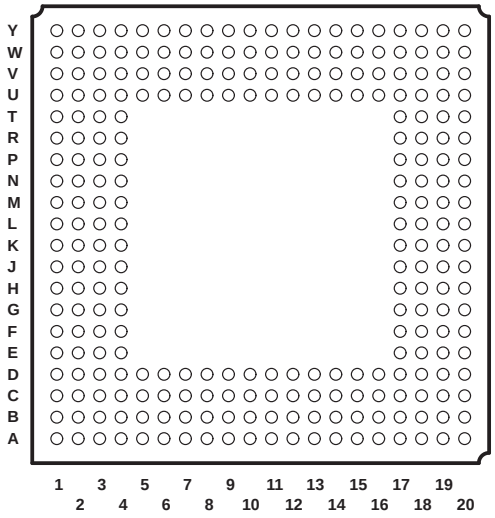
- **Best Price/Performance Floating-Point Digital Signal Processor (DSP) TMS320C6711**
  - **10-, 6.7-ns Instruction Cycle Time**
  - **100-, 150-MHz Clock Rates**
  - **Eight 32-Bit Instructions/Cycle**
  - **900 MFLOPS**
  - **Pin-Compatible With ’C6211 Fixed-Point DSP**
- **VelociTI™ Advanced Very Long Instruction Word (VLIW) ’C6700 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**
- **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)**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface to Synchronous Memories: SDRAM and SBSRAM**

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



  - **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 BGA Package (GFN Suffix)**
- **0.18-μm/5-Level Metal Process**
  - **CMOS Technology**
- **3.3-V I/Os, 1.8-V Internal**

ADVANCE INFORMATION

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

VelociTI is a trademark of Texas Instruments Incorporated.
Motorola is a trademark of Motorola, Inc.
† IEEE Standard 1149.1-1990 Standard-Test-Access Port and Boundary Scan Architecture.

description

The TMS320C67x DSPs (including the TMS320C6711 device) are the floating-point DSP family in the TMS320C6000 platform. The TMS320C6711 (’C6711) 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 900 million floating-point operations per second (MFLOPS) at a clock rate of 150 MHz, the ’C6711 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 multiply-accumulates (MACs) per cycle for a total of 300 million MACs per second (MMACS). The ’C6711 DSP also has application-specific hardware logic, on-chip memory, and additional on-chip peripherals.

The ’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 ’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.

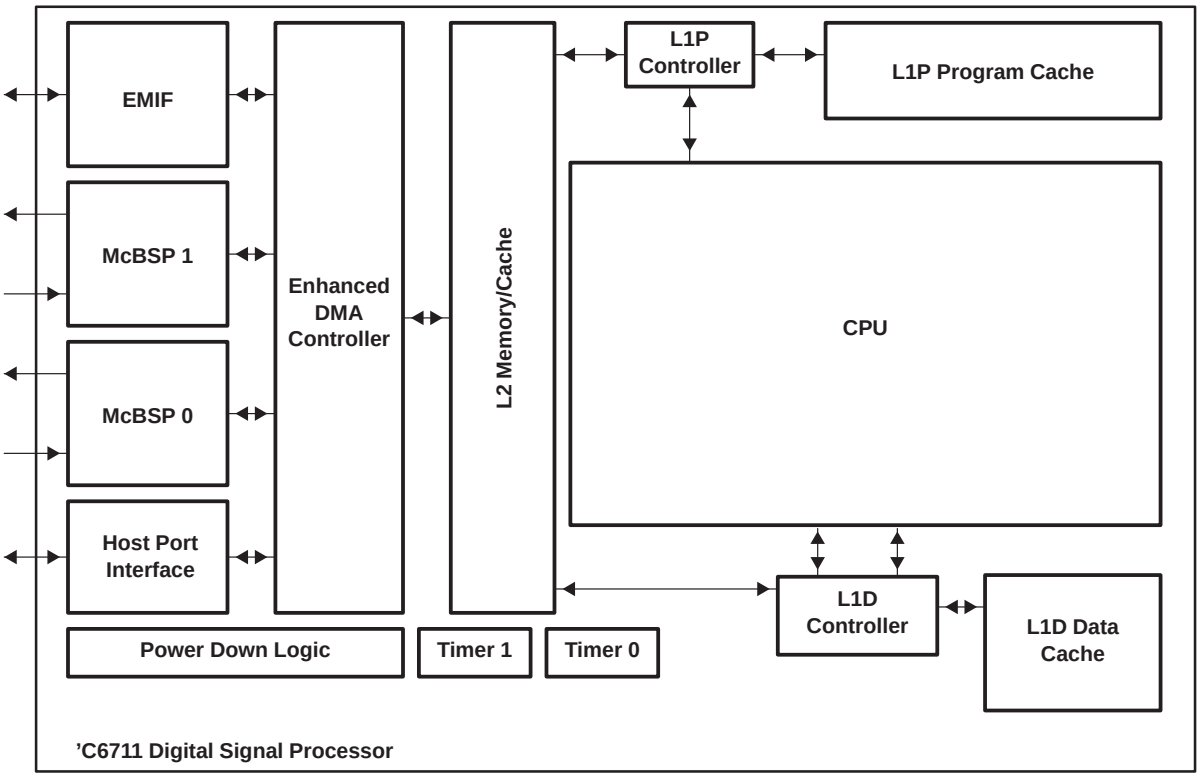
device characteristics

Table 1 provides an overview of the ’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 ’C6711 Processors

CHARACTERISTICS	DESCRIPTION
Device Number	TMS320C6711
On-Chip Memory	32-Kbit L1 Program (L1P) Cache 32-Kbit L1 Data (L1D) Cache 512-Kbit Unified Mapped RAM/Cache (L2)
Peripherals	2 Mutichannel Buffered Serial Ports (McBSP) 2 General-Purpose Timers Host-Port Interface (HPI) External Memory Interface (EMIF)
Cycle Time	6.7 ns (High-Performance Device – 150 MHz), 10 ns (Lowest-Cost Device – 100 MHz)
Package Type	27 mm × 27 mm, 256-Pin BGA (GFN)
Nominal Voltage	1.8 V Core 3.3 V I/O

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 ’C6700 CPU from other VLIW architectures.

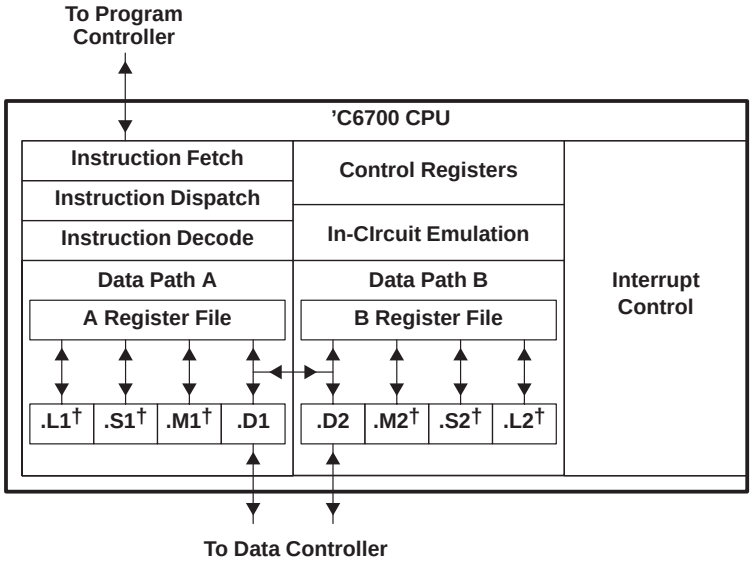
The CPU features two sets of functional units. Each set contains four units and a register file. One set contains functional units .L1, .S1, .M1, and .D1; the other set contains units .D2, .M2, .S2, and .L2. The two register files contain 16 32-bit registers each for the total of 32 general-purpose registers. 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 ’C6700 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 ’C6700 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 ’C6700 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.

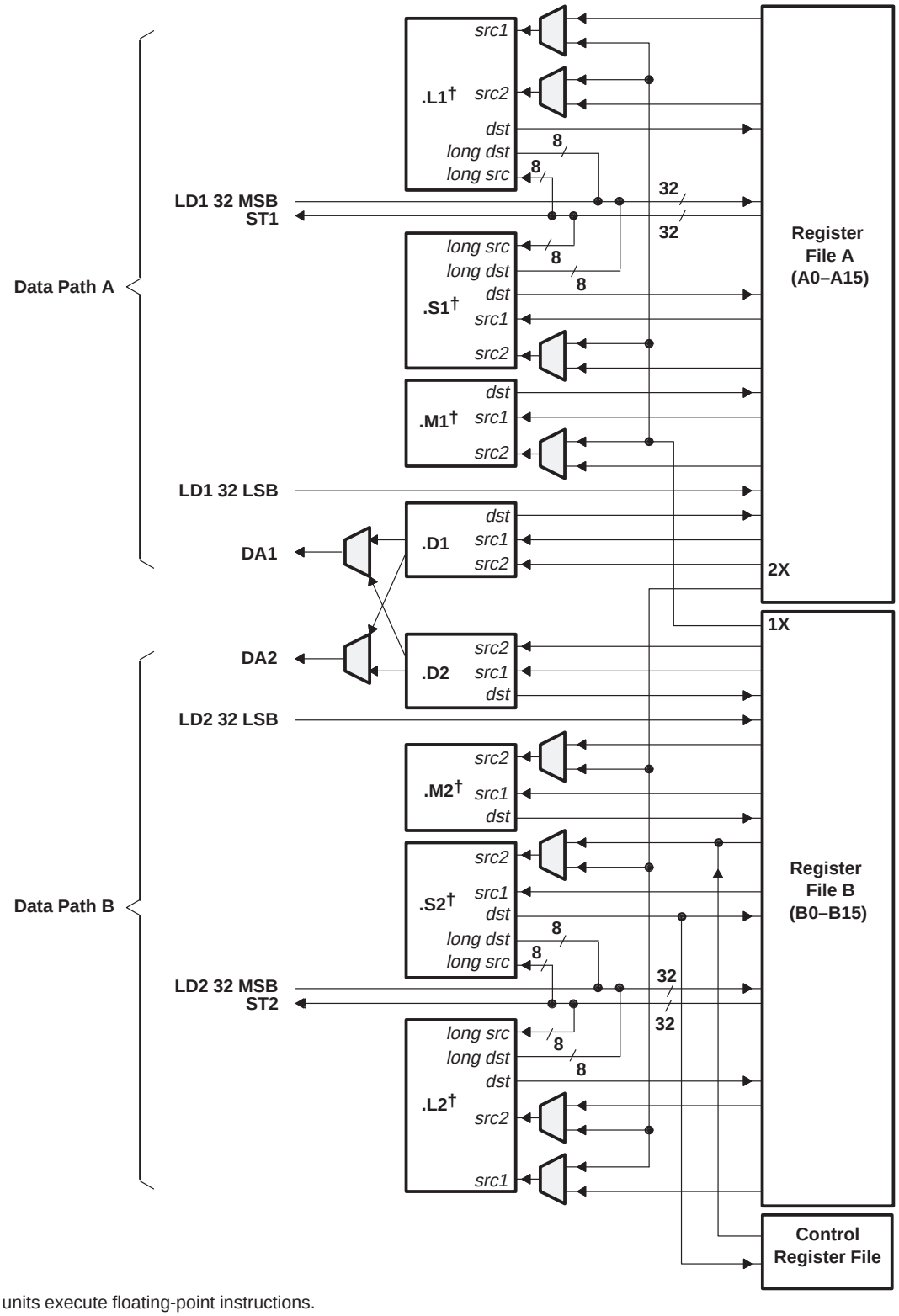
CPU description (continued)



† These functional units execute floating-point instructions.

Figure 1. TMS320C6700 CPU Block Diagram

CPU description (continued)



† These functional units execute floating-point instructions.

