TEXAS INSTRUMENTS

MSP58C80/C81/C82

Mixed-Signal Processor

User's Guide

1996

Mixed-Signal Products

MSP58C80/C81/C82 Mixed-Signal Processor User's Guide

July 1996
SPSU005B

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

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

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

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

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Printed in U.S.A. by
Great Impressions Printing & Graphics
Dallas, Texas

Preface

Read This First

About This Manual

This user's guide serves as a reference book for the MSP58C80/C81/C82 mixed-signal processor and is used in conjunction with the *TMS320C2x User's Guide*.

Related documentation is listed on page v.

How to Use This Manual

This document contains the following chapters:

- **Chapter 1** describes the design of the MSP58C80. This description includes a list of possible applications, a functional block diagram, a list of features, a pinout, signal descriptions, and a list of differences from the TMS320C25.
- **Chapter 2** describes the architecture of the MSP58C80. The memory organization is discussed in detail including program memory, data memory, memory maps, memory-mapped registers, I/O-mapped registers, alternate data space, interfacing to standard memory, and interfacing to DRAM. Interrupts, analog-to-digital converters, digital-to-analog converters, and low-power operation finish out the chapter.
- **Chapter 3** describes application information needed for the MSP58C80. This includes a software overview, the reset circuit, interrupts, and programming memory-mapped registers.
- **Chapter 4** describes the EVM58C80 development tool and related equipment. This chapter contains a discussion of the EVM58C80 development tool, hardware connection and setup, monitor-mode commands, and debug-mode commands.
- **Chapter 5** contains a list of frequently asked questions as compiled by TI's application staff. The questions range from MSP58C80 functionality to application information.

- **Appendix A** contains program code for a simple program. It also contains a list of DOS commands for assembling and linking, a step-by-step list for setting up the EVM58C80 hardware and executing a simple program, and an explanation of how to upgrade the EVM system software.

- **Appendix B** contains the electrical specifications and timings for the MSP58C80/C81/C82.

- **Appendix C** describes the results of different MEMTYPE and FREQ memory-mapped register setups and the resulting MSP58C80 functional stability.

- **Appendix D** contains customer information including an example of a new product release form and mechanical information.

- **Appendix E** contains a copy of the MSP58C20 data sheet.

- **Appendix F** contains information on programming the MSP58P80/P81. The MSP58P80/P81 incorporates a one-time programmable (OTP) 32K × 16-bit EPROM designed to function as a 64K × 8-bit EPROM (TMS27PC512) when used with a programmer adapter.

- **Appendix G** describes the differences of the MSP58C81/C82 from the MSP58C80. This description includes a functional block diagram, a list of features, a pinout, signal descriptions, differences in memory-mapped registers, differences in I/O-mapped registers, and differences in development tools.

- **Appendix H** is a glossary that defines acronyms and key words used in this book.

Notational Conventions

This document uses the following conventions:

- Program listings, program examples, and interactive displays are shown in a `special typeface` similar to that of a typewriter.

  Here is a sample program listing:

```
0011  0005  0001          .field    1, 2
0012  0005  0003          .field    3, 4
0013  0005  0006          .field    6, 3
```

Information About Cautions

This book may contain cautions and warnings.

> **This is an example of a caution statement.**
>
> **A caution statement describes a situation that could potentially damage your software or equipment.**

Related Documentation From Texas Instruments

The following book describes the TMS320C2x and related support tools. To obtain a copy of this TI document, call the Texas Instruments Literature Response Center at (800) 477–8924. When ordering, please identify the book by its title and literature number.

TMS320C2x User's Guide (SPRU014C) discusses the hardware aspects of the TMS320 family of 16-/32-bit single-chip digital signal processors. These processors combine the flexibility of a high-speed controller with the numerical capability of an array processor, offering an inexpensive alternative to custom VLSI and multichip bit-slice processors for signal processing.

From Other Sources

EVEREADY™ Battery Engineering Data: Alkaline (Volume 2A), 1990.

If You Need Assistance. . .

If you want to. . .	Do this. . .
Request more information about Texas Instruments Digital Signal Processing (DSP) products	Write to: Texas Instruments Incorporated Market Communications Manager, MS 736 P.O. Box 1443 Houston, Texas 77251–1443
Order Texas Instruments documentation	Call the TI Literature Response Center: **(800) 477–8924**
Report mistakes in this document or any other TI documentation	Send your comments to: Texas Instruments Incorporated Technical Publications Manager, MS 702 P.O. Box 1443 Houston, Texas 77251–1443

Trademarks

Apple and Macintosh are trademarks of Apple Computer, Inc.
Data I/O and Chiplab are trademarks of the Data I/O Corporation.
EVEREADY is a trademark of the Eveready Battery Company Inc.
IBM PC and PC-AT are trademarks of International Business Machines Corp.
MS-DOS is a trademark of Microsoft Corp.
Procomm Plus is a trademark of DATASTORM TECHNOLOGIES, INC.

Contents

Figures

Tables

Examples

Chapter 1

Introduction to the MSP58C80

The design of the MSP58C80 emphasizes both digital signal processing and general processing functions. The design also optimizes system cost for many applications by placing useful interfaces directly on the chip, eliminating the cost and board space associated with them. The MSP58C80 processing blocks are designed to implement voice-band DSP algorithms efficiently, leaving enough processing power to do general microcontroller functions.

Topic **Page**

1.1 Introduction

The MSP58C80 and its companion chip, the MSP58C20, integrate a 16.384 mega-instructions per second (MIPS) TMS320C25 digital signal processor (DSP) core with expanded internal memory, enhanced external memory interface, expanded I/O capability, enhanced clock control, two analog-to-digital converters (ADC), and a digital-to-analog converter (DAC). As a result, the MSP58C80 allows simple DSP system designs with few total chips.

The expanded internal memory allows complex control and DSP algorithms to be implemented with on-chip resources. The external memory interface supports standard memory (SRAM and ROM) and dynamic memory (DRAM), provides programmable wait states, and allows usage of a large alternate data space (16M words). This enhanced interface simplifies obtaining an economical solution for DSP applications with large data requirements.

Applications

The MSP58C80 robust instruction set, flexible on-chip interfaces, and fast execution speed make it an ideal choice for a variety of applications. The following list indicates some of these potential applications:

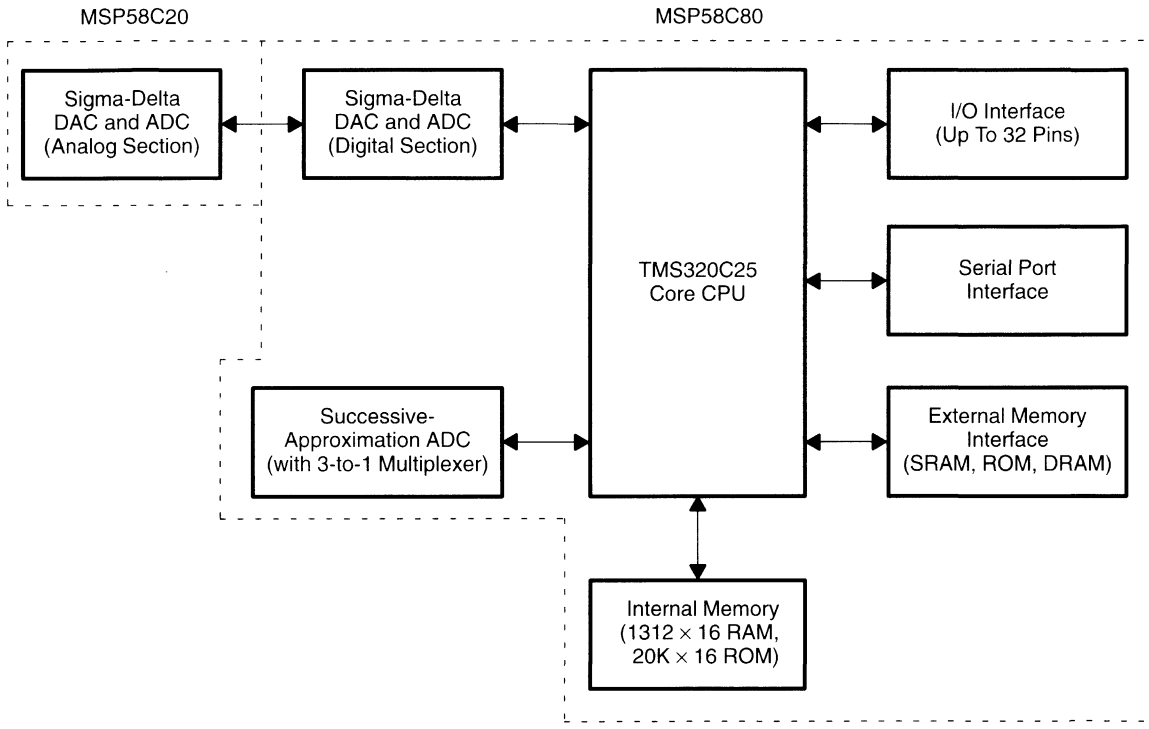
- ❑ Digital telephone answering machines
- ❑ Voice mail
- ❑ Speech vocoding
- ❑ Speech recognition
- ❑ Speaker verification
- ❑ Echo cancellation
- ❑ Modems
- ❑ Adaptive equalizers
- ❑ DTMF encoding/decoding
- ❑ FAX
- ❑ Cellular telephones
- ❑ Speaker phones
- ❑ Music synthesizers
- ❑ Toys and games
- ❑ Equipment for the handicapped
- ❑ Pattern recognition
- ❑ Laboratory instruments

Many MSP58C80 features, such as single-cycle multiply/accumulate instructions, 32-bit arithmetic unit, large auxiliary register file with a separate arithmetic unit, large on-chip RAM and ROM, DAC, and ADCs make the device particularly useful in digital signal processing systems. At the same time, general-purpose applications are greatly enhanced by the large address spaces, on-chip timers, standard and dynamic memory interfaces, serial interface, I/O interface, and multiple interrupt structure.

1.2 Description

The MSP58C80 has a DSP core, which performs both signal processing and control functions (Figure 1–1). To support high-speed processing, it has 20K words of internal ROM and 1312 words of internal RAM. It also has a flexible clock control system, which allows power consumption to be reduced during the execution of routines that do not require high-speed processing.

Figure 1–1. MSP58C80/MSP58C20 Simplified Functional Block Diagram



The external memory interface supports SRAM, ROM, DRAM, and memory-mapped peripherals, allowing a wide range of system configurations with minimal hardware design effort. The I/O interface provides up to 32 general-purpose I/O signals. In addition, eight external I/O expansion registers can be accessed by way of the memory interface. The serial port interface signals are compatible with codecs and many other serial devices with a minimum of external hardware.

The sigma-delta ADC and DAC have 16 bits of resolution and can be operated with variable sample rates. These converters are designed to work well with speech-band signals. The successive-approximation ADC has eight bits of resolution and is designed to work well with slowly-varying inputs, such as an attenuated telephone line voltage.

1.3 Features

The following is a list of features of the MSP58C80.

TMS320C25 Core CPU

- ❑ 16.384 MIPS TMS320C25 core (61-ns instruction cycle)
- ❑ 128K-word total data/program memory space
- ❑ 32-bit ALU/accumulator
- ❑ 16-bit parallel shifter between data bus and ALU
- ❑ Eight auxiliary registers with independent arithmetic unit
- ❑ 16-bit x 16-bit parallel multiplier with 32-bit product
- ❑ 8-level hardware stack
- ❑ 133-instruction TMS320C25 instruction set (see the TMS320C2x User's Guide)
  - ■ Single-cycle multiply/accumulate instructions
  - ■ Repeat instructions for efficient use of program space and enhanced execution
  - ■ Block moves for data/program management
  - ■ Instructions that support adaptive filtering, fast Fourier transform (FFT), and extended-precision arithmetic
  - ■ Bit-reversed addressing mode for radix-2 FFT
- ❑ On-chip timer for control operations
- ❑ Serial port for direct codec interface

Expanded Internal Memory

- ❑ 20K-word on-chip program ROM
- ❑ 1312-word on-chip RAM

Enhanced External Memory Interface

- ❑ Additional 16M-word alternate data space with prefetch-read and post-write capabilities
- ❑ Internal wait state generator
- ❑ Internal controller for direct memory interface that supports 8-bit wide and 16-bit wide standard memory and 1-bit wide and 4-bit wide dynamic memory
- ❑ Automatic conversion of external data to 16-bit internal format

Expanded I/O Capability

- ❑ Up to 32 terminals of general I/O (16 minimum)

Enhanced Clock Control

- ❑ Phase-locked loop (PLL) generates a clock that is software controllable between 2.048 MHz and 65.536 MHz
- ❑ Variable-speed processor clock for power consumption control
- ❑ External crystal, ceramic resonator or clock reference source reduced to 4.096 MHz
- ❑ Internal "real-time" counter tied to reference source

Analog-to-Digital and Digital-to-Analog Converters

- ❑ 16-bit resolution voice-band analog-to-digital converter
- ❑ 16-bit resolution voice-band digital-to-analog converter
- ❑ 8-bit resolution successive-approximation low-frequency analog-to-digital converter with on-chip 3-to-1 analog multiplexer

1.4 Pinout and Signal Descriptions

Figure 1–2 shows the signal assignments for the MSP58C80, Figure 1–3 illustrates the recommended clock and PLL circuits, Figure 1–4 shows the recommended reset circuit, and Table 1–1 provides signal descriptions.

