	$t_{d(CKO1-SSCLK1/2)}$	Delay time, CLKOUT1 edge to SSCLK edge (1/2 clock rate)	–1.0	3.0	ns
3	$t_{d(CKO1-CKO2)}$	Delay time, CLKOUT1 edge to CLKOUT2 edge	–1.5	2.5	ns
4	$t_{d(CKO1-SDCLK)}$	Delay time, CLKOUT1 edge to SDCLK edge	–1.5	1.9	ns

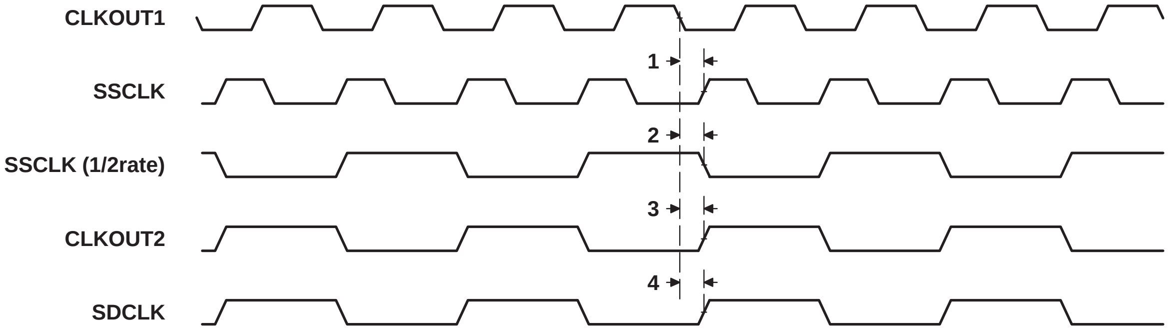


Figure 11. Relation of CLKOUT2, SDCLK, and SSCLK to CLKOUT1

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles[†] (see Figure 12 and Figure 13)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
6	$t_{su(EDV-CKO1H)}$	Setup time, read EDx valid before CLKOUT1 high	4.5		ns
7	$t_{h(CKO1H-EDV)}$	Hold time, read EDx valid after CLKOUT1 high	1.5		ns
10	$t_{su(ARDY-CKO1H)}$	Setup time, ARDY valid before CLKOUT1 high	3.5		ns
11	$t_{h(CKO1H-ARDY)}$	Hold time, ARDY valid after CLKOUT1 high	1.5		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 12 and Figure 13)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{d(CKO1H-CEV)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$x valid	–1.0	4.5	ns
2	$t_{d(CKO1H-BEV)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$x valid		4.5	ns
3	$t_{d(CKO1H-BEIV)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$x invalid	–1.0		ns
4	$t_{d(CKO1H-EAV)}$	Delay time, CLKOUT1 high to EAx valid		4.5	ns
5	$t_{d(CKO1H-EAIV)}$	Delay time, CLKOUT1 high to EAx invalid	–1.0		ns
8	$t_{d(CKO1H-AOEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AOE}}$ valid	–1.0	4.5	ns
9	$t_{d(CKO1H-AREV)}$	Delay time, CLKOUT1 high to $\overline{\text{ARE}}$ valid	–0.5	4.5	ns
12	$t_{d(CKO1H-EDV)}$	Delay time, CLKOUT1 high to EDx valid		4.5	ns
13	$t_{d(CKO1H-EDIV)}$	Delay time, CLKOUT1 high to EDx invalid	–1.0		ns
14	$t_{d(CKO1H-AWEV)}$	Delay time, CLKOUT1 high to $\overline{\text{AWE}}$ valid	–1.0	4.5	ns

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

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

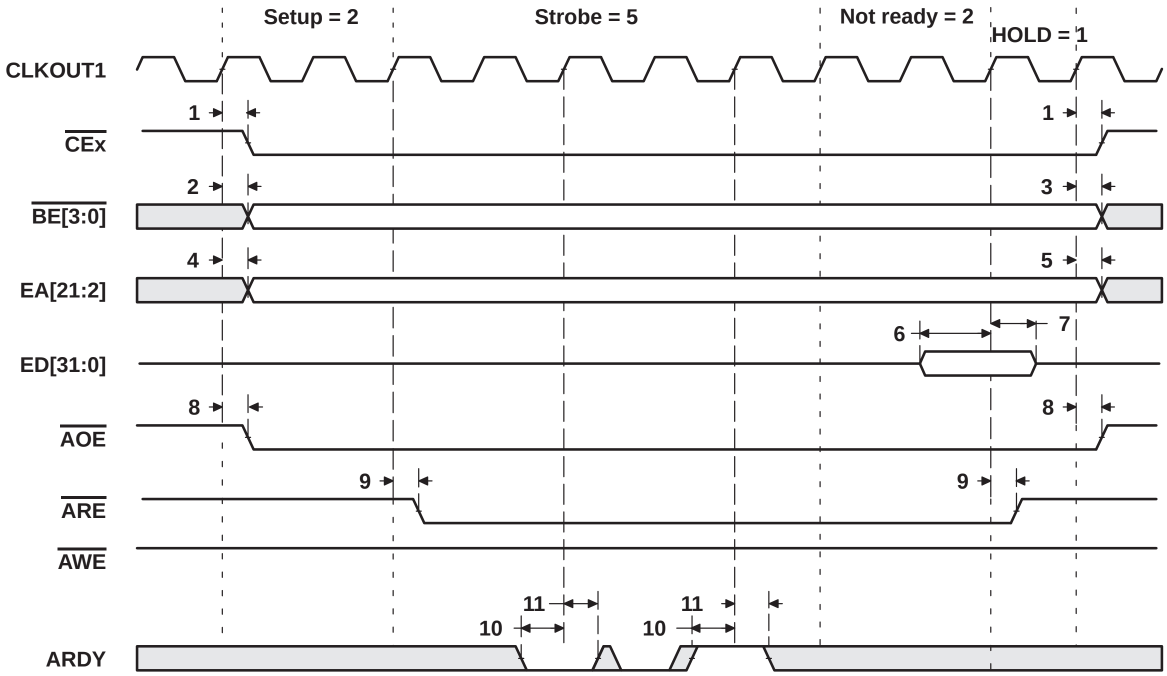


Figure 12. Asynchronous Memory Read Timing

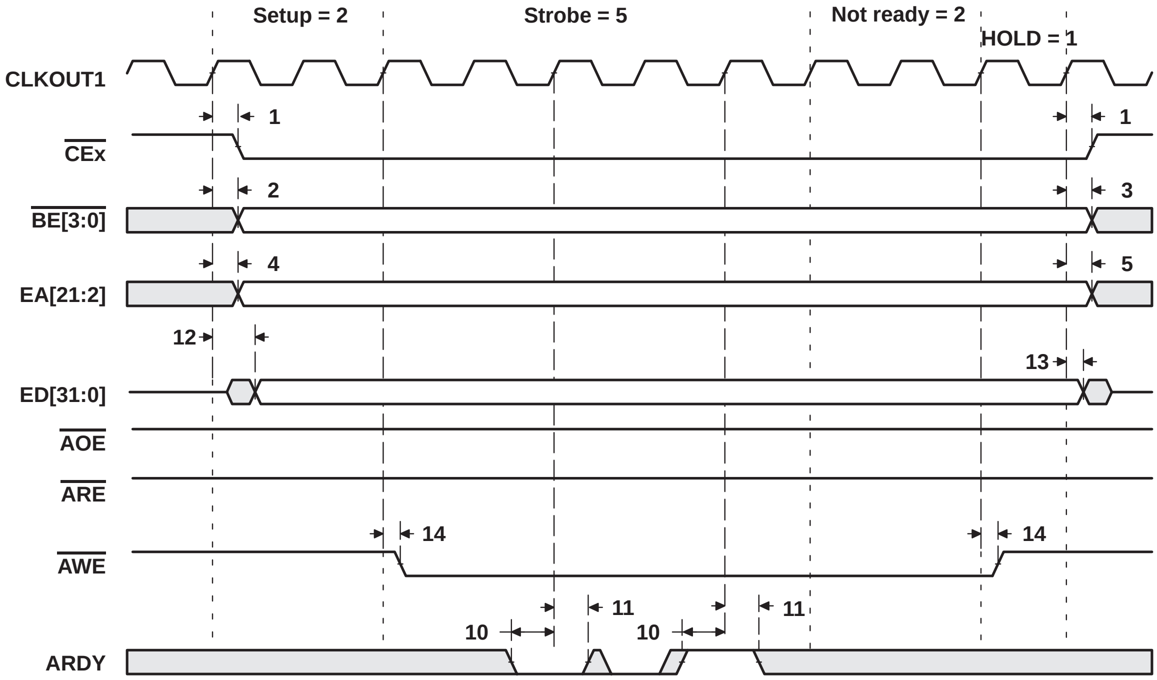


Figure 13. Asynchronous Memory Write Timing

SYNCHRONOUS-BURST MEMORY TIMING

timing requirements for synchronous-burst SRAM cycles (full-rate SSCLK) (see Figure 14)

NO.			’C6701-150		’C6701-167		UNIT
			MIN	MAX	MIN	MAX	
7	$t_{su(EDV\text{-}SSCLKH)}$	Setup time, read EDx valid before SSCLK high	2.0		2.0		ns
8	$t_{h(SSCLKH\text{-}EDV)}$	Hold time, read EDx valid after SSCLK high	2.1		2.1		ns

switching characteristics for synchronous-burst SRAM cycles[†] (full-rate SSCLK) (see Figure 14 and Figure 15)

NO.	PARAMETER		’C6701-150		’C6701-167		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{osu(CEV\text{-}SSCLKH)}$	Output setup time, $\overline{\text{CE}}$x valid before SSCLK high	0.5P – 1.5		0.5P – 1.3		ns
2	$t_{oh(SSCLKH\text{-}CEV)}$	Output hold time, $\overline{\text{CE}}$x valid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
3	$t_{osu(BEV\text{-}SSCLKH)}$	Output setup time, $\overline{\text{BE}}$x valid before SSCLK high	0.5P – 1.6		0.5P – 1.6		ns
4	$t_{oh(SSCLKH\text{-}BEIV)}$	Output hold time, $\overline{\text{BE}}$x invalid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
5	$t_{osu(EAV\text{-}SSCLKH)}$	Output setup time, EAx valid before SSCLK high	0.5P – 1.7		0.5P – 1.7		ns
6	$t_{oh(SSCLKH\text{-}EAIV)}$	Output hold time, EAx invalid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
9	$t_{osu(ADSV\text{-}SSCLKH)}$	Output setup time, $\overline{\text{SSADS}}$ valid before SSCLK high	0.5P – 1.5		0.5P – 1.3		ns
10	$t_{oh(SSCLKH\text{-}ADSV)}$	Output hold time, $\overline{\text{SSADS}}$ valid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
11	$t_{osu(OEV\text{-}SSCLKH)}$	Output setup time, $\overline{\text{SSOE}}$ valid before SSCLK high	0.5P – 1.5		0.5P – 1.3		ns
12	$t_{oh(SSCLKH\text{-}OEV)}$	Output hold time, $\overline{\text{SSOE}}$ valid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
13	$t_{osu(EDV\text{-}SSCLKH)}$	Output setup time, EDx valid before SSCLK high	0.5P – 1.5		0.5P – 1.3		ns
14	$t_{oh(SSCLKH\text{-}EDIV)}$	Output hold time, EDx invalid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns
15	$t_{osu(WEV\text{-}SSCLKH)}$	Output setup time, $\overline{\text{SSWE}}$ valid before SSCLK high	0.5P – 1.5		0.5P – 1.3		ns
16	$t_{oh(SSCLKH\text{-}WEV)}$	Output hold time, $\overline{\text{SSWE}}$ valid after SSCLK high	0.5P – 2.5		0.5P – 2.3		ns

