- **Powerful 16-Bit TMS320C511A CPU**
- **20-, 20.8-, and 21.7-ns Single-Cycle Instruction Execution Time With 5-V Operation**
- **Single-Cycle 16 × 16-Bit Multiply/Add**
- **128K Words of Total Data/Program Space**
- **6 × 4K × 16-Bit Single-Access On-Chip Program ROM**
- **1K × 16-Bit Dual-Access On-Chip Program/Data RAM**
- **Full-Duplex Synchronous Serial Port for Code/Decode (CODEC) Interface**
- **Hardware or Software Wait-State Generation Capability**
- **Repeat Instructions for Efficient Use of Program Space**
- **Multiply-by-Two and Divide-by-Two Clocking Options**
- **Block Moves for Data/Program Management**
- **On-Chip Scan-Based Emulation Logic**
- **100-Pin Quad Flat Package (PJ Suffix) and 100-Pin Thin Quad Flat Package (PZ Suffix)**
- **Low-Power Dissipation and Power-Down Modes:**
  - **47 mA (2.35 mA/MIPS) at 5 V, 40-MHz Clock (Average)**
  - **3 mA at 5 V, 40-MHz Clock (Typical IDLE2)**
  - **5 μA at 5 V, Clocks Off (Typical STANDBY)**
- **High-Performance Static CMOS Technology**
- **Databus Keepers**

description

The TMS320C511A is a member of the ’C5x generation of the Texas Instruments (TI™) TMS320 digital signal processors (DSPs). This device is fabricated with static CMOS integrated circuit technology, and its architectural design is based on that of an earlier TI DSP, the TMS320C25. The combination of advanced Harvard architecture, on-chip peripherals, on-chip memory, and a highly specialized instruction set is the basis of the operational flexibility and speed of the ’C511A device. The ’C511A executes up to 50 MIPS (million instructions per second).

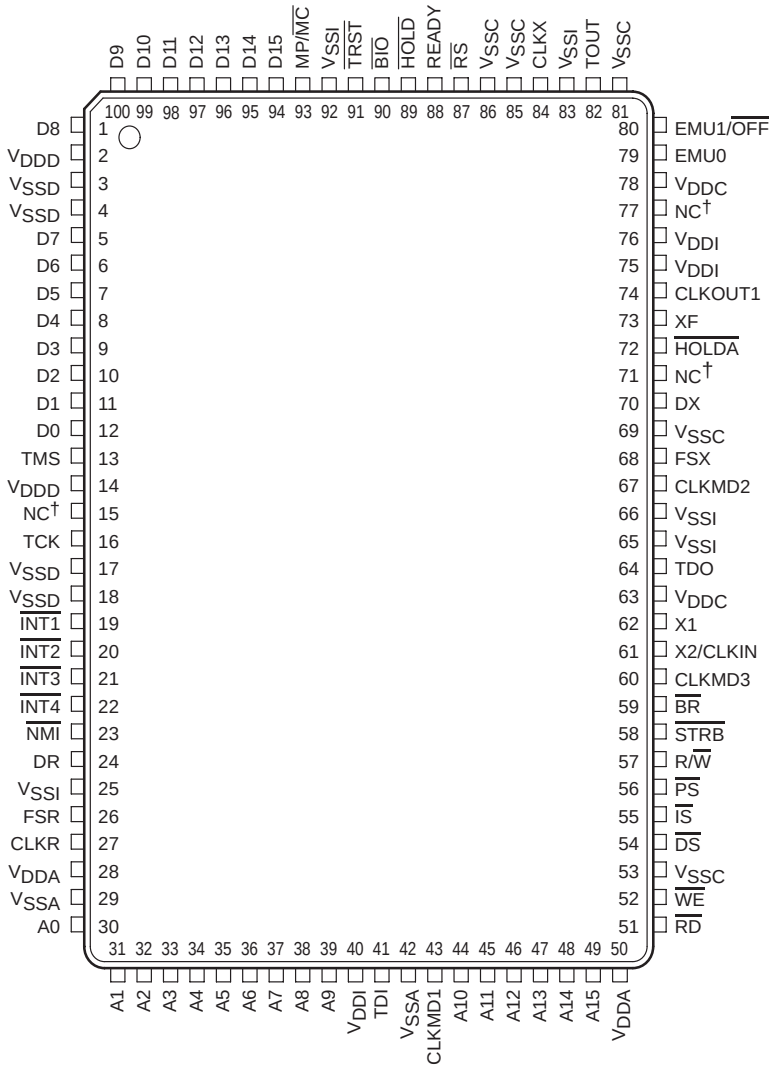
The ’C5x generation DSPs, like the ’C511A, offer these advantages:

- Enhanced TMS320 architecture for increased performance and versatility
- Modular architecture for fast development of spin-off devices
- Advanced integrated-circuit processing technology for increased performance
- Source code for ’C1x and ’C2x DSPs is upward-compatible with ’C5x generation devices like the ’C511A
- Enhanced TMS320 instruction set for faster algorithms and for optimized high-level language operation
- New static-design techniques for minimizing power consumption and maximizing radiation hardness

Table 1 lists the characteristics of the ’C511A processor: the capacity of on-chip RAM and ROM, the number of serial and parallel I/O ports, the execution time of one machine cycle, and the type of package with total pin count.

description (continued)

TMS320C511A – PJ PACKAGE
(TOP VIEW)

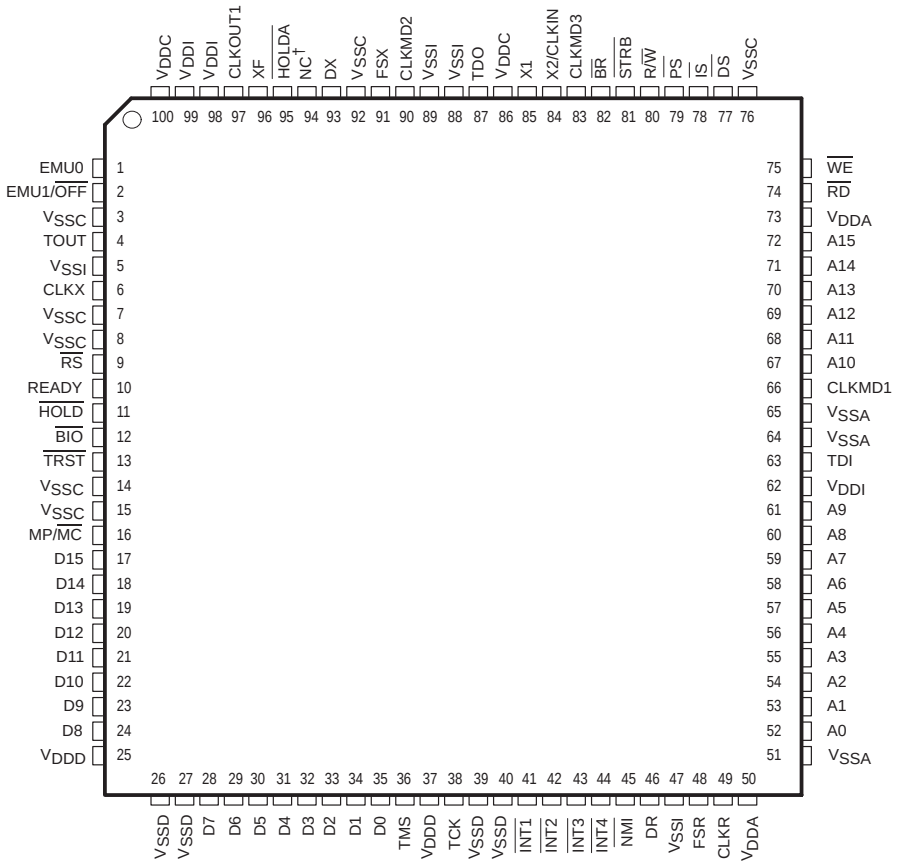


† NC = No connect (this pin is reserved)

description (continued)

TMS320C511A - PZ PACKAGE

(TOP VIEW)



†NC = No connect (this pin is reserved)

description (continued)

Table 1. Characteristics of the ’C511A Processor

DEVICE NAME	ON-CHIP MEMORY			I/O PORTS		CYCLE TIME (ns)	PACKAGE TYPES QFP‡
	RAM		ROM				
	DATA	DATA + PROG	PROG	SERIAL	PARALLEL†		
TMS320C511APJ TMS320C511APZ	544	512	24K	1	64K	21.7/20.8/20	100-pin

† Note that 16 of the 64K parallel I/O ports are memory-mapped.
‡ QFP = Quad flatpack

Pin Functions for a TMS320C511A Device in the PJ/PZ Package

SIGNAL	TYPE	DESCRIPTION
PARALLEL INTERFACE BUS		
A15–A0	I/O/Z	16-bit external address bus [most significant bit (MSB): A15; least significant bit (LSB): A0]
D15–D0	I/O/Z	16-bit external data bus (MSB: D15, LSB: D0)
$\overline{PS}$, $\overline{DS}$, $\overline{IS}$	O/Z	Program, data, and I/O space-select outputs, respectively
$\overline{STRB}$	I/O/Z	Timing strobe for external cycles and external direct memory access (DMA)
R/$\overline{W}$	I/O/Z	Read/write-select for external cycles and external DMA
$\overline{RD}$, $\overline{WE}$	O/Z	Read and write strobes, respectively, for external cycles
READY	I	External bus ready/wait-state control input
$\overline{BR}$	I/O/Z	Bus request. $\overline{BR}$ arbitrates global memory and external DMA.
SYSTEM INTERFACE/CONTROL SIGNALS		
$\overline{RS}$	I	Reset. $\overline{RS}$ initializes the device and sets the program counter (PC) to zero.
MP/$\overline{MC}$	I	Microprocessor/microcomputer mode select. MP/$\overline{MC}$ enables internal ROM.
$\overline{HOLD}$	I	$\overline{HOLD}$ puts the parallel interface (I/F) bus in the high-impedance state after the current cycle.
$\overline{HOLDA}$	O/Z	Hold acknowledge. $\overline{HOLDA}$ indicates that the external bus is in hold state.
XF	O/Z	External flag output. XF is set/cleared through software.
$\overline{BIO}$	I	I/O branch input. $\overline{BIO}$ implements conditional branches.
TOUT	O/Z	Timer output signal. TOUT indicates output of the internal timer.
$\overline{INT1}$–$\overline{INT4}$	I	External interrupt inputs
$\overline{NMI}$	I	Nonmaskable external interrupt
SERIAL PORT INTERFACE		
DR	I	Serial receive-data input
DX	O/Z	Serial transmit-data output. DX is in the high-impedance state when not transmitting.
CLKR	I	Serial receive-data clock input
CLKX	I/O/Z	Serial transmit-data clock. Internal or external source
FSR	I	Serial receive-frame-synchronization input
FSX	I/O/Z	Serial transmit-frame-synchronization signal. Internal or external source

Legend:
I = Input
O = Output
Z = High impedance
S = Supply

Pin Functions for a TMS320C511A Device in the PJ/PZ Package (Continued)

SIGNAL	TYPE	DESCRIPTION
EMULATION/IEEE 1149.1 (JTAG) INTERFACE		
TDI	I	IEEE 1149.1 test-access-port scan data input
TDO	O/Z	IEEE 1149.1 test-access-port scan data output
TMS	I	IEEE 1149.1 test-access-port mode-select input
TCK	I	IEEE 1149.1 test-access-port clock input
$\overline{\text{TRST}}$	I	IEEE 1149.1 test-access-port reset (with pulldown resistor). $\overline{\text{TRST}}$ disables JTAG when low.
EMU0	I/O/Z	Emulation control 0. EMU0 is reserved for emulation use.
EMU1/$\overline{\text{OFF}}$	I/O/Z	Emulation control 1. EMU1/$\overline{\text{OFF}}$ puts outputs in the high-impedance state when low.
CLOCK GENERATION AND CONTROL		
X1	O	Divide-by-two oscillator output
X2/CLKIN	I	Divide-by-two clock/oscillator input
CLKMD1, CLKMD2, CLKMD3	I	Clock-mode select inputs
CLKOUT1	O/Z	Device system-clock output
POWER SUPPLY CONNECTIONS		
V_{DDA}	S	Supply connection, address-bus output
V_{DDD}	S	Supply connection, data-bus output
V_{DDC}	S	Supply connection, control output
V_{DDI}	S	Supply connection, internal logic
V_{SSA}	S	Supply connection, address-bus output
V_{SSD}	S	Supply connection, data-bus output
V_{SSC}	S	Supply connection, control output
V_{SSI}	S	Supply connection, internal logic

Legend:
I = Input
O = Output
Z = High impedance
S = Supply

architecture

The ’C511A’s advanced Harvard-type architecture maximizes processing power by maintaining two memory bus structures — program and data — for full-speed execution. Instructions support data transfers between the two spaces. This architecture permits coefficients that are stored in program memory to be read into the RAM, thereby eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

Increased throughput on the ’C511A for many DSP applications is accomplished by means of single-cycle multiply/accumulate instructions with a data-move option, up to eight auxiliary registers with a dedicated arithmetic unit, and faster I/O necessary for data-intensive signal processing.

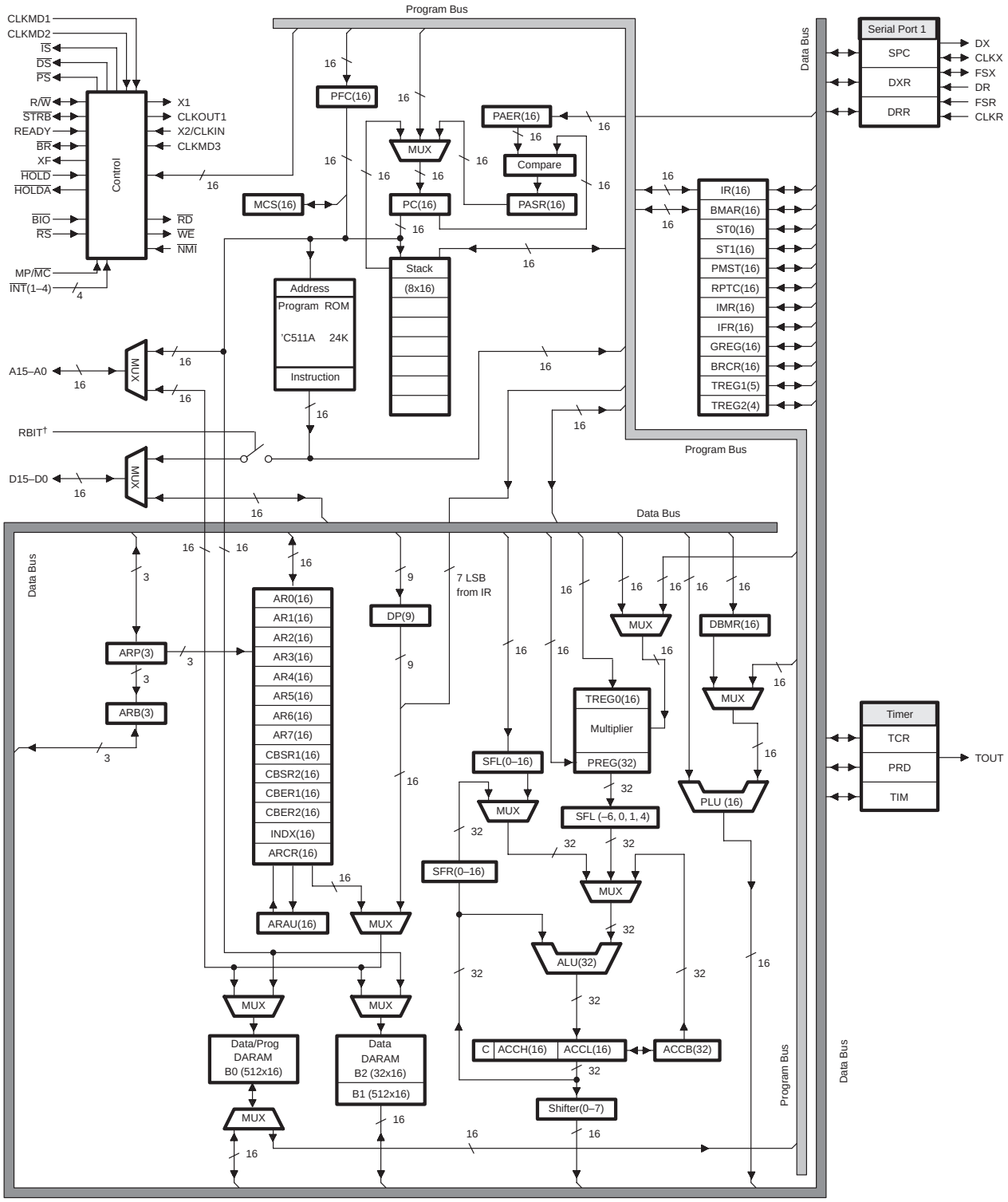
The architecture emphasizes overall speed, communication, and flexibility in processor configuration. Control signals and instructions provide floating-point support, block-memory transfers, communication to slower off-chip devices, and multiprocessing implementations (see the functional block diagram).