Figure 1–2. MSP58C80 Terminal Assignments

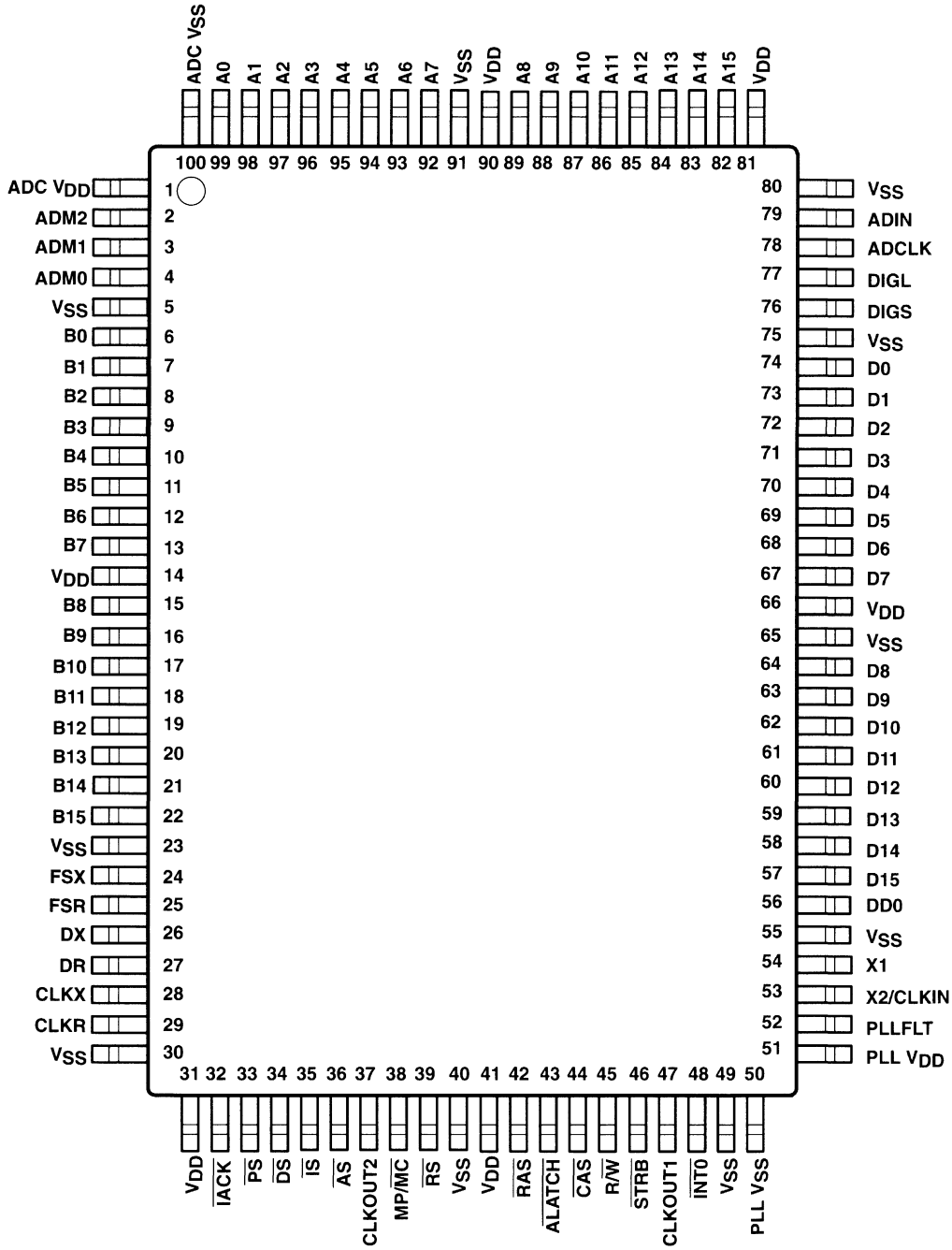
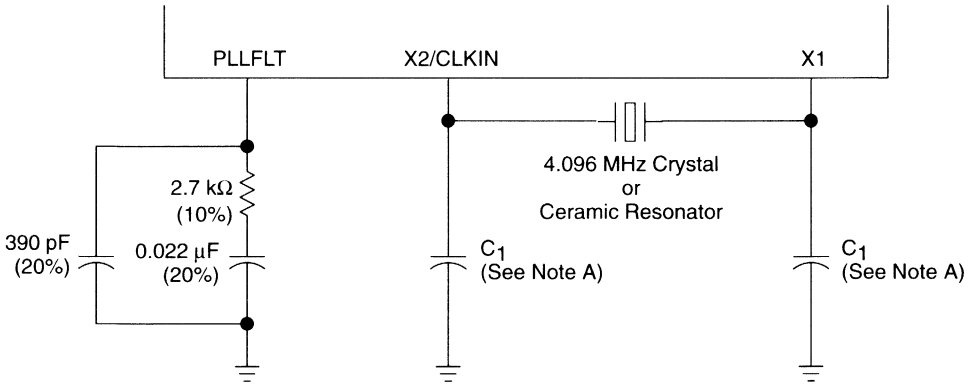


Figure 1–3. External Clock and PLL Filter



NOTE A: Capacitance is specified by crystal or ceramic resonator manufacturer.

Figure 1–4. Reset Circuit

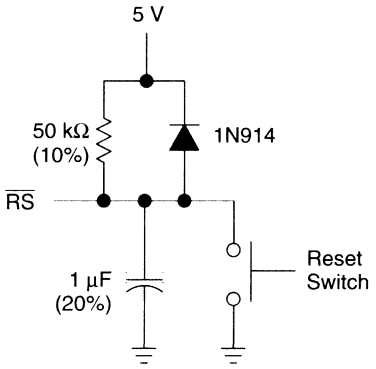


Table 1–1. MSP58C80 Signal Descriptions

Signal Name	Terminal No.	I/O/Z	Description
			Memory and I/O Lines
A0 – A15	99 – 92, 89 – 82	O	Memory address bus. It is a parallel address bus A15 (MSB) through A0 (LSB) that addresses external I/O, data, program, and alternate data spaces. There are two address bus configurations when using SRAM or ROM: ❑ in the program, data, and I/O space, the address is generated without multiplexing ❑ in the alternate data space, a 24-bit address is generated by first outputting the 8 MSBs on A8–A15 and then outputting the 16 LSBs on A0–A15. There are also two address bus configurations when using DRAM ❑ The address is placed on A0–A13 for 1-bit DRAM ❑ the address is placed on A1–A13 for 4-bit DRAM. The address is multiplexed for both DRAM configurations, with $\overline{RAS}$ and $\overline{CAS}$ providing the multiplexing signals.
B0 – B7 B8 – B15	6 – 13, 15 – 22	I/O/Z	B port is a parallel data bus [B15 (MSB) through B0 (LSB)] that provides 16 general I/O signals. The B15 signal can be configured to perform the function of the TMS320C25 XF signal, and the B0 signal can be configured to perform the function of the TMS320C25 $\overline{BIO}$ signal. B port signals are placed in a high-impedance state when not outputting or when $\overline{RS}$ is being asserted. Unused B port signals that are configured as inputs should be tied to either V_{SS} or V_{DD}.
D0 – D7 D8 – D15	74 – 67, 64 – 57	I/O/Z	D port is a parallel data bus [D15 (MSB) through D0 (LSB)] that transfers data between MSP58C80 and external I/O, data, program, and alternate data spaces. Those signals not being used to interface with any of the external spaces can be used as general I/O signals. D-port signals are placed in a high-impedance state when not outputting or when $\overline{RS}$ is being asserted. Unused D port signals that are configured as inputs should be tied to either V_{SS} or V_{DD}.
DD0	56	I/O	DRAM data signal for x1 DRAM interface that transfers data between MSP58C80 and 1-bit wide DRAM. DD0 is asserted high when inactive or when $\overline{RS}$ is being asserted.

Table 1–1. MSP58C80 Signal Descriptions (Continued)

Signal Name	Terminal No.	I/O/Z	Description
			Miscellaneous Control Signals
$\overline{IACK}$	32	O	Interrupt acknowledge signal. $\overline{IACK}$ indicates receipt of an interrupt and that the program is branching to the interrupt vector.
$\overline{INT0}$	48	I	External user interrupt input. This interrupt input is maskable by the interrupt mask register and the interrupt mode bit.
MP/$\overline{MC}$	38	I	Microprocessor/microcomputer mode select signal. In microcomputer mode (low), the signal causes the lower 32K words of program memory to be mapped internally. In microprocessor mode (high), the lower 32K words of program memory are mapped externally.
$\overline{RS}$	39	I	Reset input. $\overline{RS}$ low causes the MSP58C80 to terminate program execution and forces the program counter to zero. Driving $\overline{RS}$ high causes program execution to begin at location zero of program memory. $\overline{RS}$ affects various registers and status bits. This signal has a Schmitt-trigger input to allow the use of a simple reset circuit.
			Memory Control Signals
$\overline{ALATCH}$	43	O	Standard memory address demultiplex strobe serves as: ❑ The demultiplexing latch signal for 24-bit multiplexed address ❑ The time multiplexing signal for accessing lower/upper byte from byte-wide memories.
$\overline{CAS}$	44	O	DRAM column address strobe. $\overline{CAS}$ is asserted low after the column address has been set up for each of the columns accessed during a read or write operation. $\overline{CAS}$ is also used for $\overline{CAS}$-before-$\overline{RAS}$ refresh operations.
$\overline{RAS}$	42	O	DRAM row address strobe. $\overline{RAS}$ is asserted low after the row address has been set up for a read or write operation. $\overline{RAS}$ is also used for $\overline{CAS}$-before-$\overline{RAS}$ refresh operations.
R/$\overline{W}$	45	O	Read/Write signal. R/$\overline{W}$ controls the direction of a data transfer when communicating with an external device. It is normally high, unless asserted low to perform a write operation.
$\overline{STRB}$	46	O	Standard memory strobe signal. It provides timing for a standard memory data transfer. It is normally a high signal that goes low on address valid and returns high on data valid.
$\overline{IS}$	35	O	I/O, data, program, and alternate data space select signals. These signals are always high unless asserted low for communicating with a particular external space.
$\overline{DS}$	34	O	
$\overline{PS}$	33	O	
$\overline{AS}$	36	O	

Table 1–1. MSP58C80 Signal Descriptions (Continued)

Signal Name	Terminal No.	I/O/Z	Description
			Serial Interface Signals
CLKR	29	I	Receive clock input. CLKR is an external clock signal for clocking data from the DR signal into the RSR (serial-port receive shift register). It must be present during serial-port input.
CLKX	28	I	Transmit clock input. CLKX is an external clock signal for clocking data from the XSR (serial-port transmit shift register) to the DX signal. It must be present during serial port output.
DR	27	I	Serial-data receive input. Serial data is received in the RSR (serial-port receive shift register) by way of the DR signal.
DX	26	O/Z	Serial-data transmit output. Serial data is transmitted from the XSR (serial-port transmit shift register) by way of the DX signal. DX is placed in high-impedance state when not transmitting or when $\overline{RS}$ is being asserted.
FSR	25	I	Frame-synchronization input pulse for receive. The falling edge of the FSR pulse initiates the data-receive process, beginning the clocking of the RSR (serial-port receive shift register).
FSX	24	I/O	Frame-synchronization input/output pulse for transmit. The falling edge of the FSX pulse initiates the data-transmit process, beginning the clocking of the XSR (serial-port transmit shift register). After reset, this signal defaults to being an input. It may be programmed as an output by setting the TXM bit in status register ST1 to 1.
			Clock Control Signals
CLKOUT1	47	O	Instruction cycle clock output. The clock output has a period equal to the instruction cycle period. CLKOUT1 rises at the beginning of quarter-phase 3 (Q3) and falls at the beginning of quarter-phase 1 (Q1).
CLKOUT2	37	O	Second instruction cycle clock output. The clock output with a period equal to the instruction cycle period. CLKOUT2 rises at the beginning of quarter-phase 2 (Q2) and falls at the beginning of quarter-phase 4 (Q4).
PLLFLT	52	O	PLL filter connection. This terminal is used for attaching the external PLL filter.
X1	54	O	Crystal connection. The output signal is connected from the internal reference oscillator to the crystal or ceramic resonator. When a crystal or ceramic resonator is not used, this signal should be left unconnected.
X2/CLKIN	53	I	Crystal/clock connection. This is the input signal to the internal reference oscillator from the crystal or ceramic resonator. When a crystal or ceramic resonator is not used, a clock may be input to the device on this signal.

Table 1–1. MSP58C80 Signal Descriptions (Continued)

Signal Name	Terminal No.	I/O/Z	Description
			Audio-Band ADC/DAC Signals (MSP58C20 Interface)
ADCLK	78	O	Sigma-delta ADC/DAC modulator clock output. ADCLK is the clock output signal for connection to the MSP58C20 ADCLK signal.
ADIN	79	I	Sigma-delta ADC data input. ADIN is the digital input signal for connection to MSP58C20 ADOUT signal.
DIGL	77	O	Sigma-delta DAC level output. DIGL is the digital output signal for connection to the MSP58C20 DIGL signal.
DIGS	76	O	Sigma-delta DAC sign output. DIGS is the digital output signal for connection to the MSP58C20 DIGS signal.
			Successive-Approximation ADC Signals
ADM0	4	I	Successive-approximation low-frequency ADC MUX inputs. Three analog input signals, any one of which may be connected to the successive-approximation ADC using an on-chip analog multiplexer.
ADM1	3	I	
ADM2	2	I	
			Power Signals
ADC V_{SS}	100		Supply ground for the successive-approximation ADC.
ADC V_{DD}	1		Positive supply voltage for the successive-approximation ADC.
PLL V_{SS}	50		Supply ground for the phase-locked loop.
PLL V_{DD}	51		Positive supply voltage for the phase-locked loop.
V_{SS}	5, 23, 30, 40, 49, 55, 65, 75, 80, 91		Supply ground consists of ten ground signals tied together externally.
V_{DD}	14, 31, 41, 66, 81, 90		Positive supply voltage consists of six 5-V supply signals tied together externally.

1.5 Differences Between the MSP58C80 and the TMS320C25

The MSP58C80 is object-code compatible with the TMS320C25, except that the data, program, and I/O space mapping is different. Some of the TMS320C25 signal functions have been either modified or eliminated in order to meet the needs of MSP58C80 applications. The following list provides a summary of the differences between the TMS320C25 and the MSP58C80. For a description of the TMS320C25, refer to the *TMS320C2x User's Guide*.

1.5.1 Deletions

This section lists the deletions that have been made.

- Since the MSP58C80 was designed to allow efficient implementation of single-processor solutions, global memory allocation and multiprocessing control lines are not provided on the MSP58C80. As a result:
  - The MSP58C80 does not have a BR (bus request) line.
  - The GREG (global memory allocation register) is not available in the MSP58C80. Location 0005h in the data space is, therefore, unusable.
  - The $\overline{\text{HOLD}}$ (hold signal) and $\overline{\text{HOLDA}}$ (hold acknowledge signal) lines do not exist on the MSP58C80; therefore, changing the HM (hold mode) bit of status register ST1 with the SHM (set hold mode), RHM (reset hold mode), or LST1 (load status register ST1) instructions does not serve any functional purpose.
  - The $\overline{\text{SYNC}}$ (synchronization input) line does not exist on the MSP58C80.
- On the TMS320C25, the READY signal can be used to cause the processor to wait for slow external memory. The $\overline{\text{MSC}}$ (microstate complete signal) signal can implement a one-wait-state READY signal for slow memory. Neither the $\overline{\text{MSC}}$ signal nor the READY signal are available on the MSP58C80. Instead, software-controllable wait states have been implemented on the MSP58C80 (see the second item under enhancements).

1.5.2 Enhancements

This section lists the enhancements that have been made.

- The D port (D15–D0) was enhanced so that any signals not being used to interface with memory can be configured as general I/O signals on a bit-wise basis. In addition, the B port (B15–B0) was added to provide 16 terminals of general I/O. As a result, two of the B-port signals have been set up so that they can either provide the function of the TMS320C25 $\overline{\text{BIO}}$ (branch control input) and XF (external flag output) signals or be used as general I/O signals:
  - The B0 signal can be configured to provide the $\overline{\text{BIO}}$ function. To do this, the B0 signal must be configured as an input by setting the LSB of BDIR (B-port direction register) to 0, which is the default configuration after a reset.
  - The B15 signal can provide the XF function. To do this, the B15 signal must be configured as a low output by setting the MSB of BDIR to 1 and the MSB of BO (B-port output register) to 0 (these are not the default configurations after a reset).
  - D port lines that are being used to communicate with any of the external spaces must be programmed as inputs by setting the corresponding bits of DDIR to 0, which is the default configuration after a reset.
- A memory-mapped register (MEMTYPE) has been added to control the MSP58C80 flexible external memory interface. The MEMTYPE register can be used to program 0–7 wait states for the different external memory spaces. In addition, the type of external memory (standard or dynamic, narrow or wide) is indicated by the MEMTYPE register and refresh is enabled or disabled with this register.
- A memory-mapped register (FREQ) has been added to control the various user-adjustable clocks that are available on the MSP58C80. Since the MSP58C80 uses a PLL as part of its flexible clock-control system, the MSP58C80 requires an external PLL filter attached to the PLLFLT signal.
- The MSP58C80 provides an additional external memory space, the alternate data space. The MSP58C80 performs prefetch-read and postwrite operations when communicating with the alternate data space. As a result:

- Internal I/O-mapped registers have been added to the MSP58C80 to control the interface with the alternate data space. These registers are mapped to addresses 08h–0Fh. Therefore, IN (input data from port) or OUT (output data to port) instructions that use external I/O expansion registers 08h–0Fh on the TMS320C25 must be reassigned to use registers 00h–07h on the MSP58C80.
- To reduce power consumption and to be able to simultaneously perform a prefetch-read or postwrite operation while executing from internal program and data memory, the MSP58C80 only exercises the external address lines when external addresses or I/O-mapped registers are being accessed. Also, an interrupt vector is not placed on the address bus when the vector is internal.