[†] When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns. For CLKMODE x1, 0.5P is defined as PH (pulse duration of CLKIN high) for all output setup times; 0.5P is defined as PL (pulse duration of CLKIN low) for all output hold times.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

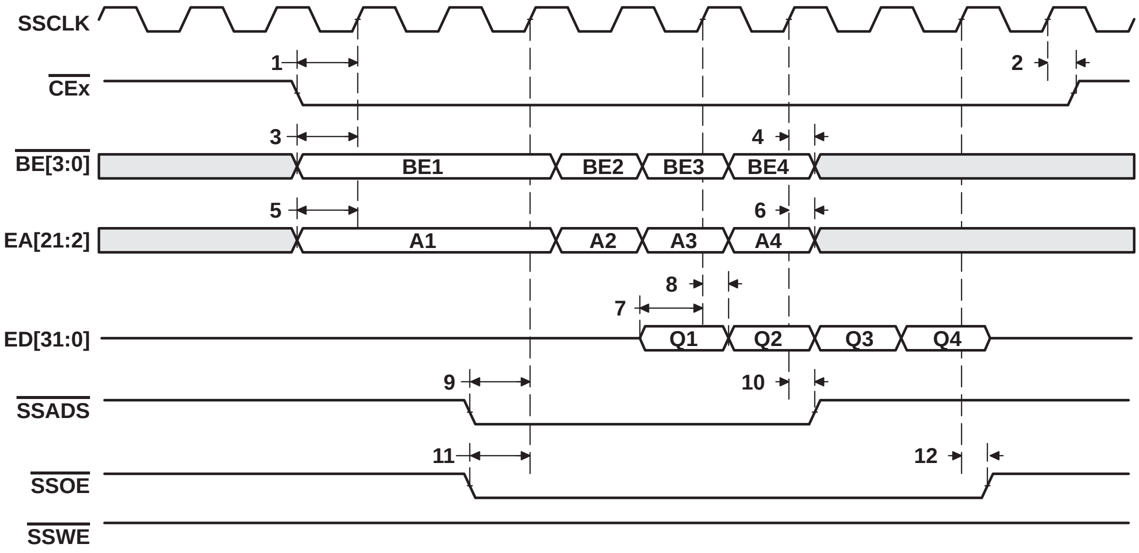


Figure 14. SBSRAM Read Timing (Full-Rate SSCLK)

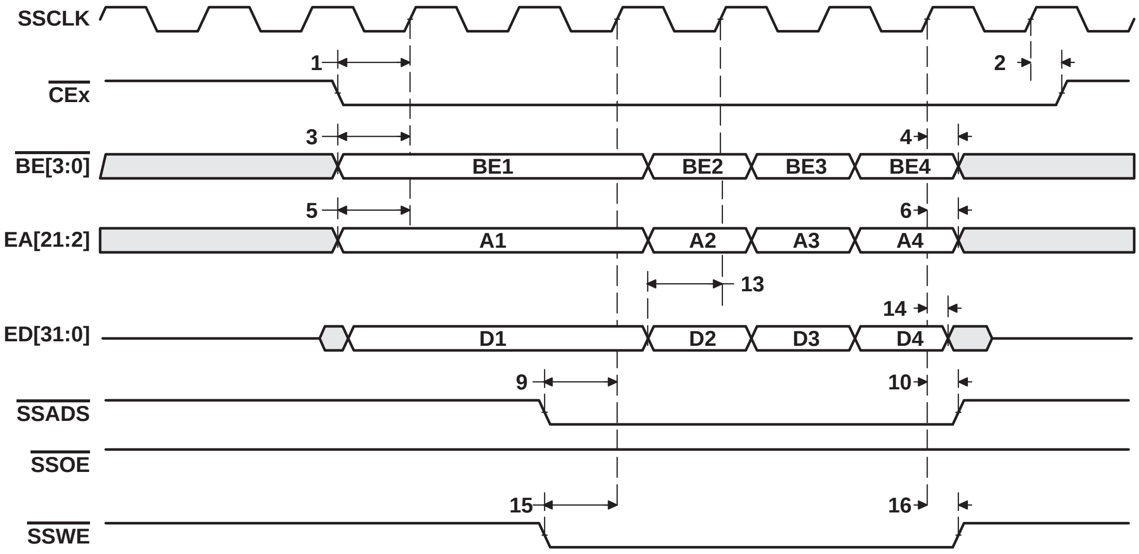


Figure 15. SBSRAM Write Timing (Full-Rate SSCLK)

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

timing requirements for synchronous-burst SRAM cycles (half-rate SSCLK) (see Figure 16)

NO.		’C6701-150		’C6701-167		UNIT
		MIN	MAX	MIN	MAX	
7	$t_{\text{su(EDV-SSCLKH)}}$ Setup time, read EDx valid before SSCLK high	3.8		3.6		ns
8	$t_{\text{h(SSCLKH-EDV)}}$ Hold time, read EDx valid after SSCLK high	2		1.5		ns

switching characteristics for synchronous-burst SRAM cycles[†] (half-rate SSCLK) (see Figure 16 and Figure 17)

NO.	PARAMETER	’C6701-150		’C6701-167		UNIT
		MIN	MAX	MIN	MAX	
1	$t_{\text{osu(CEV-SSCLKH)}}$ Output setup time, $\overline{\text{CE}}$x valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
2	$t_{\text{oh(SSCLKH-CEV)}}$ Output hold time, $\overline{\text{CE}}$x valid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
3	$t_{\text{osu(BEV-SSCLKH)}}$ Output setup time, $\overline{\text{BE}}$x valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
4	$t_{\text{oh(SSCLKH-BEIV)}}$ Output hold time, $\overline{\text{BE}}$x invalid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
5	$t_{\text{osu(EAV-SSCLKH)}}$ Output setup time, EAx valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
6	$t_{\text{oh(SSCLKH-EAIV)}}$ Output hold time, EAx invalid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
9	$t_{\text{osu(ADSV-SSCLKH)}}$ Output setup time, $\overline{\text{SSADS}}$ valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
10	$t_{\text{oh(SSCLKH-ADSV)}}$ Output hold time, $\overline{\text{SSADS}}$ valid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
11	$t_{\text{osu(OEV-SSCLKH)}}$ Output setup time, $\overline{\text{SSOE}}$ valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
12	$t_{\text{oh(SSCLKH-OEV)}}$ Output hold time, $\overline{\text{SSOE}}$ valid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
13	$t_{\text{osu(EDV-SSCLKH)}}$ Output setup time, EDx valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
14	$t_{\text{oh(SSCLKH-EDIV)}}$ Output hold time, EDx invalid after SSCLK high	0.5P – 2.3		0.5P – 2		ns
15	$t_{\text{osu(WEV-SSCLKH)}}$ Output setup time, $\overline{\text{SSWE}}$ valid before SSCLK high	1.5P – 5.5		1.5P – 4.5		ns
16	$t_{\text{oh(SSCLKH-WEV)}}$ Output hold time, $\overline{\text{SSWE}}$ valid after SSCLK high	0.5P – 2.3		0.5P – 2		ns

---

[†] When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.
For CLKMODE x1:
1.5P = P + PH, where P = 1/CPU clock frequency, and PH = pulse duration of CLKIN high.
0.5P = PL, where PL = pulse duration of CLKIN low.

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)

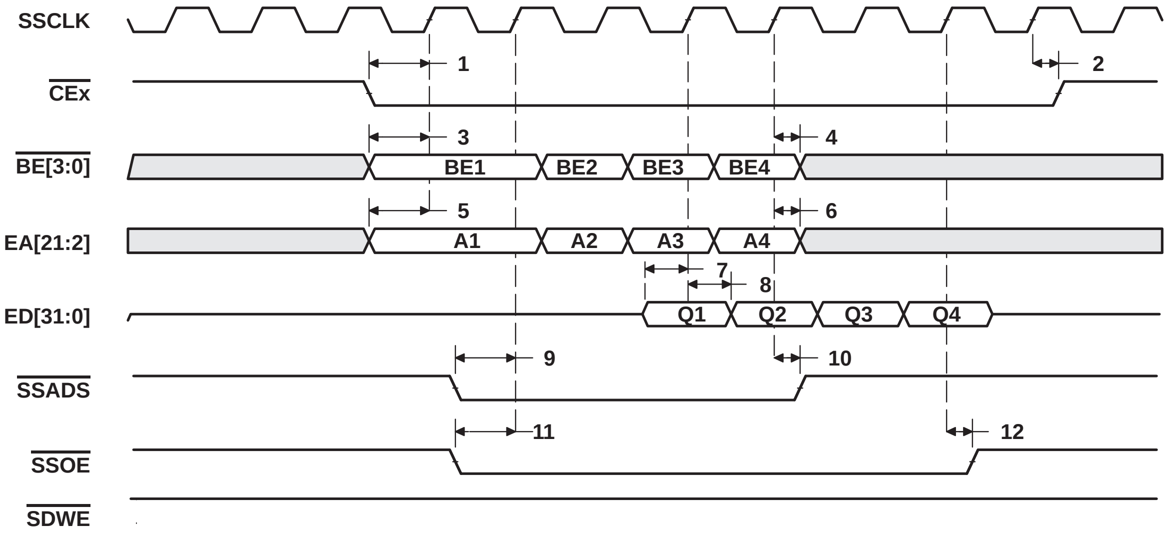


Figure 16. SBSRAM Read Timing (1/2 Rate SSCLK)