Table 2 explains the symbols that are used in the functional block diagram.

Table 2. Symbols Used in Functional Block Diagram

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
ABU	Auto-buffering unit	IMR	Interrupt-mask register
ACCB	Accumulator buffer	INDX	Indirect-addressing-index register
ACCH	Accumulator high	IR	Instruction register
ACCL	Accumulator low	MCS	Microcall stack
ALU	Arithmetic logic unit	MUX	Multiplexer
ARAU	Auxiliary-register arithmetic unit	PAER	Block-repeat-address end register
ARB	Auxiliary-register pointer buffer	PASR	Block-repeat-address start register
ARCR	Auxiliary-register compare register	PC	Program counter
ARP	Auxiliary-register pointer	PFC	Prefetch counter
AR0–AR7	Auxiliary registers	PLU	Parallel logic unit
BMAR	Block-move-address register	PMST	Processor-mode status register
BRCR	Block-repeat-counter register	PRD	Timer-period register
C	Carry bit	PREG	Product register
CBER1	Circular buffer 1 end address	RPTC	Repeat-counter register
CBER2	Circular buffer 2 end address	SFL	Left-shifter
CBSR1	Circular buffer 1 start address	SFR	Right-shifter
CBSR2	Circular buffer 2 start address	SPC	Serial-port interface-control register
DARAM	Dual-access RAM	ST0,ST1	Status registers
DBMR	Dynamic bit manipulation register	TCR	Timer-control register
DP	Data memory page pointer	TIM	Timer-count register
DRR	Serial-port data receive register	TREG0	Temporary register for multiplication
DXR	Serial-port data transmit register	TREG1	Temporary register for dynamic shift count
GREG	Global-memory allocation register	TREG2	Temporary register used as bit pointer in dynamic-bit test
IFR	Interrupt-flag register		

functional block diagram



NOTE A: Symbol descriptions appear in Table 2.
† ROM protection feature

32-bit ALU/accumulator

The 32-bit arithmetic logic unit (ALU) and accumulator (ACC) implement a wide range of arithmetic and logical functions, the majority of which execute in a single clock cycle. The ALU is a general-purpose arithmetic/logic unit that operates on 16-bit words taken from data memory or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, facilitating the bit manipulation ability required of a high-speed controller. One input to the ALU is always supplied by the ACC, and the other input can be furnished from the product register (PREG) of the multiplier, the accumulator buffer (ACCB), or the output of the scaling shifter (which has been read from data memory or from the ACC). After the ALU performs the arithmetic or logical operation, the result is stored in the ACC where additional operations, such as shifting, can be performed. Data input to the ALU can be scaled by the scaling shifter. The 32-bit ACC is split into two 16-bit segments for storage in data memory. Shifters at the output of the ACC provide a left shift of 0 to 7 places. This shift is performed while the data is being transferred to the data bus for storage. The contents of the ACC remain unchanged. When the postscaling shifter is used on the high word of the ACC (bits 31–16), the most significant bits (MSBs) are lost and the least significant bits (LSBs) are filled with bits shifted in from the low word (bits 15–0). When the postscaling shifter is used on the low word, the LSBs are filled with zeros.

The ’C511A supports floating-point operations for applications requiring a large dynamic range. By performing left shifts, the normalization instruction (NORM) is used to normalize fixed-point numbers contained in the ACC. The four bits of TREG1 (the temporary register for dynamic shift count) define a variable shift through the scaling shifter for the ADDT/LACT/SUBT instructions (add to/load to/subtract from ACC with shift specified by TREG1). These instructions are useful in denormalizing a number (that is, converting from floating point to fixed point). They are also useful for executing an automatic gain control (AGC) going into a filter.

The single-cycle 1-bit to 16-bit right shift of the ACC efficiently aligns the ACC’s contents. This, coupled with the 32-bit temporary buffer on the ACC, enhances the effectiveness of the ALU in extended-precision arithmetic. The ACCB provides a temporary storage place for a fast save of the ACC. The ACCB also can be used as an input to the ALU. The minimum or maximum value in a string of numbers is found by comparing the contents of the ACCB with the contents of the ACC. The minimum or maximum value is placed in both registers, and, if the condition is met, the carry bit (C) is set to 1. The minimum and maximum functions are executed by the CRLT and CRGT instructions, respectively. See Table 4 for a list of ’C511A instructions.

scaling shifters

The ’C511A provides a scaling shifter that has a 16-bit input connected to the data bus and a 32-bit output connected to the ALU. This scaling shifter produces a left shift of 0 to 16 bits on the input data. The shift count is specified by a constant embedded in the instruction word or by the value in TREG1. The LSBs of the output are filled with zeros; the MSBs may either be filled with zeros or sign-extended, depending upon the value of the sign-extension mode (SXM) bit of status register ST1.

The ’C511A also contains several other shifters that allow it to perform numerical scaling, bit extraction, extended-precision arithmetic, and overflow prevention. These shifters are connected to the output of the product register and the ACC.

parallel logic unit

The parallel logic unit (PLU) is a second logic unit, additional to the main ALU, that executes logic operations on data without affecting the contents of the ACC. The PLU provides the bit-manipulation ability required of a high-speed controller and simplifies control/status register operations. The PLU provides a direct logic operation path to data memory space and can set, clear, test, or toggle multiple bits directly in a data memory location, a control/status register, or any register that is mapped into data memory space.

16 × 16-bit parallel multiplier

The ’C511A uses a 16 × 16-bit hardware multiplier that is capable of computing a signed or an unsigned 32-bit product in a single machine cycle. All multiply instructions, except the MPYU (multiply unsigned) instruction, perform a signed multiply operation in the multiplier. That is, two numbers being multiplied are treated as 2s-complement numbers, and the result is a 32-bit 2s-complement number.

There are two registers associated with the multiplier: TREG0, a 16-bit temporary register that holds one of the operands for the multiplier, and PREG, the 32-bit product register that holds the product. Four product-shift modes (PM) are available at the PREG’s output. These shift modes are useful for performing multiply/accumulate operations, performing fractional arithmetic, or justifying fractional products. The PM field of status register ST1 specifies the PM.

The multiply instruction (MPY) allows the product to be shifted one bit to compensate for the extra sign bit gained in multiplying two 16-bit 2s-complement numbers. A 4-bit shift is used in conjunction with the MPY instruction with a short-immediate value (13 bits or less) to eliminate the four extra sign bits gained in multiplying a 16-bit number by a 13-bit number. Finally, the output of PREG can, instead, be right-shifted 6 bits to enable the execution of up to 128 consecutive multiply/accumulates without the possibility of overflow.

The load-TREG0 (LT) instruction normally loads TREG0 to provide one operand (from the data bus), and the MPY instruction provides the second operand (also from the data bus). A multiplication also can be performed with a short- or long-immediate operand by using the MPY instruction with an immediate operand. A product is obtained every two cycles except when a long-immediate operand is used.

Four multiply/accumulate instructions (MAC, MACD, MADD, and MADS as defined in Table 4) fully utilize the computational bandwidth of the multiplier, allowing both operands to be processed simultaneously. The data for these operations is transferred to the multiplier during each cycle through the program and data buses. This facilitates single-cycle multiply/accumulates when used with repeat (RPT and RPTZ) instructions. In these instructions, the coefficient addresses are generated by the PC, while the data addresses are generated by the auxiliary register arithmetic unit. This allows the repeated instruction to access the values sequentially from the coefficient table and step through the data in any of the indirect addressing modes. The RPTZ instruction also clears the accumulator and the product register to initialize the multiply/accumulate operation.

The MACD and MADD instructions, when repeated, support filter constructs (weighted running averages) so that as the sum-of-products is executed, the sample data is shifted in memory to make room for the next sample and to eliminate the oldest sample. Circular addressing with MAC and MADS instructions also can be used to support filter implementation.

auxiliary registers and auxiliary-register arithmetic unit (ARAU)

The ’C511A provides a register file containing eight auxiliary registers (AR0–AR7). The auxiliary registers are used for indirect addressing of the data memory or for temporary data storage. Indirect auxiliary-register addressing allows placement of the data memory address of an instruction operand into one of the auxiliary registers. These registers are referenced with a 3-bit auxiliary-register pointer (ARP) that is loaded with a value from 0 through 7, designated AR0 through AR7, respectively. The auxiliary registers and the ARP can be loaded from data memory, the accumulator, the product register, or by an immediate operand defined in the instruction. The contents of these registers can be stored in data memory or used as inputs to the central arithmetic logic unit (CALU). These registers are accessible as memory-mapped locations within the ’C5x data-memory space.

The auxiliary register file (AR0–AR7) is connected to the ARAU. The ARAU can autoindex the current auxiliary register while the data memory location is being addressed. Indexing can be performed either by adding or subtracting 1 or by the contents of the INDX register. As a result, accessing tables of information does not require the CALU for address manipulation; therefore, the CALU is free for other operations in parallel.

memory

The ’C511A implements three separate address spaces for program memory, data memory, and input/output (I/O). Each space accommodates a total of 64K 16-bit words (see Figure 1). Within the 64K words of data space, the 256 to 32K words at the top of the address range can be defined to be external global memory in increments of powers of two, as specified by the contents of the global-memory allocation register (GREG). Access to global memory is arbitrated using the global memory bus request ($\overline{BR}$) signal.

The ’C511A device includes a considerable amount of on-chip memory to aid in system performance and integration including ROM and dual-access RAM (DARAM). Refer to Table 1 for the amount and types of memory available on this device.

On the ’C511A, the first 96 (0–5Fh) data-memory locations are allocated for memory-mapped registers. This memory-mapped register space contains various control and status registers including those for the CPU, serial port, timer, and software wait-state generators. Additionally, the first 16 I/O port locations are mapped into this data-memory space, allowing them to be accessed either as data memory using single-word instructions or as I/O locations with two-word instructions. Two-word instructions allow access to the full 64K words of I/O space.

The ’C511A contains 24K words of mask-programmable on-chip ROM located in program memory space. This ROM can be programmed with contents unique to to any particular application. The ROM is enabled or disabled by the state of the MP/$\overline{MC}$ control input upon resetting the device or by manipulating the MP/$\overline{MC}$ bit in the PMST status register after reset. The ROM occupies the first 24K words of internal program space (0–5FFFh) when enabled (which is the lowest block of program memory). When disabled, these addresses are located in the device’s external program-memory space.

The ’C511A also has a mask-programmable option that provides security protection for the contents of on-chip ROM. When this internal option bit is programmed, no externally-originating instruction can access the on-chip ROM. This feature can be used to provide security for proprietary algorithms.

The ’C511A also provides a total of 1 056 16-bit words of on-chip data RAM, divided into three separate blocks: block 0 (B0), block 1 (B1), and block 2 (B2). Of the 1 056 words, 544 words (blocks B1 and B2) are always data memory and 512 words (block B0) are programmable as either data or program memory. A data-memory size of 1056 words allows the ’C511A to handle a data array of 1 024 words (512 words if on-chip RAM is used for program memory) while still leaving 32 locations for intermediate storage. When using block B0 as program memory, instructions can be downloaded from external program memory into on-chip RAM and then executed.

The CLRC CNF (configure block B0 as data memory) and SETC CNF (configure block B0 as program memory) instructions allow dynamic configuration of the memory maps through software. Regardless of the configuration, code still can be executed from external program memory.

When using on-chip RAM, ROM, or high-speed external memory, the ’C511A runs at full speed with no wait states. The ability of the DARAM to allow two accesses to be performed in one cycle, coupled with the parallel nature of the ’C511A architecture, enables the device to perform three concurrent memory accesses in any given machine cycle. Externally, the READY line can be used to interface the ’C511A to slower, less expensive external memory. Downloading programs from slow off-chip memory to on-chip RAM can speed processing while cutting system costs.

memory (continued)

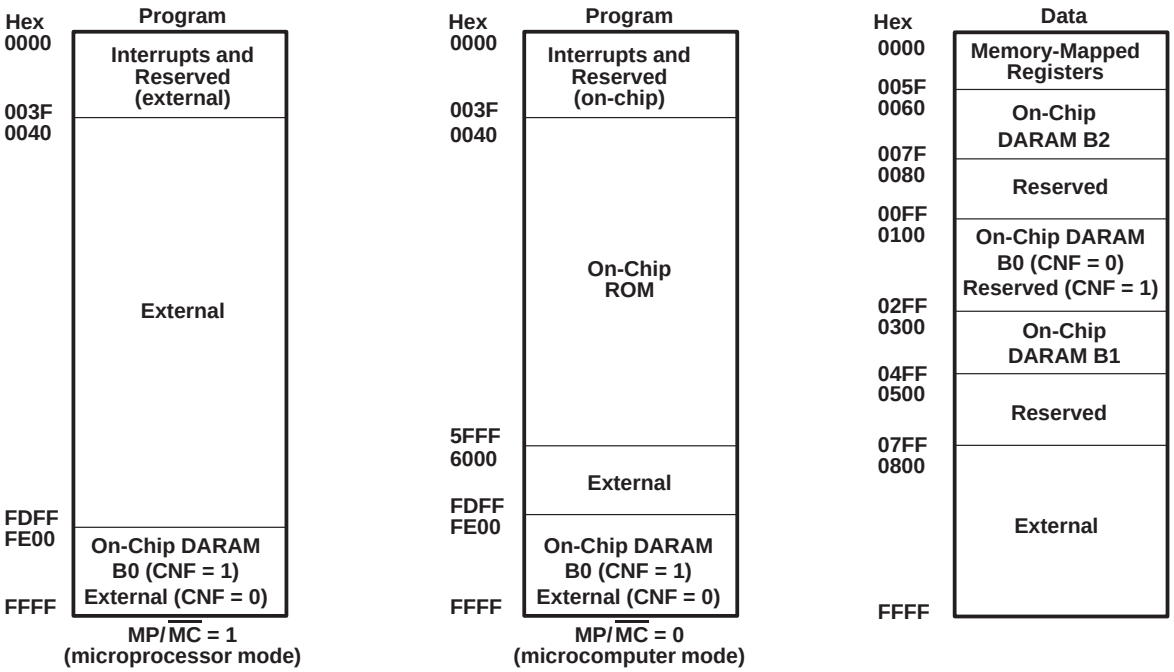


Figure 1. TMS320C511A Memory Map

interrupts and subroutines

The ’C511A implements four general-purpose interrupts, $\overline{\text{INT4}}$–$\overline{\text{INT1}}$, along with reset ($\overline{\text{RS}}$) and the nonmaskable interrupt ($\overline{\text{NMI}}$), which are available for external devices to request the attention of the processor. Internal interrupts are generated by the serial port (RINT and XINT), by the timer (TINT), and by the software-interrupt (TRAP, INTR, and NMI) instructions. Interrupts are prioritized with $\overline{\text{RS}}$ having the highest priority, followed by $\overline{\text{NMI}}$, and $\overline{\text{INT4}}$ having the lowest priority. Additionally, any interrupt except $\overline{\text{RS}}$ and $\overline{\text{NMI}}$ can be masked individually with a dedicated bit in the interrupt-mask register (IMR) and can be cleared, set, or tested using its own dedicated bit in the interrupt-flag register (IFR). The reset and NMI functions are not maskable.