❑ A sigma-delta interrupt and a real-time counter interrupt have been added to the MSP58C80. These interrupts occupy the same vectors and have the same priorities that the $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ external user interrupts have on the TMS320C25. As a result, the only user interrupt signal provided by the MSP58C80 is the $\overline{\text{INT0}}$ signal.

❑ Since the MSP58C80 has more internal ROM and RAM than the TMS320C25, the two devices have different program and data memory maps. Therefore, even though the CNFD (configure block as data memory) instruction configures RAM block B0 as data memory and the CNFP (configure block as program memory) instruction configures RAM block B0 as program memory on both devices, the resultant memory maps are different. Also, even though driving MP/$\overline{\text{MC}}$ low causes the lower portion of program memory to be mapped internally and driving MP/$\overline{\text{MC}}$ high causes the lower portion of program memory to be mapped externally on both devices, the resultant memory maps are different.

❑ The MSP58C80 $\overline{\text{RS}}$ signal has a Schmitt-trigger input in order to minimize the required external circuitry.

Chapter 2

MSP58C80 Architecture

The MSP58C80 architecture permits a high degree of parallelism. The data bus and program bus can carry operands simultaneously, permitting single-cycle multiply/accumulate operations. While the arithmetic logic unit (ALU) is performing one operation, the auxiliary register arithmetic unit (ARAU) can simultaneously perform another operation. Also, while instructions are being executed from internal program and data memory, the MSP58C80 can externally perform a prefetch from or a postwrite to the alternate data space.

Topic **Page**

2.1 Memory Organization

Adequate internal memory is included to implement many applications, eliminating the need for external expansion. However, if external memory is needed, the MSP58C80 memory interface directly supports communication with both standard and dynamic memory in the same application and can automatically generate wait states and refresh cycles. The MSP58C80 can address a total of 64K words of data memory, 16M words of alternate data memory, and 64K words of program memory.

This section explains memory management by discussing program and data memory, memory maps, memory-mapped registers, I/O-mapped registers, alternate data memory addressing, standard memory interfacing, DRAM interfacing, and I/O interfacing.

2.1.1 Program and Data Memory

The MSP58C80 has 1312 words of on-chip RAM that are divided into three blocks (B0, B1, and B2). Of these three blocks, block B0 (512 words) is configurable as either data or program memory, and blocks B1 and B2 (total of 800 words) are always configured as data memory.

The MSP58C80 can address a total of 64K words of data memory. The on-chip data memory and internally reserved locations are mapped into the lower 2K words of the data space.

In addition to the 64K-word data space, the MSP58C80 can also address a 16M-word alternate data space. Typically, the external memory interface is primarily used to store data to and recall data from this alternate data space at a low data-transfer rate, relying on fast internal memory for speed-sensitive algorithms and data manipulations. In such an application environment, the prefetch-read and postwrite operations available for the alternate data space can effectively eliminate the impact of slow external memory components on the internal performance of the DSP. For this to happen, the programmer must allow enough time between prefetch-read and postwrite operations so a given prefetch-read or postwrite operation is completed before the next one is initiated. Otherwise, the processor is forced to halt until the prior operation is completed.

The MSP58C80 has 20K words of on-chip program ROM and can address a total of 64K words of program space. When the MP/$\overline{\text{MC}}$ signal is held low, the on-chip program memory and internally reserved locations are mapped into the lower 32K words of the program space. When the MP/$\overline{\text{MC}}$ signal is held high, the program space is addressed externally. In either case, internal RAM block B0 can be configured as program memory.

The external program, data, and alternate data spaces can use standard or dynamic memory. Each of these spaces can be independently configured for narrow or wide dynamic memory (i.e., 1-bit or 4-bit DRAM) and narrow or wide standard memory (i.e., 8-bit or 16-bit SRAM or ROM) with the MEMTYPE memory-mapped register. The configuration that is chosen for the program space is also used for the I/O space.

Programmable wait states can be used to interface with slow, inexpensive external memory. When using programmable wait states, a different number of wait states can be specified for the external program, data, and alternate data spaces by using the MEMTYPE memory-mapped register. The number of wait states chosen for the program space is also used for the I/O space. A program that is stored in slow external memory can be executed at full speed by downloading it to RAM block B0 and then executing it internally.

2.1.2 Memory Maps

The MSP58C80 provides separate address spaces for program memory, data memory, alternate data memory, and I/O, as shown in Figure 2–1 through Figure 2–4. These spaces are distinguished externally by means of the $\overline{PS}$, $\overline{DS}$, $\overline{AS}$, and $\overline{IS}$ (program, data, alternate data, and I/O space select) signals.

The data space can be addressed in the direct addressing mode by placing the seven LSBs of the address in the instruction opcode and placing the nine MSBs of the address in the DP (data memory page pointer) register. Each page selected by the DP register consists of 128 words.

Program/data RAM block B0 (512 words) resides in pages 4–7 of the data space when configured as data memory and in the highest 512 words of the program space when configured as program memory. Data RAM block B1 (768 words) resides in pages 8–13 of the data space, and data RAM block B2 resides in the upper 32 words of data page 0. The remainder of page 0 of the data space is composed of memory-mapped registers and internally reserved locations. Pages 1–3 and 14–15 consist of internally reserved locations. The internally reserved locations cannot be used for storage, and their contents are undefined when read.

Block B0 is mapped into either the data or program space, based on the value of the CNF bit of status register ST1. The CNFD instruction clears the CNF bit, causing block B0 to be configured as data memory. The CNFP instruction sets the CNF bit, causing block B0 to be configured as program memory. A reset configures block B0 as data memory.

The program memory mapping is determined by the status of the MP/$\overline{MC}$ signal. When the MP/$\overline{MC}$ signal is a logic low, the lower 32K words of the program

space are mapped internally. Within the lower 32K words: a) the first 32 locations consist of interrupt vectors and reserved locations, b) the next 20 368 locations are mapped to program ROM, c) the next 80 locations are reserved for TI test code, and d) the last 12K locations are reserved for future expansion and return FFFFh when accessed. When the MP/$\overline{MC}$ signal is a logic high, all of the program space is mapped externally, with the exception that RAM block B0 can still be mapped as program memory with the CNFP instruction.

The MSP58C80 is capable of addressing 16M words in the alternate space and 16 locations in the I/O space. The lower eight locations in the I/O space are external I/O expansion registers and the upper eight locations are internal I/O-mapped registers used for communicating with the alternate data space.

Figure 2–1. System Memory Maps With MP/$\overline{MC}$ = 0 After a CNFD Instruction

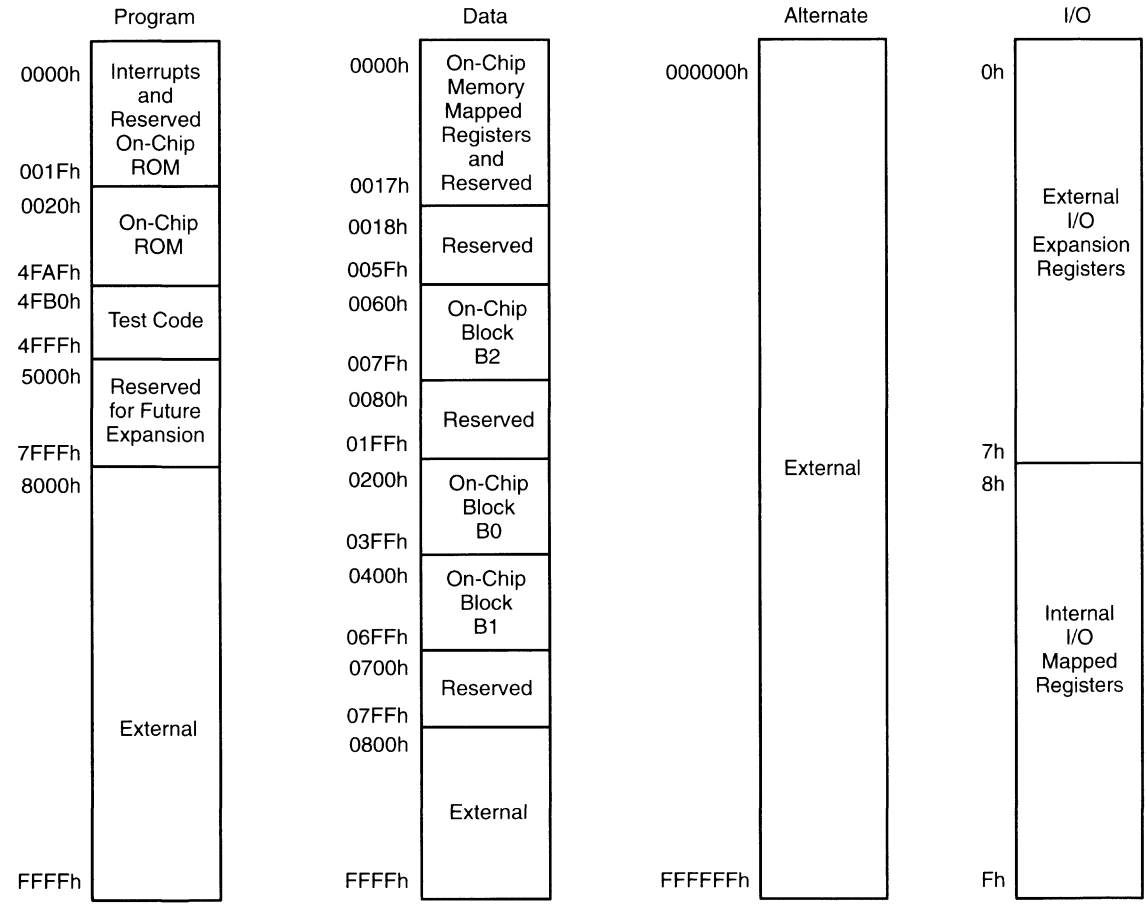


Figure 2–2. System Memory Maps With MP/$\overline{MC}$ = 0 After a CNFP Instruction

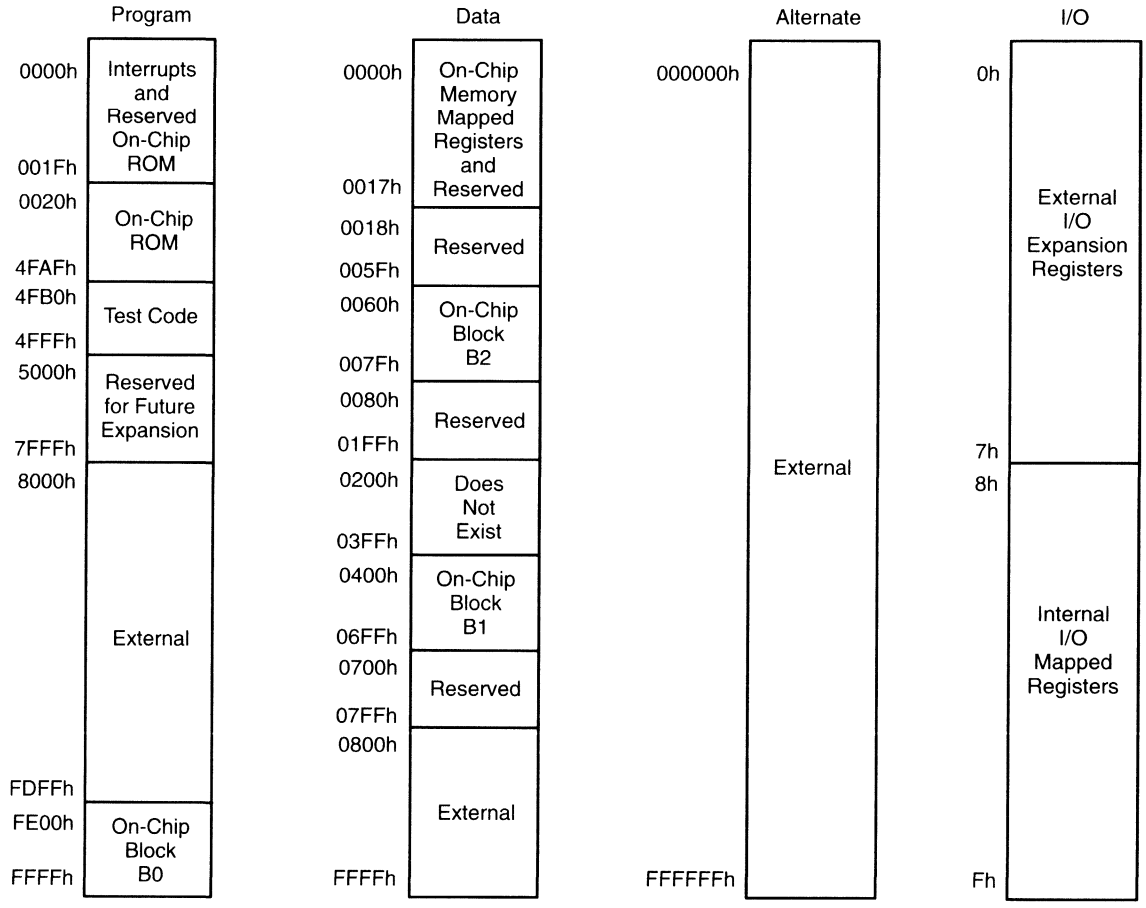


Figure 2–3. System Memory Maps With MP/$\overline{MC}$ = 1 After a CNFD Instruction

