SPRU010

TMS32011 User's Guide

Digital Signal Processor Products

Texas Instruments

TMS32011
User's Guide

Digital Signal Processor Products

Texas
Instruments

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes in the devices or the device specifications identified in this publication without notice. TI advises its customers to obtain the latest version of device specifications to verify, before placing orders, that the information being relied upon by the customer is current.

In the absence of written agreement to the contrary, TI assumes no liability for TI applications assistance, customer's product design, or infringement of patents or copyrights of third parties by or arising from use of semiconductor devices described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor devices might be or are used.

Contents

Section *Page*

Illustrations

Figure *Page*

Tables

Table *Page*

1. Introduction

The TMS32011, a spinoff from the TMS32010, is a dedicated microcomputer developed for large-volume applications. The TMS32011 is designed to support a wide range of Digital Signal Processing (DSP) applications. The TMS320 family supports digital signal processing applications, such as telecommunications, modems, image processing, speech processing, spectrum analysis, audio processing, digital filtering, high-speed control, graphics, and other computation-intensive applications.

The TMS32011 incorporates all the TMS32010 hardware features, such as a 200-ns instruction cycle time, 16 x 16 parallel multiplier, and 32-bit ALU/accumulator. The instruction sets are object-code compatible, enabling existing TMS32010 development tools to be used in TMS32011 applications. The device also provides a dual-channel serial port capable of full-duplex direct interface to combo-codec circuits. Two receive and two transmit channels permit direct interface to two combo-codec circuits in telecommunications systems.

Development tools and applications support are key advantages to using the TMS32011. Existing TMS32010 development tools (macro assembler/linker, software simulator, evaluation module, and full-speed in-circuit emulator) provide advanced development operations for rapid application design cycles. The device features a peripheral mode that allows utilization of the TMS32010 development tools for TMS32011 application development.

1.1 General Description

The TMS32011 consists of a TMS320M10 microcomputer with added hardware for enhanced performance in serial interface applications. The device features a dual-channel serial port capable of full-duplex communication and direct interface to combo-codec circuits in telecommunications systems. Framing pulses for serial-port transmission are selectable as either external to the device or internally generated. Timing logic supports either connection to PCM highways via combo-codec circuits or standalone serial communications with an internally generated serial-port clock.

On-chip hardware performs μ-law or A-law conversions between sign-magnitude linear PCM and 8;bit logarithmically compressed PCM code words. Conversion is accomplished according to either of the two international standards: the μ-255 law companding characteristic used in the United States and Japan, or the European A-law companding characteristic. These conversions are usually performed in software by the host processor (see the application report, "Companding Routines for the TMS32010/TMS32020," in the book, *Digital Signal Processing Applications with the TMS320 Family,* published by Texas Instruments). Conversions are incorporated in hardware to improve the processor CPU program space and utilization.

1.2 Key Features

The following key features distinguish the TMS32011 microcomputer:

- Object-code compatible with TMS32010 instruction set
- Compatible with TMS32010 development support tools
- Peripheral mode to TMS32010 for application development
- Dual-channel serial port for full-duplex serial communication
- Direct interface to combo-codec and PCM highway systems
- Serial-port timer for standalone serial communications and PCM highway clocks
- Internal framing-pulse generation for serial communications or a programmable timer for control applications
- On-chip companding hardware for μ-law and A-law PCM conversions
- Eight input and eight output channels, six of which are user-definable
- Four maskable interrupts to fully support serial-port interface
- Programmable output flag for peripheral control.

The following TMS32010 features are common to the TMS32011:

- 200-ns instruction cycle
- 144-word on-chip data RAM
- 1536-word on-chip program ROM
- 16-bit instruction/data word
- 32-bit ALU/accumulator
- 16 x 16-bit multiply in 200 ns

- 0 to 16-bit barrel shifter
- 16-bit bidirectional data bus with 40-Mbit/s transfer rate
- Signed two's-complement fixed-point arithmetic
- Interrupt with full context save
- Single 5-V supply
- 40-pin dual-in-line package (DIP).

1.3 How To Use This Manual

This user's guide is designed as a supplement to the TMS32010 User's Guide. TMS32011 features are presented along with detailed descriptions of hardware configuration and operation. The user is referred to the TMS32010 User's Guide for a complete description of TMS320 architecture and design methodology.

The following table lists each section and briefly describes the section contents.

Section 2. Pinout and signal descriptions. Drawing of the 40-pin dual-in-line package for the TMS32011 microcomputer. A functional listing of the signals with their descriptions.

Section 3. Architecture. TMS32011 design description, hardware components, and their functions. Functional block diagram.

Section 4. Processor Resource Management. Detailed description of the on-chip hardware configuration. Assembly language coding examples of initialization, interrupt, and companding routines.

Section 5. Codec Interface. Hardware design technique for serial-port interface to combo-codec.

Section 6. Peripheral Mode Interface. A description of the TMS32011 peripheral mode for interfacing to TMS32010 development tools.

These appendices are included to provide additional information.

Appendix A. TMS32011 Data Sheet. Electrical specifications, timing, and mechanical data for the TMS32011.

Appendix B. TMS32010 Data Sheet. Electrical specifications, timing, and mechanical data for the TMS32010.

Appendix C. TMS32010-TMS32011 System Migration. A description of the changes in the hardware and programming features of the TMS32011 from those of the TMS32010.

Appendix D. TMS32011 Development Support/Part Order Information. A listing of TMS32010 support tools for performing TMS32011 development operations.

2. Pinout and Signal Descriptions

The TMS32011 is produced in a 40-pin dual-in-line package (DIP). Mechanical data is provided in Appendix A, the TMS32011 Data Sheet. Figure 2-1 shows a pinout of the TMS32011 package with signal location/pin numbers.

Table 2-1 lists the TMS32011 signals, their pin numbers, and input, output, or high-impedance status (I/O/Z). The function of each signal is described. The signals in Table 2-1 are grouped according to function.

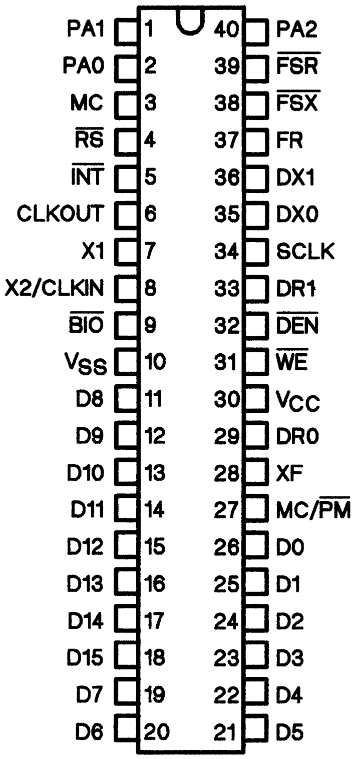


Figure 2-1. TMS32011 Pin Assignments

Table 2-1. TMS32011 Signal Descriptions

SIGNAL	PIN	I/O/Z†	DESCRIPTION
POWER			
V_{CC}	30	I	Power supply (+5 V NOM)
V_{SS}	10	I	Ground reference
CLOCKS			
X2/CLKIN	8	I	Input pin to the internal oscillator from the crystal (X2). If a crystal is not used, a clock may be input to the device on this pin (CLKIN).
X1	7	O	Crystal output pin from the internal oscillator. If a crystal is not used, this pin should be left unconnected.
CLKOUT	6	O	Clock output signal. The frequency is 1/4 the oscillator input (external oscillator) or crystal frequency (internal oscillator).
CONTROL			
$\overline{\text{WE}}$	31	O	Write enable. When active (low), $\overline{\text{WE}}$ indicates that valid output data from the TMS32011 is available on the data bus (D15-D0). $\overline{\text{WE}}$ is only active during the first cycle of the OUT instruction. $\overline{\text{DEN}}$ is always inactive (high) when $\overline{\text{WE}}$ is active.
$\overline{\text{DEN}}$	32	O	Data enable. When active (low), $\overline{\text{DEN}}$ indicates that the TMS32011 is accepting data from the data bus (D15-D0). $\overline{\text{DEN}}$ is only active during the first cycle of the IN instruction. $\overline{\text{WE}}$ is always inactive (high) when $\overline{\text{DEN}}$ is active.
INTERRUPTS AND EXTERNAL CONTROL			
$\overline{\text{RS}}$	4	I	Reset. Used to initialize the device. When an active low is placed on the $\overline{\text{RS}}$ pin for a minimum of five clock cycles, $\overline{\text{DEN}}$ and $\overline{\text{WE}}$ are forced high, and the data bus (D15-D0) goes to the high-impedance state. The serial-port clock and transmit outputs also go to the high-impedance state. The program counter (PC) and the port address bus (PA2-PA0) are then synchronously cleared after the next complete clock cycle from the falling edge of $\overline{\text{RS}}$.
$\overline{\text{INT}}$	5	I	Interrupt. The interrupt signal is generated by applying a negative-going edge to the $\overline{\text{INT}}$ pin. The edge is used to latch the system control register flag bit (CR0) until an interrupt is granted by the device.
$\overline{\text{BIO}}$	9	I	I/O branch control. If $\overline{\text{BIO}}$ is active (low) upon execution of the BIOZ instruction, the device branches to the address specified by the instruction.
XF	28	O	External logic output flag. Programmable via system control register bit 10 (CR10). This pin is the direct output of the CR10 latch.
MICROCOMPUTER CONTROL			
MC	3	I	Microcomputer-mode select. This pin is used for internal Texas Instruments testing purposes and should be tied to V_{CC} for the device to function correctly.
MC/$\overline{\text{PM}}$	27	I	Microcomputer or peripheral mode select. When high, the device functions in normal microcomputer mode. When low, the device functions as a peripheral to a TMS32010 for developing TMS32011 applications. See Section 6 for a detailed description of the peripheral mode.
BIDIRECTIONAL DATA BUS			
D15 MSB	18	I/O/Z	External parallel data bus D15 through D0. The data bus is always in the high-impedance state, except when $\overline{\text{WE}}$ is active (low) or when executing an IN instruction from port 0 or port 1.
D14	17		
D13	16		
D12	15		
D11	14		
D10	13		
D9	12		
D8	11		
D7	19		
D6	20		
D5	21		
D4	22		
D3	23		
D2	24		
D1	25		
D0 LSB	26		

† Input/Output/High-impedance state

Table 2-1. TMS32011 Signal Descriptions (Concluded)

SIGNAL	PIN	I/O/Z†	DESCRIPTION
PORT ADDRESS BUS			
PA2 MSB PA1 PA0 LSB	40 1 2	O	Port addresses PA2 through PA0. During the IN and OUT instructions, these pins carry the port address. These pins always output the three LSBs of the program counter.
SERIAL PORT SIGNALS			
SCLK	34	I/O/Z	Serial-port clock. Master clock for transmitting and receiving serial-port data. Configurable as an input or output. SCLK must always be present for serial-port operation. As an input, SCLK is the external clock that controls data transfers with the serial port. As an output, SCLK provides the serial clock for data transfers and framing-pulse synchronization. Its frequency is derived from the TMS32011 system clock, X2/CLKIN, and system control register bits CR27-CR24. Reset ($\overline{\text{RS}}$) forces SCLK to the high-impedance state.
DX1 DX0	36 35	O/Z O/Z	Serial-port transmit-channel outputs. Serial data is transmitted from the transmit registers on these pins. These outputs are in the high-impedance state when not transmitting.
DR1 DR0	33 29	I I	Serial-port receive-channel inputs. Serial data is received in the receive registers via these pins.
$\overline{\text{FSR}}$	39	I	External serial-port receive-framing input. If external framing is enabled via the system control register, data is received via the receive pins (DR1 and DR0) on the active (low) FSR input. The falling edge of FSR initiates the receive process, and the rising edge sets the flag bit (CR1) in the system control register, causing an interrupt to occur if enabled.
$\overline{\text{FSX}}$	38	I	External serial-port transmit-framing input. If external framing is enabled, data is transmitted on the transmit pins (DX1,DX0) on the active (low) FSX input. The falling edge of FSX initiates the transmit process, and the rising edge sets the flag bit (CR2) in the system control register, causing an interrupt to occur if enabled.
FR	37	O	Internal serial-port framing output. If internal framing is enabled, serial-port transmit and receive operations occur simultaneously on an active (high) FR framing pulse. Both short and long FR pulses are selectable to provide fixed and variable data-rate framing pulses for combo-codec interface. The FR frequency is derived from the serial-port clock (SCLK) and system control register bits CR23-CR16.

† Input/Output/High-impedance state

3. Architecture

The TMS32011 microcomputer architectural design is identical to that of the TMS32010. The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of the instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into RAM, eliminating the need for separate coefficient ROM. It also makes available immediate instructions and subroutine calls based on computed values.

The architectural elements common to both the TMS32011 and TMS32010 are briefly described in this section. A detailed description of the architecture can be found in the TMS32010 User's Guide. References to the TMS32010 User's Guide are enclosed in brackets; e.g., [Section X.X] refers to the section in the TMS32010 User's Guide, which discusses a particular architectural element.

Major topics discussed in this section are listed below.

- Architectural Overview (Section 3.1 on page 3-2)
  Functional block diagram
- Arithmetic Elements (Section 3.2 on page 3-4)
  ALU
  Accumulator
  Multiplier
  Shifters
- Memory Configuration (Section 3.3 on page 3-5)
  Data memory
  Registers
  Program memory
  Program counter and stack
- Hardware Control (Section 3.4 on page 3-6)
  System control register
  Interrupts
  Input/output functions
- Serial Port (Section 3.5 on page 3-9)
  Receive/transmit registers
  Timing and framing control
- Companding Hardware (Section 3.6 on page 3-12)
  μ-law/A-law encoder/decoder

3.1 Architectural Overview

The TMS32011 consists of a TMS320M10 microcomputer with added hardware for interfacing ease in serial applications. This additional hardware is interfaced to the microcomputer portion of the device via the external data bus (D15-D0).

Two registers, collectively called the system control register, control the added hardware. The lower control register bits (CR15-CR0) control interrupts, companding modes of operation, and serial-port functions. The upper control register bits (CR28-CR16) control the internal framing (FR) signal output, the serial-port clock (SCLK) frequency as an output, and the internal framing output (FR) pulse width. These two control registers are mapped into I/O port 0 and port 1.

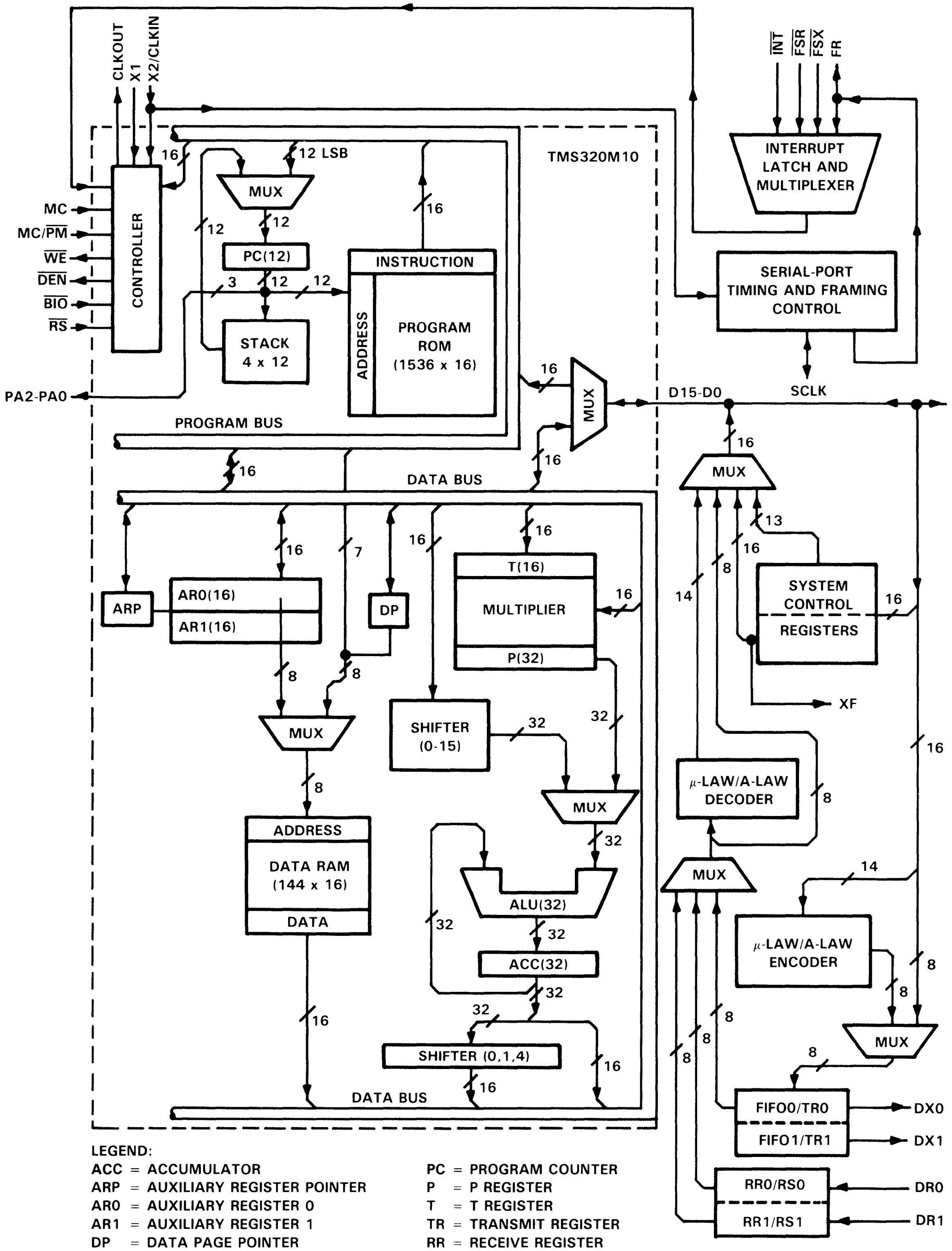
The dual-channel serial port is capable of full-duplex communications and direct interface to combo-codec circuits. The serial port operates with an 8-bit sample length for log PCM data. Two transmit registers and two receive registers are mapped into I/O port 1. The registers are accessed with OUT and IN instructions to port 1.

The companding hardware performs conversions between sign-magnitude linear PCM and 8-bit logarithmically compressed PCM data samples. Both international companding standards, u-255 law and A-law, are available for companding operation. The hardware operates on either serial data from the port or parallel data from the on-chip data RAM.

All hardware features of the TMS320M10 are still valid. The modified Harvard architecture provides increased flexibility. The 32-bit ALU and accumulator are unchanged to yield TMS32010 performance and flexibility. The shifters are also unchanged.

The architecture is built around two major buses: the program bus and the data bus. The program bus carries the instruction code and immediate operands from the program memory. The data bus interconnects various elements, such as the ALU/accumulator, multiplier, and data RAM. The serial-port logic, companding hardware, and system control register are multiplexed to the device through the external data bus.

The functional block diagram shown in Figure 3-1 outlines the principal blocks and data paths within the processor. Further details of functional blocks are given in the succeeding subsections. The block diagram also shows all of the device interface pins.



LEGEND:

ACC = ACCUMULATOR
ARP = AUXILIARY REGISTER POINTER
AR0 = AUXILIARY REGISTER 0
AR1 = AUXILIARY REGISTER 1
DP = DATA PAGE POINTER

PC = PROGRAM COUNTER
P = P REGISTER
T = T REGISTER
TR = TRANSMIT REGISTER
RR = RECEIVE REGISTER

Figure 3-1. Block Diagram of the TMS32011 Digital Signal Processor

3.2 Arithmetic Elements

The TMS32011 provides four basic arithmetic elements: the ALU, accumulator, multiplier, and shifters. All arithmetic operations are performed using two's-complement arithmetic.

Most arithmetic instructions access a word in data RAM, either directly or indirectly, and pass it through the barrel shifter. This shifter can left-shift a word 0 to 16 bits, depending on the value specified by the instruction. The data word then enters the ALU where it is loaded into or added/subtracted from the accumulator. After a result is obtained in the accumulator, it can be stored in data RAM. Since the accumulator operates with a 32-bit wordlength, both halves must be stored separately. A parallel left-shifter is present at the accumulator output to aid in scaling results as they are being moved to data RAM.

3.2.1 ALU

The ALU is a general-purpose arithmetic logic unit that operates with a 32-bit data word. The unit adds, subtracts, and performs logical operations. The accumulator is always the destination and the primary operand. [Section 2.2.1]

3.2.2 Accumulator

The accumulator stores the output from the ALU and is also often an input to the ALU. It operates with a 32-bit wordlength. The accumulator is divided into a high-order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing the high- and low-order accumulator words in memory. [Section 2.2.2]

3.2;3 Multiplier

The 16 x 16-bit parallel multiplier consists of three units: the T register, P register, and multiplier array. The T register is a 16-bit register that stores the multiplicand, and the P register is a 32-bit register that stores the product. The multiplier operation is unchanged from the TMS32010. [Section 2.2.3]

3.2.4 Shifters

Two shifters are available for manipulating data. The ALU barrel shifter performs a left-shift of 0 to 16 places on data memory words loaded into the ALU. The accumulator parallel shifter performs a left-shift of 0, 1, or 4 places on the entire accumulator for the SACH (store high-order accumulator word) instruction and places the high-order accumulator bits into data RAM. [Section 2.2.4]

3.3 Memory Configuration

The TMS32011 features 144 words of 16-bit data RAM and 1536 words of 16-bit program ROM. The device is identical to the TMS320M10 without the external-memory implementation. The microprocessor mode of the TMS32010 is not available on the TMS32011.

3.3.1 Data Memory

Data memory consists of 144 words of 16-bit on-chip RAM. All nonimmediate data operands reside within this RAM.

Sometimes it is convenient to store data operands in program ROM or external memory and read them into the on-chip RAM as they are needed. Two means are available for doing this. First, the TBLR (table read) instruction transfers data from on-chip program ROM to on-chip data RAM. The TMS32010 user should note that the TBLW (table write) instruction should not be used on the TMS32011. This instruction transferred data from the on-chip data RAM to external memory. The TMS32011 does not directly interface to external memory since the port address bits (PA2-PA0) are the only address lines external to the device. The second method of addressing off-chip data RAM is via the IN and OUT instructions. With some extra hardware, the IN and OUT instructions can be used to read and write from data RAM to large amounts of external storage addressed as a peripheral. [Section 6.1.3]

The TMS32011 uses the same three forms of data memory addressing (indirect, direct, and immediate) as for the TMS32010. Indirect addressing uses the lower eight bits of the auxiliary registers as the data memory address. Direct addressing uses the seven bits of the instruction word concatenated with the data page pointer to form the data memory address. Immediate addressing uses part of the instruction word for data rather than data RAM. [Section 2.3.1]

3.3.2 Registers

The two auxiliary registers and auxiliary register pointer are unchanged from the TMS32010. The two registers are used for temporary storage, indirect addressing of data memory, and loop control. The auxiliary register pointer is a single bit in the status register, which indicates the auxiliary register currently selected. [Section 2.4]

3.3.3 Program Memory

Program memory consists of 1536 words of 16-bit width. Of these 1536 words, 1524 are available for program use. Locations 1524 through 1535 are reserved for Texas Instruments testing purposes. [Section 2.5]

The TMS32010 user should note that there is no microprocessor mode of operation on the TMS32011. All program memory resides on-chip in the 1536-word ROM.

3.3.4 Program Counter and Stack

The program counter (PC) is a 12-bit register that contains the program memory address of the next instruction to be executed. The device reads the instruction from the program memory location addressed by the PC, and increments the PC in preparation for the next instruction prefetch. The PC is initialized to zero by activating the reset ($\overline{RS}$) line. The stack is 12 bits wide and four layers deep. [Section 2.6]

Note that the program counter uses only 11 of the 12 bits for internal memory addressing of 1.5K words. No external memory implementation is allowed on the TMS32011.

3.4 Hardware Control

The TMS32011 consists of a TMS320M10 with the added hardware of a system control register, interrupt latch, serial port, and companding hardware. The serial port and companding hardware are controlled via the system control register. Interrupts have been expanded to support the serial-port interface. Input/output to the device is via IN and OUT instructions to the I/O ports.

3.4.1 System Control Register

Operation of the serial-port and companding hardware is controlled by the 29 bits of the system control register. This control register accesses all serial-port and companding hardware, eliminating any additions to the TMS32010 instruction set. A block diagram is shown in Figure 3-2. The system control register is mapped into I/O port 0 and port 1.

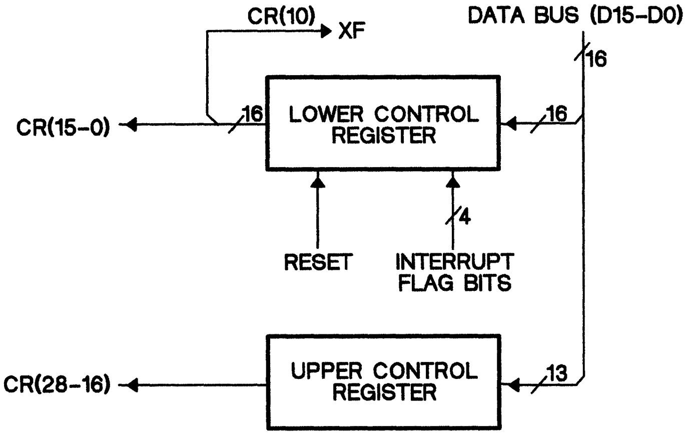


Figure 3-2. System Control Register

The lower 16 register bits (CR15-CR0) are accessed through port 0. These bits control interrupts, serial-port configuration, the external logic output flag, internal and external framing pulses, and the μ-law/A-law encoder and decoder. The interrupt inputs ($\overline{INT}$, $\overline{FSX}$, $\overline{FSR}$, and FR) are synchronized to CLKOUT and control the interrupt flag bits (CR3-CR0). The interrupts are maskable via the interrupt enable bits (CR7-CR4). Bit 8 (CR8) controls I/O port 1 configuration.

The upper 13 bits (CR28-CR16) are accessed through port 1. These bits control the internal framing pulse (FR) output frequency, serial-clock divide ratios, and pulse-width control for the FR framing pulse (see Section 4 for detailed control register bit descriptions).

The external data bus provides on-chip communication with the system control register, serial port, and companding hardware. With a write to port 0, the lower control register is addressed and data latched into the register by the rising edge of the write enable ($\overline{\text{WE}}$) signal. To write to the upper control register bits, bit 8 of the lower control register must be set to logic 1. If CR8 is logic 0, a write to port 1 accesses the serial port and companding hardware.

The control register bits are configured through OUT instructions to port 0 and port 1. $\overline{\text{WE}}$ goes low during the first cycle of the OUT instruction, enabling the port data onto the external data bus. The control register bits are latched on the rising edge of $\overline{\text{WE}}$. There is a propagation delay time for these bits to access the appropriate hardware (see Appendix A, the TMS32011 Data Sheet, for further timing information). An allowance for this write delay should be made when reconfiguring the control register. The most critical factor is receiving an external framing pulse while reconfiguring the control register. If an external framing pulse is received at that time, it may not be detected and the serial-port registers will contain random data (see Section 5 for further details).