Figure 2. TMS320C6700 CPU Data Paths

signal groups description

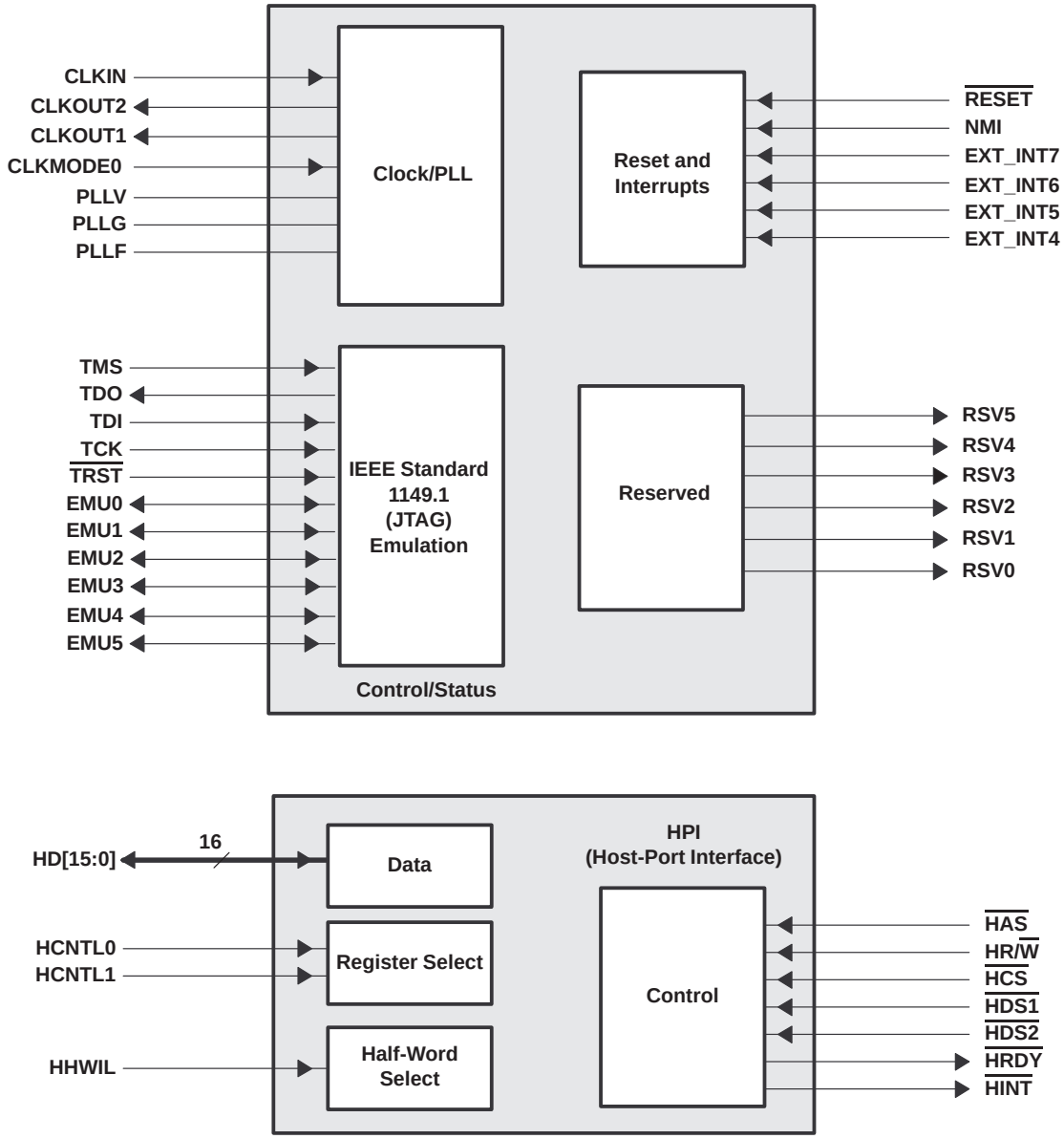


Figure 3. CPU and Peripheral Signals

ADVANCE INFORMATION

signal groups description (continued)

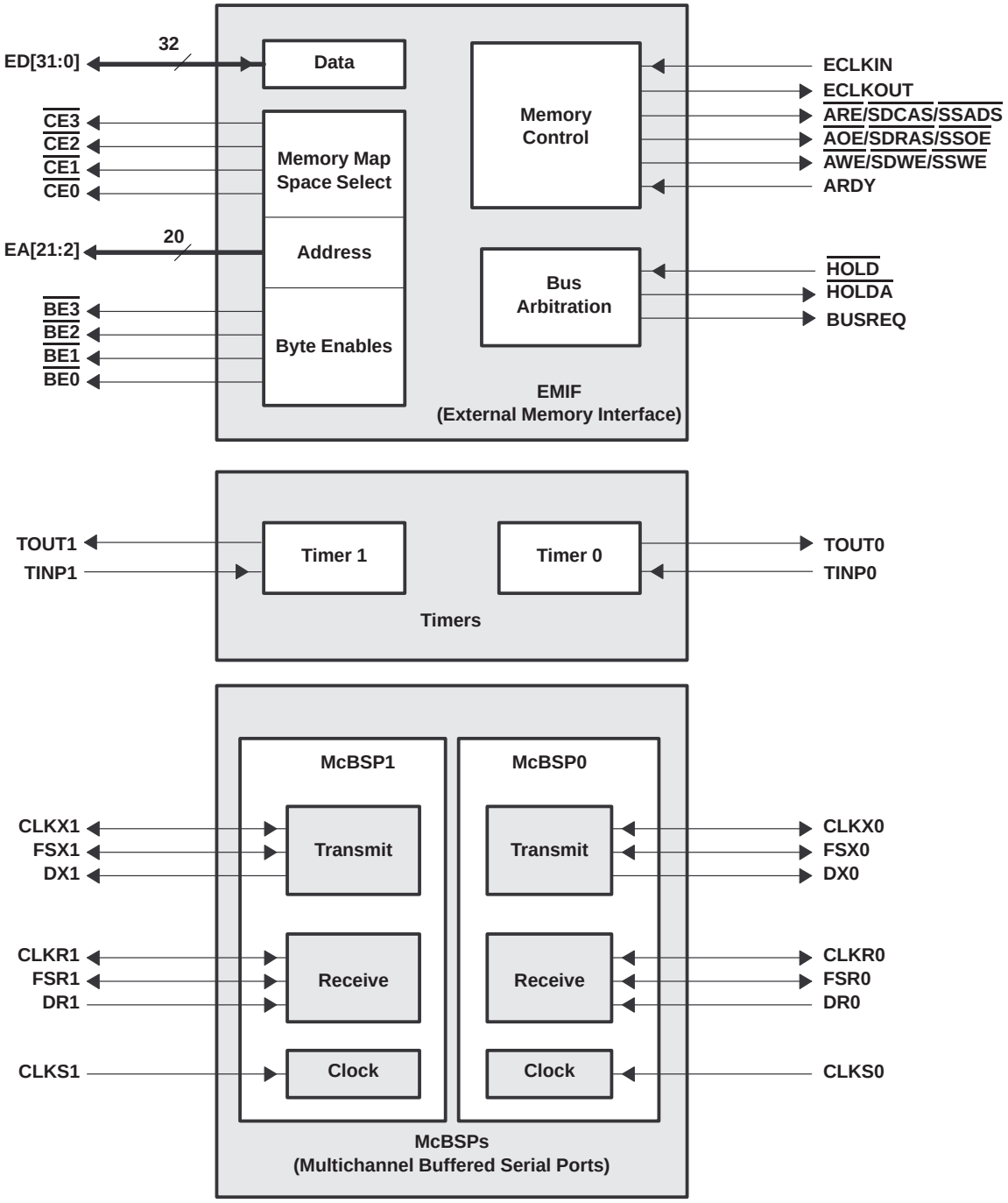


Figure 4. Peripheral Signals

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#
EMU4	C11	I/O/Z	IPU	Emulation pin 4#
EMU3	B10	I/O/Z	IPU	Emulation pin 3#
EMU2	D10	I/O/Z	IPU	Emulation pin 2#
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 [Most 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)

For interfacing details, see *JTAG/MPSD Emulation Technical Reference* (literature number SPDU079).

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 – Little endian/Big endian – Boot mode
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 [Most 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	SIGNAL 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 [Most 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 [Most 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	SIGNAL 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 [Most 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

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

Texas Instruments (TI) offers an extensive line of development tools for the ’C6000 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:
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 ’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 3 for a complete listing of development-support tools for the ’C6000. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Table 2. TMS320C6xx Development-Support Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Software		
C Compiler/Assembler/Linker/Assembly Optimizer	Win32™	TMDX3246855-07
C Compiler/Assembler/Linker/Assembly Optimizer	SPARC™ Solaris™	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.

ADVANCE INFORMATION

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, GFN) and the device speed range in megahertz (for example, -150 is 150 MHz). Table 3 provides a legend for reading the complete device name for any TMS320 family member.

device and development-support tool nomenclature (continued)

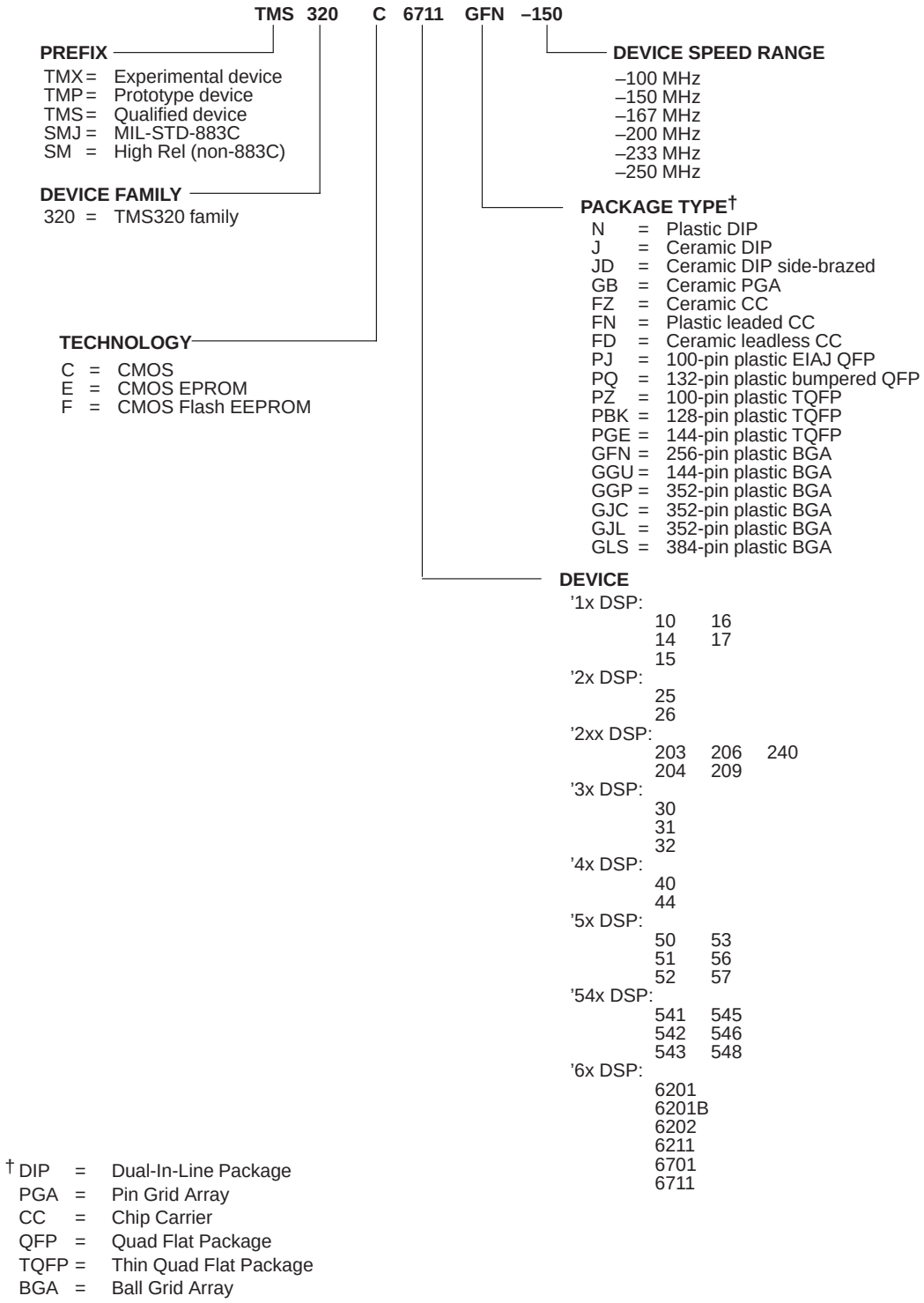


Figure 5. TMS320 Device Nomenclature (Including TMS320C6711)

ADVANCE INFORMATION

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.

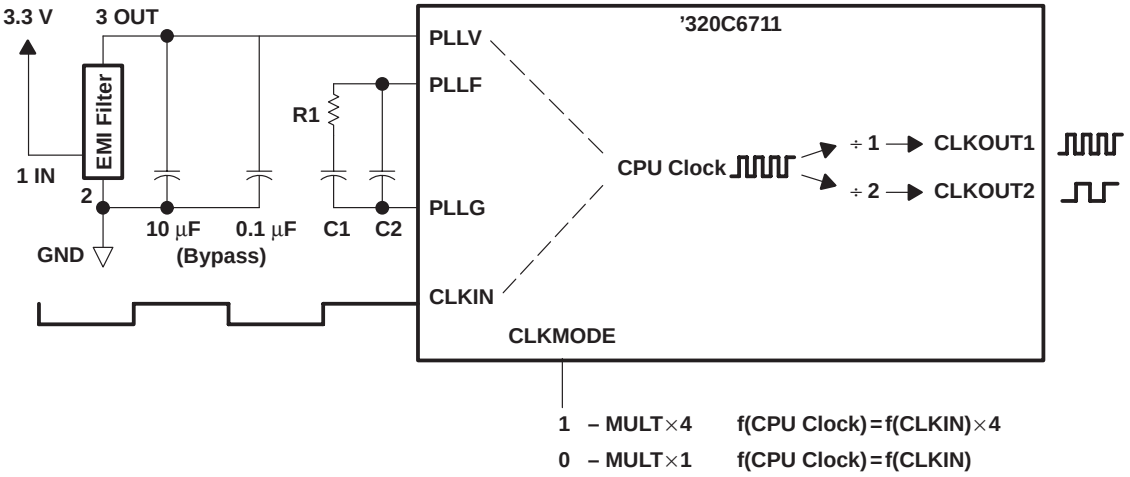
Information regarding TI DSP products is also available on the Worldwide Web at http://www.ti.com uniform resource locator (URL).

clock PLL

All of the internal ’C6711 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 ’C6711 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 ’C6711 device and the external clock oscillator circuit. The minimum CLKIN rise and fall times should also be observed. See the *input and output clocks* section for input clock timing requirements.