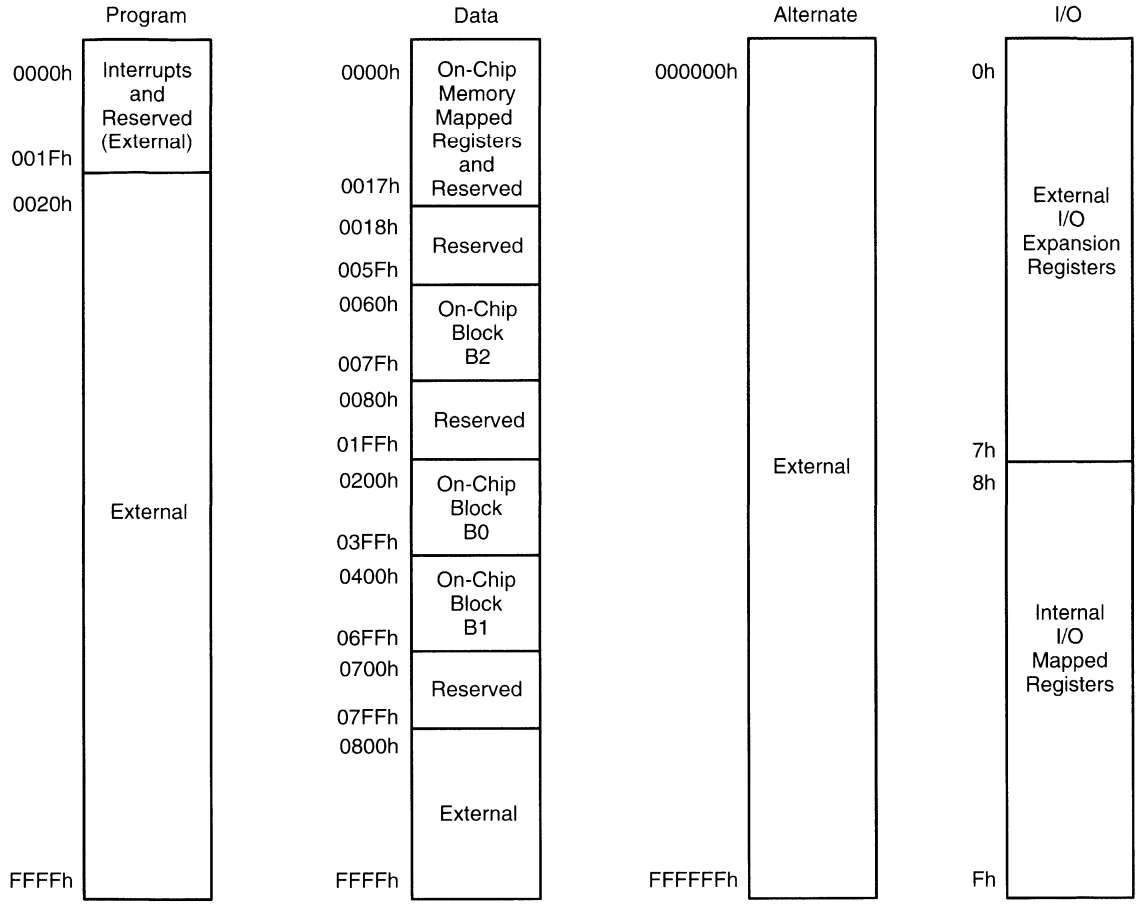
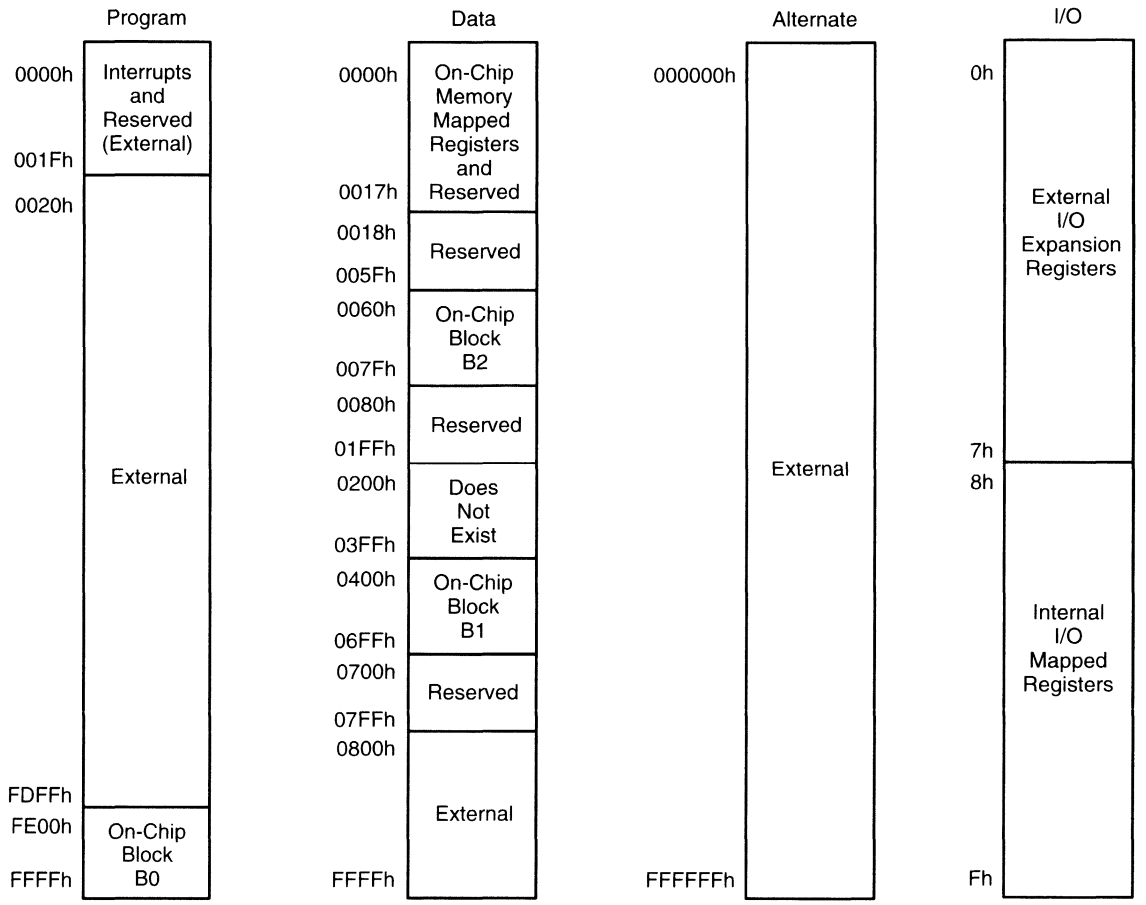


Figure 2–4. System Memory Maps With MP/$\overline{MC}$ = 1 After a CNFP Instruction



2.1.3 Memory-Mapped Registers

Memory-mapped registers may be accessed in the same manner as any other data memory location, with the exception that the BLKD (block move from data memory to data memory) and DMOV (data move in data memory) instructions cannot be performed with memory-mapped registers.

The lower 24 words of the MSP58C80 data space consist of 18 memory-mapped registers and 6 reserved locations (see Table 2–1). The reserved locations cannot be used for storage, and their contents are undefined when read. These locations are listed in Table 2–1 and described in the following text. The addresses of these locations are given in hexadecimal.

Table 2–1. Memory-Mapped Registers

Address	Abbreviation	Name
0h	DRR	Serial-port data receive register
1h	DXR	Serial-port data transmit register
2h	TIM	Timer register
3h	PRD	Period register
4h	IMR	Interrupt mask register
5h	—	Reserved
6h	RTC	Real-time counter
7h	FREQ	Frequency control register
8h	MEMTYPE	External-memory interface register
9h	DI	D-port input register
Ah	DO	D-port output register
Bh	DDIR	D-port direction register
Ch	BI	B-port input register
Dh	BO	B-port output register
Eh	BDIR	B-port direction register
Fh	—	Reserved
10h	ADAC	Sigma-delta ADC/DAC control register
11h	SDAD	Sigma-delta ADC input register
12h	SDDA	Sigma-delta DAC output register
13h	SAAD	Successive-approximation ADC register
14h	—	Reserved
15h	—	Reserved
16h	—	Reserved
17h	—	Reserved

2.1.3.1 DRR — Serial-Port Data Receive Register

The serial-port data receive register (DRR) is a 16-bit read/write register that holds the data received by the serial port. It may be operated in 8-bit byte or 16-bit word mode. When DRR is used as a general-purpose register, CLKR or FSR should be tied to ground.

2.1.3.2 DXR — Serial-Port Data Transmit Register

The serial-port data transmit register (DXR) is a 16-bit read/write register that holds the data to be transmitted by the serial port. It may be operated in 8-bit byte or 16-bit word mode. It also may be used as a general-purpose register.

2.1.3.3 TIM — Timer Register

The timer register (TIM) is a 16-bit read/write register that is initialized by a reset to the maximum value of FFFFh. TIM begins decrementing after $\overline{RS}$ is deasserted and decrements once for every CLKOUT1 period. When the timer interrupt (TINT) is enabled, an interrupt is generated when TIM decrements to zero. TIM is reloaded from period register (PRD) one instruction cycle after TIM decrements to zero. When a new value is written to TIM, it begins decrementing from that value without generating an interrupt.

2.1.3.4 PRD — Period Register

The period register (PRD) is a 16-bit read/write register that is initialized by a reset to the maximum value of FFFFh. One instruction cycle after TIM decrements to zero, TIM is reloaded with the value in PRD. Therefore, interrupts can be programmed to happen at intervals of (PRD + 1) instruction cycles. When a new value is written to PRD, it does not modify the current countdown. A PRD value of 0 is illegal. When PRD is used as a general-purpose register, the TINT interrupt should be disabled (either by clearing the TINT bit of the interrupt mask register (IMR) memory-mapped register or by setting the interrupt mode bit (INTM) of status register ST0).

2.1.3.5 IMR — Interrupt Mask Register

The interrupt mask register (IMR) is a 6-bit read/write register that allows the following interrupts to be masked (listed from highest priority to lowest priority):

Interrupt 0	External user interrupt ($\overline{INT0}$)
Interrupt 1	Sigma-delta interrupt (SDINT)
Interrupt 2	RTC interrupt (CINT)
Interrupt 3	Timer interrupt (TINT)
Interrupt 4	Serial-port receive interrupt (RINT)
Interrupt 5	Serial-port transmit interrupt (XINT)

To disable an interrupt, a logic zero must be written to the appropriate bit position of the IMR (see Table 2–2). Writing a logic one enables the interrupt, provided the INTM bit of status register ST0 is cleared.

Table 2–2. Interrupt Mask Register (IMR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										XINT	RINT	TINT	CINT	SDINT	$\overline{INT0}$

2.1.3.6 RTC — Real-Time Counter Register

The real-time counter register (RTC) is a 16-bit read/write register initialized by a reset to 0000h. The RTC decrements once for every 64 reference oscillator periods after a nonzero value is written to it. The first time a value is written to the RTC after reset, the RTC and its preset latch are both changed. Upon all subsequent writes to the RTC, the new value is written only to the preset latch. When reading from the RTC, the preset latch value is returned. When the RTC decrements to zero, an interrupt is generated if the CINT bit of the IMR memory-mapped register equals one and the INTM bit of status register ST0 equals zero. When the RTC underflows, it is reloaded from the preset latch. This means that an RTC interrupt can be generated once every 64 × (preset latch value + 1) reference oscillator periods. A value should not be written to the RTC more frequently than once every 64 periods of the external crystal. With a 4.096 MHz external crystal, the maximum rate of writing to the RTC is 64 kHz. The RTC provides a fixed time base even if system clocks are changing.

2.1.3.7 FREQ — Frequency Control Register

The frequency control register (FREQ) is a 13-bit read/write register that allows four MSP58C80 clock rates to be adjusted (see Table 2–3 and Table 2–4). These clocks are the PLL clock, the refresh clock, the sigma-delta clock, and the processor clock. The PLL clock and the refresh clock are defined relative to the reference oscillator, which is either provided externally or controlled by an external crystal or ceramic resonator. The sigma-delta clock and the processor clock are defined relative to the PLL clock. Table 2–4 lists the binary setting for each of the clock control parameters along with the value which the binary setting represents. Figure 2–5 provides a block diagram of the MSP58C80 clock system.

Table 2–3. Frequency Control Register (FREQ)

15	14	13	12	11 10 9 8	7 6 5	4 3 2 1 0
1	1	1	Refresh Divider (RD)	Sigma-Delta Predivider (SDPD)	Processor-Clock Predivider (PCPD)	Phase-Locked Loop Frequency Gain (PLLFG)

Table 2–4. Clock Control Settings in the FREQ Memory-Mapped Register

Refresh Divider (RD)	Sigma-Delta Predivider (SDPD)	Processor-Clock Predivider (PCPD)	Phase-Locked Loop Frequency Gain (PLLFG)
0 → 32†	0000 → 2	000 → 1†	00000 → 0.5†
1 → 64	0001 → 4	001 → 2	00001 → 1
	0010 → 6	010 → 4	00010 → 1.5
	0011 → 8	011 → 8	00011 → 2
	0100 → 10	100 → 16	00100 → 2.5
	0101 → 12	101 → 16	00101 → 3
	0110 → 14	110 → 16	00110 → 3.5
	0111 → 16	111 → 16	00111 → 4
	1000 → 18		01000 → 4.5
	1001 → 20†		01001 → 5
	1010 → 22		01010 → 5.5
	1011 → 24		01011 → 6
	1100 → 26		01100 → 6.5
	1101 → 28		01101 → 7
	1110 → 30		01110 → 7.5
	1111 → 32		01111 → 8
			10000 → 8.5
			10001 → 9
			10010 → 9.5
			10011 → 10
			10100 → 10.5
			10101 → 11
			10110 → 11.5
			10111 → 12
			11000 → 12.5
			11001 → 13
			11010 → 13.5
			11011 → 14
			11100 → 14.5
			11101 → 15
			11110 → 15.5
			11111 → 16

† This is the default value after a reset.

The following paragraphs describe the MSP58C80 clocks and how they are affected by settings of the FREQ memory-mapped register.

Reference Oscillator — The reference oscillator (4.096 MHz) can be provided externally or controlled using an external crystal or ceramic resonator. The rate at which the RTC (real-time counter) decrements is determined by the reference oscillator. Therefore, even when the rate of other internal clocks is changed, the rate of CINT (real-time counter interrupt) does not change.

PLL Clock — The PLL clock is the output of the PLL frequency synthesizer and has 32 possible settings, ranging from one-half the reference oscillator to 16 times the reference oscillator. The specific setting of the PLL clock is determined by the PLLFG bits of the FREQ memory-mapped register. The PLL-clock rate can be calculated from the reference oscillator rate as follows:

$$\text{PLL-Clock Rate} = \text{Reference Oscillator Rate} \times \text{PLLFG} \quad (1)$$

With a 4.096-MHz reference oscillator, the PLL clock can vary between 2.048 MHz and 65.536 MHz in steps of 2.048 MHz. The highest PLL-clock setting ensured to work is 65.536 MHz. Therefore, if the reference oscillator is greater than 4.096 MHz, exercise caution selecting a gain with PLLFG bits that result in a PLL-clock rate greater than 65.536 MHz.

When the PLL clock is changed using the FREQ memory-mapped register, the frequency changes exponentially if the LSB of FREQ is toggled (10 ms to change by 99% of the difference between the old and new frequencies). However, when the LSB of FREQ remains unchanged, the frequency undergoes close to a step-function change.

Wait states must be modified so that they are consistent with the PLL-clock setting. An increase in wait state value should precede an increase of the PLL-clock rate, and a decrease in wait-state value should follow a decrease of the PLL-clock rate.

Processor Clock — The processor clock, CLKOUT1 (instruction-cycle clock output), and CLKOUT2 (second instruction-cycle clock output) have 5 speed settings relative to the PLL clock, allowing division by 1, 2, 4, 8, or 16. The specific setting of these clocks is determined by the PCPD bits of the FREQ memory-mapped register. The processor-

clock rate and the CLKOUT1/CLKOUT2 rate can be calculated from the PLL-clock rate as follows:

$$\text{Processor-Clock Rate} = \frac{\text{PLL-Clock Rate}}{\text{PCPD}} \quad (2)$$

$$\text{CLKOUT1, CLKOUT2 Rate} = \frac{\text{Processor-Clock Rate}}{4} \quad (3)$$

A fixed divider is responsible for the division by 4 in the second equation. For a 4.096-MHz reference oscillator, the processor clock can vary between 65.536 MHz (= 65.536 MHz / 1) and 128 kHz (= 2.048 MHz / 16). Since there are four processor clock periods per instruction cycle, the MSP58C80 instruction execution speed can be varied between approximately 16 million instructions per second and 32 thousand instructions per second.

The power consumption of the MSP58C80 is largely determined by the rate of the processor clock. Therefore, reducing the rate of the processor clock can be an effective method of reducing power consumption.

The number of wait states programmed does not need to be changed when the processor-clock predivider is changed because the timing of external memory cycles is based on the PLL clock, not the processor clock. However, when the processor clock is slowed down relative to the PLL clock, fewer instruction cycles are required to perform an external memory access.