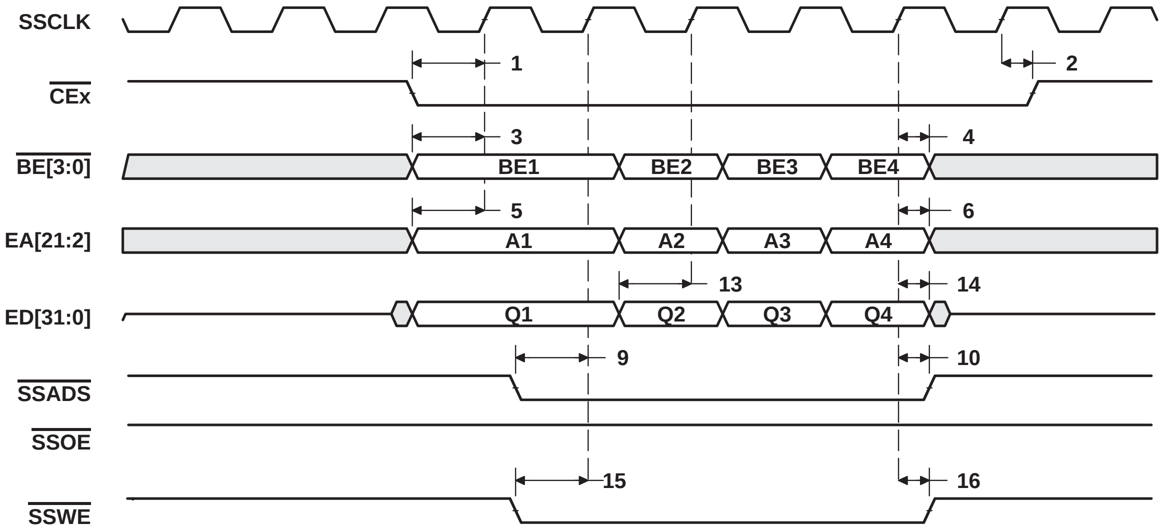


Figure 17. SBSRAM Write Timing (1/2 Rate SSCLK)

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles (see Figure 18)

NO.			’C6701-150		’C6701-167		UNIT
			MIN	MAX	MIN	MAX	
7	$t_{\text{su(EDV-SDCLKH)}}$	Setup time, read EDx valid before SDCLK high	1.8		1.8		ns
8	$t_{\text{h(SDCLKH-EDV)}}$	Hold time, read EDx valid after SDCLK high	3		3		ns

switching characteristics for synchronous DRAM cycles[†] (see Figure 18–Figure 23)

NO.	PARAMETER		’C6701-150		’C6701-167		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{\text{osu(CEV-SDCLKH)}}$	Output setup time, $\overline{\text{CE}}$x valid before SDCLK high	1.5P – 5		1.5P – 4		ns
2	$t_{\text{oh(SDCLKH-CEV)}}$	Output hold time, $\overline{\text{CE}}$x valid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
3	$t_{\text{osu(BEV-SDCLKH)}}$	Output setup time, $\overline{\text{BE}}$x valid before SDCLK high	1.5P – 5		1.5P – 4		ns
4	$t_{\text{oh(SDCLKH-BEIV)}}$	Output hold time, $\overline{\text{BE}}$x invalid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
5	$t_{\text{osu(EAV-SDCLKH)}}$	Output setup time, EAx valid before SDCLK high	1.5P – 5		1.5P – 4		ns
6	$t_{\text{oh(SDCLKH-EAIV)}}$	Output hold time, EAx invalid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
9	$t_{\text{osu(SDCAS-SDCLKH)}}$	Output setup time, $\overline{\text{SDCAS}}$ valid before SDCLK high	1.5P – 5		1.5P – 4		ns
10	$t_{\text{oh(SDCLKH-SDCAS)}}$	Output hold time, $\overline{\text{SDCAS}}$ valid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
11	$t_{\text{osu(EDV-SDCLKH)}}$	Output setup time, EDx valid before SDCLK high	1.5P – 5		1.5P – 4		ns
12	$t_{\text{oh(SDCLKH-EDIV)}}$	Output hold time, EDx invalid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
13	$t_{\text{osu(SDWE-SDCLKH)}}$	Output setup time, $\overline{\text{SDWE}}$ valid before SDCLK high	1.5P – 5		1.5P – 4		ns
14	$t_{\text{oh(SDCLKH-SDWE)}}$	Output hold time, $\overline{\text{SDWE}}$ valid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
15	$t_{\text{osu(SDA10V-SDCLKH)}}$	Output setup time, SDA10 valid before SDCLK high	1.5P – 5		1.5P – 4		ns
16	$t_{\text{oh(SDCLKH-SDA10IV)}}$	Output hold time, SDA10 invalid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns
17	$t_{\text{osu(SDRAS-SDCLKH)}}$	Output setup time, $\overline{\text{SDRAS}}$ valid before SDCLK high	1.5P – 5		1.5P – 4		ns
18	$t_{\text{oh(SDCLKH-SDRAS)}}$	Output hold time, $\overline{\text{SDRAS}}$ valid after SDCLK high	0.5P – 1.9		0.5P – 1.5		ns

[†] When the PLL is used (CLKMODE x4), P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.
For CLKMODE x1:
1.5P = P + PH, where P = 1/CPU clock frequency, and PH = pulse duration of CLKIN high.
0.5P = PL, where PL = pulse duration of CLKIN low.

SYNCHRONOUS DRAM TIMING (CONTINUED)

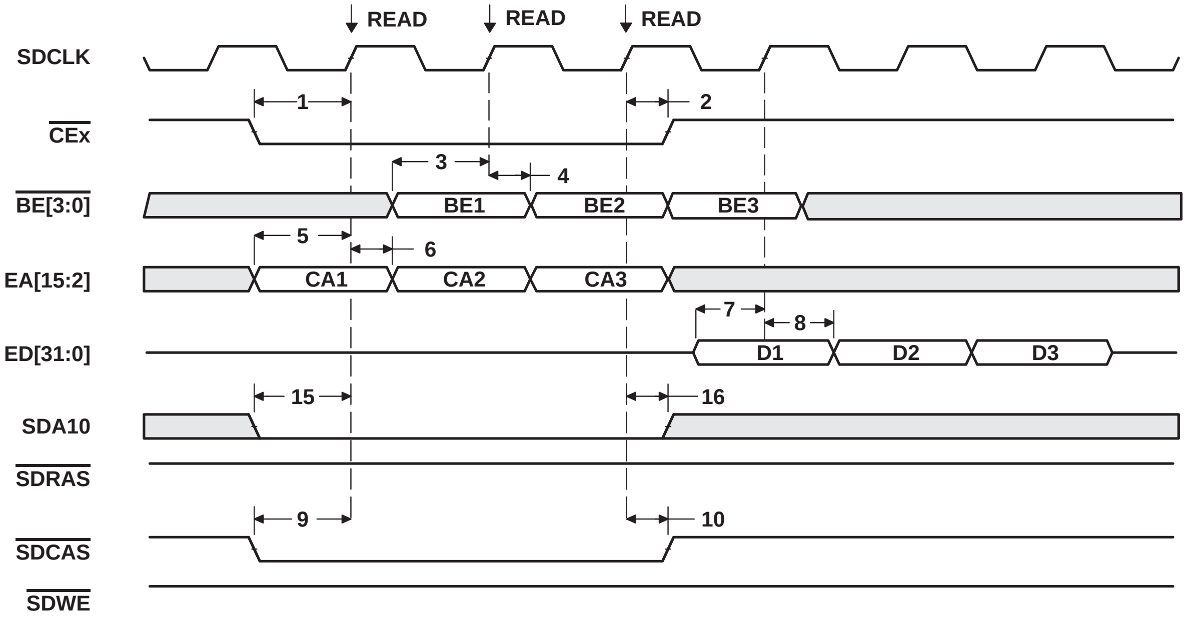


Figure 18. Three SDRAM Read Commands

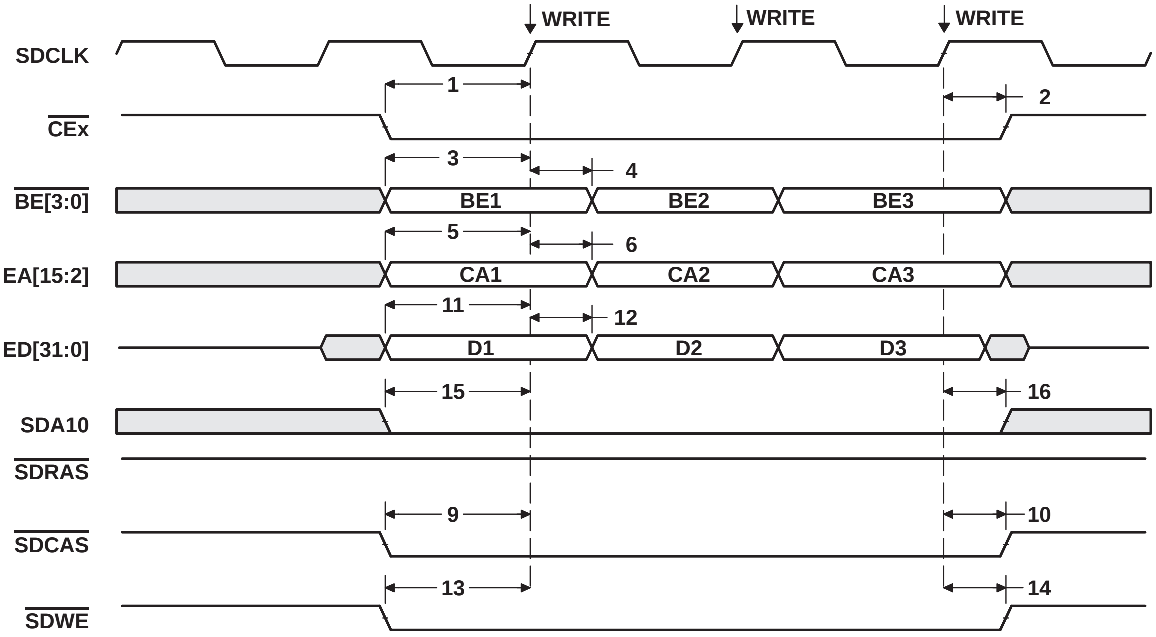


Figure 19. Three SDRAM Write Commands

SYNCHRONOUS DRAM TIMING (CONTINUED)