NOTES: A. The ’C6711 PLL can generate CPU clock frequencies in the range of 65 MHz to 150 MHz. For frequencies below 65 MHz, the PLL should be configured to operate in bypass mode.
B. For the ’C6711, values for C1, C2, and R1 are fixed and apply to all valid frequency ranges of CLKIN and CLKOUT.
C. 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.
D. The 3.3-V supply for the EMI filter (and PLLV) must be from the same 3.3-V power plane supplying the I/O voltage, DV_{DD}.
E. EMI filter manufacturer TDK part number ACF451832-153-T

Figure 6. PLL Block Diagram

Table 3. TMS320C6711 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.

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.

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		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	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

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

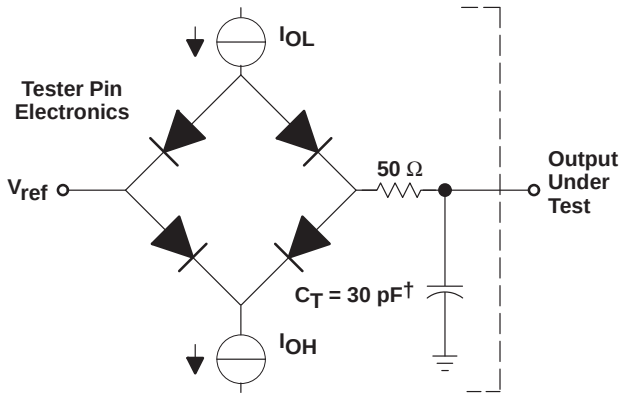
PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	DV_{DD} = MIN,	I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	DV_{DD} = MIN,	I_{OL} = MAX			0.6	V
I_I	Input current	$V_I = V_{SS}$ to DV_{DD}				±125	uA
I_{OZ}	Off-state output current	$V_O = DV_{DD}$ or 0 V				±10	uA
I_{DD2V}	Supply current, CPU + CPU memory access‡	CV_{DD} = NOM,	CPU clock = 150 MHz		TBD		mA
I_{DD2V}	Supply current, peripherals§	CV_{DD} = NOM,	CPU clock = 150 MHz		TBD		mA
I_{DD3V}	Supply current, I/O pins¶	DV_{DD} = NOM,	CPU clock = 150 MHz		TBD		mA
C_i	Input capacitance					5	pF
C_o	Output capacitance					5	pF

‡ 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):
25% of time: Reads from external SDRAM
25% of time: Writes to external SDRAM
50% of time: No activity

PARAMETER MEASUREMENT INFORMATION



† Typical distributed load circuit capacitance

Figure 7. Test Load Circuit

signal transition levels

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

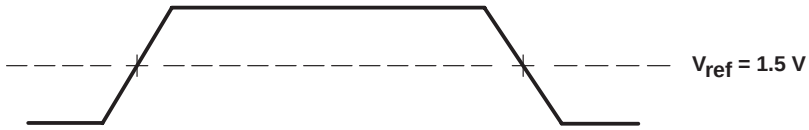


Figure 8. Input and Output Voltage Reference Levels for ac Timing Measurements

INPUT AND OUTPUT CLOCKS

timing requirements for CLKIN† (see Figure 9)

NO.			’C6711-100				’C6711-150				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	16		4.5		10.9		3		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	16		4.5		10.9		3		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}.

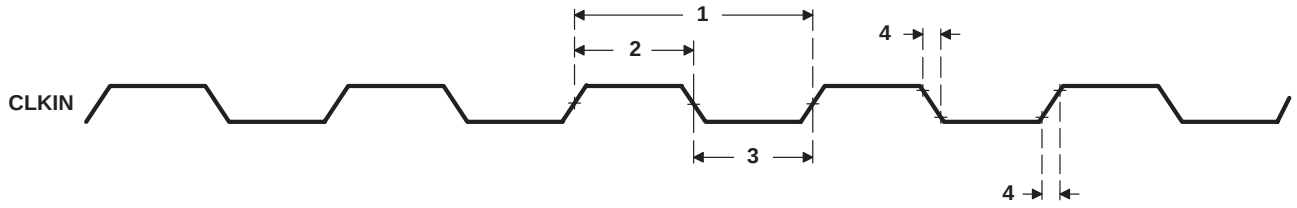


Figure 9. CLKIN Timings

switching characteristics for CLKOUT1‡§ (see Figure 10)

NO.	PARAMETER		’C6711-100 ’C6711-150				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)

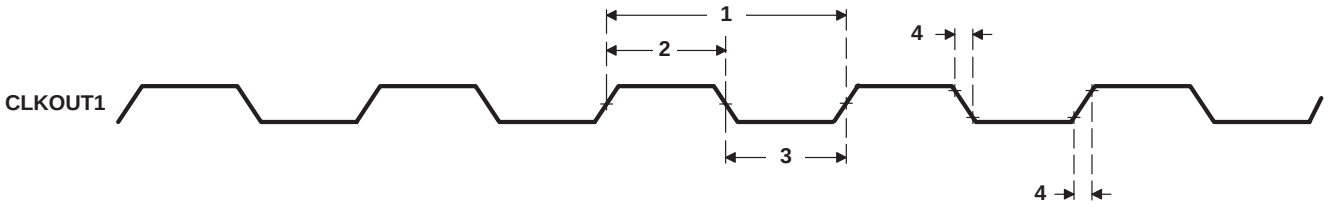


Figure 10. CLKOUT1 Timings

ADVANCE INFORMATION

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for CLKOUT2† (see Figure 11)

NO.	PARAMETER		’C6711-100 ’C6711-150		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

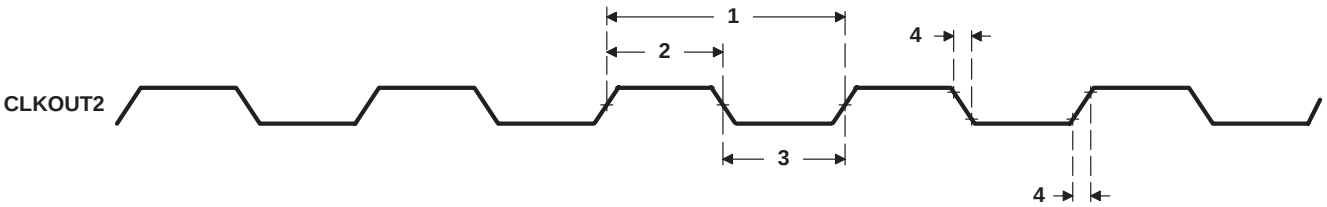


Figure 11. CLKOUT2 Timings

timing requirements for ECLKIN‡ (see Figure 12)

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{c(EKI)}$	Cycle time, ECLKIN	10		ns
2	$t_{w(EKIH)}$	Pulse duration, ECLKIN high	4.5		ns
3	$t_{w(EKIL)}$	Pulse duration, ECLKIN low	4.5		ns
4	$t_{t(EKI)}$	Transition time, ECLKIN		3	ns

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

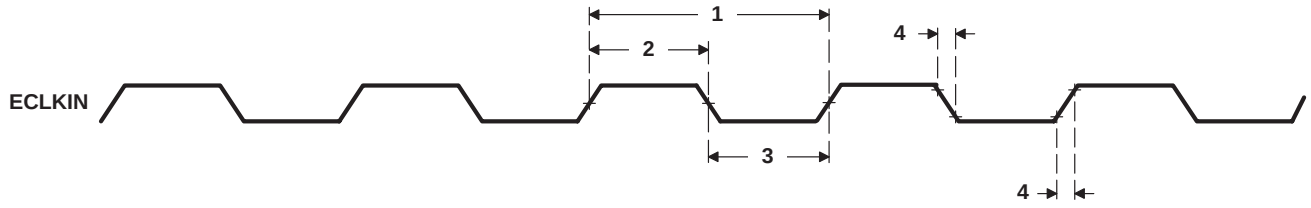


Figure 12. ECLKIN Timings

ADVANCE INFORMATION

INPUT AND OUTPUT CLOCKS (CONTINUED)

switching characteristics for ECLKOUT†‡§ (see Figure 13)

NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
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.

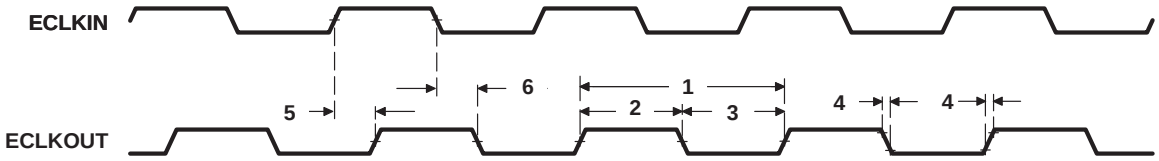


Figure 13. ECLKOUT Timings

ADVANCE INFORMATION

ASYNCHRONOUS MEMORY TIMING

timing requirements for asynchronous memory cycles† (see Figure 14–Figure 15)

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
6	$t_{su(EDV-EKOH)}$	Setup time, read EDx valid before ECLKOUT high	1.5		ns
7	$t_{h(EKOH-EDV)}$	Hold time, read EDx valid after ECLKOUT high	1		ns
10	$t_{su(ARDY-EKOH)}$	Setup time, ARDY valid before ECLKOUT high	1.5		ns
11	$t_{h(EKOH-ARDY)}$	Hold time, ARDY valid after ECLKOUT high	1		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 14–Figure 15)

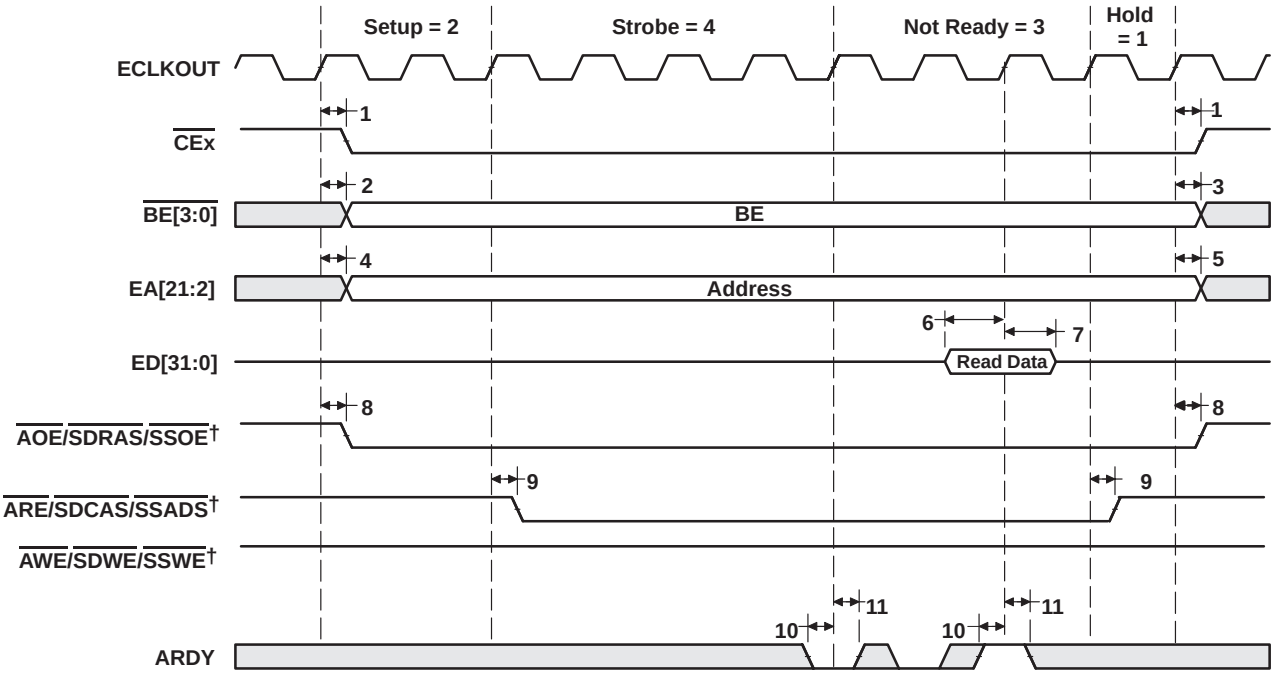
NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{d(EKOH-CEV)}$	Delay time, ECLKOUT high to $\overline{CEx}$ valid	1.5	6	ns
2	$t_{d(EKOH-BEV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ valid	1.5	6	ns
3	$t_{d(EKOH-BEIV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ invalid	1.5	6	ns
4	$t_{d(EKOH-EAV)}$	Delay time, ECLKOUT high to EAx valid	1.5	6	ns
5	$t_{d(EKOH-EAIV)}$	Delay time, ECLKOUT high to EAx invalid	1.5	6	ns
8	$t_{d(EKOH-AOEV)}$	Delay time, ECLKOUT high to $\overline{AOE}$/$\overline{SDRAS}$/$\overline{SSOE}$ valid	1.5	6	ns
9	$t_{d(EKOH-AREV)}$	Delay time, ECLKOUT high to $\overline{ARE}$/$\overline{SDCAS}$/$\overline{SSADS}$ valid	1.5	6	ns
12	$t_{d(EKOH-EDV)}$	Delay time, ECLKOUT high to EDx valid		6	ns
13	$t_{d(EKOH-EDIV)}$	Delay time, ECLKOUT high to EDx invalid	1.5		ns
14	$t_{d(EKOH-AWEV)}$	Delay time, ECLKOUT high to $\overline{AWE}$/$\overline{SDWE}$/$\overline{SSWE}$ valid	1.5	6	ns

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

§ $\overline{AOE}$/$\overline{SDRAS}$/$\overline{SSOE}$, $\overline{ARE}$/$\overline{SDCAS}$/$\overline{SSADS}$, and $\overline{AWE}$/$\overline{SDWE}$/$\overline{SSWE}$ operate as $\overline{AOE}$ (identified under select signals), $\overline{ARE}$, and $\overline{AWE}$, respectively, during asynchronous memory accesses.