3.4.2 Interrupts

The TMS32011 has four maskable interrupts: $\overline{\text{INT}}$, $\overline{\text{FSR}}$, $\overline{\text{FSX}}$, and FR. The TMS32011 has expanded the TMS32010 interrupts to fully support the serial-port interface. An interrupt latch and multiplexer is used to generate the master interrupt signal, which functions identically to the $\overline{\text{INT}}$ interrupt on the TMS32010. Thus, all the maskable interrupts have the same priority and require the use of interrupt polling techniques when multiple interrupts are enabled.

Two steps must be taken to enable an active interrupt to the device. First, the individual interrupt must be enabled by writing a logic 1 to the appropriate system control register bit (CR7-CR4). Then, the master interrupt circuitry is enabled via the EINT instruction. In a reset initialization routine, the interrupt flag bits (CR3-CR0) should be cleared before the EINT instruction to insure that a false interrupt does not occur (see Section 4.3 for detailed interrupt bit descriptions).

The interrupt latch synchronizes all interrupts to the device output clock (CLKOUT). A block diagram is shown in Figure 3-3. The external interrupt ($\overline{\text{INT}}$) is either an asynchronous input to the device for external control or a master processor interrupt signal. The other three interrupts are all associated with the serial-port framing signals, although the external framing pulse interrupts ($\overline{\text{FSX}}$ and $\overline{\text{FSR}}$) may be used as system interrupts when not being utilized by the serial port.

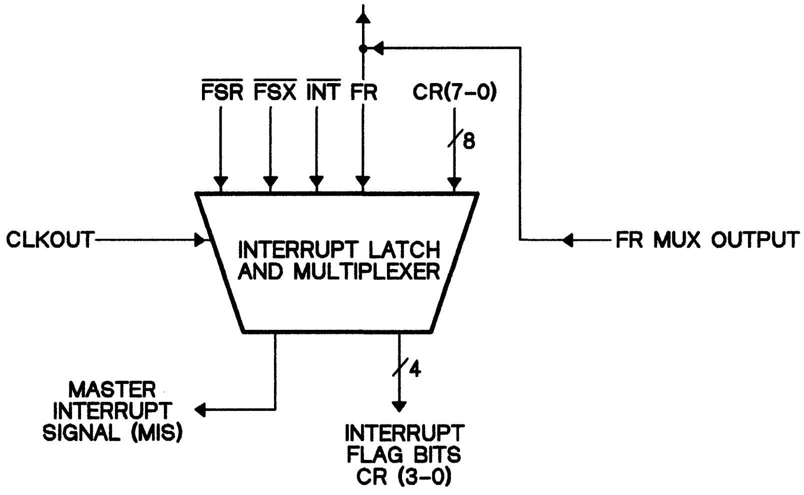

$\overline{\text{FSR}}$ $\overline{\text{FSX}}$ $\overline{\text{INT}}$ FR
CR(7-0)
8
CLKOUT
INTERRUPT LATCH AND MULTIPLEXER
FR MUX OUTPUT
MASTER INTERRUPT SIGNAL (MIS)
4
INTERRUPT FLAG BITS CR (3-0)

Figure 3-3. Interrupt Latch and Multiplexer

Due to the asynchronous operation of the interrupts, the time between the occurrence of an active interrupt signal and the device actually vectoring to ROM location 2 is four CLKOUT cycles (see Appendix A, the TMS32011 Data Sheet, for further timing information).

3.4.3 Input/Output Functions

The external address bus consists of three port address output pins (PA2-PA0). These address pins output the port address during OUT and IN instructions. PA2-PA0 output the three LSBs of the program counter, and are inputs only in the peripheral mode. These three pins address the serial port and companding hardware. During reset, the pins along with the program counter are synchronously cleared to zero during the cycle following $\overline{\text{RS}}$ low. Because all program and data memory are contained on-chip, only these three address lines go external to the device. The TMS32011 user who has experience with the TMS32010 should note the absence of the memory enable ($\overline{\text{MEN}}$) signal on the TMS32011 since all instruction execution is from on-chip program ROM.

Input/output of data to and from a peripheral is accomplished by the IN and OUT instructions. Data is transferred over the 16-bit data bus to and from data memory by two independent strobes: data enable ($\overline{\text{DEN}}$) and write enable ($\overline{\text{WE}}$).

The bidirectional external data bus is always in the high-impedance state, except when $\overline{\text{WE}}$ is active (low) or during an IN instruction from port 0 or port 1. $\overline{\text{WE}}$ goes low during the first cycle of the OUT instruction to provide the write strobe for writing data to a peripheral. During an IN from port 0 or port 1, the external data bus is driven to transfer the data from the serial-port peripheral onto the external data bus and then into data RAM.

The system control register, serial-port transmit and receive registers, and the companding hardware have been mapped into I/O ports 0 and 1. During an OUT or IN instruction to port 0 or port 1, data appears on the external data bus (D15-D0). The data bus is not in the high-impedance state while accessing these dedicated I/O ports. Peripheral device interface should be to port addresses 2 through 7 to prevent bus conflicts with the system control register and serial port. Six 16-bit multiplexed input ports and six 16-bit multiplexed output ports are available for interfacing to peripheral devices.

Execution of an IN instruction generates the $\overline{\text{DEN}}$ strobe for transferring data from a peripheral device to data RAM. The IN instruction is the only instruction for which $\overline{\text{DEN}}$ becomes active. Execution of an OUT instruction generates the $\overline{\text{WE}}$ strobe for transferring data from data RAM to a peripheral device. $\overline{\text{WE}}$ becomes active only during the OUT instruction (see Appendix A, the TMS32011 Data Sheet, for further timing information).

3.5 Serial Port

The dual-channel serial port is capable of full-duplex serial communications and direct interface to combo-codec PCM systems. A block diagram is shown in Figure 3-4. The serial port consists of two receive and two transmit registers that operate with 8-bit data samples. The serial port is mapped into I/O port 1 and is accessed through OUT and IN instructions. All data transfers are MSB first.

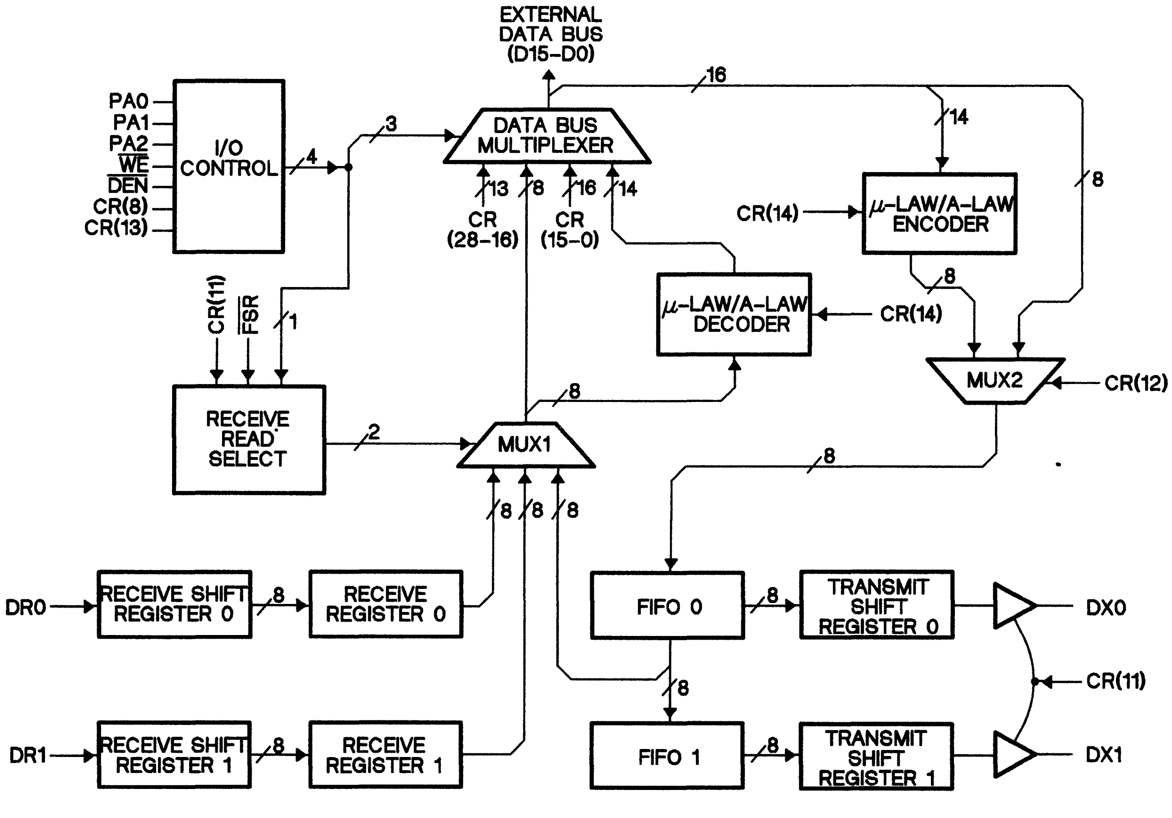


Figure 3-4. Serial Port and Companding Hardware

3.5.1 Receive Registers

Two receive registers are mapped into I/O port 1 via the port decode logic. Data is clocked into the shift registers on the next eight negative serial clock (SCLK) transitions after an active framing pulse is detected. SCLK controls the bit-level timing for all serial-port data transfers.

On an active framing pulse, serial data is clocked into the receive registers from the DR pins. Channel 0 data is received in shift register 0 from pin DR0, and channel 1 data is received in shift register 1 from pin DR1. To read the data from the registers, an IN instruction is executed from port 1. On the first IN instruction after a framing pulse, channel 0 data is output onto the external data bus. On the second IN, channel 1 data is output onto the external data bus.

An active framing pulse initiates the receive operation. External framing pulses ($\overline{\text{FSR}}$) are active low, and the internal framing (FR) signal is active high. With external framing ($\overline{\text{FSR}}$), the falling edge of the framing pulse gates the serial-port clock to the receive shift registers, and the data is clocked into the shift registers on the next eight consecutive negative transitions of the clock. The rising edge of the framing pulse transfers the data from the receive shift registers to the receive registers and sets the $\overline{\text{FSR}}$ flag bit (CR1) in the system control register, causing an interrupt to occur if the $\overline{\text{FSR}}$ is enabled.

Internal (FR) framing pulses are selectable for either fixed data-rate or variable data-rate modes for combo-codec interface. With the fixed data-rate mode, the FR pulse is one SCLK cycle wide, and appears in the cycle preceding the first data bit. The falling edge of the pulse initiates both the transmit and receive operations. Received data is clocked into the receive shift registers on the next eight consecutive negative transitions of the clock. After data bit 8 has been received, data is transferred from the receive shift registers to the receive registers, and an interrupt is generated when the FR flag bit (CR3) is set in the system control register, thus causing an interrupt to occur if enabled. In the variable data-rate mode, the FR pulse is eight SCLK cycles wide, and appears in the same SCLK cycle as the first data bit. The rising edge of the pulse initiates the transmit and receive operations. The falling edge of the pulse transfers data from the receive shift registers to the receive registers and sets the FR flag bit (CR3) in the system control register, causing an interrupt to occur if enabled.

3.5.2 Transmit Registers

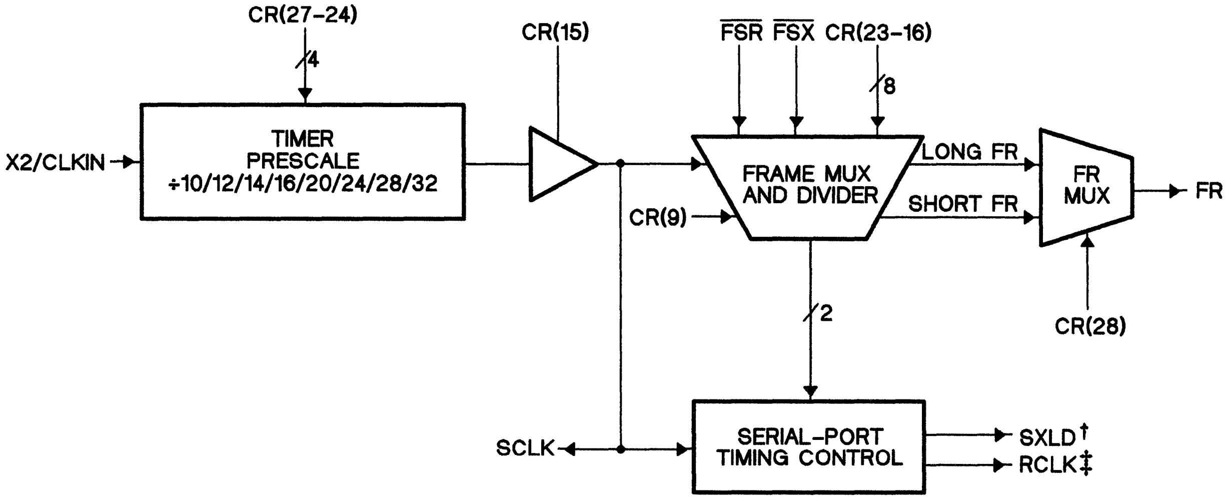
Two transmit registers are mapped into I/O port 1 via the port decode logic. The transmit registers are connected to the port 1 data bus in a First-In First-Out (FIFO) configuration. On the first OUT instruction to port 1 after a framing pulse, the data to be transmitted is put into FIFO buffer 0. On the next framing pulse, the buffer contents are latched into transmit shift register 0 and the data is transmitted on channel 0 (pin DX0) on the next eight positive transitions of the serial-port clock (SCLK). Data sent to port 1 is always put into the FIFO buffers. Only when control register bit 11 (CR11) is high will the data be enabled onto the transmit pins. The transmit pins are in the high-impedance state when not transmitting.

When two OUT instructions to port 1 are executed between framing pulses, both FIFO buffers are loaded with data for transmission. The first OUT instruction loads data into FIFO buffer 0. The second OUT pushes the data from FIFO buffer 0 into FIFO buffer 1 and puts the new data into FIFO buffer 0. On an active framing pulse edge, the FIFO buffer contents are latched into the transmit shift registers and the data clocked out on the next eight consecutive positive transitions of SCLK. Thus, for single-channel operation, only one OUT to port 1 should be executed between framing pulses to insure data transmission on channel 0. Only FIFO buffer 0 may be read back to the serial-port data bus by an IN instruction. This feature is used for the parallel companding mode.

Both transmit channels always output data on an active framing pulse when CR11 is high. During single-channel operation (using channel 0), channel 1 still transmits the data from transmit register 1. Transmit-channel 1 cannot be disabled during single-channel operation.

3.5.3 Timing and Framing Control

The serial-port timing and framing control is shown in Figure 3-5. The serial-port clock (SCLK) provides the timing control for data transfers with the serial port. It may be configured as either an input or output through the control register. As an input, SCLK is an external serial system clock that provides the framing synchronization and timing for the serial port. As an output, SCLK provides the system clock for standalone serial applications and is derived from the microcomputer system clock (X2/CLKIN). Prescale divide ratios are selectable through the system control register.



†SXLD = Load transmit shift registers (TR0 and TR1) from FIFO 0 and FIFO 1.
‡RCLK = Load receive registers (RR0 and RR1) from receive shift registers.

Figure 3-5. Serial-Port Timing and Framing Control

The serial-port clock prescaler determines the divide ratio for SCLK as an output. The TMS32011 system clock (X2/CLKIN) is input to the prescaler, along with control register bits CR27-CR24. These bits determine the divide ratio, selectable as divide by 10, 12, 14, 16, 20, 24, 28, and 32. The divide ratios are available only for SCLK when it is configured as an output from the device (see Section 4 for control register bit configurations).

The frame multiplexer determines which framing pulses cause serial-port data transfers to occur and configures the internal framing pulse (FR) frequency. The inputs to the multiplexer are SCLK, control register bit 9 (CR9), control register bits CR23-CR16, external transmit framing (FSX) pulse, and external receive framing (FSR) pulse. The outputs of the multiplexer go to the serial-port control for receive and transmit timing generation for the serial-port registers and to the FR multiplexer for determining which FR framing pulse will be generated.

SCLK is an input to the frame divider for the FR divide ratio determination. Control register bits CR23-CR16 determine the divide ratio, which is equal to SCLK/[(CR23-CR16) + 2]. The outputs of the frame counter are input to the FR multiplexer for selection of long or short FR pulses. The short FR pulse provides fixed data-rate framing pulses for standalone serial interface to the Texas Instruments TCM29XX family of combo-codec circuits. The long FR framing pulse provides variable data-rate framing pulses to the combo-codec.

The FR frequency is determined at the beginning of the framing pulse cycle. When reconfiguring the frequency, the upper control register bits determine the new divide ratio. However, the new frequency is not implemented until the next FR framing pulse.

3.6 Companding Hardware

The on-chip companding hardware enables the TMS32011 to compress and expand data in either μ-law or A-law format. Configuration and connections of the encoder and decoder (see Figure 3-4) are controlled through the system control register. The hardware operates on sign-magnitude data that must be converted to two's-complement notation for computations within the microcomputer. Note that for μ-law encoding, the bias of 33 must be added to the magnitude before encoding; likewise, after μ-law decoding, the bias of 33 must be subtracted from the magnitude. For A-law companding, no bias is required.

Data may be companded via four modes: serial-port encode, serial-port decode, parallel encode, and parallel decode. In the serial mode, transmitted data is encoded according to the specified companding law, and received data is decoded to sign-magnitude format. In the parallel modes, encoding or decoding is performed on data from the RAM for computations within the device.

3.6.1 μ-Law/A-Law Encoder

The encoder compresses sign-magnitude linear PCM (13 magnitude bits plus 1 sign bit for μ-law format or 12 magnitude bits plus 1 sign bit for A-law format) to 8-bit logarithmic PCM (7 magnitude bits plus 1 sign bit). Selection between μ-law or A-law conversion is determined by the system control register bit 14 (CR14). This bit is input directly to the encoder to determine the conversion law to be used. The μ-255 law conversion is performed if CR14 is logic 0, and A-law conversion if CR14 is logic 1. Data is input to the encoder from the data bus with an OUT instruction to port 1. The converted 8-bit log PCM sample is then presented to the multiplexer (MUX2). The multiplexer controls whether the encoder output or the eight low-order data bus bits are input to FIFO buffer 0 of the serial port. The encoder compresses data written to port 1 at all times, but the output will be enabled to the FIFO buffer only when CR12 is logic 1.

In the serial-encode mode, data written to port 1 is encoded, and the value put into FIFO buffer 0. The transmit register is then loaded with the 8-bit value on an active framing pulse, and the 8 bits are clocked out on the positive edge of SCLK.

For the parallel-encode mode, the sign-magnitude linear-PCM value is written to port 1 with an OUT instruction. The encoded 8-bit value is then stored in FIFO buffer 0. An IN instruction from port 1 reads the FIFO buffer to the data bus for storage in RAM. Care should be taken to have only one OUT and one IN instruction to port 1 for each data sample in the parallel-encode mode. With two OUTs to port 1, the first sample will be pushed into FIFO buffer 1, which cannot be read back to the data bus.

3.6.2 μ-Law/A-Law Decoder

The μ-law/A-law decoder converts 8-bit log-PCM samples to sign-magnitude linear PCM. The conversion-law selection is governed by control register bit 14 (CR14). The μ-law conversion is performed if CR14 is logic 0, and A-law conversion if CR14 is logic 1. Data input to the decoder may come from either the serial-port receive registers or FIFO buffer 0. The multiplexer (MUX1) sends data to the data bus either through the decoder or directly to the bus. This multiplexer is controlled in part by control register bit 13 (CR13). If this bit is logic 0, the multiplexer output is sent to the data bus directly. If the bit is logic 1, the multiplexer output is sent to the data bus through the decoder.

For the serial-decode mode, received data from the serial-port receive registers is input to the decoder from the multiplexer, and the received data is decoded according to either μ-law or A-law format.

For the parallel-decode mode, the 8-bit PCM sample to be decoded is written to port 1 with an OUT instruction. This stores the sample in FIFO buffer 0. The sample is then decoded by reading the value from port 1 with an IN instruction. The IN instruction brings the sample from the FIFO buffer through the multiplexer (MUX1) to the decoder, which performs the expansion on the 8-bit sample. Again, there should be only one OUT and one IN instruction to port 1 for each sample to be decoded, to avoid losing a sample in FIFO buffer 1.

4. Processor Resource Management

The TMS32011 provides many instructions for controlling internal device functions. The system control register controls the serial port and companding logic operation. There are several control register settings to utilize this hardware. An understanding of the instructions and control register settings is necessary to make efficient use of the device.

Major topics discussed in this section are listed below.

- Reset (Section 4.1 on page 4-2)
  Reset operation
  Processor initialization

- Interrupt Management (Section 4.2 on page 4-4)

- Control Register Bit Definitions (Section 4.3 on page 4-5)

- Companding Hardware Operation (Section 4.4 on page 4-7)

4.1 Reset

The reset function is enabled when an active-low level is placed on the $\overline{\text{RS}}$ pin for a minimum of five clock cycles. Reset may be applied at any time to put the TMS32011 into a known state. It is typically applied after powerup when the machine is in a random state.

4.1.1 Reset Operation

When reset is activated by applying a low level to the reset ($\overline{\text{RS}}$) input, the TMS32011 asynchronously terminates execution and forces the program counter to zero. Program memory location 0 normally contains a Branch (B) instruction to direct program execution to the system initialization routine. The reset affects the device as follows:

1) $\overline{\text{RS}}$ forces $\overline{\text{DEN}}$ and $\overline{\text{WE}}$ high and the data bus (D15-D0) to the high-impedance state.

2) Port address outputs and the program counter are asynchronously cleared to zero after the next complete CLKOUT cycle from $\overline{\text{RS}}$.

3) Normal program execution begins after one complete CLKOUT cycle from $\overline{\text{RS}}$.

4) The serial-port clock (SCLK) and the transmit outputs (DX1 and DX0) are placed in the high-impedance state.

5) The I/O port address is set to port 0.

6) Control register bit 11 (CR11) is set to logic 0, indicating that the serial-port transmit and receive registers are disabled and the parallel-companding mode is enabled.

7) Control register bit 15 (CR15) is set to logic 1, making SCLK an input to the device.

8) Interrupts are disabled. The master interrupt circuitry is disabled, along with the individual interrupts. The interrupts are disabled in the system control register, but the flags are not cleared. Control register bits CR3-CR0 should be cleared before enabling interrupts.

9) The receive register pointer is set to serial-port receive register 0. To read receive register 0, the serial-port registers must be enabled by writing a logic 1 to control register bit 11 (CR11).

The following items must be noted for effective reset operation:

1) $\overline{\text{RS}}$ must be maintained for a minimum of five CLKOUT cycles.

2) During a write cycle, $\overline{\text{RS}}$ may produce an invalid write address.

3) Due to the synchronizing action on $\overline{\text{RS}}$, the time to execute the function can vary dependent upon when $\overline{\text{RS}}$ occurs in the CLKOUT cycle.

4) The MC/$\overline{\text{PM}}$ pin must be high during reset, or $\overline{\text{BIO}}$ will be configured as an output.

5) Only control register bits CR4-CR7, CR11, and CR15 are affected by reset. All other bits are in a random state after reset, and should be configured in an initialization routine to insure the device is in a known state.

Although the transmit pins are three-stated on reset, care should be taken not to enable the serial-port registers while a framing pulse is still being generated. The framing pulses are level active, and the serial port registers continue to transmit or receive data as long as the framing pulse is active. If a reset occurs during a transmit or receive operation and the reset initialization routine execution is much shorter than the period of the framing pulse, invalid data will be transmitted or received.

4.1.2 Processor Initialization

After reset, the device must be initialized to meet the system requirements. Instructions should be executed to set up operational modes, memory pointers, interrupts, and the system control register configuration.

To configure the processor after reset, the following internal functions must be initialized:

- Interrupt structure
- Serial-port framing-pulse generation selection
- Serial-port connection
- Companding hardware
- Serial-port clock
- Auxiliary registers and auxiliary register pointer
- Data-memory page pointer
- Overflow mode.

Example 4-1 shows coding for initializing the TMS32011 serial-port and companding hardware to the following machine state:

1) Set the lower control register bit 8 (CR8) to enable port 1 to access the upper control register. To insure safe system operation, SCLK should be left as an input to the device (CR15 set to logic 0). This prevents any invalid serial-port timing during the initialization routine. The data operand is >3988.

2) The upper control register is set as follows:
   - Long FR pulse (variable data-rate selected)
   - SCLK divide ratio of 10
   - FR frequency at SCLK/256 for 8-kHz framing pulse.

   The data operand is >1C7F.

3) The lower control register is then configured as follows:
   - Interrupt flags cleared
   - Active FR interrupt enabled
   - Port transfers enabled by active FR
   - SCLK as an output.

   The data operand is >3C88.

Example 4-1. Processor Initialization

```
* A BRANCH INSTRUCTION AT PROGRAM MEMORY LOCATION 0 DIRECTS
* PROCESSOR EXECUTION HERE.  THE CONTROL REGISTER VALUES ARE
* STORED IN ROM STARTING AT LOCATION 5.  THESE VALUES ARE
* THEN READ INTO RAM FOR THE OUT INSTRUCTIONS TO THE CONTROL
* REGISTERS.  MEMORY LOCATIONS SET1-SET3 ARE LOCATED ON RAM
* PAGE 1.
*
INIT   DINT            * DISABLE INTERRUPTS
       LDPK   1        * WORK IN RAM PAGE 1
       SOVM            * SET OVERFLOW MODE
       LARP   0        * USE AUXILIARY REGISTER 0
       LACK   >5       * START AT ROM LOCATION 5
       TBLR   SET1     * READ VALUE >3988 TO RAM
       ADD    ONE,0    * INCREMENT ADDRESS
       TBLR   SET2     * READ VALUE >1C7F TO RAM
       ADD    ONE,0    * INCREMENT ADDRESS
       TBLR   SET3     * READ VALUE >3C88 TO RAM
       OUT    SET1,0   * CONFIGURE LOWER CONTROL REGISTER
       OUT    SET2,1   * CONFIGURE UPPER CONTROL REGISTER
       OUT    SET3,0   * CONFIGURE LOWER CONTROL REGISTER
       LDPK   0        * RESET RAM PAGE TO 0
*
* THE PROCESSOR IS INITIALIZED.  THE REST OF THE SYSTEM THAT IS
* APPLICATION-DEPENDENT SHOULD BE INITIALIZED BEFORE THE EINT
* INSTRUCTION.
*
       EINT            * ENABLE INTERRUPTS
```

4.2 Interrupt Management

The TMS32011 has four maskable interrupts: $\overline{\text{INT}}$, $\overline{\text{FSR}}$, $\overline{\text{FSX}}$, and FR. The interrupts are maskable via the system control register. All interrupts are synchronized and multiplexed to input the master interrupt circuitry and have the same priority. Software polling techniques are used to determine which input caused the device interrupt when multiple interrupts are enabled.

An example interrupt service routine for a system with three active interrupts enabled is given in Example 4-2.

