- **Best Price/Performance Digital Signal Processors (DSPs)**
  Fixed-Point: TMS320C6211
  Floating-Point: TMS320C6711
  - **10-, 6.7-ns Instruction Cycle Time**
  - **100-, 150-MHz Clock Rates**
  - **Eight 32-Bit Instructions/Cycle**
  - **1200 MIPS ('C6211)**
  - **900 MFLOPS ('C6711)**
  - **'C6211 and 'C6711 are Pin-Compatible**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) 'C6200 CPU Core ('C6211)**
  - **Eight Highly 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**
- **VelociTI Advanced Very Long Instruction Word (VLIW) 'C6700 CPU Core ('C6711)**
  - **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 ('C6711 Only)**
  - **Hardware Support for IEEE Double-Precision Instructions ('C6711 Only)**
  - **Byte-Addressable (8-, 16-, 32-Bit Data)**
  - **8-Bit Overflow Protection**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit-Counting**
  - **Normalization**
- **L1/L2 Memory Architecture**
  - **32K-Bit (4K-Byte) L1P Program Cache (Direct Mapped)**
  - **32K-Bit (4K-Byte) L1D Data Cache (2-Way Set-Associative)**
  - **512K-Bit (64K-Byte) L2 Unified Mapped RAM/Cache (Flexible Data/Program Allocation)**
  - **1024M-Byte Addressable External Memory Space**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Synchronous Memories: SDRAM and SBSRAM**
  - **Glueless Interface to Asynchronous Memories: SRAM and EPROM**
- **Enhanced Direct-Memory-Access (EDMA) Controller**
- **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**
- **256-Pin Ball Grid Array (BGA) Package (GFN Suffix)**
- **0.18-μm/5-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1.8-V Internal**

ADVANCE INFORMATION

† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

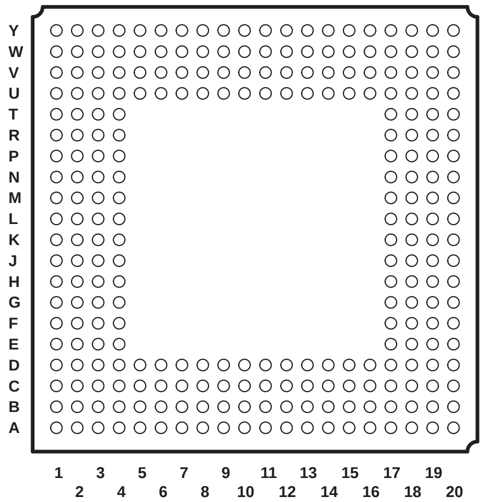
ADVANCE INFORMATION concerns new products in the sampling or preproduction phase of development. Characteristic data and other specifications are subject to change without notice.

Table of Contents

ADVANCE INFORMATION

GFN BGA package (bottom view)

GFN 256-PIN BALL GRID ARRAY (BGA) PACKAGE (BOTTOM VIEW)



description

The TMS320C62x™ DSPs (including the TMS320C6211 device) are the fixed-point DSP family in the TMS320C6000™ platform. The TMS320C67x™ DSPs (including the TMS320C6711 device) are the floating-point DSP family in the TMS320C6000 platform. The TMS320C6211 (’C6211) and TMS320C6711 (’C6711) devices are based on the high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making these DSPs an excellent choice for multichannel and multifunction applications.

With performance of up to 1200 million instructions per second (MIPS) at a clock rate of 150 MHz, the ’C6211 device offers cost-effective solutions to high-performance DSP programming challenges. The ’C6211 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 six arithmetic logic units (ALUs) for a high degree of parallelism and two 16-bit multipliers for a 32-bit result. The ’C6211 can produce two multiply-accumulates (MACs) per cycle for a total of 300 million MACs per second (MMACS).

With performance of up to 900 million floating-point operations per second (MFLOPS) at a clock rate of 150 MHz, the ’C6711 device also offers cost-effective solutions to high-performance DSP programming challenges. The 100-MHz device is the lowest-cost DSP in the ’C6000 family. The ’C6711 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 ’C6711 can produce two MACs per cycle for a total of 300 MMACS.

Both the ’C6211 and ’C6711 DSPs have the same application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The ’C6211/’C6711 uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 32-Kbit direct mapped cache and the Level 1 data cache (L1D) is a 32-Kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 512-Kbit memory space that is shared between program and data space. L2 memory can be configured as mapped memory, cache, or combinations of the two.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, SBSRAM and asynchronous peripherals.

The ’C6211/’C6711 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.

ADVANCE INFORMATION

TMS320C62x, TMS320C6000, and TMS320C67x are trademarks of Texas Instruments.
Windows is a registered trademark of the Microsoft Corporation.

device characteristics

Table 1 provides an overview of the ’C6211/’C6711 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 ’C6211/’C6711 Processors

HARDWARE FEATURES		’C6211 (FIXED-POINT DSP)	’C6711 (FLOATING-POINT DSP)
Peripherals	EMIF	1	1
	EDMA	1	1
	HPI	1	1
	McBSPs	2	2
	32-Bit Timers	2	2
On-Chip Memory	Size (Bytes)	72K	72K
	Organization	4K-Byte (4KB) L1 Program (L1P) Cache 4KB L1 Data (L1D) Cache 64KB Unified Mapped RAM/Cache (L2)	4K-Byte (4KB) L1 Program (L1P) Cache 4KB L1 Data (L1D) Cache 64KB Unified Mapped RAM/Cache (L2)
Frequency	MHz	150	150, 100
Cycle Time	ns	6.7 ns (’6211-150)	6.7 ns (’6711-150) 10 ns (’6711-100 [Lowest-Cost Device])
Voltage	Core (V)	1.8	1.8
	I/O (V)	3.3	3.3
PLL Options	CLKIN frequency multiplier	Bypass (x1), x4	Bypass (x1), x4
BGA Package	27 x 27 mm	256-Pin BGA (GFN)	256-Pin BGA (GFN)
Process Technology	µm	0.18 µm	0.18 µm
Product Status	Product Preview (PP) Advance Information (AI) Production Data (PD)	AI	AI

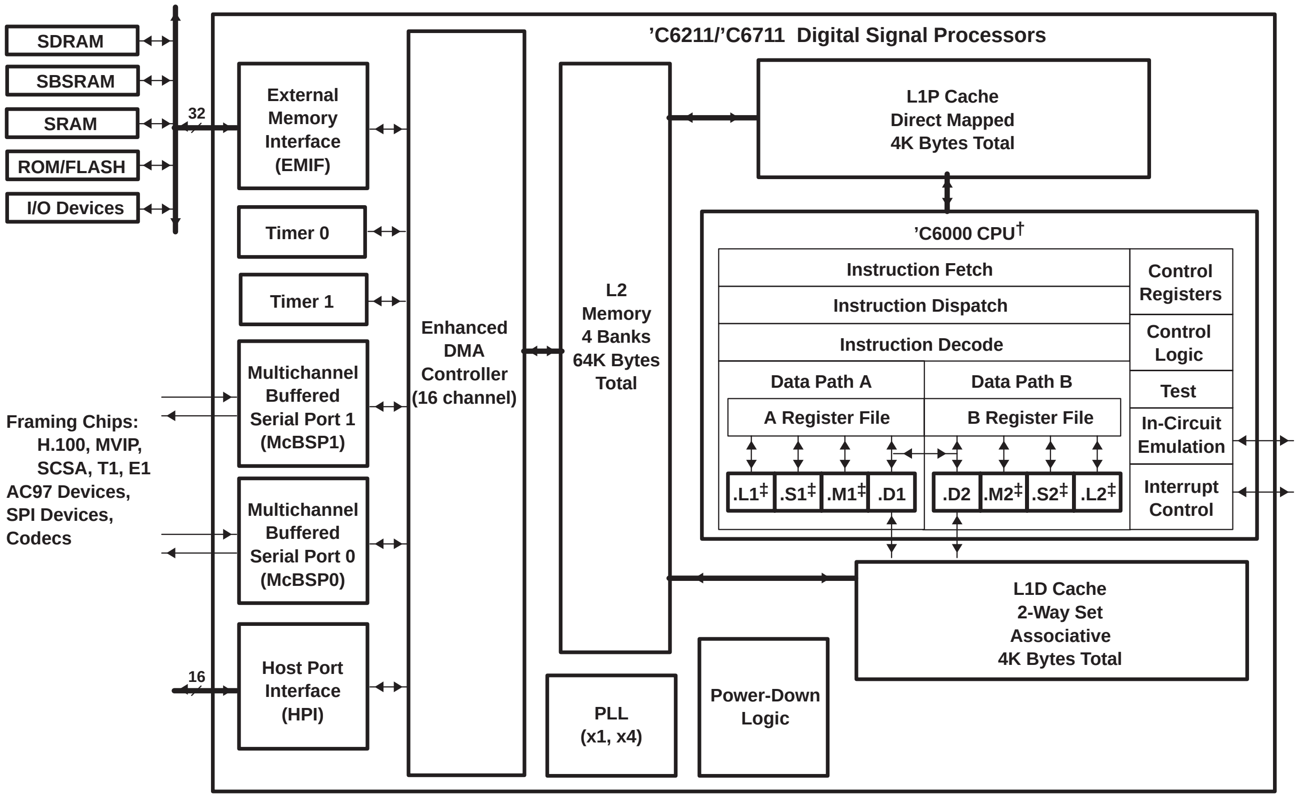
device compatibility

The TMS320C6211 and ’C6711 devices are pin-compatible and have the same peripheral set; thus, making new system designs easier and providing faster time to market. The following list summarizes the device characteristic differences between the ’C6211 and ’C6711 devices:

- The ’C6211 device has a fixed-point ’C62x CPU, while the ’C6711 device has a floating-point ’C67x CPU.
- A 100-MHz version of the ’C6711 is available, providing the lowest-cost entry in the TMS320C6000™ platform.

For a more detailed discussion on the similarities/differences between the ’C6211 and ’C6711 devices, see the *How to Begin Development Today with the TMS320C6211 DSP* and *How to Begin Development Today with the TMS320C6711 DSP* application reports (literature number SPRA474 and SPRA522, respectively).

functional block and CPU diagram



† The ’C6211 device has a fixed-point ’C62x CPU, while the ’C6711 device has a floating-point ’C67x CPU.
‡ For the ’C6711 device only, in addition to fixed-point instructions, these functional units execute floating-point instructions.

ADVANCE INFORMATION

CPU description

The CPU fetches VelociTI advanced very-long instruction words (VLIW) (256 bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. The VelociTI VLIW architecture features controls by which all eight units do not have to be supplied with instructions if they are not ready to execute. The first bit of every 32-bit instruction determines if the next instruction belongs to the same execute packet as the previous instruction, or whether it should be executed in the following clock as a part of the next execute packet. Fetch packets are always 256 bits wide; however, the execute packets can vary in size. The variable-length execute packets are a key memory-saving feature, distinguishing the ’C62x and ’C67x CPUs from other VLIW architectures.

The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files each contain 16 32-bit registers for a total of 32 general-purpose registers. The two sets of functional units, along with two register files, compose sides A and B of the CPU (see the functional block and CPU diagram and Figure 1 for the ’C6211 device; and see the functional block and CPU diagram and Figure 2 for the ’C6711 device). 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.

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 ’C62x/’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 ’C62x/’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)

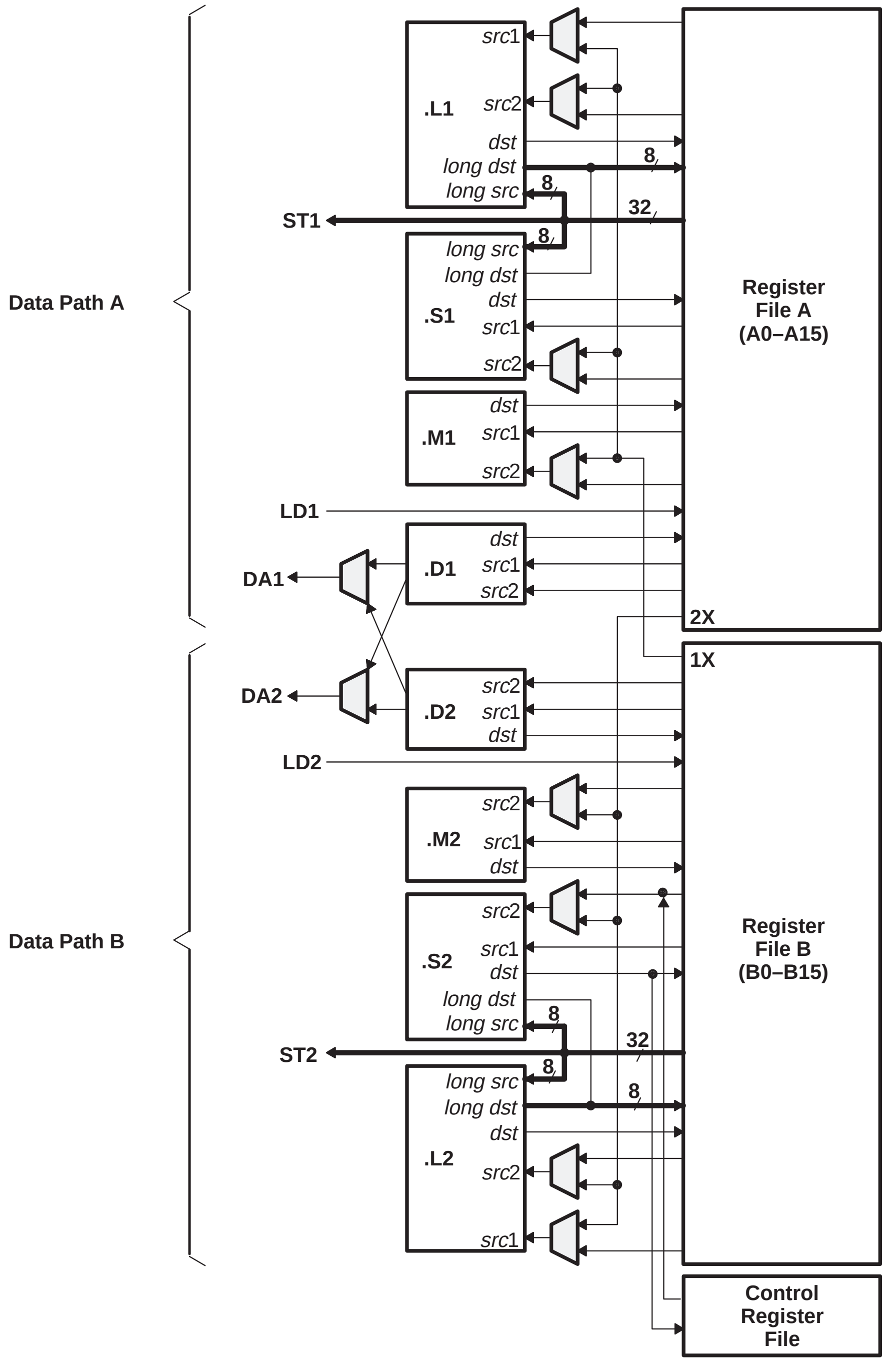
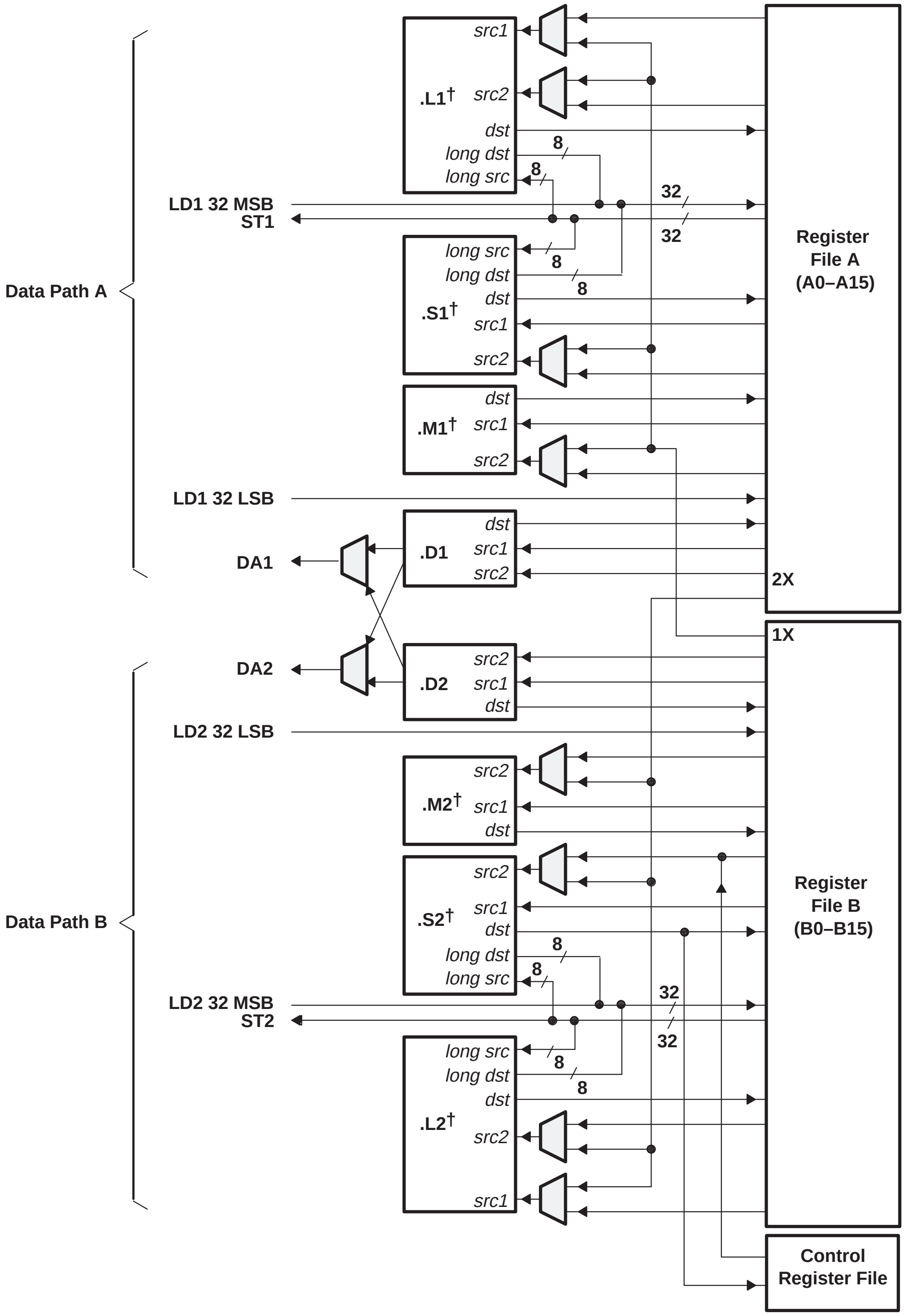


Figure 1. TMS320C62x CPU Data Paths

ADVANCE INFORMATION

CPU description (continued)



† For the ’C6711 device only, in addition to fixed-point instructions, these functional units execute floating-point instructions.