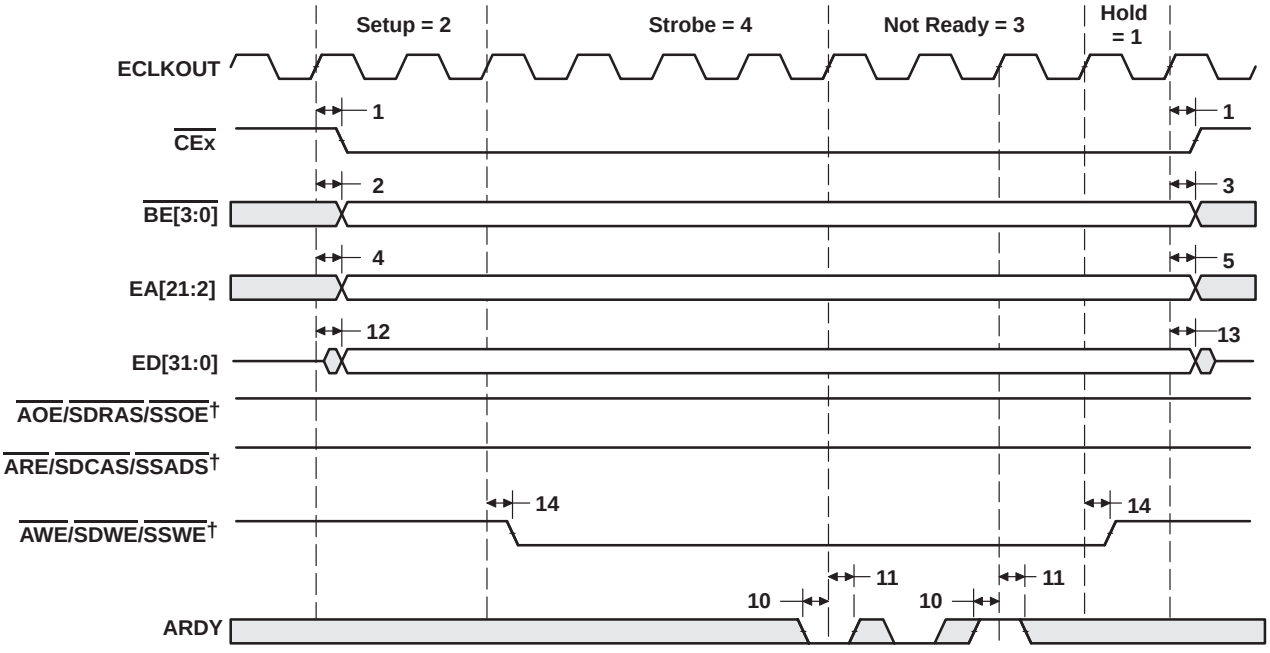
ADVANCE INFORMATION

ASYNCHRONOUS MEMORY TIMING (CONTINUED)



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

Figure 14. Asynchronous Memory Read Timing



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

Figure 15. Asynchronous Memory Write Timing

SYNCHRONOUS-BURST MEMORY TIMING

timing requirements for synchronous-burst SRAM cycles† (see Figure 16)

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
6	$t_{su(EDV-EKOH)}$	Setup time, read EDx valid before ECLKOUT high	1.5		ns
7	$t_{h(EKOH-EDV)}$	Hold time, read EDx valid after ECLKOUT high	1.0		ns

† The ’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 16 and Figure 17)

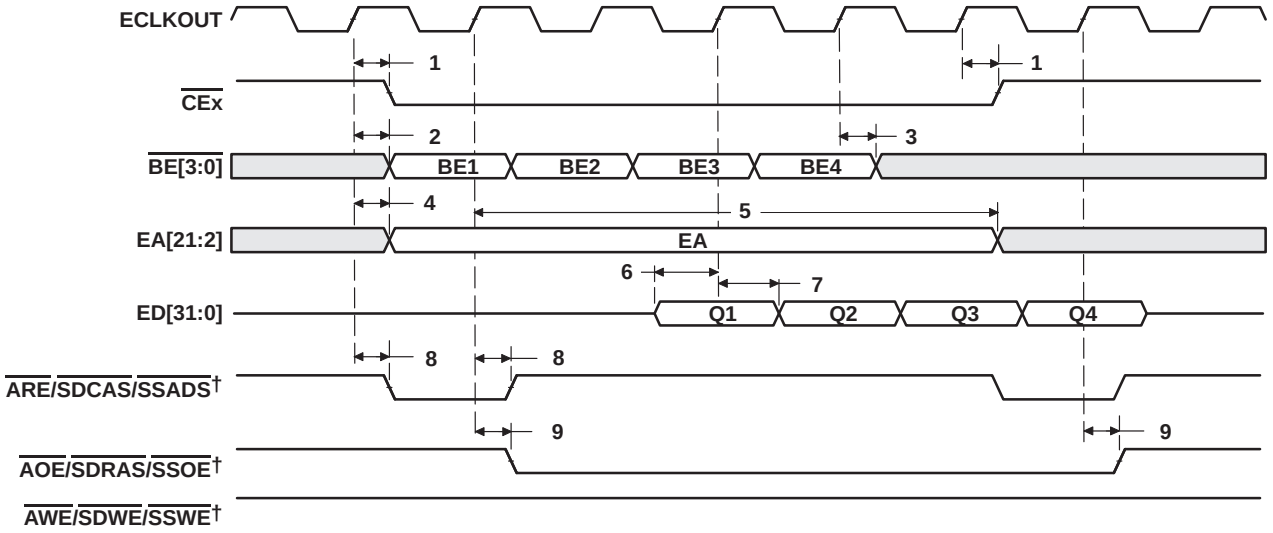
NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{d(EKOH-CEV)}$	Delay time, ECLKOUT high to $\overline{CEx}$ valid	1.5	6	ns
2	$t_{d(EKOH-BEV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ valid	1.5	6	ns
3	$t_{d(EKOH-BEIV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ invalid	1.5	6	ns
4	$t_{d(EKOH-EAV)}$	Delay time, ECLKOUT high to EAx valid	1.5	6	ns
5	$t_{d(EKOH-EAIV)}$	Delay time, ECLKOUT high to EAx invalid	1.5	6	ns
8	$t_{d(EKOH-ADSV)}$	Delay time, ECLKOUT high to $\overline{ARE}/\overline{SDCAS}/\overline{SSADS}$ valid	1.5	6	ns
9	$t_{d(EKOH-OEV)}$	Delay time, ECLKOUT high to, $\overline{AOE}/\overline{SDRAS}/\overline{SSOE}$ valid	1.5	6	ns
10	$t_{d(EKOH-EDV)}$	Delay time, ECLKOUT high to $\overline{EDx}$ valid	1.5	6	ns
11	$t_{d(EKOH-EDIV)}$	Delay time, ECLKOUT high to $\overline{EDx}$ invalid	1.5	6	ns
12	$t_{d(EKOH-WEV)}$	Delay time, ECLKOUT high to $\overline{AWE}/\overline{SDWE}/\overline{SSWE}$ valid	1.5	6	ns

† The ’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.

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

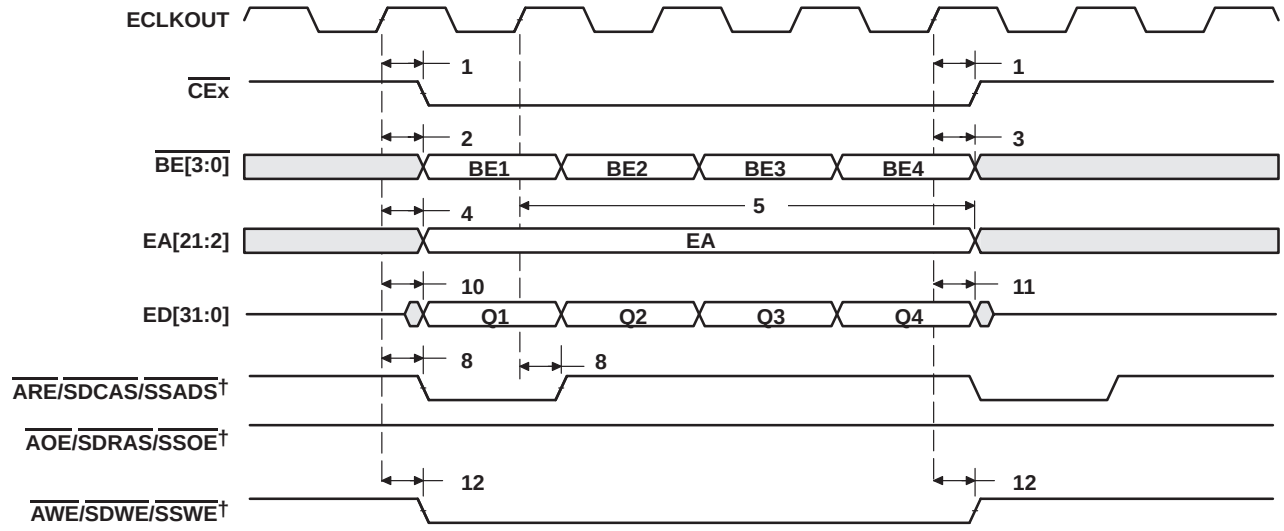
ADVANCE INFORMATION

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 16. SBSRAM Read Timing



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

Figure 17. SBSRAM Write Timing

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING

timing requirements for synchronous DRAM cycles† (see Figure 18)

NO.			'C6711-100 'C6711-150		UNIT
			MIN	MAX	
6	$t_{su(EDV-EKOH)}$	Setup time, read EDx valid before ECLKOUT high	1.5		ns
7	$t_{h(EKOH-EDV)}$	Hold time, read EDx valid after ECLKOUT high	1		ns

† The '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 18–Figure 24)

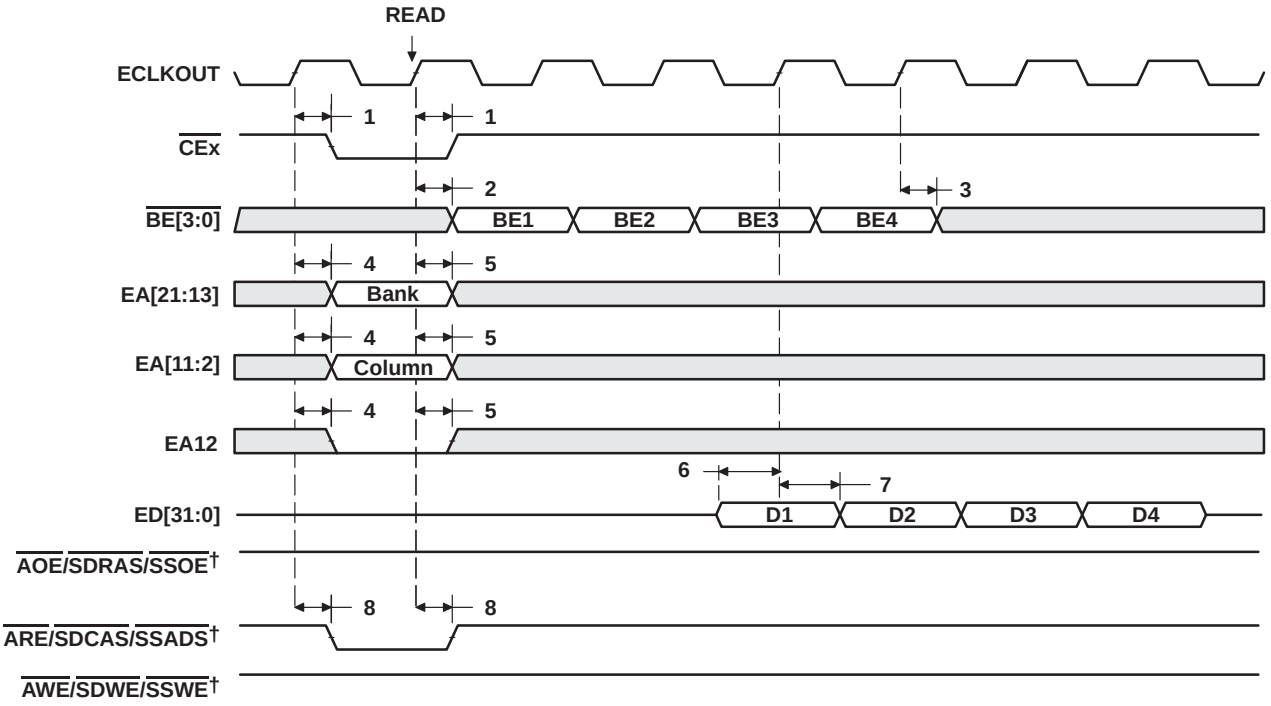
NO.	PARAMETER		'C6711-100 'C6711-150		UNIT
			MIN	MAX	
1	$t_{d(EKOH-CEV)}$	Delay time, ECLKOUT high to $\overline{CEx}$ valid	1.5	6	ns
2	$t_{d(EKOH-BEV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ valid	1.5	6	ns
3	$t_{d(EKOH-BEIV)}$	Delay time, ECLKOUT high to $\overline{BEx}$ invalid	1.5	6	ns
4	$t_{d(EKOH-EAV)}$	Delay time, ECLKOUT high to EAx valid	1.5	6	ns
5	$t_{d(EKOH-EAIV)}$	Delay time, ECLKOUT high to EAx invalid	1.5	6	ns
8	$t_{d(EKOH-CASV)}$	Delay time, ECLKOUT high to $\overline{ARE}$/$\overline{SDCAS}$/$\overline{SSADS}$ valid	1.5	6	ns
9	$t_{d(EKOH-EDV)}$	Delay time, ECLKOUT high to $\overline{EDx}$ valid	1.5	6	ns
10	$t_{d(EKOH-EDIV)}$	Delay time, ECLKOUT high to $\overline{EDx}$ invalid	1.5	6	ns
11	$t_{d(EKOH-WEV)}$	Delay time, ECLKOUT high to $\overline{AWE}$/$\overline{SDWE}$/$\overline{SSWE}$ valid	1.5	6	ns
12	$t_{d(EKOH-RAS)}$	Delay time, ECLKOUT high to, $\overline{AOE}$/$\overline{SDRAS}$/$\overline{SSOE}$ valid	1.5	6	ns

† The '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.