All interrupt vector locations are on two-word boundaries so that branch instructions can be accommodated in those locations. While normally located at program memory address 0, the interrupt vectors can be remapped to the beginning of any 2K-word page in program memory by modifying the contents of the interrupt vector pointer (IPTR) located in the PMST status register.

A built-in mechanism protects multicycle instructions from interrupts. If an interrupt occurs during a multicycle instruction, the interrupt is not processed until the instruction completes execution. This mechanism applies to instructions that are repeated (using the RPT instruction) and to instructions that become multicycle because of wait states.

Each time an interrupt is serviced or a subroutine is entered, the PC is pushed onto an internal hardware stack, providing a mechanism for returning to the previous context. The stack contains eight locations, allowing interrupts or subroutines to be nested up to eight levels deep.

In addition to the eight-level hardware PC stack, eleven key CPU registers are equipped with an associated single-level stack or shadow register into which the registers’ contents are saved upon servicing an interrupt. The contents are restored into their particular CPU registers once a return-from-interrupt instruction (RETE or RETI) is executed. The registers that have the shadow-register feature include the ACC and buffer, product register, status registers, and several other key CPU registers. The shadow-register feature allows sophisticated context save and restore operations to be handled automatically in cases where nested interrupts are not required or if interrupt servicing is performed serially.

power-down modes

The ’C511A implements several power-down modes in which the ’C5x core enters a dormant state and dissipates considerably less power. A power-down mode is invoked either by executing the IDLE/IDLE2 instructions or by driving the $\overline{\text{HOLD}}$ input low. When the $\overline{\text{HOLD}}$ signal initiates the power-down mode, on-chip peripherals continue to operate; this power-down mode is terminated when $\overline{\text{HOLD}}$ goes inactive.

While the ’C511A is in a power-down mode, all internal contents are maintained; this allows operation to continue unaltered when the power-down mode is terminated. All CPU activities are halted when the IDLE instruction is executed, but the CLKOUT1 pin remains active. The peripheral circuits continue to operate, allowing peripherals such as serial ports and timers to take the CPU out of its powered-down state. A power-down mode, when initiated by an IDLE instruction, is terminated upon receipt of an interrupt.

The IDLE2 instruction is used for a complete shutdown of the core CPU as well as all on-chip peripherals. In IDLE2, the power is reduced significantly because the entire device is stopped. The power-down mode is terminated by activating any of the external interrupt pins ($\overline{\text{RS}}$, $\overline{\text{NMI}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$, $\overline{\text{INT3}}$, and $\overline{\text{INT4}}$) for at least five machine cycles.

bus-keeper circuitry

The ’C511A device provides built-in bus-keeper circuitry which holds the last state driven on the data bus by either the DSP or an external device after the bus is no longer being driven. This capability prevents excess power consumption caused by a floating bus, thereby allowing optimization of power consumption without the need for external pullup resistors.

external interface

The ’C511A supports a wide range of system-interfacing requirements. Program, data, and I/O address spaces provide interface to memory and I/O, maximizing system throughput. The full 16-bit address and data bus, along with the $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ space select signals, allow addressing of 64K 16-bit words in each of the three spaces.

Input/output design is simplified by having I/O treated the same way as memory. Input/output devices are mapped into the I/O address space using the processor’s external address and data buses in the same manner as memory-mapped devices.

The ’C511A external parallel interface provides various control signals to facilitate interfacing to the device. The R/$\overline{W}$ output signal is provided to indicate whether the current cycle is a read or a write. The $\overline{STRB}$ output signal provides a timing reference for all external cycles. For convenience, the device also provides the $\overline{RD}$ and the $\overline{WE}$ output signals, which indicate a read and a write cycle, respectively, along with timing information for those cycles. The availability of these signals minimizes external gating necessary for interfacing external devices to the ’C511A.

Interface to memory and I/O devices of varying speeds is accomplished by using the READY line. When transactions are made with slower devices, the ’C511A processor waits until the other device completes its function and signals the processor through the READY line. Once a ready indication is provided back to the ’C511A from the external device, execution continues.

The bus request ($\overline{BR}$) signal is used in conjunction with the other ’C511A interface signals to arbitrate external global-memory accesses. Global memory is external data-memory space in which the $\overline{BR}$ signal is asserted at the beginning of the access. When an external global-memory device receives the the bus request, the external device responds by asserting the READY signal after the global-memory access is arbitrated and completed.

serial port

The ’C511A provides a high-speed full-duplex serial port that allows direct interface with other ’C511A devices, CODECs, and other devices in a system. This serial port is capable of operating at up to one-fourth the machine cycle rate (CLKOUT1).

The serial port uses two memory-mapped registers for data transfer: the data-transmit register (DXR) and the data-receive register (DRR). Both registers can be accessed in the same manner as any other memory location. The transmit and receive sections of the serial port each have associated clocks, frame-synchronization pulses, and serial-shift registers. Serial data can be transferred either in bytes or in 16-bit words. Serial port receive and transmit operations can generate their own maskable transmit and receive interrupts (XINT and RINT), allowing serial-port transfers to be managed through software. The ’C511A serial port is double-buffered and fully static.

The ’C511A serial port can be used to activate one of the various methods of multiprocessing on the device. This can be accomplished by programming one device to transmit while the others are in the receive mode, thereby broadcasting a serial transfer to multiple receiving devices simultaneously.

software wait-state generators

Software wait-state generation is incorporated in the ’C511A without any external hardware for interfacing with slower off-chip memory and I/O devices. The circuitry consists of 16 wait-state generating circuits and is user-programmable to operate with 0, 1, 2, 3, or 7 wait states. For off-chip memory accesses, these wait-state generators are mapped on 16K-word boundaries in program memory, data memory, and the I/O ports.

timer

The ’C511A features a 16-bit timing circuit with a 4-bit prescaler. This timer clocks between one-half and one thirty-second the machine rate of the device itself, depending on the programmable timer’s divide-down ratio. This timer can be stopped, restarted, reset, or disabled by specific status bits.

The timer can be used to generate CPU interrupts periodically. The timer is decremented by one at every CLKOUT1 cycle. A timer interrupt (TINT) and a pulse that is equal to the duration of a CLKOUT1 cycle on the external TOUT pin are generated each time the counter decrements to zero. The timer provides a convenient means of performing periodic I/O or other functions. When the timer is stopped, the internal clocks to the timer are shut off, allowing the device to run in a low-power mode of operation.

JTAG interface

The JTAG interface is for emulation purposes only. The JTAG scan logic is interfaced with other internal scanning logic circuitry, which has access to all of the on-chip resources. The ’C511A can perform on-board emulation by means of the JTAG serial scan pins and the emulation-dedicated pins.

multiprocessing

The flexibility of the ’C511A allows configurations to satisfy a wide range of system requirements. The device can be used in a variety of system configurations, including but not limited to the following:

- A stand-alone processor
- A multiprocessor with devices in parallel
- A slave/host multiprocessor with global-memory space
- A peripheral processor interfaced through processor-controlled signals to another device

For multiprocessing applications, the ’C5x can allocate global-memory space and communicate with that space by the $\overline{\text{BR}}$ and ready-control signals. Global memory is data memory that is shared by more than one device. Global-memory access must be arbitrated. The 8-bit memory-mapped global-memory-allocation register (GREG) specifies part of the ’C511A’s data memory as external global memory. The contents of the register determine the size of the global-memory space. If the current instruction addresses an operand within that space, $\overline{\text{BR}}$ is asserted to request control of the bus. The length of the memory cycle is controlled by the READY line.

The ’C511A supports direct memory access to its external program, data, and I/O spaces using the $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ signals. Another device can take complete control of the ’C511A’s external-memory interface by asserting $\overline{\text{HOLD}}$ low. This causes the ’C511A to place its address, data, and control lines in the high-impedance state, and to assert $\overline{\text{HOLDA}}$. When external memory is being accessed, program execution from on-chip memory can proceed concurrently when the device is in the hold mode.

Multiple ’C511As can be interconnnected through the serial port. This form of interconnection allows information to be transferred at high speed while using a minimum number of signal connections. A complete full-duplex serial-port interconnection between multiple processors can be accomplished with as few as four signal lines.

If more than two devices are being interconnected, one device may be programmed to transmit while the others are in the receive mode, thereby broadcasting a serial transfer to multiple receiving devices simultaneously.

instruction set

The ’C511A microprocessor implements a comprehensive instruction set that supports numeric-intensive signal-processing operations and general-purpose applications such as multiprocessing and high-speed control. Source code for the ’C1x and ’C2x DSPs is upward-compatible with the ’C5x generation devices like the ’C511A.

instruction set (continued)

For maximum throughput, the next instruction is prefetched while the current one is being executed. Because the same data lines are used to communicate to external data, program, or I/O space, the number of cycles an instruction requires to execute varies, depending on whether the next data-operand fetch is from internal or external memory. Highest throughput is achieved by maintaining data memory on chip and by using either internal or fast external program memory.

addressing modes

The ’C511A instruction set provides six basic memory-addressing modes: direct, indirect, immediate, register, memory-mapped, and circular addressing.

In direct addressing, the instruction word contains the lower seven bits of the data-memory address. This field is concatenated with the nine bits of the data-memory page pointer (DP) to form the 16-bit data-memory address. Therefore, in the direct-addressing mode, data memory is effectively paged with a total of 512 pages, each page containing 128 words.

Indirect addressing accesses data memory through the auxiliary registers. In this addressing mode, the address of the instruction operand is contained in the currently selected auxiliary register. Eight auxiliary registers (AR0–AR7) provide flexible and powerful indirect addressing. To select a specific auxiliary register, the auxiliary register pointer (ARP) is loaded with a value from 0 to 7 for AR0 through AR7, respectively.

There are seven types of indirect addressing: autoincrement or autodecrement, postindexing by either adding or subtracting the contents of AR0, single-indirect addressing with no increment or decrement, and bit-reversed addressing [used in Fast Fourier Transforms (FFTs)] with increment or decrement. All operations are performed on the current auxiliary register in the same cycle as the original instruction, following which the current auxiliary register and ARP can be modified.

In immediate addressing, the actual operand data is provided in a portion of the instruction word or words. There are two types of immediate addressing: short and long. In short-immediate addressing, the data is contained in the lower eight bits in a single-word instruction. In long-immediate addressing, the data is contained in the second word of a two-word instruction. The immediate-addressing mode is useful for data that does not need to be stored or used more than once during the course of program execution, such as initialization values, constants, and so on.

The register-addressing mode uses operands in CPU registers either explicitly, such as with a direct reference to a specific register, or implicitly with instructions that intrinsically reference certain registers. In either case, operand reference is simplified because 16-bit values can be used without specifying a full 16-bit operand address or immediate value.

Memory-mapped addressing provides easy access to memory-mapped registers located on page zero of data memory. The flexibility of memory-mapped addressing results because accesses using this addressing mode are made independent of actual DP value and without having to provide a complete address of the memory location being accessed. Commonly used on-board registers can be accessed with a simplified addressing scheme.

Circular addressing is the most sophisticated ’C511A addressing mode. This addressing mode allows specified buffers in memory to be accessed sequentially with a pointer that automatically wraps around to the beginning of the buffer when the last location is accessed. A total of two independent circular buffers can be allocated at any given time.

Five dedicated registers are allocated for circular addressing: a beginning-of-buffer and an end-of-buffer register for each of the two independent circular buffers and a control register. Additionally, one of the auxiliary registers is used as the pointer into the circular buffer. All registers used in circular addressing must be properly initialized prior to performing any circular-buffer access.

addressing modes (continued)

The circular-addressing mode allows circular buffers, which permit data structures used in finite impulse response (FIR) filters, convolution and correlation algorithms, and waveform generators. Having the capability to access circular buffers automatically with no overhead allows these types of data structures to be used most efficiently.

repeat feature

The repeat function can be used with instructions such as multiply/accumulates (MAC and MACD), block moves (BLDD and BLPD), I/O transfers (IN/OUT), and table read/writes (TBLR/TBLW). These instructions, although normally multicycle, are pipelined when the repeat feature is used, and they effectively become single-cycle instructions. For example, the table-read instruction may take three or more cycles to execute, but when the instruction is repeated, a table location can be read every cycle.

The repeat counter (RPTC) is a 16-bit register that, when loaded with a number N, causes the next single instruction to be executed N + 1 times. The RPTC register is loaded by either the RPT or the RPTZ instruction. This results in a maximum of 65536 executions of a given instruction. RPTC is cleared by reset. The RPTZ instruction clears both ACC and PREG before the next instruction starts repeating. Once a repeat instruction (RPT or RPTZ) is decoded, all interrupts, including $\overline{\text{NMI}}$ (except reset), are masked until the completion of the repeat loop. However, the device responds to the $\overline{\text{HOLD}}$ signal while executing an RPT/RPTZ loop.

The ’C5x also implements a block-repeat feature that provides zero-overhead looping for FOR and DO loops. The function is controlled by three registers (PASR, PAER, and BRCR) and the BRAF bit in the PMST register. The block-repeat counter register (BRCR) is loaded with a loop count of 0 to 65535. Then, execution of the RPTB (repeat block) instruction loads the block-repeat-address start register (PASR) with the address of the instruction following the RPTB instruction and loads the block-repeat-address end register (PAER) with its long-immediate operand. The long-immediate operand is the address of the instruction following the last instruction in the loop minus one. (The repeat block must contain at least three instruction words.) Execution of the RPTB instruction automatically sets active the BRAF bit. With each PC update, the PAER contents are compared to the PC. If they are equal, the BRCR contents are compared to zero. If the BRCR contents are greater than zero, BRCR is decremented and the PASR is loaded into the PC, thereby starting the loop over. If not, the BRAF bit is set low and the processor resumes execution past the end of the code’s loop.

The equivalent of a WHILE loop can be implemented by setting the BRAF bit to zero if the exit condition is met. The program then completes the current pass through the loop but does not go back to the top. To exit, the bit must be reset at least four instruction words before the end of the loop. It is possible to exit block-repeat loops and return to them without stopping and restarting the loop. Branches, calls, and interrupts do not necessarily affect the loop. When program control is returned to the loop, loop execution is resumed.

instruction set summary

This section summarizes the operational codes (opcodes) of the instruction set for the ’C5x DSPs. The instruction set is a superset of the ’C1x and ’C2x instruction sets. The instructions are arranged according to function and are alphabetized by mnemonic within each category. The symbols in Table 3 describe the opcode symbols used in Table 4. The Texas Instruments ’C5x assembler accepts ’C2x instructions as well as ’C5x instructions.

The number of words that an instruction occupies in program memory is specified in column 4 of Table 4. In these cases, different forms of the instruction occupy a different number of words. For example, the ADD instruction occupies one word when the operand is a short-immediate value or two words if the operand is a long-immediate value.

The number of cycles that an instruction requires to execute is listed in column 5 of Table 4. All instructions are assumed to be executed from internal program memory and internal data dual-access memory. The cycle timings are for single-instruction execution, not for repeat mode.

instruction set summary (continued)

A read or write access to any peripheral memory-mapped register in data memory locations 20h–4Fh adds one cycle to the cycle time shown because all peripherals perform these accesses over the internal peripheral bus.