Sigma-Delta Clock — The sigma-delta clock (ADCLK) and the sigma-delta sample rate have 16 settings relative to the PLL clock, allowing division by 2, 4, 6, ... , 28, 30, or 32. The specific ADCLK rate and sigma-delta sample rate are determined by the SDPD bits of the FREQ memory-mapped register.

The ADCLK rate and the sigma-delta sample rate can be calculated from the PLL-clock rate as follows:

$$\text{ADCLK Rate} = \frac{\text{PLL-Clock Rate}}{(\text{SDPD} \times 4)} \quad (4)$$

$$\text{Nominal Sigma-Delta Sample Rate} = \frac{\text{PLL-Clock Rate}}{(\text{SDPD} \times 512)} \quad (5)$$

The sigma-delta clock buffering scheme is responsible for the factor of 4 in equation (4) and the factor of 512 in equation (5). The MSP58C80 was designed for use in voice-band products with a 0-kHz to 4-kHz or 0-kHz to 5-kHz range. As a result, the SDPD settings were chosen so that an 8-kHz to 10-kHz sample rate could be obtained at several settings of the PLL clock. In order to obtain an 8-kHz nominal sample rate with a 4.096-MHz reference oscillator, the following settings can be used:

PLLFG = 2, SDPD = 2
PLLFG = 4, SDPD = 4
PLLFG = 6, SDPD = 6
PLLFG = 8, SDPD = 8
PLLFG = 10, SDPD = 10
PLLFG = 12, SDPD = 12
PLLFG = 14, SDPD = 14
PLLFG = 16, SDPD = 16

Note:

See Table 2–4 to determine the binary values that results in the settings listed above.

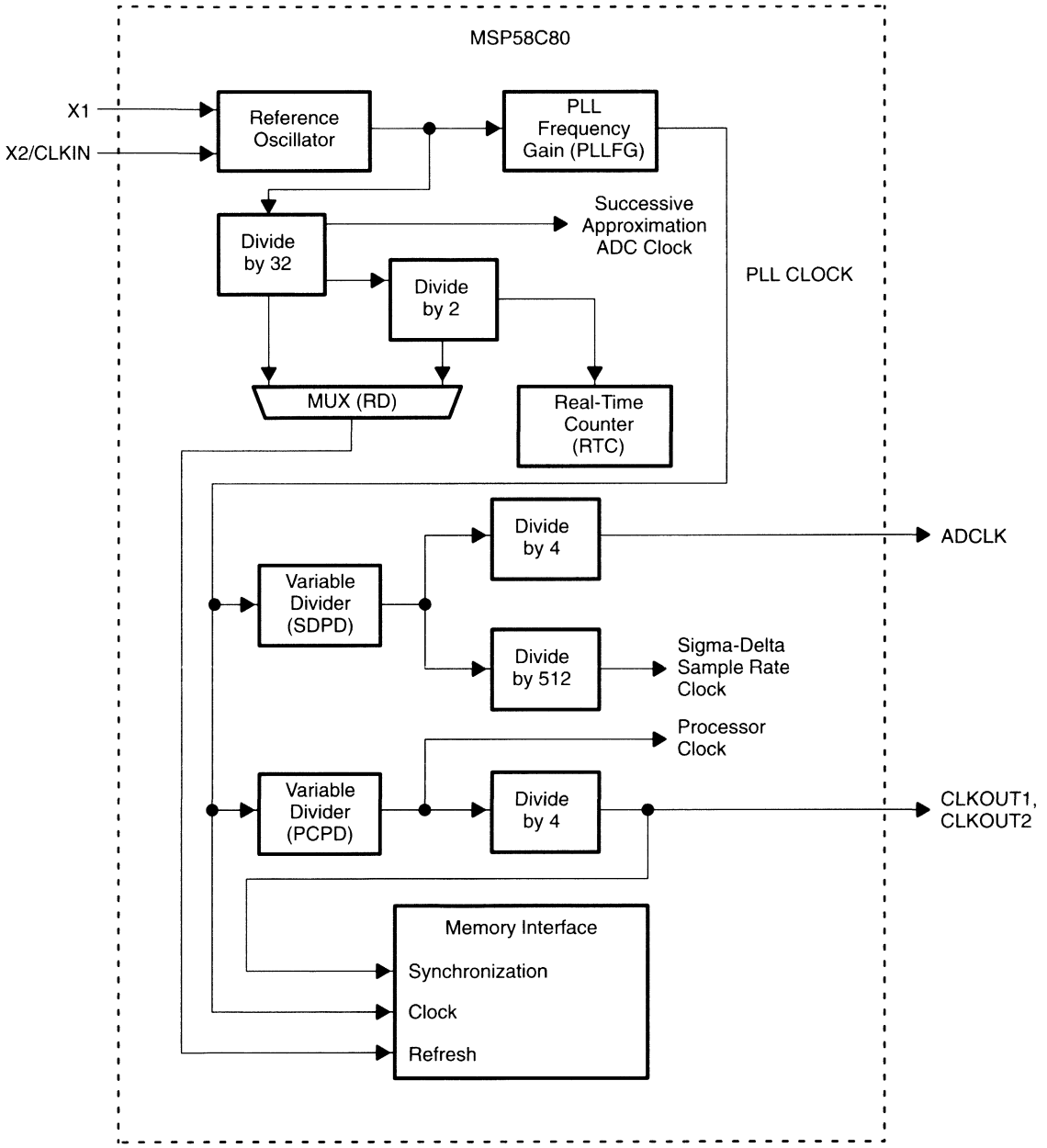
The sigma-delta DAC can either run at the nominal rate or at twice the nominal rate. The sigma-delta ADC can either run at the nominal rate or, when the ADC finite impluse response (FIR) filter is bypassed, at four times the nominal rate. See subsection 2.1.3.15, *ADAC — Sigma-Delta ADC/DAC Control Register*, for the description of the ADAC memory-mapped register and information on determining whether or not the nominal rate is used.

DRAM Refresh Rate — The DRAM refresh rate has 2 settings, allowing division of the reference oscillator by 32 or 64. The specific DRAM refresh rate is determined by the RD bit of the FREQ memory-mapped register and can be calculated as follows:

$$\text{Refresh Rate} = \frac{\text{Reference Oscillator Rate}}{\text{RD}} \qquad (6)$$

For a 4.096-MHz reference oscillator, the DRAM refresh rate can be either 64 kHz (1024 cycle refresh in 16 ms) or 128 kHz (1024 cycle refresh in 8 ms). The 64-kHz refresh rate can be used to meet the refresh requirements of some commercially available DRAM chips, and the 128-kHz rate can be used to meet the refresh requirements of some commercially available ARAM chips.

Figure 2–5. Clock Generation and Distribution Block Diagram



2.1.3.8 MEMTYPE — External-Memory Interface Register

The external-memory interface register (MEMTYPE) is a 16-bit read/write register that specifies the following settings for each of the external-memory spaces:

- ❑ Standard (RAM/ROM) or dynamic (DRAM) memory
- ❑ Wide or narrow data format
- ❑ Number of wait states

MEMTYPE also contains a global external-memory-refresh enable bit (see Table 2–5).

Table 2–5. External-Memory Interface Register (MEMTYPE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RF	D/S	N/W	NWS			D/S	N/W	NWS			D/S	N/W	NWS		
Global	Alternate Data Space					Data Space					Program and I/O Spaces				

RF: Global refresh enable, 1 = enable, 0 = disable
D/S: Dynamic = 1, Standard = 0
N/W: Narrow = 1, Wide = 0
NWS: Number of wait states, ranging from 0 (000) through 7 (111)

Depending on the type of external memory that is used, between 16 and 32 terminals are available for general I/O. When narrow dynamic memory is used, the DD0 signal is used for the memory interface and all of the B-port and D-port signals can be used for general I/O. When wide dynamic memory or narrow standard memory is used, the LSBs of the D port are used for interfacing with memory and are not available for general I/O. When wide standard memory is used, the entire D port is used for interfacing with memory and is not available for general I/O (see Table 2–6). Those signals that are used for interfacing with memory must be programmed as inputs with the DDIR memory-mapped register (see subsection 2.1.3.11, *DDIR — D-Port Direction Register*).

Table 2–6. Terminal Usage for Different External Memory Types

External Memory Type	D-Port Terminals Required for Memory Interface	Number of Terminals Available for General I/O
Dynamic, Narrow (1 bit)	None	32
Dynamic, Wide (4 bits)	D0 – D3	28
Standard, Narrow (8 bits)	D0 – D7	24
Standard, Wide (16 bits)	D0 – D15	16

All external accesses are affected by the corresponding number of wait states (NWS) parameter. Refreshes are also affected by wait states: the NWS for refreshes is equal to the NWS setting for the alternate space. See Table 2–7 to determine the time required to perform read, write, and refresh operations.

Table 2–7. Time Required to Perform Read, Write, and Refresh Operations

External Memory Type	Operation	External Space	Time Required (In Instruction Cycles)†
Standard, Wide	Read/Write	Data, Program, and I/O	(NWS+1)/PCPD
Standard, Wide	Read/Write	Alternate	(NWS+2)/PCPD
Standard, Narrow	Read/Write	Data, Program, and I/O	(2×NWS+2)/PCPD
Standard, Narrow	Read/Write	Alternate	(2×NWS+3)/PCPD
Dynamic, Wide	Read/Write	Any	(6×NWS+6)/PCPD
Dynamic, Narrow	Read/Write	Any	(18×NWS+18)/PCPD
Dynamic	Refresh	Any	(3×NWS+3)/PCPD

† For write operations, add one instruction cycle to the time required. To determine the read or write time when refresh is enabled, add twice the refresh time to the read or write time. When refresh is enabled, this adjustment must be made even if the read or write operation involves standard memory. To determine time in seconds, multiply the time in instruction cycles by $4 \times \frac{\text{PCPD}}{\text{PLL Clock Rate}}$.

2.1.3.9 DI — D-Port Input Register

The D-port input register (DI) is a 16-bit read-only register that contains data being read from the D-port lines.

2.1.3.10 DO — D-Port Output Register

The D-port output register (DO) is a 16-bit read/write register. This register contains data being written to the D-port lines that are configured as outputs.

Note:

All D-port lines default to input on reset.

2.1.3.11 DDIR — D-Port Direction Register

The D-port direction register (DDIR) is a 16-bit read/write register that allows individual lines of the D-port to be specified as either input (logic 0) or output (logic 1) lines. Those signals being used to interface with memory must be programmed as inputs (logic 0).

2.1.3.12 BI — B-Port Input Register

The B-port input register (BI) is a 16-bit read-only register that contains data being read from the B-port lines.

2.1.3.13 BO — B-Port Output Register

The B-port output register (BO) is a 16-bit read/write register. This register contains data being written to B-port lines that are configured as outputs. When the B0 (zero) signal is configured as an output, the value of the LSB of the BO register is used to determine whether or not to branch when a BIOZ (branch on I/O status = 0) instruction is executed (see subsection 2.1.3.14, *BDIR — B-Port Direction Register*). When the B15 signal is configured as an output, its value is the logical OR of the XF bit of status register ST1 and the MSB of the BO register. Since the XF bit defaults to a logic high on reset, the XF bit must be cleared if you desire the MSB of the BO register to determine the output value of the B15 signal. On the other hand, since the MSB of the BO register does not have a defined value upon reset, the MSB of the BO register must be cleared if you desire the XF bit to determine the output value of the B15 signal.

Note:

All B-port lines default to input on reset.

2.1.3.14 BDIR — B-Port Direction Register

The B-port direction register is a 16-bit read/write register that allows individual lines of the B port to be specified as either input (logic 0) or output (logic 1) lines. Programming the B0 signal as an input allows it to be used the same as the $\overline{\text{BIO}}$ signal of the TMS320C25. However, if the B0 signal is programmed as an output and a BIOZ instruction is executed, the value of the LSB of the BO register determines whether or not the branch is taken. The B15 signal can be used the same as the XF signal of the TMS320C25 by programming it as an output and clearing the MSB of the BO register.

2.1.3.15 ADAC — Sigma-Delta ADC/DAC Control Register

The sigma-delta ADC/DAC control register is a 6-bit register that:

- Allows the sigma-delta ADC and DAC blocks on the MSP58C80 to be placed in a low-power state.
- Indicates the status of ADC and DAC processing.
- Allows the ADC FIR filter to be bypassed.
- Sets the DAC sample rate to be the nominal rate or twice the nominal rate.

By default, the sigma-delta ADC and the sigma-delta DAC run at the same rate. By setting the ADC FIR disable bit in the ADAC memory-mapped register, the FIR filter is bypassed and ADC samples are written to the sigma-delta ADC (SDAD) memory-mapped register at four times the nominal rate. In this case,

a software filter can be used in place of the hardware FIR filter. By setting the DAC rate bit in the ADAC memory-mapped register, the DAC samples in the sigma-delta DAC (SDDA) memory-mapped register are read and converted at twice the nominal rate. In this case, the DAC filter high-frequency cutoff is twice the nominal cutoff and stop-band noise increases.

An SDINT interrupt is generated whenever an ADC conversion or a DAC conversion (or both) has been completed. The ADC status bit in the ADAC register is set when an ADC conversion has been completed, and the DAC status bit is set when a DAC conversion has been completed.

The ADC and DAC status bits are both cleared by a reset. In addition, the ADC status bit is cleared by reading the SDAD memory-mapped register or by disabling analog-to-digital conversion (by clearing bit 0 of the ADAC memory-mapped register). The DAC status bit is cleared by writing to the SDDA memory-mapped register or by disabling digital-to-analog conversion (by clearing bit 1 of the ADAC memory-mapped register).

Table 2–8. Sigma-Delta ADC/DAC Control Register (ADAC)

Bit	Name	Control Description
0	ADC Enable (Read/Write)	Logic 1 enables analog-to-digital conversion
1	DAC Enable (Read/Write)	Logic 1 enables digital-to-analog conversion
2	ADC Status (Read only)	Logic 1 indicates a valid ADC sample has been buffered
3	DAC Status (Read only)	Logic 1 indicates a valid DAC buffer has been read by the DAC
4	ADC FIR Disable (Read/Write)	Logic 1 disables the ADC FIR low-pass filter, allowing the hardware filter to be replaced by a software filter. When the FIR filter is disabled, samples are written to the SDAD buffer at four times the nominal rate.
5	DAC Rate (Read/Write)	Logic 1 causes the DAC conversion to be performed at twice the nominal DAC sample rate.

2.1.3.16 SDAD — Sigma-Delta ADC Input Register

The sigma-delta ADC input register (SDAD) is a 16-bit read-only register that contains the input value read by the sigma-delta ADC.

2.1.3.17 SDDA — Sigma-Delta DAC Output Register

The sigma-delta DAC output register (SDDA) is a 16-bit read/write register that contains the output value to be written to the sigma-delta DAC.

2.1.3.18 SAAD — Successive-Approximation ADC Register

The successive-approximation ADC register (SAAD) is a 10-bit register that contains the result of the successive-approximation ADC conversion. It also contains control bits to select which analog input is used.

Table 2–9. Successive-Approximation ADC Register (SAAD)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	MX1 (Read/Write)	MX0 (Read/Write)	ADC Value (Read Only)							

Table 2–10. Successive-Approximation ADC Input Selection

MX1	MX0	Result
0	0	ADM0 selected
0	1	ADM1 selected
1	0	ADM2 selected
1	1	No channel selected

The successive-approximation ADC initiates a conversion immediately following a read from the SAAD register. When the MSP58C80 is reset, the SAAD register contains the value FF00h, which indicates that no analog input channel is selected. Before reading the first data value after the reset, the following sequence must happen.