Example 4-2. Interrupt Service Routine

```
* THIS ROUTINE MAY BE LOCATED AT PROGRAM MEMORY LOCATION 2 OR
* A BRANCH INSTRUCTION AT LOCATION 2 DIRECTS PROGRAM EXECUTION
* HERE.  MEMORY LOCATIONS A-D ARE LOCATED ON RAM PAGE 1 SINCE
* THEY ARE SELDOM-USED VALUES.  MEMORY LOCATION "ONE" CONTAINS
* THE VALUE 1.  MEMORY LOCATION RBUF IS USED TO STORE THE
* RECEIVED DATA FROM THE SERIAL PORT.
*
ISERV   SST     A           * SAVE STATUS
        LDPK    1           * WORK IN RAM PAGE 1
        SACL    B           * SAVE LOW ACCUMULATOR
        SACH    C           * SAVE HIGH ACCUMULATOR
*
* THIS ROUTINE CHECKS FOR THREE ACTIVE INTERRUPTS OCCURRING
* AND SERVICES THEM ACCORDINGLY. AFTER AN INTERRUPT FLAG IS
* SET, IT MUST BE RESET BY THE INTERRUPT SERVICE ROUTINE TO
* AVOID BEING INTERRUPTED AGAIN ON THE RETURN FROM THE
* SUBROUTINE.
*
        IN      0,D         * READ LOWER CONTROL REGISTER
        LAC     ONE,0       * INT INTERRUPT MASK
        AND     D           * INT FLAG SET?
        BNZ     XINT        * INT SERVICE ROUTINE
        LAC     ONE,2       * FSX INTERRUPT MASK
        AND     D           * FSX FLAG SET?
        BNZ     TRANS       * FSX SERVICE ROUTINE
        OUT     D,0         * RESTORE PORT 0 DATA
        IN      RBUF,1      * READ CHANNEL 0 DATA
        ZALH    C           * RESTORE HIGH ACCUMULATOR
        ADDS    B           * RESTORE LOW ACCUMULATOR
        LST     A           * LOAD STATUS
        LDPK    0           * RETURN TO RAM PAGE 0
        EINT                * ENABLE INTERRUPTS FOR RETURN
        RET                 * RETURN
```

4.3 Control Register Bit Definitions

The system control register has been mapped into I/O port 0 and port 1. To configure the register, a typical situation is to initialize constants in data RAM corresponding to the appropriate control bit values. An OUT instruction to the appropriate port (0 or 1) then writes the data memory contents to the control register. A delay time occurs from the execution of the OUT to when the control register bit values become valid and the device hardware is configured. To insure proper operation, the control register should be configured in an initialization routine that allows the bits time to become valid and places the device into a known state. Table 4-1 is a detailed description of the control register bits and their operation (see Table 1 in Appendix A, the TMS32011 Data Sheet, for a chart of the control register bits).

Table 4-1. Control Register Bit Definitions

CR BIT #	DESCRIPTION
0-3	Interrupt flags. When an interrupt occurs on any of the four maskable interrupts, the appropriate flag is set to logic 1 whether the interrupt is enabled or disabled. To clear the flag, a logic 1 is written to the appropriate bit by an OUT instruction to port 0. The bits may be read by an IN instruction to determine interrupt sources when multiple interrupts are enabled. Bit # Flag 0 $\overline{\text{INT}}$ 1 $\overline{\text{FSR}}$ 2 $\overline{\text{FSX}}$ 3 FR
4-7	Interrupt enable bits. When one of these bits is set to logic 1, an interrupt occurring on that input sets the appropriate flag and activates the microcomputer interrupt circuitry. When disabled, the interrupt flag is still set, but the device is not interrupted. Bit # Flag 4 $\overline{\text{INT}}$ 5 $\overline{\text{FSR}}$ 6 $\overline{\text{FSX}}$ 7 FR
8	Port 1 control bit. When set to logic 0, I/O port 1 is connected to either the serial-port registers or the companding hardware, depending on the state of CR11. This bit must be set with an OUT instruction to port 0 before port 1 may access the upper control register bits CR28-CR16.
9	External framing enable. This bit controls which framing pulses cause serial-port data transmission to occur. When set to logic 0, serial-port transmit and receive operations occur simultaneously and are controlled by the internal framing (FR) pulse. When set to logic 1, transmit operations are controlled by the external transmit framing pulse ($\overline{\text{FSX}}$), and receive operations are controlled by the external receive framing pulse ($\overline{\text{FSR}}$).
10	XF output latch. This bit controls the logic level of the external logic output flag (XF) pin. A write delay time occurs when reconfiguring this latch (see Appendix A for timing information).
11	Serial-port enable. When set to logic 0, the transmit and receive registers are disabled in order to use the parallel companding mode. When set to logic 1, the serial-port registers are enabled and data transfers with the serial-port are via OUT and IN instructions to port 1.
12	µ-law/A-law encoder enable. When set to logic 0, the encoder is disabled. When set to logic 1, the encoder is enabled and data written to port 1 is µ-law or A-law encoded. The encoder must be enabled for compression of linear sign-magnitude data in both the serial and parallel modes of operation.
13	µ-law/A-law decoder enable. When set to logic 0, the decoder is disabled. When set to logic 1, the decoder is enabled and data written to port 1 is µ-law or A-law decoded to sign- magnitude format. The decoder must be enabled for expansion of log PCM data in both the serial and parallel modes of operation.
14	µ-law or A-law select. When set to logic 0, the companding hardware performs µ-255-law conversion. When set to logic 1, the companding hardware performs A-law conversion.
15	Serial-clock control. When set to logic 0, the serial-port clock (SCLK) is an output, and its frequency is derived from the microcomputer system clock, X2/CLKIN. When set to logic 1, SCLK is an input to the device and controls all data transfers with the serial port.
16-23	FR frequency control. The value of these bits determines the divide ratio for the FR output frequency. The FR frequency is given as SCLK/[(CR23-CR16) + 2] The following should be noted when configuring the divide ratio: 1. All ones in these bits represents a degenerative state and shouldbe avoided. 2. The divide ratio must be greater than seven.

Table 4-1. Control Register Bit Definitions (Concluded)

CR BIT #	DESCRIPTION
24-27	SCLK prescale divide-ratio control. As an output, SCLK is derived from the microcomputer system clock, X2/CLKIN. Prescale divide ratios are selectable through these control bits (see Table 2 in Appendix A, the TMS32011 Data Sheet, for the available divide ratios).
28	FR pulse-width control. This bit controls the pulse width of the FR output to select data-transfer rates with combo-codec circuits. When set to logic 0, the FR output framing pulse is one SCLK cycle wide for the fixed data-rate mode and appears in the serial-clock cycle preceding the first serial-bit transmission. When set to logic 1, the FR output framing pulse is eight SCLK cycles wide for the variable data-rate mode. In this mode, the framing pulse is active high for the duration of the eight bits transmitted and received.

4.4 Companding Hardware Operation

Four modes are available for the companding hardware operation: serial encode, serial decode, parallel encode, and parallel decode. In the serial encode mode, transmitted data is encoded according to either μ-law or A-law format. In the serial decode mode, received data is decoded to sign-magnitude format according to the specified companding law.

In the parallel modes, either the encoder or decoder is enabled, and then data written to port 1 is compressed or expanded. To convert sign-magnitude linear PCM to 8-bit log PCM, the encoder is enabled for parallel operation, and the sample is written to port 1. An IN instruction from port 1 returns the converted 8-bit log-PCM value. To convert 8-bit log PCM to sign-magnitude linear PCM, the decoder is enabled for parallel operation, and the 8-bit sample is written to port 1. The expanded sign-magnitude value is returned on the IN instruction from port 1. Enabling both the encoder and decoder is an undesirable state and should be avoided for the parallel mode. Care should be taken to have one OUT-IN instruction sequence to port 1 for each data sample, because the execution of two OUT instructions to port 1 in succession pushes the first sample into FIFO buffer 1, preventing access for read purposes. OUT instructions to port addresses 2 through 7 do not affect the serial-port operation.

The companding hardware operates on sign-magnitude data that must be converted to two's-complement notation for computation in the microcomputer. Sign-magnitude notation consists of a sign bit in the MSB: a 0 indicating a positive value, and a 1 indicating a negative number. All bits between the sign bit and the MSB of the data value are set to 0. For conversions between μ-law and sign-magnitude linear PCM, the hexadecimal value >1FFF represents the most positive value of 8191 and the value >9FFF represents the most negative value of -8191. For conversions between A-law and sign-magnitude linear PCM, the hexadecimal value >0FFF represents the most positive value of 4095 and the value >8FFF represents the most negative value of -4095. For detailed information on μ-law and A-law companding, see the application report, "Companding Routines for the TMS32010/TMS32020," in the book, *Digital Signal Processing Applications with the TMS320 Family,* published by Texas Instruments.

Conversion between sign-magnitude and two's-complement data for μ-law encoding and decoding is implemented with the code shown in Example 4-3 and Example 4-4, respectively. The code to be used for A-law encoding and decoding is given in Example 4-5 and Example 4-6, respectively.

Example 4-3. Two's-Complement to Sign-Magnitude Conversion for μ-Law Encoding

```
* THIS ROUTINE CONVERTS A TWO'S-COMPLEMENT NUMBER TO 14-BIT
* SIGN-MAGNITUDE FORMAT AND ADDS THE BIAS OF 33 FOR MU-LAW
* ENCODING. MEMORY LOCATION 1 CONTAINS THE VALUE 1 AND
* MEMORY LOCATION 2 (BIAS) CONTAINS +33.
*
OUTPUT EQU   $
       LAC   SAMPLE     * GET THE LINEAR DATA FOR OUTPUT.
       BGEZ  POSOUT     * IF POSITIVE, THEN CHECK POS MAX VALUE;
       ABS              * IF NEGATIVE, THEN CHECK ABSOLUTE VALUE.
       ADD   BIAS       * ADD IN THE BIAS OF >21.
       ADD   ONE,15     * SET THE SIGN BIT NEGATIVE.
       SACL  SAMPLE     * HOLD FOR LATER.
       SUB   NEGMAX,7   * COMPARE TO NEGATIVE MAX = >9FFF.
       BLEZ  DONE       * IF WITHIN MAX, THEN SEND IT.
       LAC   NEGMAX,7   * ELSE LOAD THE VALUE WITH THE LARGEST
       SACL  SAMPLE     * NEGATIVE IN RANGE
       B     DONE       * AND SEND IT.
POSOUT ADD   BIAS       * ADD IN THE BIAS OF >21.
       SACL  SAMPLE     * AND SAVE IT.
       SUB   POSMAX,7   * COMPARE TO POSITIVE MAX = >1FFF.
       BLEZ  DONE       * IF WITHIN MAX, THEN SEND IT.
       LAC   POSMAX,7   * ELSE LOAD THE VALUE WITH THE LARGEST
       SACL  SAMPLE     * POSITIVE VALUE IN RANGE
DONE   OUT   SAMPLE,PA1 * AND SEND IT TO ENCODER.
* ...  CONTINUE CODE HERE.
```

Example 4-4. Sign-Magnitude to Two's-Complement Conversion for μ-Law Decoding

```
* THIS ROUTINE CONVERTS A 14-BIT SIGN-MAGNITUDE NUMBER TO
* TWO'S-COMPLEMENT NOTATION AND REMOVES THE BIAS OF 33 FOR
* MU-LAW DECODING.  MEMORY LOCATION 1 CONTAINS THE VALUE 1
* AND MEMORY LOCATION 2 (BIAS) CONTAINS 33.
*
INPUT  EQU   $
       IN    SAMPLE,PA1 * READ INPUT FROM SERIAL PORT AND DECODE.
       LAC   SAMPLE     * MOVE INPUT TO ACCUMULATOR.
       SUB   BIAS       * REMOVE BIAS VALUE.
       BGEZ  POS        * IF POSITIVE, THEN SAVE IT.
       ADD   ONE,15     * ELSE DELETE SIGN BIT BY CARRY.
       SACL  SAMPLE     * SAVE MAGNITUDE VALUE.
       ZAC              * NEGATE THE INPUT BY
       SUB   SAMPLE     * SUBTRACTING FROM ZERO AND SAVE
POS    SACL  SAMPLE     * FULLY EXPANDED LINEAR DATA.
* ...  CONTINUE CODE HERE.
```

Example 4-5. Two's-Complement to Sign-Magnitude Conversion for A-Law Encoding

```
* THIS ROUTINE CONVERTS A TWO'S-COMPLEMENT NUMBER TO 13-BIT
* SIGN-MAGNITUDE FORMAT FOR A-LAW ENCODING. MEMORY LOCATION 1
* CONTAINS THE VALUE 1.
*
OUTPUT EQU    $
       LAC    SAMPLE      * GET THE LINEAR DATA FOR OUTPUT.
       BGEZ   POSOUT      * IF POSITIVE, THEN CHECK POS MAX VALUE;
       ABS                * IF NEGATIVE, THEN CHECK ABSOLUTE VALUE.
       ADD    ONE,15      * SET THE SIGN BIT NEGATIVE.
       SACL   SAMPLE      * HOLD FOR LATER.
       SUB    NEGMAX,7    * COMPARE TO NEGATIVE MAX = >8FFF.
       BLEZ   DONE        * IF WITHIN MAX, THEN SEND IT.
       LAC    NEGMAX,7    * ELSE LOAD THE VALUE WITH THE LARGEST
       SACL   SAMPLE      * NEGATIVE IN RANGE
       B      DONE        * AND SEND IT.
POSOUT SACL   SAMPLE      * SAVE IT.
       SUB    POSMAX,7    * COMPARE TO POSITIVE MAX = >0FFF.
       BLEZ   DONE        * IF WITHIN MAX, THEN SEND IT.
       LAC    POSMAX,7    * ELSE LOAD THE VALUE WITH THE LARGEST
       SACL   SAMPLE      * POSITIVE VALUE IN RANGE
DONE   OUT    SAMPLE,PA1 * AND SEND IT TO ENCODER.
* ...  CONTINUE CODE HERE.
```

Example 4-6. Sign-Magnitude to Two's-Complement Conversion for A-Law Decoding

```
* THIS ROUTINE CONVERTS A 13-BIT SIGN-MAGNITUDE NUMBER TO
* TWO'S-COMPLEMENT NOTATION FOR A-LAW DECODING.  MEMORY
* LOCATION 1 CONTAINS THE VALUE 1.
*
INPUT  EQU    $
       IN     SAMPLE,PA1 * READ INPUT FROM SERIAL PORT AND DECODE.
       LAC    SAMPLE      * MOVE INPUT TO ACCUMULATOR.
       BGEZ   POS         * IF POSITIVE, THEN SAVE IT.
       ADD    ONE,15      * ELSE DELETE SIGN BIT BY CARRY.
       SACL   SAMPLE      * SAVE MAGNITUDE VALUE.
       ZAC                * NEGATE THE INPUT BY
       SUB    SAMPLE      * SUBTRACTING FROM ZERO AND SAVE
POS    SACL   SAMPLE      * FULLY EXPANDED LINEAR DATA.
* ...  CONTINUE CODE HERE.
```

To further illustrate the serial and parallel modes of the companding hardware operation, Table 4-2 shows the control register bit combinations that determine the serial or parallel modes of operation.

Note that the serial and parallel companding modes require separate control register settings. When using the serial mode, parallel companding is not available unless the control register is reconfigured.

Table 4-2. Serial- and Parallel-Mode Bit Configurations

CR BIT #			MODE OF OPERATION
13	12	11	
0	0	0	Parallel mode. Encoder and decoder are disabled. No operation performed on data written to or read from port 1.
0	0	1	Serial mode. Encoder and decoder are disabled. The transmit registers are enabled for data transmission on an active framing pulse. The 8-bit value written to port 1 is transmitted and the 8-bit value in the receive register is read with an IN instruction from port 1.
0	1	0	Parallel encode. Encoder is enabled. A sign-magnitude sample written to port 1 with an OUT instruction is compressed to 8-bit log PCM. The 8-bit value is then read from port 1 with an IN instruction.
0	1	1	Serial encode. Encoder is enabled. A sign-magnitude sample written to port 1 is compressed to 8-bit log PCM and put into the transmit register for transmission on an active framing pulse.
1	0	0	Parallel decode. Decoder is enabled. An 8-bit log PCM data written to port 1 is decoded to sign-magnitude notation with an IN instruction from port 1.
1	0	1	Serial decode. Decoder is enabled. An 8-bit log PCM sample from one of the receive registers is expanded to sign-magnitude notation with an IN instruction from port 1.
1	1	0	Parallel encode and decode. Encoder and decoder enabled. This is not a usual state, since data is compressed on an OUT instruction to port 1 and then expanded with the IN instruction from the port.
1	1	1	Serial encode and decode. Encoder and decoder enabled. Sign-magnitude data written to port 1 is encoded and put into one of the transmit registers for serial transmission. The 8-bit log PCM data from one of the receive registers is decoded with an IN instruction from port 1.

5. Codec Interface

A sample interface utilizing the serial port of the TMS32011 is described in this section. The TMS32011 is capable of direct interface to combo-codec circuits, reducing chip count and improving system throughput. The TMS32011 can also interface a PCM highway system through the combo-codec, with the timing and framing pulses received from the highway. In the PCM highway interface, the combo-codec provides the external framing pulses to the TMS32011, and the serial-port clock is provided from the PCM highway.

The TMS32011 is interfaced to a TCM2913 combo-codec to demonstrate the direct serial-port interface capability. A standalone full-duplex serial interface is shown in Figure 5-1, in which the TMS32011 provides the serial clock for bit transmission. The codec is sampled every 125 us (8-kHz frequency), at which time an 8-bit PCM byte is exchanged between the two devices.

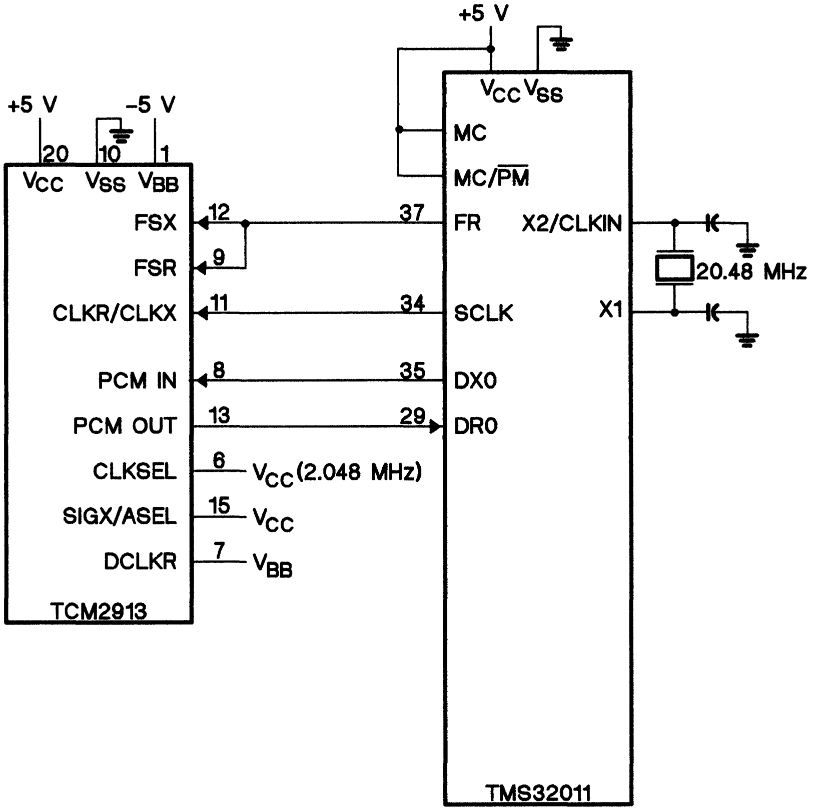


Figure 5-1. Codec Interface for Standalone Serial Operation

5.1 Timing Considerations

Timing for the serial interface system is controlled by the TMS32011 serial-port clock (SCLK). SCLK is configured as an output from the TMS32011, and its frequency is set to 2.048 MHz (see Section 4.4). A 20.48-MHz crystal is input to the TMS32011 as its system clock. The SCLK frequency is derived from this system clock by a divide-by-10 in the SCLK prescale control logic. SCLK is connected to CLKR/CLKX on the TCM2913 to provide the transmit and receive master clock. CLKSEL on the codec is tied to V_{CC} to select the 2.048-MHz master clock mode.

5.2 Framing Pulse Generation

Framing pulses are generated by the TMS32011 on the FR output pin. The frequency of these pulses is set to 8 kHz by dividing the serial clock (SCLK) by 256. The short FR framing pulses provide the codec with framing pulses for the fixed data-rate mode. FR is input to both the $\overline{\text{FSX}}$ and $\overline{\text{FSR}}$ inputs on the codec. The FR output causes simultaneous transmit and receive operations from the TMS32011 serial port. The $\overline{\text{FSX}}$ input on the codec causes the device to transmit PCM data on the next eight consecutive positive transitions of the serial-port clock (SCLK). The $\overline{\text{FSR}}$ input on the codec causes the device to receive PCM data on the next eight consecutive negative transitions of the serial-port clock (SCLK). With this timing, the codec transmits and receives one 8-bit PCM sample every 125 us.

5.3 Transmit and Receive Operation

The TMS32011 transmits its PCM sample via the DX0 pin. The sample is received by the TCM2913 on the PCM IN pin. The TMS32011 receives PCM samples on its DR0 pin, which is the output of the PCM OUT pin of the TCM2913. With this setup, single-channel operation is realized with the TMS32011. All data transmission occurs on channel 0, requiring one IN instruction from port 1 to receive the PCM sample and one OUT instruction to port 1 to send a sample to the codec.

5.4 Companding Law Selection

In the serial interface configuration, μ-255 law companding is selected. The TMS32011 companding hardware selects μ-law companding by setting system control register bit 14 (CR14) to logic 0. The TCM2913 is put into the μ-law companding mode by connecting the SIGX/ASEL pin to V_{CC}.

6. Peripheral Mode Interface

The TMS32011 features a peripheral mode for the development and testing of ROM code in applications development. In this peripheral mode, the device functions as a serial-port peripheral to a standard TMS32010 digital signal processor. Only the serial port, interrupt synchronizer, system control register, and the μ-law/A-law hardware are functional. All other logic on the device is unused. This mode allows realtime emulation capability for the development of TMS32011 applications.

When the MC/$\overline{\text{PM}}$ pin is tied to V_{SS}, the TMS32011 is in the peripheral mode. In this mode, the TMS32011 is a serial port to a standard TMS32010 microprocessor. The following signals differ in their function and use when in the peripheral mode:

1) $\overline{\text{BIO}}$ becomes an output from the TMS32011, signalling the TMS32010 that an interrupt has occured on the serial port and that the port needs to be serviced. The master interrupt signal is enabled onto the $\overline{\text{BIO}}$ pin. This signal is input to the $\overline{\text{INT}}$ interrupt on the TMS32010.

2) CLKOUT on the TMS32011 becomes an input for the synchronization of interrupts and serial-port framing control. This insures synchronous operation between the two processors. The CLKOUT pins are connected between the two devices.

3) $\overline{\text{WE}}$ becomes an input indicating that valid data from the TMS32010 is on the data bus. $\overline{\text{WE}}$ indicates that the TMS32010 is performing an OUT instruction and that the TMS32011 should accept the data into one of its I/O ports. Configuring the control register or sending data to the serial port requires an OUT instruction to port 0 or port 1.

4) $\overline{\text{DEN}}$ becomes an input to the TMS32011, indicating that the TMS32010 is accepting data on the data bus during an IN instruction. The TMS32010 receives serial port data from the TMS32011 via IN instructions from port 1. The TMS32010 outputs $\overline{\text{DEN}}$ active low to signal the TMS32011 that data should be output from either port 0 or port 1.

5) PA2-PA0 become inputs to the TMS32011 to signal which I/O port is being accessed by the TMS32010 during OUT and IN instructions. These pins control the serial port, companding hardware, and the system control register.

6) CLKIN on the TMS32011 must be synchronized with CLKIN on the TMS32010.

7) $\overline{\text{RS}}$ should be connected as an input to both the TMS32011 and the TMS32010.

In the peripheral mode, only the serial port and interrupt sychronization logic of the TMS32011 is enabled. During OUT and IN instructions with the TMS32010, the TMS32011 will transfer data on the data bus (D15-D0) on active $\overline{\text{DEN}}$ or $\overline{\text{WE}}$ from the TMS32010. When the TMS32010 executes an OUT instruction, $\overline{\text{WE}}$ goes active (low) to signal that data is being output onto the data bus. $\overline{\text{WE}}$ is an input to the TMS32011, signalling that it should accept data from the data bus on one of its I/O ports. When the TMS32010 executes an IN instruction, it outputs $\overline{\text{DEN}}$ low to signal that data is being accepted from the data bus via one of its ports. The $\overline{\text{DEN}}$ pin on the TMS32011 is an input used to signal that the device should output data onto the data bus from one of the port addresses.

The TMS32011 data bus is in the high-impedance state, except when port 0 or port 1 is being accessed. This prevents bus conflicts if the TMS32010 is accessing I/O port addresses above 1. The TMS32011 in the peripheral mode acts as a serial port peripheral connected to port 0 and port 1 of the TMS32010.

In the peripheral mode, all timings are identical to the normal microcomputer mode. Therefore, the TMS32011 in the peripheral mode connected to a TMS32010 provides realtime in-circuit emulation for the development of TMS32011 applications.

The TMS32011 prototyping system, shown in Figure 6-1, uses the TMS32011 in the peripheral mode. The TMS32010 and a pair of TBP28S166 2K-byte PROMs are used as the processor and ROM in the development of TMS32011 applications. The serial port pins (DR0, DR1, DX0, DX1, XF, SCLK, FR, $\overline{\text{FSX}}$, and $\overline{\text{FSR}}$) can be connected to a codec as the application requires (see Section 5).

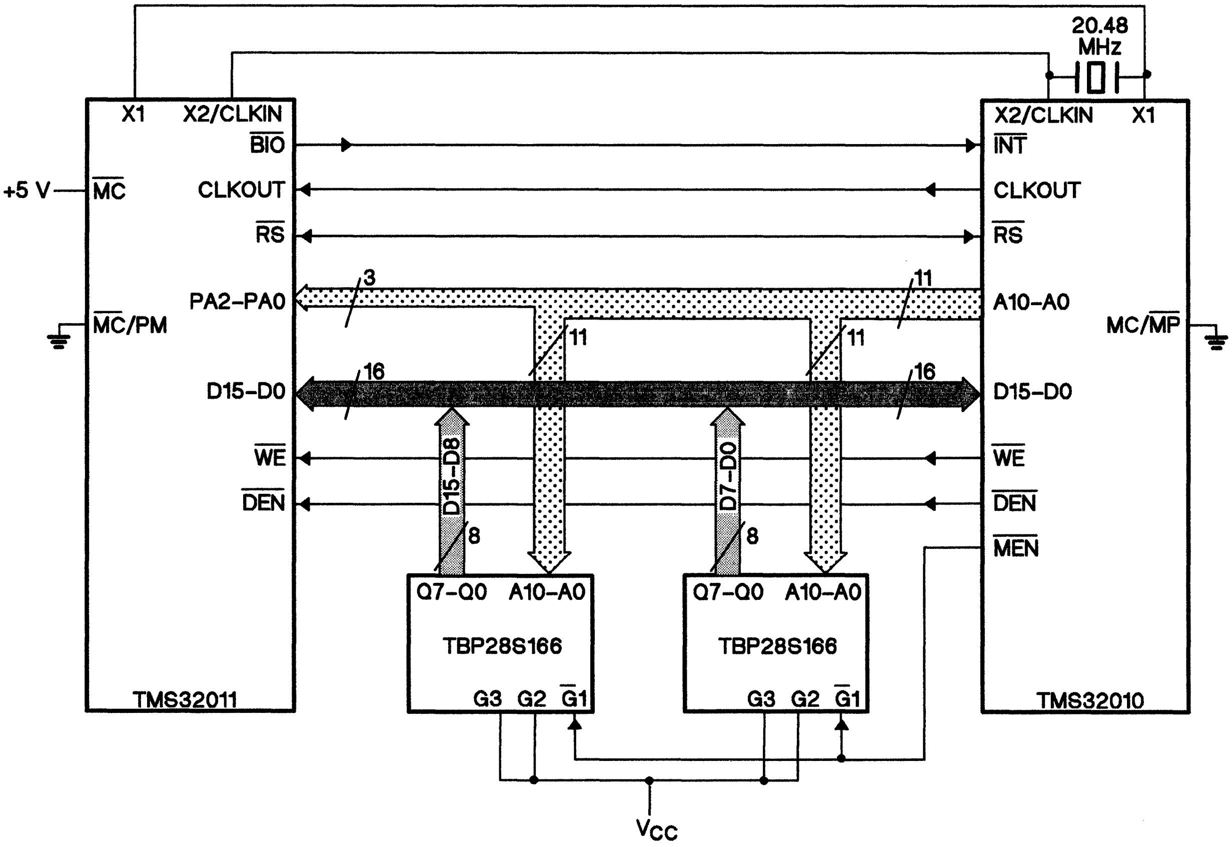


Figure 6-1. TMS32011 Standalone Prototyping System

Figure 6-2 shows an interface of the TMS32011 in the peripheral mode to the TMS32010 Emulator (XDS/22) and Evaluation Module (EVM) via the TMS32010 target connector. This interface provides increased flexibility for development of applications with the TMS32011. An enhanced XDS/22, which will have a TMS32011 target connector, will be available in the second half of 1986.