Figure 2. TMS320C67x CPU Data Paths

signal groups description

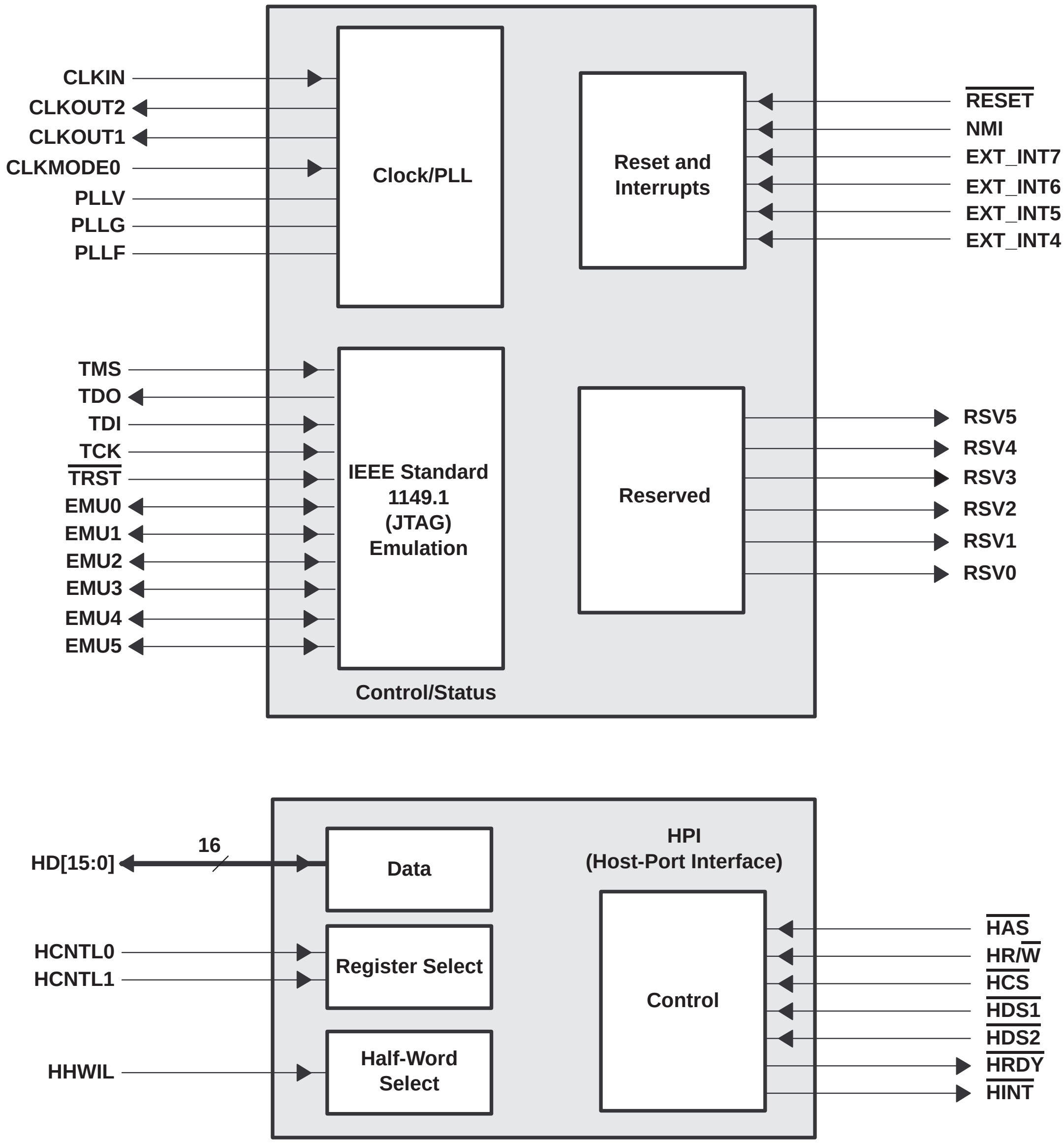


Figure 3. CPU and Peripheral Signals

ADVANCE INFORMATION

signal groups description (continued)

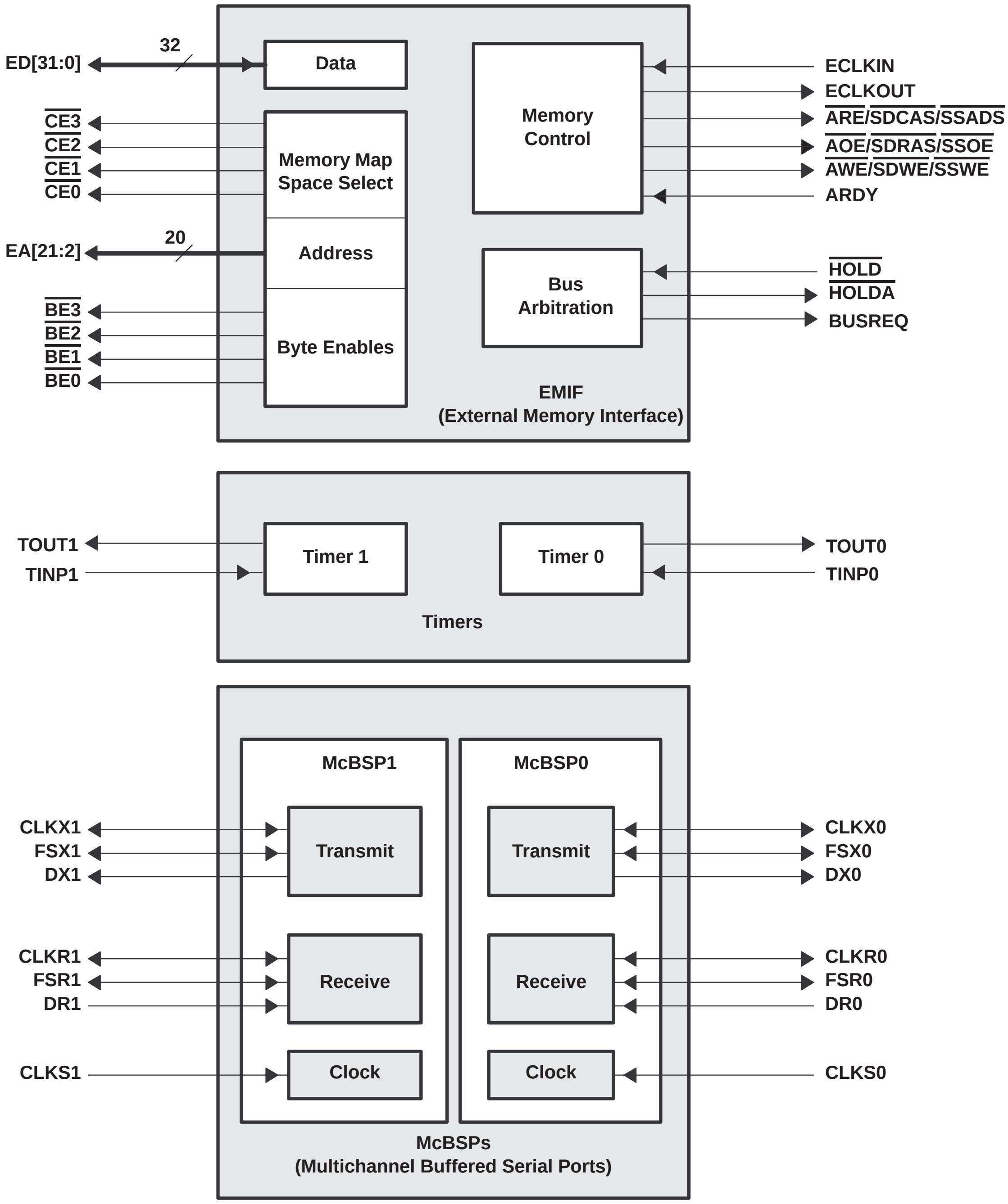


Figure 4. Peripheral Signals

ADVANCE INFORMATION

Signal Descriptions

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
CLOCK/PLL				
CLKIN	A3	I	IPU	Clock Input
CLKOUT1	D7	O	IPD	Clock output at device speed
CLKOUT2	Y12	O	IPD	Clock output at half of device speed
CLKMODE0	C4	I	IPU	Clock mode select • Selects whether the CPU clock frequency = input clock frequency x4 or x1
PLLV§	A4	A¶		PLL analog V_{CC} connection for the low-pass filter
PLLG§	C6	A¶		PLL analog GND connection for the low-pass filter
PLLF	B5	A¶		PLL low-pass filter connection to external components and a bypass capacitor
JTAG EMULATION				
TMS	B7	I	IPU	JTAG test-port mode select
TDO	A8	O/Z	IPU	JTAG test-port data out
TDI	A7	I	IPU	JTAG test-port data in
TCK	A6	I	IPU	JTAG test-port clock
$\overline{\text{TRST}}$	B6	I	IPD	JTAG test-port reset
EMU5	B12	I/O/Z	IPU	Emulation pin 5. Reserved for future use, leave unconnected.
EMU4	C11	I/O/Z	IPU	Emulation pin 4. Reserved for future use, leave unconnected.
EMU3	B10	I/O/Z	IPU	Emulation pin 3. Reserved for future use, leave unconnected.
EMU2	D10	I/O/Z	IPU	Emulation pin 2. Reserved for future use, leave unconnected.
EMU1	B9	I/O/Z	IPU	Emulation pin 1#
EMU0	D9	I/O/Z	IPU	Emulation pin 0#
RESETS AND INTERRUPTS				
$\overline{\text{RESET}}$	A13	I	IPU	Device reset
NMI	C13	I	IPD	Nonmaskable interrupt • Edge-driven (rising edge)
EXT_INT7	E3	I	IPU	External interrupts • Edge-driven (rising edge)
EXT_INT6	D2			
EXT_INT5	C1			
EXT_INT4	C2			
HOST-PORT INTERFACE (HPI)				
$\overline{\text{HINT}}$	J20	O	IPU	Host interrupt (from DSP to host)
HCNTL1	G19	I	IPU	Host control – selects between control, address, or data registers
HCNTL0	G18	I	IPU	Host control – selects between control, address, or data registers
HHWIL	H20	I	IPU	Host half-word select – first or second half-word (not necessarily high or low order)
HR/$\overline{\text{W}}$	G20	I	IPU	Host read or write select

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = internal pullup. [These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a resistor in the range of 4.7 kΩ to 5.1 kΩ should be used.]
§ 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)
# The EMU0 and EMU1 pins are internally pulled up with 30-kΩ resistors; therefore, for emulation and normal operation, no external pullup/pulldown resistors are necessary. However, for boundary scan operation, pull down the EMU1 and EMU0 pins with an external dedicated resistor in the range of 4.7 kΩ to 5.1 kΩ.

ADVANCE INFORMATION

ADVANCE INFORMATION

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
HOST-PORT INTERFACE (HPI) (CONTINUED)				
HD15	B14	I/O/Z	IPU	Host-port data • Used for transfer of data, address, and control • Also controls initialization of DSP modes at reset via pullup/pulldown resistors – Device Endian Mode HD8: 0 – Big Endian 1 – Little Endian – Boot mode HD[4:3]: 00 – HPI boot 01 – 8-bit ROM boot with default timings 10 – 16-bit ROM boot with default timings 11 – 32-bit ROM boot with default timings
HD14	C14		IPU	
HD13	A15		IPU	
HD12	C15		IPU	
HD11	A16		IPU	
HD10	B16		IPU	
HD9	C16		IPU	
HD8	B17		IPU	
HD7	A18		IPU	
HD6	C17		IPU	
HD5	B18		IPU	
HD4	C19		IPD	
HD3	C20		IPU	
HD2	D18		IPU	
HD1	D20		IPU	
HD0	E20		IPU	
$\overline{\text{HAS}}$	E18	I	IPU	Host address strobe
$\overline{\text{HCS}}$	F20	I	IPU	Host chip select
$\overline{\text{HDS1}}$	E19	I	IPU	Host data strobe 1
$\overline{\text{HDS2}}$	F18	I	IPU	Host data strobe 2
$\overline{\text{HRDY}}$	H19	O	IPU	Host ready (from DSP to host)
EMIF – CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY				
$\overline{\text{CE3}}$	V6	O/Z	IPU	Memory space enables • Enabled by bits 28 through 31 of the word address • Only one asserted during any external data access
$\overline{\text{CE2}}$	W6	O/Z	IPU	
$\overline{\text{CE1}}$	W18	O/Z	IPU	
$\overline{\text{CE0}}$	V17	O/Z	IPU	
$\overline{\text{BE3}}$	V5	O/Z	IPU	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)
$\overline{\text{BE2}}$	Y4	O/Z	IPU	
$\overline{\text{BE1}}$	U19	O/Z	IPU	
$\overline{\text{BE0}}$	V20	O/Z	IPU	

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = internal pullup. [These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a resistor in the range of 4.7 kΩ to 5.1 kΩ should be used.]

Signal Descriptions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/IPU‡	DESCRIPTION
EMIF – BUS ARBITRATION				
$\overline{\text{HOLDA}}$	J18	O/Z	IPU	Hold-request-acknowledge to the host
$\overline{\text{HOLD}}$	J17	I	IPU	Hold request from the host
BUSREQ	J19	O/Z	IPU	Bus request output
EMIF – ASYNCHRONOUS/SYNCHRONOUS DRAM/SYNCHRONOUS BURST SRAM MEMORY CONTROL				
ECLKIN	Y11	I	IPD	EMIF input clock
ECLKOUT	Y10	O	IPD	EMIF output clock (based on ECLKIN)
$\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$	V11	O/Z	IPU	Asynchronous memory read enable/SDRAM column-address strobe/SBSRAM address strobe
$\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$	W10	O/Z	IPU	Asynchronous memory output enable/SDRAM row-address strobe/SBSRAM output enable
$\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$	V12	O/Z	IPU	Asynchronous memory write enable/SDRAM write enable/SBSRAM write enable
ARDY	Y5	I	IPU	Asynchronous memory ready input
EMIF – ADDRESS				
EA21	U18	O/Z	IPU	External address (word address)
EA20	Y18			
EA19	W17			
EA18	Y16			
EA17	V16			
EA16	Y15			
EA15	W15			
EA14	Y14			
EA13	W14			
EA12	V14			
EA11	W13			
EA10	V10			
EA9	Y9			
EA8	V9			
EA7	Y8			
EA6	W8			
EA5	V8			
EA4	W7			
EA3	V7			
EA2	Y6			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = internal pullup. [These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a resistor in the range of 4.7 kΩ to 5.1 kΩ should be used.]