1) Select a MUX channel.
2) Wait at least one successive-approximation ADC clock cycle (see Figure 2-5).
3) Prime the SAAD register with a dummy read.

To ensure data integrity of successive-approximation ADC values, all of the following conditions must be met.

- ❑ Allow eight successive-approximation ADC clock cycles between reads of the SAAD register.
- ❑ Allow one successive-approximation ADC clock cycle between selecting a MUX input and reading the SAAD register. This delay allows the input to stabilize before the conversion is initiated.
- ❑ Allow two successive-approximation ADC clock cycles between reading the SAAD register and changing the MUX input. This delay prevents the input from changing before it is sampled.

With a 4.096-MHz external crystal, the successive-approximation ADC clock rate is 128 kHz. Therefore, reads from the SAAD can be performed at 16 kHz.

2.1.4 I/O-Mapped Registers

The MSP58C80 can address 16 locations in the I/O space using the IN and OUT instructions. The lower eight addresses are external I/O expansion registers, and the upper eight addresses are internal I/O-mapped registers and reserved registers. If the MSP58C80 is performing a prefetch-read or a postwrite and an internal I/O-mapped register is read from or written to, program execution is halted until the prefetch-read or postwrite is finished. The I/O registers are numbered in Table 2–11 from 0h through Fh, with the number of each register indicating its I/O address.

Table 2–11. I/O-Mapped Registers

Address	Abbreviation	Name
0h – 7h	—	External I/O expansion registers
8h	ADB	Alternate data buffer for read without prefetch
9h	AAL	Alternate address low register
Ah	AAH	Alternate address high register
Bh	—	Reserved
Ch	ADBP	Alternate data buffer for prefetch-read/ postwrite
Dh	AALP	Alternate address low register for prefetch-read
Eh – Fh	—	Reserved

2.1.4.1 ADB — Alternate Data Buffer for Read Without Prefetch

The alternate data buffer for read without prefetch (ADB) is a 16-bit register. Performing an IN from this location reads a value that has been fetched from the alternate data space by a previous prefetch-read. Using this location does not initiate a new prefetch-read.

2.1.4.2 AAL — Alternate Address Low Register

The alternate address low register (AAL) is a 16-bit register. Performing an OUT to this location writes to the lower 16 bits of the 24-bit alternate address register . Since writing to the AAL does not initiate a prefetch-read, this register is generally used when preparing for a postwrite operation.

2.1.4.3 AAH — Alternate Address High Register

The alternate address high register (AAH) is an 8-bit register. Performing an OUT to this location writes to the upper 8 bits of the 24-bit alternate address register.

2.1.4.4 ADBP — Alternate Data Buffer for Prefetch-Read/Postwrite

The alternate data buffer for prefetch-read/postwrite (ADBP) is a 16-bit register. Performing an IN from this location reads a value that has been fetched from the alternate data space by a previous prefetch-read and also initiates a new prefetch-read. Performing an OUT to this location initiates a postwrite.

2.1.4.5 AALP — Alternate Address Low Register for Prefetch-Read

The alternate address low register for prefetch-read (AALP) is a 16-bit register. Performing an OUT to this location writes to the lower 16 bits of the 24-bit alternate address register and also initiates a prefetch-read.

2.1.5 Addressing the Alternate Data Space

The MSP58C80 can address 16M words of alternate data memory. Prefetch-read from and postwrite to alternate data memory are done by way of internal I/O-mapped registers. The procedures for performing these operations are described in the following subsections.

2.1.5.1 Prefetch-Read From Alternate Data Space

The following procedure performs a prefetch-read from an alternate data space (see Example 2-1).

1) Perform an OUT instruction with the AAH I/O-mapped register to load the upper 8 bits of the 24-bit alternate address register. When there is a prior prefetch-read or postwrite in progress, the MSP58C80 completes the prior operation before continuing with the OUT instruction.

2) Perform an OUT instruction with the AALP I/O-mapped register to load the lower 16 bits of the 24-bit alternate address register. The MSP58C80 initiates a prefetch and then increment the alternate address register.

3) Perform an IN instruction with the ADBP I/O-mapped register. The MSP58C80 provides the data immediately after the previous prefetch has been completed. Otherwise, the MSP58C80 waits for the previous prefetch to complete before providing the data. In either case, the MSP58C80 initiates a new prefetch and then increments the alternate address register. This step can be repeated to get more data from consecutive locations. To take advantage of the prefetch-read capability, the IN instructions must be spaced far enough apart so that one prefetch-read has time to finish before the next is initiated. Also, the instructions between the IN instructions must be located in the internal program space and must address only internal locations.

4) Perform an IN instruction with the ADB I/O-mapped register to read the last value in the sequence.

Example 2–1. Prefetch-Read From Alternate Data Space

```
LDPK page        ; Set up data memory page pointer
LACK value1      ; [ACC] = upper portion of alternate space address
SACL upper       ; ACC value stored in internal location "upper"
LALK value2      ; [ACC] = lower portion of alternate space address
SACL lower       ; ACC value stored in internal location "lower"
LARP AR0         ; Select auxiliary register 0
LRLK AR0,data    ; [AR0] = internal address where data is stored
OUT upper,aah    ; Set up alternate address register
OUT lower,aalp   ; and initiate a prefetch-read

; If you want to read only one value from the alternate space,
; the "IN *+,adbp" instructions can be eliminated. If you want
; to take advantage of the prefetch-read, all of the instructions that
; access the aalp, adbp, and adb registers need to be separated by instructions
; located in the internal program space that do not access any of the
; external spaces.
IN *+,adbp       ; Read prefetched value and initiate next prefetch
.
.
.
IN *+,adbp       ; Read prefetched value and initiate next prefetch
IN *+,adb        ; Read final value without initiating another prefetch
```

2.1.5.2 Postwrite to Alternate Data Space

The following procedure performs a postwrite to an alternate data space (see Example 2-2).

1) Perform an OUT instruction with the AAH I/O-mapped register to load the upper 8 bits of the 24-bit alternate address register. When there is a prior prefetch-read or postwrite in progress, the MSP58C80 completes the prior operation before continuing with the OUT instruction.

2) Perform an OUT instruction with the AAL I/O-mapped register to load the lower 16 bits of the 24-bit alternate address register.

3) Perform an OUT instruction with the ADBP I/O-mapped register. The MSP58C80 begins a postwrite and immediately continues program execution unless a prior postwrite has not been completed. In that case, the MSP58C80 halts program execution until it is completed, then begins

the new postwrite and continues with program execution. In either case, the MSP58C80 increments the alternate address register after completing the postwrite. This step can be repeated to write more data to consecutive locations. To take advantage of the postwrite capability, the OUT instructions must be spaced far enough apart so that one postwrite has time to finish before the next one is initiated, and the instructions between the OUT instructions must be located in the internal program space and address only internal locations.

Example 2–2. Postwrite to Alternate Data Space

```
LDPK page        ; Set up data memory page pointer
LACK value1      ; [ACC] = upper portion of alternate space address
SACL upper       ; ACC value stored in internal location "upper"
LALK value2      ; [ACC] = lower portion of alternate space address
SACL lower       ; ACC value stored in internal location "lower"
LARP AR0         ; Select auxiliary register 0
LRLK AR0,data    ; [AR0] = internal address where data is stored
OUT upper,aah    ; Set up alternate address
OUT lower,aal    ; register

; If you want to take advantage of the postwrite, all of the following
; OUT instructions need to be separated by instructions located in the
; internal program space that do not access any of the external spaces.

OUT *+,adbp      ; Postwrite to alternate space
.
.
.
OUT *+,adbp      ; Postwrite to alternate space
```

2.1.6 Interfacing with Standard Memory

The following signals are used for controlling standard memory accesses:

- R/$\overline{W}$ controls the direction of data transfer.
- $\overline{STRB}$ provides timing for the data transfer.
- $\overline{ALATCH}$ doubles as a demultiplexing latch signal for 24-bit multiplexed addresses and as a time-multiplexing signal to select the lower or upper byte when interfacing to byte-wide memories.
- $\overline{AS}$, $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ indicate which of the external spaces is being accessed.

The external memory interface can be configured to work with both wide (16-bit) and narrow (8-bit) standard memory. For wide memory, a single access is performed for each 16-bit word, but for narrow memory a lower-byte access and an upper-byte access are performed for each 16-bit word. Table 2–12 shows how the address bus is used for standard memory cycles.

Table 2–12. Standard Memory Address Bus Bit Assignments†

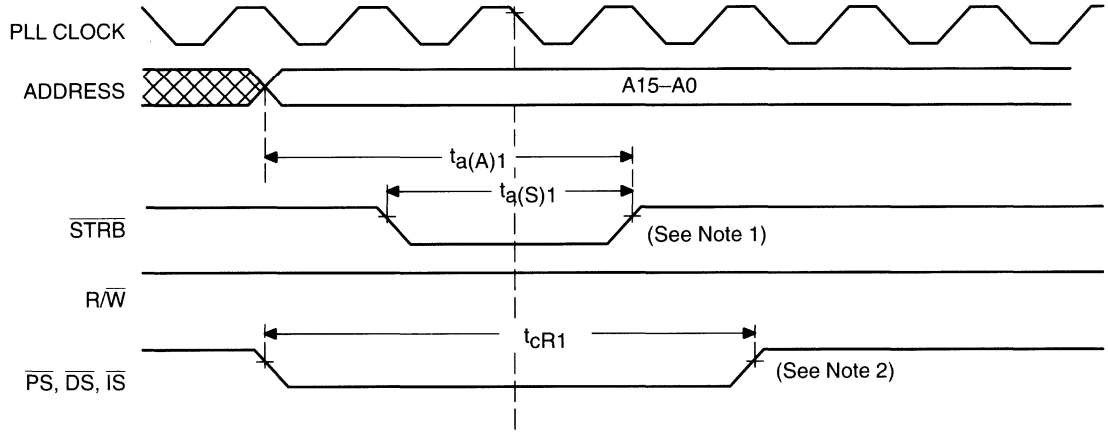
External Address Lines on Address Bus	Internal Address Bits for Program, Data, or I/O Space	Internal Address Bits for Alternate Data Space	
	Transfer Data Word (Wide Memory) or Data Bytes (Narrow Memory)	Latch High-Order Bits	Transfer Data Word (Wide Memory) or Data Bytes (Narrow Memory)
$\overline{\text{ALATCH}}$‡	c0	—	c0
A0	a0	a0	a0
A1	a1	a1	a1
A2	a2	a2	a2
A3	a3	a3	a3
A4	a4	a4	a4
A5	a5	a5	a5
A6	a6	a6	a6
A7	a7	a7	a7
A8	a8	a16	a8
A9	a9	a17	a9
A10	a10	a18	a10
A11	a11	a19	a11
A12	a12	a20	a12
A13	a13	a21	a13
A14	a14	a22	a14
A15	a15	a23	a15

† — = undetermined
Ax = external address line
ax = internal address bit (generated by software)
cx = internal byte counter (generated by the MSP58C80 memory interface logic)
‡ $\overline{\text{ALATCH}}$ is only used as an address line for narrow-memory data transfers.

If standard wide memory is being used, none of the 16 D-port signals are available for use as general I/O signals. However, if standard narrow memory is being used, the eight most significant D-port signals are available for use as general I/O signals. The signals used for memory interfacing must be programmed as inputs by appropriately loading the DDIR memory-mapped register (see subsection 2.1.3.11, *DDIR — D-Port Direction Register*).

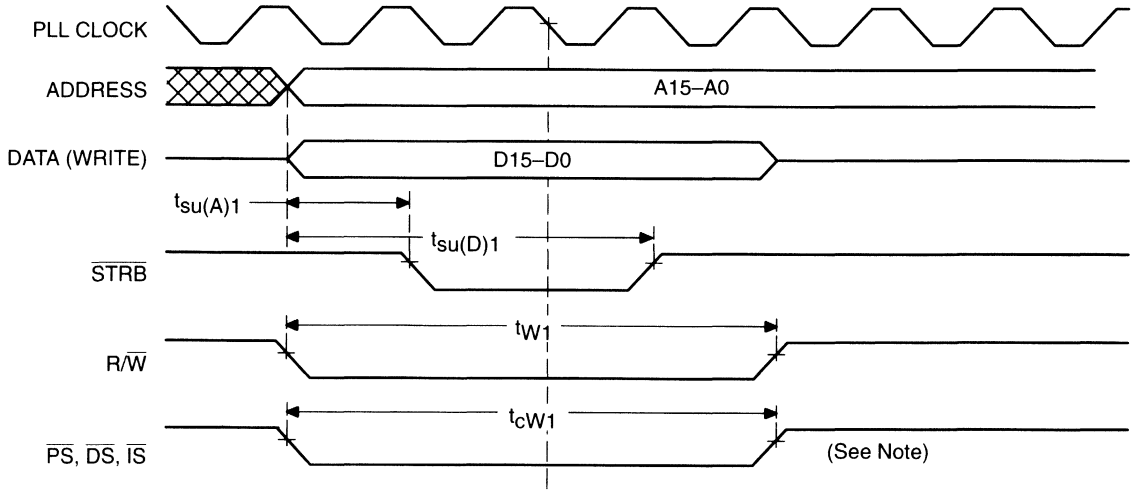
For standard wide memory in the program, data, or I/O space, the address bus lines carry the internal 16-bit address with no need for a multiplexing signal (see Figure 2–6 and Figure 2–7). The $\overline{\text{STRB}}$ signal going low indicates the address is valid, and the $\overline{\text{STRB}}$ signal going high indicates the read or write operation can be performed (i.e., data valid).

Figure 2–6. Standard Wide-Memory Read Cycle (Program, Data, I/O Space)



Notes 1) Data is latched on the rising edge of $\overline{\text{STRB}}$.
2) t_{cR1} (read cycle time) = 4 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical line). R/$\overline{\text{W}}$ is high during entire read cycle.

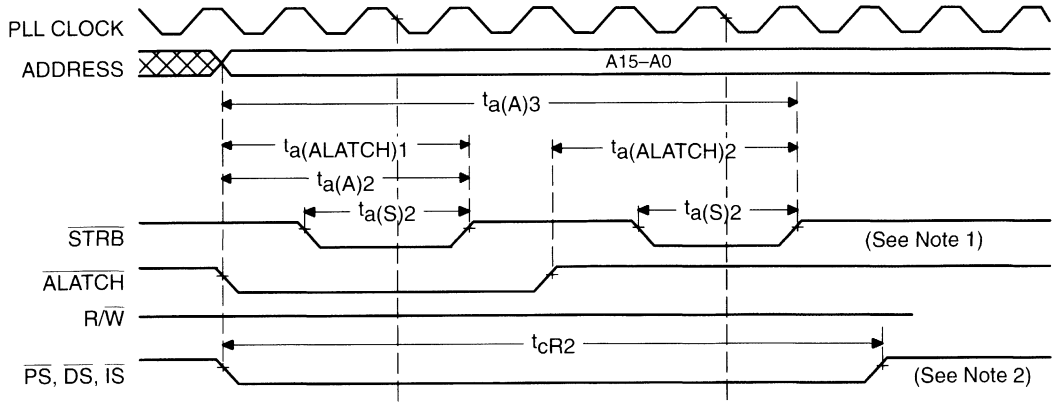
Figure 2–7. Standard Wide-Memory Write Cycle (Program, Data, I/O Space)



Note t_{cW1} (write cycle time) = 4 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical line).