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

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)

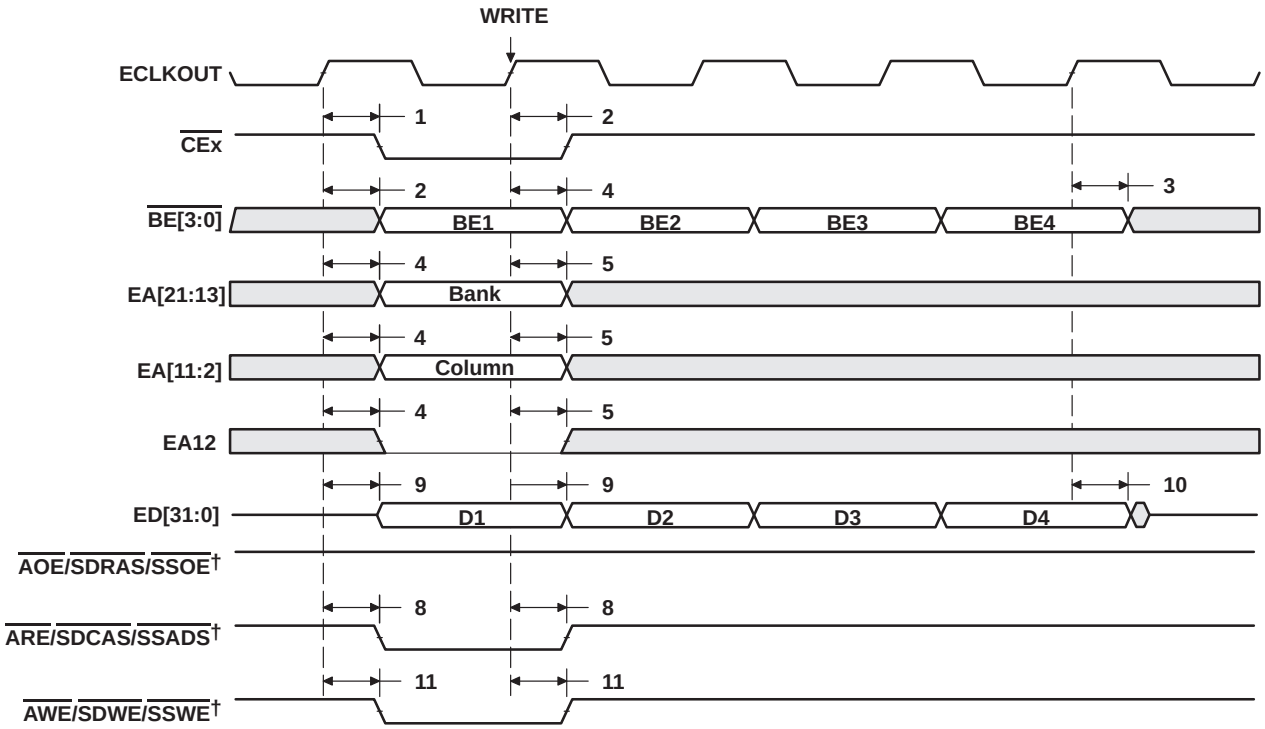


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 18. SDRAM Read Command (CAS Latency 3)

ADVANCE INFORMATION

SYNCHRONOUS DRAM TIMING (CONTINUED)