ADVANCE INFORMATION

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
EMIF – DATA				
ED31	N3	I/O/Z	IPU	External data
ED30	P3			
ED29	P2			
ED28	P1			
ED27	R2			
ED26	R3			
ED25	T2			
ED24	T1			
ED23	U3			
ED22	U1			
ED21	U2			
ED20	V1			
ED19	V2			
ED18	Y3			
ED17	W4			
ED16	V4			
ED15	T19			
ED14	T20			
ED13	T18			
ED12	R20			
ED11	R19			
ED10	P20			
ED9	P18			
ED8	N20			
ED7	N19			
ED6	N18			
ED5	M20			
ED4	M19			
ED3	L19			
ED2	L18			
ED1	K19			
ED0	K18			

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = internal pullup. [These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a resistor in the range of 4.7 kΩ to 5.1 kΩ should be used.]

Signal Descriptions (Continued)

SIGNAL NAME	NO.	TYPE†	IPD/ IPU‡	DESCRIPTION
TIMERS				
TOUT1	F1	O	IPD	Timer 1 or general-purpose output
TINP1	F2	I	IPD	Timer 1 or general-purpose input
TOUT0	G1	O	IPD	Timer 0 or general-purpose output
TINP0	G2	I	IPD	Timer 0 or general-purpose input
MULTICHANNEL BUFFERED SERIAL PORT 1 (McBSP1)				
CLKS1	E1	I	IPD	External clock source (as opposed to internal)
CLKR1	M1	I/O/Z	IPD	Receive clock
CLKX1	L3	I/O/Z	IPD	Transmit clock
DR1	M2	I	IPU	Receive data
DX1	L2	O/Z	IPU	Transmit data
FSR1	M3	I/O/Z	IPD	Receive frame sync
FSX1	L1	I/O/Z	IPD	Transmit frame sync
MULTICHANNEL BUFFERED SERIAL PORT 0 (McBSP0)				
CLKS0	K3	I	IPD	External clock source (as opposed to internal)
CLKR0	H3	I/O/Z	IPD	Receive clock
CLKX0	G3	I/O/Z	IPD	Transmit clock
DR0	J1	I	IPU	Receive data
DX0	H2	O/Z	IPU	Transmit data
FSR0	J3	I/O/Z	IPD	Receive frame sync
FSX0	H1	I/O/Z	IPD	Transmit frame sync
RESERVED FOR TEST				
RSV0	C12	O	IPU	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV1	D12	O	IPU	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV2	A5	O	IPU	Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV3	D3	O		Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV4	N2	O		Reserved (leave unconnected, ***do not*** connect to power or ground)
RSV5	Y20	O		Reserved (leave unconnected, ***do not*** connect to power or ground)

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground
‡ IPD = Internal pulldown, IPU = internal pullup. [These IPD/IPU signal pins feature a 30-kΩ IPD or IPU resistor. To pull up a signal to the opposite supply rail, a resistor in the range of 4.7 kΩ to 5.1 kΩ should be used.]

ADVANCE INFORMATION

ADVANCE INFORMATION

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
SUPPLY VOLTAGE PINS			
DV_{DD}	A17, B3, B8, B13, C5, C10, D1, D16, D19, F3, H18, J2, M18, N1, R1, R18, T3, U5, U7, U12, U16, V13, V15, V19, W3, W9, W12, Y7, Y17	S	3.3-V supply voltage
CV_{DD}	A9, A10, A12, B2, B19, C3, C7, C18, D5, D6, D11, D14	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
SUPPLY VOLTAGE PINS (CONTINUED)			
CV_{DD}	D15	S	1.8-V supply voltage
	F4		
	F17		
	K1		
	K4		
	K17		
	L4		
	L17		
	L20		
	R4		
	R17		
	U6		
	U10		
	U11		
	U14		
	U15		
	V3		
	V18		
	W2		
	W19		
GROUND PINS			
V_{SS}	A1	GND	Ground pins
	A2		
	A11		
	A14		
	A19		
	A20		
	B1		
	B4		
	B11		
	B15		
	B20		
	C8		
	C9		
	D4		
	D8		
	D13		
	D17		
	E2		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

ADVANCE INFORMATION

Signal Descriptions (Continued)

SIGNAL NAME	SIGNAL NO.	TYPE†	DESCRIPTION
GROUND PINS (CONTINUED)			
V_{SS}	E4	GND	Ground pins
	E17		
	F19		
	G4		
	G17		
	H4		
	H17		
	J4		
	K2		
	K20		
	M4		
	M17		
	N4		
	N17		
	P4		
	P17		
	P19		
	T4		
	T17		
	U4		
	U8		
	U9		
	U13		
	U17		
	U20		
	W1		
	W5		
	W11		
	W16		
	W20		
	Y1		
	Y2		
	Y13		
	Y19		

† I = Input, O = Output, Z = High impedance, S = Supply voltage, GND = Ground

development support

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

The following products support development of ’C6000-based applications:

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

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

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

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

ADVANCE INFORMATION

development support (continued)

Table 2. TMS320C6000 Development-Support Tools

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

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

ADVANCE INFORMATION

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, GFN) and the device speed range in megahertz (for example, -150 is 150 MHz). Figure 5 provides a legend for reading the complete device name for any TMS320 family member.

ADVANCE INFORMATION

ADVANCE INFORMATION

device and development-support tool nomenclature (continued)

TMS 320 C 6211 GFN ( ) 150

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, 120 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
GLW = 340-pin plastic BGA
GHK = 288-pin plastic MicroStar BGA™

DEVICE
’1x DSP: 10, 14, 15, 16, 17
’2x DSP: 25, 26
’2xx DSP: 203, 204, 206, 209, 240
’3x DSP: 30, 31, 32
’4x DSP: 40, 44
’5x DSP: 50, 51, 52, 53, 56, 57
’54x DSP: 541, 542, 543, 545, 546, 548
’6x DSP: 6201, 6202, 6202B, 6203, 6204, 6205, 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 TMS320C6211 and TMS320C6711 devices)

MicroStar BGA is a trademark of Texas Instruments.

documentation support

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

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

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

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

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

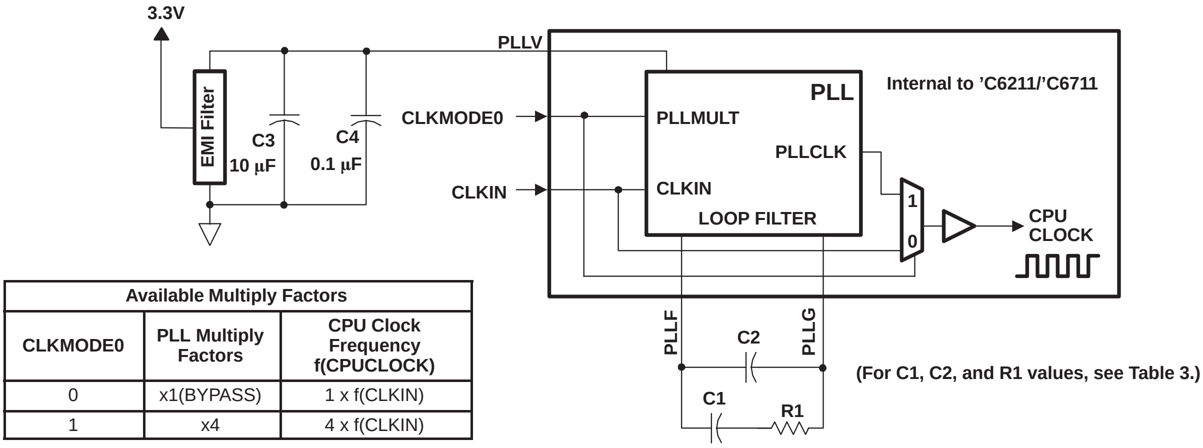
See the Worldwide Web URL for the application reports *How To Begin Development Today with the TMS320C6211 DSP* (literature number SPRA474) and *How To Begin Development Today with the TMS320C6711 DSP* (literature number SPRA522) which describe in more detail the similarities/differences between the 'C6211 and 'C6711 devices.

clock PLL

All of the internal ’C62x/’C67x clocks are generated from a single source through the CLKIN pin. This source clock either drives the PLL, which multiplies the source clock in frequency to generate the internal CPU clock, or bypasses the PLL to become the internal CPU clock.

To use the PLL to generate the CPU clock, the external PLL filter circuit must be properly designed. Figure 6 shows the external PLL circuitry for either x1 (PLL bypass) or x4 PLL multiply modes. Figure 7 shows the external PLL circuitry for a system with ONLY x1 (PLL bypass) mode.

To minimize the clock jitter, a single clean power supply should power both the ’C62x/’C67x device and the external clock oscillator circuit. Noise coupling into PLLF will directly impact PLL clock jitter. The minimum CLKIN rise and fall times should also be observed. For the input clock timing requirements, see the *input and output clocks* electricals section.



Available Multiply Factors		
CLKMODE0	PLL Multiply Factors	CPU Clock Frequency f(CPUCLOCK)
0	x1(BYPASS)	1 x f(CLKIN)
1	x4	4 x f(CLKIN)

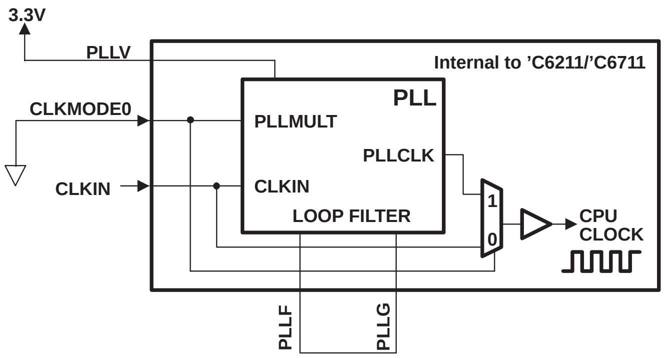
NOTES: A. Keep the lead length and the number of vias between the PLLF pin, the PLLG pin, and R1, C1, and C2 to a minimum. In addition, place all PLL external components (R1, C1, C2, C3, C4, and the EMI Filter) as close to the ’C6000 device as possible. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than the ones shown.

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

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

D. EMI filter manufacturer: TDK part number ACF451832-333, 223, 153, 103. Panasonic part number EXCCET103U.

Figure 6. External PLL Circuitry for Either PLL x4 Mode or x1 (Bypass) Mode



NOTES: A. For a system with ONLY PLL x1 (bypass) mode, short the PLLF terminal to the PLLG terminal.

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

Figure 7. External PLL Circuitry for x1 (Bypass) Mode Only

clock PLL (continued)

Table 3. ’C6211/’C6711 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	16.3–37.5	65–150	32.5–75	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.

ADVANCE INFORMATION

ADVANCE INFORMATION

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, Core‡		1.71	1.8	1.89	V
DV_{DD}	Supply voltage, I/O‡		3.14	3.30	3.46	V
V_{SS}	Supply ground		0	0	0	V
V_{IH}	High-level input voltage		2.0			V
V_{IL}	Low-level input voltage				0.8	V
I_{OH}	High-level output current	All signals except CLKOUT1 and CLKOUT2			–4	mA
		CLKOUT1 and CLKOUT2			–8	mA
I_{OL}	Low-level output current	All signals except CLKOUT1 and CLKOUT2			4	mA
		CLKOUT1 and CLKOUT2			8	mA
T_C	Operating case temperature		0		90	°C

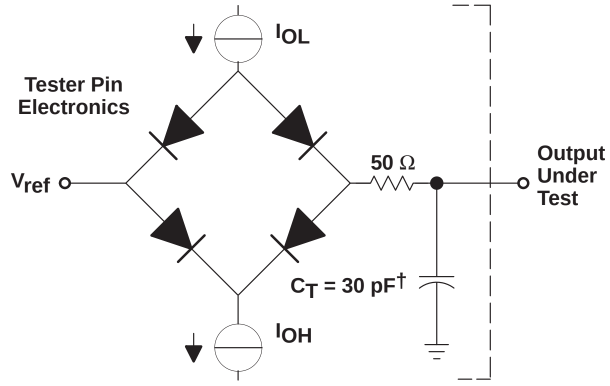
‡ TI DSP’s do not require specific power sequencing between the core supply and the I/O supply. However, systems should be designed to ensure that neither supply is powered up for extended periods of time if the other supply is below the proper operating voltage. Excessive exposure to these conditions can adversely affect the long term reliability of the device. System-level concerns such as bus contention may require supply sequencing to be implemented. In this case, the core supply should be powered up at the same time as, or prior to (and powered down after), the I/O buffers. For additional power supply sequencing information, see the *Power Supply Sequencing Solutions For Dual Supply Voltage DSPs* application report (literature number SLVA073).

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}			±125	uA
I_{OZ}	Off-state output current	$V_O = DV_{DD}$ or 0 V			±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access§	’C6211, CV_{DD} = NOM, CPU clock = 150 MHz		270		mA
		’C6711, CV_{DD} = NOM, CPU clock = 150 MHz		TBD		mA
I_{DD2V}	Supply current, peripherals§	’C6211, CV_{DD} = NOM, CPU clock = 150 MHz		220		mA
		’C6711, CV_{DD} = NOM, CPU clock = 150 MHz		TBD		mA
I_{DD3V}	Supply current, I/O pins§	’C6211, DV_{DD} = NOM, CPU clock = 150 MHz		60		mA
		’C6711, DV_{DD} = NOM, CPU clock = 150 MHz		TBD		mA
C_i	Input capacitance				5	pF
C_o	Output capacitance				5	pF

§ Measured with average activity (50% high/50% low power). For more details on CPU, peripheral, and I/O activity, refer to the *TMS320C6000 Power Consumption Summary* application report (literature number SPRA486).

PARAMETER MEASUREMENT INFORMATION



† Typical distributed load circuit capacitance

Figure 8. Test Load Circuit

signal transition levels

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

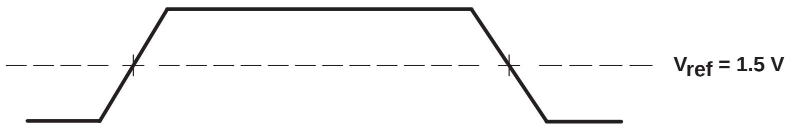


Figure 9. Input and Output Voltage Reference Levels for ac Timing Measurements

ADVANCE INFORMATION

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN†‡ (see Figure 10)

NO.			–100				–150				UNIT
			CLKMODE = x4		CLKMODE = x1		CLKMODE = x4		CLKMODE = x1		
			MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, CLKIN	40		10		26.7		6.7		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	0.4C		0.45C		0.4C		0.45C		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	0.4C		0.45C		0.4C		0.45C		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}.
‡ C = CLKIN cycle time in ns. For example, when CLKIN frequency is 10 MHz, use C = 100 ns.