Table 3. Opcode Symbols

SYMBOL	DESCRIPTION
A	Address
ACC	Accumulator
ACCB	Accumulator buffer
ACCU	Accumulator
ARX	Auxiliary register value (0–7)
BITX	4-bit field that specifies which bit to test for the BIT instruction
BMAR	Block-move address register
DBMR	Dynamic bit-manipulation register
I	Addressing-mode bit
II...II	Immediate operand value
INTM	Interrupt-mode flag bit
INTR#	Interrupt vector number
N	Field for the XC instruction, indicating the number of instructions (one or two) to execute conditionally
PREG	Product register
PROG	Program memory
RPTC	Repeat counter
SHF, SHFT	3/4 bit shift value
TC	Test-control bit
T P	Two bits used by the conditional execution instructions to represent the following conditions: T P Meaning 0 0 BIO low 0 1 TC = 1 1 0 TC = 0 1 1 None of the above conditions
TREGn	Temporary register n (n = 0, 1, or 2) TREG0: Temporary register for multiplication TREG1: Temporary register for dynamic shift count TREG2: Temporary register used as bit pointer in dynamic-bit test
Z L V C	4-bit field representing the following conditions: Z: ACC = 0 L: ACC < 0 V: Overflow C: Carry A conditional instruction contains two of these 4-bit fields. The 4-LSB field of the instruction is a 4-bit mask field. A 1 in the corresponding mask bit indicates that the condition is being tested. The second 4-bit field (bits 4–7) indicates the state of the conditions designated by the mask bits as being tested. For example, to test for ACC ≥ 0, the Z and L fields are set while the V and C fields are not set. The next 4-bit field contains the state of the conditions to be tested. The Z field is set to indicate testing the condition ACC = 0, and the L field is reset to indicate testing the condition ACC ≥ 0. The conditions possible with these 8 bits are shown in the BCND, CC, and XC instructions. To determine if the conditions are met, the 4-LSB bit mask is ANDed with the conditions. If any bits are set, the conditions are met.

instruction set summary (continued)

Table 4. TMS320C511A Instruction Set Opcodes

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Absolute value of ACC	ABS	1011 1110 0000 0000	1	1
Add ACCB to ACC with carry	ADCB	1011 1110 0001 0001	1	1
Add to ACC with shift	ADD	0010 SHFT IAAA AAAA	1	1
Add to ACCL short immediate	ADD	1011 1000 IIII IIII	1	1
Add to ACC long immediate with shift	ADD	1011 1111 1001 SHFT	2	2
Add to ACC with shift of 16	ADD	0110 0001 IAAA AAAA	1	1
Add ACCB to ACC	ADDB	1011 1110 0001 0000	1	1
Add to ACC with carry	ADDC	0110 0000 IAAA AAAA	1	1
Add to low ACC with sign extension suppressed	ADDS	0110 0010 IAAA AAAA	1	1
Add to ACC with shift specified by TREG1[3–0]	ADDT	0110 0011 IAAA AAAA	1	1
AND ACC with data value	AND	0110 1110 IAAA AAAA	1	1
AND with ACC long immediate with shift	AND	1011 1111 1011 SHFT	2	2
AND with ACC long immediate with shift of 16	AND	1011 1110 1000 0001	2	2
AND ACCB with ACC	ANDB	1011 1110 0001 0010	1	1
Barrel-shift ACC right	BSAR	1011 1111 1110 SHFT	1	1
Complement ACC	CMPL	1011 1110 0000 0001	1	1
Store ACC in ACCB if ACC > ACCB	CRGT	1011 1110 0001 1011	1	1
Store ACC in ACCB if ACC< ACCB	CRLT	1011 1110 0001 1100	1	1
Exchange ACCB with ACC	EXAR	1011 1110 0001 1101	1	1
Load ACC with ACCB	LACB	1011 1110 0001 1111	1	1
Load ACC with shift	LACC	0001 SHFT IAAA AAAA	1	1
Load ACC long immediate with shift	LACC	1011 1111 1000 SHFT	2	2
Load ACC with shift of 16	LACC	0110 1010 IAAA AAAA	1	1
Load ACCL with immediate	LACL	1011 1001 IIII IIII	1	1
Load ACCL	LACL	0110 1001 IAAA AAAA	1	1
Load ACC with shift specified by TREG1[3–0]	LACT	0110 1011 IAAA AAAA	1	1
Load ACCL with memory-mapped register	LAMM	0000 1000 IAAA AAAA	1	1 or 2
Negate ACC	NEG	1011 1110 0000 0010	1	1
Normalize ACC	NORM	1010 0000 IAAA AAAA	1	1
OR ACC with data value	OR	0110 1101 IAAA AAAA	1	1
OR with ACC long immediate with shift	OR	1011 1111 1100 SHFT	2	2
OR with ACC long immediate with shift of 16	OR	1011 1110 1000 0010	2	2
OR ACCB with ACC	ORB	1011 1110 0001 0011	1	1
Rotate ACC one bit left	ROL	1011 1110 0000 1100	1	1
Rotate ACCB and ACC left	ROLB	1011 1110 0001 0100	1	1
Rotate ACC one bit right	ROR	1011 1110 0000 1101	1	1
Rotate ACCB and ACC right	RORB	1011 1110 0001 0101	1	1
Store ACC in ACCB	SACB	1011 1110 0001 1110	1	1
Store high ACC with shift	SACH	1001 1SHF IAAA AAAA	1	1
Store ACCL with shift	SACL	1001 0SHF IAAA AAAA	1	1
Store ACCL to memory-mapped register	SAMM	1000 1000 IAAA AAAA	1	1 or 2
Shift ACC 16 bits right if TREG1[4] = 0	SATH	1011 1110 0101 1010	1	1
Shift ACC[15–0] right as specified by TREG1[3–0]	SATL	1011 1110 0101 1011	1	1
Subtract ACCB from ACC	SBB	1011 1110 0001 1000	1	1
Subtract ACCB from ACC with borrow	SBBB	1011 1110 0001 1001	1	1
Shift ACC one bit left	SFL	1011 1110 0000 1001	1	1
Shift ACCB and ACC left	SFLB	1011 1110 0001 0110	1	1
Shift ACC one bit right	SFR	1011 1110 0000 1010	1	1
Shift ACCB and ACC right	SFRB	1011 1110 0001 0111	1	1
Subtract from ACC with shift	SUB	0011 SHFT IAAA AAAA	1	1
Subtract from ACC with shift of 16	SUB	0110 0101 IAAA AAAA	1	1
Subtract from ACC short immediate	SUB	1011 1010 IIII IIII	1	1
Subtract from ACC long immediate with shift	SUB	1011 1111 1010 SHFT	2	2

instruction set summary (continued)

Table 4. TMS320C511A Instruction Set Opcodes (Continued)

ACCUMULATOR MEMORY REFERENCE INSTRUCTIONS (CONTINUED)				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Subtract from ACC with borrow	SUBB	0110 0100 IAAA AAAA	1	1
Conditional subtract	SUBC	0000 1010 IAAA AAAA	1	1
Subtract from ACC with sign extension suppressed	SUBS	0110 0110 IAAA AAAA	1	1
Subtract from ACC, shift specified by TREG1 [3–0]	SUBT	0110 0111 IAAA AAAA	1	1
XOR ACC with data value	XOR	0110 1100 IAAA AAAA	1	1
XOR with ACC long immediate with shift	XOR	1011 1111 1101 SHFT	2	2
XOR with ACC long immediate with shift of 16	XOR	1011 1110 1000 0011	2	2
XOR ACCB with ACC	XORB	1011 1110 0001 1010	1	1
Zero ACC, load high ACC with rounding	ZALR	0110 1000 IAAA AAAA	1	1
Zero ACC and product register	ZAP	1011 1110 0101 1001	1	1
AUXILIARY REGISTERS AND DATA PAGE POINTER INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Add to AR short immediate	ADRK	0111 1000 IIII IIII	1	1
Compare AR with CMPR	CMPR	1011 1111 0100 01CM	1	1
Load AR from addressed data	LAR	0000 0ARX IAAA AAAA	1	1
Load AR short immediate	LAR	1011 0ARX IIII IIII	1	1
Load AR long immediate	LAR	1011 1111 0000 1ARX	2	2
Load data page pointer with addressed data	LDP	0000 1101 IAAA AAAA	1	2
Load data page immediate	LDP	1011 110I IIII IIII	1	2
Modify auxiliary register	MAR	1000 1011 IAAA AAAA	1	1
Store AR to addressed data	SAR	1000 0ARX IAAA AAAA	1	1
Subtract from AR short immediate	SBRK	0111 1100 IIII IIII	1	1
BRANCH INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Branch unconditional with AR update	B	0111 1001 1AAA AAAA	2	4
Branch addressed by ACC	BACC	1011 1110 0010 0000	1	4
Branch addressed by ACC delayed	BACCD	1011 1110 0010 0001	1	2
Branch AR ≠ 0 with AR update	BANZ	0111 1011 1AAA AAAA	2	2 or 4
Branch AR ≠ 0 with AR update delayed	BANZD	0111 1111 1AAA AAAA	2	2
Branch conditional	BCND	1110 00TP ZLVC ZLVC	2	2 or 4
Branch conditional delayed	BCNDD	1111 00TP ZLVC ZLVC	2	2
Branch unconditional with AR update delayed	BD	0111 1101 1AAA AAAA	2	2
Call subroutine addressed by ACC	CALA	1011 1110 0011 0000	1	4
Call subroutine addressed by ACC delayed	CALAD	1011 1110 0011 1101	1	2
Call unconditional with AR update	CALL	0111 1010 1AAA AAAA	2	4
Call unconditional with AR update delayed	CALLD	0111 1110 1AAA AAAA	2	2
Call conditional	CC	1110 10TP ZLVC ZLVC	2	2 or 4
Call conditional delayed	CCD	1111 10TP ZLVC ZLVC	2	2
Software interrupt	INTR	1011 1110 011I NTR#	1	4
Nonmaskable interrupt	NMI	1011 1110 0101 0010	1	4
Return	RET	1110 1111 0000 0000	1	4
Return conditional	RETC	1110 11TP ZLVC ZLVC	1	2 or 4
Return conditionally, delayed	RETCD	1111 11TP ZLVC ZLVC	1	2
Return, delayed	RETD	1111 1111 0000 0000	1	2
Return from interrupt with enable	RETE	1011 1110 0011 1010	1	4
Return from interrupt	RETI	1011 1110 0011 1000	1	4
Trap	TRAP	1011 1110 0101 0001	1	4
Execute next one or two INST on condition	XC	111N 01TP ZLVC ZLVC	1	1

instruction set summary (continued)

Table 4. TMS320C511A Instruction Set Opcodes (Continued)

I/O AND DATA MEMORY OPERATIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Block-move from data to data memory	BLDD	1010 1000 IAAA AAAA	2	3
Block-move data to data DEST long immediate	BLDD	1010 1001 IAAA AAAA	2	3
Block-move data to data with source in BMAR	BLDD	1010 1100 IAAA AAAA	1	2
Block-move data to data with DEST in BMAR	BLDD	1010 1101 IAAA AAAA	1	2
Block-move data to PROG with DEST in BMAR	BLDP	0101 0111 IAAA AAAA	1	2
Block-move from program to data memory	BLPD	1010 0101 IAAA AAAA	2	3
Block-move PROG to data with source in BMAR	BLPD	1010 0100 IAAA AAAA	1	2
Data-move in data memory	DMOV	0111 0111 IAAA AAAA	1	1
Input external access	IN	1010 1111 IAAA AAAA	2	2
Load memory-mapped register	LMMR	1000 1001 IAAA AAAA	2	2 or 3
Out external access	OUT	0000 1100 IAAA AAAA	2	3
Store memory-mapped register	SMMR	0000 1001 IAAA AAAA	2	2 or 3
Table read	TBLR	1010 0110 IAAA AAAA	1	3
Table write	TBLW	1010 0111 IAAA AAAA	1	3
PARALLEL LOGIC UNIT INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
AND DBMR with data value	APL	0101 1010 IAAA AAAA	1	1
AND long immediate with data value	APL	0101 1110 IAAA AAAA	2	2
Compare DBMR to data value	CPL	0101 1011 IAAA AAAA	1	1
Compare data with long immediate	CPL	0101 1111 IAAA AAAA	2	2
OR DBMR to data value	OPL	0101 1001 IAAA AAAA	1	1
OR long immediate with data value	OPL	0101 1101 IAAA AAAA	2	2
Store long immediate to data	SPLK	1010 1110 IAAA AAAA	2	2
XOR DBMR to data value	XPL	0101 1000 IAAA AAAA	1	1
XOR long immediate with data value	XPL	0101 1100 IAAA AAAA	2	2
T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Add PREG to ACC	APAC	1011 1110 0000 0100	1	1
Load high PREG	LPH	0111 0101 IAAA AAAA	1	1
Load TREG0	LT	0111 0011 IAAA AAAA	1	1
Load TREG0 and accumulate previous product	LTA	0111 0000 IAAA AAAA	1	1
Load TREG0, accumulate previous product, and move data	LTD	0111 0010 IAAA AAAA	1 1	1 1
Load TREG0 and load ACC with PREG	LTP	0111 0001 IAAA AAAA	1	1
Load TREG0 and subtract previous product	LTS	0111 0100 IAAA AAAA	1	1
Multiply/accumulate	MAC	1010 0010 IAAA AAAA	2	3
Multiply/accumulate with data shift	MACD	1010 0011 IAAA AAAA	2	3
Mult/ACC w/source ADRS in BMAR and DMOV	MADD	1010 1011 IAAA AAAA	1	3
Mult/ACC with source address in BMAR	MADS	1010 1010 IAAA AAAA	1	3
Multiply data value times TREG0	MPY	0101 0100 IAAA AAAA	1	1
Multiply TREG0 by 13-bit immediate	MPY	110I IIII IIII IIII	1	1
Multiply TREG0 by long immediate	MPY	1011 1110 1000 0000	2	2
Multiply TREG0 by data, add previous product	MPYA	0101 0000 IAAA AAAA	1	1
Multiply TREG0 by data, subtract PREG from ACC	MPYS	0101 0001 IAAA AAAA	1	1
Multiply unsigned data value times TREG0	MPYU	0101 0101 IAAA AAAA	1	1
Load ACC with product register	PAC	1011 1110 0000 0011	1	1
Subtract product from ACC	SPAC	1011 1110 0000 0101	1	1
Store high product register	SPH	1000 1101 IAAA AAAA	1	1
Store low product register	SPL	1000 1100 IAAA AAAA	1	1
Set PREG shift count	SPM	1011 1111 0000 00PM	1	1
Data to TREG0, square it, add PREG to ACC	SQRA	0101 0010 IAAA AAAA	1	1
Data to TREG0, square it, subtract PREG from ACC	SQRS	0101 0011 IAAA AAAA	1	1
Zero product register	ZPR	1011 1110 0101 1000	1	1

instruction set summary (continued)

Table 4. TMS320C511A Instruction Set Opcodes (Continued)

CONTROL INSTRUCTIONS				
INSTRUCTION	**MNEMONIC**	**OPCODE**	**WORDS**	**CYCLES**
Test bit specified immediate	BIT	0100 BITX IAAA AAAA	1	1
Test bit in data value as specified by TREG2[3–0]	BITT	0110 1111 IAAA AAAA	1	1
Reset overflow mode	CLRC	1011 1110 0100 0010	1	1
Reset sign-extension mode	CLRC	1011 1110 0100 0110	1	1
Reset hold mode	CLRC	1011 1110 0100 1000	1	1
Reset TC bit	CLRC	1011 1110 0100 1010	1	1
Reset carry	CLRC	1011 1110 0100 1110	1	1
Reset CNF bit	CLRC	1011 1110 0100 0100	1	1
Reset INTM bit	CLRC	1011 1110 0100 0000	1	1
Reset XF pin	CLRC	1011 1110 0100 1100	1	1
Idle	IDLE	1011 1110 0010 0010	1	1
Idle until interrupt — low-power mode	IDLE2	1011 1110 0010 0011	1	1
Load status register 0	LST	0000 1110 IAAA AAAA	1	2
Load status register 1	LST	0000 1111 IAAA AAAA	1	2
No operation	NOP	1000 1011 0000 0000	1	1
Pop PC stack to ACCL	POP	1011 1110 0011 0010	1	1
Pop stack to data memory	POPD	1000 1010 IAAA AAAA	1	1
Push data memory value onto PC stack	PSHD	0111 0110 IAAA AAAA	1	1
Push low ACC to PC stack	PUSH	1011 1110 0011 1100	1	1
Repeat instruction as specified by data	RPT	0000 1011 IAAA AAAA	1	2
Repeat next INST specified by long immediate	RPT	1011 1110 1100 0100	2	2
Repeat INST specified by short immediate	RPT	1011 1011 IIII IIII	1	2
Block repeat	RPTB	1011 1110 1100 0110	2	2
Clear ACC/PREG and repeat next INST long immediate	RPTZ	1011 1110 1100 0101	2	2
Set overflow mode	SETC	1011 1110 0100 0011	1	1
Set sign-extension mode	SETC	1011 1110 0100 0111	1	1
Set hold mode	SETC	1011 1110 0100 1001	1	1
Set TC bit	SETC	1011 1110 0100 1011	1	1
Set carry	SETC	1011 1110 0100 1111	1	1
Set XF pin high	SETC	1011 1110 0100 1101	1	1
Set CNF bit	SETC	1011 1110 0100 0101	1	1
Set INTM bit	SETC	1011 1110 0100 0001	1	1
Store status register 0	SST	1000 1110 IAAA AAAA	1	1
Store status register 1	SST	1000 1111 IAAA AAAA	1	1

development support

Texas Instruments offers an extensive line of development tools for the ’C511A, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and to integrate fully and debug software and hardware modules.