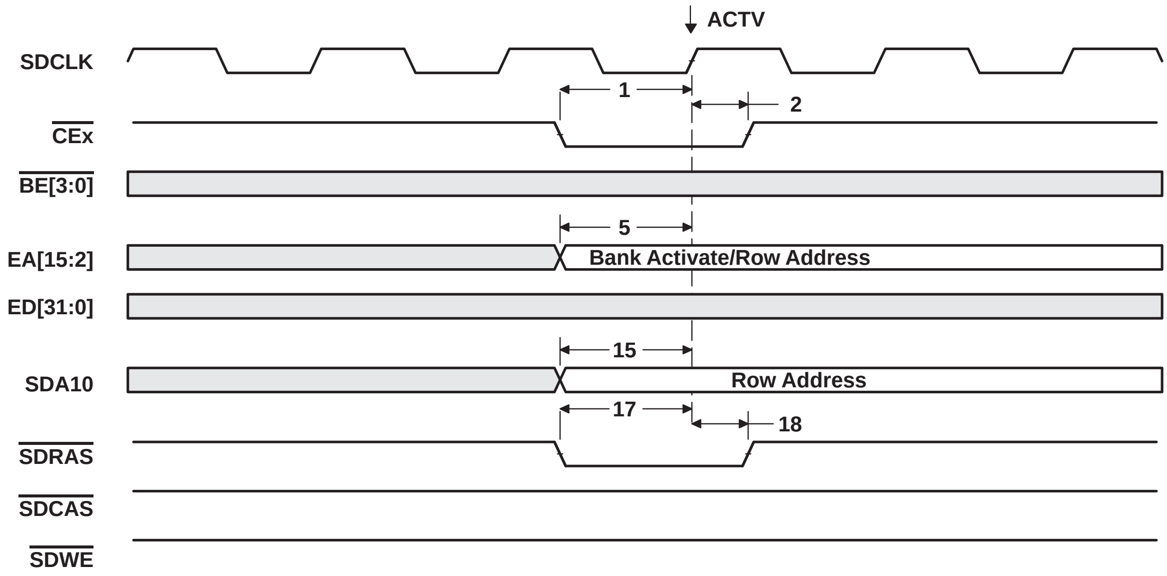


Figure 20. SDRAM ACTV Command

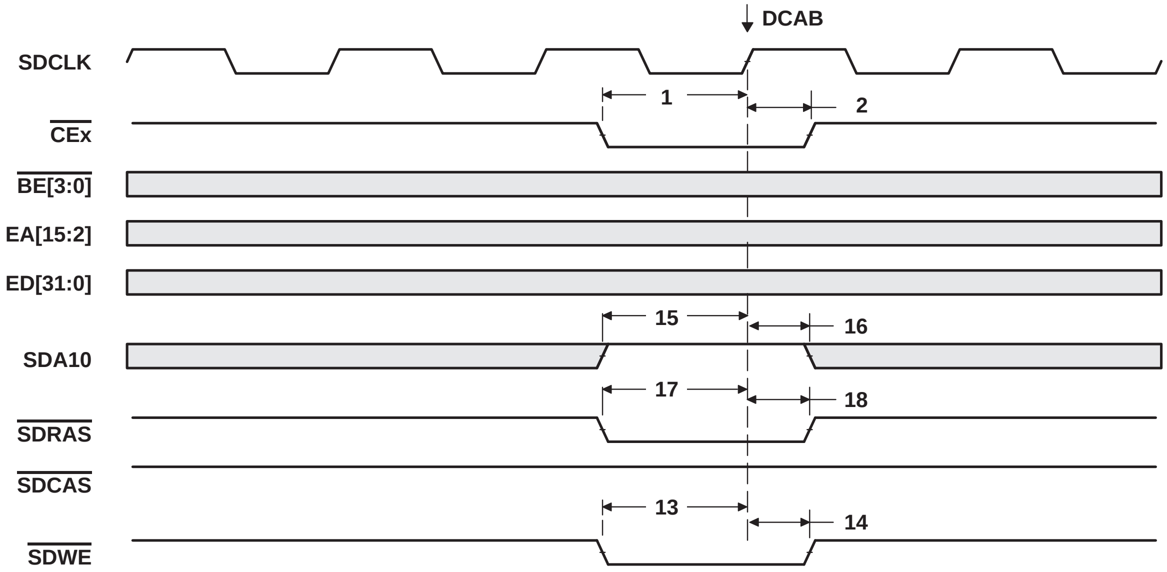


Figure 21. SDRAM DCAB Command

SYNCHRONOUS DRAM TIMING (CONTINUED)

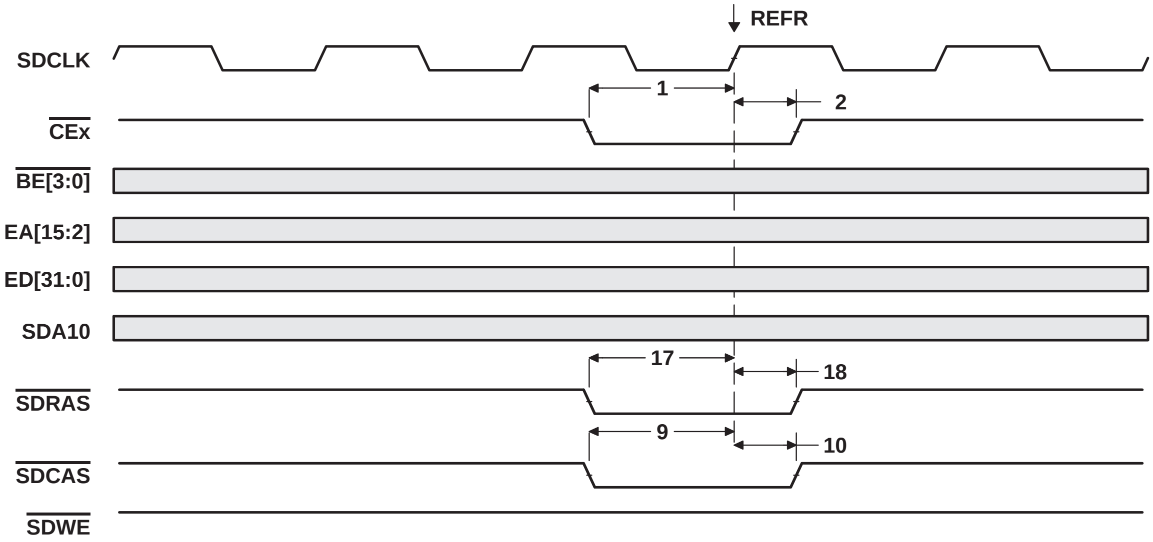


Figure 22. SDRAM REFR Command

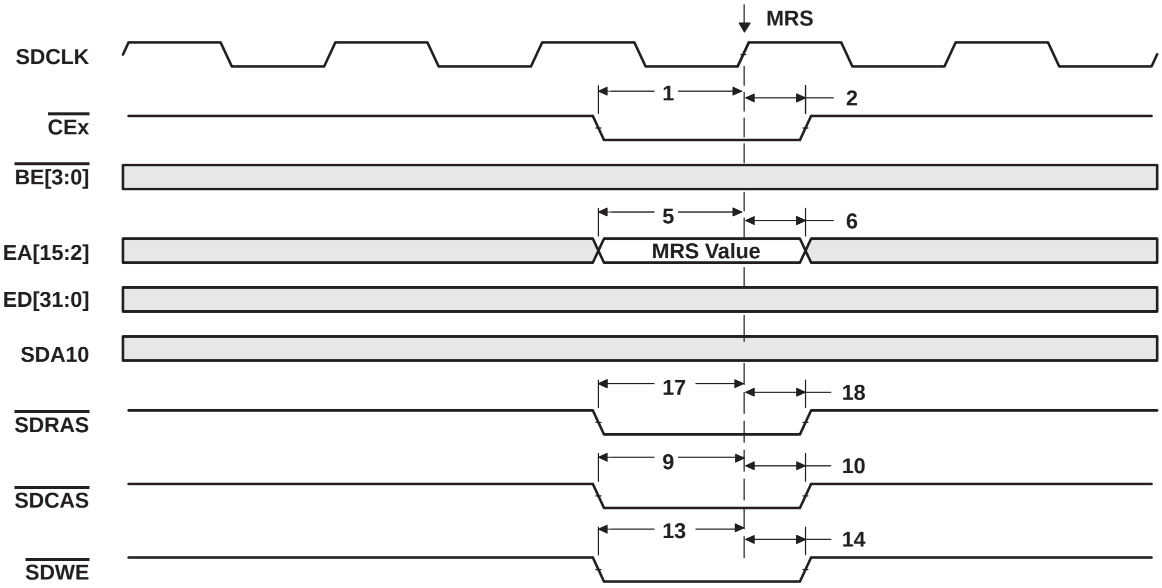


Figure 23. SDRAM MRS Command

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

timing requirements for the hold/hold acknowledge cycles[†] (see Figure 24)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{su(HOLDH-CKO1H)}$	Setup time, $\overline{\text{HOLD}}$ high before CLKOUT1 high	5		ns
2	$t_{h(CKO1H-HOLDL)}$	Hold time, $\overline{\text{HOLD}}$ low after CLKOUT1 high	2		ns

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

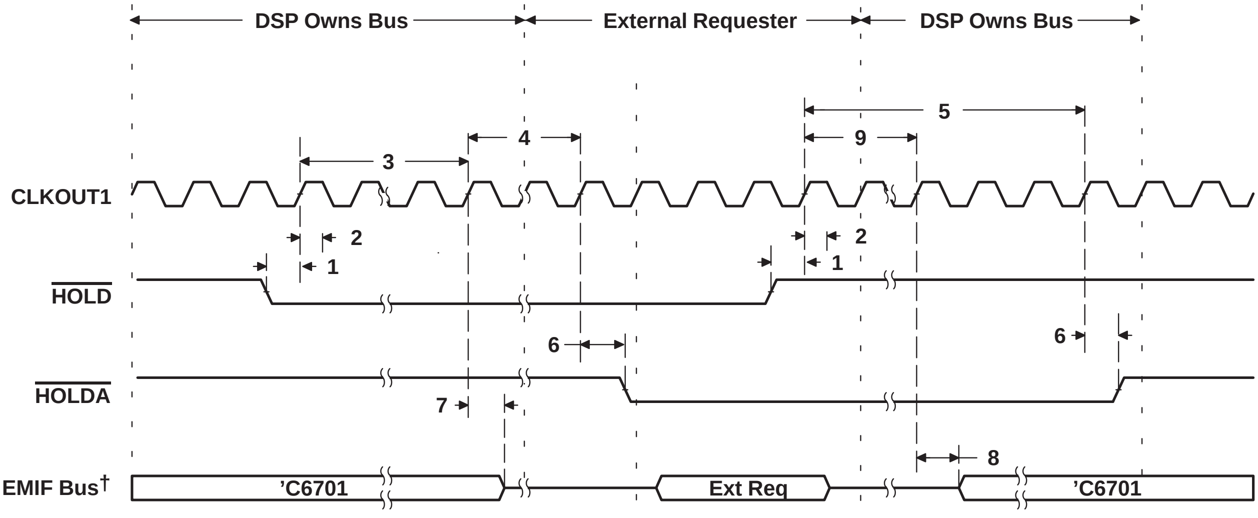
switching characteristics for the hold/hold acknowledge cycles[‡] (see Figure 24)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
3	$t_{R(HOLDL-EMHZ)}$	Response time, $\overline{\text{HOLD}}$ low to EMIF high impedance	4P	[§]	ns
4	$t_{R(EMHZ-HOLDAL)}$	Response time, EMIF high impedance to $\overline{\text{HOLDA}}$ low		2P	ns
5	$t_{R(HOLDH-HOLDAH)}$	Response time, $\overline{\text{HOLD}}$ high to $\overline{\text{HOLDA}}$ high	4P	7P	ns
6	$t_{d(CKO1H-HOLDAL)}$	Delay time, CLKOUT1 high to $\overline{\text{HOLDA}}$ valid	1	8	ns
7	$t_{d(CKO1H-BHZ)}$	Delay time, CLKOUT1 high to EMIF Bus high impedance[¶]	1	8	ns
8	$t_{d(CKO1H-BLZ)}$	Delay time, CLKOUT1 high to EMIF Bus low impedance[¶]	1	12	ns
9	$t_{R(HOLDH-BLZ)}$	Response time, $\overline{\text{HOLD}}$ high to EMIF Bus low impedance[¶]	3P	6P	ns

[‡] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[§] All pending EMIF transactions are allowed to complete before $\overline{\text{HOLDA}}$ is asserted. The worst cases for this is an asynchronous read or write with external ARDY used or a minimum of eight consecutive SDRAM reads or writes when RBTR8 = 1. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting the NOHOLD = 1.