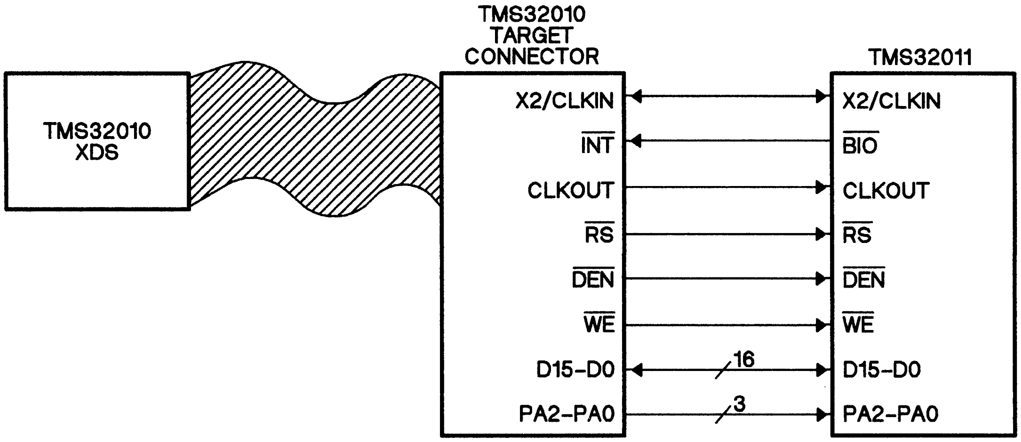
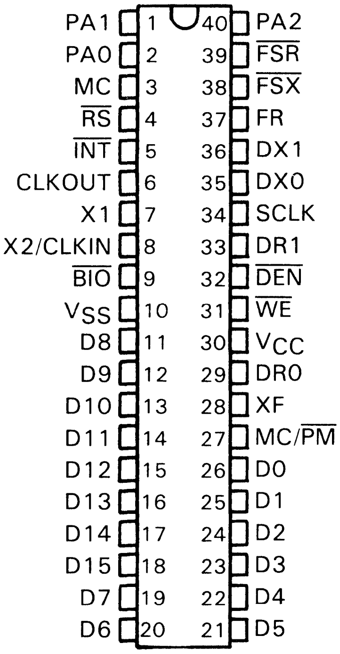


Figure 6-2. XDS/TMS32011 Peripheral Mode Interface

FEBRUARY 1986

- Object-Code Compatible with TMS32010 Instruction Set and Development Tools
- Dual-Channel Serial Port with Direct Interface to Combo-Codec
- On-Chip μ-Law/A-Law Companding Hardware
- 200-ns Instruction Cycle
- 144-Word On-Chip Data RAM
- 1536-Word On-Chip Program ROM
- 16-Bit Instruction/Data Word
- 32-Bit ALU/Accumulator
- 16 x 16-Bit Multiply in 200 ns
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels (Six of Each User-Definable)
- 16-Bit Bidirectional Data Bus with 40-Megabits-per-Second Transfer Rate
- Interrupt with Full Context Save
- Signed Two's-Complement Fixed-Point Arithmetic
- 2.4-Micron NMOS Technology
- Single 5-V Supply

N PACKAGE
(TOP VIEW)



description

The TMS32011 is a dedicated microcomputer designed to support a wide range of Digital Signal Processing (DSP) applications. The TMS32011 is a member of the TMS32010 architectural generation of digital signal processors. Execution capability of five million instructions per second, coupled with the 16 × 16 parallel multiplier and single-cycle multiply instructions, make the TMS320 family ideally suited for many numeric-intensive applications, such as telecommunications, modems, and speech processing.

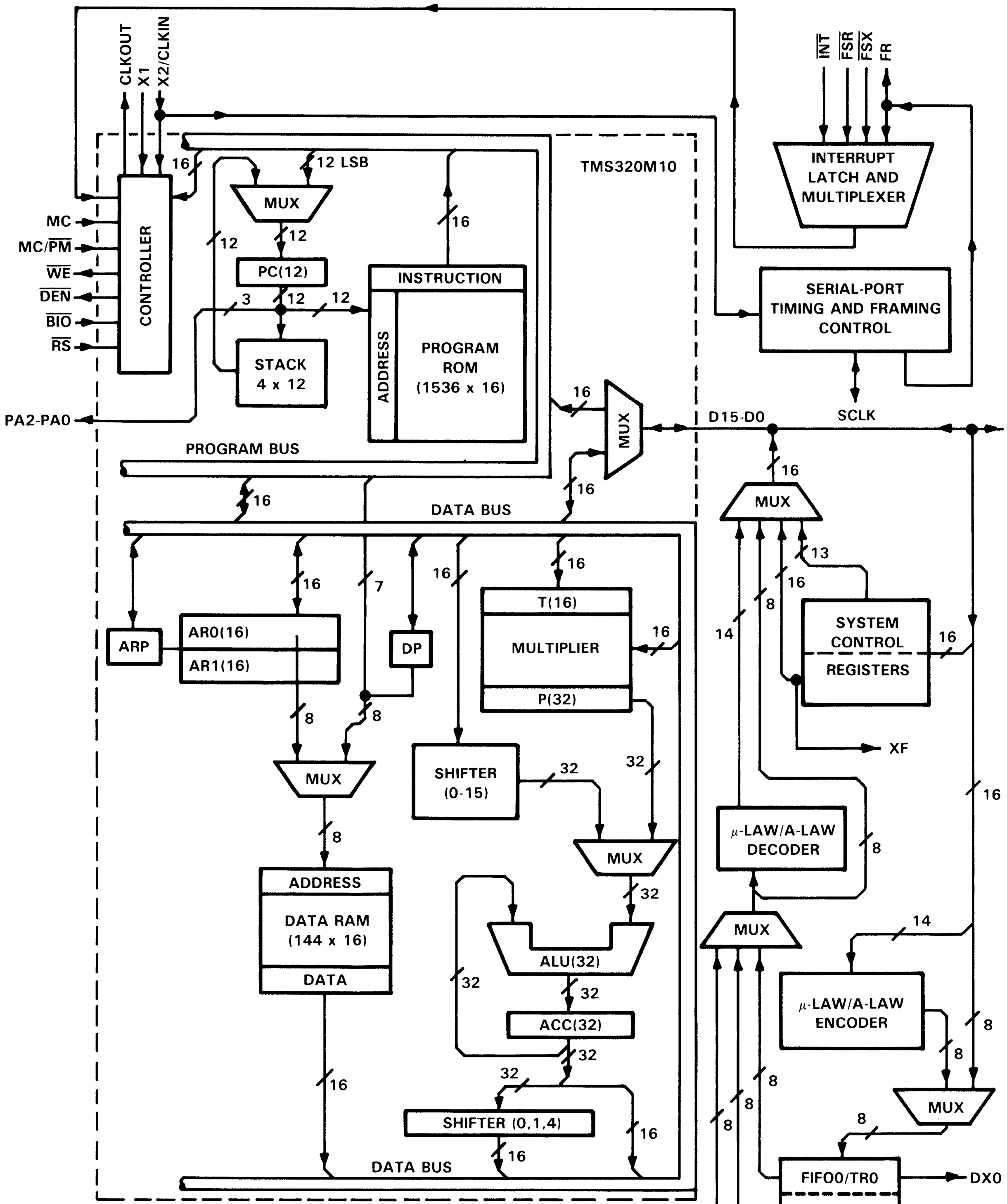
The TMS32011 is identical to a TMS320M10 microcomputer without the external memory expansion capability. The TMS32011 instruction set is fully compatible with the TMS32010, enabling all existing TMS32010 development tools to be used in TMS32011 application development. The device features a dual-channel serial port with two transmit and two receive registers for full-duplex communication. The serial port provides a direct interface to combo-codec circuits via three selectable framing pulses and timing logic to support external PCM highway communications and standalone serial applications. On-chip hardware enables the TMS32011 to compand (COMpress/exPAND) data in either μ-law or A-law conversion format, thus providing increased throughput and reduced ROM space requirements in PCM applications. The TMS32011 offers a cost-effective solution in many DSP applications by reducing chip count for designs that require a serial interface.

TEXAS
INSTRUMENTS
POST OFFICE BOX 1443 ● HOUSTON, TEXAS 77001

PIN NOMENCLATURE

NAME	I/O	DEFINITION	NAME	I/O	DEFINITION
PA2-PA0	O	I/O port address outputs	DX1, DX0	O	Serial-port transmit-channel outputs
D15-D0	I/O	16-bit parallel data bus	DR1, DR0	I	Serial-port receive-channel inputs
$\overline{\text{DEN}}$	O	Data enable for device input data on D15-D0	XF	O	External-flag output pin
$\overline{\text{WE}}$	O	Write enable for device output data on D15-D0	MC	I	Microcomputer select input, tied to V_{CC}.
$\overline{\text{RS}}$	I	Reset for initializing the device	MC/$\overline{\text{PM}}$	I	Microcomputer or peripheral mode select input.
$\overline{\text{BIO}}$	I	External polling input for branch instructions			
$\overline{\text{INT}}$	I	External interrupt input	CLKOUT	O	System clock output, ¼ crystal/ CLKIN frequency
$\overline{\text{FSR}}$	I	External serial-port receive framing input	X1	O	Crystal output for internal oscillator
$\overline{\text{FSX}}$	I	External serial-port transmit framing input	X2/CLKIN	I	Crystal input for internal oscillator or external system clock input
FR	O	Internal serial-port framing output	V_{CC}	I	+5-V Supply
SCLK	I/O	Serial-port clock	V_{SS}	I	Ground

functional block diagram



LEGEND:

ACC	= ACCUMULATOR	PC	= PROGRAM COUNTER
ARP	= AUXILIARY REGISTER POINTER	P	= P REGISTER
AR0	= AUXILIARY REGISTER 0	T	= T REGISTER
AR1	= AUXILIARY REGISTER 1	TR	= TRANSMIT REGISTER
DP	= DATA PAGE POINTER	RR	= RECEIVE REGISTER

architecture

The TMS32011 consists of four major functional units: the TMS320M10 microcomputer, a system control register, a full-duplex dual-channel serial port, and μ-law/A-law companding hardware.

microcomputer

The TMS32011 features TMS320M10 microcomputer architecture and performance. The TMS320 family utilizes a modified Harvard architecture, which permits transfers between program and data memory for increased flexibility and performance.

The TMS32011 utilizes hardware to implement functions that other processors typically perform in software. This hardware includes a hardware multiplier to perform a 16 × 16-bit multiply in a single 200-ns instruction cycle. A hardware barrel shifter is used to shift data going into the ALU. Two auxiliary registers, which provide indirect data RAM addresses, can be configured in an autoincrement/decrement mode for single-cycle manipulation of data tables.

The TMS32011 instruction set is fully compatible with the TMS32010 instruction set. The Table Write (TBLW) instruction should not be used on the TMS32011 or a potential bus conflict may occur.

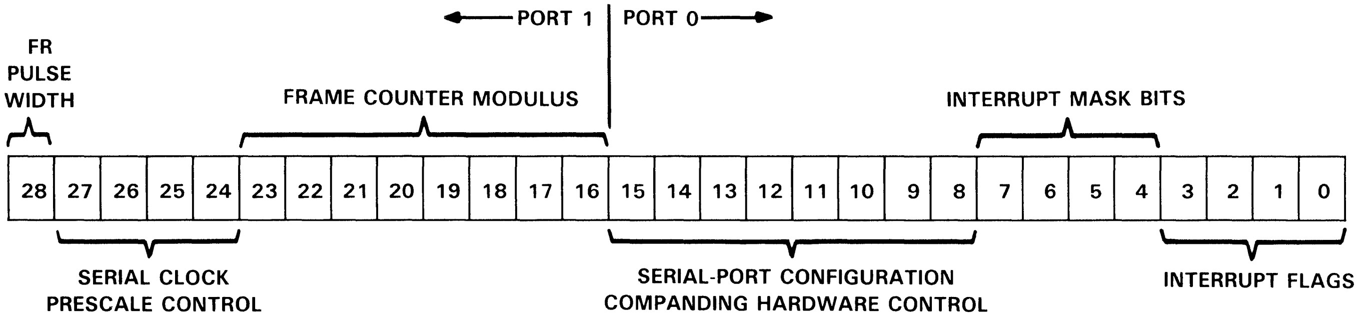
Two of the I/O ports are dedicated to the serial port and companding hardware. I/O port 0 is dedicated to the lower system control register, which controls the serial port, interrupts, and companding hardware. I/O port 1 accesses the upper control register, as well as both serial port channels and the companding hardware. Six I/O ports are available for external interface.

Interrupts have been expanded on the TMS32011 to fully support the serial port interface. Four maskable interrupts ($\overline{\text{INT}}$; FR, $\overline{\text{FSX}}$, and $\overline{\text{FSR}}$) are mapped into I/O port 0 via the lower system control register. When disabled, these interrupts may be used as single-bit logic inputs polled by software.

system control register

Operation of the serial port and companding hardware is determined by the 29 bits of the system control register (see Table 1 for the control register bit definitions). The control register is mapped into I/O port 0 and port 1. The lower 16 register bits CR15-CR0 are accessed through port 0. Port 1 accesses the upper 13 control bits CR28-CR16, as well as both serial port channels and the companding hardware. Communication with the control register is via OUT and IN instructions to port 0 and port 1.

TABLE 1. CONTROL REGISTER CONFIGURATION



BIT	DESCRIPTION AND CONFIGURATION
0	$\overline{\text{INT}}$ interrupt flag (Note 1).
1	$\overline{\text{FSR}}$ interrupt flag (Note 1).
2	$\overline{\text{FSX}}$ interrupt flag (Note 1).
3	FR interrupt flag (Note 1)
4	$\overline{\text{INT}}$ interrupt enable mask. When set to logic 1, an interrupt on $\overline{\text{INT}}$ activates device interrupt circuitry.
5	$\overline{\text{FSR}}$ interrupt enable mask. Same as $\overline{\text{INT}}$ control.
6	$\overline{\text{FSX}}$ interrupt enable mask. Same as $\overline{\text{INT}}$ control.
7	FR interrupt enable mask. Same as $\overline{\text{INT}}$ control.
8	Port 1 configuration control: 0 = port 1 connected to either serial-port registers or companding hardware. 1 = port 1 accesses CR28-CR16.
9	External framing enable: 0 = serial-port data transfers controlled by active FR. 1 = serial-port data transfers controlled by active $\overline{\text{FSX}}/\overline{\text{FSR}}$.
10	XF external logic output flag latch
11	Serial-port enable: 0 = parallel companding mode; serial port disabled. 1 = serial companding mode; serial port registers enabled.
12	μ-law/A-law encoder enable: 0 = disabled. 1 = data written to port 1 is μ-law or A-law encoded.
13	μ-law/A-law decoder enable: 0 = disabled. 1 = data written to port 1 is μ-law or A-law decoded.
14	μ-law or A-law encode/decode select: 0 = companding hardware performs μ-law conversion 1 = companding hardware performs A-law conversion
15	Serial clock control: 0 = SCLK is an output, derived from the prescaler in timing logic. 1 = SCLK is an input that provides the clock for serial port and frame counter in timing logic.
23-16	Frame counter modulus. Controls FR frequency = SCLK/CNT + 2 where CNT is binary value of CR23-CR16 (Notes 2, 3, 4)
27-24	SCLK prescale control bits. (See Table 2 for divide ratios.)
28	FR pulse-width control: 0 = Fixed-data rate; FR is 1 SCLK cycle wide. 1 = Variable-data rate; FR is 8 SCLK cycles wide.

NOTES: 1. Interrupt flag is cleared by writing a logic 1 to the bit with an OUT to port 0.
2. All ones in CR23-CR16 indicate a degenerative state and should be avoided.
3. Bits are operational whether SCLK is an input or an output.
4. The FR divide ratio must be greater than 7.

Configuration of the serial port and companding hardware is via port 0. These 16 bits control the interrupts, serial port connections, and companding hardware operation. Port 1 accesses the 13 bits of the upper control register (CR28-CR16) when the lower control register bit 8 (CR8) is logic 0. When bit 8 is logic 1, port 1 communicates with the serial port and companding hardware.

serial port

The dual-channel serial port is capable of full-duplex serial communication and offers direct interface to combo-codecs. Two receive registers and two transmit registers are mapped into I/O port 1. These registers operate with 8-bit data samples. Either internal (FR) or external ($\overline{\text{FSR}}$ and $\overline{\text{FSX}}$) framing signals for serial data transfers are selected through the system control register. The serial port clock, SCLK, provides the bit timing for transfers with the serial port, and may be either an input or output. As an input, an external clock provides the timing for data transfers and framing pulse synchronization. As an output, SCLK provides the timing for standalone serial communication and is derived from the TMS32011 system clock, X2/CLKIN and system control register bits CR27-CR24. See Table 2 for the available divide ratios. The internal (FR) framing pulse frequency is derived from the serial port clock (SCLK) and system control register bits CR23-CR16. This framing pulse signal provides framing pulses for combo-codec circuits, an 8-kHz sample clock for voice-band systems, or a timer for control applications.

TABLE 2. SERIAL CLOCK (SCLK) DIVIDE RATIOS
X2/CLKIN = 20.48 MHz

CR27	CR26	CR25	CR24	DIVIDE RATIO	SCLK FREQUENCY	UNIT
0	0	0	0	32	0.640	MHz
0	0	0	1	28	0.731	MHz
0	0	1	X	24	$0.85\overline{3}$	MHz
0	1	X	X	20	1.024	MHz
1	0	0	0	16	1.280	MHz
1	0	0	1	14	1.463	MHz
1	0	1	X	12	$1.70\overline{6}$	MHz
1	1	X	X	10	2.048	MHz

μ-law/A-law companding hardware

On-chip hardware enables the TMS32011 to compand data in either μ-law or A-law format. The companding logic operation is configured via the system control register. Data may be companded in either a serial mode for operation on serial port data or a parallel mode for computation inside the device. μ-255-law or A-law companding is selectable through the system control register bit 14. Note that for μ-law encoding, the bias of 33 must be added to the magnitude before encoding; likewise, after μ-law decoding, the bias of 33 must be subtracted from the magnitude. For A-law companding, no bias is required.

In the serial mode, sign-magnitude linear PCM (13 magnitude bits plus 1 sign bit for μ-law format or 12 magnitude bits plus 1 sign bit for A-law format) is compressed to 8-bit sign-magnitude logarithmic PCM by the encoder and sent to the transmit register for transmission on an active framing pulse. The decoder converts 8-bit sign-magnitude log PCM from the serial port receive registers to linear PCM.

In the parallel mode, the serial port registers are disabled to allow parallel data from internal memory to be encoded or decoded for computation inside the device. In the parallel encode mode, the encoder is enabled and a 14-bit sign-magnitude value written to port 1. The encoded value is returned with an IN instruction from port 1. In the parallel decode mode, the decoder is enabled and an 8-bit sign-magnitude log PCM value written to port 1. On the successive IN instruction from port 1, the decoded value is returned. For further information on companding techniques, see the Texas Instruments application report "Companding Routines for the TMS32010."

instruction set

The TMS32011's comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. The instruction set, explained in Tables 3 and 4, consists primarily of single-cycle single-word instructions, permitting execution rates of five million instructions per second. Only infrequently used branch and I/O instructions are multicycle.

The TMS32011 also contains a number of instructions that shift data as part of an arithmetic operation. These instructions all execute in a single cycle and are useful for scaling data in parallel with other operations.

Three main addressing modes are available with the TMS32011 instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words, and the second page contains 16 words. In a typical application, infrequently accessed variables, such as those used for servicing an interrupt, are stored on the second page. The instruction format for direct addressing is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								0	dma						

Bit 7 = 0 defines the direct addressing mode. The opcode is contained in bits 15 through 8. Bits 6 through 0 contain the data memory address (dma). The seven bits of dma can directly address up to 128 words (1 page) of data memory. Use of the data memory page pointer is required to address the full 144 words of data memory.

Direct addressing can be used with all instructions requiring data operands except for the immediate operand instruction.

indirect addressing

Indirect addressing forms the data memory address from the least-significant eight bits of one of the two auxiliary registers, AR0 and AR1. The Auxiliary Register Pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. The instruction format for indirect addressing is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								1	0	INC	DEC	ARP	0	0	ARP

Bit 7 = 1 defines the indirect addressing mode. The opcode is contained in bits 15 through 8. Bits 7 through 0 contain indirect addressing control bits. Bits 3 and 0 control the ARP. If bit 3 = 0, the content of bit 0 is loaded into the ARP; if bit 3 = 1, the content of ARP remains unchanged. ARP = 0 defines the contents of AR0 as memory address; ARP = 1 defines the contents of AR1 as memory address.

Bits 5 and 4 control the auxiliary registers. If bit 5 = 1, the ARP defines which auxiliary register is to be incremented by 1. If bit 4 = 1, the ARP defines which auxiliary register is to be decremented by 1. If bits 5 and 4 are zero, then neither auxiliary register is incremented or decremented. Bits 6, 2, and 1 are reserved and should always be programmed to zero.

Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

The TMS32011 instruction set contains special ''immediate'' instructions. These instructions derive data from part of the instruction word rather than from the data RAM. Some useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

instruction set summary

Table 3 lists the symbols and abbreviations used in Table 4, the instruction set summary. Table 4 contains a short description along with the opcode for each TMS32011 instruction. The summary is arranged according to function and alphabetized within each functional group.

TABLE 3. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
AR	Auxiliary register (AR0 and AR1 are predefined assembler symbols equal to 0 and 1, respectively.)
ARP	Auxiliary register pointer
D	Data memory address field
DATn	Label assigned to data memory location n
dma	Data memory address
DP	Data page pointer
I	Addressing mode bit
INTM	Interrupt mode flag bit
K	Immediate operand field
>nn	Indicates nn is a hexadecimal number. All others are assumed to be decimal values.
OVM	Overflow (saturation) mode flag bit
P	Product (P) register
PA	Port address (PA0 through PA7 are predefined assembler symbols equal to 0 through 7, respectively).
PC	Program counter
pma	Program memory address
PRGn	Label assigned to program memory location n
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
T	T register
TOS	Top of stack
X	3-bit accumulator left-shift field
→	Is assigned to
| |	Indicates an absolute value.
< >	Items within angle brackets are defined by user.
[ ]	Items within brackets are optional.
( )	Indicates ''contents of''.
{ }	Items within braces are alternative items; one of them must be entered.
< >	Angle brackets back-to-back indicate ''not equal''.
	Blanks or spaces are significant.

TABLE 4. TMS32011 INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	◄—S—►				I	◄—D—►						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	◄—D—►						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	◄—D—►						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	◄—D—►						
LAC	Load accumulator with shift	1	1	0	0	1	0	◄—S—►				I	◄—D—►						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	◄—K—►							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	◄—D—►						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	◄X►			I	◄—D—►						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	◄—D—►						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	◄—S—►				I	◄—D—►						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	◄—D—►						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	◄—D—►						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	◄—D—►						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	◄—D—►						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	◄—D—►						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	◄—D—►						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	◄—D—►						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	◄—K—►							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	◄—D—►						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	◄—D—►						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	◄—D—►						

TABLE 4. TMS32011 INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BIOZ	Branch on $\overline{\text{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	← D →						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	← D →						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	← D →						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	← D →						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	← K →												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 4. TMS32011 INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◄——D——►						
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◄——D——►						

I/O AND DATA MEMORY OPERATIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next location	1	1	0	1	1	0	1	0	0	1	I	◄——D——►						
IN	Input data from port	2	1	0	1	0	0	0	◄PA►			I	◄——D——►						
OUT	Output data to port	2	1	0	1	0	0	1	◄PA►			I	◄——D——►						
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◄——D——►						

development systems and software support

Texas Instruments offers concentrated development support and complete documentation for designing a TMS32011-based microcomputer system. When developing an application, tools are provided to evaluate the performance of the processor, to develop the algorithm implementation, and to fully integrate the design's software and hardware modules. When questions arise, additional support can be obtained by contacting the Texas Instruments Customer Response Center (CRC) hotline number, 1-800-232-3200.

Sophisticated TMS32011 development operations are performed with the existing TMS32010 Emulator (XDS), Evaluation Module (EVM), Macro Assembler/Linker, and Simulator. Using the unique features of the TMS32011 in the peripheral mode, along with the emulation capability of the TMS32010 XDS and EVM, TMS32011 applications can be developed. In the second half of 1986, Texas Instruments will provide a full in-circuit emulator for the TMS32011. No changes in software support are necessary for the TMS32011, enabling all existing TMS32010 development tools to be used in TMS32011 development operations.

A complete list of TMS32011 software and hardware development tools is given in Table 5.