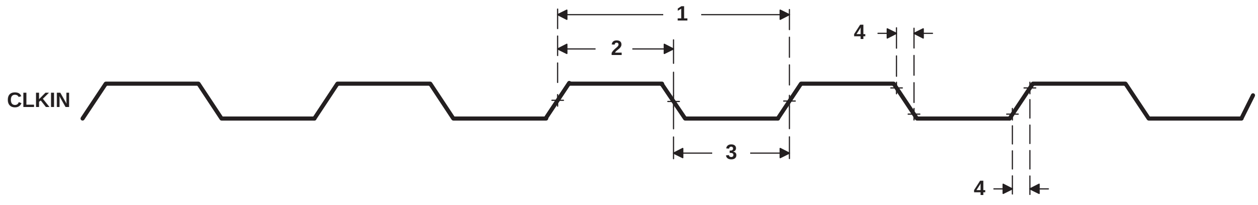


Figure 10. CLKIN Timings

switching characteristics for CLKOUT1§¶ (see Figure 11)

NO.	PARAMETER		–100 –150				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 CLKIN in ns and PL is the low period of CLKIN in ns.
¶ P = 1/CPU clock frequency in nanoseconds (ns)

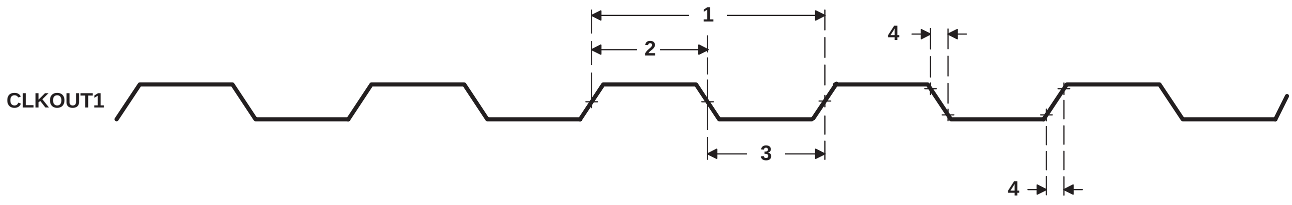


Figure 11. CLKOUT1 Timings

ADVANCE INFORMATION

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT2† (see Figure 12)

NO.	PARAMETER		–100 –150 MIN	MAX	UNIT
1	$t_{c(CKO2)}$	Cycle time, CLKOUT2	2P – 0.7	2P + 0.7	ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	P – 0.7	P + 0.7	ns
3	$t_{w(CKO2L)}$	Pulse duration, CLKOUT2 low	P – 0.7	P + 0.7	ns
4	$t_{t(CKO2)}$	Transition time, CLKOUT2		0.6	ns

† P = 1/CPU clock frequency in ns

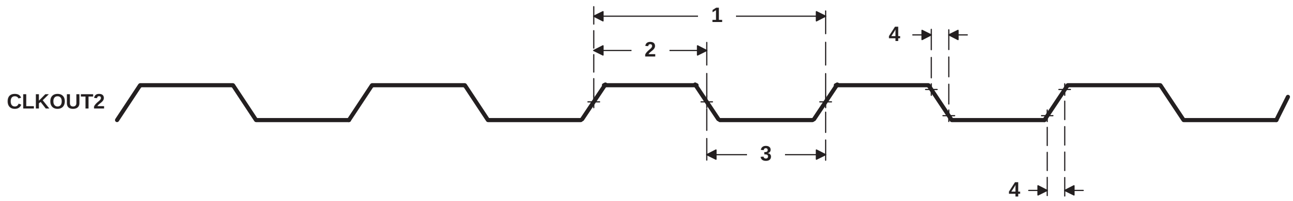


Figure 12. CLKOUT2 Timings

timing requirements for ECLKIN‡ (see Figure 13)

NO.			–100 MIN	–100 MAX	–150 MIN	–150 MAX	UNIT
1	$t_{c(EKI)}$	Cycle time, ECLKIN	15		10		ns
2	$t_{w(EKIH)}$	Pulse duration, ECLKIN high	6.8		4.5		ns
3	$t_{w(EKIL)}$	Pulse duration, ECLKIN low	6.8		4.5		ns
4	$t_{t(EKI)}$	Transition time, ECLKIN		3		3	ns

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

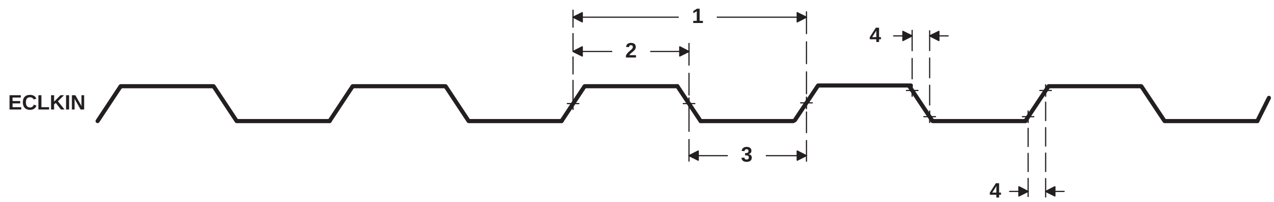


Figure 13. ECLKIN Timings

ADVANCE INFORMATION

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for ECLKOUT†‡§ (see Figure 14)

NO.	PARAMETER		–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{c(EKO)}$	Cycle time, ECLKOUT	E – 0.7	E + 0.7	ns
2	$t_{w(EKOH)}$	Pulse duration, ECLKOUT high	EH – 0.7	EH + 0.7	ns
3	$t_{w(EKOL)}$	Pulse duration, ECLKOUT low	EL – 0.7	EL + 0.7	ns
4	$t_{t(EKO)}$	Transition time, ECLKOUT		0.6	ns
5	$t_{d(EKIH-EKOH)}$	Delay time, ECLKIN high to ECLKOUT high	1	3	ns
6	$t_{d(EKIL-EKOL)}$	Delay time, ECLKIN low to ECLKOUT low	1	3	ns

† The reference points for the rise and fall transitions are measured at 20% and 80%, respectively, of V_{IH}.
‡ E = ECLKIN period in ns
§ EH is the high period of ECLKIN in ns and EL is the low period of ECLKIN in ns.

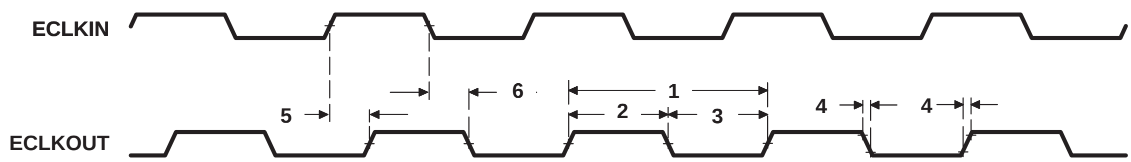


Figure 14. ECLKOUT Timings

ADVANCE INFORMATION

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles†‡§ (see Figure 15–Figure 16)

NO.			–100		–150		UNIT
			MIN	MAX	MIN	MAX	
3	$t_{su(EDV-AREH)}$	Setup time, EDx valid before $\overline{\text{ARE}}$ high	6		3		ns
4	$t_{h(AREH-EDV)}$	Hold time, EDx valid after $\overline{\text{ARE}}$ high	1		1		ns
6	$t_{su(ARDY-EKOH)}$	Setup time, ARDY valid before ECLKOUT high	6		2		ns
7	$t_{h(EKOH-ARDY)}$	Hold time, ARDY valid after ECLKOUT high	1		1		ns

† To ensure data setup time, simply program the strobe width wide enough. ARDY is internally synchronized. The ARDY signal is recognized in the cycle for which the setup and hold time is met. To use ARDY as an asynchronous input, the pulse width of the ARDY signal should be wide enough (e.g., pulse width = 2E) to ensure setup and hold time is met.
‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the EMIF CE space control registers.
§ E = ECLKOUT period in ns

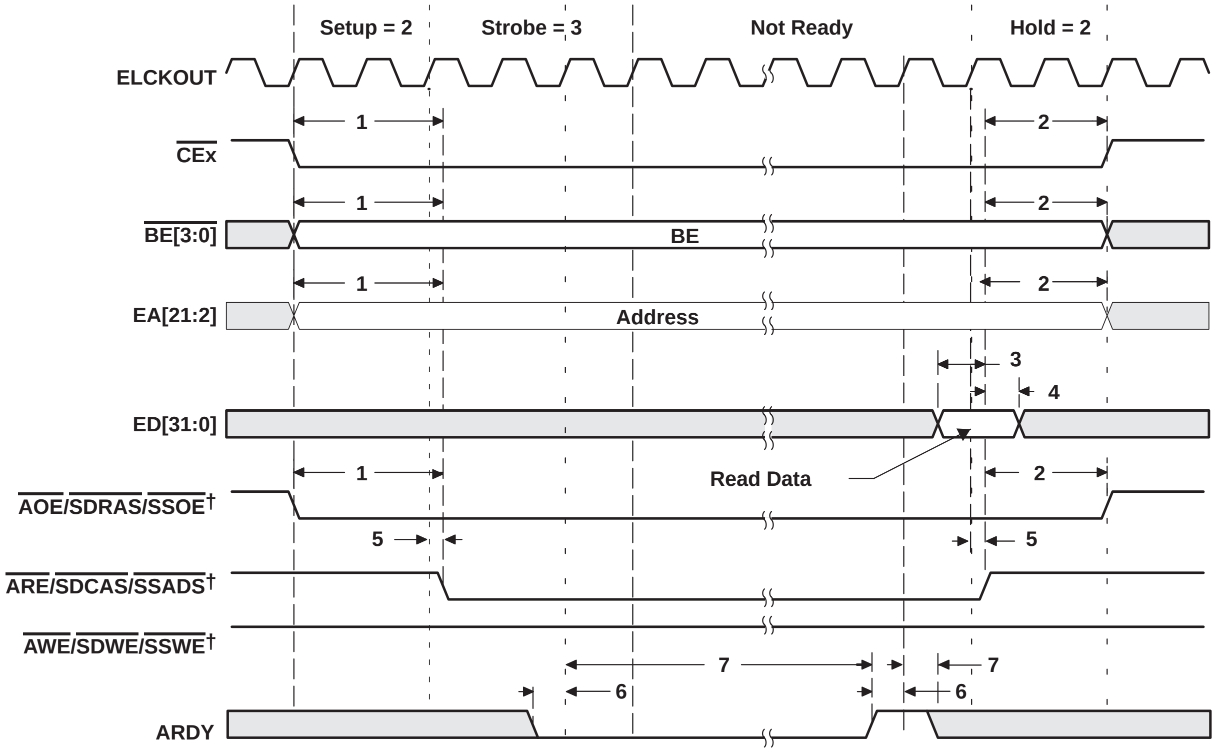
switching characteristics for asynchronous memory cycles‡§¶ (see Figure 15–Figure 16)

NO.	PARAMETER		–100		–150		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{osu(SELV-AREL)}$	Output setup time, select signals valid to $\overline{\text{ARE}}$ low	RS * E – 2		RS * E – 2		ns
2	$t_{oh(AREH-SELIV)}$	Output hold time, $\overline{\text{ARE}}$ high to select signals invalid	RH * E – 2		RH * E – 2		ns
5	$t_{d(EKOH-AREV)}$	Delay time, ECLKOUT high to $\overline{\text{ARE}}$ vaild	2	11	1.5	6	ns
8	$t_{osu(SELV-AWEL)}$	Output setup time, select signals valid to $\overline{\text{AWE}}$ low	WS * E – 2		WS * E – 2		ns
9	$t_{oh(AWEH-SELIV)}$	Output hold time, $\overline{\text{AWE}}$ high to select signals invalid	WH * E – 2		WH * E – 2		ns
10	$t_{d(EKOH-AWEV)}$	Delay time, ECLKOUT high to $\overline{\text{AWE}}$ vaild	2	11	1.5	6	ns

‡ RS = Read Setup, RST = Read Strobe, RH = Read Hold, WS = Write Setup, WST = Write Strobe, WH = Write Hold. These parameters are programmed via the EMIF CE space control registers.
§ E = ECLKOUT period in ns
¶ Select signals include: $\overline{\text{CEx}}$, $\overline{\text{BE[3:0]}}$, EA[21:2], $\overline{\text{AOE}}$; and for writes, include ED[31:0].

ADVANCE INFORMATION

ASYNCHRONOUS MEMORY TIMING (CONTINUED)

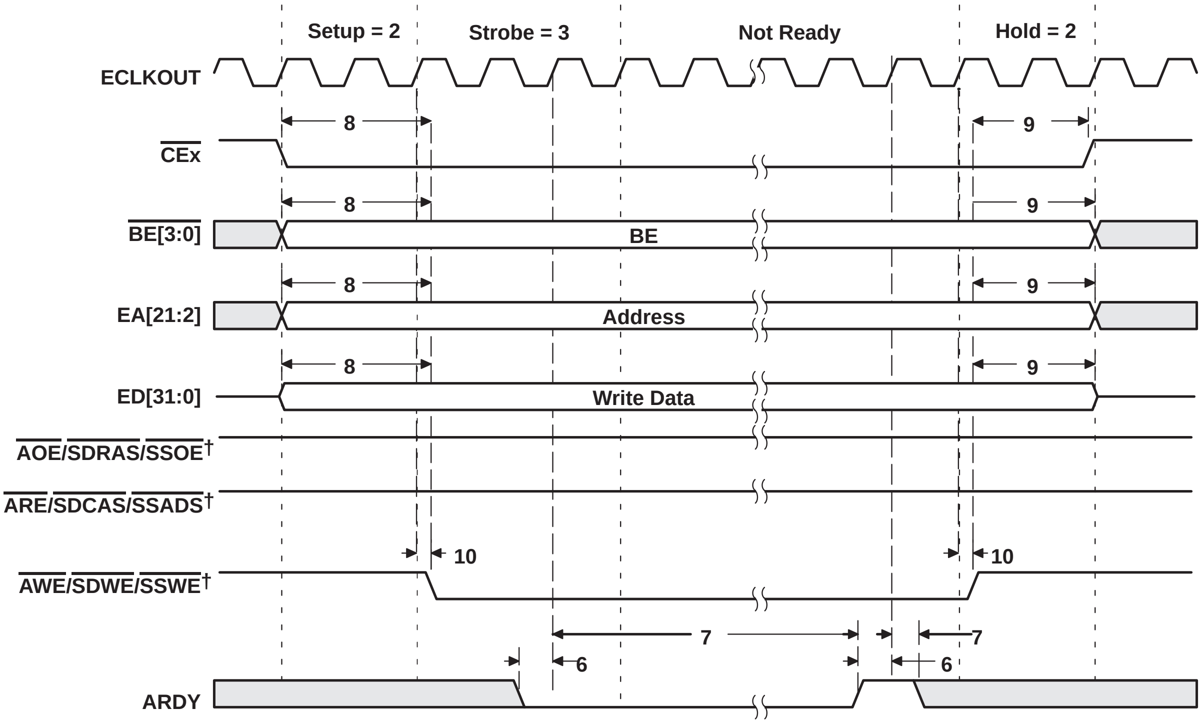


† AOE/SDRAS/SSOE, ARE/SDCAS/SSADS, and AWE/SDWE/SSWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

Figure 15. Asynchronous Memory Read Timing

ADVANCE INFORMATION

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



† AOE/SDRAS/SSOE, ARE/SDCAS/SSADS, and AWE/SDWE/SSWE operate as AOE (identified under select signals), ARE, and AWE, respectively, during asynchronous memory accesses.

Figure 16. Asynchronous Memory Write Timing

ADVANCE INFORMATION

SYNCHRONOUS-BURST MEMORY TIMING

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

NO.			–100		–150		UNIT
			MIN	MAX	MIN	MAX	
6	$t_{su(EDV-EKOH)}$	Setup time, read EDx valid before ECLKOUT high	6		1.5		ns
7	$t_{h(EKOH-EDV)}$	Hold time, read EDx valid after ECLKOUT high	1		1.0		ns

† The ’C6211/’C6711 SBSRAM interface takes advantage of the internal burst counter in the SBSRAM. Accesses default to incrementing 4-word bursts, but random bursts and decrementing bursts are done by interrupting bursts in progress. All burst types can sustain continuous data flow.

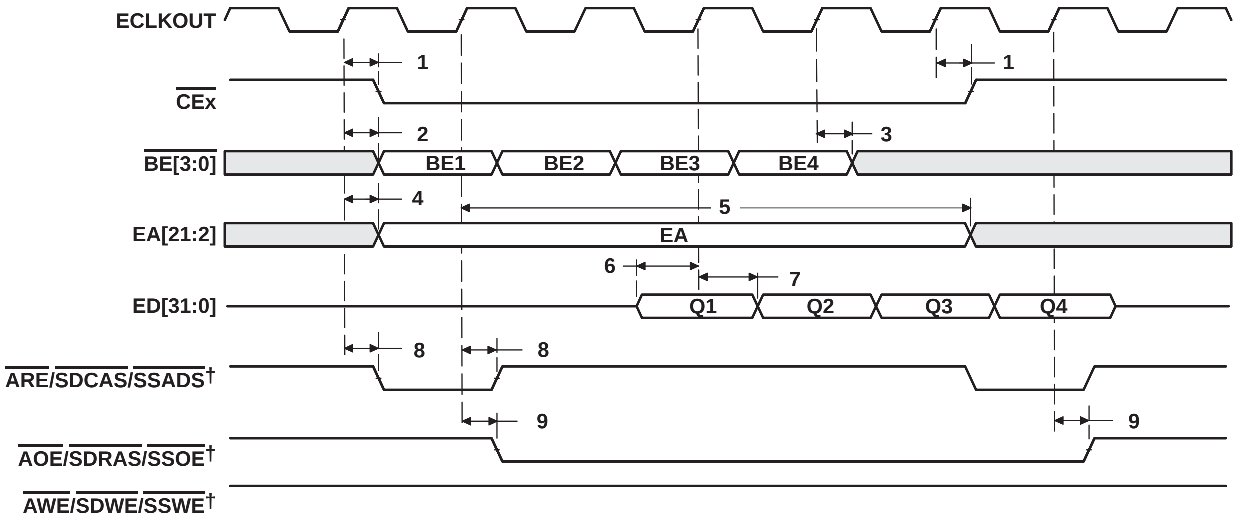
switching characteristics for synchronous-burst SRAM cycles†‡ (see Figure 17 and Figure 18)

NO.	PARAMETER		–100		–150		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(EKOH-CEV)}$	Delay time, ECLKOUT high to $\overline{\text{CE}}$x valid	2	11	1.5	6	ns
2	$t_{d(EKOH-BEV)}$	Delay time, ECLKOUT high to $\overline{\text{BE}}$x valid		11		6	ns
3	$t_{d(EKOH-BEIV)}$	Delay time, ECLKOUT high to $\overline{\text{BE}}$x invalid	2		1.5		ns
4	$t_{d(EKOH-EAV)}$	Delay time, ECLKOUT high to EAx valid		11		6	ns
5	$t_{d(EKOH-EAIV)}$	Delay time, ECLKOUT high to EAx invalid	2		1.5		ns
8	$t_{d(EKOH-ADSV)}$	Delay time, ECLKOUT high to $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid	2	11	1.5	6	ns
9	$t_{d(EKOH-OEV)}$	Delay time, ECLKOUT high to, $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid	2	11	1.5	6	ns
10	$t_{d(EKOH-EDV)}$	Delay time, ECLKOUT high to $\overline{\text{ED}}$x valid		11		6	ns
11	$t_{d(EKOH-EDIV)}$	Delay time, ECLKOUT high to $\overline{\text{ED}}$x invalid	2		1.5		ns
12	$t_{d(EKOH-WEV)}$	Delay time, ECLKOUT high to $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid	2	11	1.5	6	ns

† The ’C6211/’C6711 SBSRAM interface takes advantage of the internal burst counter in the SBSRAM. Accesses default to incrementing 4-word bursts, but random bursts and decrementing bursts are done by interrupting bursts in progress. All burst types can sustain continuous data flow.
‡ $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$, and $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ operate as $\overline{\text{SSADS}}$, $\overline{\text{SSOE}}$, and $\overline{\text{SSWE}}$, respectively, during SBSRAM accesses.