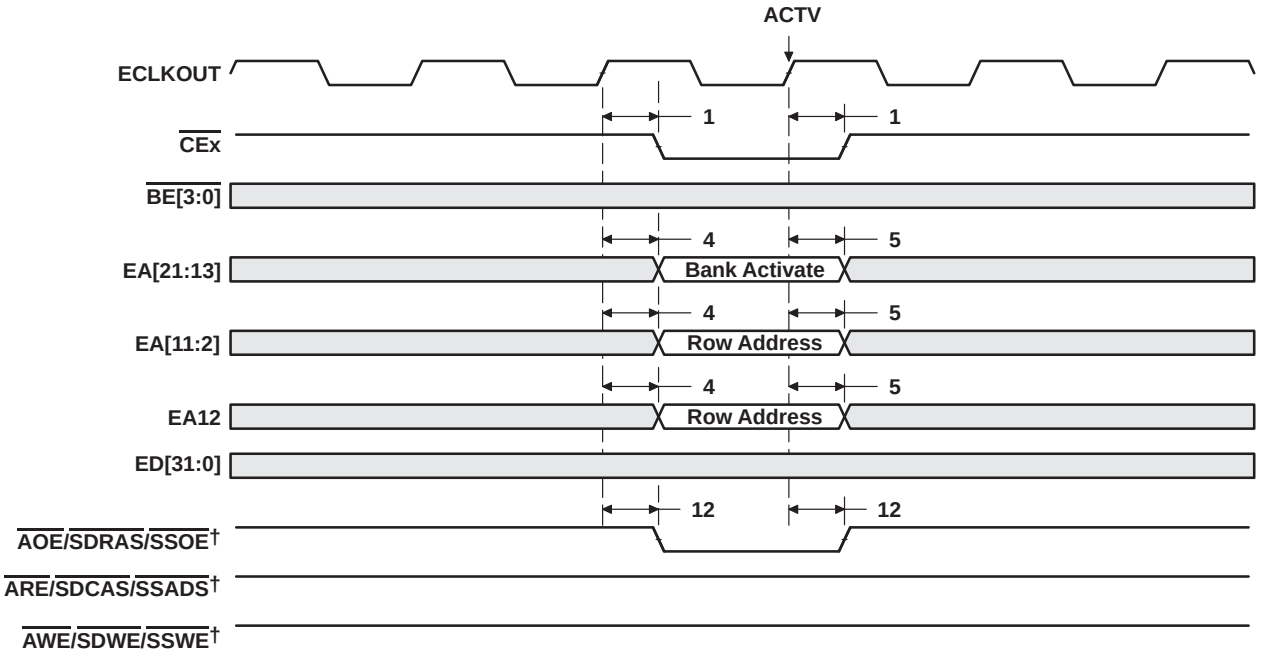


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 19. SDRAM Write Command

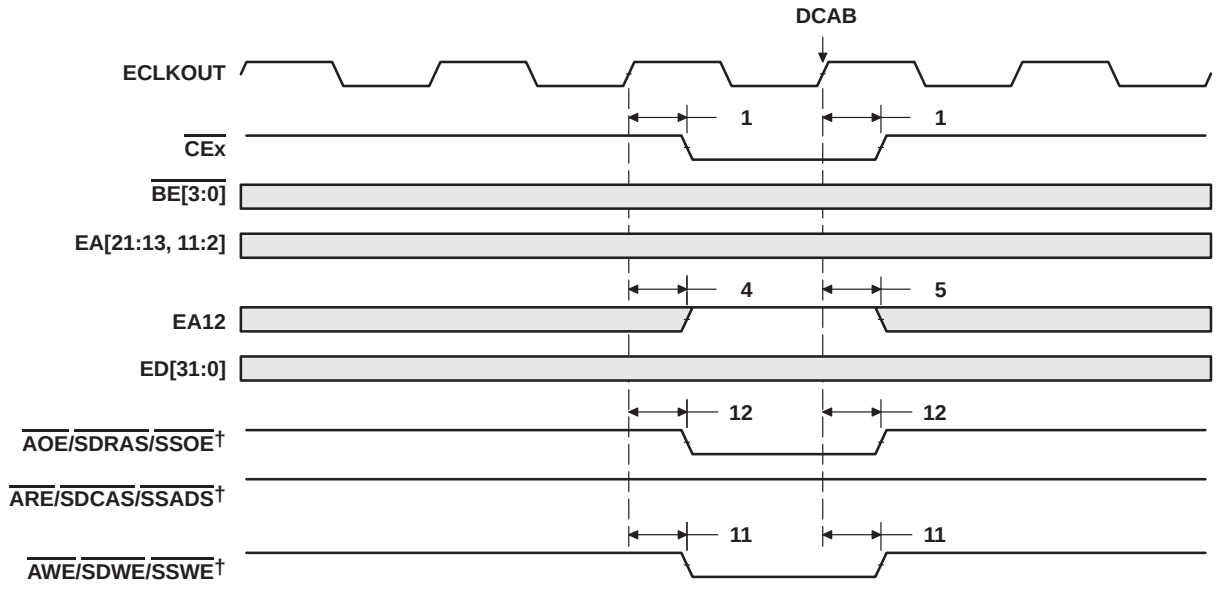
ADVANCE INFORMATION

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 20. SDRAM ACTV Command

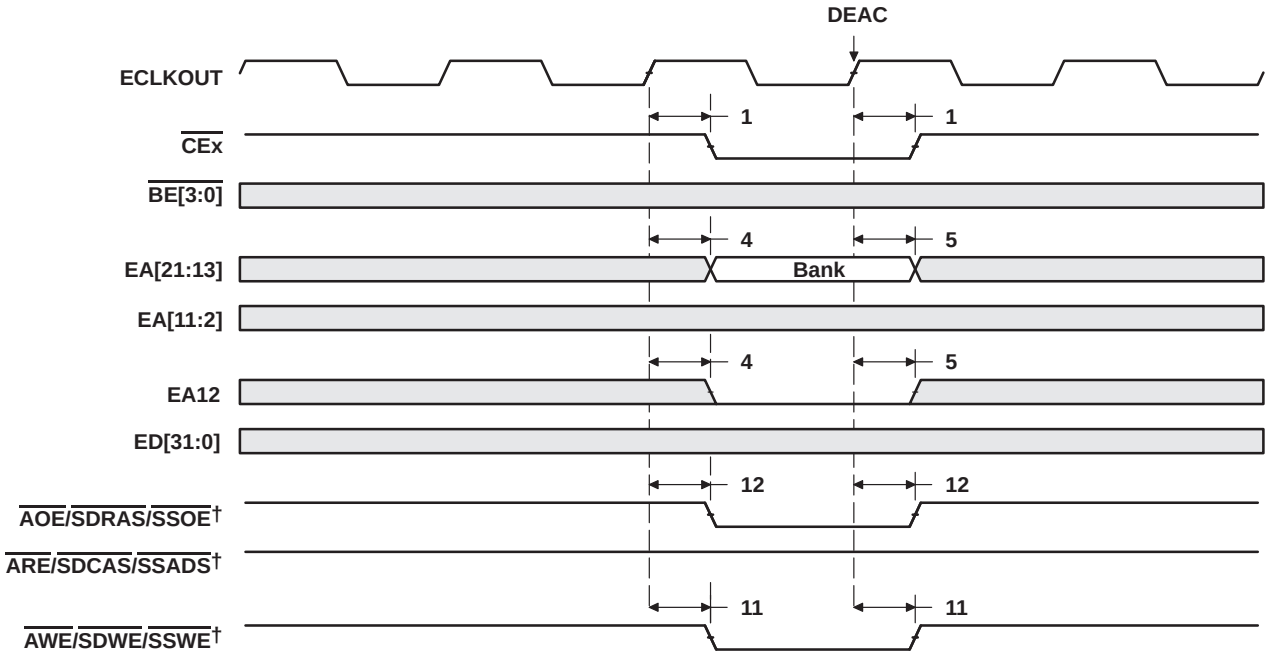


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 21. SDRAM DCAB Command

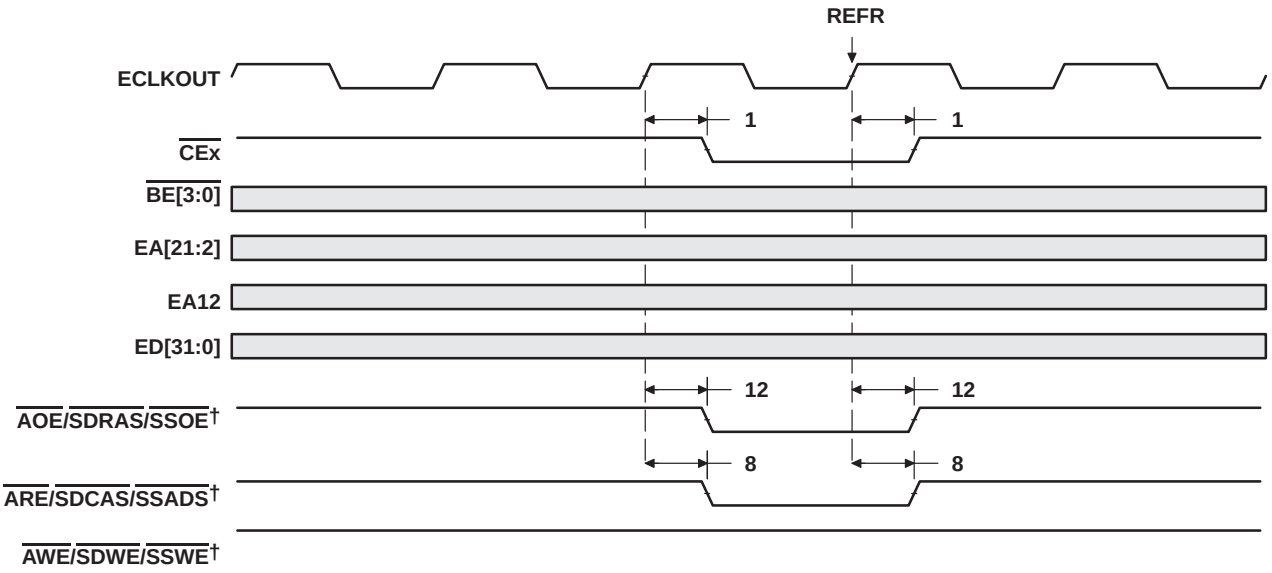
ADVANCE INFORMATION

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 22. SDRAM DEAC Command

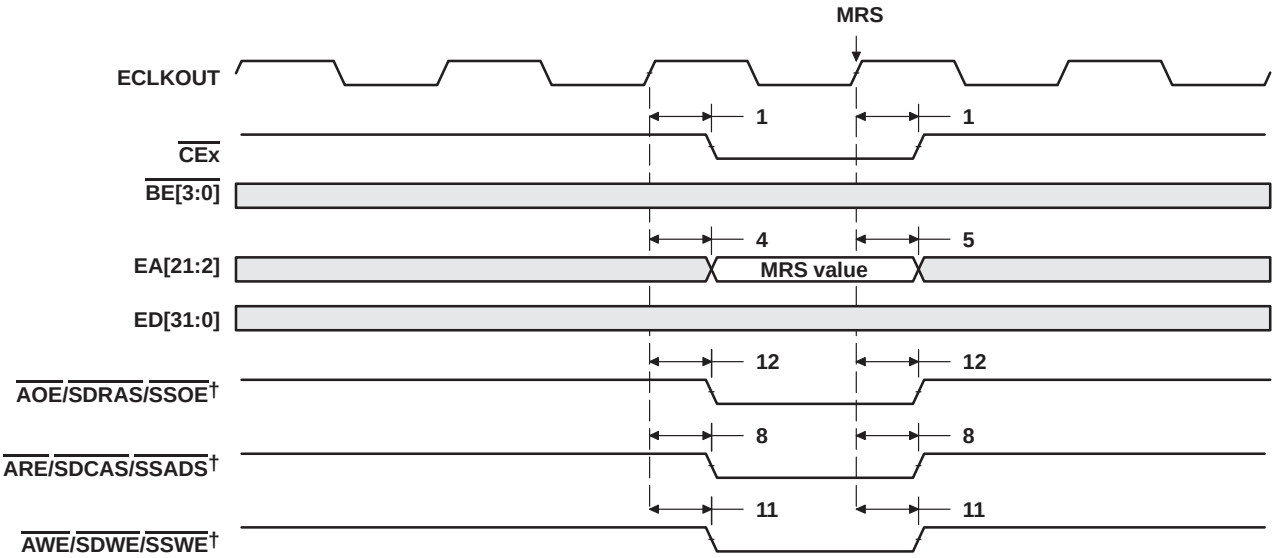


† ARE/SDCAS/SSADS, AWE/SDWE/SSWE, and AOE/SDRAS/SSOE operate as SDCAS, SDWE, and SDRAS, respectively, during SDRAM accesses.

Figure 23. SDRAM REFR 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 24. SDRAM MRS Command

ADVANCE INFORMATION

HOLD/HOLDA TIMING

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

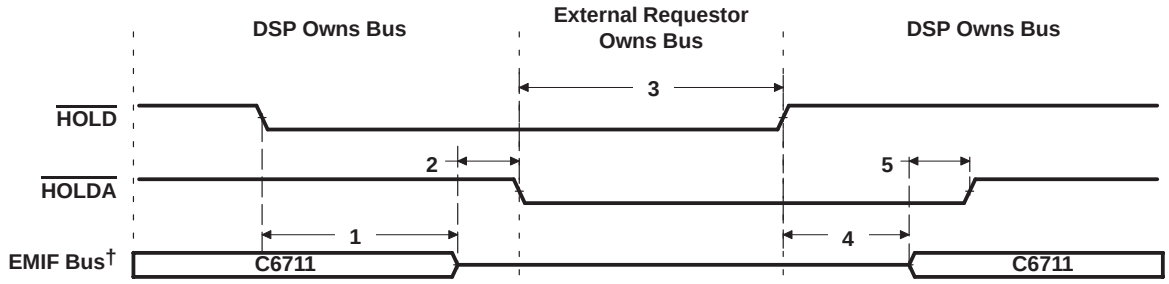
NO.		’C6711-100 ’C6711-150		UNIT
		MIN	MAX	
3	$t_{oh(HOLDAL-HOLD}$ Hold time, HOLD low after HOLDA low	E		ns

† E = ECLKIN period in ns

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

NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{R(HOLDL-EMHZ)}$	Response time, HOLD low to EMIF Bus high impedance	2E	§	ns
2	$t_{d(EMHZ-HOLDAL)}$	Delay time, EMIF Bus high impedance to HOLDA low	0	2E	ns
4	$t_{R(HOLDH-EMLZ)}$	Response time, HOLD high to EMIF Bus low impedance	2E	7E	ns
5	$t_{d(EMLZ-HOLDAH)}$	Delay time, EMIF Bus low impedance to HOLDA high	0	2E	ns

† E = ECLKIN period in ns
‡ EMIF Bus consists of CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE/SDCAS/SSADS, AOE/SDRAS/SSOE, and AWE/SDWE/SSWE.
§ All pending EMIF transactions are allowed to complete before 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 CE[3:0], BE[3:0], ED[31:0], EA[21:2], ARE/SDCAS/SSADS, AOE/SDRAS/SSOE, and AWE/SDWE/SSWE.

Figure 25. HOLD/HOLDA Timing

ADVANCE INFORMATION

RESET TIMING

timing requirements for reset (see Figure 26)

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{w(RST)}$	Width of the $\overline{\text{RESET}}$ pulse (PLL stable)	10		CLKOUT1 cycles
		Width of the $\overline{\text{RESET}}$ pulse (PLL needs to sync up)†	250		µs

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

switching characteristics during reset‡§ (see Figure 26)

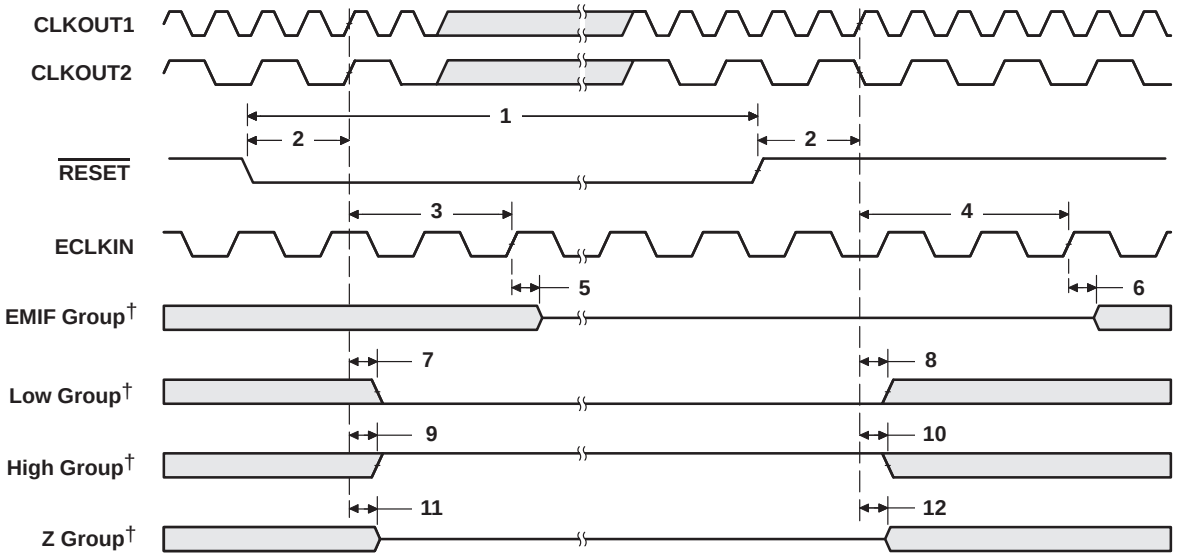
NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
2	$t_{R(RST)}$	Response time to change of value in $\overline{\text{RESET}}$ signal	2		CLKOUT1 cycles
3	$t_{R(RSTL-ECKI)}$	Response time, $\overline{\text{RESET}}$ active to ECLKIN synchronized	2E		ns
4	$t_{R(RSTH-ECKI)}$	Response time, $\overline{\text{RESET}}$ inactive to ECLKIN synchronized	3E		ns
5	$t_{d(ECKIH-EMIFHZ)}$	Delay time, ECLKIN high to EMIF group high impedance	–1	10	ns
6	$t_{d(ECKIH-EMIFV)}$	Delay time, ECLKIN high to EMIF group valid	–1	10	ns
7	$t_{d(CKO1H-LOWIV)}$	Delay time, CLKOUT1 high to low group invalid	–1	10	ns
8	$t_{d(CKO1H-LOWV)}$	Delay time, CLKOUT1 high to low group valid	–1	10	ns
9	$t_{d(CKO1H-HIGHIV)}$	Delay time, CLKOUT1 high to high group invalid	–1	10	ns
10	$t_{d(CKO1H-HIGHV)}$	Delay time, CLKOUT1 high to high group valid	–1	10	ns
11	$t_{d(CKO1H-ZHZ)}$	Delay time, CLKOUT1 high to Z group high impedance	–1	10	ns
12	$t_{d(CKO1H-ZV)}$	Delay time, CLKOUT1 high to Z group valid	–1	10	ns

‡ E = ECLKIN period in ns

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

Figure 26. Reset Timing

ADVANCE INFORMATION

EXTERNAL INTERRUPT/RESET TIMING

timing requirements for external interrupts† (see Figure 27)

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
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

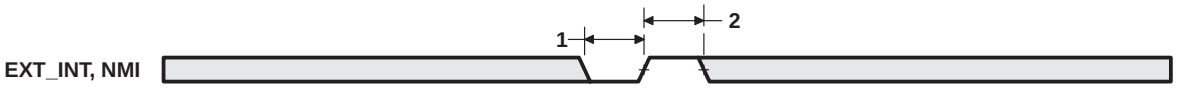


Figure 27. External/NMI Interrupt Timing

HOST-PORT INTERFACE TIMING

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

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{su(SELV-HSTBL)}$	Setup time, select signals§ valid before $\overline{\text{HSTROBE}}$ low	5		ns
2	$t_{h(HSTBL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HSTROBE}}$ low	2		ns
3	$t_{w(HSTBL)}$	Pulse duration, $\overline{\text{HSTROBE}}$ low	4P		ns
4	$t_{w(HSTBH)}$	Pulse duration, $\overline{\text{HSTROBE}}$ high between consecutive accesses	4P		ns
10	$t_{su(SELV-HASL)}$	Setup time, select signals§ valid before $\overline{\text{HAS}}$ low	5		ns
11	$t_{h(HASL-SELV)}$	Hold time, select signals§ valid after $\overline{\text{HAS}}$ low	2		ns
12	$t_{su(HDV-HSTBH)}$	Setup time, host data valid before $\overline{\text{HSTROBE}}$ high	5		ns
13	$t_{h(HSTBH-HDV)}$	Hold time, host data valid after $\overline{\text{HSTROBE}}$ high	2		ns
14	$t_{h(HRDYL-HSTBL)}$	Hold time, $\overline{\text{HSTROBE}}$ low after $\overline{\text{HRDY}}$ low. $\overline{\text{HSTROBE}}$ should not be inactivated until $\overline{\text{HRDY}}$ is active (low); otherwise, HPI writes will not complete properly.	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 150 MHz, use P = 6.7 ns.
§ Select signals include: HCNTL[1:0], HR/$\overline{\text{W}}$, and HHWIL.