TABLE 5. TMS32011 SOFTWARE AND HARDWARE SUPPORT

HOST COMPUTER	OPERATING SYSTEM	PART NUMBER
TMS32010 MACRO ASSEMBLERS/LINKERS		
DEC VAX	VMS	TMDS3240210-08
TI/IBM PC	MS/PC-DOS	TMDS3240810-02
TMS32010 SIMULATORS		
DEC VAX	VMS	TMDS3240211-08
TI/IBM PC	MS/PC-DOS	TMDS3240811-02
TMS32010 DIGITAL FILTER DESIGN PACKAGE (DFDP)		
TI PC	MS-DOS	DFDP-TI001
IBM PC	PC-DOS	DFDP-IBM001
TMS32010 HARDWARE		
Evaluation Module (EVM)		RTC/EVM320A-03
Analog Interface Board (AIB)		RTC/EVM320C-06
Emulator:		
XDS/22		TMDS3262210
Enhanced XDS/22 (available the second half of 1986)		TMDS3262211

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

Supply voltage, V_{CC}‡	−0.3 V to 7 V
All input voltages	−0.3 V to 15 V
Output voltage	−0.3 V to 15 V
Continuous power dissipation	1.5 W
Air temperature range above operating device	0°C to 70°C
Storage temperature range	−55°C to +150°C

†Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	2.8			
V_{IL}	Low-level input voltage (all inputs)				0.8	V
I_{OH}	High-level output current (all outputs)				300	μA
I_{OL}	Low-level output current (all outputs)				2	mA
T_A	Operating free-air temperature		0		70	°C

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 5.25 V			20	μA
				V_O = 0 V			−20	
I_I	Input current		V_I = V_{SS} to V_{CC}				±50	μA
I_{CC}‡	Supply current		V_{CC} = MAX	T_A = 0°C		180	275	mA
				T_A = 70°C			235§	mA
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25		pF
		All others				15		
C_o	Output capacitance	Data bus				25		pF
		All others				10		

†All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.
‡I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.
§Value derived from characterization data and is not tested.

CLOCK CHARACTERISTICS AND TIMING

The TMS32011 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	MIN	NOM	MAX	UNIT
Crystal frequency f_x	0°C – 70°C	6.7		20.5	MHz
C1, C2	0°C – 70°C		10		pF

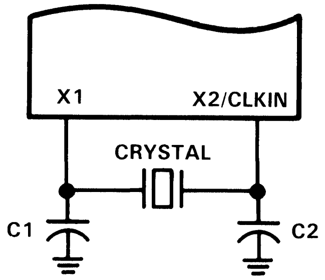


FIGURE 1. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{c(C)}$	CLKOUT cycle time[†]	R_L = 825 Ω, C_L = 100 pF, See Figure 2	195.12			ns
$t_{r(C)}$	CLKOUT rise time			10		ns
$t_{f(C)}$	CLKOUT fall time			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓[‡]		25		60	ns

[†] $t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).

[‡] Values given were derived from characterization data and are not tested.

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{c(MC)}$	Master clock cycle time	48.78		150	ns
$t_{r(MC)}$	Rise time master clock input		5	10	ns
$t_{f(MC)}$	Fall time master clock input		5	10	ns
$t_{w(MCP)}$	Pulse duration master clock	$0.475t_{c(C)}$		$0.525t_{c(C)}$	ns
$t_{w(MCL)}$	Pulse duration master clock low, $t_{c(MC)}$ = 50 ns		20		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)}$ = 50 ns		20		ns

PARAMETER MEASUREMENT INFORMATION

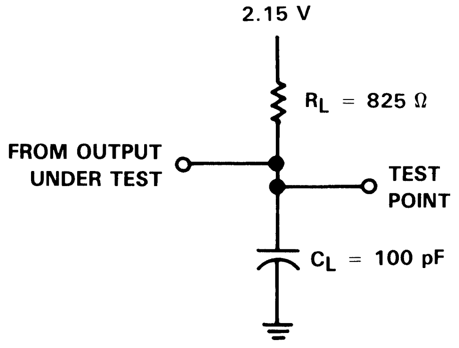
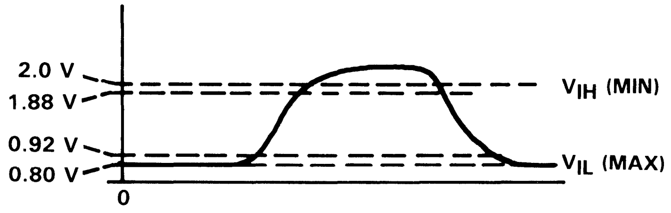
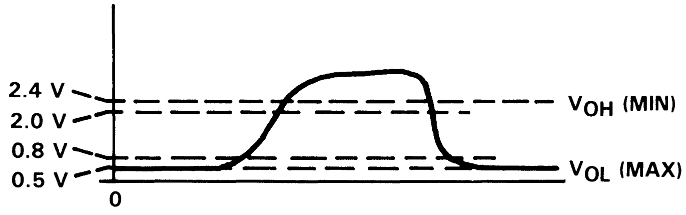


FIGURE 2. TEST LOAD CIRCUIT



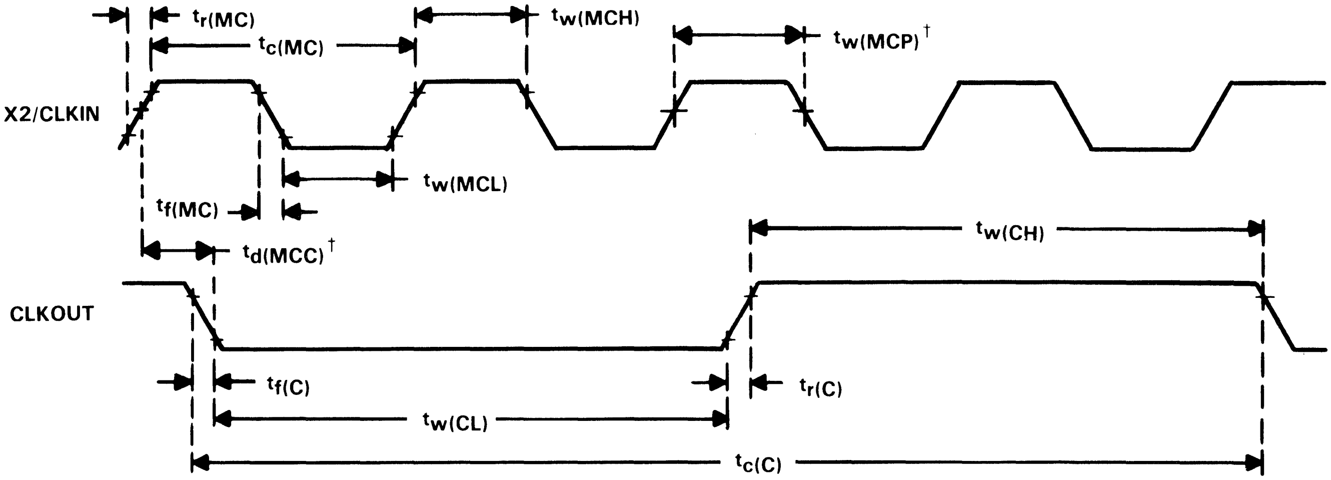
(a) INPUT



(b) OUTPUTS

FIGURE 3. VOLTAGE REFERENCE LEVELS

clock timing



NOTE 5: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
[†]$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid (see Note 6)	R_L = 825 Ω, C_L = 100 pF, See Figure 2	10[†]		50	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\mathrm{DEN}}$↓		$\frac{1}{4}t_{c(C)}-5$[†]		$\frac{1}{4}t_{c(C)}+15$	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\mathrm{DEN}}$↑		-10[†]		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\mathrm{WE}}$↓		$\frac{1}{2}t_{c(C)}-5$[†]		$\frac{1}{2}t_{c(C)}+15$	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\mathrm{WE}}$↑		-10[†]		15	ns
t_{d6}	Delay time CLKOUT↓ to data bus OUT valid				$\frac{1}{4}t_{c(C)}+65$	ns
t_{d7}	Time after CLKOUT↓ that data bus starts to be driven		$\frac{1}{4}t_{c(C)}-5$[†]			ns
t_{d8}	Time after CLKOUT↓ that data bus stops being driven				$\frac{1}{4}t_{c(C)}+30$[†]	ns
t_v	Data bus OUT valid after CLKOUT↓		$\frac{1}{4}t_{c(C)}-10$			ns

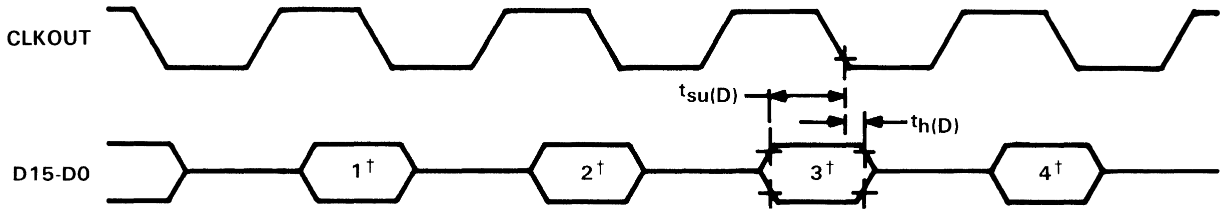
NOTE 6: Address bus will be valid upon $\overline{\mathrm{WE}}$↑, $\overline{\mathrm{DEN}}$↑, or $\overline{\mathrm{MEN}}$↑.
[†]These values were derived from characterization data and are not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2	50			ns
$t_{su(A-D)}$	Address bus setup time prior to $\overline{\mathrm{DEN}}$↓		5			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓		0			ns

NOTE 7: Data may be removed from the data bus upon $\overline{\mathrm{DEN}}$↑ preceding CLKOUT↓.

TBLR instruction timing



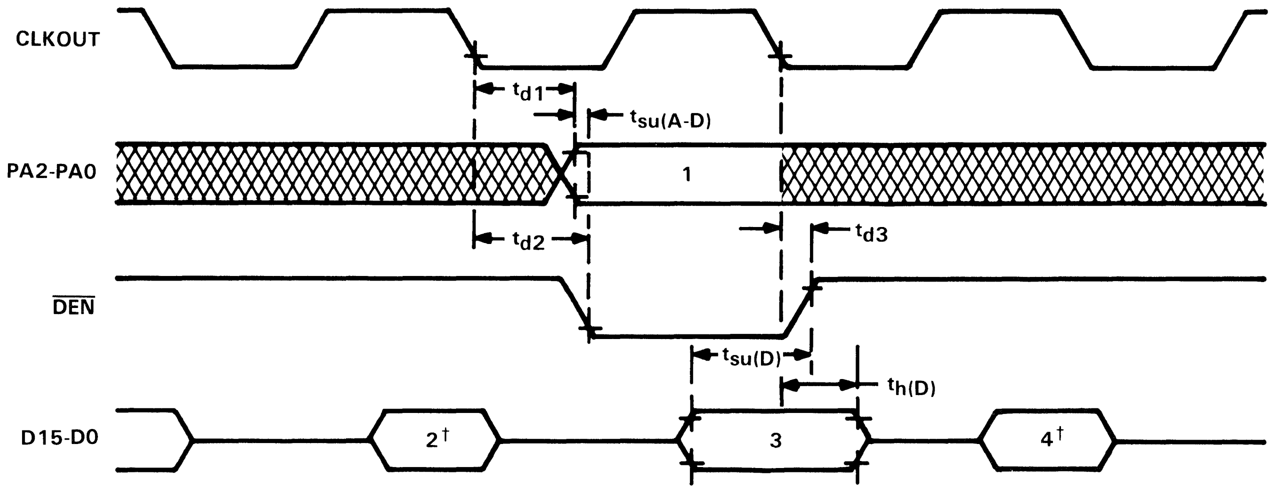
LEGEND:

1. TBLR OPCODE VALID	3. DATA IN VALID
2. DUMMY OPCODE	4. NEXT INSTRUCTION OPCODE VALID

[†]Data appears only on the internal data bus and is shown for clarification.

NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

IN instruction timing



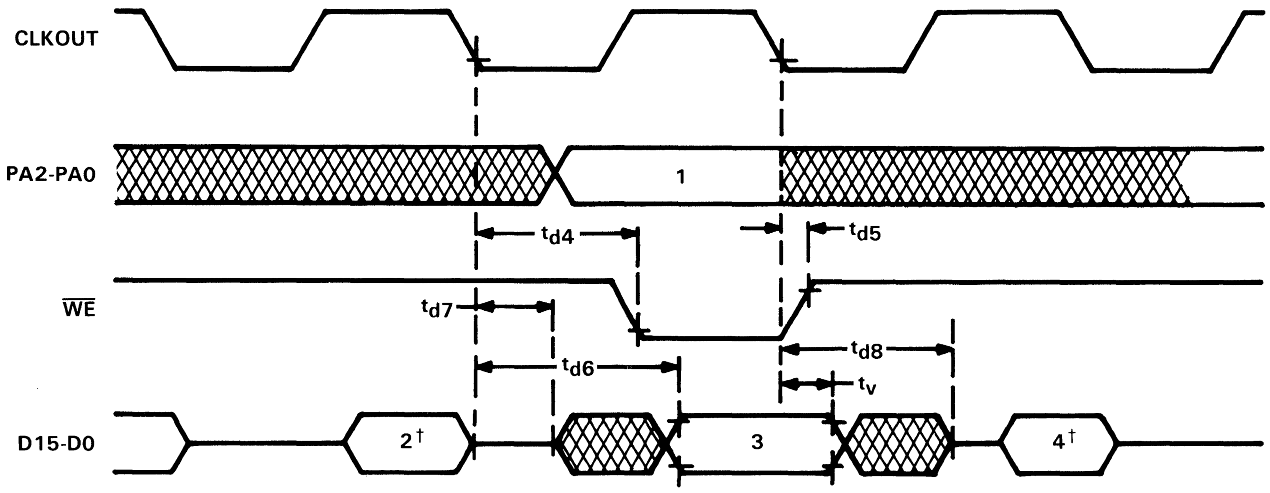
LEGEND:

1. PORT ADDRESS VALID	3. PORT DATA VALID
2. IN OPCODE VALID	4. NEXT INSTRUCTION OPCODE VALID

[†]Data appears only on the internal data bus and is shown for clarification.

NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

OUT instruction timing



LEGEND:

1. PORT ADDRESS VALID
2. IN OPCODE VALID
3. PORT DATA VALID
4. NEXT INSTRUCTION OPCODE VALID

†Data appers only on the internal data bus and is shown for clarification.

NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

$\overline{RS}$, $\overline{INT}$, $\overline{BIO}$, and XF TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d9}	Delay time $\overline{DEN}$↑ and $\overline{WE}$↑ from $\overline{RS}$↓	R_L = 825 Ω, C_L = 100 pF, See Figure 2			$\frac{1}{2}t_{c(C)}+50$†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{RS}$↓				$\frac{1}{4}t_{c(C)}+50$†	ns
t_{d10}	Delay time SCLK three-state from $\overline{RS}$↓		100†		200†	ns
t_{d11}	Delay time, DX1, DX0 three-state from $\overline{RS}$↓		100†		200†	ns
$t_{d(XF)}$	Delay time CLKOUT↓ to valid XF		5†		115	ns
$t_{d(BIO)}$	Delay time CLKOUT↓ to $\overline{BIO}$↓ (in the peripheral mode)		$0.5t_{c(C)}$†		$0.5t_{c(C)}+50$	ns

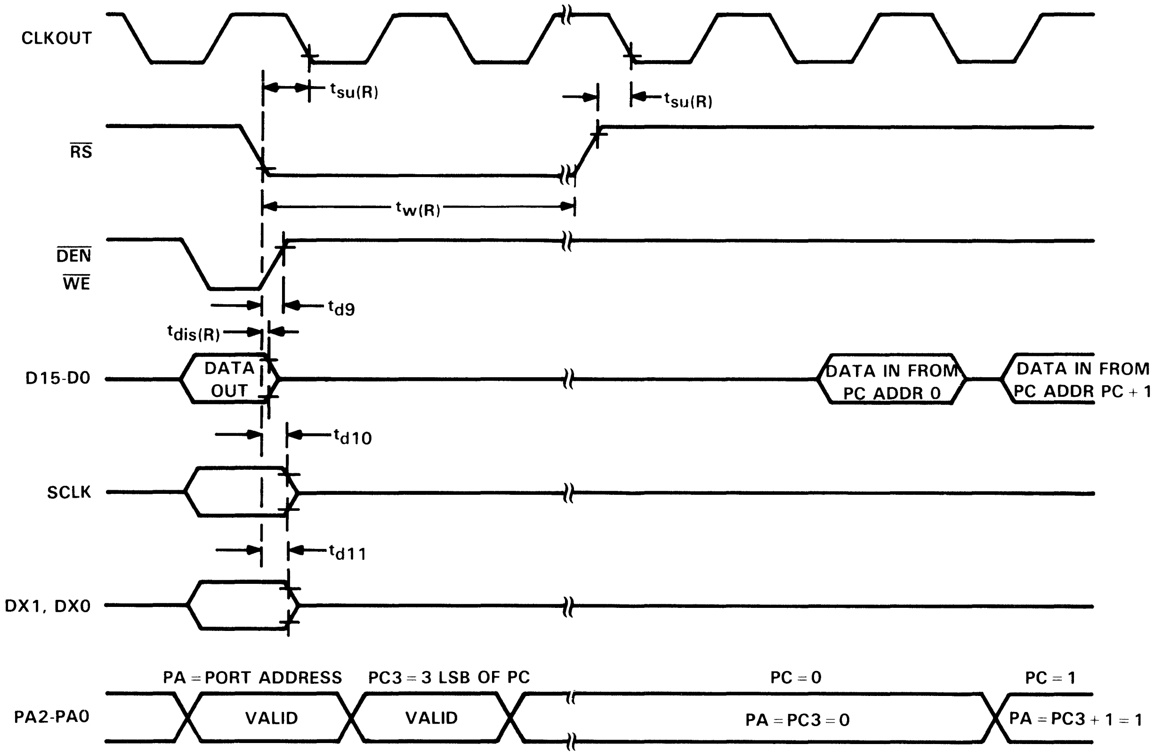
†These values were derived from characterization data and are not tested.

timing requirements over recommended operating conditions

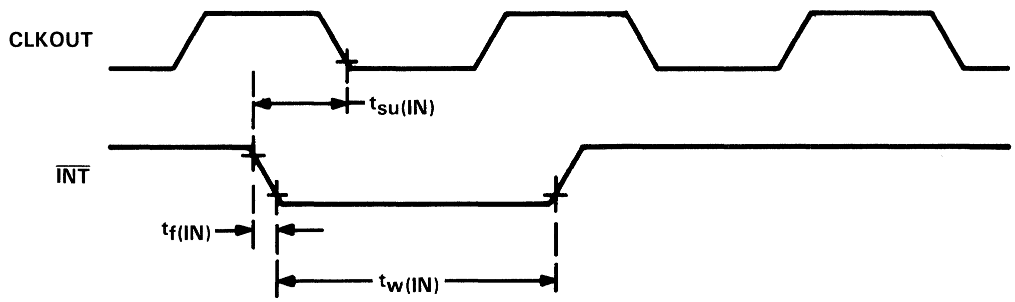
		MIN	NOM	MAX	UNIT
$t_{su(R)}$	Reset ($\overline{RS}$) setup time prior to CLKOUT. (See Note 9)	50			ns
$t_{w(R)}$	$\overline{RS}$ pulse duration	$5t_{c(C)}$			ns
$t_{f(IN)}$	$\overline{INT}/\overline{BIO}$ fall time			15	ns
$t_{w(IN)}$	$\overline{INT}/\overline{BIO}$ pulse duration	$t_{c(C)}$			ns
$t_{su(IN)}$	$\overline{INT}/\overline{BIO}$ setup time before CLKOUT↓	50			ns

NOTE 9: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

reset timing

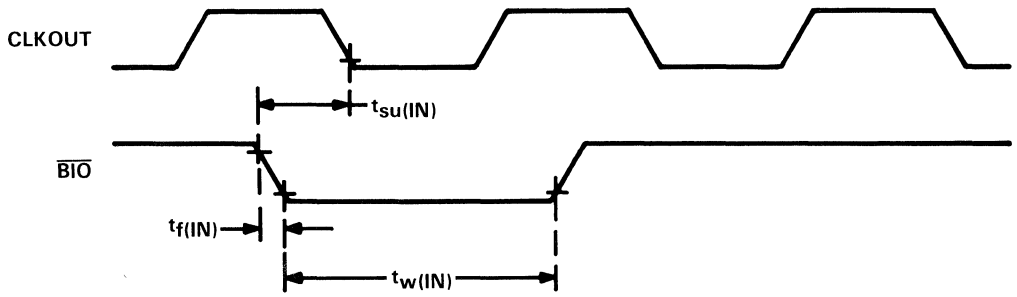


interupt timing



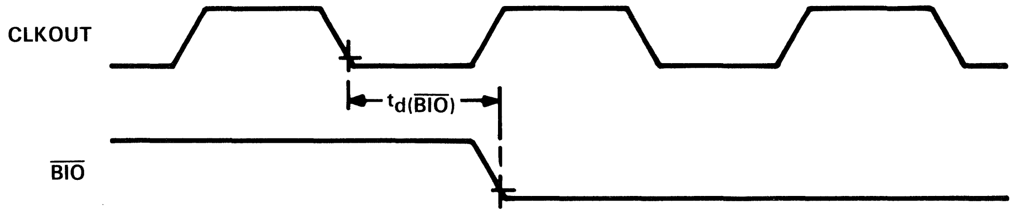
NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

$\overline{\text{BIO}}$ timing (microcomputer mode only)



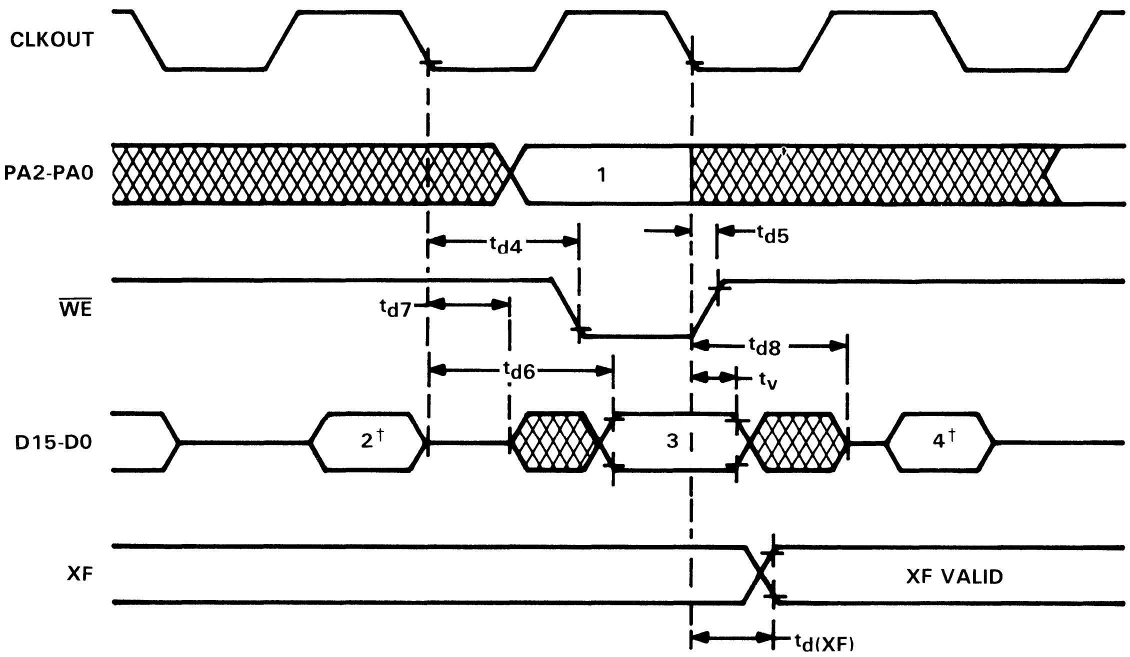
NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

$\overline{\text{BIO}}$ timing (peripheral mode only)



NOTE 8: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

XF timing



LEGEND:

1. PORT ADDRESS VALID	3. PORT DATA VALID
2. OUT OPCODE VALID	4. NEXT INSTRUCTION OPCODE VALID

†Data appears only on the internal data bus and is shown for clarification.

SERIAL PORT TIMING

switching characteristics over recommended operating conditions

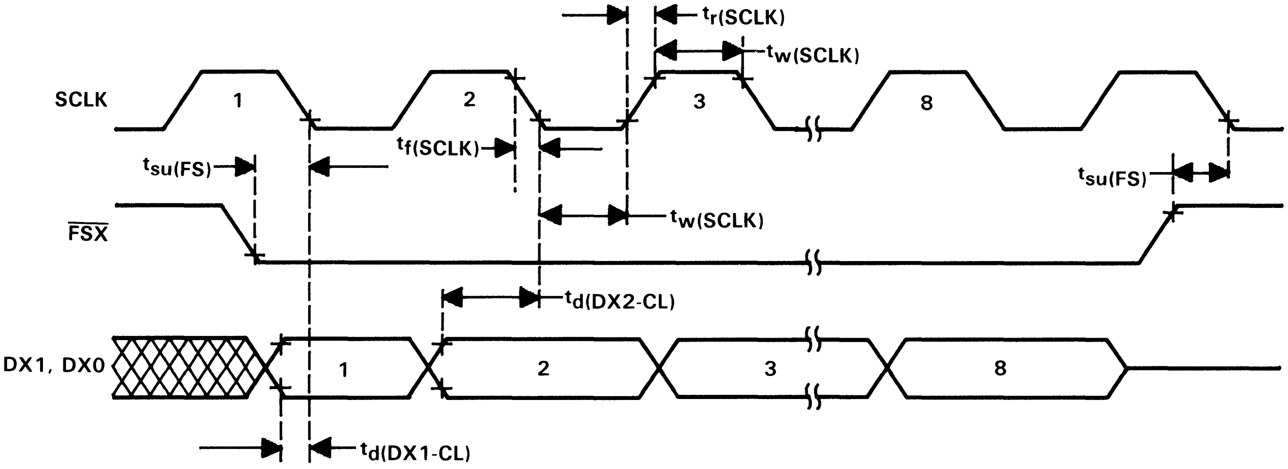
PARAMETER		MIN	TYP	MAX	UNIT
$t_{d(CH\text{-}FR)}$	Internal framing (FR) delay from SCLK rising edge	20		115	ns
$t_{d(DX1\text{-}CL)}$	DX bit 1 valid before SCLK falling edge	20			ns
$t_{d(DX2\text{-}CL)}$	DX bit 2 valid before SCLK falling edge	20			ns

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{c(SCLK)}$	Serial port clock (SCLK) cycle time (see Note 10)	390		4770	ns
$t_{f(SCLK)}$	Serial port clock (SCLK) fall time			50	ns
$t_{r(SCLK)}$	Serial port clock (SCLK) rise time			50	ns
$t_{w(SCLK)}$	Serial Port Clock (SCLK) low-pulse duration (see Note 11)	185		2500	ns
$t_{w(SCLK)}$	Serial Port Clock (SCLK) high-pulse duration (see Note 11)	185		2500	
$t_{su(FS)}$	$\overline{FSX}/\overline{FSR}$ setup time before SCLK falling edge	100			ns
$t_{su(DR)}$	DR setup time before SCLK falling edge	20			ns
$t_{h(DR)}$	DR hold time after SCLK falling edge	20			ns

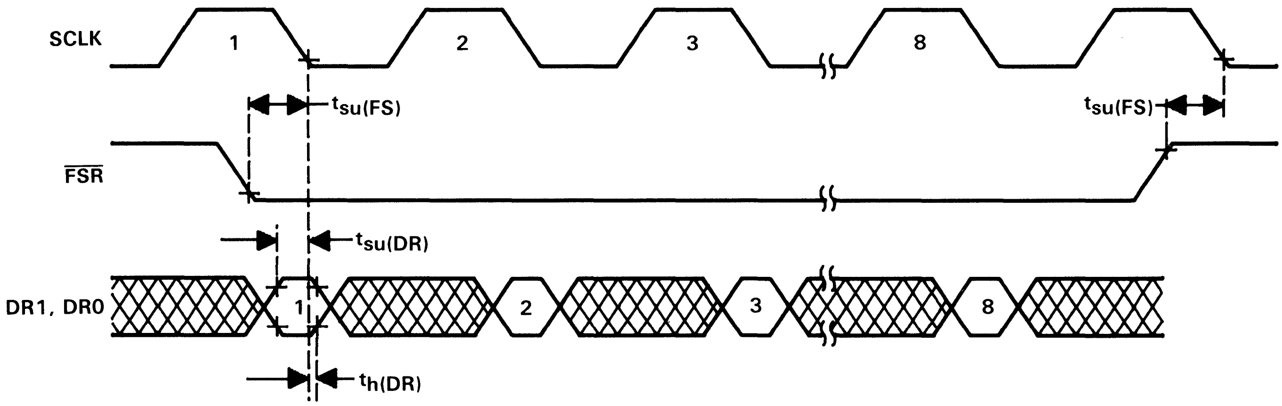
NOTES: 10. Minimum cycle time is $2t_{c(C)}$ where $t_{c(C)}$ is CLKOUT cycle time.
11. The duty cycle of the serial port clock must be within 45 to 55 percent.

external framing: transmit timing

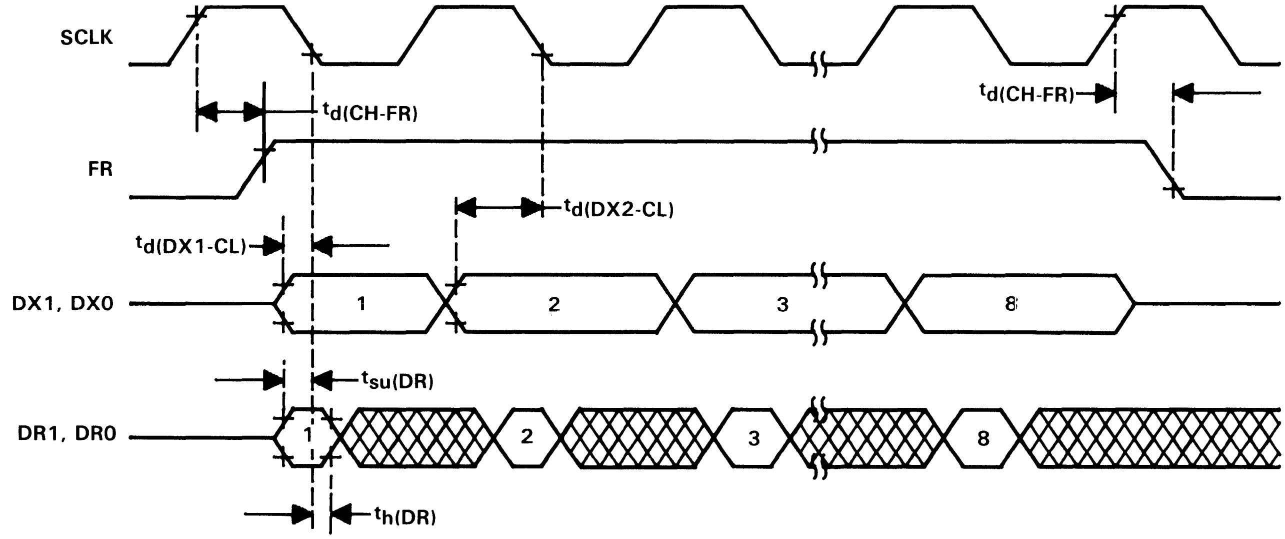


NOTE 12: Data valid on transmit outputs until SCLK rises.