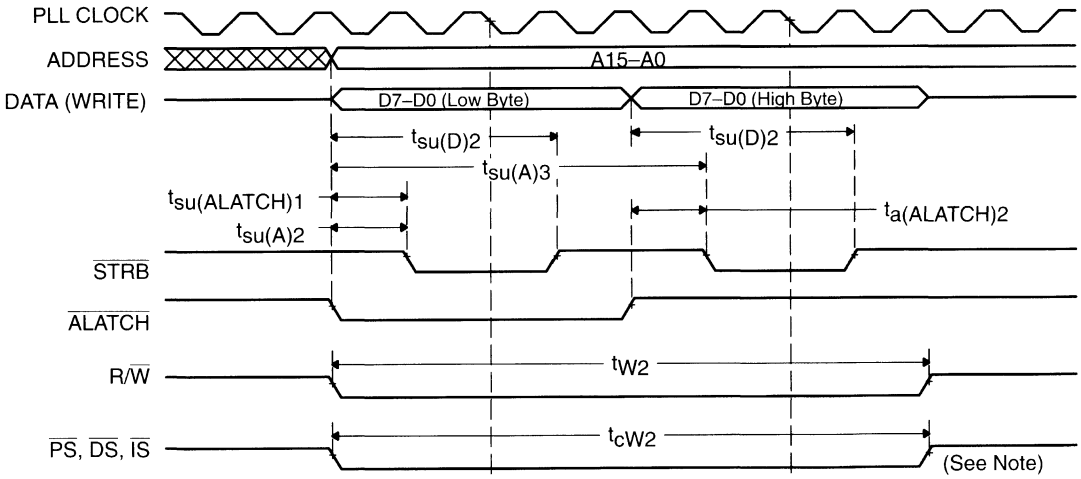
For standard narrow memory in the program, data, or I/O space, the address bus lines carry the internal 16-bit address, with $\overline{\text{ALATCH}}$ providing the LSbyte for the physical memory devices to multiplex the upper and lower bytes of a given 16-bit word (see Figure 2–8 and Figure 2–9). When $\overline{\text{ALATCH}}$ is low, the LSbyte of the internal data word is transferred. When $\overline{\text{ALATCH}}$ is high the MSbyte of the internal data word is transferred. $\overline{\text{STRB}}$ pulses low and high once while $\overline{\text{ALATCH}}$ is low and once after $\overline{\text{ALATCH}}$ goes high. In this way, $\overline{\text{STRB}}$ supplies the timing for two read or write operations, indicating that the address is valid when it goes low and indicating that the read or write operation can be performed when it goes high.

Figure 2–8. Standard Narrow-Memory Read Cycle (Program, Data, I/O Space)



Notes 1) Data is latched on the rising edge of $\overline{\text{STRB}}$.
2) t_{cR2} (read cycle time) = 8 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines). R/$\overline{\text{W}}$ is high during entire read cycle.

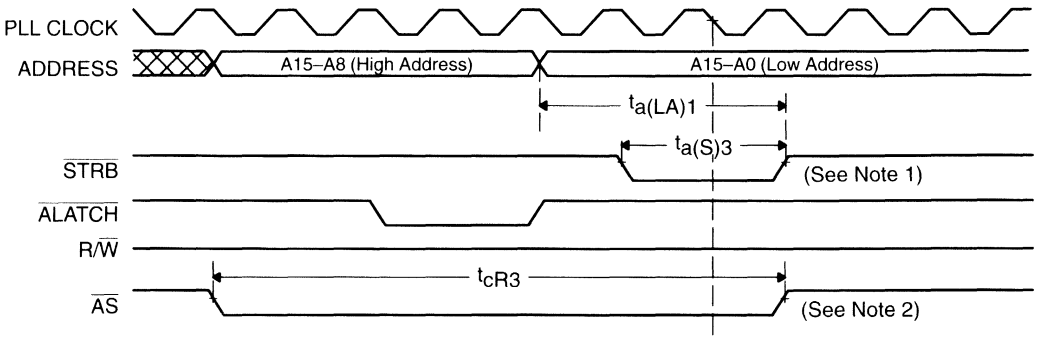
Figure 2–9. Standard Narrow-Memory Write Cycle (Program, Data, I/O Space)



Note: t_{cW2} (write cycle time) = 8 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines)

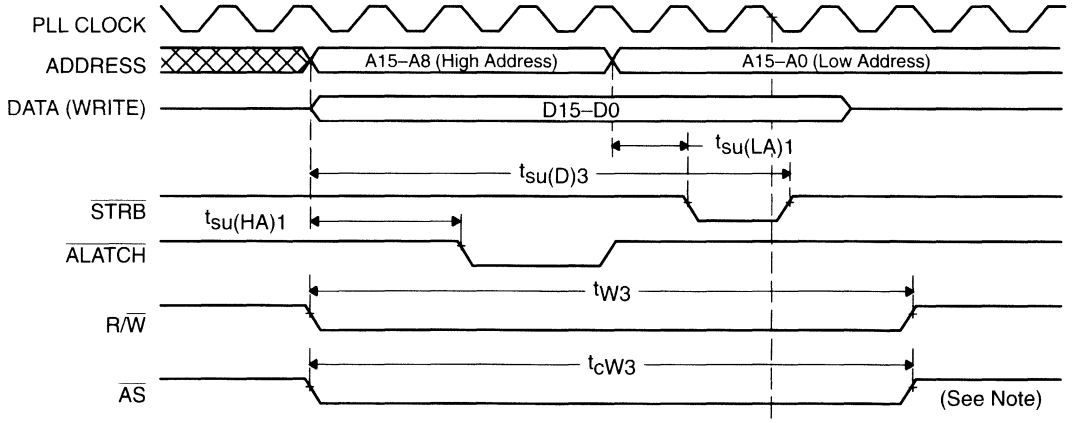
For standard wide memory in the alternate data space, $\overline{\text{ALATCH}}$ goes low when internal address bits a16 – a23 need to be latched from address bus lines A8–A15 (see Figure 2–10 and Figure 2–11). $\overline{\text{STRB}}$ then goes low to indicate that internal address bits a0–a15 are available on address bus lines A0–A15 and goes high to indicate that the read or write operation can be performed.

Figure 2–10. Standard Wide-Memory Read Cycle (Alternate Data Space)



Notes 1) Data is latched on the rising edge of $\overline{\text{STRB}}$.
2) t_{cR3} (read cycle time) = 8 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical line). R/$\overline{\text{W}}$ is high during entire read cycle.

Figure 2–11. Standard Wide-Memory Write Cycle (Alternate Data Space)

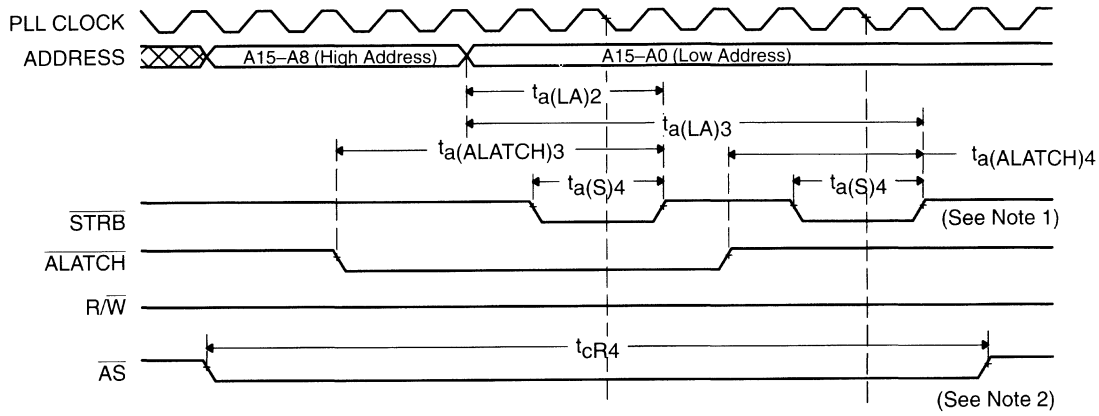


Note: t_{cW3} (write cycle time) = 8 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical line).

For standard narrow memory in the alternate data space, $\overline{\text{ALATCH}}$ goes low when internal address bits a16–a23 need to be latched from address bus lines A8–A15 (see Figure 2–12 and Figure 2–13). $\overline{\text{STRB}}$ pulses low and high once while $\overline{\text{ALATCH}}$ is still low and once after $\overline{\text{ALATCH}}$ goes high. In this way, $\overline{\text{STRB}}$ supplies the timing for two read or write operations, indicating that address bits

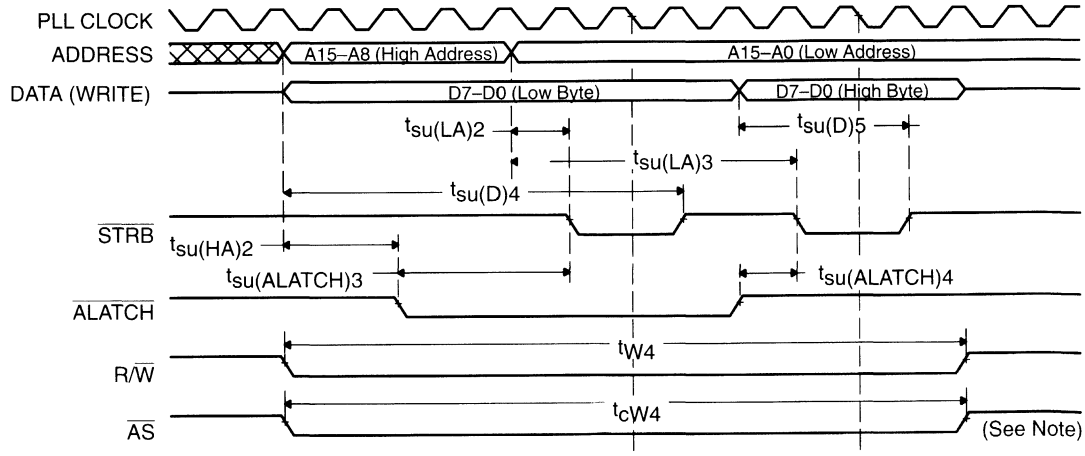
a0–a15 are available on address bus lines A0 – A15 when STRB goes low and indicating that the read or write operation can be performed when it goes high. In addition, $\overline{\text{ALATCH}}$ provides the LSbyte for the physical memory devices to multiplex the upper and lower bytes of a given 16-bit word. When $\overline{\text{ALATCH}}$ is low, the LSbyte of the internal data word is transferred. When $\overline{\text{ALATCH}}$ is high the MSbyte of the internal data word is transferred.

Figure 2–12. Standard Narrow-Memory Read Cycle (Alternate Data Space)



Notes 1) Data is latched on the rising edge of $\overline{\text{STRB}}$.
2) t_{cR4} (read cycle time) = 12 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines)

Figure 2–13. Standard Narrow-Memory Write Cycle (Alternate Data Space)



Note: t_{cW4} (write cycle time) = 12 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines).

The time base for standard memory operations is based on the PLL clock and, in the case of write operations is synchronized to the instruction cycle. For both wide and narrow standard memory, the number of wait states controls the length of time that the $\overline{STRB}$ signal is held low for a given access. For a complete discussion of memory access time (with and without wait states), see the discussion of the MEMTYPE memory-mapped register in subsection 2.1.3.8, *MEMTYPE — External-Memory Interface Register.*

2.1.7 Interfacing with DRAM

The following signals are used for controlling DRAM accesses:

- R/$\overline{W}$ controls the direction of data transfer.
- $\overline{RAS}$ and $\overline{CAS}$ provide the row and column address strobes and the refresh signal.
- $\overline{AS}$, $\overline{PS}$, $\overline{DS}$, and $\overline{IS}$ indicate which of the external spaces is being accessed.

Table 2–13 shows the internal address source is multiplexed to the external memory in two different sequences depending on whether narrow (1-bit) or wide (4-bit) DRAMs are being used. The bit counter for narrow DRAM causes the least significant bits of an internal data word to be mapped to lower external addresses and the most significant bits to be mapped to higher external addresses. Similarly, the nibble counter for wide DRAM causes the least significant nibbles of an internal data word to be mapped to lower external addresses and the most significant nibbles to be mapped to higher external addresses.

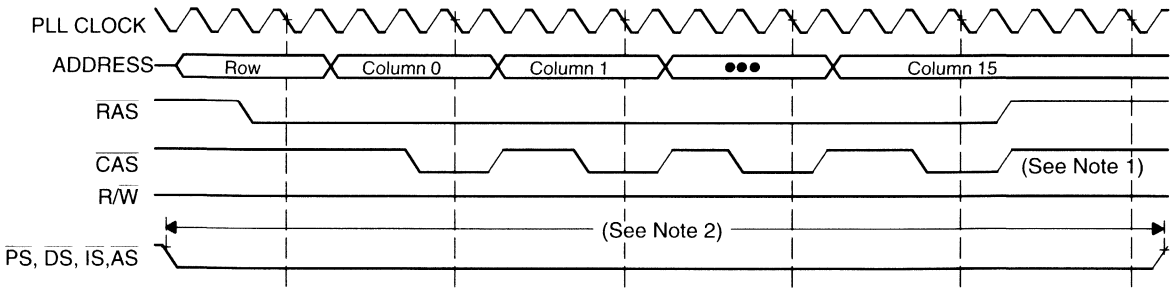
Table 2–13. DRAM Address-Bus Bit Assignments†

External Address Lines on Address Bus	Internal Address Bits for 1-Bit DRAM		Internal Address Bits for 4-Bit DRAM	
	Row Address	Column Address	Row Address	Column Address
A0	a5	c0	—	—
A1	a6	c1	a6	a5
A2	a7	c2	a7	c0
A3	a8	c3	a8	c1
A4	a9	a0	a9	a0
A5	a10	a1	a10	a1
A6	a11	a2	a11	a2
A7	a12	a3	a12	a3
A8	a13	a4	a13	a4
A9	a15	a14	a15	a14
A10	a17	a16	a17	a16
A11	a19	a18	a19	a18
A12	a21	a20	a21	a20
A13	a23	a22	a23	a22
A14	0	0	0	0
A15	0	0	0	0

† — = undetermined
cx = internal nibble or bit counter (generated by MSP58C80 memory interface logic)
ax = internal address bit (generated by software)
Ax = external address line

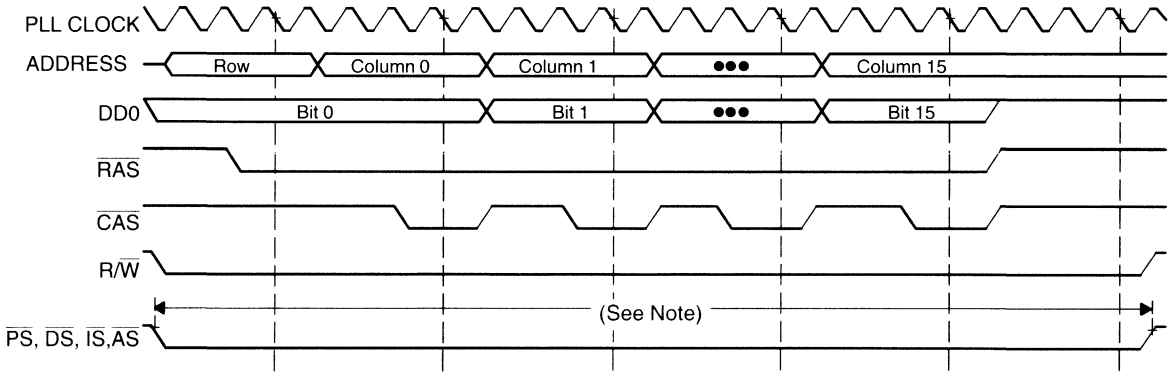
For narrow DRAMs, only data signal DD0 is used for data transmission, leaving all 16 D-port terminals available for use as general I/O signals (see Figure 2–14 and Figure 2–15). When narrow DRAMs are used in the data or program space, either the 9 or 10 LSbytes of the address bus should be connected to the DRAMs, permitting a DRAM space of either 256K x 1 or 1M x 1. When narrow DRAMs are used in the alternate data space, between 9 and 14 LSbytes of the address bus should be connected to the DRAMs, permitting a DRAM space of between 256K x 1 and 256M x 1.