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



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

Figure 24. $\overline{\text{HOLD}}$/$\overline{\text{HOLDA}}$ Timing

RESET TIMING

timing requirements for reset (see Figure 25)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{w(RESET)}$	Width of the RESET pulse (PLL stable)[†]	10		CLKOUT1 cycles
		Width of the RESET pulse (PLL needs to sync up)[‡]	250		µs

[†] This parameter applies to CLKMODE x1 when CLKIN is stable and applies to CLKMODE x4 when CLKIN and PLL are stable.

[‡] This parameter only applies to CLKMODE x4. The 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.

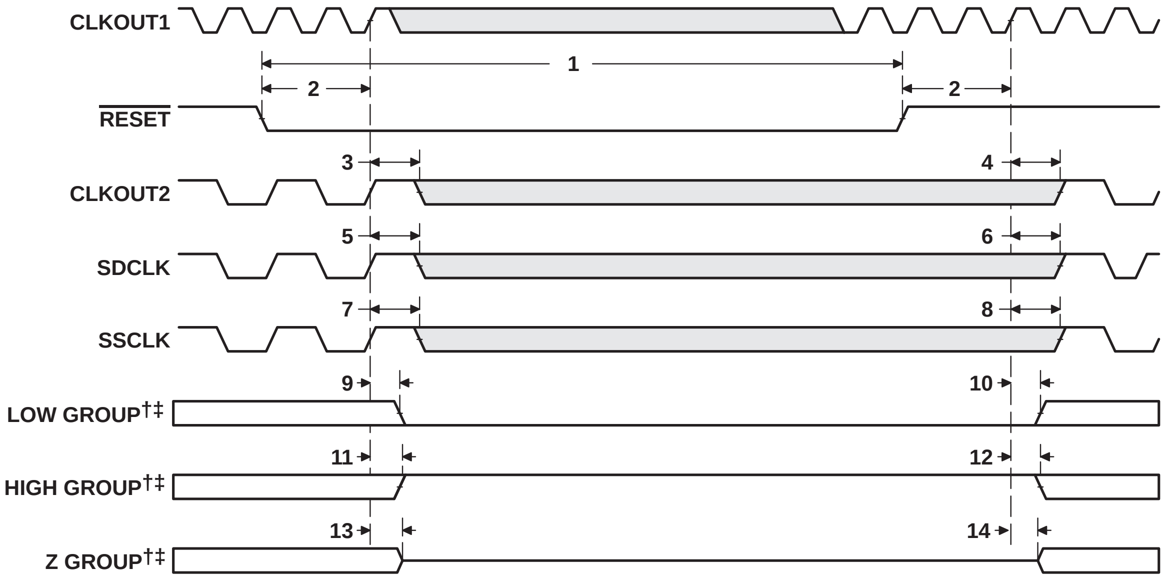
switching characteristics during reset[§][¶] (see Figure 25)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
2	$t_{R(RESET)}$	Response time to change of value in RESET signal	1		CLKOUT1 cycles
3	$t_{d(CKO1H-CKO2IV)}$	Delay time, CLKOUT1 high to CLKOUT2 invalid	–1		ns
4	$t_{d(CKO1H-CKO2V)}$	Delay time, CLKOUT1 high to CLKOUT2 valid		10	ns
5	$t_{d(CKO1H-SDCLKIV)}$	Delay time, CLKOUT1 high to SDCLK invalid	–1		ns
6	$t_{d(CKO1H-SDCLKV)}$	Delay time, CLKOUT1 high to SDCLK valid		10	ns
7	$t_{d(CKO1H-SSCKIV)}$	Delay time, CLKOUT1 high to SSCLK invalid	–1		ns
8	$t_{d(CKO1H-SSCKV)}$	Delay time, CLKOUT1 high to SSCLK valid		10	ns
9	$t_{d(CKO1H-LOWIV)}$	Delay time, CLKOUT1 high to low group invalid	–1		ns
10	$t_{d(CKO1H-LOWV)}$	Delay time, CLKOUT1 high to low group valid		10	ns
11	$t_{d(CKO1H-HIGHIV)}$	Delay time, CLKOUT1 high to high group invalid	–1		ns
12	$t_{d(CKO1H-HIGHV)}$	Delay time, CLKOUT1 high to high group valid		10	ns
13	$t_{d(CKO1H-ZHZ)}$	Delay time, CLKOUT1 high to Z group high impedance	–1		ns
14	$t_{d(CKO1H-ZV)}$	Delay time, CLKOUT1 high to Z group valid		10	ns

[§] Low group consists of: IACK, INUM[3:0], DMAC[3:0], PD, TOUT0, and TOUT1.
High group consists of: 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.

[¶] HRDY is gated by input HCS.
If HCS = 0 at device reset, HRDY belongs to the high group.
If HCS = 1 at device reset, HRDY belongs to the low group.

RESET TIMING (CONTINUED)



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

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

Figure 25. Reset Timing

EXTERNAL INTERRUPT TIMING

timing requirements for interrupt response cycles[†‡] (see Figure 26)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
2	$t_{w(ILOW)}$	Width of the interrupt pulse low	2P		ns
3	$t_{w(IHIGH)}$	Width of the interrupt pulse high	2P		ns

[†] Interrupt signals are synchronized internally and are potentially recognized one cycle later if setup and hold times are violated. Thus, they can be connected to asynchronous inputs.

[‡] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

switching characteristics during interrupt response cycles[§] (see Figure 26)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{R(EINTH\text{-}IACKH)}$	Response time, EXT_INTx high to IACK high	9P		ns
4	$t_{d(CKO2L\text{-}IACKV)}$	Delay time, CLKOUT2 low to IACK valid	–0.5P	13 – 0.5P	ns
5	$t_{d(CKO2L\text{-}INUMV)}$	Delay time, CLKOUT2 low to INUMx valid		10 – 0.5P	ns
6	$t_{d(CKO2L\text{-}INUMIV)}$	Delay time, CLKOUT2 low to INUMx invalid	–0.5P		ns

[§] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.
When the PLL is used (CLKMODE x4), 0.5P = 1/(2 × CPU clock frequency).
For CLKMODE x1: 0.5P = PH, where PH is the high period of CLKIN.

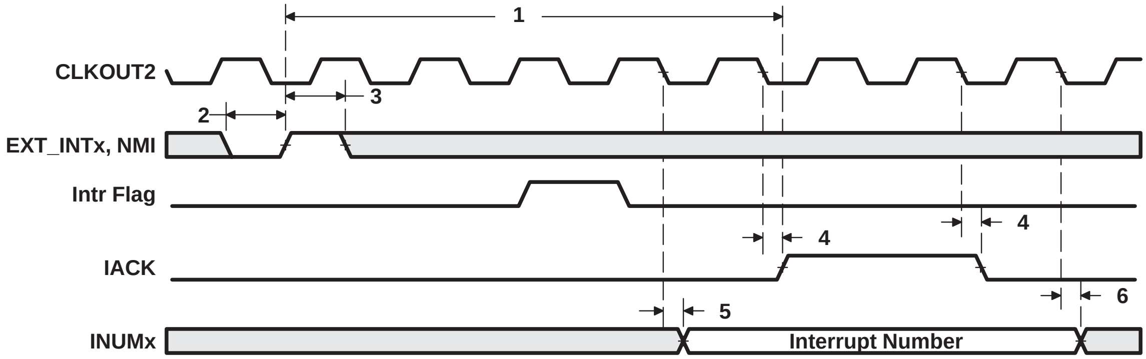


Figure 26. Interrupt Timing

HOST-PORT INTERFACE TIMING

timing requirements for host-port interface cycles[†][‡] (see Figure 27, Figure 28, Figure 29, and Figure 30)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{\text{su(SEL-HSTBL)}}$	Setup time, select signals[§] valid before $\overline{\text{HSTROBE}}$ low	4		ns
2	$t_{\text{h(HSTBL-SEL)}}$	Hold time, select signals[§] valid after $\overline{\text{HSTROBE}}$ low	2		ns
3	$t_{\text{w(HSTBL)}}$	Pulse duration, $\overline{\text{HSTROBE}}$ low	2P		ns
4	$t_{\text{w(HSTBH)}}$	Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses	2P		ns
10	$t_{\text{su(SEL-HASL)}}$	Setup time, select signals[§] valid before $\overline{\text{HAS}}$ low	4		ns
11	$t_{\text{h(HASL-SEL)}}$	Hold time, select signals[§] valid after $\overline{\text{HAS}}$ low	2		ns
12	$t_{\text{su(HDV-HSTBH)}}$	Setup time, host data valid before $\overline{\text{HSTROBE}}$ high	3		ns
13	$t_{\text{h(HSTBH-HDV)}}$	Hold time, host data valid after $\overline{\text{HSTROBE}}$ high	2		ns
14	$t_{\text{h(HRDYL-HSTBL)}}$	Hold time, $\overline{\text{HSTROBE}}$ low after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly.	1		ns
18	$t_{\text{su(HASL-HSTBL)}}$	Setup time, $\overline{\text{HAS}}$ low before $\overline{\text{HSTROBE}}$ low	2		ns
19	$t_{\text{h(HSTBL-HASL)}}$	Hold time, $\overline{\text{HAS}}$ low after $\overline{\text{HSTROBE}}$ low	2		ns

[†] $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.

[‡] The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[§] Select signals include: HCNTRL[1:0], HR/$\overline{\text{W}}$, and HHWIL.

switching characteristics during host-port interface cycles[†][‡] (see Figure 27, Figure 28, Figure 29, and Figure 30)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
5	$t_{\text{d(HCS-HRDY)}}$	Delay time, $\overline{\text{HCS}}$ to $\overline{\text{HRDY}}$[¶]	1	12	ns
6	$t_{\text{d(HSTBL-HRDYH)}}$	Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high[#]	1	12	ns
7	$t_{\text{oh(HSTBL-HDLZ)}}$	Output hold time, HD low impedance after $\overline{\text{HSTROBE}}$ low for an HPI read	4		ns
8	$t_{\text{d(HDV-HRDYL)}}$	Delay time, HD valid to $\overline{\text{HRDY}}$ low	P – 3	P + 3	ns
9	$t_{\text{oh(HSTBH-HDV)}}$	Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high	3	12	ns
15	$t_{\text{d(HSTBH-HDHZ)}}$	Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance	3	12	ns
16	$t_{\text{d(HSTBL-HDV)}}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD valid	3	12	ns
17	$t_{\text{d(HSTBH-HRDYH)}}$	Delay time, $\overline{\text{HSTROBE}}$ high to $\overline{\text{HRDY}}$ high[‖]	1	12	ns

[†] $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.