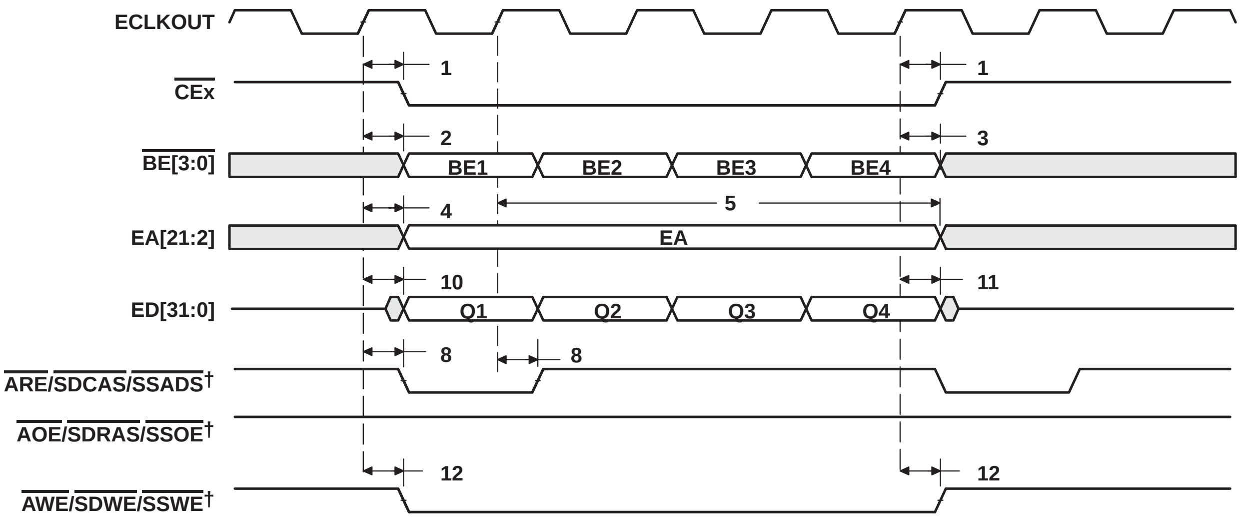
ADVANCE INFORMATION

SYNCHRONOUS-BURST MEMORY TIMING (CONTINUED)



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

Figure 17. SBSRAM Read Timing



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

Figure 18. SBSRAM Write Timing

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles† (see Figure 19)

NO.			–100		–150		UNIT
			MIN	MAX	MIN	MAX	
6	$t_{su(EDV-EKOH)}$	Setup time, read EDx valid before ECLKOUT high	6		1.5		ns
7	$t_{h(EKOH-EDV)}$	Hold time, read EDx valid after ECLKOUT high	1		1		ns

† The ’C6211/’C6711 SDRAM interface takes advantage of the internal burst counter in the SDRAM. Accesses default to incrementing 4-word bursts, but random bursts and decrementing bursts are done by interrupting bursts in progress. All burst types can sustain continuous data flow.

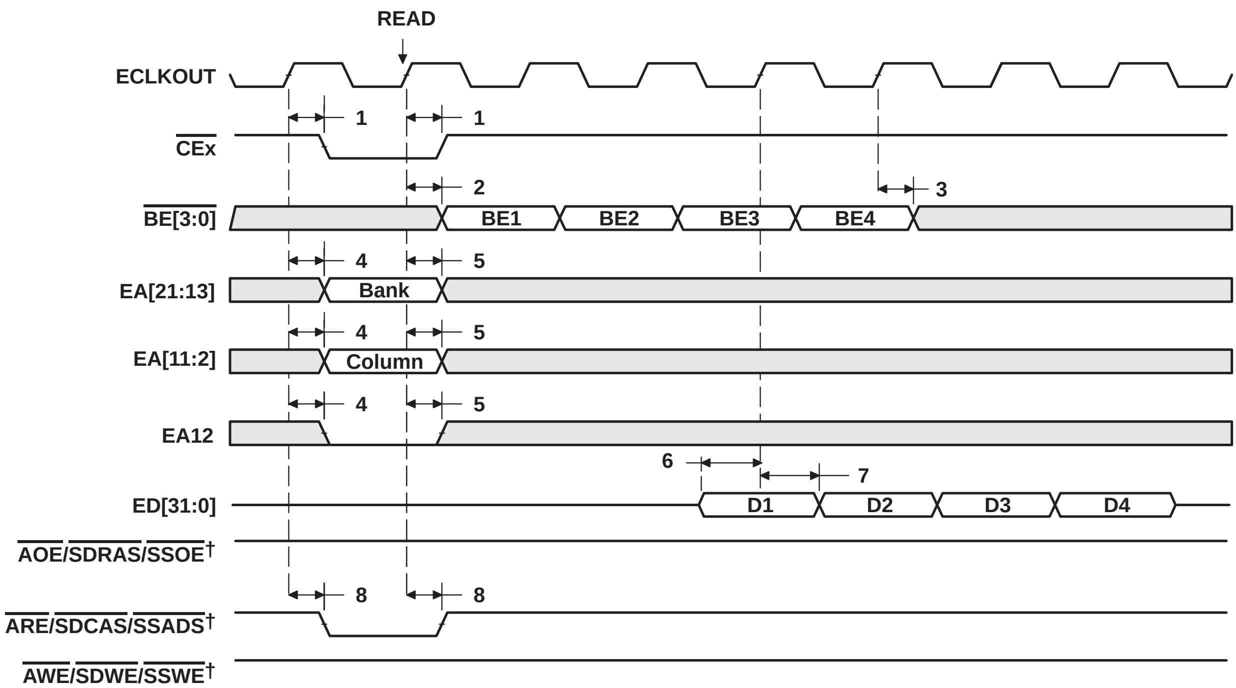
switching characteristics for synchronous DRAM cycles†‡ (see Figure 19–Figure 25)

NO.	PARAMETER		–100		–150		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{d(EKOH-CEV)}$	Delay time, ECLKOUT high to $\overline{\text{CE}}$x valid	2	11	1.5	6	ns
2	$t_{d(EKOH-BEV)}$	Delay time, ECLKOUT high to $\overline{\text{BE}}$x valid		11		6	ns
3	$t_{d(EKOH-BEIV)}$	Delay time, ECLKOUT high to $\overline{\text{BE}}$x invalid	2		1.5		ns
4	$t_{d(EKOH-EAV)}$	Delay time, ECLKOUT high to EAx valid		11		6	ns
5	$t_{d(EKOH-EAIV)}$	Delay time, ECLKOUT high to EAx invalid	2		1.5		ns
8	$t_{d(EKOH-CASV)}$	Delay time, ECLKOUT high to $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$ valid	2	11	1.5	6	ns
9	$t_{d(EKOH-EDV)}$	Delay time, ECLKOUT high to $\overline{\text{ED}}$x valid		11		6	ns
10	$t_{d(EKOH-EDIV)}$	Delay time, ECLKOUT high to $\overline{\text{ED}}$x invalid	2		1.5		ns
11	$t_{d(EKOH-WEV)}$	Delay time, ECLKOUT high to $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$ valid	2	11	1.5	6	ns
12	$t_{d(EKOH-RAS)}$	Delay time, ECLKOUT high to, $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ valid	2	11	1.5	6	ns

† The ’C6211/’C6711 SDRAM interface takes advantage of the internal burst counter in the SDRAM. Accesses default to incrementing 4-word bursts, but random bursts and decrementing bursts are done by interrupting bursts in progress. All burst types can sustain continuous data flow.
‡ $\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$, $\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$, and $\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$ operate as $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, and $\overline{\text{SDRAS}}$, respectively, during SDRAM accesses.

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)

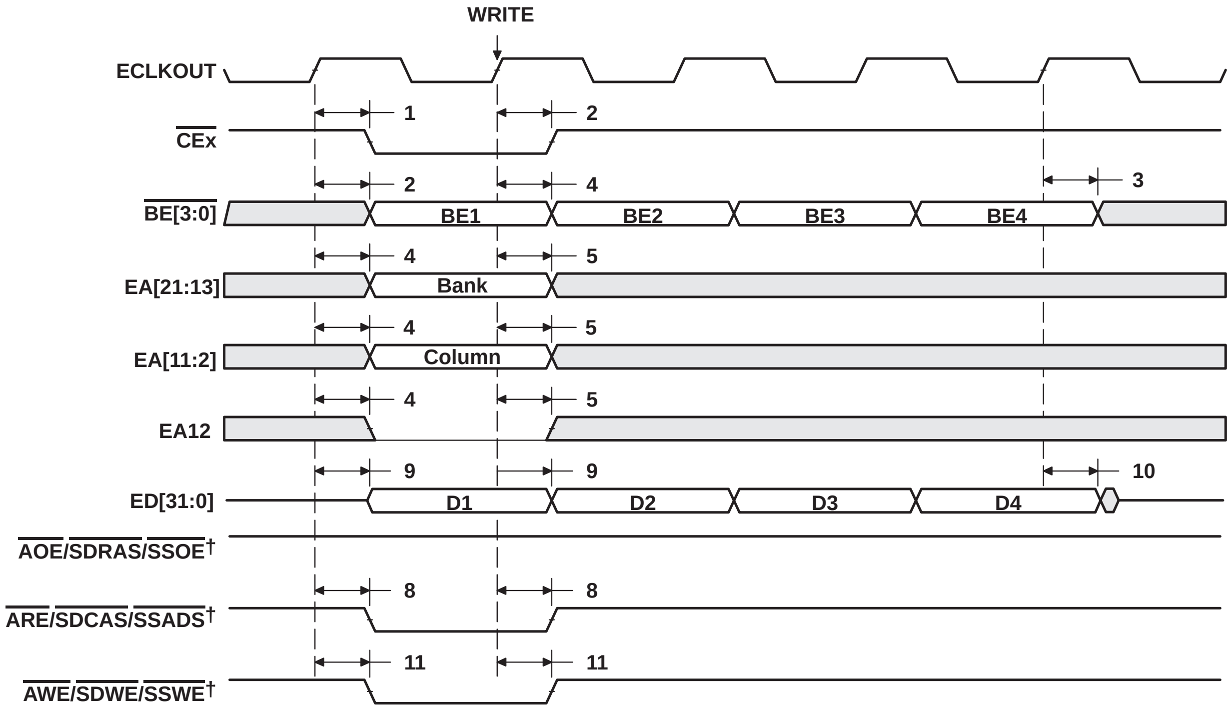


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 19. SDRAM Read Command (CAS Latency 3)

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)

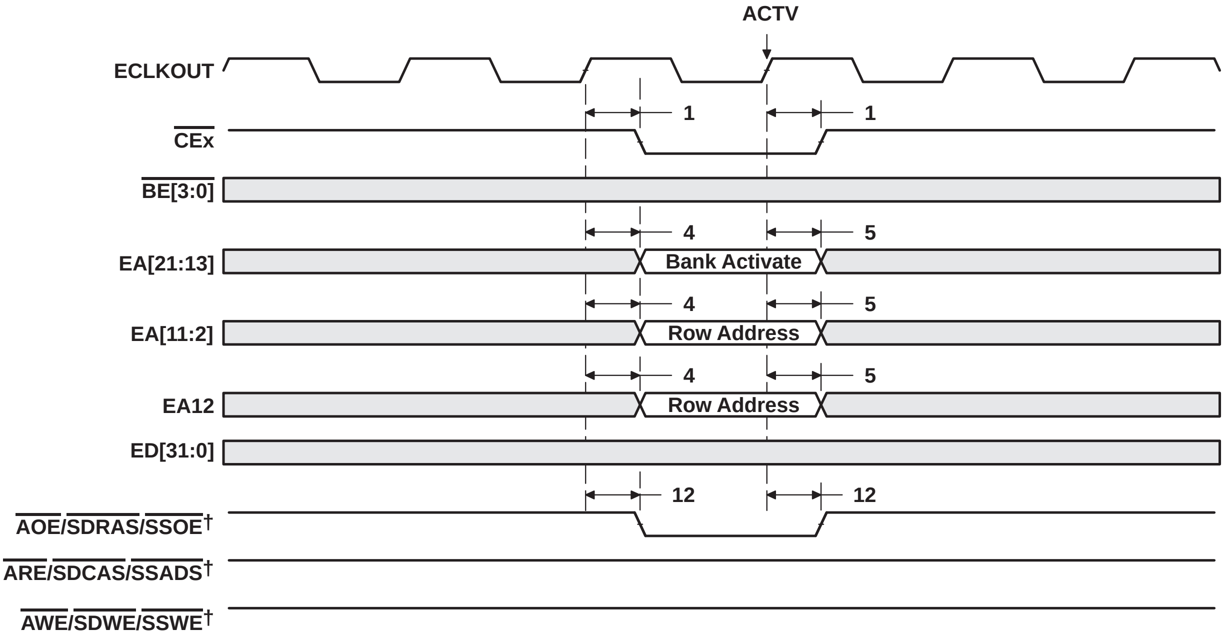


† $\overline{ARE}/\overline{SDCAS}/\overline{SSADS}$, $\overline{AWE}/\overline{SDWE}/\overline{SSWE}$, and $\overline{AOE}/\overline{SDRAS}/\overline{SSOE}$ operate as $\overline{SDCAS}$, $\overline{SDWE}$, and $\overline{SDRAS}$, respectively, during SDRAM accesses.