The following products support development of ’C5x-based applications:

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

development support (continued)

Hardware Development Tools:
Extended development system (XDS™) emulator (supports ’C5x multiprocessor system debug)
’C5x EVM (Evaluation Module)
’C5x DSK (DSP Starter Kit)

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

See Table 5 for a complete listing of development support tools for the ’C511A. For information on pricing and availability, contact the nearest TI field sales office or authorized distributor.

Table 5. TMS320C511A Development-Support Tools

DEVELOPMENT TOOL	PLATFORM	PART NUMBER
Software		
Compiler/Assembler/Linker	PC-DOS™, OS/2™	TMDS3242855-02
Compiler/Assembler/Linker	SPARC™, HP™	TMDS3242555-08
Assembler/Linker	PC-DOS, OS/2	TMDS3242850-02
Simulator	PC-DOS, WIN™	TMDS3245851-02
Simulator	SPARC	TMDS3245551-09
Digital Filter Design Package	PC-DOS	DFDP
Debugger/Emulation Software	PC-DOS, OS/2, WIN	TMDS3240150
Debugger/Emulation Software	SPARC	TMDS3240650
Hardware		
XDS-510 XL Emulator	PC-DOS, OS/2	TMD000510
XDS-510 WS Emulator	SPARC	TMDS000510WS
Evaluation Module (EVM)	PC-DOS, WIN	TMDS3260050
DSP Starter Kit (DSK)	PC-DOS	TMDS3200051

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 its support tools: TMDX and TMDS. These prefixes represent evolutionary stages of product development from engineering prototypes (TMX/TMDX) through fully qualified production devices/tools (TMS/TMDS). This development flow is defined below.

Device development evolutionary flow:

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

PC-DOS and OS/2 are trademarks of International Business Machines Corp.
SPARC is a trademark of SPARC International, Inc.
WIN is a trademark of Microsoft Corp.
HP is a trademark of Hewlett-Packard Company.
XDS is a trademark of Texas Instruments Incorporated.

device and development support tool nomenclature (continued)

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, N, FN, or GB) and temperature range (for example, L). Figure 2 provides a legend for reading the complete device name for any TMS320 or TMX320 family member.

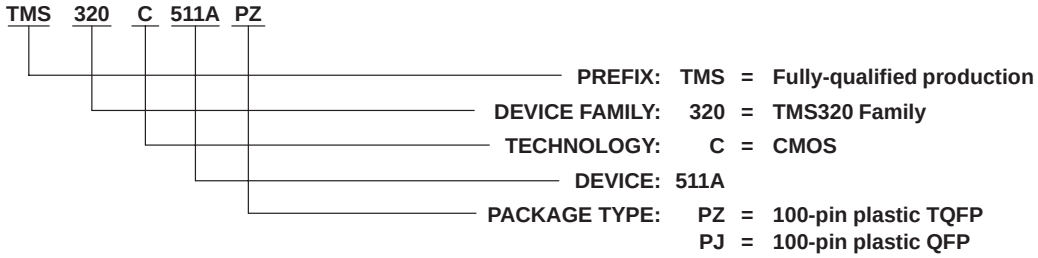


Figure 2. Device Nomenclature

documentation support

Extensive documentation supports all of the TMS320 family devices from product announcement through applications development. The types of documentation available include data sheets with design specifications, user's guides for all devices and development support tools, and three volumes of *Digital Signal Processing Applications with the TMS320 Family* (literature numbers SPRA012, SPRA016, and SPRA017).

The application book series describes hardware and software applications, including algorithms, for fixed and floating-point TMS320 family devices. The *TMS320C5x User's Guide* (literature number SPRU056), which describes in detail the fifth-generation TMS320 products, is also currently available.

A series of DSP textbooks is published by Prentice-Hall and John Wiley & Sons to support digital signal processing 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 713/274-2323.

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

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

Supply voltage range, V_{DD} (see Note 1) . . . . . – 0.3 V to 7 V
Input voltage range, V_I . . . . . – 0.3 V to 7 V
Output voltage range, V_O . . . . . – 0.3 V to 7 V
Operating case temperature range, T_C . . . . . 0°C to 85°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
V_{DD}	Supply voltage		4.75	5	5.25	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	X2/CLKIN	3		V_{DD}+0.3	V
		CLKX, CLKR	2.5		V_{DD}+0.3	
		All other inputs	2		V_{DD}+0.3	
V_{IL}	Low-level input voltage	X2/CLKIN, CLKX, CLKR	– 0.3		0.7	V
		All other inputs	– 0.3		0.8	
I_{OH}	High-level output current (see Note 2)				– 300	µA
I_{OL}	Low-level output current				2	mA
T_C	Operating case temperature		0		85	°C

NOTE 2: Figure 3 shows the test load circuit: Figure 4 and Figure 5 show the voltage reference levels.

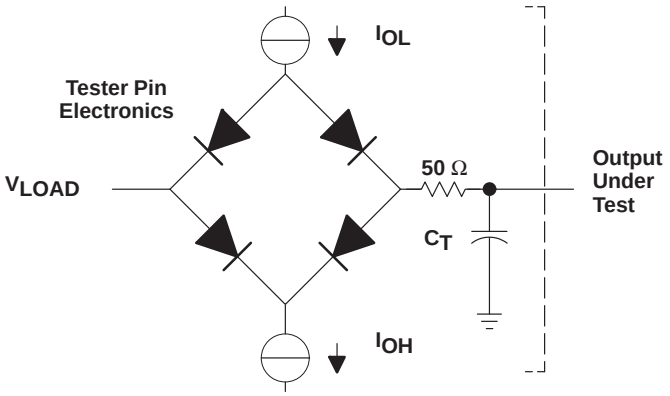
electrical characteristics over recommended ranges of supply voltage and operating case temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage (see Note 2)	I_{OH} = – 300 µA	2.4	3		V
V_{OL}	Low-level output voltage (see Note 2)	I_{OL} = 2 mA		0.3	0.6	V
I_{OZ}	High-impedance output current (V_{DD} = 5.25 V)	$\overline{BR}$ (with internal pullup)	–500		20	µA
		All other 3-state outputs	– 20		20	
I_I	Input current (V_I = V_{SS} to V_{DD})	$\overline{TRST}$ (with internal pulldown)	– 10		800	µA
		TMS, TCK, TDI (with internal pullups)	–500		10	
		X2/CLKIN	– 50		50	
		All other inputs	– 10		10	
$I_{DD(core)}$	Supply current, core CPU	V_{DD} = 5.25 V, f_x = 92 MHz		102		mA
		V_{DD} = 5.25 V, f_x = 96 MHz		106		
		V_{DD} = 5.25 V, f_x = 100 MHz		110		
$I_{DD(pins)}$	Supply current, pins	V_{DD} = 5.25 V, f_x = 92 MHz		69		mA
		V_{DD} = 5.25 V, f_x = 96 MHz		72		
		V_{DD} = 5.25 V, f_x = 100 MHz		75		
$I_{DD(standby)}$	Supply current, standby	IDLE2, divide-by-two clock mode, clocks shut off		5		µA
C_i	Input capacitance			15		pF
C_o	Output capacitance			15		pF

† Typical values are at V_{DD} = 5 V, ambient-air temperature = 25°C, unless otherwise specified.

NOTE 2: Figure 3 shows the test load circuit; Figure 4 and Figure 5 show the voltage reference levels.

PARAMETER MEASUREMENT INFORMATION



Where: I_{OL} = 2 mA (all outputs)
I_{OH} = 300 μA (all outputs)
V_{LOAD} = 1.5 V
C_T = 80-pF typical load-circuit capacitance

Figure 3. Test Load Circuit

signal transition levels

TTL-output levels are driven to a minimum logic-high level of 2.4 V and to a maximum logic-low level of 0.6 V. Figure 4 shows the TTL-level outputs.

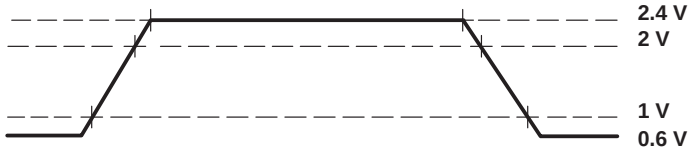


Figure 4. TTL-Level Outputs

TTL-output transition times are specified as follows:

- For a *high-to-low transition*, the level at which the output is said to be no longer high is 2 V, and the level at which the output is said to be low is 1 V.
- For a *low-to-high transition*, the level at which the output is said to be no longer low is 1 V, and the level at which the output is said to be high is 2 V.

Figure 5 shows the TTL-level inputs.

Figure 5. TTL-Level Inputs

TTL-compatible input transition times are specified as follows:

- For a *high-to-low transition* on an input signal, the level at which the input is said to be no longer high is 2 V, and the level at which the input is said to be low is 0.8 V.
- For a *low-to-high transition* on an input signal, the level at which the input is said to be no longer low is 0.8 V, and the level at which the input is said to be high is 2 V.

PARAMETER MEASUREMENT INFORMATION

timing parameter symbology

Timing parameter symbols used are created in accordance with JEDEC Standard 100-A. To shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows:

Lowercase subscripts and their meanings:	
a	access time
c	cycle time (period)
d	delay time
dis	disable time
en	enable time
f	fall time
h	hold time
r	rise time
su	setup time
t	transition time
v	valid time
w	pulse duration (width)
X	Unknown, changing, or don't care level

Letters and symbols and their meanings:	
H	High
L	Low
V	Valid
Z	High impedance

CLOCK CHARACTERISTICS AND TIMING

The ’C511A can use either its internal oscillator or an external frequency source for a clock. The clock mode is determined by the clock mode pins (CLKMD1, CLKMD2, and CLKMD3). Table 6 shows the clock options available on the ’C511A device. Also for the ’C511A device, X2/CLKIN functions as the external frequency input when using the PLL options.

Table 6. PLL Clock Option for TMS320C511A

CLKMD1	CLKMD2	CLKMD3	CLOCK SOURCE
0	0	1	External divide-by-two option with oscillator disabled
0	1	1	PLL multiply-by-two
1	1	1	External/Internal divide-by-two with oscillator enabled

internal divide-by-two clock option with external crystal

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN. The frequency of CLKOUT1 is one-half the crystal’s oscillating frequency. The crystal should be in either fundamental or overtone operation and parallel resonant, with an effective series resistance of 30 Ω and a power dissipation of 1 mW; it should be specified at a load capacitance of 20 pF. Overtone crystals require an additional tuned LC circuit. Figure 6 shows an external crystal (fundamental frequency) connected to the on-chip oscillator.

recommended operating conditions for internal divide-by-two clock option

		’320C511A-92			’320C511A-96			’320C511A-100			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	MIN	NOM	MAX	
f_X	Input clock frequency	0†		92	0†		96	0†		100	MHz
C1, C2	Load capacitance		10			10			10		pF

† This device uses a fully static design and therefore can operate with $t_{C(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz but is tested at f_X = 6.7 MHz to meet device test time requirements.

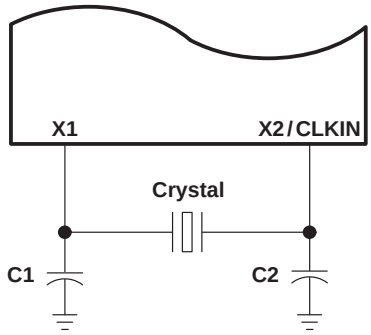


Figure 6. Internal Clock Option

CLOCK CHARACTERISTICS AND TIMING (CONTINUED)

external divide-by-two clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected, and CLKMD1, CLKMD2 and CLKMD3 pins configured appropriately. Refer to Table 6 for the appropriate pin configuration to generate the external divide-by-two clock option. This external frequency is divided by two to generate the internal machine cycle. The external frequency injected must conform to the specifications listed in the timing requirements table.

switching characteristics over recommended operating conditions [H = 0.5 $t_{c(CO)}$] (see Figure 7)

PARAMETER		'320C511A-92			'320C511A-96			'320C511A-100			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT1	21.7	$2t_{c(CI)}$	†	20.8	$2t_{c(CI)}$	†	20	$2t_{c(CI)}$	†	ns
$t_{d(CIH-COH/L)}$	Delay time, X2/CLKIN high to CLKOUT1 high/low	1	9	18	1	9	18	1	9	18	ns
$t_{f(CO)}$	Fall time, CLKOUT1		4			4			4		ns
$t_{r(CO)}$	Rise time, CLKOUT1		4			4			4		ns
$t_{w(COL)}$	Pulse duration, CLKOUT1 low	H – 3	H	H + 2	H – 3	H	H + 2	H – 3	H	H + 2	ns
$t_{w(COH)}$	Pulse duration, CLKOUT1 high	H – 3	H	H + 2	H – 3	H	H + 2	H – 3	H	H + 2	ns

† This device utilizes a fully static design and, therefore, can operate with $t_{c(CI)}$ approaching infinity. The device is characterized at frequencies approaching 0 Hz but is tested at $t_{c(CO)}$ = 300 ns to meet device test time requirements.

timing requirements over recommended ranges of supply voltage and operating ambient-air temperature (see Figure 7)

		'320C511A-92		'320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	10.85	‡	10.4	‡	10	‡	ns
$t_{f(CI)}$	Fall time, X2/CLKIN§		4		4		4	ns
$t_{r(CI)}$	Rise time, X2/CLKIN§		4		4		4	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low	5	‡	5	‡	5	‡	ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high	5	‡	5	‡	5	‡	ns

‡ This device utilizes a fully static design and, therefore, can operate with $t_{c(CI)}$ approaching ∞. The device is characterized at frequencies approaching 0 Hz, but is tested at a minimum of $t_{c(CI)}$ = 150 ns to meet device test time requirements.

§ Values are derived from characterization data and are not tested.