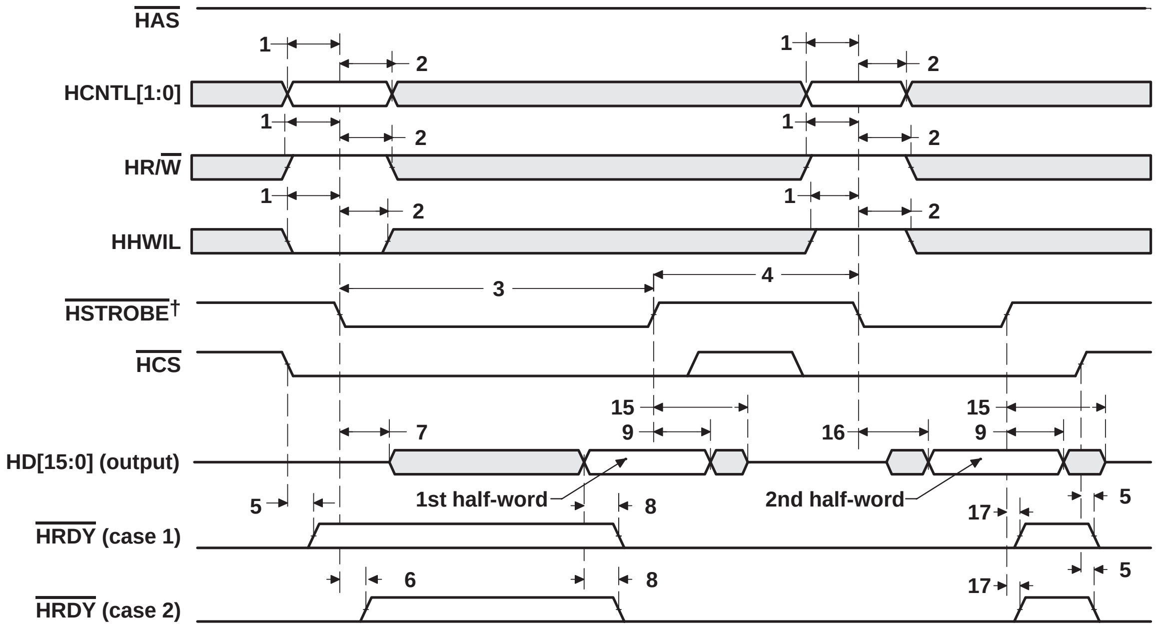
[‡] The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[¶] $\overline{\text{HCS}}$ enables $\overline{\text{HRDY}}$, and $\overline{\text{HRDY}}$ is always low when $\overline{\text{HCS}}$ is high. The case where $\overline{\text{HRDY}}$ goes high when $\overline{\text{HCS}}$ falls indicates that HPI is busy completing a previous HPID write or READ with autoincrement.

[#] This parameter is used during an HPID read. At the beginning of the first half-word transfer on the falling edge of $\overline{\text{HSTROBE}}$, the HPI sends the request to the DMA auxiliary channel, and $\overline{\text{HRDY}}$ remains high until the DMA auxiliary channel loads the requested data into HPID.

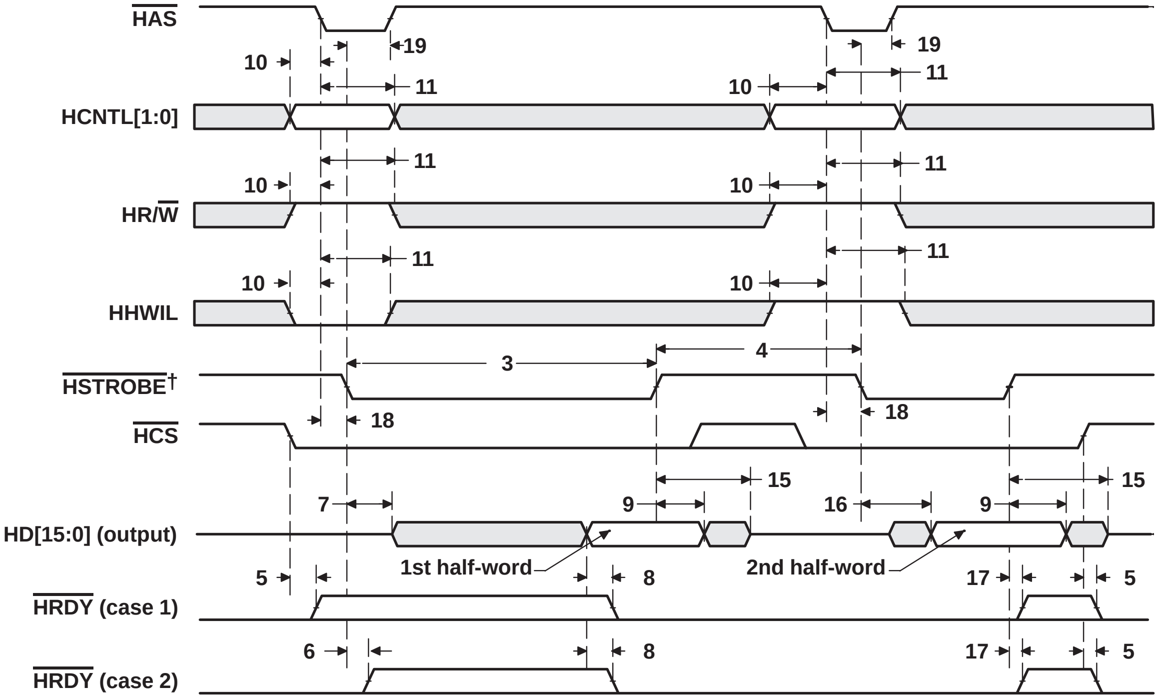
[‖] This parameter is used after the second half-word of an HPID write or autoincrement read. $\overline{\text{HRDY}}$ remains low if the access is not an HPID write or autoincrement read. Reading or writing to HPIC or HPIA does not affect the $\overline{\text{HRDY}}$ signal.

HOST-PORT INTERFACE TIMING (CONTINUED)



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

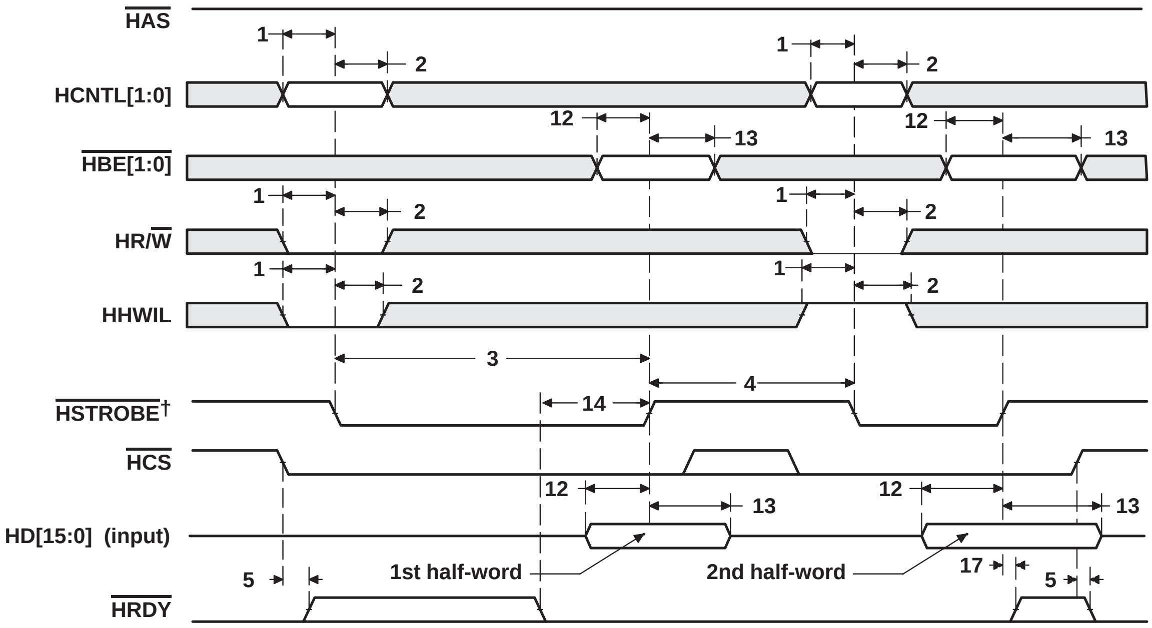
Figure 27. HPI Read Timing ($\overline{\text{HAS}}$ Not Used, Tied High)



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

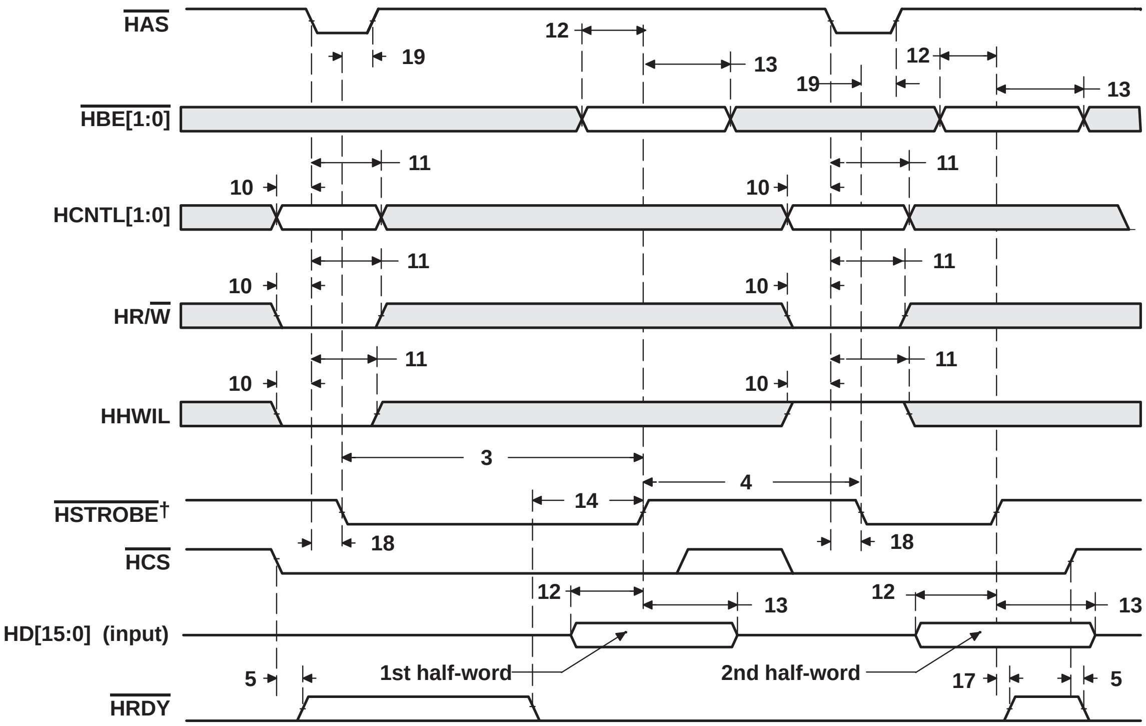
Figure 28. HPI Read Timing ($\overline{\text{HAS}}$ Used)