Figure 20. SDRAM Write Command

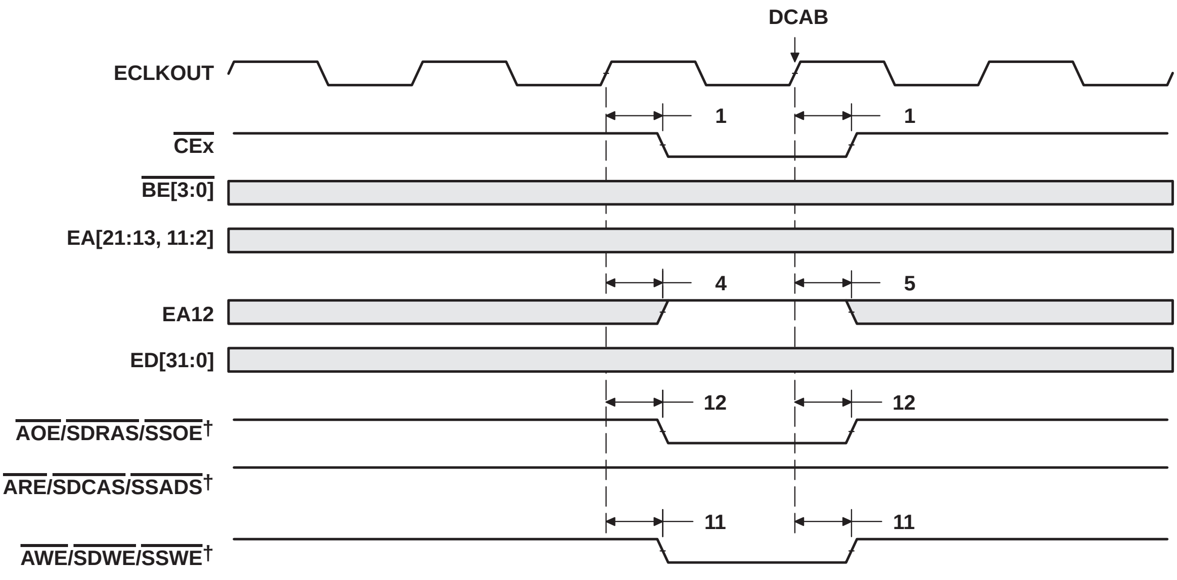
ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

Figure 21. SDRAM ACTV Command

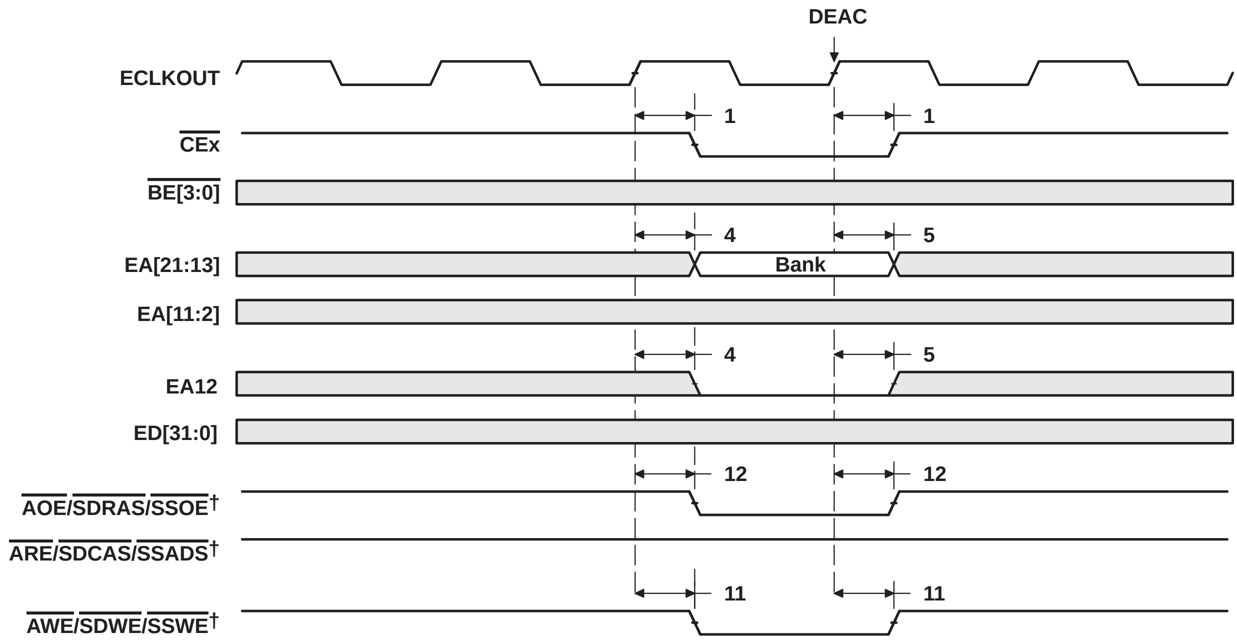


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 22. SDRAM DCAB Command

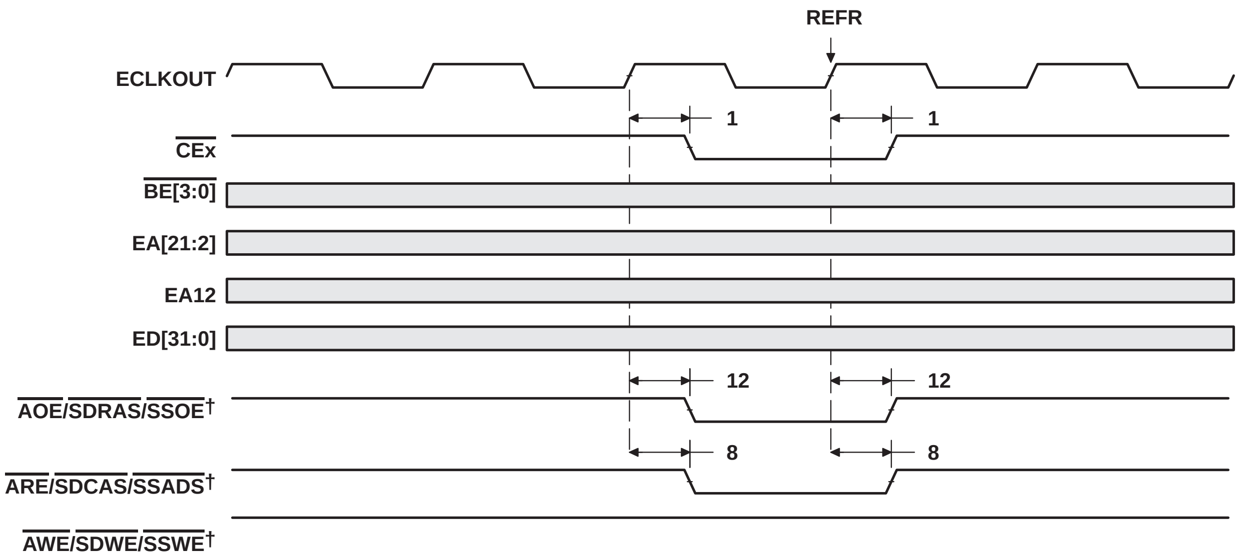
ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

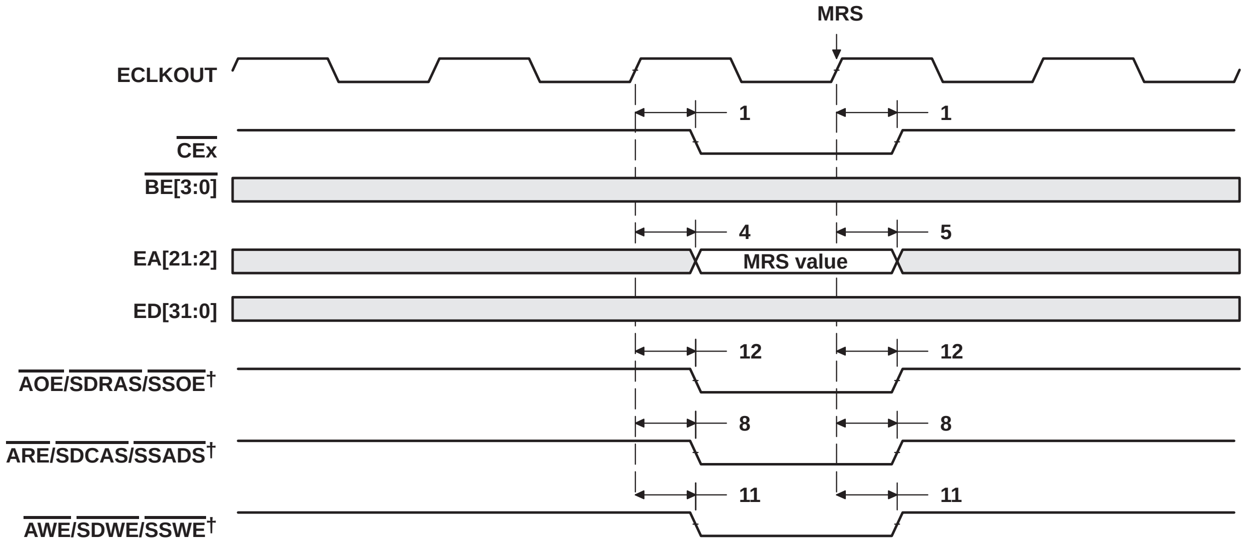
Figure 23. SDRAM DEAC Command



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

Figure 24. SDRAM REFR Command

SYNCHRONOUS DRAM TIMING (CONTINUED)



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

Figure 25. SDRAM MRS Command

ADVANCE INFORMATION

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

timing requirements for the $\overline{HOLD}$/$\overline{HOLDA}$ cycles† (see Figure 26)

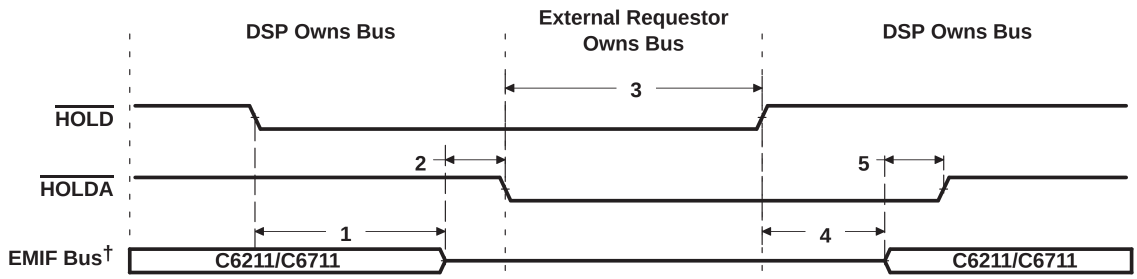
NO.			–100 –150 MIN	–100 –150 MAX	UNIT
3	$t_{oh(HOLDAL-HOLDL)}$	Hold time, $\overline{HOLD}$ low after $\overline{HOLDA}$ low	E		ns

† E = ECLKIN period in ns

switching characteristics for the $\overline{HOLD}$/$\overline{HOLDA}$ cycles†‡ (see Figure 26)

NO.	PARAMETER		–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{R(HOLDL-EMHZ)}$	Response time, $\overline{HOLD}$ low to EMIF Bus high impedance	2E	§	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to $\overline{HOLDA}$ low	0	2E	ns
4	$t_{R(HOLDH-EMLZ)}$	Response time, $\overline{HOLD}$ high to EMIF Bus low impedance	2E	7E	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to $\overline{HOLDA}$ high	0	2E	ns

† E = ECLKIN period in ns
‡ EMIF Bus consists of $\overline{CE[3:0]}$, $\overline{BE[3:0]}$, ED[31:0], EA[21:2], $\overline{ARE}$/$\overline{SDCAS}$/$\overline{SSADS}$, $\overline{AOE}$/$\overline{SDRAS}$/$\overline{SSOE}$, and $\overline{AWE}$/$\overline{SDWE}$/$\overline{SSWE}$.
§ All pending EMIF transactions are allowed to complete before $\overline{HOLDA}$ is asserted. If no bus transactions are occurring, then the minimum delay time can be achieved. Also, bus hold can be indefinitely delayed by setting NOHOLD = 1.



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

Figure 26. $\overline{HOLD}$/$\overline{HOLDA}$ Timing

ADVANCE INFORMATION

RESET TIMING

timing requirements for reset† (see Figure 27)

NO.			–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{w(RST)}$	Width of the RESET pulse (PLL stable)‡	10P		ns
		Width of the RESET pulse (PLL needs to sync up)§	250		µs
14	$t_{su(HD)}$	Setup time, HD boot configuration bits valid before RESET high¶	2P		ns
15	$t_{h(HD)}$	Hold time, HD boot configuration bits valid after RESET high¶	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.
‡ This parameter applies to CLKMODE x1 when CLKIN is stable, and applies to CLKMODE x4 when CLKIN and PLL are stable.
§ This parameter applies to CLKMODE x4 only (it does not apply to CLKMODE x1). 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 power up 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.
¶ HD[4:3] are the boot configuration pins during device reset.

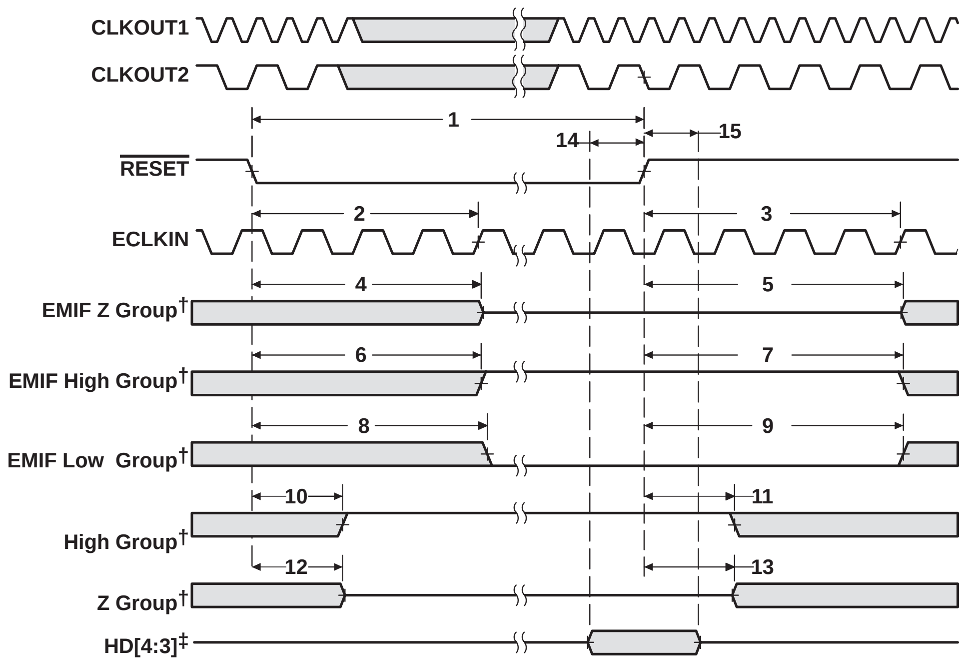
switching characteristics during reset†#|| (see Figure 27)

NO.	PARAMETER		–100 –150 MIN	–100 –150 MAX	UNIT
2	$t_{R(RSTL-ECKI)}$	Response time, RESET low to ECLKIN synchronized	2P + 3E	3P + 4E	ns
3	$t_{R(RSTH-ECKI)}$	Response time, RESET high to ECLKIN synchronized	2P + 3E	3P + 4E	ns
4	$t_{d(RSTL-EMIFZHZ)}$	Delay time, RESET low to EMIF Z group high impedance	2P + 3E		ns
5	$t_{d(RSTH-EMIFZV)}$	Delay time, RESET high to EMIF Z group valid		3P + 4E	ns
6	$t_{d(RSTL-EMIFHIV)}$	Delay time, RESET low to EMIF high group invalid	2P + 3E		ns
7	$t_{d(RSTH-EMIFHV)}$	Delay time, RESET high to EMIF high group valid		3P + 4E	ns
8	$t_{d(RSTL-EMIFLIV)}$	Delay time, RESET low to EMIF low group invalid	2P + 3E		ns
9	$t_{d(RSTH-EMIFLV)}$	Delay time, RESET high to EMIF low group valid		3P + 4E	ns
10	$t_{d(RSTL-HIGHIV)}$	Delay time, RESET low to high group invalid	2P		ns
11	$t_{d(RSTH-HIGHV)}$	Delay time, RESET high to high group valid		4P	ns
12	$t_{d(RSTL-ZHZ)}$	Delay time, RESET low to Z group high impedance	2P		ns
13	$t_{d(RSTH-ZV)}$	Delay time, RESET high to Z group valid	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.
# E = ECLKIN period in ns
|| EMIF Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE
EMIF high group consists of: HOLDA
EMIF low group consists of: BUSREQ
High group consists of: HRDY and HINT
Z group consists of: HD[15:0], CLKX0, CLKX1, FSX0, FSX1, DX0, DX1, CLKR0, CLKR1, FSR0, FSR1, TOUT0, and TOUT1.

ADVANCE INFORMATION

RESET TIMING (CONTINUED)



† EMIF Z group consists of: EA[21:2], ED[31:0], CE[3:0], BE[3:0], ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE
EMIF high group consists of: HOLDA
EMIF low group consists of: BUSREQ
High group consists of: HRDY and HINT
Z group consists of: HD[15:0], CLKX0, CLKX1, FSX0, FSX1, DX0, DX1, CLKR0, CLKR1, FSR0, FSR1, TOUT0, and TOUT1.
‡ HD[4:3] are the boot configuration pins during device reset.