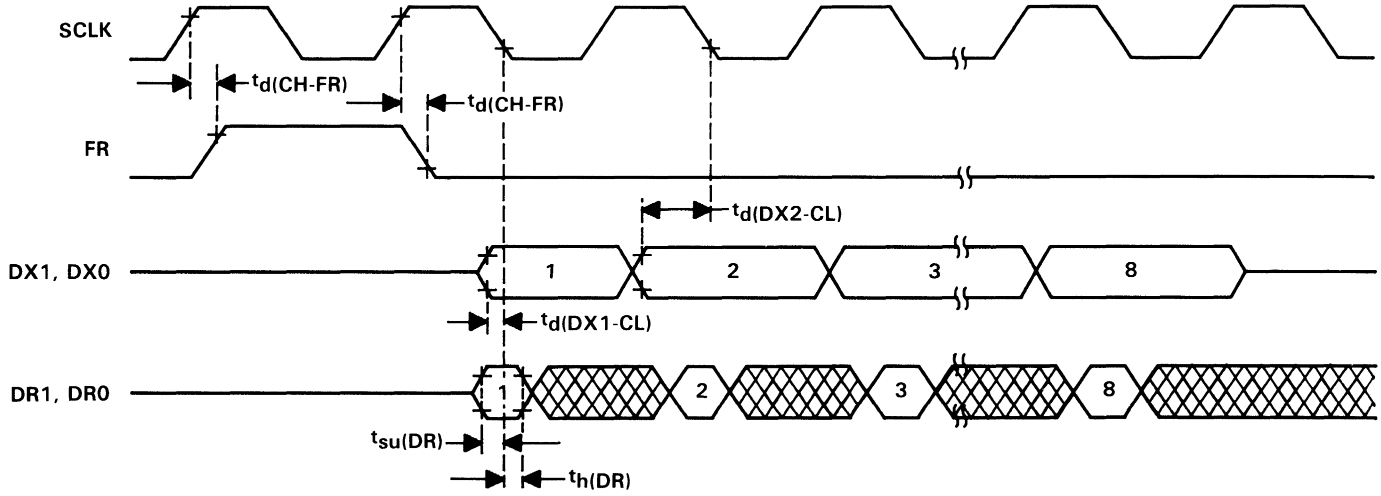
external framing: receive timing



internal framing: variable-data rate

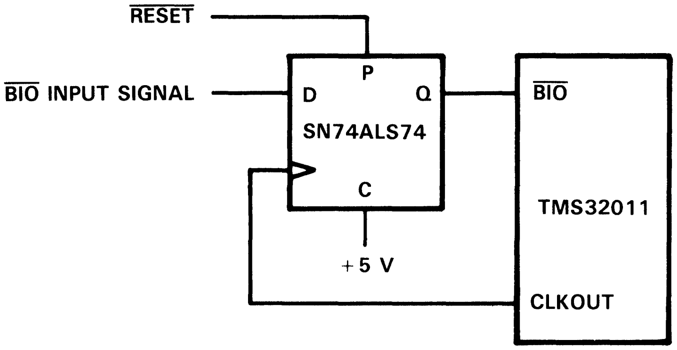


internal framing: fixed-data rate



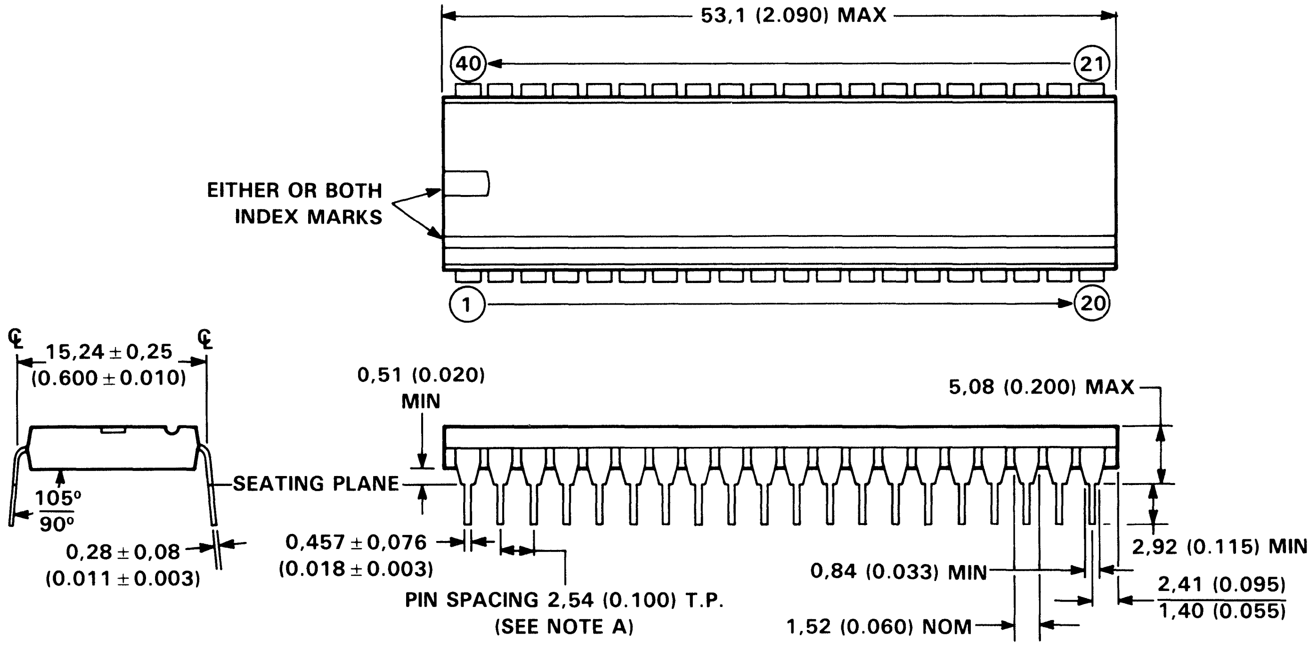
$\overline{\text{BIO}}$ synchronization requirements

For systems using asynchronous inputs to the $\overline{\text{BIO}}$ pin on the TMS32011, the external hardware shown in the diagram is recommended to ensure proper execution of the BIOZ instruction. This hardware synchronizes the $\overline{\text{BIO}}$ input signal with the rising edge of CLKOUT on the TMS32011. The pulse width required for this input signal is $t_{c(C)}$, which is one TMS32011 clock cycle, plus sufficient setup time for the flip-flop (dependent upon the flip-flop used). Note that the $\overline{\text{INT}}$ input is totally asynchronous on the TMS32011 and does not require the flip-flop to insure proper interrupt execution as on the TMS32010.


$\overline{\text{RESET}}$
$\overline{\text{BIO}}$ INPUT SIGNAL
P
D
Q
SN74ALS74
C
+5 V
$\overline{\text{BIO}}$
TMS32011
CLKOUT

MECHANICAL DATA

40-pin plastic dual-in-line package



ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

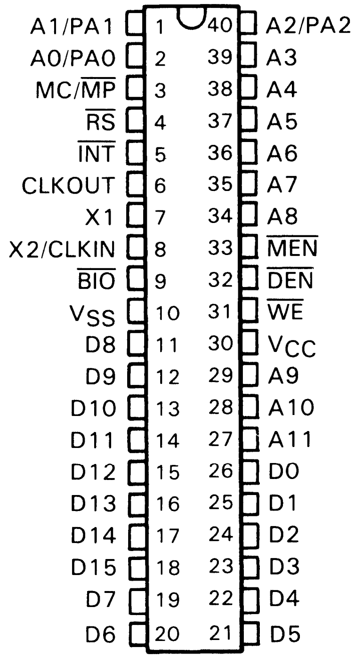
NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.

TMS32010
DIGITAL SIGNAL PROCESSOR

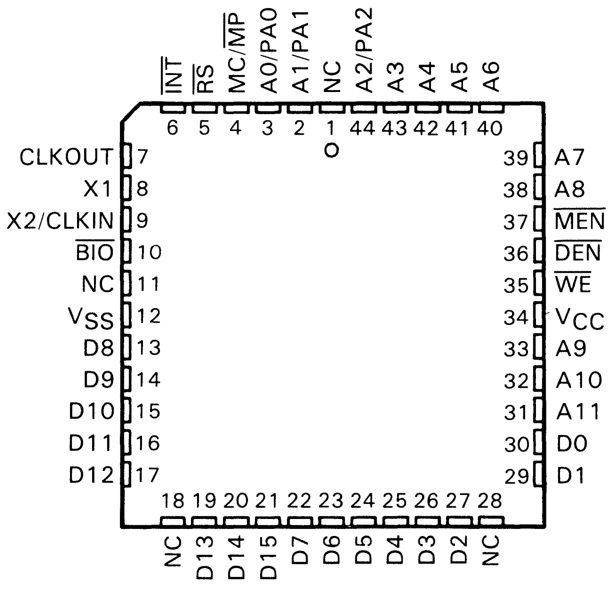
MAY 1983 – REVISED OCTOBER 1985

- 160-ns Instruction Cycle
- 144-Word On-Chip Data RAM
- ROMless Version – TMS32010
- 1.5K-Word On-Chip Program ROM – TMS320M10
- External Memory Expansion to a Total of 4K Words at Full Speed
- 16-Bit Instruction/Data Word
- 32-Bit ALU/Accumulator
- 16 × 16-Bit Multiply in 160-ns
- 0 to 16-Bit Barrel Shifter
- Eight Input and Eight Output Channels
- 16-Bit Bidirectional Data Bus with 50-Megabits-per-Second Transfer Rate
- Interrupt with Full Context Save
- Signed Two's-Complement Fixed-Point Arithmetic
- NMOS Technology
- Single 5-V Supply
- Two Versions Available
  TMS32010 . . . 20.5 MHz Clock
  TMS32010-25 . . . 25.0 MHz Clock

N PACKAGE
(TOP VIEW)



FN PACKAGE
(TOP VIEW)



description

The TMS32010 is the first member of the new TMS320 digital signal processing family, designed to support a wide range of high-speed or numeric-intensive applications. This 16/32-bit single-chip microcomputer combines the flexibility of a high-speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The TMS320 family contains the first MOS microcomputers capable of executing better than 6 million instructions per second. This high throughput is the result of the comprehensive, efficient, and easily programmed instruction set and of the highly pipelined architecture. Special instructions have been incorporated to speed the execution of digital signal processing (DSP) algorithms.

The TMS320 family's unique versatility and power give the design engineer a new approach to a variety of complex applications. In addition, these microcomputers are capable of providing the multiple functions often required for a single application. For example, the TMS320 family can enable an industrial robot to

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS
POST OFFICE BOX 1443 ● HOUSTON, TEXAS 77001

PIN NOMENCLATURE

NAME	I/O	DEFINITION
A11-A0/PA2-PA0	O	External address bus. I/O port address multiplexed over PA2-PA0.
$\overline{\text{BIO}}$	I	External polling input for bit test and jump operations.
CLKOUT	O	System clock output, ¼ crystal/CLKIN frequency.
D15-D0	I/O	16-bit data bus.
$\overline{\text{DEN}}$	O	Data enable indicates the processor accepting input data on D15-D0.
$\overline{\text{INT}}$	I	Interrupt.
MC/$\overline{\text{MP}}$	I	Memory mode select pin. High selects microcomputer mode. Low selects microprocessor mode.
$\overline{\text{MEN}}$	O	Memory enable indicates that D15-D0 will accept external memory instruction.
NC		No connection.
$\overline{\text{RS}}$	I	Reset used to initialize the device.
V_{CC}	I	Power.
V_{SS}	I	Ground.
$\overline{\text{WE}}$	O	Write enable indicates valid data on D15-D0.
X1	I	Crystal input.
X2/CLKIN	I	Crystal input or external clock input.

synthesize and recognize speech, sense objects with radar or optical intelligence, and perform mechanical operations through digital servo loop computations.

architecture

The TMS320 family utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory lie in two separate spaces, permitting a full overlap of the instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfers between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

The TMS32010 utilizes hardware to implement functions that other processors typically perform in software. For example, this device contains a hardware multiplier to perform a multiplication in a single 160-ns cycle. There is also a hardware barrel shifter for shifting data on its way into the ALU. Finally, extra hardware has been included so that auxiliary registers, which provide indirect data RAM addresses, can be configured in an autoincrement/decrement mode for single-cycle manipulation of data tables. This hardware-intensive approach gives the design engineer the type of power previously unavailable on a single chip.

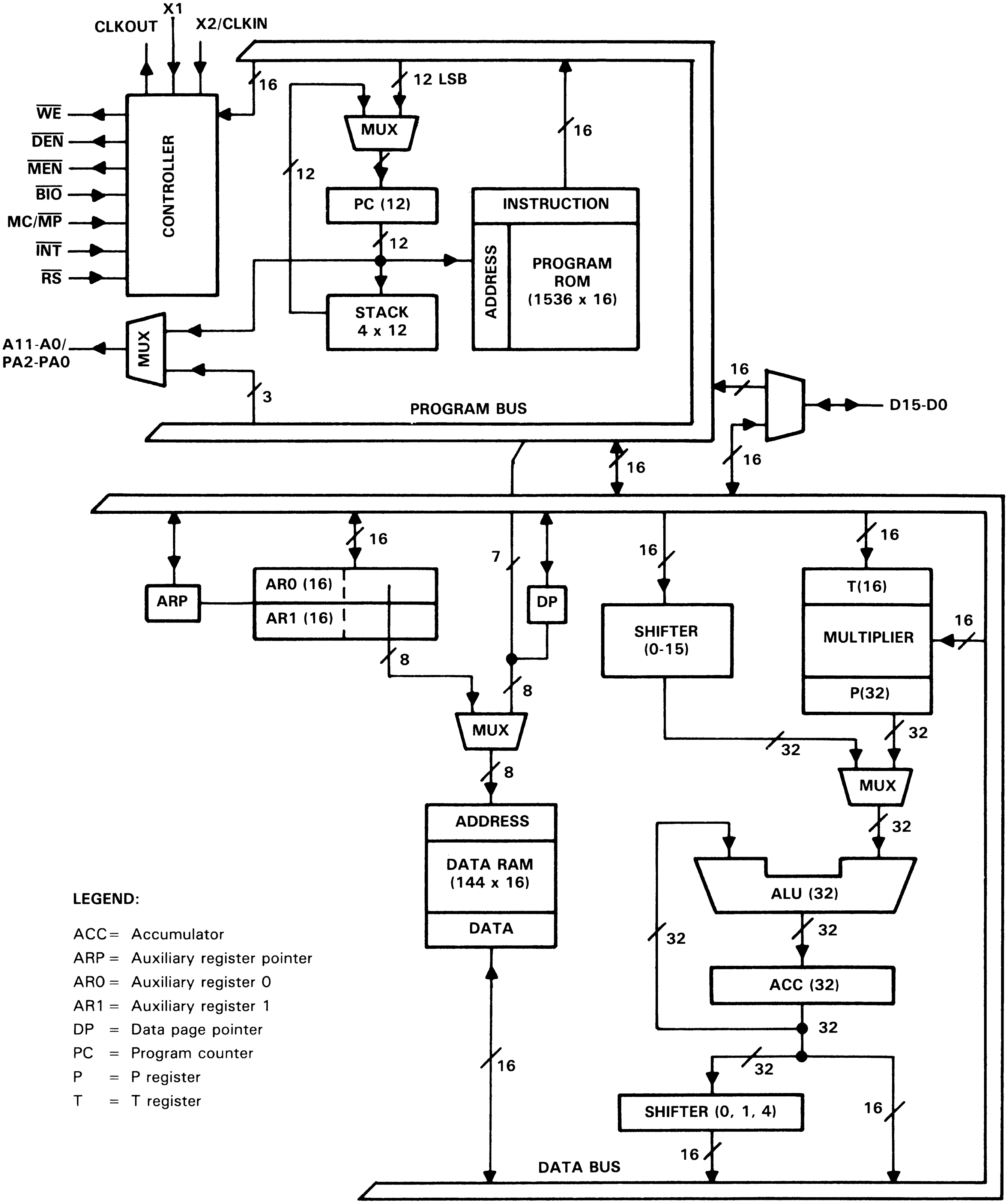
32-bit ALU/accumulator

The TMS32010 contains a 32-bit ALU and accumulator that support double-precision arithmetic. The ALU operates on 16-bit words taken from the data RAM or derived from immediate instructions. Besides the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller.

shifters

A barrel shifter is available for left-shifting data 0 to 15 places before it is loaded into, subtracted from, or added to the accumulator. This shifter extends the high-order bit of the data word and zero-fills the low-order bits for two's-complement arithmetic. A second shifter left-shifts the upper half of the accumulator 0, 1, or 4 places while it is being stored in the data RAM. Both shifters are very useful for scaling and bit extraction.

functional block diagram



LEGEND:

ACC =	Accumulator
ARP =	Auxiliary register pointer
AR0 =	Auxiliary register 0
AR1 =	Auxiliary register 1
DP =	Data page pointer
PC =	Program counter
P =	P register
T =	T register

16 × 16-bit parallel multiplier

The TMS32010's multiplier performs a 16 × 16-bit, two's-complement multiplication in one 160-ns instruction cycle. The 16-bit T Register temporarily stores the multiplicand; the P Register stores the 32-bit result. Multiplier values either come from the data memory or are derived immediately from the MPYK (multiply immediate) instruction word. The fast on-chip multiplier allows the TMS32010 to perform such fundamental operations as convolution, correlation, and filtering at the rate of better than 3 million samples per second.

program memory expansion

The TMS320M10 is equipped with a 1536-word ROM which is mask-programmed at the factory with a customer's program. It can also execute from an additional 2560 words of off-chip program memory at full speed. This memory expansion capability is especially useful for those situations where a customer has a number of different applications that share the same subroutines. In this case, the common subroutines can be stored on-chip while the application specific code is stored off-chip.

The TMS320M10 can operate in either of the following memory modes via the MC/$\overline{\text{MP}}$ pin:

Microcomputer Mode (MC) — Instruction addresses 0-1535 fetched from on-chip ROM. Those with addresses 1536-4095 fetched from off-chip memory at full speed.

Microprocessor Mode ($\overline{\text{MP}}$) — Full-speed execution from all 4096 off-chip instruction addresses.

The TMS32010 is identical to the TMS320M10, except that the TMS32010 operates only in the microprocessor mode. Henceforth, TMS32010 refers to both versions.

The ability of the TMS32010 to execute at full speed from off-chip memory provides the following important benefits:

- Easier prototyping and development work than is possible with a device that can address only on-chip ROM,
- Purchase of a standard off-the-shelf product rather than a semi-custom mask-programmed device,
- Ease of updating code,
- Execution from external RAM,
- Downloading of code from another microprocessor, and
- Use of off-chip RAM to expand data storage capability.

input/output

The TMS32010's 16-bit parallel data bus can be utilized to perform I/O functions at burst rates of 50 million bits per second. Available for interfacing to peripheral devices are 128 input and 128 output bits consisting of eight 16-bit multiplexed input ports and eight 16-bit multiplexed output ports. In addition, a polling input for bit test and jump operations ($\overline{\text{BIO}}$) and an interrupt pin ($\overline{\text{INT}}$) have been incorporated for multitasking.

interrupts and subroutines

The TMS32010 contains a four-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the TMS32010's complete context. The instructions, PUSH stack from accumulator, and POP stack to accumulator permit a level of nesting restricted only by the amount of available RAM. The interrupts used in the TMS32010 are maskable.

instruction set

The TMS32010's comprehensive instruction set supports both numeric-intensive operations, such as signal processing, and general-purpose operations, such as high-speed control. The instruction set, explained in Tables 1 and 2, consists primarily of single-cycle single-word instructions, permitting execution rates of better than 6 million instructions per second. Only infrequently used branch and I/O instructions are multicycle.

The TMS32010 also contains a number of instructions that shift data as part of an arithmetic operation. These all execute in a single cycle and are very useful for scaling data in parallel with other operations.

Three main addressing modes are available with the TMS32010 instruction set: direct, indirect, and immediate addressing.

direct addressing

In direct addressing, seven bits of the instruction word concatenated with the data page pointer form the data memory address. This implements a paging scheme in which the first page contains 128 words and the second page contains 16 words. In a typical application, infrequently accessed variables, such as those used for servicing an interrupt, are stored on the second page. The instruction format for direct addressing is shown below.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								0	dma						

Bit 7 = 0 defines direct addressing mode. The opcode is contained in bits 15 through 8. Bits 6 through 0 contain data memory address.

The seven bits of the data memory address (dma) field can directly address up to 128 words (1 page) of data memory. Use of the data memory page pointer is required to address the full 144 words of data memory.

Direct addressing can be used with all instructions requiring data operands except for the immediate operand instructions.

indirect addressing

Indirect addressing forms the data memory address from the least significant eight bits of one of two auxiliary registers, AR0 and AR1. The auxiliary register pointer (ARP) selects the current auxiliary register. The auxiliary registers can be automatically incremented or decremented in parallel with the execution of any indirect instruction to permit single-cycle manipulation of data tables. The instruction format for indirect addressing is as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPCODE								1	0	INC	DEC	ARP	0	0	ARP

Bit 7 = 1 defines indirect addressing mode. The opcode is contained in bits 15 through 8. Bits 7 through 0 contain indirect addressing control bits.

Bit 3 and bit 0 control the Auxiliary Register Pointer (ARP). If bit 3 = 0, then the content of bit 0 is loaded into the ARP. If bit 3 = 1, then the content of ARP remains unchanged. ARP = 0 defines the contents of ARO as memory address. ARP = 1 defines the contents of AR1 as memory address.

Bit 5 and bit 4 control the auxiliary registers. If bit 5 = 1, then the ARP defines which auxiliary register is to be incremented by 1. If bit 4 = 1, then the ARP defines which auxiliary register is to be decremented by 1. If bit 5 and bit 4 are zero, then neither auxiliary register is incremented or decremented. Bits 6, 2, and 1 are reserved and should always be programmed to zero.

Indirect addressing can be used with all instructions requiring data operands, except for the immediate operand instructions.

immediate addressing

The TMS32010 instruction set contains special ''immediate'' instructions. These instructions derive data from part of the instruction word rather than from the data RAM. Some very useful immediate instructions are multiply immediate (MPYK), load accumulator immediate (LACK), and load auxiliary register immediate (LARK).