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 29. HPI Write 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 30. HPI Write Timing ($\overline{\text{HAS}}$ Used)

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP[†][‡] (see Figure 31)

NO.				’C6701-150 ’C6701-167		UNIT
				MIN	MAX	
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1		ns
5	$t_{su(FRH\text{-}CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	13		ns
			CLKR ext	4		
6	$t_{h(CKRL\text{-}FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	7		ns
			CLKR ext	4		
7	$t_{su(DRV\text{-}CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	10		ns
			CLKR ext	1		
8	$t_{h(CKRL\text{-}DRV)}$	Hold time, DR valid after CLKR low	CLKR int	4		ns
			CLKR ext	4		
10	$t_{su(FXH\text{-}CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	13		ns
			CLKX ext	4		
11	$t_{h(CKXL\text{-}FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	7		ns
			CLKX ext	3		

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[‡] CLKRP = CLKXP = FSRP = FSXP = 0 in the pin control register (PCR). If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics for McBSP†‡§ (see Figure 31)

NO.	PARAMETER			’C6701-150 ’C6701-167		UNIT
				MIN	MAX	
1	$t_{\text{d(CKSH-CKRXH)}}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		3	15	ns
2	$t_{\text{c(CKRX)}}$	Cycle time, CLKR/X	CLKR/X int	2P		ns
3	$t_{\text{w(CKRX)}}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C – 1¶	C + 1¶	ns
4	$t_{\text{d(CKRH-FRV)}}$	Delay time, CLKR high to internal FSR valid	CLKR int	–4	4	ns
9	$t_{\text{d(CKXH-FXV)}}$	Delay time, CLKX high to internal FSX valid	CLKX int	–4	5	ns
			CLKX ext	3	16	
12	$t_{\text{dis(CKXH-DXHZ)}}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	–3	2	ns
			CLKX ext	2	9	
13	$t_{\text{d(CKXH-DXV)}}$	Delay time, CLKX high to DX valid.	CLKX int	–2	4	ns
			CLKX ext	3	16	
14	$t_{\text{d(FXH-DXV)}}$	Delay time, FSX high to DX valid. ONLY applies when in data delay 0 (XDATDLY = 00b) mode.	FSX int	–2	4	ns
			FSX ext	2	10	

† CLKRP = CLKXP = FSRP = FSXP = 0 in the pin control register (PCR). If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.

‡ Minimum delay times also represent minimum output hold times.

§ P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

¶ C = H or L
S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

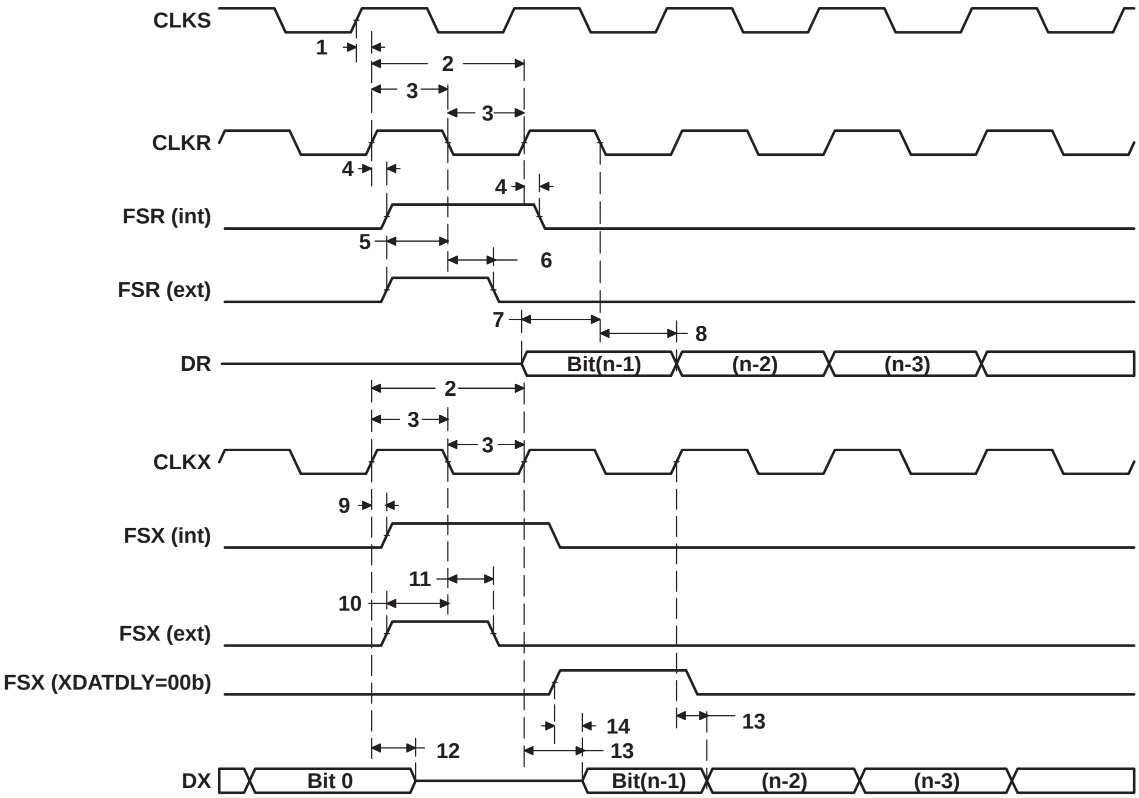


Figure 31. McBSP Timings

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 32)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{su(FRH-CKSH)}$	Setup time, FSR high before CLKS high	4		ns
2	$t_{h(CKSH-FRH)}$	Hold time, FSR high after CLKS high	4		ns

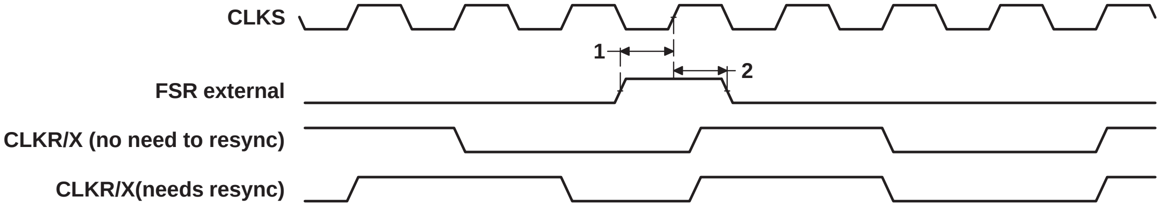


Figure 32. FSR Timing When GSYNC = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 33)

NO.			’C6701-150 ’C6701-167				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 6P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 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 33)

NO.	PARAMETER		’C6701-150 ’C6701-167				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T – 4	T + 4			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L – 4	L + 4			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–4	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	L – 2	L + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			P + 4	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 3	4P + 12	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.
‡ For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
§ S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
¶ FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP
# FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

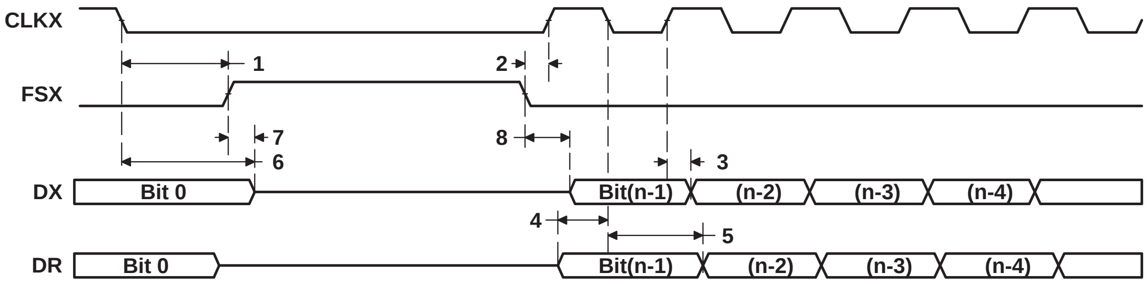


Figure 33. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0[†][‡] (see Figure 34)

NO.			’C6701-150 ’C6701-167				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 6P		ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 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 34)

NO.	PARAMETER		’C6701-150 ’C6701-167				UNIT
			MASTER[§]		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low[¶]	L – 4	L + 4			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high[#]	T – 4	T + 4			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–4	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	–2	4	3P + 4	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H – 2	H + 3	2P + 3	4P + 12	ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[‡] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

[§] S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

[¶] FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP

[#] FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

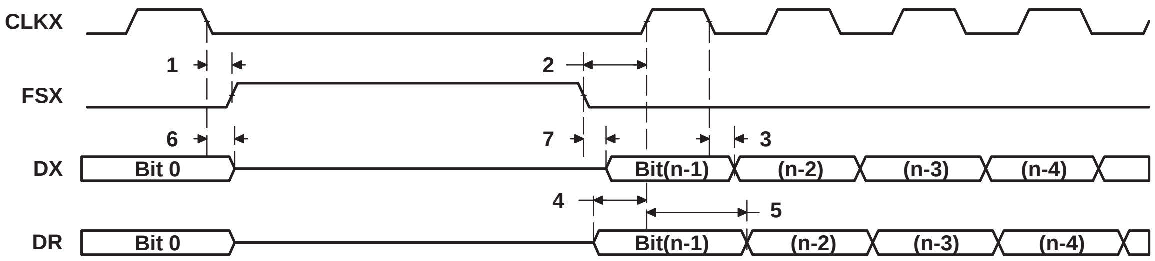


Figure 34. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1[†‡] (see Figure 35)

NO.			’C6701-150 ’C6701-167				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 3P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 6P		ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 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 35)

NO.	PARAMETER		’C6701-150 ’C6701-167				UNIT
			MASTER[§]		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high[¶]	T – 4	T + 4			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low[#]	H – 4	H + 4			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–4	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	H – 2	H + 3			ns
7	$t_{dis(FXH-DXHZ)}$	Disable time, DX high impedance following last data bit from FSX high			P + 4	3P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			2P + 3	4P + 12	ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

[‡] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.