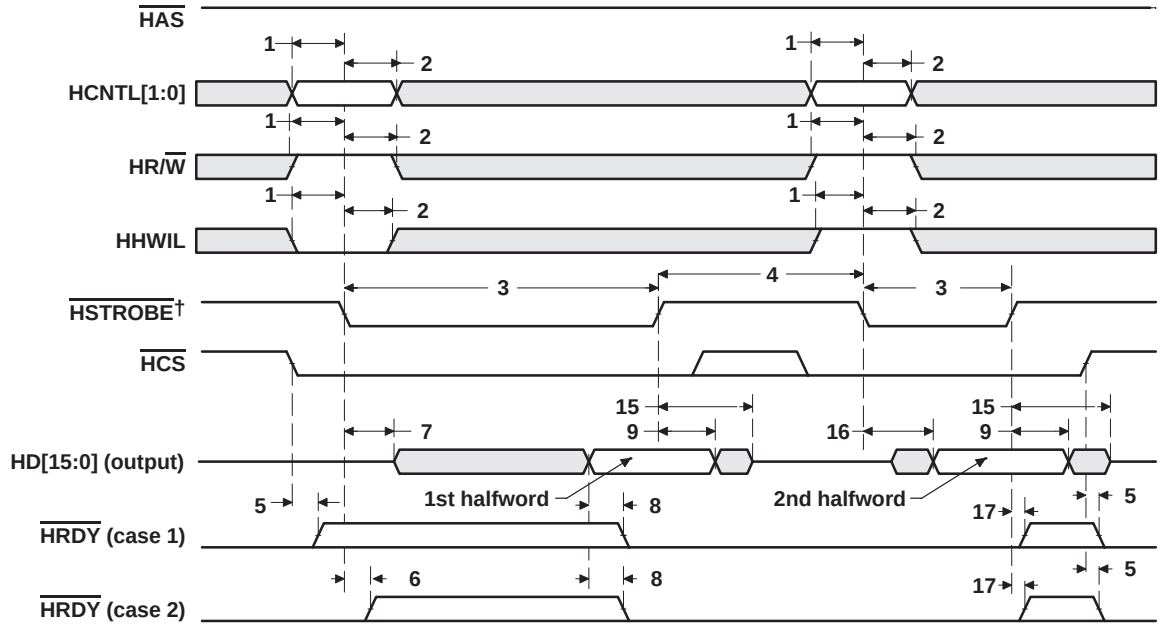
switching characteristics during host-port interface cycles†‡ (see Figure 28, Figure 29, Figure 30, and Figure 31)

NO.	PARAMETER		’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
5	$t_{d(HCS-HRDY)}$	Delay time, $\overline{\text{HCS}}$ to $\overline{\text{HRDY}}$¶	1	7	ns
6	$t_{d(HSTBL-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ low to $\overline{\text{HRDY}}$ high#	3	12	ns
7	$t_{oh(HSTBL-HDLZ)}$	Output hold time, HD low impedance after $\overline{\text{HSTROBE}}$ low for an HPI read	2		ns
8	$t_{d(HDV-HRDYL)}$	Delay time, HD valid to $\overline{\text{HRDY}}$ low	2P – 4	2P	ns
9	$t_{oh(HSTBH-HDV)}$	Output hold time, HD valid after $\overline{\text{HSTROBE}}$ high	3	12	ns
15	$t_{d(HSTBH-HDHZ)}$	Delay time, $\overline{\text{HSTROBE}}$ high to HD high impedance	3	12	ns
16	$t_{d(HSTBL-HDV)}$	Delay time, $\overline{\text{HSTROBE}}$ low to HD valid	3	12	ns
17	$t_{d(HSTBH-HRDYH)}$	Delay time, $\overline{\text{HSTROBE}}$ high to $\overline{\text{HRDY}}$ high‖	3	12	ns

† $\overline{\text{HSTROBE}}$ refers to the following logical operation on $\overline{\text{HCS}}$, $\overline{\text{HDS1}}$, and $\overline{\text{HDS2}}$: [NOT($\overline{\text{HDS1}}$ XOR $\overline{\text{HDS2}}$)] OR $\overline{\text{HCS}}$.
‡ The effects of internal clock jitter are included at test. There is no need to adjust timing numbers for internal clock jitter. P = 1/CPU clock frequency in ns. For example, when running parts at 150 MHz, use P = 6.7 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.

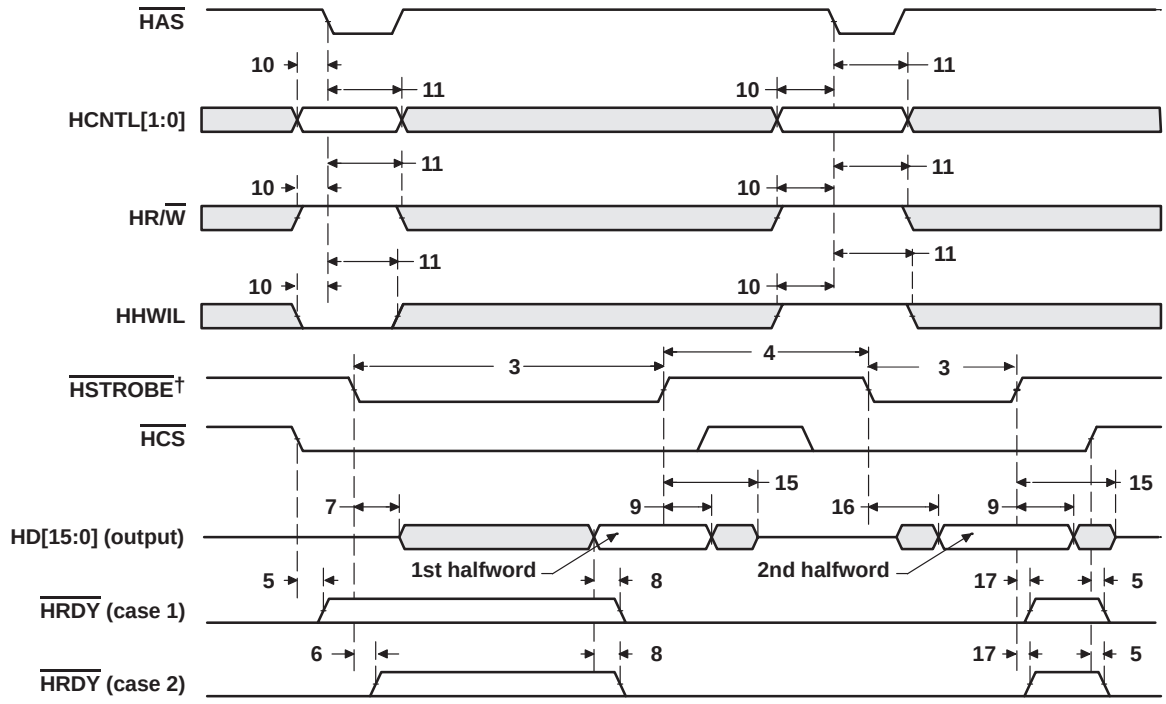
ADVANCE INFORMATION

HOST-PORT INTERFACE TIMING (CONTINUED)



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

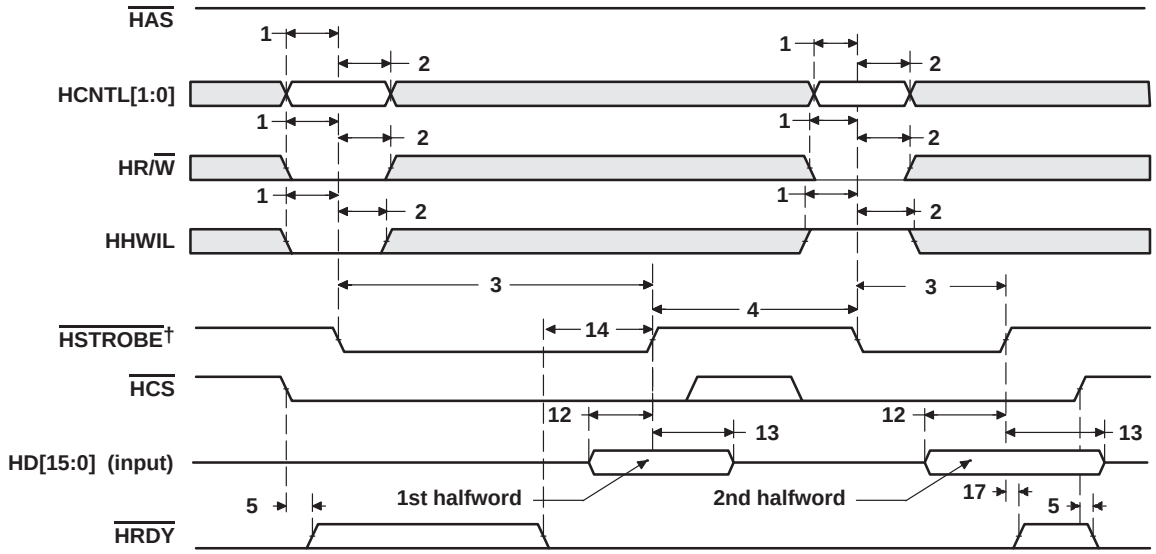
Figure 28. HPI Read Timing (HAS Not Used, Tied High)



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

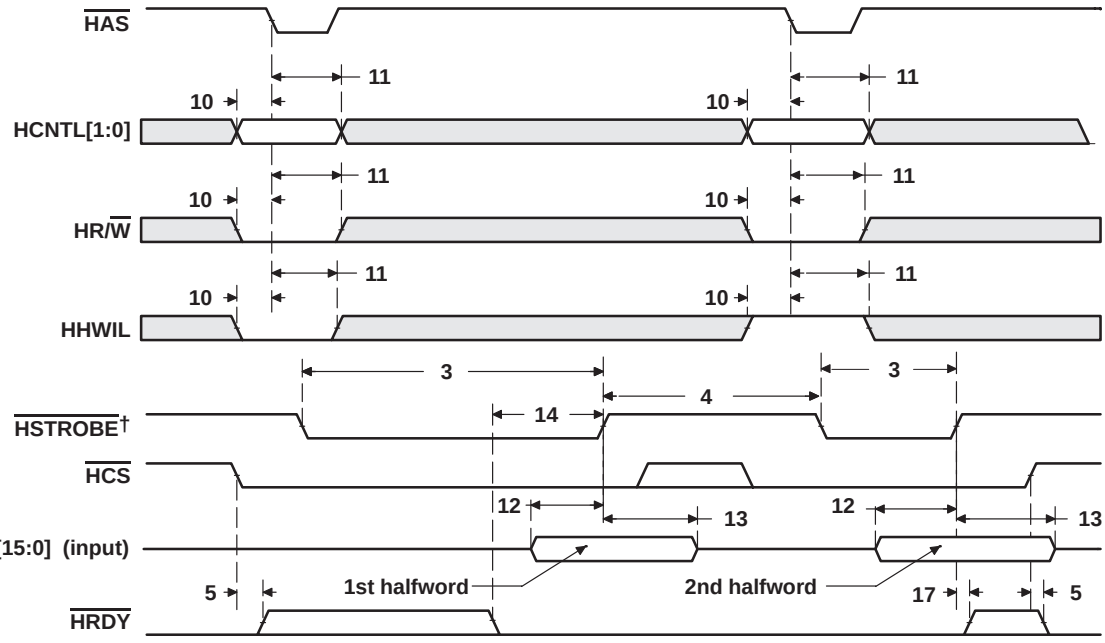
Figure 29. HPI Read Timing (HAS Used)

HOST-PORT INTERFACE TIMING (CONTINUED)



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

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



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

Figure 31. HPI Write Timing (HAS Used)

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING

timing requirements for McBSP†‡ (see Figure 32)

NO.				’C6711-100 ’C6711-150		UNIT
				MIN	MAX	
2	$t_{c(CKRX)}$	Cycle time, CLKR/X	CLKR/X ext	2P		ns
3	$t_{w(CKRX)}$	Pulse duration, CLKR/X high or CLKR/X low	CLKR/X ext	P – 1		ns
5	$t_{su(FRH-CKRL)}$	Setup time, external FSR high before CLKR low	CLKR int	9		ns
			CLKR ext	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. For example, when running parts at 150 MHz, use P = 6.7 ns.