Figure 27. Reset Timing

EXTERNAL INTERRUPT TIMING

timing requirements for external interrupts† (see Figure 28)

NO.			–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{w(ILOW)}$	Width of the interrupt pulse low	2E		ns
2	$t_{w(IHIGH)}$	Width of the interrupt pulse high	2E		ns

† E = ECLKIN period in ns

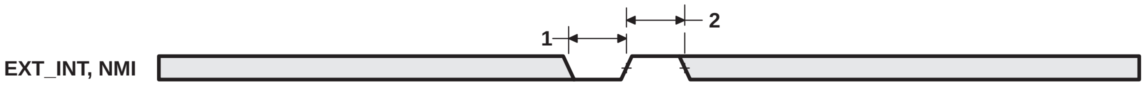


Figure 28. External/NMI Interrupt Timing

ADVANCE INFORMATION

HOST-PORT INTERFACE TIMING

timing requirements for host-port interface cycles†‡ (see Figure 29, Figure 30, Figure 31, and Figure 32)

NO.			–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{su(SELV-HSTBL)}$	Setup time, select signals§ valid before HSTROBE low	5		ns
2	$t_{h(HSTBL-SELV)}$	Hold time, select signals§ valid after HSTROBE low	2		ns
3	$t_{w(HSTBL)}$	Pulse duration, HSTROBE low	4P		ns
4	$t_{w(HSTBH)}$	Pulse duration, HSTROBE high between consecutive accesses	4P		ns
10	$t_{su(SELV-HASL)}$	Setup time, select signals§ valid before HAS low	5		ns
11	$t_{h(HASL-SELV)}$	Hold time, select signals§ valid after HAS low	2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before HSTROBE high	5		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after HSTROBE high	2		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.	2		ns
18	$t_{su(HASL-HSTBL)}$	Setup time, HAS low before HSTROBE low	2		ns
19	$t_{h(HSTBL-HASL)}$	Hold time, HAS low after HSTROBE low	2		ns

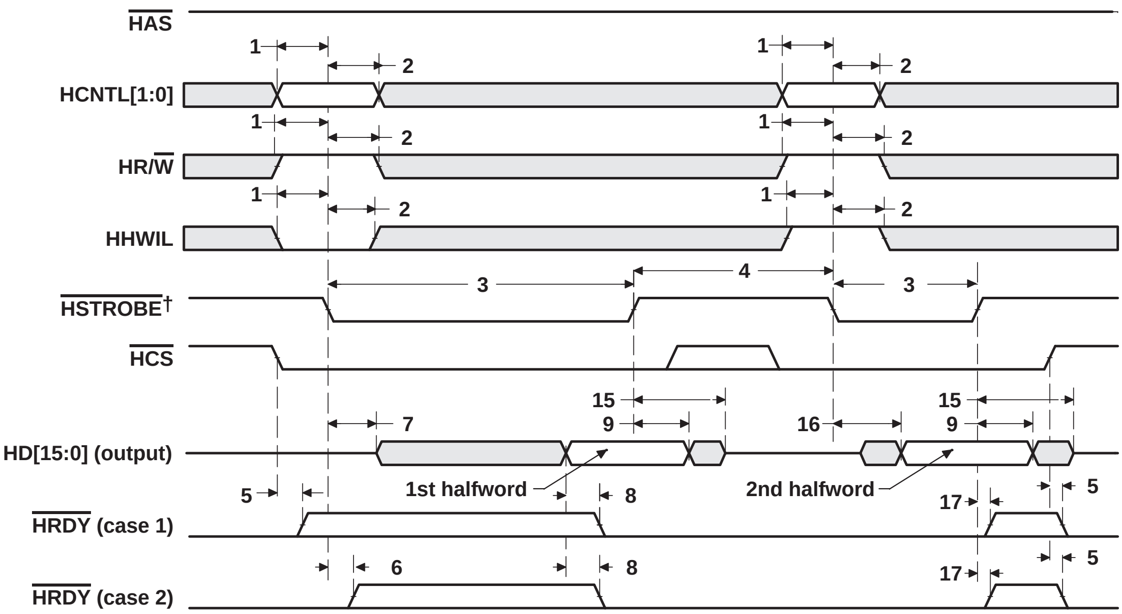
† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.
§ Select signals include: HCNTL[1:0], HR/W, and HHWIL.

switching characteristics during host-port interface cycles†‡ (see Figure 29, Figure 30, Figure 31, and Figure 32)

NO.	PARAMETER		–100 –150 MIN	–100 –150 MAX	UNIT
5	$t_{d(HCS-HRDY)}$	Delay time, HCS to HRDY¶	1	7	ns
6	$t_{d(HSTBL-HRDYH)}$	Delay time, HSTROBE low to HRDY high#	3	12	ns
7	$t_{d(HSTBL-HDLZ)}$	Delay time, HSTROBE low to HD low impedance for an HPI read	2		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to HRDY low	2P – 4	2P	ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after HSTROBE high	3	12	ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, HSTROBE high to HD high impedance	3	12	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, HSTROBE low to HD valid	3	12	ns
17	$t_{d(HSTBH-HRDYH)}$	Delay time, HSTROBE high to HRDY high‖	3	12	ns
20	$t_{d(HASL-HRDYH)}$	Delay time, HAS low to HRDY high	3	12	ns

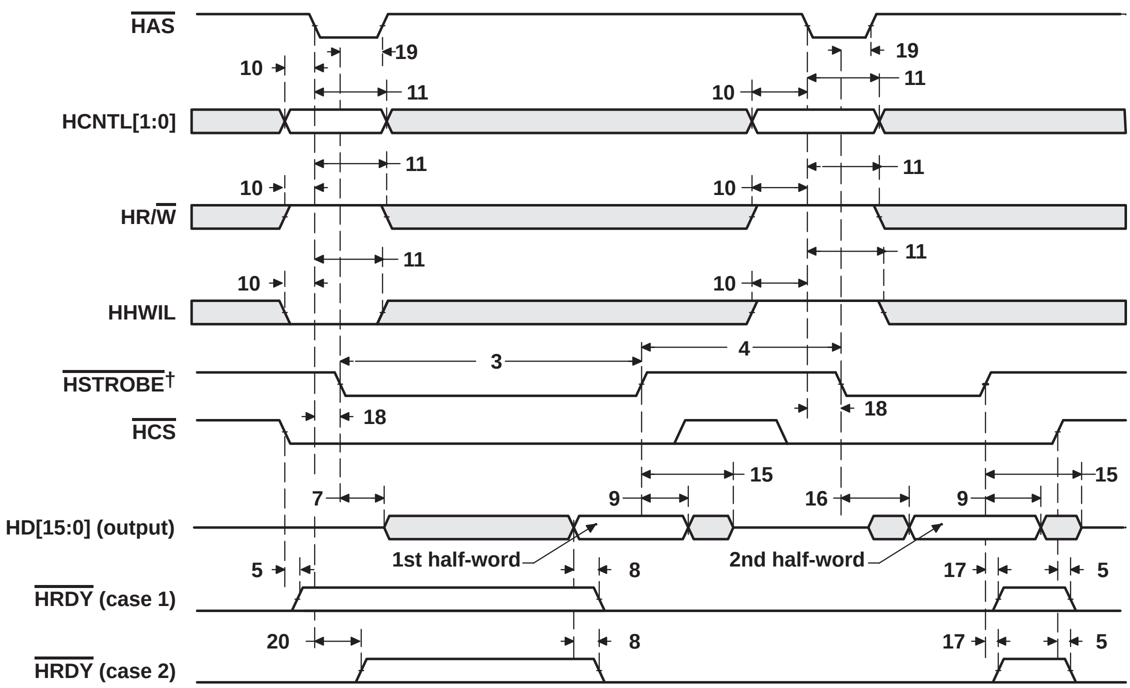
† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.
‡ P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.
¶ HCS enables HRDY, and HRDY is always low when HCS is high. The case where HRDY goes high when HCS falls 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 EDMA internal address generation hardware, and HRDY remains high until the EDMA internal address generation hardware 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)



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

Figure 29. HPI Read Timing (HAS Not Used, Tied High)

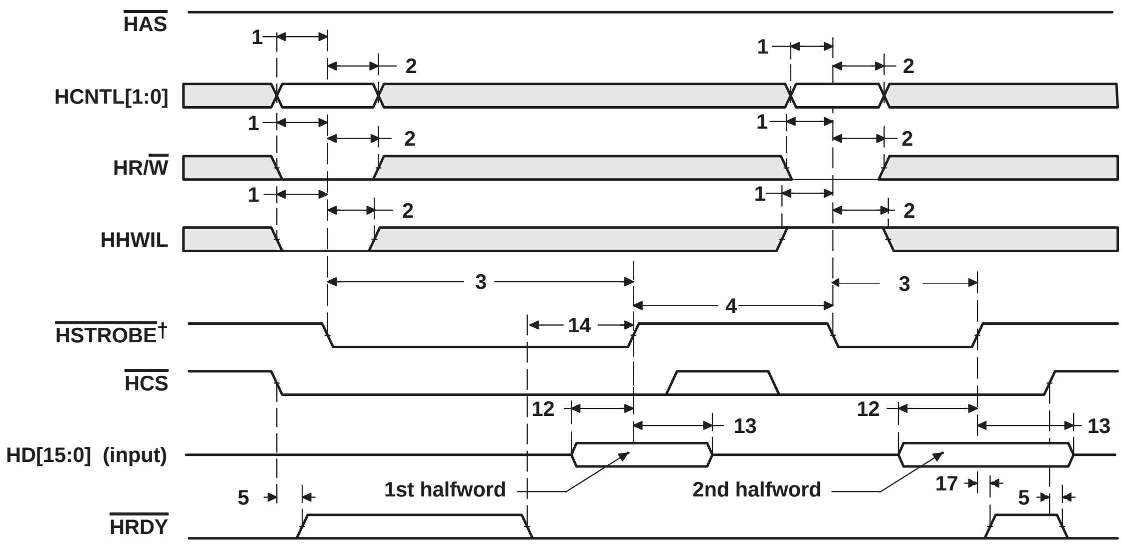


† HSTROBE refers to the following logical operation on HCS, HDS1, and HDS2: [NOT(HDS1 XOR HDS2)] OR HCS.

Figure 30. HPI Read Timing (HAS Used)

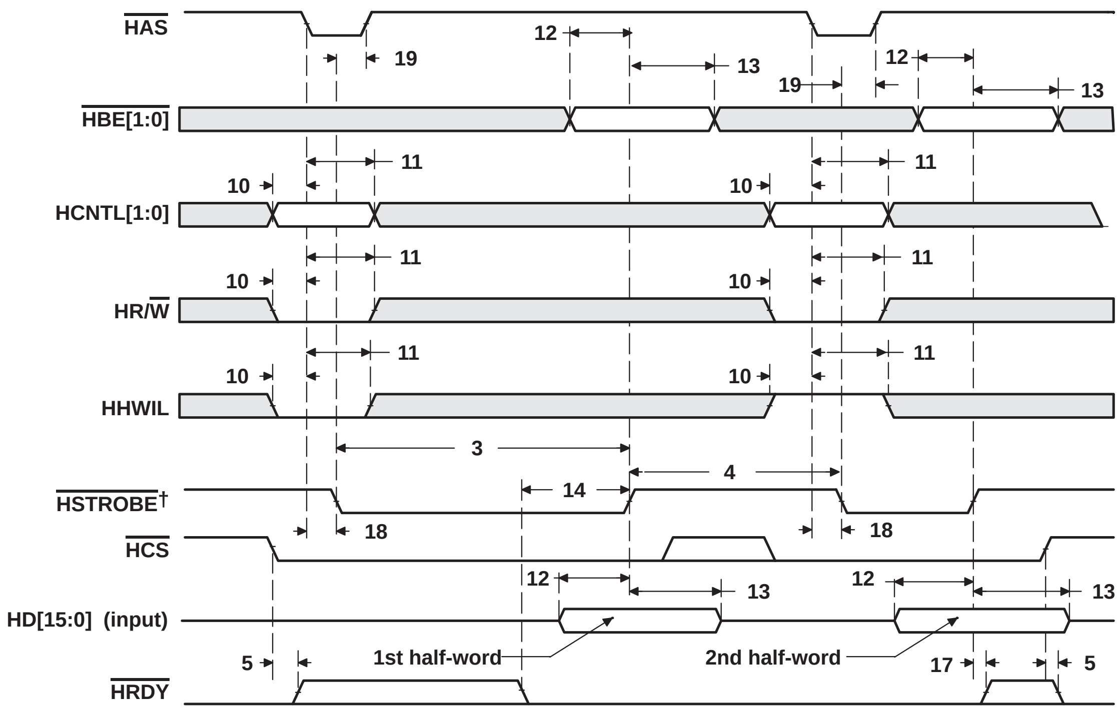
ADVANCE INFORMATION

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 31. 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 32. HPI Write Timing ($\overline{\text{HAS}}$ Used)

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP†‡ (see Figure 33)

NO.				–100 –150 MIN	–100 –150 MAX	UNIT
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P§		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1¶		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9		ns
			CLKR ext	1		
6	$t_{h(CKRL-FRH)}$	Hold time, external FSR high after CLKR low	CLKR int	6		ns
			CLKR ext	3		
7	$t_{su(DRV-CKRL)}$	Setup time, DR valid before CLKR low	CLKR int	8		ns
			CLKR ext	0		
8	$t_{h(CKRL-DRV)}$	Hold time, DR valid after CLKR low	CLKR int	3		ns
			CLKR ext	3		
10	$t_{su(FXH-CKXL)}$	Setup time, external FSX high before CLKX low	CLKX int	9		ns
			CLKX ext	1		
11	$t_{h(CKXL-FXH)}$	Hold time, external FSX high after CLKX low	CLKX int	6		ns
			CLKX ext	3		

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ P = 1/CPU clock frequency in ns.
§ The maximum McBSP bit rate is 50 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 20 ns (50 MHz), whichever value is larger. For example, when running parts at 150 MHz (P = 6.7 ns), use 20 ns as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 80 MHz (P = 12.5 ns), use 2P = 25 ns (40 MHz) as the minimum CLKR/X clock cycle. The maximum McBSP bit rate applies to the following hardware configuration: the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
¶ The minimum CLKR/X pulse duration is either (P–1) or 9 ns, whichever is larger. For example, when running parts at 150 MHz (P = 6.7 ns), use 9 ns as the minimum CLKR/X pulse duration. When running parts at 80 MHz (P = 12.5 ns), use (P–1) = 11.5 ns as the minimum CLKR/X pulse duration.

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics for McBSP[†‡] (see Figure 33)

NO.	PARAMETER			–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		4	10	ns
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X int	2P[§¶]		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X int	C – 1[#]	C + 1[#]	ns
4	$t_{d(CKRH-FRV)}$	Delay time, CLKR high to internal FSR valid	CLKR int	–2	3	ns
9	$t_{d(CKXH-FXV)}$	Delay time, CLKX high to internal FSX valid	CLKX int	–2	3	ns
			CLKX ext	3	9	
12	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	CLKX int	–1	4	ns
			CLKX ext	3	9	
13	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	CLKX int	–1 + D[‖]	4 + D[‖]	ns
			CLKX ext	3 + D[‖]	9 + D[‖]	
14	$t_{d(FXH-DXV)}$	Delay time, FSX high to DX valid ONLY applies when in data delay 0 (XDATDLY = 00b) mode	FSX int	–1	3	ns
			FSX ext	3	9	

† CLKRP = CLKXP = FSRP = FSXP = 0. If polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
‡ Minimum delay times also represent minimum output hold times.
§ P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.
¶ The maximum McBSP bit rate is 50 MHz; therefore, the minimum CLKR/X clock cycle is either twice the CPU cycle time (2P), or 20 ns (50 MHz), whichever value is larger. For example, when running parts at 150 MHz (P = 6.7 ns), use 20 ns as the minimum CLKR/X clock cycle (by setting the appropriate CLKGDV ratio or external clock source). When running parts at 80 MHz (P = 12.5 ns), use 2P = 25 ns (40 MHz) as the minimum CLKR/X clock cycle. The maximum McBSP bit rate applies to the following hardware configuration: the serial port is a master of the clock and frame syncs (with CLKR connected to CLKX, FSR connected to FSX, CLKXM = FSXM = 1, and CLKRM = FSRM = 0) in data delay 1 or 2 mode (R/XDATDLY = 01b or 10b) and the other device the McBSP communicates to is a slave.
# C = H or L
S = sample rate generator input clock = 2P if CLKSM = 1 (P = 1/CPU clock frequency)
= sample rate generator input clock = P_clks if CLKSM = 0 (P_clks = CLKS period)
H = CLKX high pulse width = (CLKGDV/2 + 1) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
L = CLKX low pulse width = (CLKGDV/2) * S if CLKGDV is even
= (CLKGDV + 1)/2 * S if CLKGDV is odd or zero
CLKGDV should be set appropriately to ensure the McBSP bit rate does not exceed the 50 MHz limit.
‖ Extra delay from CLKX high to DX valid applies *only* to the first data bit of a device, if and only if DXENA = 1 in SPCR.
D = extra delay from CLKX high to DX vaild = 0 if DXENA = 0
= extra delay from CLKX high to DX vaild = 2P if DXENA = 1