[§] S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

[¶] FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP

[#] FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

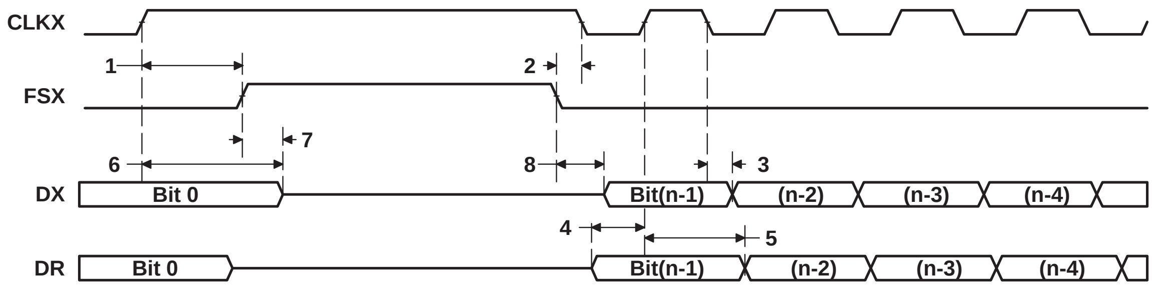


Figure 35. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1[†][‡] (see Figure 36)

NO.			’C6701-150 ’C6701-167				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 3P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 6P		ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 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 36)

NO.	PARAMETER		’C6701-150 ’C6701-167				UNIT
			MASTER[§]		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high[¶]	H – 4	H + 4			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low[#]	T – 4	T + 4			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–4	4	3P + 4	5P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	4	3P + 4	5P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L – 2	L + 3	2P + 3	4P + 12	ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.
[‡] For all SPI slave modes, CLKG is programmed as 1/2 of the CPU clock by setting CLKSM = CLKGDV = 1.
[§] S = sample rate generator input clock = P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
T = CLKX period = (1 + CLKGDV) * S
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero

[¶] FSRP = FSXP = 1. As a SPI master, FSX is inverted to provide active-low slave-enable output. As a slave, the active-low signal input on FSX and FSR is inverted before being used internally.
CLKXM = FSXM = 1, CLKRM = FSRM = 0 for master McBSP
CLKXM = CLKRM = FSXM = FSRM = 0 for slave McBSP

[#] FSX should be low before the rising edge of clock to enable slave devices and then begin a SPI transfer at the rising edge of the master clock (CLKX).

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

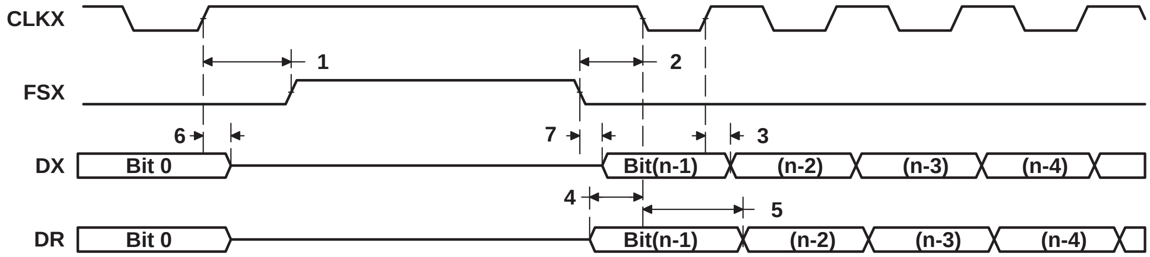


Figure 36. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

DMAC, TIMER, POWER-DOWN TIMING

switching characteristics for DMAC outputs (see Figure 37)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{d(CKO1H-DMACV)}$	Delay time, CLKOUT1 high to DMAC valid	2	11	ns

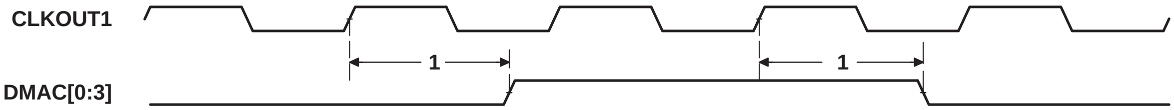


Figure 37. DMAC Timing

timing requirements for timer inputs (see Figure 38)[†]

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{w(TINPH)}$	Pulse duration, TINP high	2P		ns

[†] P = 1/CPU clock frequency in ns. For example, when running parts at 167 MHz, use P = 6 ns.

switching characteristics for timer outputs (see Figure 38)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
2	$t_{d(CKO1H-TOUTV)}$	Delay time, CLKOUT1 high to TOUT valid	1	10	ns

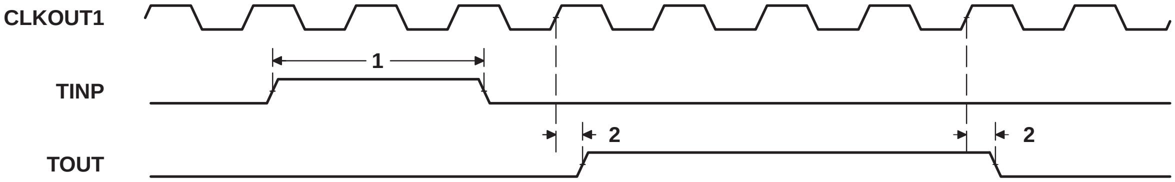


Figure 38. Timer Timing

switching characteristics for power-down outputs (see Figure 39)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{d(CKO1H-PDV)}$	Delay time, CLKOUT1 high to PD valid	1	9	ns

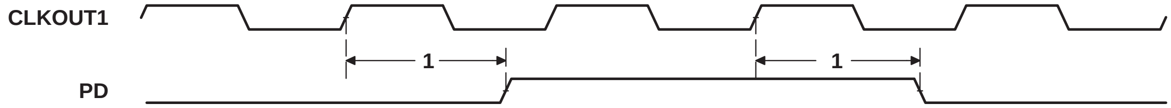


Figure 39. Power-Down Timing

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 40)

NO.			’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
1	$t_{c(TCK)}$	Cycle time, TCK		50	ns
3	$t_{su(TDIV\text{-}TCKH)}$	Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high	10		ns
4	$t_{h(TCKH\text{-}TDIV)}$	Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high	5		ns

switching characteristics for JTAG test port (see Figure 40)

NO.	PARAMETER		’C6701-150 ’C6701-167		UNIT
			MIN	MAX	
2	$t_{d(TCKL\text{-}TDOV)}$	Delay time, TCK low to TDO valid	0	15	ns

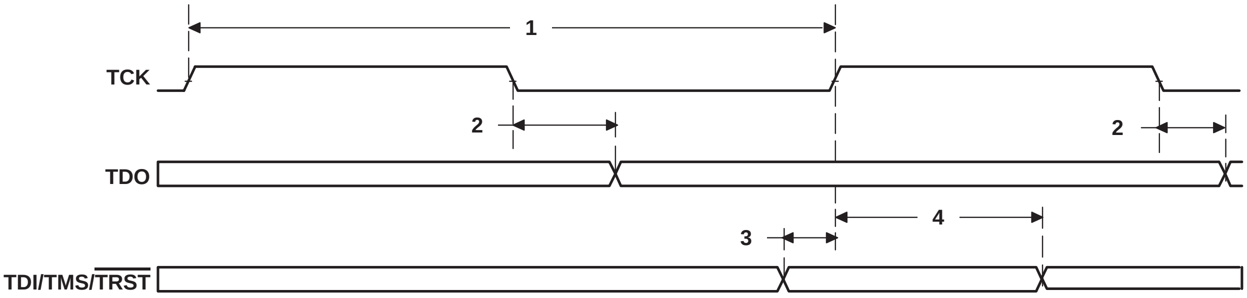
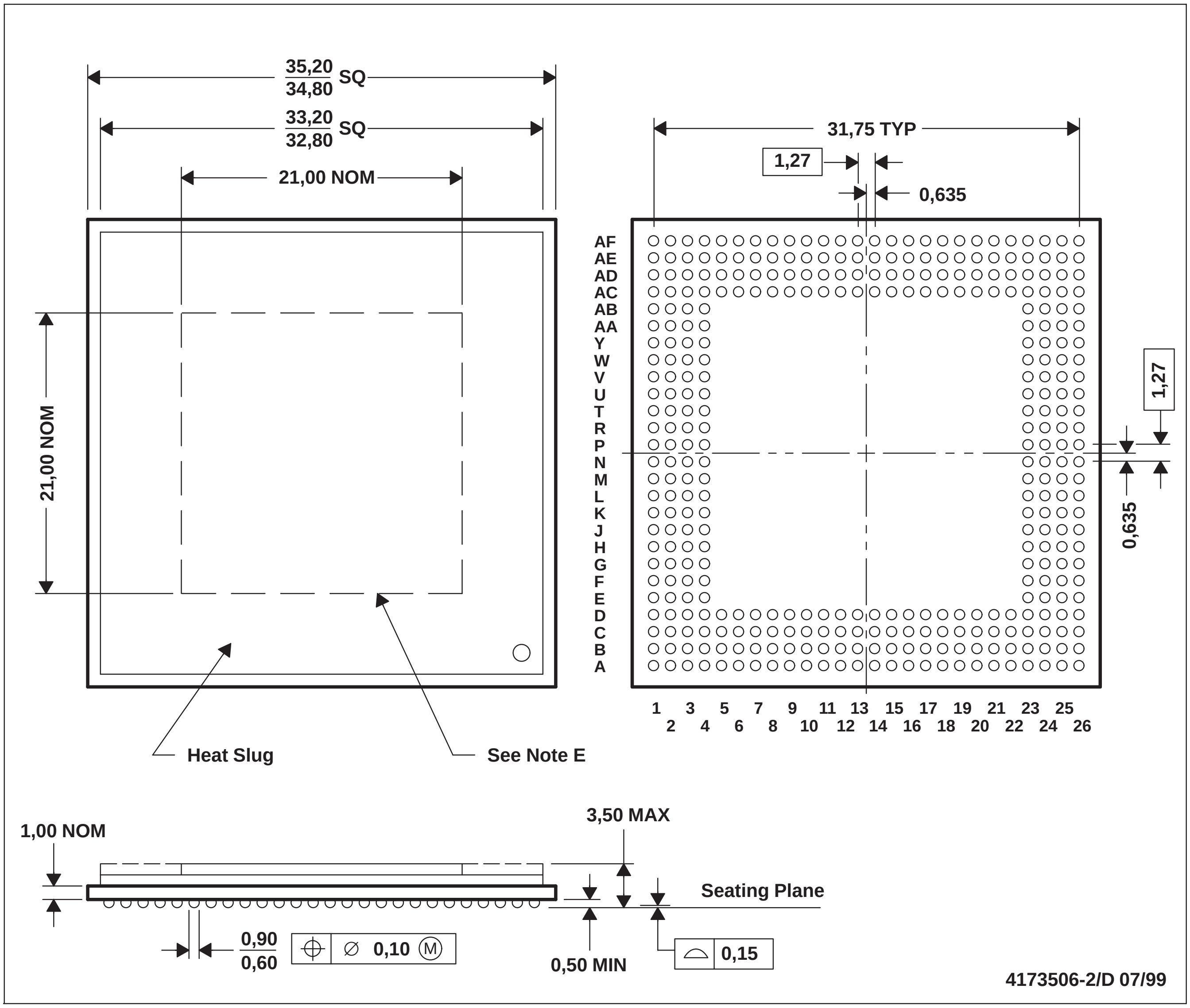


Figure 40. JTAG Test-Port Timing

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. Flip chip application only
E. Possible protrusion in this area, but within 3,50 max package height specification
F. Falls within JEDEC MO-151/BAR-2

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

IMPORTANT NOTICE

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

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

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

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

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

Copyright © 1999, Texas Instruments Incorporated