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

switching characteristics for McBSP†‡ (see Figure 32)

NO.	PARAMETER			'C6711-100 'C6711-150		UNIT
				MIN	MAX	
1	$t_{d(CKSH-CKRXH)}$	Delay time, CLKS high to CLKR/X high for internal CLKR/X generated from CLKS input		4	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.
¶ 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
# 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

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

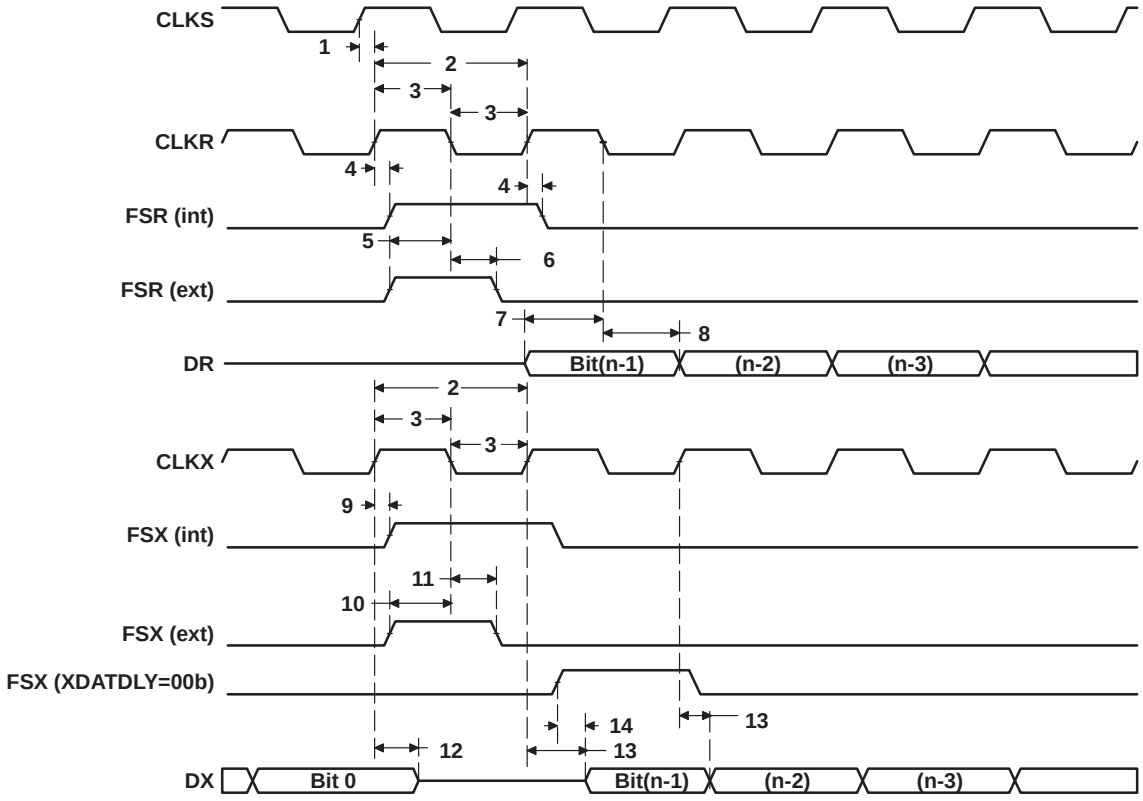


Figure 32. McBSP Timings

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			’C6711-100 ’C6711-150		UNIT
			MIN	MAX	
1	$t_{su(FRH-CKSH)}$	Setup time, FSR high before CLKS high	4		ns
2	$t_{h(CKSH-FRH)}$	Hold time, FSR high after CLKS high	4		ns

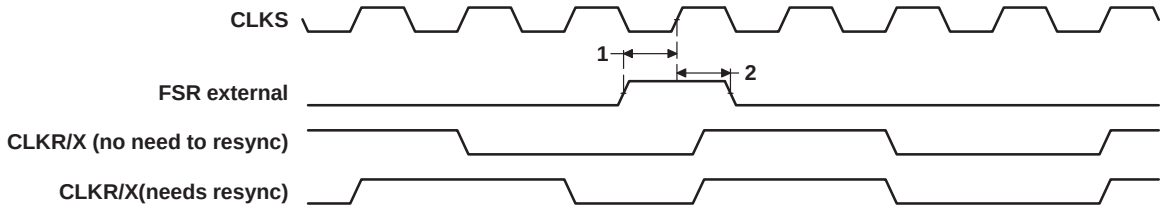


Figure 33. FSR Timing When GSYNC = 1

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			'C6711-100 'C6711-150				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

† 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 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 34)

NO.	PARAMETER		'C6711-100 'C6711-150				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

† 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 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)

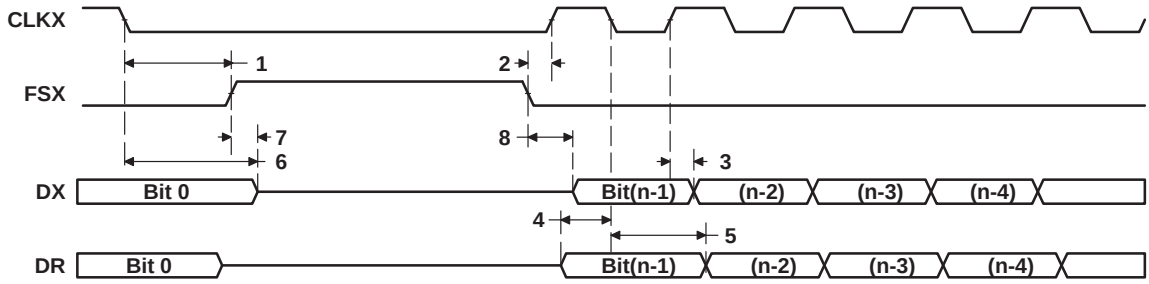


Figure 34. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 0

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			’C6711-100 ’C6711-150				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

† 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 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 35)

NO.	PARAMETER		’C6711-100 ’C6711-150				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

† 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 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).

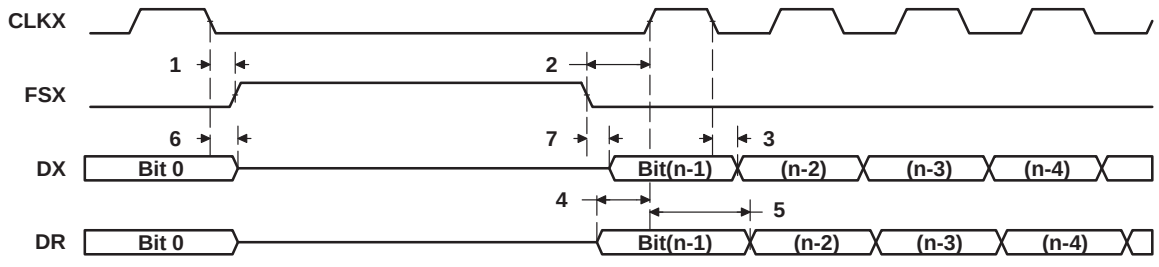


Figure 35. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 0

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			’C6711-100 ’C6711-150				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

† 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 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 36)

NO.	PARAMETER		’C6711-100 ’C6711-150				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

† 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 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)

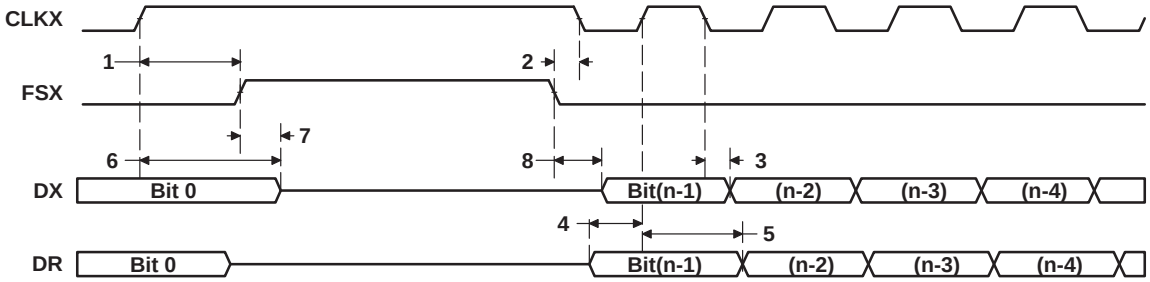


Figure 36. McBSP Timing as SPI Master or Slave: CLKSTP = 10b, CLKXP = 1

ADVANCE INFORMATION

MULTICHANNEL BUFFERED SERIAL PORT TIMING (CONTINUED)

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

NO.			’C6711-100 ’C6711-150				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

† 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 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 37)

NO.	PARAMETER		’C6711-100 ’C6711-150				UNIT
			MASTER§		SLAVE		
			MIN	MAX	MIN	MAX	
1	$t_{h(CKXH-FXL)}$	Hold time, FSX low after CLKX high¶	H – 2	H + 3			ns
2	$t_{d(FXL-CKXL)}$	Delay time, FSX low to CLKX low#	T – 2	T + 1			ns
3	$t_{d(CKXH-DXV)}$	Delay time, CLKX high to DX valid	–2	4	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

† 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 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)

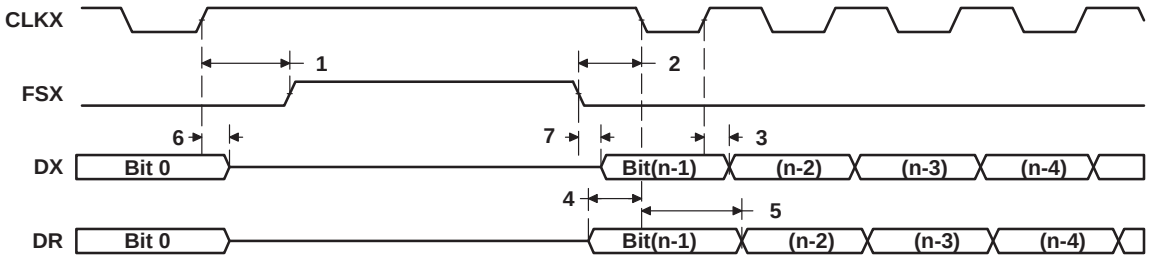


Figure 37. McBSP Timing as SPI Master or Slave: CLKSTP = 11b, CLKXP = 1

TIMER TIMING

timing requirements for timer inputs† (see Figure 38)

NO.			'C6711-100 'C6711-150		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 38)

NO.	PARAMETER		'C6711-100 'C6711-150		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.

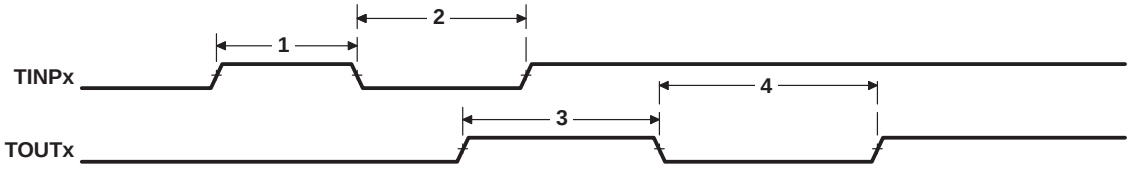


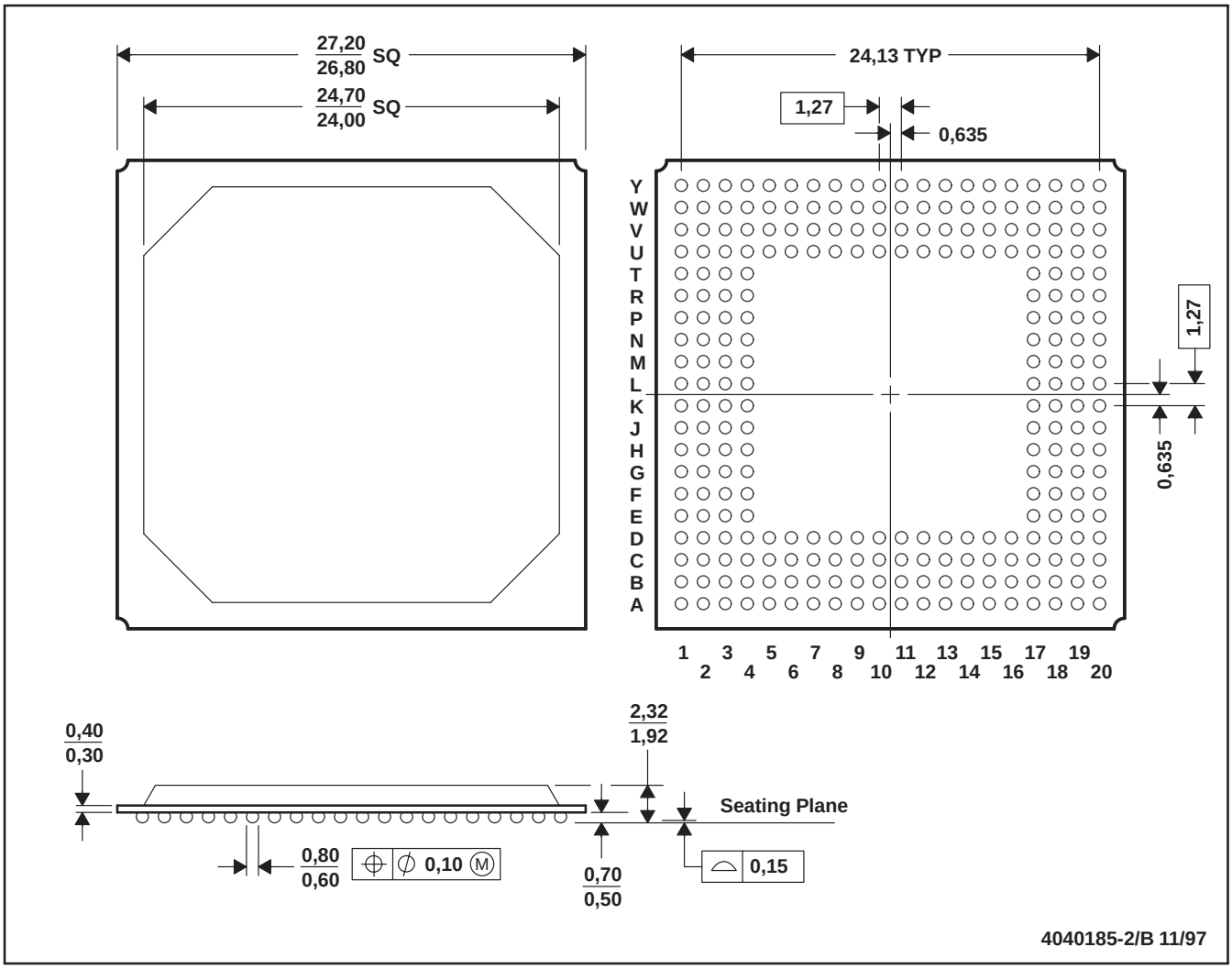
Figure 38. Timer Timing

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 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.