TABLE 1. INSTRUCTION SYMBOLS

SYMBOL	MEANING
ACC	Accumulator
D	Data memory address field
I	Addressing mode bit
K	Immediate operand field
PA	3-bit port address field
R	1-bit operand field specifying auxiliary register
S	4-bit left-shift code
X	3-bit accumulator left-shift field

TABLE 2. TMS32010 INSTRUCTION SET SUMMARY

ACCUMULATOR INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABS	Absolute value of accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0
ADD	Add to accumulator with shift	1	1	0	0	0	0	◄— S —►				I	◄— D —►						
ADDH	Add to high-order accumulator bits	1	1	0	1	1	0	0	0	0	0	I	◄— D —►						
ADDS	Add to accumulator with no sign extension	1	1	0	1	1	0	0	0	0	1	I	◄— D —►						
AND	AND with accumulator	1	1	0	1	1	1	1	0	0	1	I	◄— D —►						
LAC	Load accumulator with shift	1	1	0	0	1	0	◄— S —►				I	◄— D —►						
LACK	Load accumulator immediate	1	1	0	1	1	1	1	1	1	0	◄— K —►							
OR	OR with accumulator	1	1	0	1	1	1	1	0	1	0	I	◄— D —►						
SACH	Store high-order accumulator bits with shift	1	1	0	1	0	1	1	◄X►			I	◄— D —►						
SACL	Store low-order accumulator bits	1	1	0	1	0	1	0	0	0	0	I	◄— D —►						
SUB	Subtract from accumulator with shift	1	1	0	0	0	1	◄— S —►				I	◄— D —►						
SUBC	Conditional subtract (for divide)	1	1	0	1	1	0	0	1	0	0	I	◄— D —►						
SUBH	Subtract from high-order accumulator bits	1	1	0	1	1	0	0	0	1	0	I	◄— D —►						
SUBS	Subtract from accumulator with no sign extension	1	1	0	1	1	0	0	0	1	1	I	◄— D —►						
XOR	Exclusive OR with accumulator	1	1	0	1	1	1	1	0	0	0	I	◄— D —►						
ZAC	Zero accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1
ZALH	Zero accumulator and load high-order bits	1	1	0	1	1	0	0	1	0	1	I	◄— D —►						
ZALS	Zero accumulator and load low-order bits with no sign extension	1	1	0	1	1	0	0	1	1	0	I	◄— D —►						

AUXILIARY REGISTER AND DATA PAGE POINTER INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LAR	Load auxiliary register	1	1	0	0	1	1	1	0	0	R	I	◄— D —►						
LARK	Load auxiliary register immediate	1	1	0	1	1	1	0	0	0	R	◄— K —►							
LARP	Load auxiliary register pointer immediate	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	0	0	K
LDP	Load data memory page pointer	1	1	0	1	1	0	1	1	1	1	I	◄— D —►						
LDPK	Load data memory page pointer immediate	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	K
MAR	Modify auxiliary register and pointer	1	1	0	1	1	0	1	0	0	0	I	◄— D —►						
SAR	Store auxiliary register	1	1	0	0	1	1	0	0	0	R	I	◄— D —►						

TABLE 2. TMS32010 INSTRUCTION SET SUMMARY (CONTINUED)

BRANCH INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B	Branch unconditionally	2	2	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BANZ	Branch on auxiliary register not zero	2	2	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGEZ	Branch if accumulator ≥ 0	2	2	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BGZ	Branch if accumulator > 0	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BIOZ	Branch on $\overline{\text{BIO}}$ = 0	2	2	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLEZ	Branch if accumulator ≤ 0	2	2	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BLZ	Branch if accumulator < 0	2	2	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BNZ	Branch if accumulator ≠ 0	2	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BV	Branch on overflow	2	2	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
BZ	Branch if accumulator = 0	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
CALA	Call subroutine from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0
CALL	Call subroutine immediately	2	2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	← BRANCH ADDRESS →											
RET	Return from subroutine or interrupt routine	2	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	1

T REGISTER, P REGISTER, AND MULTIPLY INSTRUCTIONS

MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APAC	Add P register to accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
LT	Load T register	1	1	0	1	1	0	1	0	1	0	I	← D →						
LTA	LTA combines LT and APAC into one instruction	1	1	0	1	1	0	1	1	0	0	I	← D →						
LTD	LTD combines LT, APAC, and DMOV into one instruction	1	1	0	1	1	0	1	0	1	1	I	← D →						
MPY	Multiply with T register, store product in P register	1	1	0	1	1	0	1	1	0	1	I	← D →						
MPYK	Multiply T register with immediate operand; store product in P register	1	1	1	0	0	← K →												
PAC	Load accumulator from P register	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0
SPAC	Subtract P register from accumulator	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0

TABLE 2. TMS32010 INSTRUCTION SET SUMMARY (CONCLUDED)

CONTROL INSTRUCTIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DINT	Disable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1
EINT	Enable interrupt	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0
LST	Load status register	1	1	0	1	1	1	1	0	1	1	I	◄			D			►
NOP	No operation	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
POP	POP stack to accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1
PUSH	PUSH stack from accumulator	2	1	0	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0
ROVM	Reset overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0
SOVM	Set overflow mode	1	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1
SST	Store status register	1	1	0	1	1	1	1	1	0	0	I	◄			D			►

I/O AND DATA MEMORY OPERATIONS																			
MNEMONIC	DESCRIPTION	NO. CYCLES	NO. WORDS	OPCODE INSTRUCTION REGISTER															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMOV	Copy contents of data memory location into next location	1	1	0	1	1	0	1	0	0	1	I	◄			D			►
IN	Input data from port	2	1	0	1	0	0	0	◄	PA	►	I	◄			D			►
OUT	Output data to port	2	1	0	1	0	0	1	◄	PA	►	I	◄			D			►
TBLR	Table read from program memory to data RAM	3	1	0	1	1	0	0	1	1	1	I	◄			D			►
TBLW	Table write from data RAM to program	3	1	0	1	1	1	1	1	0	1	I	◄			D			►

development systems and software support

Texas Instruments offers concentrated development support and complete documentation for designing a TMS32010-based microprocessor system. When developing an application, tools are provided to evaluate the performance of the processor, to develop the algorithm implementation, and to fully integrate the design's software and hardware modules. When questions arise, additional support can be obtained by calling the nearest Texas Instruments Regional Technology Center (RTC).

Sophisticated development operations are performed with the TMS32010 Evaluation Module (EVM), Macro Assembler/Linker, Simulator, and Emulator (XDS). In the initial phase of developing an application, the evaluation module is used to characterize the performance of the TMS32010. Once this evaluation phase is completed, the macro assembler and linker are used to translate program modules into object code and link them together. This puts the program modules into a form which can be loaded into the TMS32010 Evaluation Module, Simulator, or Emulator. The simulator provides a quick means for initially debugging TMS32010 software while the emulator provides real-time in-circuit emulation necessary to perform system level debug efficiently.

A complete list of TMS32010 software and hardware development tools is given in Table 3.

TABLE 3. TMS32010 SOFTWARE AND HARDWARE SUPPORT

HOST COMPUTER	OPERATING SYSTEM	PART NUMBER
	MACRO ASSEMBLERS/LINKERS	
DEC VAX	VMS	TMDS3240210-08
DEC VAX	Berkeley UNIX 4.1	TMDS3240220-08
IBM	MVS	TMDS3240310-08
IBM	CMS	TMDS3240320-08
TI/IBM PC	MS/PC-DOS	TMDS3240810-02
	SIMULATOR	
DEC VAX	VMS	TMDS3240211-08
TI/IBM PC	MS/PC-DOS	TMDS3240811-02
	DIGITAL FILTER DESIGN PACKAGE (DFDP)	
TI PC	MS-DOS	DFDP-TI001
IBM PC	PC-DOS	DFDP-IBM001
	HARDWARE	
Evaluation Module (EVM)		RTC/EVM320A-03
Analog Interface Board (AIB)		RTC/EVM320C-06
Emulator (XDS/22)		TMDS3262210

absolute maximum ratings over specified temperature range (unless otherwise noted)[†]

Supply voltage, V_{CC}[‡]	−0.3 V to 7 V
All input voltages	−0.3 V to 15 V
Output voltage	−0.3 V to 15 V
Continuous power dissipation	1.5 W
Air temperature range above operating device	0°C to 70°C
Storage temperature range	−55°C to +150°C

[†] Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

[‡] All voltage values are with respect to V_{SS}.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.75	5	5.25	V
V_{SS}	Supply voltage			0		V
V_{IH}	High-level input voltage	All inputs except CLKIN	2			V
		CLKIN	2.8			
V_{IL}	Low-level input voltage (all inputs)				0.8	V
I_{OH}	High-level output current (all outputs)				300	μA
I_{OL}	Low-level output current (all outputs)				2	mA
T_A	Operating free-air temperature		0		70	°C

NOTES: 1. Case temperature (T_C) for the TMS32010-25 and TMS32010FDL must be maintained below 90°C.

2. For dual-in-line package:
$R_{\theta JA}$ = 51.6°C/Watt
$R_{\theta JC}$ = 16.6°C/Watt.
For plastic chip-carrier package:
$R_{\theta JA}$ = 70°C/Watt
$R_{\theta JC}$ = 20°C/Watt.

electrical characteristics over specified temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage		I_{OH} = MAX		2.4	3		V
V_{OL}	Low-level output voltage		I_{OL} = MAX			0.3	0.5	V
I_{OZ}	Off-state output current		V_{CC} = MAX	V_O = 2.4 V			20	μA
				V_O = 0.4 V			−20	
I_I	Input current		V_I = V_{SS} to V_{CC}				±50	μA
I_{CC}‡	Supply current		V_{CC} = MAX	T_A = 0°C		180	275	mA
				T_A = 70°C			235§	mA
C_i	Input capacitance	Data bus	f = 1 MHz, All other pins 0 V			25		pF
		All others				15		
C_o	Output capacitance	Data bus				25		pF
		All others				10		

†All typical values except for I_{CC} are at V_{CC} = 5 V, T_A = 25°C.
‡I_{CC} characteristics are inversely proportional to temperature; i.e., I_{CC} decreases approximately linearly with temperature.
§Value derived from characterization data and not tested.

CLOCK CHARACTERISTICS AND TIMING

The TMS32010 can use either its internal oscillator or an external frequency source for a clock.

internal clock option

The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN (see Figure 1). The frequency of CLKOUT is one-fourth the crystal fundamental frequency. The crystal should be fundamental mode, and parallel resonant, with an effective series resistance of 30 ohms, a power dissipation of 1 mW, and be specified at a load capacitance of 20 pF.

PARAMETER	TEST CONDITIONS	TMS32010			TMS32010-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Crystal frequency f_x	0°C − 70°C	6.7		20.5	6.7		25.0	MHz
C1, C2	0°C − 70°C		10			10		pF

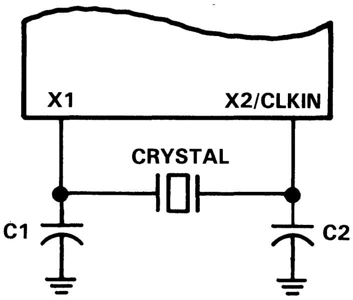


FIGURE 1. INTERNAL CLOCK OPTION

external clock option

An external frequency source can be used by injecting the frequency directly into X2/CLKIN with X1 left unconnected. The external frequency injected must conform to the specifications listed in the table below.

timing requirements over recommended operating conditions

		TMS32010			TMS32010-25			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(MC)}$	Master clock cycle time	48.78		150	40		150	ns
$t_{r(MC)}$	Rise time master clock input		5	10		5	10	ns
$t_{f(MC)}$	Fall time master clock input		5	10		5	10	ns
$t_{w(MCP)}$	Pulse duration master clock	$0.475t_{c(C)}$		$0.525t_{c(C)}$	$0.475t_{c(C)}$		$0.525t_{c(C)}$	ns
$t_{w(MCL)}$	Pulse duration master clock low, $t_{c(MC)} = 50$ ns		20			18		ns
$t_{w(MCH)}$	Pulse duration master clock high, $t_{c(MC)} = 50$ ns		20			18		ns

switching characteristics over recommended operating conditions

PARAMETER		TEST	TMS32010			TMS32010-25			UNIT
		CONDITIONS	MIN	NOM	MAX	MIN	NOM	MAX	
$t_{c(C)}$	CLKOUT cycle time†	$R_L = 870\ \Omega$ $C_L = 100$ pF, See Figure 2	195.12			160			ns
$t_{r(C)}$	CLKOUT rise time			10			10		ns
$t_{f(C)}$	CLKOUT fall time			8			8		ns
$t_{w(CL)}$	Pulse duration, CLKOUT low			92			74		ns
$t_{w(CH)}$	Pulse duration, CLKOUT high			90			72		ns
$t_{d(MCC)}$	Delay time CLKIN↑ to CLKOUT↓‡		25		60	25		60	ns

†$t_{c(C)}$ is the cycle time of CLKOUT, i.e., 4*$t_{c(MC)}$ (4 times CLKIN cycle time if an external oscillator is used).
‡Values given were derived from characterization data and are not tested.

PARAMETER MEASUREMENT INFORMATION

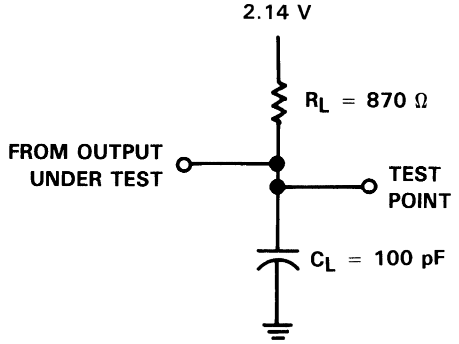
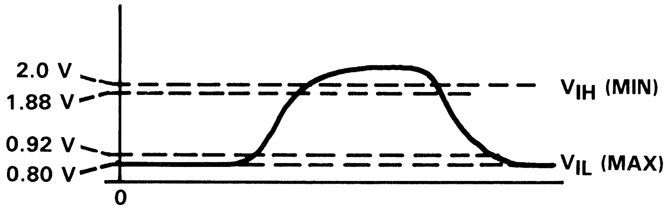
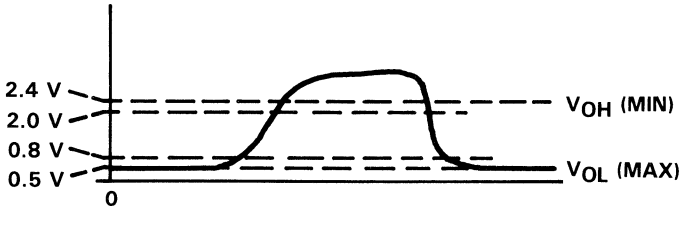


FIGURE 2. TEST LOAD CIRCUIT



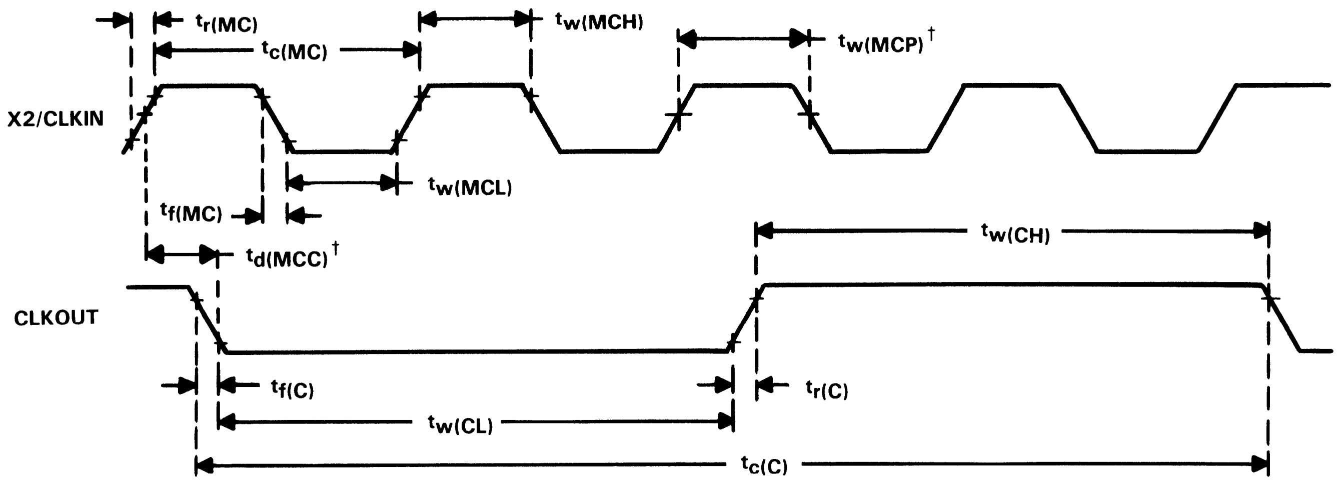
(a) INPUT



(b) OUTPUTS

FIGURE 3. VOLTAGE REFERENCE LEVELS

clock timing



NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
†$t_{d(MCC)}$ and $t_{w(MCP)}$ are referenced to an intermediate level of 1.5 volts on the CLKIN waveform.

MEMORY AND PERIPHERAL INTERFACE TIMING

switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d1}	Delay time CLKOUT↓ to address bus valid (see Note 4)	R_L = 870 Ω, C_L = 100 pF, See Figure 2	10†		50	ns
t_{d2}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↓		¼$t_{c(C)}$ − 5†		¼$t_{c(C)}$ + 15	ns
t_{d3}	Delay time CLKOUT↓ to $\overline{\text{MEN}}$↑		− 10†		15	ns
t_{d4}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↓		¼$t_{c(C)}$ − 5†		¼$t_{c(C)}$ + 15	ns
t_{d5}	Delay time CLKOUT↓ to $\overline{\text{DEN}}$↑		− 10†		15	ns
t_{d6}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↓		½$t_{c(C)}$ − 5†		½$t_{c(C)}$ + 15	ns
t_{d7}	Delay time CLKOUT↓ to $\overline{\text{WE}}$↑		− 10†		15	ns
t_{d8}	Delay time CLKOUT↓ to data bus OUT valid				¼$t_{c(C)}$ + 65	ns
t_{d9}	Time after CLKOUT↓ that data bus starts to be driven		¼$t_{c(C)}$ − 5†			ns
t_{d10}	Time after CLKOUT↓ that data bus stops being driven				¼$t_{c(C)}$ + 30†	ns
t_v	Data bus OUT valid after CLKOUT↓		¼$t_{c(C)}$ − 10			ns

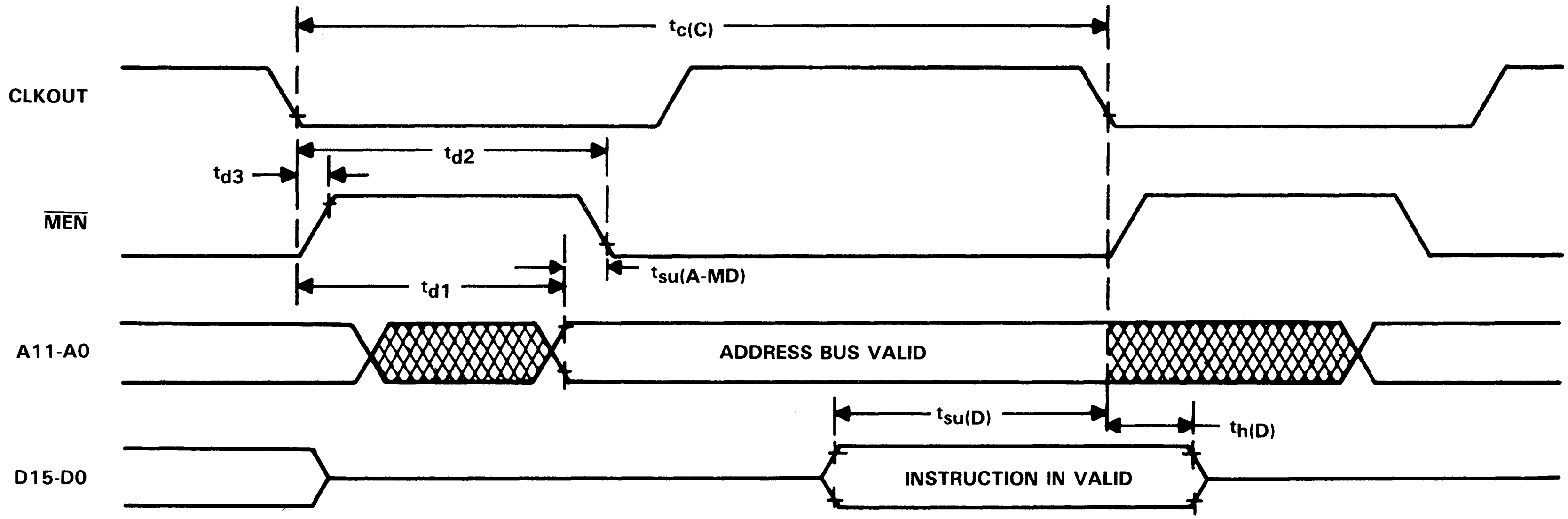
NOTE 4: Address bus will be valid upon $\overline{\text{WE}}$↑, $\overline{\text{DEN}}$↑, or $\overline{\text{MEN}}$↑.
†These values were derived from characterization data and are not tested.

timing requirements over recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
$t_{su(D)}$	Setup time data bus valid prior to CLKOUT↓	R_L = 870 Ω,	50			ns
$t_{su(A\text{-}MD)}$	Address bus setup time prior to $\overline{\text{MEN}}$↓ or $\overline{\text{DEN}}$↓	C_L = 100 pF,	¼$t_{c(C)-45}$			ns
$t_{h(D)}$	Hold time data bus held valid after CLKOUT↓	See Figure 2	0			ns

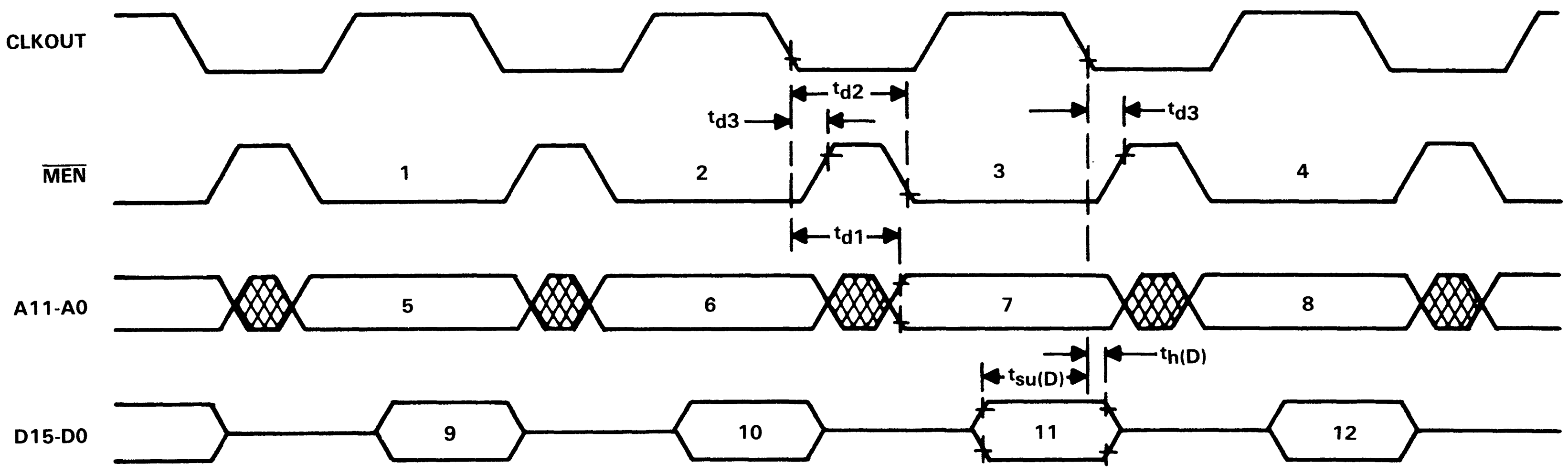
NOTE 5: Data may be removed from the data bus upon $\overline{\text{MEN}}$↑ or $\overline{\text{DEN}}$↑ preceding CLKOUT↓.

memory read



NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

TBLR instruction timing

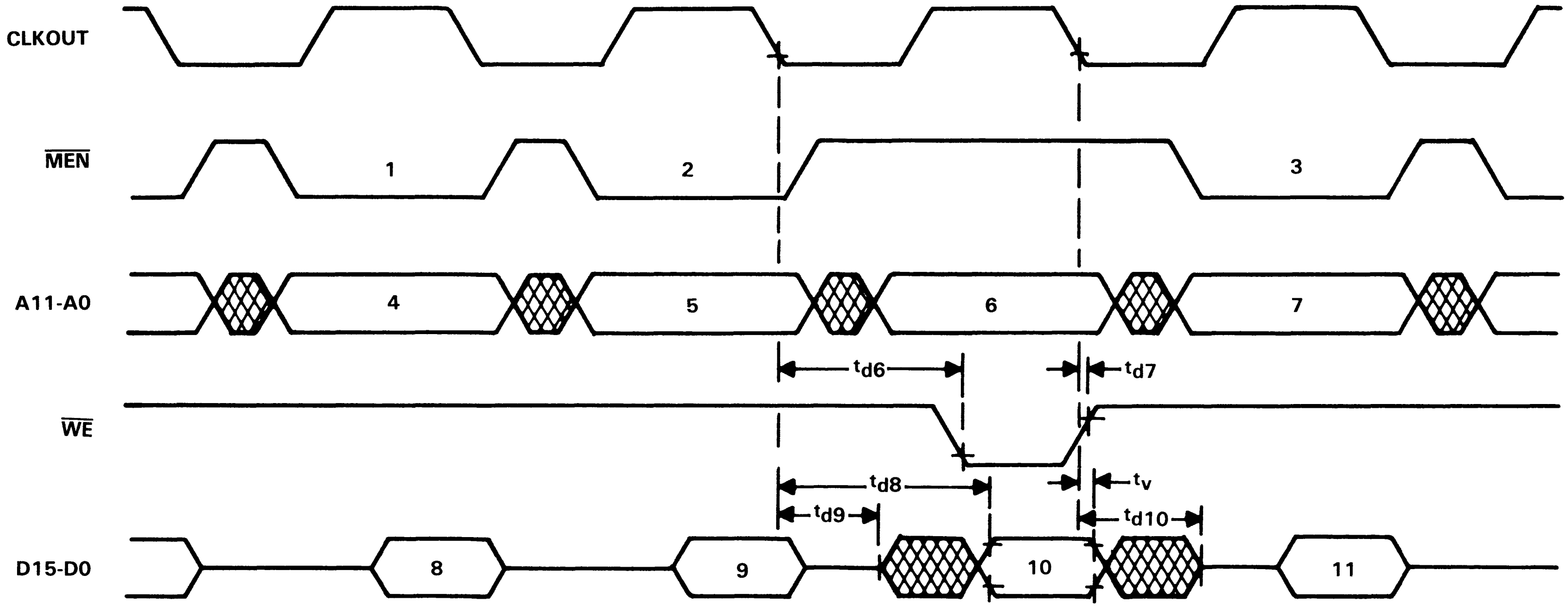


LEGEND:

1. TBLR INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. DATA FETCH
4. NEXT INSTRUCTION PREFETCH
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. ADDRESS BUS VALID
9. INSTRUCTION IN VALID
10. INSTRUCTION IN VALID
11. DATA IN VALID
12. INSTRUCTION IN VALID

NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

TBLW instruction timing

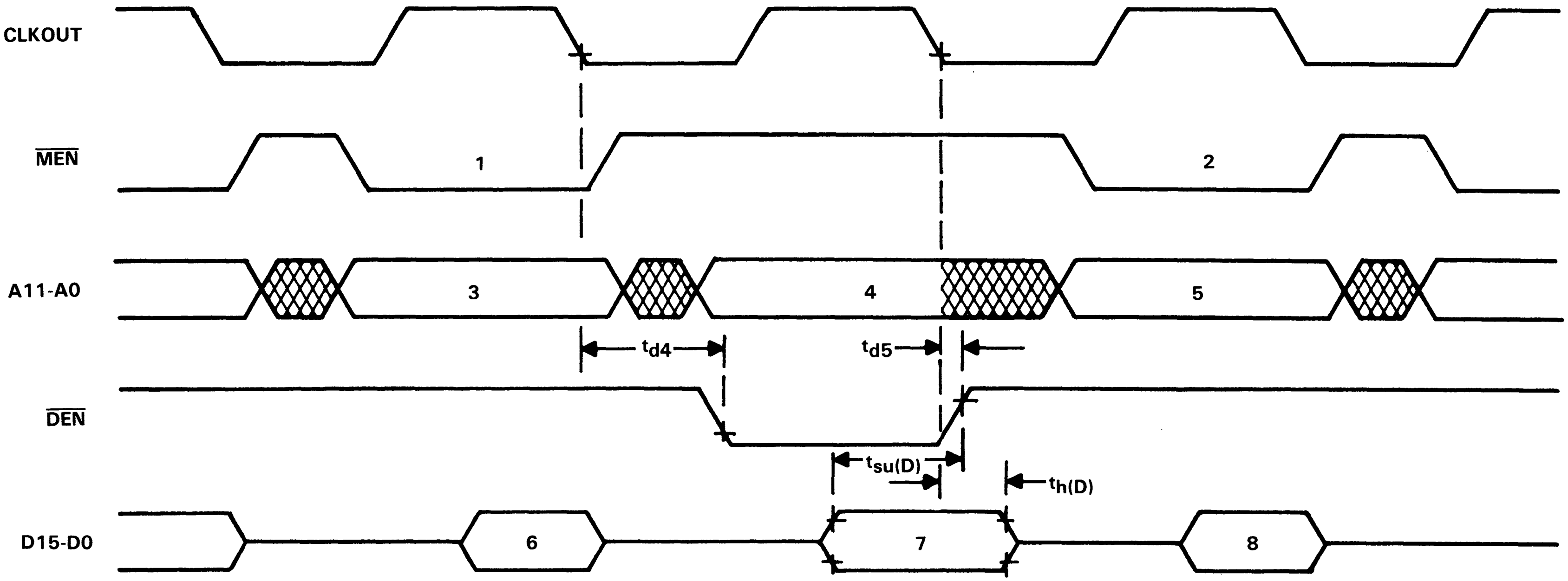


LEGEND:

1. TBLW INSTRUCTION PREFETCH
2. DUMMY PREFETCH
3. NEXT INSTRUCTION PREFETCH
4. ADDRESS BUS VALID
5. ADDRESS BUS VALID
6. ADDRESS BUS VALID
7. ADDRESS BUS VALID
8. INSTRUCTION IN VALID
9. INSTRUCTION IN VALID
10. DATA OUT VALID
11. INSTRUCTION IN VALID

NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

IN instruction timing

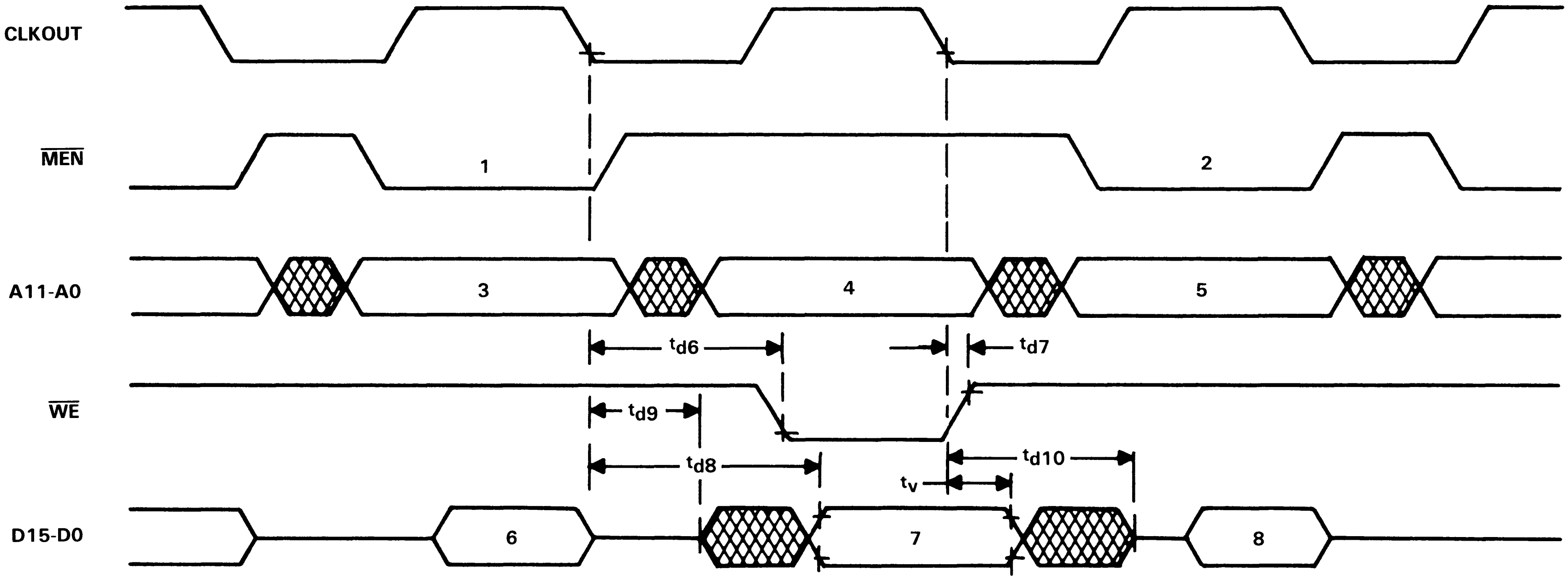


LEGEND:

1. IN INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION IN VALID
7. DATA IN VALID
8. INSTRUCTION IN VALID

NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

OUT instruction timing



LEGEND:

1. OUT INSTRUCTION PREFETCH
2. NEXT INSTRUCTION PREFETCH
3. ADDRESS BUS VALID
4. PERIPHERAL ADDRESS VALID
5. ADDRESS BUS VALID
6. INSTRUCTION IN VALID
7. DATA OUT VALID
8. INSTRUCTION IN VALID

NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

RESET ($\overline{RS}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{su(R)}$	Reset ($\overline{RS}$) setup time prior to CLKOUT. See Note 6.	50			ns
$t_{w(R)}$	$\overline{RS}$ pulse duration	$5t_{c(C)}$			ns

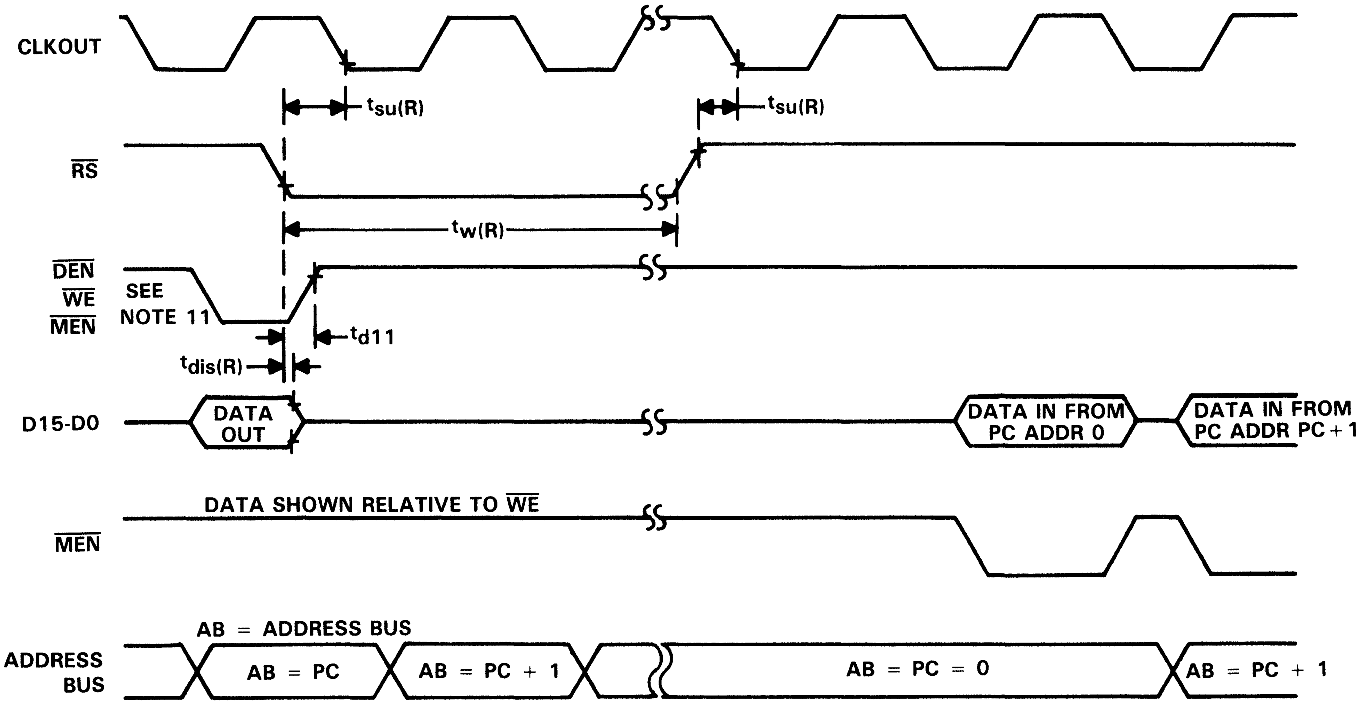
switching characteristics over recommended operating conditions

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{d11}	Delay time $\overline{DEN}$↑, $\overline{WE}$↑, and $\overline{MEN}$↑ from $\overline{RS}$	R_L = 870 Ω, C_L = 100 pF, See Figure 2			$\frac{1}{2}t_{c(C)}+50$†	ns
$t_{dis(R)}$	Data bus disable time after $\overline{RS}$				$\frac{1}{4}t_{c(C)}+50$†	ns

NOTE 6: $\overline{RS}$ can occur anytime during a clock cycle. Time given is minimum to ensure synchronous operation.

†These values were derived from characterization data and are not tested.

reset timing



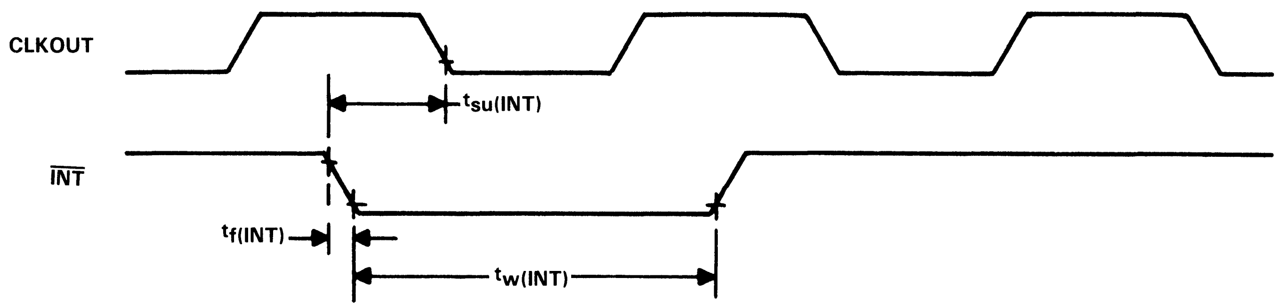
NOTES:
3. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.
7. $\overline{RS}$ forces $\overline{DEN}$, $\overline{WE}$, and $\overline{MEN}$ high and tristates data bus D0 through D15. AB outputs (and program counter) are synchronously cleared to zero after the next complete CLK cycle from ↓$\overline{RS}$.
8. $\overline{RS}$ must be maintained for a minimum of five clock cycles.
9. Resumption of normal program will commence after one complete CLK cycle from ↑$\overline{RS}$.
10. Due to the synchronizing action on $\overline{RS}$, time to execute the function can vary dependent upon when ↑$\overline{RS}$ or ↓$\overline{RS}$ occur in the CLK cycle.
11. Diagram shown is for definition purpose only. $\overline{DEN}$, $\overline{WE}$, and $\overline{MEN}$ are mutually exclusive.
12. During a write cycle, $\overline{RS}$ may produce an invalid write address.

INTERRUPT ($\overline{\text{INT}}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(INT)}$	Fall time $\overline{\text{INT}}$			15	ns
$t_{w(INT)}$	Pulse duration $\overline{\text{INT}}$	$t_{c(C)}$			ns
$t_{su(INT)}$	Setup time $\overline{\text{INT}}$↓ before CLKOUT↓	50			ns

interrupt timing



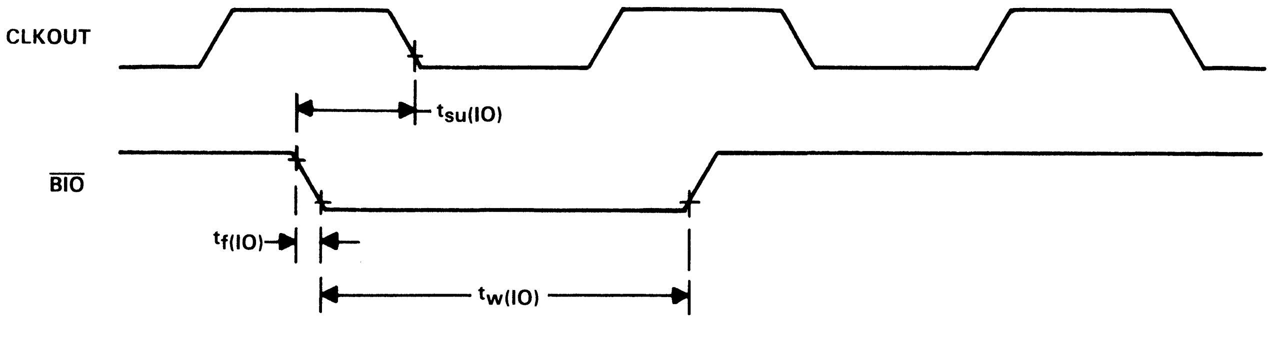
NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

I/O ($\overline{\text{BIO}}$) TIMING

timing requirements over recommended operating conditions

		MIN	NOM	MAX	UNIT
$t_{f(IO)}$	Fall time $\overline{\text{BIO}}$			15	ns
$t_{w(IO)}$	Pulse duration $\overline{\text{BIO}}$	$t_{c(C)}$			ns
$t_{su(IO)}$	Setup time $\overline{\text{BIO}}$↓ before CLKOUT↓	50			ns

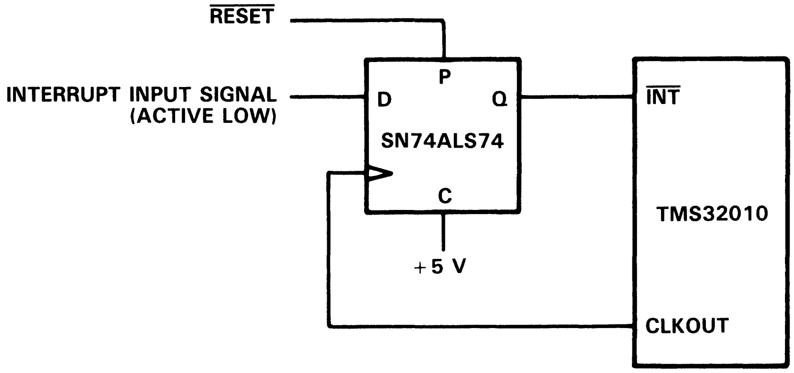
$\overline{\text{BIO}}$ timing

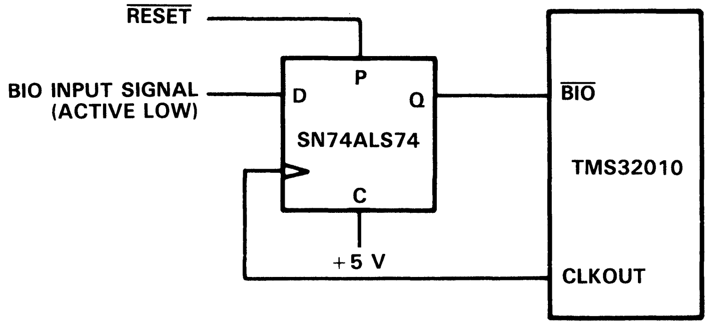


NOTE 3: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

input synchronization requirements

For systems using asynchronous inputs to the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ pins on the TMS32010, the external hardware shown in the diagrams below is recommended to ensure proper execution of interrupts and the BIOZ instruction. This hardware synchronizes the $\overline{\text{INT}}$ and $\overline{\text{BIO}}$ input signals with the rising edge of CLKOUT on the TMS32010. The pulse width required for these input signals is $t_{c(C)}$, which is one TMS32010 clock cycle, plus sufficient setup time for the flip-flop (dependent upon the flip-flop used).





TI standard symbolization for devices without on-chip ROM

SYMBOLIZATION

line 1:	(a)		(b)	TMS32010NL
line 2:	(c)	©1983 TI	(d)	DCU8327
line 3:	(e)	24655		

MEANINGS OF SYMBOLS

(a)	Texas Instruments trademark
(b)	Standard device number
(c)	TI design copyright
(d)	Tracking mark and date code
(e)	Lot code

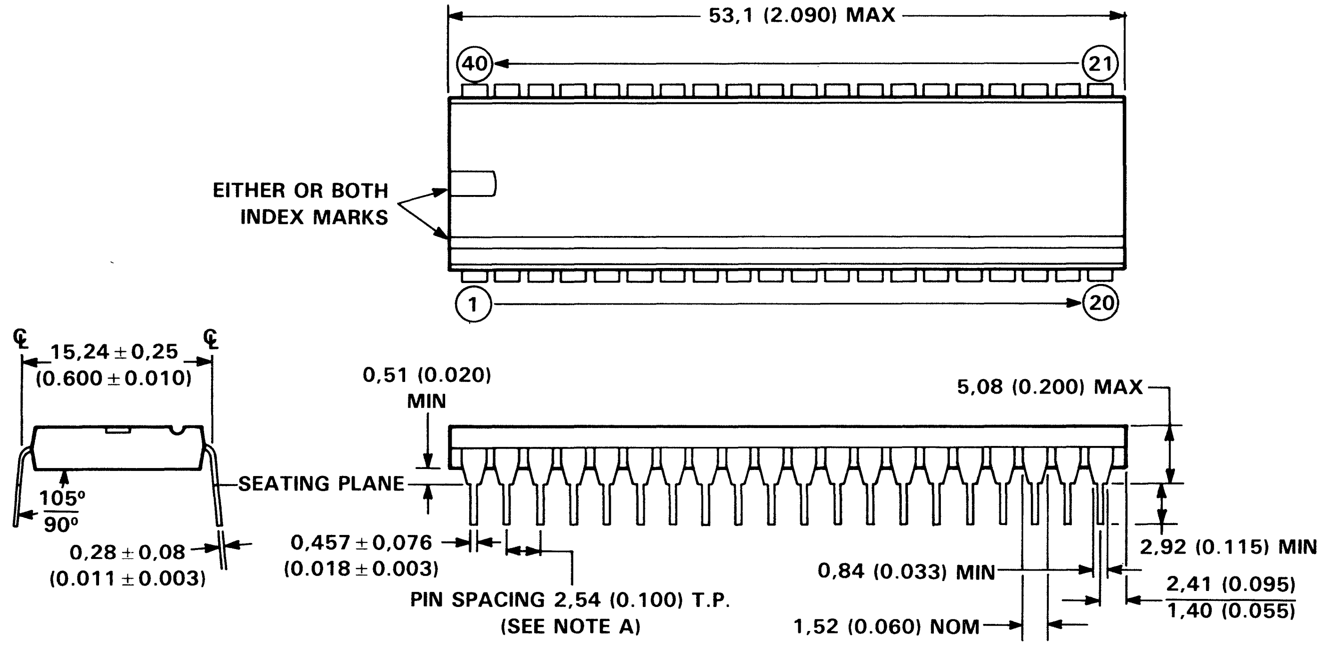
THERMAL DATA

thermal resistance characteristics

PACKAGE	RθJA (°C/W)	RθJC (°C/W)
40-pin plastic dual-in-line package	51.6	16.6
44-lead plastic chip carrier package	70	20

MECHANICAL DATA

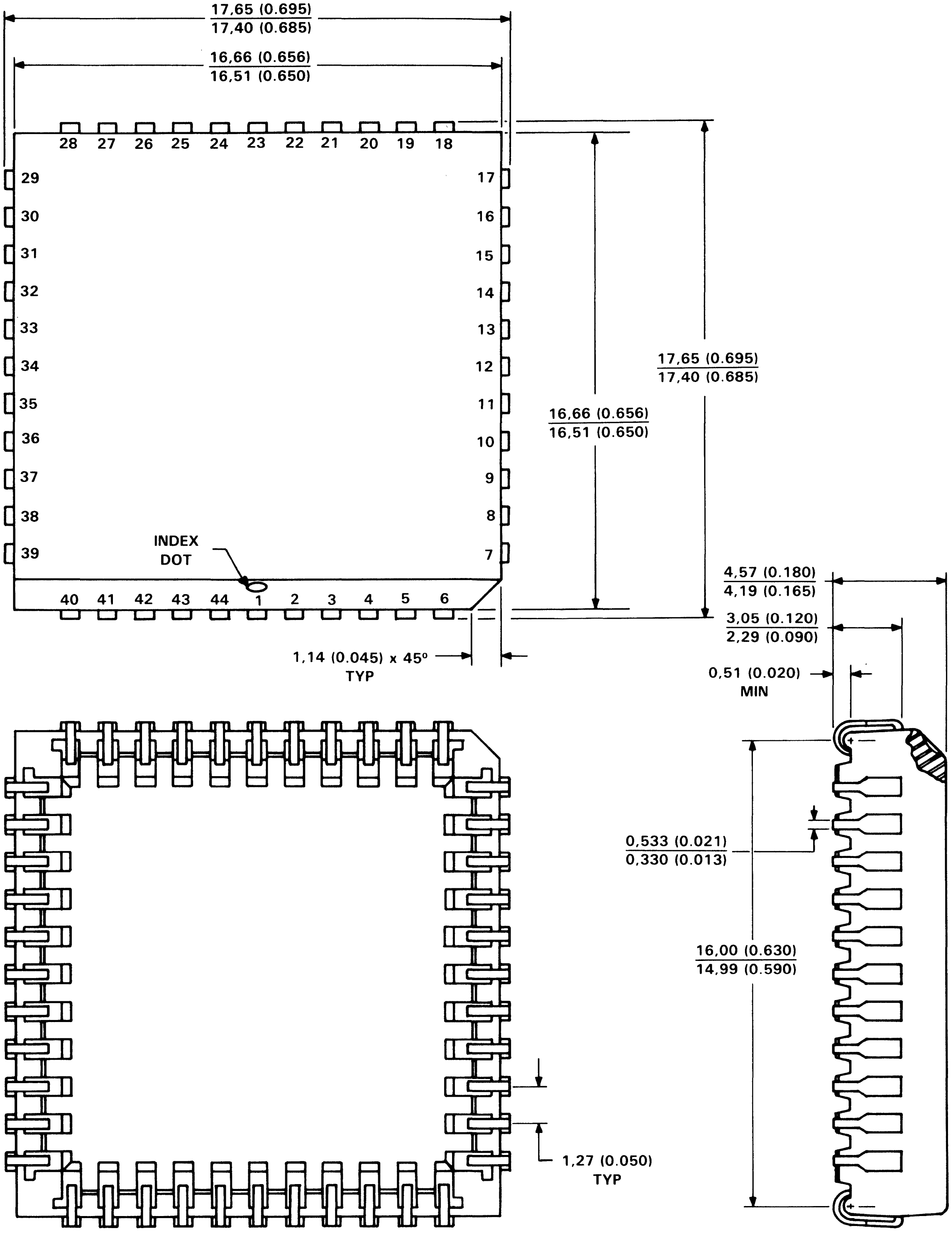
40-pin plastic dual-in-line package



ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

NOTE A: Each pin centerline is located within 0,254 (0.010) of its true longitudinal position.

44-lead plastic chip carrier package



ALL DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

C. TMS32010/TMS32011 System Migration

This appendix describes the changes in the hardware and programming features of the TMS32011 from those of the TMS32010.

1) The TMS32011 memory is configured as 144 words of on-chip data RAM and 1536 words of program ROM. The microprocessor mode of the TMS32010 is no longer available on the TMS32011 to utilize off-chip memory. The MC pin is utilized as a test pin on the TMS32011 and must be tied to V_{CC} to insure proper operation of the device.

2) The memory enable signal ($\overline{\text{MEN}}$) is not implemented on the TMS32011 for off-chip memory control.

3) The table write (TBLW) instruction should not be used on the TMS32011 or a potential bus conflict may occur. This instruction was used on the TMS32010 to write from internal RAM to external memory. Since the TMS32011 is a dedicated microcomputer with all program and data memory residing on-chip, the TBLW is not implemented.

4) I/O port 0 and port 1 on the TMS32011 are dedicated to the system control register and serial-port interface, leaving six input and six output channels for external interface.

5) $\overline{\text{INT}}$ is now an asynchronous interrupt input to the TMS32011. Only $\overline{\text{BIO}}$ requires the external flip-flop synchronization as on the TMS32010. The external interrupt is no longer at the active-low level; it is only edge-sensitive.

6) Three new maskable interrupts have been added to the TMS32011 to support the serial-port interface. $\overline{\text{FSX}}$, $\overline{\text{FSR}}$, and FR are the framing pulses used to control serial-port data transfers, and may be enabled to interrupt the TMS32011 when a data transfer has occurred. These interrupts are controlled through the system control register bits CR7-CR4.

7) Only the three lower-order internal address bits are external to the device for port addressing (PA2-PA0). These pins always output the three LSBs of the program counter and are inputs only in the peripheral mode.

8) A programmable logic output pin has been added to the TMS32011 for external peripheral control. The level present on the XF pin is controlled by the system control register bit 10 (CR10).

9) Only 11 bits are utilized in the program counter. Bit 12 of the PC must be a zero for all branch instructions.

D. TMS32011 Development Support/Part Order Information

All existing TMS32010 development tools can be used in TMS32011 development operations. No changes in software are necessary for designing a TMS32011-based microcomputer system when using the TMS32010 Evaluation Module (EVM), Macro/Assembler/Linker, and Simulator. The TMS32010 Emulator (XDS) connected to the TMS32011 in the peripheral mode provides realtime in-circuit emulation capability. (See Appendix C in the TMS32010 User's Guide for further information about the development tools.)

In the second half of 1986, Texas Instruments will provide a full in-circuit emulator for the TMS32011. This emulator will also support the TMS32010, TMS320C10, and TMS32010-25 devices (see the TMS320 Family Development Support Reference Guide for more information).

Table D-1 lists the software and hardware support available for the TMS32011.

Table D-1. TMS32011 Software and Hardware Support

HOST	OPERATING SYSTEM	PART NUMBER
TMS32010 MACRO ASSEMBLERS/LINKERS		
DEC VAX	VMS	TMDS3240210-08
TI/IBM PC	MS/PC-DOS	TMDS3240810-02
TMS32010 SIMULATORS		
DEC VAX	VMS	TMDS3240211-08
TI/IBM PC	MS/PC-DOS	TMDS3240811-02
TMS32010 DIGITAL FILTER DESIGN PACKAGE (DFDP)		
TI PC	MS-DOS	DFDP-TI001
IBM PC	PC-DOS	DFDP-IBM001
HARDWARE TOOL		**PART NUMBER**
Evaluation Module (EVM)		RTC/EVM320A-03
Analog Interface Board (AIB)		RTC/EVM320C-06
Emulators:		
XDS/22		TMDS3262210
Enhanced XDS/22 (available the second half of 1986)		TMDS3262211

Index

TEXAS INSTRUMENTS

Creating useful products
and services for you.

Printed in U.S.A.