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

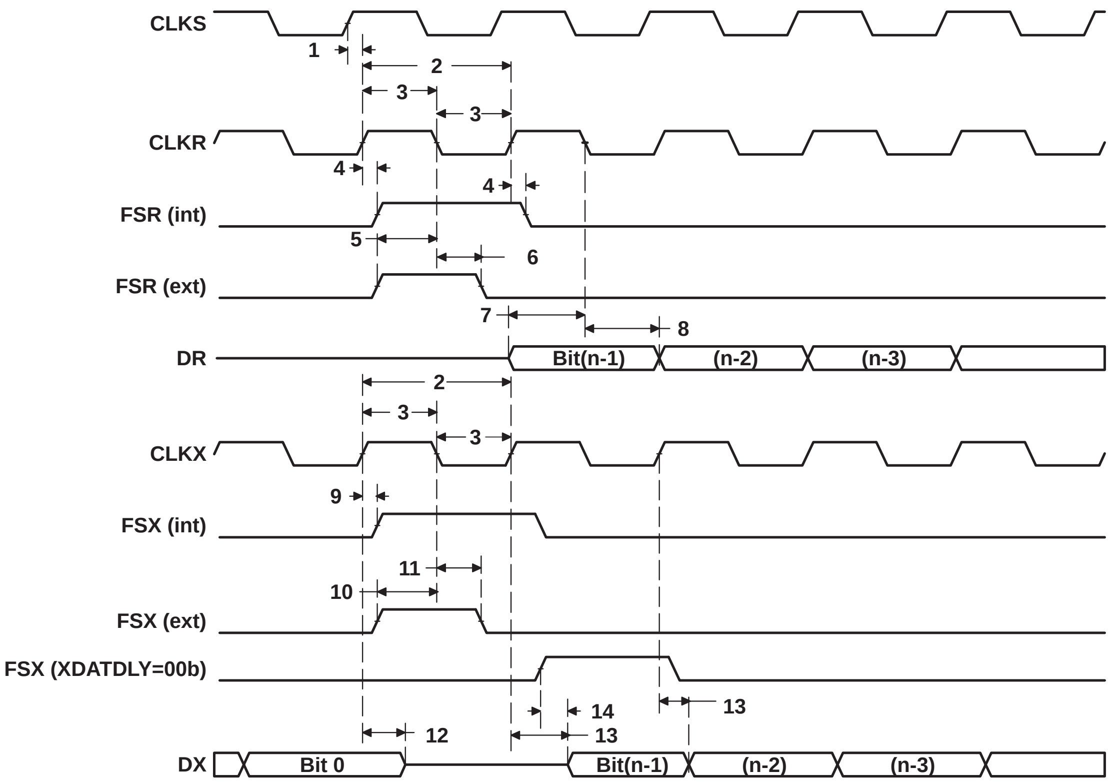


Figure 33. McBSP Timings

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for FSR when GSYNC = 1 (see Figure 34)

NO.			–100 –150 MIN	–100 –150 MAX	UNIT
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

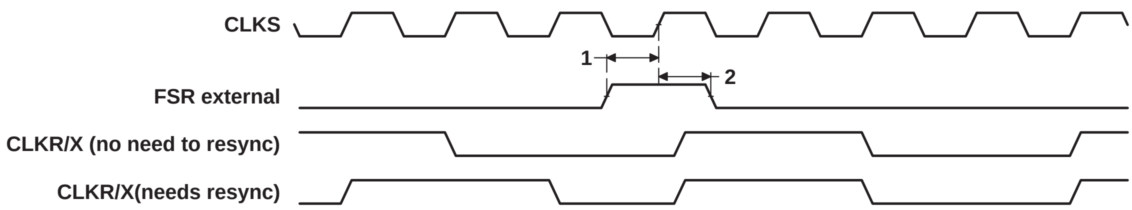


Figure 34. FSR Timing When GSYNC = 1

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 0†‡ (see Figure 35)

NO.			–100 –150				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXL)}$	Setup time, DR valid before CLKX low	12		2 – 6P		ns
5	$t_{h(CKXL-DRV)}$	Hold time, DR valid after CLKX low	4		5 + 12P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 35)

NO.	PARAMETER		–100 –150				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXL-FXL)}$	Hold time, FSX low after CLKX low¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXH)}$	Delay time, FSX low to CLKX high#	L – 2	L + 3			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	6P + 4	10P + 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			2P + 3	6P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			4P + 2	8P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 = 2P 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).

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

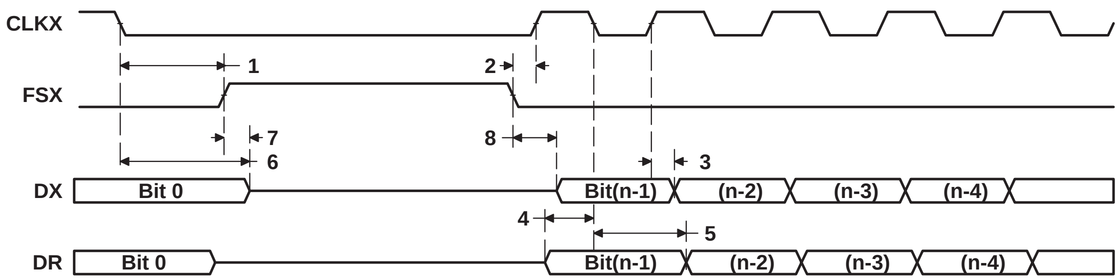


Figure 35. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 0†‡ (see Figure 36)

NO.			–100 –150				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 6P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 12P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 36)

NO.	PARAMETER		–100 –150				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	6P + 4	10P + 17	ns
6	$t_{dis(CKXL-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX low	–2	4	6P + 3	10P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	H – 2	H + 4	4P + 2	8P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 = 2P 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).

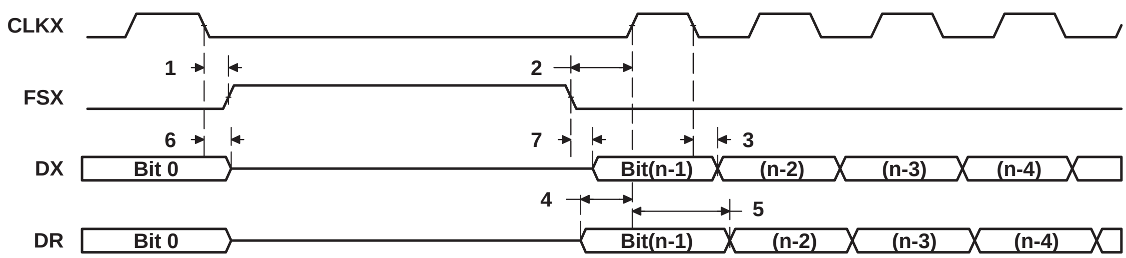


Figure 36. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 10b, CLKXP = 1†‡ (see Figure 37)

NO.			–100 –150				UNIT
			MASTER		SLAVE		
			MIN	MAX	MIN	MAX	
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 6P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 12P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 37)

NO.	PARAMETER		–100 –150				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	T – 2	T + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	H – 2	H + 3			ns
3	$t_{d(CKXL-DXV)}$	Delay time, CLKX low to DX valid	–2	4	6P + 4	10P + 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			2P + 3	6P + 17	ns
8	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid			4P + 2	8P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 = 2P 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).

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

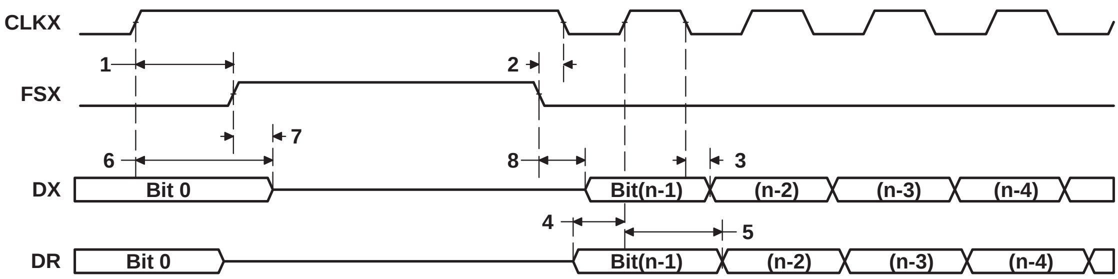


Figure 37. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

timing requirements for McBSP as SPI master or slave: CLKSTP = 11b, CLKXP = 1†‡ (see Figure 38)

NO.			–100 –150 MASTER MIN	MASTER MAX	SLAVE MIN	SLAVE MAX	UNIT
4	$t_{su(DRV-CKXH)}$	Setup time, DR valid before CLKX high	12		2 – 6P		ns
5	$t_{h(CKXH-DRV)}$	Hold time, DR valid after CLKX high	4		5 + 12P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 38)

NO.	PARAMETER		–100 –150 MASTER§ MIN	MASTER§ MAX	SLAVE MIN	SLAVE MAX	UNIT
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	6P + 4	10P + 17	ns
6	$t_{dis(CKXH-DXHZ)}$	Disable time, DX high impedance following last data bit from CLKX high	–2	4	6P + 3	10P + 17	ns
7	$t_{d(FXL-DXV)}$	Delay time, FSX low to DX valid	L – 2	L + 4	4P + 2	8P + 17	ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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 = 2P 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).

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

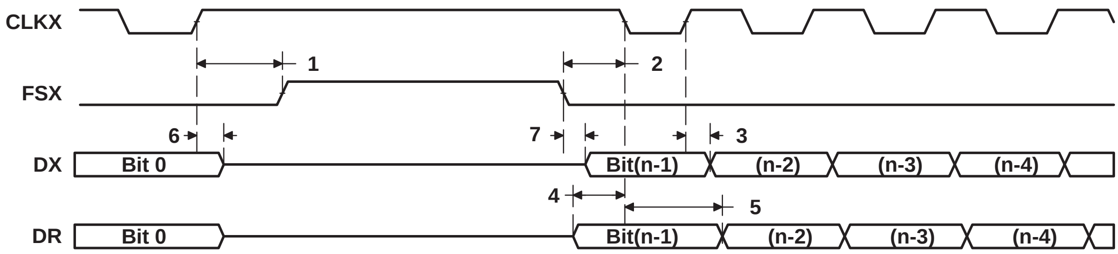


Figure 38. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

ADVANCE INFORMATION

TIMER TIMING

timing requirements for timer inputs† (see Figure 39)

NO.			–100 –150		UNIT
			MIN	MAX	
1	$t_{w(TINPH)}$	Pulse duration, TINP high	2P		ns
2	$t_{w(TINPL)}$	Pulse duration, TINP low	2P		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.

switching characteristics for timer outputs† (see Figure 39)

NO.	PARAMETER		–100 –150		UNIT
			MIN	MAX	
3	$t_{w(TOUTH)}$	Pulse duration, TOUT high	4P–3		ns
4	$t_{w(TOUTL)}$	Pulse duration, TOUT low	4P–3		ns

† P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 ns.

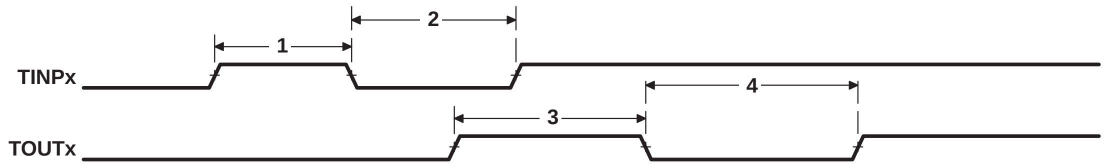


Figure 39. Timer Timing

ADVANCE INFORMATION

JTAG TEST-PORT TIMING

timing requirements for JTAG test port (see Figure 40)

NO.			–100 –150 MIN	–100 –150 MAX	UNIT
1	$t_{c(TCK)}$	Cycle time, TCK	35		ns
3	$t_{su(TDIV\text{-}TCKH)}$	Setup time, TDI/TMS/$\overline{\text{TRST}}$ valid before TCK high	10		ns
4	$t_{h(TCKH\text{-}TDIV)}$	Hold time, TDI/TMS/$\overline{\text{TRST}}$ valid after TCK high	9		ns

switching characteristics for JTAG test port (see Figure 40)

NO.	PARAMETER		–100 –150 MIN	–100 –150 MAX	UNIT
2	$t_{d(TCKL\text{-}TDOV)}$	Delay time, TCK low to TDO valid	–3	12	ns

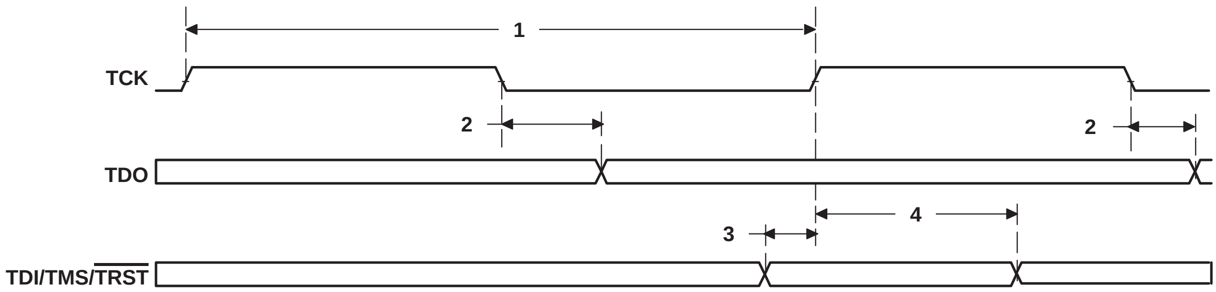


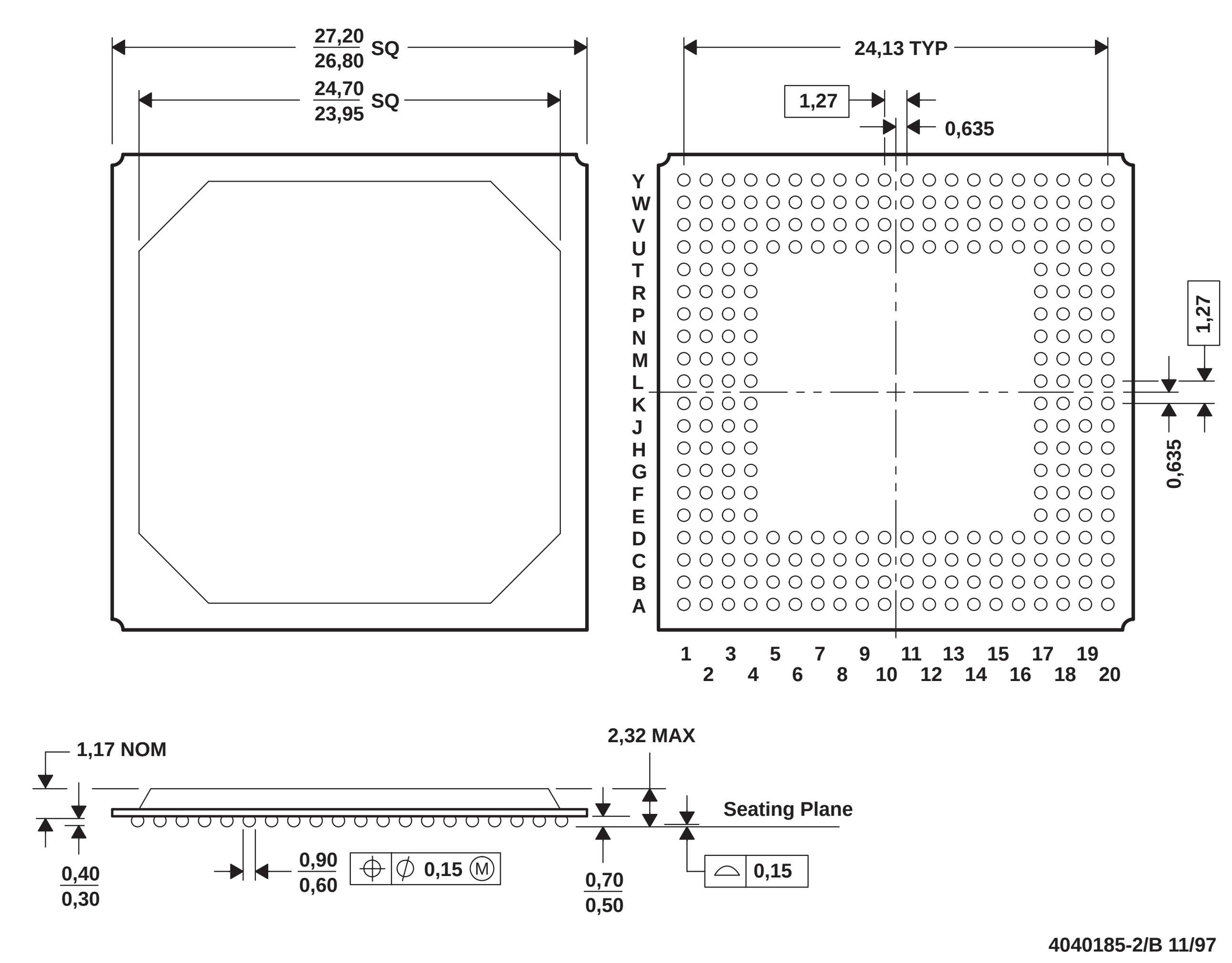
Figure 40. JTAG Test-Port Timing

ADVANCE INFORMATION

ADVANCE INFORMATION

MECHANICAL DATA

GFN (S-PBGA-N256) **PLASTIC BALL GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

thermal resistance characteristics (S-PBGA package)

NO			°C/W	Air Flow LFPM†
1	$R\Theta_{JC}$	Junction-to-case	6.4	N/A
2	$R\Theta_{JA}$	Junction-to-free air	25.2	0
3	$R\Theta_{JA}$	Junction-to-free air	23.1	100
4	$R\Theta_{JA}$	Junction-to-free air	21.9	250
5	$R\Theta_{JA}$	Junction-to-free air	20.6	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 acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

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

Customers are responsible for their applications using TI components.

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

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