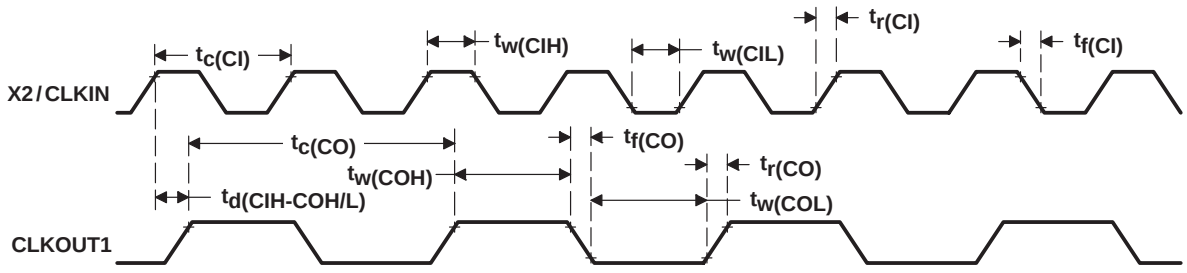


Figure 7. External Divide-by-Two Clock Timing

CLOCK CHARACTERISTICS AND TIMING (CONTINUED)

PLL clock generator option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. This external frequency is multiplied by the factors shown in Table 6 to generate the internal machine cycle. A multiplication factor of 2 is available on the ’C511A device. Refer to Table 6 for the appropriate configuration of the CLKMD1, CLKMD2, and CLKMD3 pins to generate this desired PLL multiplication factor. The external frequency injected must conform to the specifications listed in the timing requirements table.

switching characteristics over recommended operating conditions [H = 0.5 $t_{c(CO)}$] (see Figure 8)

PARAMETER		’320C511A-92			’320C511A-96			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT1	21.7		55	20.8		50	ns
$t_{f(CO)}$	Fall time, CLKOUT1		4			4		ns
$t_{r(CO)}$	Rise time, CLKOUT1		4			4		ns
$t_{w(COL)}$	Pulse duration, CLKOUT1 low	H – 3†	H	H + 2†	H – 3†	H	H + 2†	ns
$t_{w(COH)}$	Pulse duration, CLKOUT1 high	H – 3†	H	H + 2†	H – 3†	H	H + 2†	ns
$t_{d(CIH-COH)}$	Delay time, X2/CLKIN high to CLKOUT1 high	1	8	15	1	8	15	ns
$t_{d(TP)}$	Delay time, transitory phase—PLL synchronized after X2/CLKIN supplied			$1500t_{c(CI)}$†			$1500t_{c(CI)}$†	ns

PARAMETER		’320C511A-100			UNIT
		MIN	TYP	MAX	
$t_{c(CO)}$	Cycle time, CLKOUT1	20		45	ns
$t_{f(CO)}$	Fall time, CLKOUT1		4		ns
$t_{r(CO)}$	Rise time, CLKOUT1		4		ns
$t_{w(COL)}$	Pulse duration, CLKOUT1 low	H – 3†	H	H + 2†	ns
$t_{w(COH)}$	Pulse duration, CLKOUT1 high	H – 3†	H	H + 2†	ns
$t_{d(CIH-COH)}$	Delay time, X2/CLKIN high to CLKOUT1 high	1	8	15	ns
$t_{d(TP)}$	Delay time, transitory phase—PLL synchronized after X2/CLKIN supplied			$1500t_{c(CI)}$†	ns

† Values are assured by design and are not tested.

CLOCK CHARACTERISTICS AND TIMING (CONTINUED)

timing requirements over recommended ranges of supply voltage and operating case temperature (see Figure 8)

			'320C511A-92		'320C511A-96		'320C511A-100		UNIT
			MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(CI)}$	Cycle time, X2/CLKIN	Multiply-by-two	43.4	110†	41.6	100†	40	90†	ns
$t_{f(CI)}$	Fall time, X2/CLKIN‡			4		4		4	ns
$t_{r(CI)}$	Rise time, X2/CLKIN‡			4		4		4	ns
$t_{w(CIL)}$	Pulse duration, X2/CLKIN low		8	$t_{c(CI)}-8$	8	$t_{c(CI)}-8$	7	$t_{c(CI)}-7$	ns
$t_{w(CIH)}$	Pulse duration, X2/CLKIN high		8	$t_{c(CI)}-8$	8	$t_{c(CI)}-8$	7	$t_{c(CI)}-7$	ns

† Clocks can be stopped only while executing IDLE2 when using the PLL clock generator option. The $t_{d(TP)}$ (the transitory phase) occurs when restarting clock from IDLE2 in this mode.
‡ Values are derived from characterization data and are not tested.

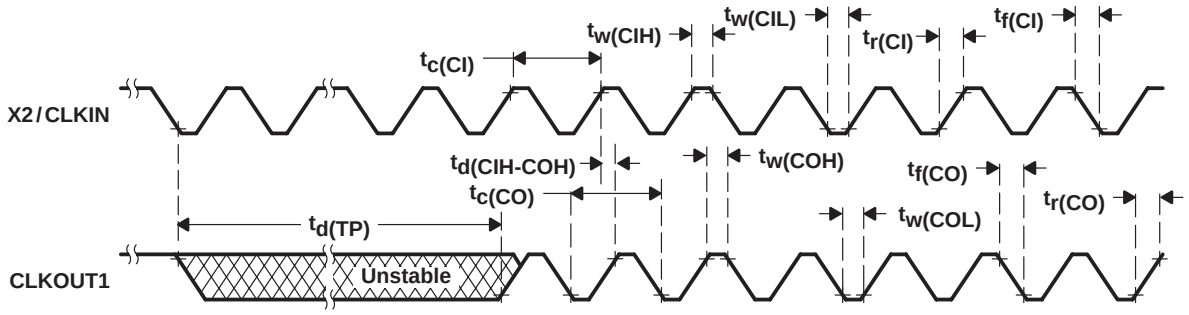


Figure 8. PLL Clock Generator Timing

MEMORY AND PARALLEL I/O INTERFACE READ

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 9)

PARAMETER		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(AV-RDL)}$	Setup time, address valid before $\overline{RD}$ low†	H – 7‡		H – 6‡		ns
$t_{h(RDH-AV)}$	Hold time, address valid after $\overline{RD}$ high†	0‡		0‡		ns
$t_{w(RDL)}$	Pulse duration, $\overline{RD}$ low§¶#	H – 2	H + 2	H – 2	H + 2	ns
$t_{w(RDH)}$	Pulse duration, $\overline{RD}$ high§¶#	H – 2		H – 2		ns
$t_{d(CO-ST)}$	Delay time, CLKOUT1 to $\overline{STRB}$ rising or falling edge§¶	– 2	2	– 2	2	ns
$t_{d(CO-RD)}$	Delay time, CLKOUT1 to $\overline{RD}$ rising or falling edge§¶	– 3	1	– 3	1	ns
$t_{d(RDH-WEL)}$	Delay time, $\overline{RD}$ high to $\overline{WE}$ low	2H – 4		2H – 4		ns

timing requirements over recommended ranges of supply voltage and operating case temperature [H = 0.5$t_{c(CO)}$] (see Figure 9)

		'320C511A-92		'320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{a(RD-AV)}$	Access time, read data from address valid		2H – 6.7†		2H – 8.8†		2H – 8†	ns
$t_{a(RDL-RD)}$	Access time, read data after $\overline{RD}$ low		H – 3		H – 3		H – 3	ns
$t_{su(RD-RDH)}$	Setup time, read data before $\overline{RD}$ high	3		3		3		ns
$t_{h(RDH-RD)}$	Hold time, read data after $\overline{RD}$ high	2		2		2		ns

† A0–A15, $\overline{PS}$, $\overline{DS}$, $\overline{IS}$, R/$\overline{W}$, and $\overline{BR}$ timings are all included in timings referenced as address.
‡ See Figure 10 for address-bus timing variation with load capacitance.
§ These timings are for the cycles following the first cycle after reset, which is always seven wait states.
¶ Values are derived from characterization data and are not tested.
# Timings are valid for zero wait-state cycles only.

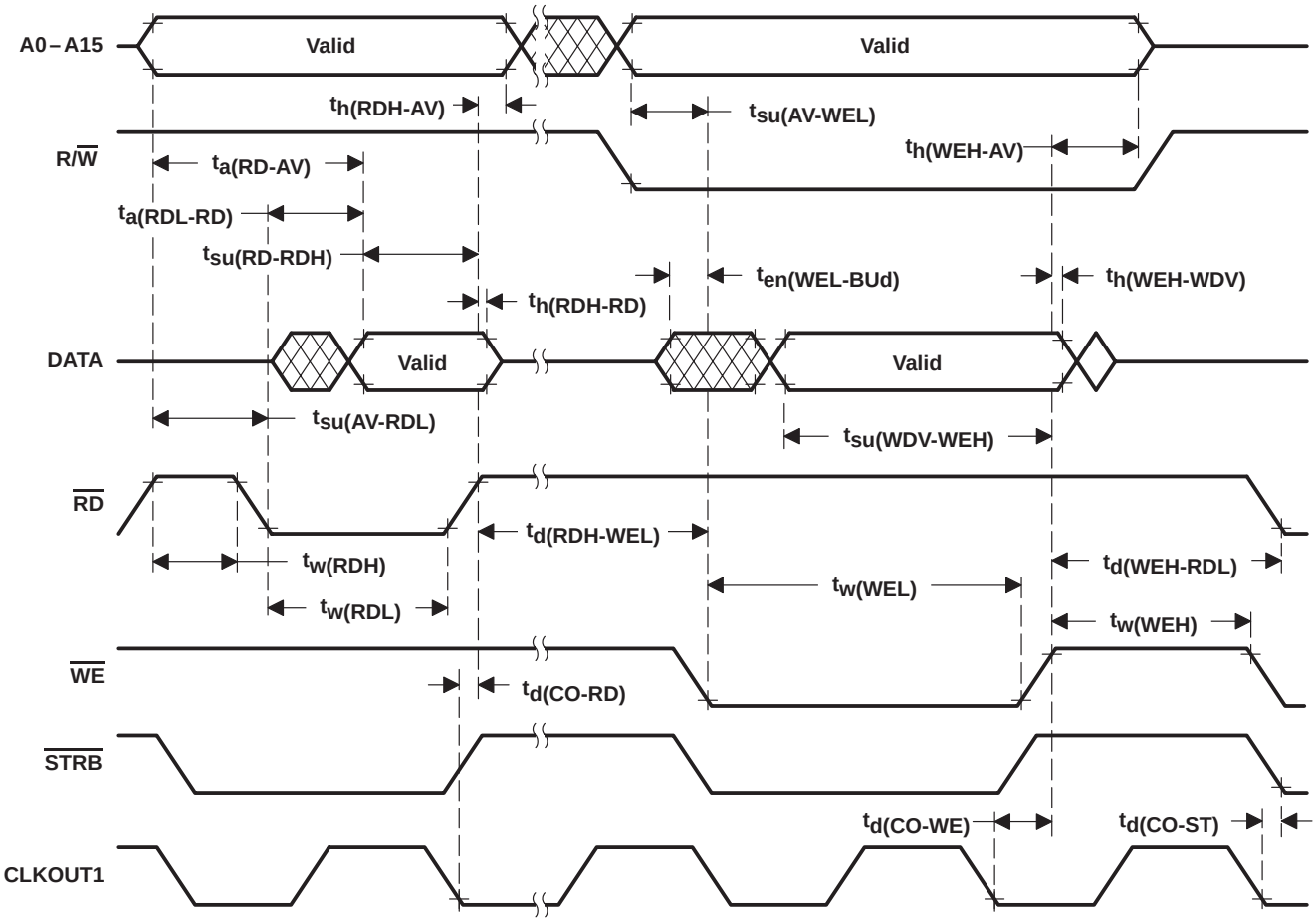
MEMORY AND PARALLEL I/O INTERFACE WRITE

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 9)

PARAMETER		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(AV-WEL)}$	Setup time, address valid before $\overline{WE}$ low†	H – 4‡		H – 3‡		ns
$t_{su(WDV-WEH)}$	Setup time, write data valid before $\overline{WE}$ high	2H – 14	2H§¶	2H – 14	2H§¶	ns
$t_{h(WEH-AV)}$	Hold time, address valid after $\overline{WE}$ high†	H – 7‡		H – 7‡		ns
$t_{h(WEH-WDV)}$	Hold time, write data valid after $\overline{WE}$ high	H – 4	H + 7§	H – 4	H + 7§	ns
$t_{w(WEL)}$	Pulse duration, $\overline{WE}$ low§¶	2H – 2	2H + 2	2H – 2	2H + 2	ns
$t_{w(WEH)}$	Pulse duration, $\overline{WE}$ high§	2H – 2		2H – 2		ns
$t_{d(CO-ST)}$	Delay time, CLKOUT1 to $\overline{STRB}$ rising or falling edge§	– 2	2	– 2	2	ns
$t_{d(CO-WE)}$	Delay time, CLKOUT1 to $\overline{WE}$ rising or falling edge§	– 1	3	– 1	3	ns
$t_{d(WEH-RDL)}$	Delay time, $\overline{WE}$ high to $\overline{RD}$ low	3H – 7		3H – 7		ns
$t_{en(WEL-BUd)}$	Enable time, $\overline{WE}$ low to data bus driven	– 4§		– 4§		ns

† A0–A15, $\overline{PS}$, $\overline{DS}$, $\overline{IS}$, R/$\overline{W}$, and $\overline{BR}$ timings are all included in timings referenced as address.
‡ See Figure 10 for address-bus timing variation with load capacitance.
§ Values are derived from characterization data and are not tested.
¶ This value holds true for zero wait states or one software wait state only.

MEMORY AND PARALLEL I/O INTERFACE READ AND WRITE



NOTES: A. All timings are for 0 wait states. However, external writes always require two cycles to prevent external bus conflicts. The diagram illustrates a one-cycle read and a two-cycle write and is not drawn to scale. All external writes immediately preceded by an external read or immediately followed by an external read require three machine cycles.

B. Refer to Appendix B of *TMS320C5x User's Guide* (literature number SPRU056) for logical timings of external interface.

Figure 9. Memory and Parallel I/O Interface Read and Write Timing

MEMORY AND PARALLEL I/O INTERFACE READ AND WRITE (CONTINUED)

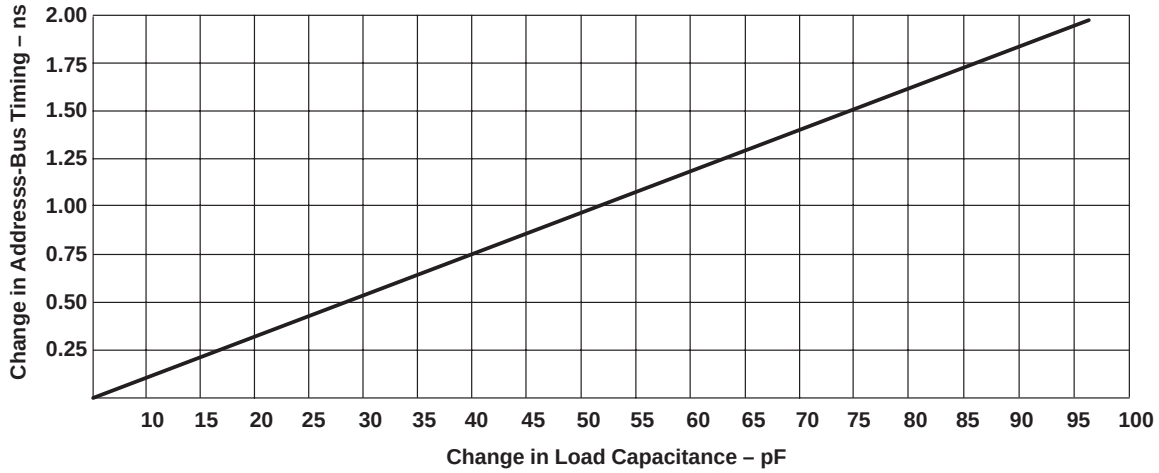


Figure 10. Address-Bus Timing Variation With Load Capacitance

READY TIMING FOR EXTERNALLY GENERATED WAIT STATES

timing requirements over recommended ranges of supply voltage and operating case temperature (see Note 3) (see Figure 11 and Figure 12)

		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(RY-COH)}$	Setup time, READY before CLKOUT1 rising edge	7		6		ns
$t_{su(RY-RDL)}$	Setup time, READY before $\overline{RD}$ falling edge	7		6		ns
$t_{h(COH-RYH)}$	Hold time, READY after CLKOUT1 rising edge	0		0		ns
$t_{h(RDL-RY)}$	Hold time, READY after $\overline{RD}$ falling edge	0		0		ns
$t_{h(WEL-RY)}$	Hold time, READY after $\overline{WE}$ falling edge	H + 4		H + 3		ns
$t_{v(WEL-RY)}$	Valid time, READY after $\overline{WE}$ falling edge		H – 10		H – 8	ns

NOTE 3: The external READY input is sampled only after the internal software wait states are completed.

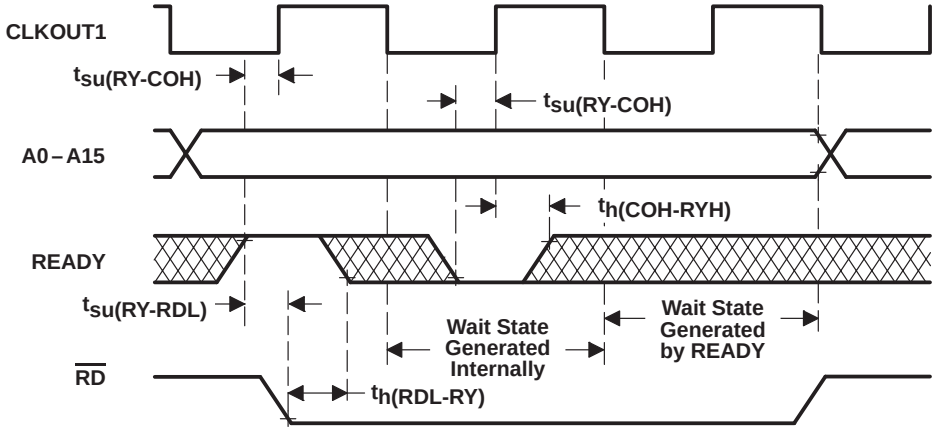


Figure 11. Ready Timing for Externally Generated Wait States During an External Read Cycle

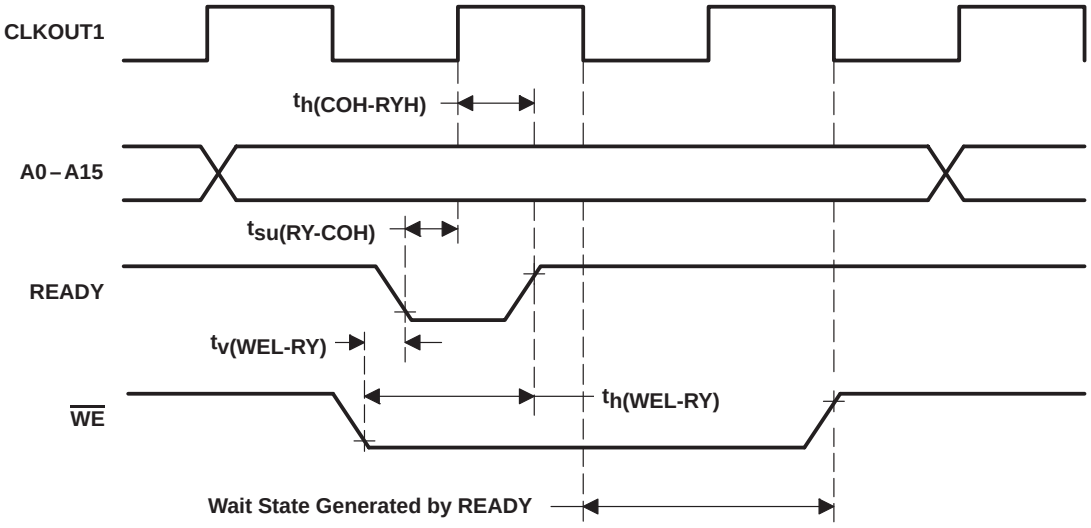


Figure 12. Ready Timing for Externally Generated Wait States During an External Write Cycle

RESET, INTERRUPT, AND $\overline{\text{BIO}}$

timing requirements over recommended ranges of supply voltage and operating case temperature [H = 0.5$t_{c(CO)}$] (see Figure 13)

		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{su(IN-COL)}$	Setup time, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ before CLKOUT1 low †	10		10		ns
$t_{su(RS-COL)}$	Setup time, $\overline{\text{RS}}$ before CLKOUT1 low	10	2H – 5‡	10	2H – 5‡	ns
$t_{su(RS-CIL)}$	Setup time, $\overline{\text{RS}}$ before X2/CLKIN low	7		7		ns
$t_{su(BI-COL)}$	Setup time, $\overline{\text{BIO}}$ before CLKOUT1 low	10		10		ns
$t_{h(COL-IN)}$	Hold time, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ after CLKOUT1 low †	0		0		ns
$t_{h(COL-BI)}$	Hold time, $\overline{\text{BIO}}$ after CLKOUT1 low	0		0		ns
$t_{w(INL)SYN}$	Pulse duration, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ low, synchronous	4H + 10§		4H + 10§		ns
$t_{w(INH)SYN}$	Pulse duration, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ high, synchronous	2H + 10§		2H + 10§		ns
$t_{w(INL)ASY}$	Pulse duration, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ low, asynchronous ‡	6H + 10§		6H + 10§		ns
$t_{w(INH)ASY}$	Pulse duration, $\overline{\text{INT1}}$–$\overline{\text{INT4}}$, $\overline{\text{NMI}}$ high, asynchronous ‡	4H + 10§		4H + 10§		ns
$t_{w(RSL)}$	Pulse duration, $\overline{\text{RS}}$ low	12H		12H		ns
$t_{w(BIL)SYN}$	Pulse duration, $\overline{\text{BIO}}$ low, synchronous	10		10		ns
$t_{w(BIL)ASY}$	Pulse duration, $\overline{\text{BIO}}$ low, asynchronous ‡	H + 10		H + 10		ns
$t_{d(RSH)}$	Delay time, $\overline{\text{RS}}$ high to reset vector fetch	34H		34H		ns

† These parameters must be met to use the synchronous timings. Both reset and the interrupts can operate asynchronously. The pulse durations require an extra half-cycle to ensure internal synchronization.
‡ These values are derived from characterization data and are not tested.
§ If in IDLE2, add 4H to these timings.

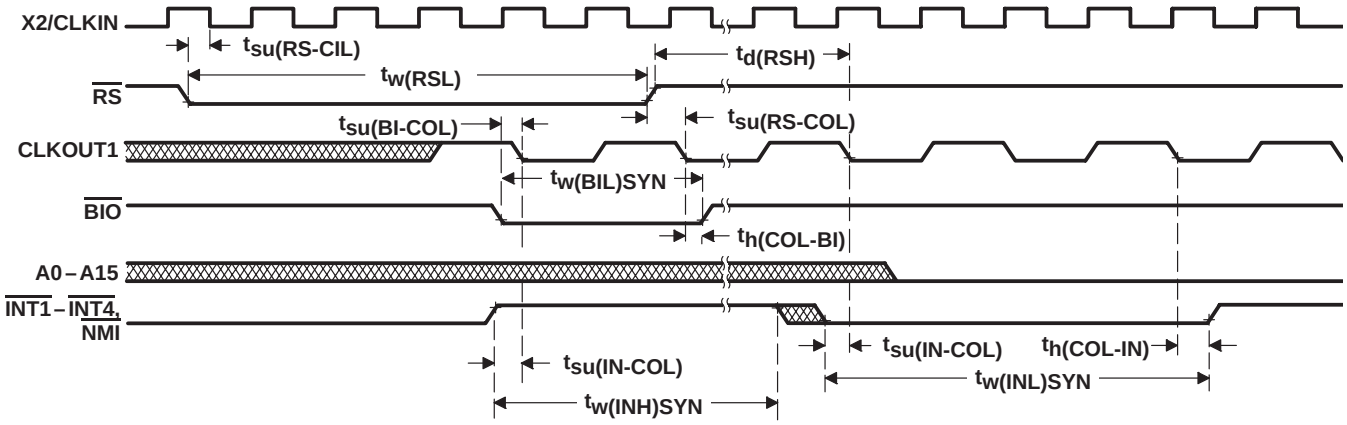


Figure 13. Reset, Interrupt, and $\overline{\text{BIO}}$ Timings

EXTERNAL FLAG (XF) AND TOUT

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 14)

PARAMETER		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{w(TUH)}$	Pulse duration, TOUT high	2H – 9		2H – 9		ns
$t_{d(CO-TU)}$	Delay time, CLKOUT1 falling to TOUT	– 6	6	– 6	6	ns
$t_{d(CO-XFV)}$	Delay time, XF valid after CLKOUT1	0	9	0	9	ns

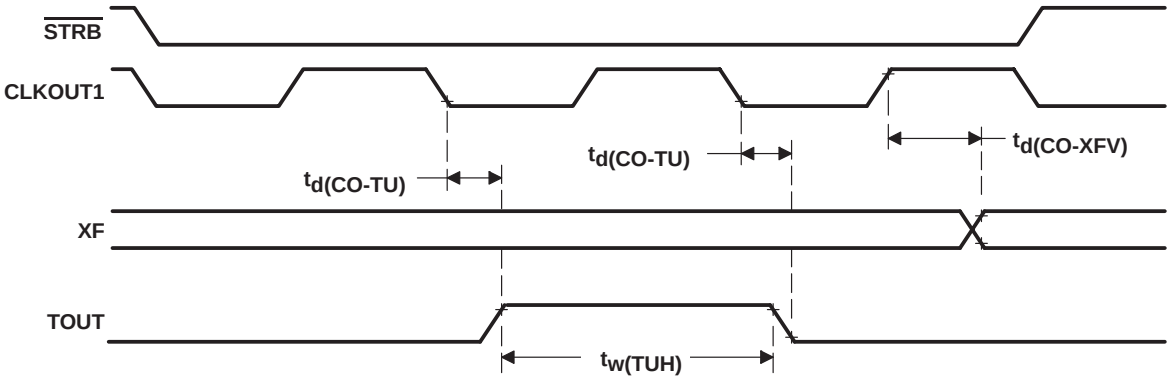


Figure 14. TOUT and XF Timing Example

EXTERNAL DMA

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Figure 15)

PARAMETER		'320C511A-92 '320C511A-96		'320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{d(HOL-HAL)}$	Delay time, $\overline{HOLD}$ low to $\overline{HOLDA}$ low	4H	†	4H	†	ns
$t_{d(HOH-HAH)}$	Delay time, $\overline{HOLD}$ high before $\overline{HOLDA}$ high	2H		2H		ns
$t_{h(AZ-HAL)}$	Hold time, $\overline{HOLDA}$ low after address high-impedance‡	H – 10§		H – 8§		ns
$t_{en(HAH-Ad)}$	Enable time, $\overline{HOLDA}$ high to address driven	H – 4§		H – 3§		ns

† $\overline{HOLD}$ is not acknowledged until current external access request is complete.
‡ This parameter includes all memory control lines.
§ Values are derived from characterization data and are not tested.

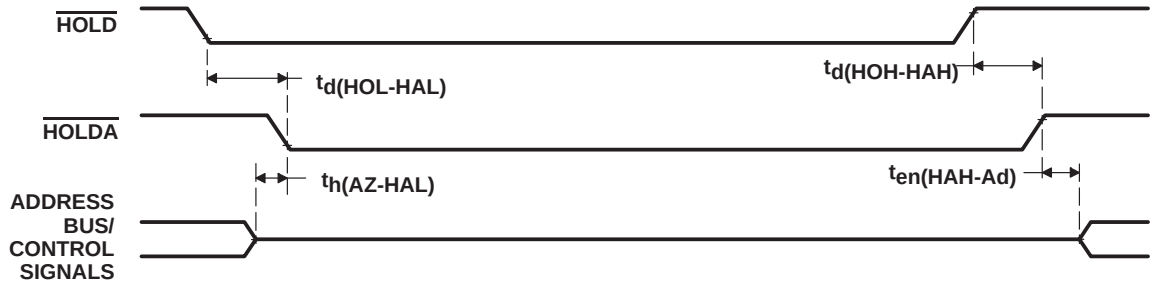


Figure 15. External DMA Timing

SERIAL-PORT RECEIVE

timing requirements over recommended ranges of supply voltage and operating case temperature [H = 0.5$t_{c(CO)}$] (see Figure 16)

		’320C511A-92 ’320C511A-96		’320C511A-100		UNIT
		MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial-port clock	5.2H†	‡	5.2H†	‡	ns
$t_{f(SCK)}$	Fall time, serial-port clock		6§		6§	ns
$t_{r(SCK)}$	Rise time, serial-port clock		6§		6§	ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	2.1H†		2.1H†		ns
$t_{su(FS-CK)}$	Setup time, FSR before CLKR falling edge	7		6		ns
$t_{su(DR-CK)}$	Setup time, DR before CLKR falling edge	7		6		ns
$t_{h(CK-FS)}$	Hold time, FSR after CLKR falling edge	7		6		ns
$t_{h(CK-DR)}$	Hold time, DR valid after CLKR falling edge	7		6		ns

† Values are assured by design but are not tested.
‡ The serial-port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching ∞. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
§ Values are derived from characterization data and are not tested.

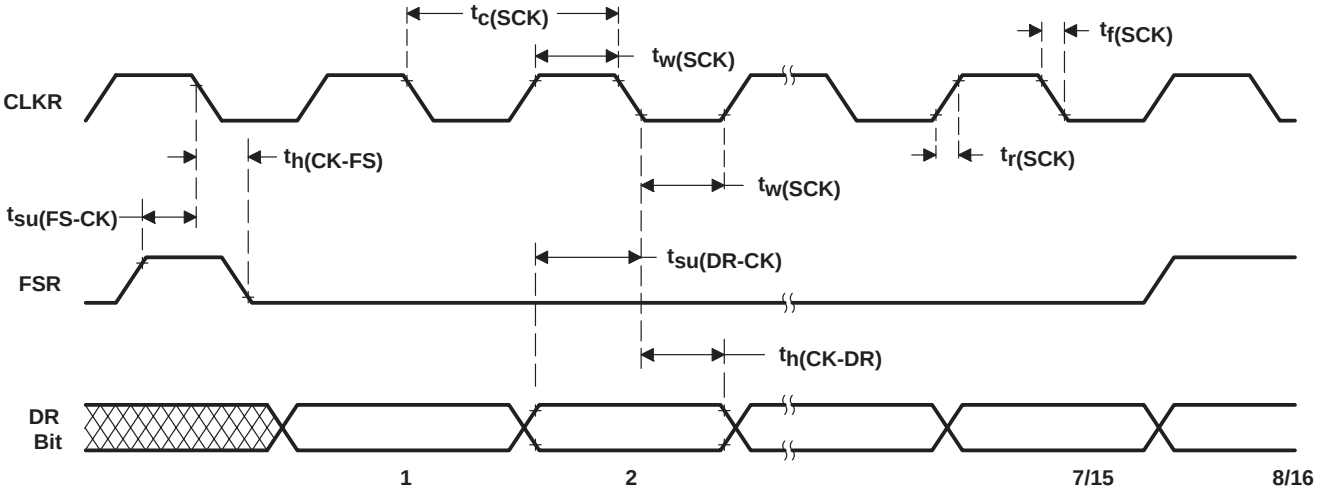


Figure 16. Serial-Port Receive Timing

SERIAL-PORT TRANSMIT TIMING, EXTERNAL CLOCKS AND EXTERNAL FRAMES

switching characteristics over recommended operating conditions (see Note 4 and Figure 17)

PARAMETER		MIN	MAX	UNIT
$t_{d(CXH-DXV)}$	Delay time, DX valid after CLKX high		25	ns
$t_{dis(CXH-DX)}$	Disable time, DX invalid after CLKX high		40†	ns
$t_{h(CXH-DXV)}$	Hold time, DX valid after CLKX high	– 5		ns

timing requirements over recommended ranges of supply voltage and operating case temperature [H = $0.5t_{c(CO)}$] (see Note 4 and Figure 17)

		’320C511A-92		’320C511A-96		’320C511A-100		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
$t_{c(SCK)}$	Cycle time, serial-port clock	5.2H‡	§	5.2H‡	§	5.2H‡	§	ns
$t_{f(SCK)}$	Fall time, serial-port clock		6†		6†		6†	ns
$t_{r(SCK)}$	Rise time, serial-port clock		6†		6†		6†	ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	2.1H‡		2.1H‡		2.1H‡		ns
$t_{d(CXH-FXH)}$	Delay time, FSX high after CLKX high		3H – 8		3H – 8		3H – 5	ns
$t_{h(CXL-FXL)}$	Hold time, FSX low after CLKX low	7		7		6		ns
$t_{h(CXH-FXL)}$	Hold time, FSX low after CLKX high		3H – 8¶		3H – 8¶		3H – 5¶	ns

† Values are derived from characterization data and are not tested.
‡ Values are assured by design but are not tested.
§ The serial-port design is fully static and, therefore, can operate with $t_{c(SCK)}$ approaching ∞. It is characterized approaching an input frequency of 0 Hz but tested at a much higher frequency to minimize test time.
¶ If the FSX pulse does not meet this specification, the first bit of serial data is driven on the DX pin until the falling edge of FSX. After the falling edge of FSX, data is shifted out on the DX pin. The transmit-buffer-empty interrupt is generated when the $t_{h(CXL-FXL)}$ and $t_{h(CXH-FXL)}$ specifications are met.

NOTE 4: Internal clock with external FSX and vice versa are also allowable. However, FSX timings to CLKX always are defined depending on the source of FSX, and CLKX timings always are dependent on the source of CLKX. Specifically, the relationship of FSX to CLKX is independent of the source of CLKX.

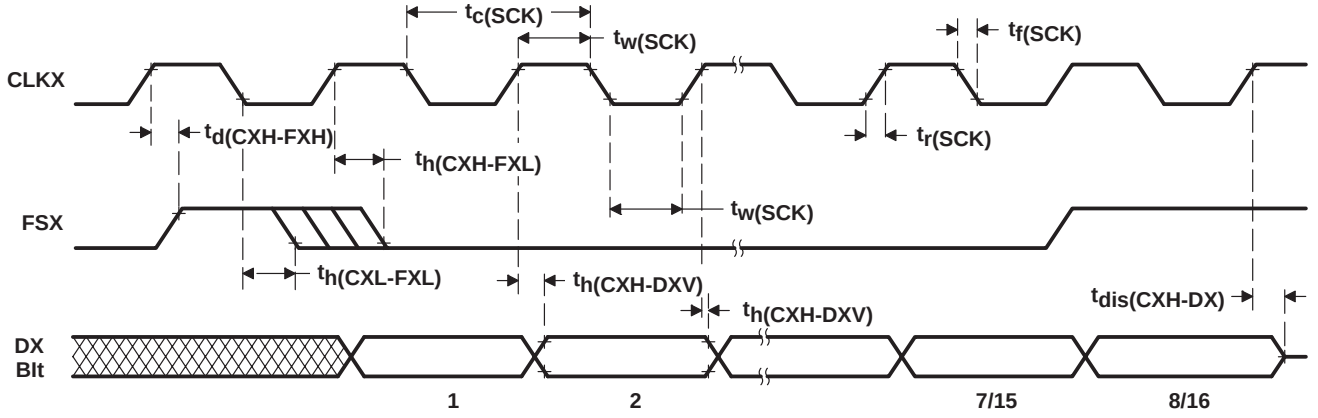


Figure 17. Serial-Port Transmit Timing of External Clocks and External Frames

SERIAL-PORT TRANSMIT TIMING, INTERNAL CLOCKS AND INTERNAL FRAMES

switching characteristics over recommended operating conditions [H = 0.5$t_{c(CO)}$] (see Note 4 and Figure 18)

PARAMETER		'320C511A-92 '320C511A-96			'320C511A-100			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	
$t_{d(CX-FX)}$	Delay time, CLKX rising edge to FSX	– 4		18	– 4		18	ns
$t_{d(CX-DX)}$	Delay time, CLKX rising edge to DX			18			18	ns
$t_{dis(CX-DX)}$	Disable time, CLKX rising edge to DX			29†			29†	ns
$t_{c(SCK)}$	Cycle time, serial-port clock		8H			8H		ns
$t_{f(SCK)}$	Fall time, serial-port clock		4			4		ns
$t_{r(SCK)}$	Rise time, serial-port clock		4			4		ns
$t_{w(SCK)}$	Pulse duration, serial-port clock low/high	4H – 14			4H – 14			ns
$t_{h(CXH-DXV)}$	Hold time, DX valid after CLKX high	– 4			– 4			ns

† Values are derived from characterization data and are not tested.

NOTE 4: Internal clock with external FSX and vice versa are also allowable. However, FSX timings to CLKX always are defined depending on the source of FSX, and CLKX timings always are dependent on the source of CLKX. Specifically, the relationship of FSX to CLKX is independent of the source of CLKX.

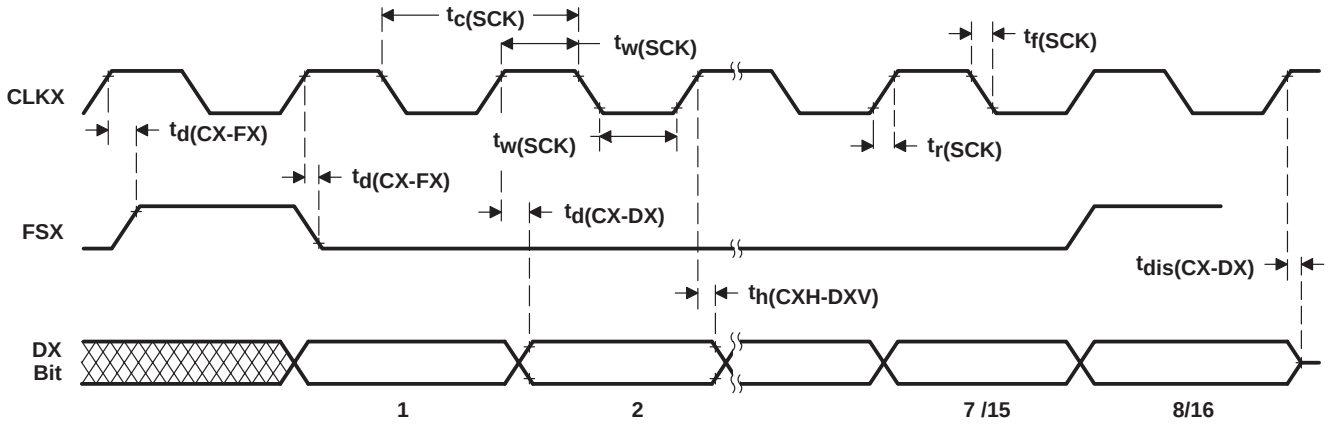
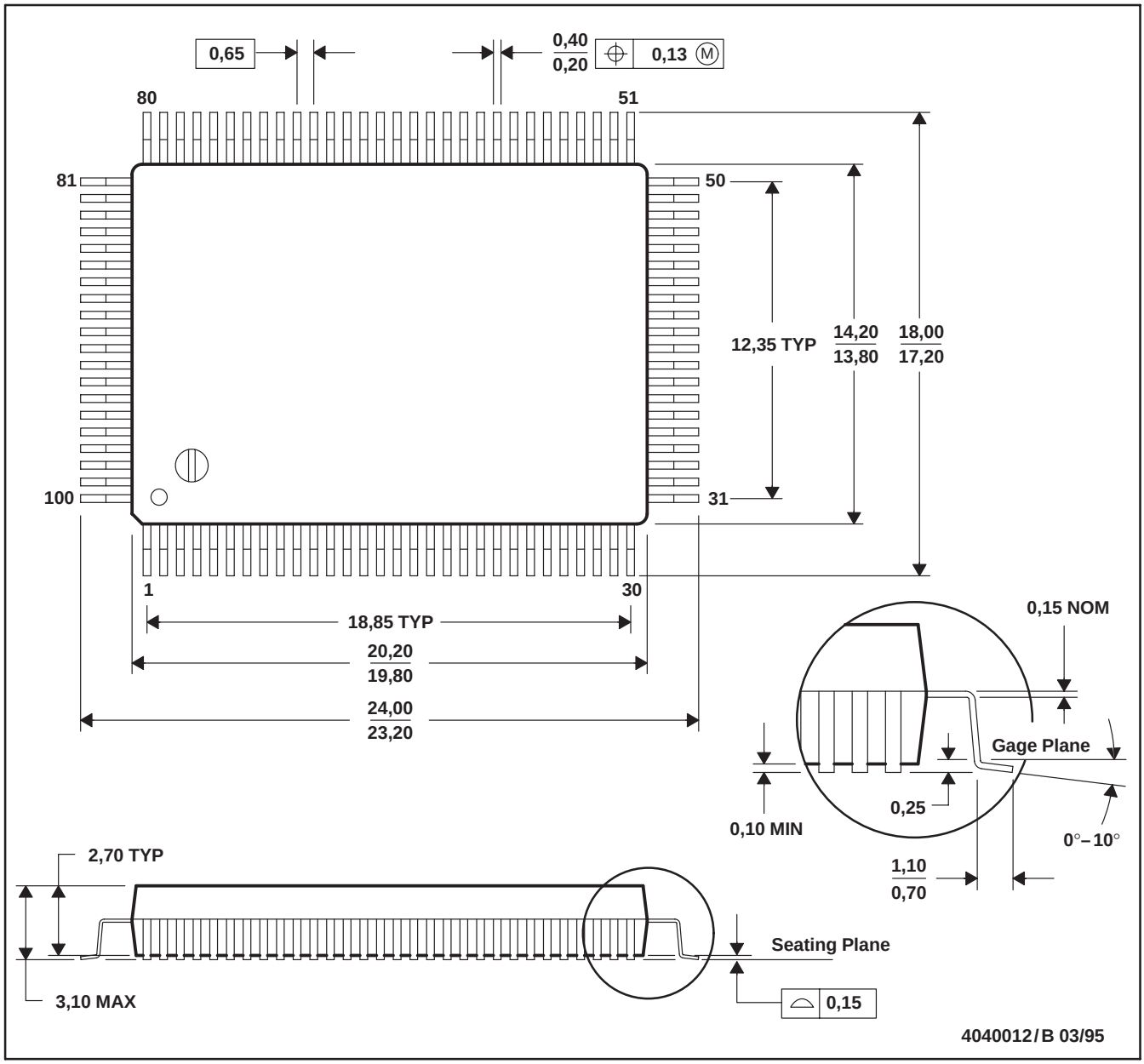


Figure 18. Serial-Port Transmit Timing of Internal Clocks and Internal Frames

MECHANICAL DATA

PJ (R-PQFP-G100) **PLASTIC QUAD FLATPACK**



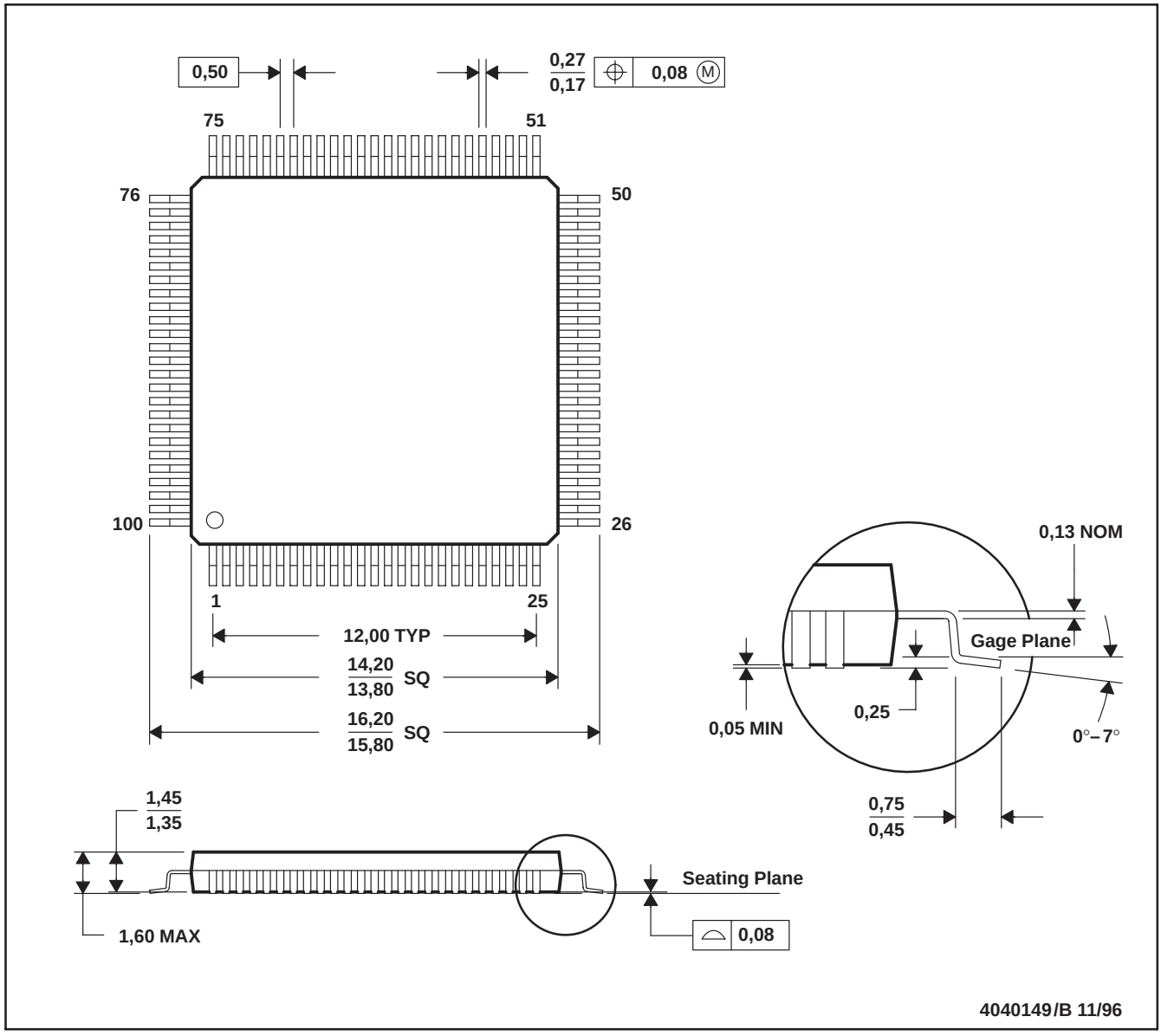
NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Contact field sales office to determine if a tighter coplanarity requirement is available for this package.

Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	78
$R_{\Theta JC}$	13

MECHANICAL DATA

PZ (S-PQFP-G100) **PLASTIC QUAD FLATPACK**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026

Thermal Resistance Characteristics

PARAMETER	°C/W
$R_{\Theta JA}$	58
$R_{\Theta JC}$	10

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.