Figure 2–14. 16-Bit DRAM Memory Read Cycle Using 1-Bit Wide Memory



Notes 1) Data is latched on the rising edge of $\overline{CAS}$.
2) Read cycle time = 72 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines). R/$\overline{W}$ is high during entire read cycle.

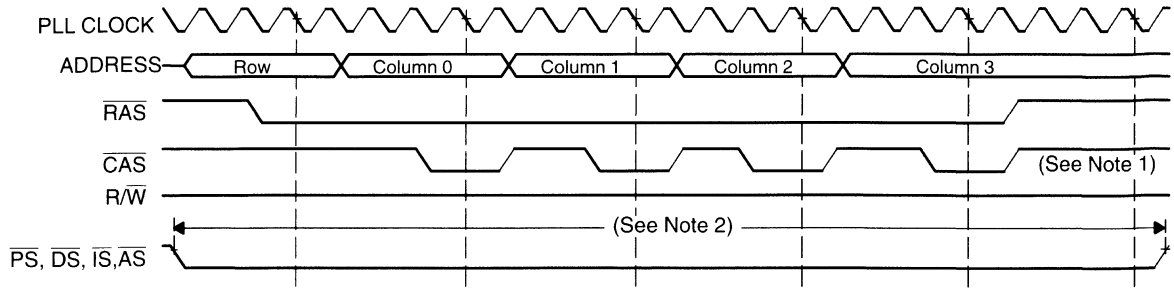
Figure 2–15. 16-Bit DRAM Memory Write Cycle Using 1-Bit Wide Memory



Note: Write cycle time = 72 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines).

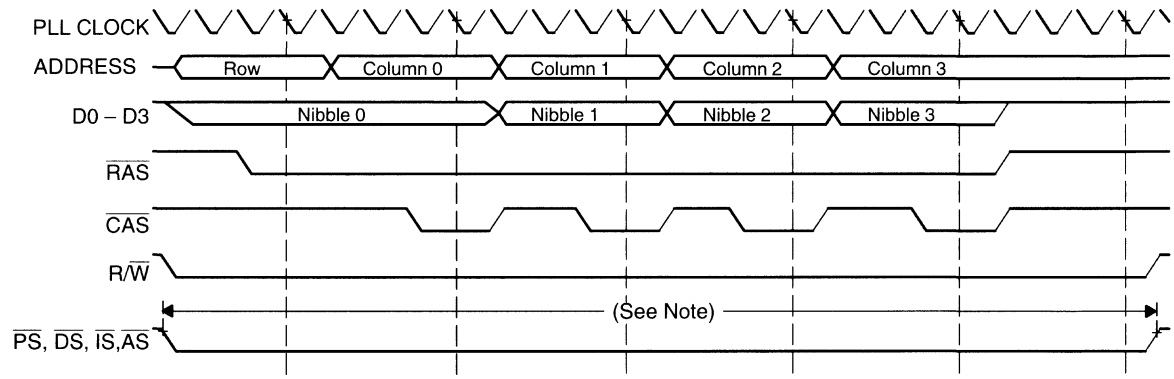
For wide DRAMs, data signals D0–D3 are used for data transmission, leaving the 12 high-order D-port terminals available for use as general I/O signals (see Figure 2–16 and Figure 2–17). The programmer must specify the signals being used for memory interfacing as inputs by appropriately setting the DDIR memory-mapped register (see subsection 2.1.3.11, *DDIR — D-Port Direction Register*). When wide DRAMs are used in the data or program space, address line A0 should be ignored and address lines A1–A8 or A1–A9 should be connected to the DRAMs, permitting a DRAM space of either 64K x 4 or 256K x 4. When wide DRAMs are used in the alternate data space, address line A0 should be ignored and between 8 and 13 of the remaining LSBs of the address bus should be connected to the DRAMs. This allows a DRAM space of between 64K x 4 and 64M x 4.

Figure 2–16. 16-Bit DRAM Memory Read Cycle Using 4-Bit Wide Memory



Notes 1) Data is latched on the rising edge of CAS.
2) Read cycle time = 24 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines). R/W is high during entire read cycle.

Figure 2–17. 16-Bit DRAM Memory Write Cycle Using 4-Bit Wide Memory

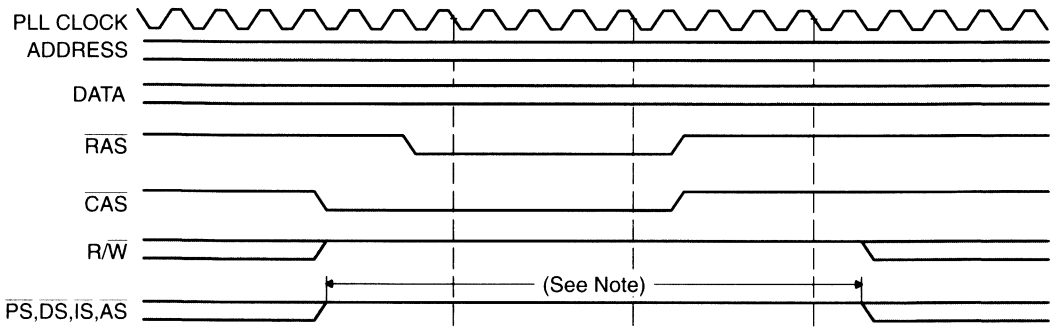


Note: Write cycle time = 24 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines).

The time base for DRAM operations is based on the PLL clock and, in the case of write operations is synchronized to the instruction cycle. DRAM cycles in both the wide and narrow modes utilize the enhanced-page-mode feature of commercially available DRAMs. This feature permits a significant reduction in the time it takes to read 16 successive bits from 1-bit DRAMs or 4 successive nibbles from 4-bit DRAMs.

A DRAM refresh cycle has priority over any other external memory operation but does not interrupt an active acquisition, waiting until the completion of the operation before performing the refresh cycle. The DRAM refresh cycle is CAS-before-RAS, eliminating the need for a refresh counter (see Figure 2–18). The number of refresh-cycle wait states is equal to the number of wait states selected for the alternate data space in the MEMTYPE memory-mapped register (see subsection 2.1.3.8, *MEMTYPE — External-Memory Interface Register*).

Figure 2–18. DRAM Refresh Sequence (All Widths)



Note: Refresh cycle time = 12 PLL periods with zero wait states (wait state delay of 4 PLL periods per wait state is inserted at the vertical lines).

2.1.8 Interfacing With External I/O Registers

The IN and OUT instructions supply 4-bit I/O addresses with the upper 12 bits of the address bus being zero-filled. R/$\overline{W}$ controls the direction of the data transfer, the $\overline{IS}$ strobe goes active low, and additional control signals are used based on the selections in the MEMTYPE memory-mapped register. The NWS (number of wait states), N/W (narrow/wide format), and D/S (dynamic/standard memory) parameters selected for the program space in the MEMTYPE register control accesses to the I/O space.

2.2 Interrupts

Vector locations and priorities for all internal and external interrupts are shown in Table 2–14.

Table 2–14. Interrupt Locations and Priorities

Interrupt	Interrupt Location	Interrupt Priority	Interrupt Description
$\overline{RS}$	0	1 (Highest)	External reset signal
$\overline{INT0}$	2	2	External user interrupt
SDINT	4	3	Sigma-delta interrupt
CINT	6	4	Real-time counter interrupt
—	8 – 23	—	Reserved locations
TINT	24	5	Internal timer interrupt
RINT	26	6	Serial-port receive interrupt
XINT	28	7 (Lowest)	Serial-port transmit interrupt
TRAP	30	—	TRAP instruction

The TRAP instruction, used for software interrupts, is not prioritized but is included here because it has its own vector location. Each interrupt address has been spaced apart by two locations so that branch instructions can be accommodated in those locations if desired.

When an interrupt occurs, it is stored in the 6-bit interrupt flag register (IFR). This register is set by the external user interrupt $\overline{INT0}$ and the internal interrupts SDINT, CINT, TINT, RINT, and XINT. Each interrupt is stored in the IFR until it is recognized, and then automatically cleared by the $\overline{IACK}$ (interrupt acknowledge) signal or the $\overline{RS}$ (reset) signal. No instructions are provided for reading from or writing to the IFR.

The six interrupts that are stored in the IFR may be masked by using the interrupt mask register (IMR). A logic 1 in bit positions 0 through 5 of the IMR enables the corresponding interrupt, provided that the INTM (interrupt mode) bit of status register ST0 is cleared. A logic zero in bit positions 0 through 5 of the IMR disables the corresponding interrupt. For further details on the IMR, see subsection 2.1.3.5, *IMR — Interrupt Mask Register.*

The INTM bit enables or disables all maskable interrupts. INTM = 0 enables all the unmasked interrupts, and INTM = 1 disables all maskable interrupts. INTM is set by the DINT (disable interrupt) instruction, the $\overline{IACK}$ signal, and

the $\overline{RS}$ signal, and it is cleared by the EINT (enable interrupt) instruction. Note that INTM does not actually modify the IMR or IFR.

When an interrupt occurs during a multicycle instruction, the interrupt is not processed until the instruction is completed. Also, when an instruction is being repeated because of the RPT (repeat instruction as specified by data memory value) or RPTK (repeat instruction as specified by immediate value) instruction, the interrupt is not processed until the repeated execution is finished.

Interrupts cannot be processed between the EINT instruction and the next instruction in a program sequence. This allows a subroutine to have EINT as its last instruction before the RET (return from subroutine) instruction, while ensuring that the return is executed before any interrupts are processed.

2.2.1 Reset Interrupt

Reset ($\overline{RS}$) is a nonmaskable external interrupt that can be used at any time to put the MSP58C80 into a known state. Reset is typically applied after power up when the machine is in a random state. At power up, the state of the processor is undefined. For correct system operation after power up, a reset signal must be asserted low for at least three cycles of CLKOUT1 to ensure a reset of the device.

Execution starts from location 0 of program memory when the $\overline{RS}$ signal is taken high. The MSP58C80 can be held in the reset state indefinitely.

The following list describes the state of the MSP58C80 that results from a reset operation. In this list, binary digits that are not changed by the reset operation are indicated by the letter X. Any item listed as unchanged is unchanged on reset provided that power has not been removed. However, if power has been removed, all such items are undetermined.

1) The following data lines are placed in a high-impedance state:
   - B15–B0 = hi–Z
   - D15–D0 = hi–Z
   - DX = hi–Z
2) External memory control signals are deasserted by being set to a logic 1:
   - $\overline{CAS}$ = 1 (no refresh cycles while $\overline{RS}$ is asserted)
   - $\overline{RAS}$ = 1 (no refresh cycles while $\overline{RS}$ is asserted)
   - R/$\overline{W}$ = 1
   - $\overline{ALATCH}$ = 1
   - $\overline{STRB}$ = 1
   - $\overline{IS}$ = 1
   - $\overline{DS}$ = 1
   - $\overline{PS}$ = 1
   - $\overline{AS}$ = 1

3) The remaining output lines are set as follows:

- A15–A0 = undetermined
- CLKOUT1, CLKOUT2 = running at default speed of 512 kHz
- DD0 = 1
- $\overline{\text{IACK}}$ = 0 (as for a maskable interrupt)
- DIGS = 0
- DIGL = 0
- ADCLK = 0

4) Table 2–15 lists the settings for memory-mapped registers.

Table 2–15. Memory-Mapped Register Settings After Reset

Address	Name	Status†
0h	DRR	Unchanged (serial-port receive register is unchanged)
1h	DXR	Unchanged (serial-port transmit register is unchanged)
2h	TIM	Set to FFFFh (timer register has maximum value; begins decrementing after $\overline{\text{RS}}$ is deasserted)
3h	PRD	Set to FFFFh (period register has maximum value)
4h	IMR	Unchanged (interrupt masking is unchanged)
5h	—	Reserved
6h	RTC	Set to 0000h (does not begin decrementing until a non-zero value is written to it)
7h	FREQ	Set to E900h (for a 4.096 MHz external crystal this gives: PLL clock = 2.048 MHz; processor clock = 2.048 MHz; sigma-delta sample rate = 200 Hz; refresh rate = 128 kHz)
8h	MEMTYPE	Set to 0000h (refresh disabled; all memory is standard wide with zero wait states)
9h	DI	Not controlled by MSP58C80 (D-port input is determined by external levels)
Ah	DO	Unchanged (D-port output is unchanged)

† Those registers listed as unchanged are unchanged only when power has not been removed. Otherwise, they are undetermined.

Table 2–15. Memory-Mapped Register Settings After Reset (Continued)

Address	Name	Status†
Bh	DDIR	Set to 0000h (D port configured as input; since MEMTYPE is initialized to work with standard wide memory, DDIR is initialized to allow the entire D port to be used for interfacing with memory)
Ch	BI	Not controlled by MSP58C80 (B-port input is determined by external levels)
Dh	BO	Unchanged (B-port output is unchanged)
Eh	BDIR	Set to 0000h (B port configured as input)
Fh	—	Reserved
10h	ADAC	Set to FFC0h (sigma-delta ADC and DAC are disabled; no ADC or DAC samples processed; ADC and DAC operating at nominal sample rate)
11h	SDAD	Set to 0000h (sigma-delta ADC input is cleared)
12h	SDDA	Set to 0000h (sigma-delta DAC output is cleared)
13h	SAAD	Set to FF00h (successive-approximation ADC input is cleared; no channel is selected)
14h – 17h	—	Reserved

† Those registers listed as unchanged are unchanged only when power has not been removed. Otherwise, they are undetermined.

5) Internal I/O-mapped registers are unchanged.

6) The status registers are set as per the TMS320C25 (see Table 2–16 and Table 2–17).

Table 2–16. Status Register ST0 Reset Setting

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARP			OV	OVM	1	INTM	DP								
ST0 =	X	X	X	0	X	1	1	X	X	X	X	X	X	X	X	X

Table 2–17. Status Register ST1 Reset Setting

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARB			CNF	TC	SXM	C	1	1	HM	FSM	XF	FO	TXM	PM	
ST1 =	X	X	X	0	X	1	1	1	1	1	1	1	0	0	0	0

These settings indicate that:

- No overflow has occurred.
- The interrupts are disabled.
- The internal RAM block B0 is configured as data memory.
- The sign-extension mode is on.
- The carry bit is set.
- The serial port is configured to use frame synchronization pulses.
- The external flag is set.
- The serial-port registers are configured for 16-bit words.
- The FSX signal is configured as an input.
- The multiplier 32-bit product is loaded to the ALU without shifting.

7) Other registers are set as follows:

   - IFR = 0 (no interrupts pending)
   - RPTC = 0 (repeat counter is cleared)
   - PC = 0000h (execution begins at program memory location 0)

8) RAM is unchanged.

2.3 Analog and Digital Converters

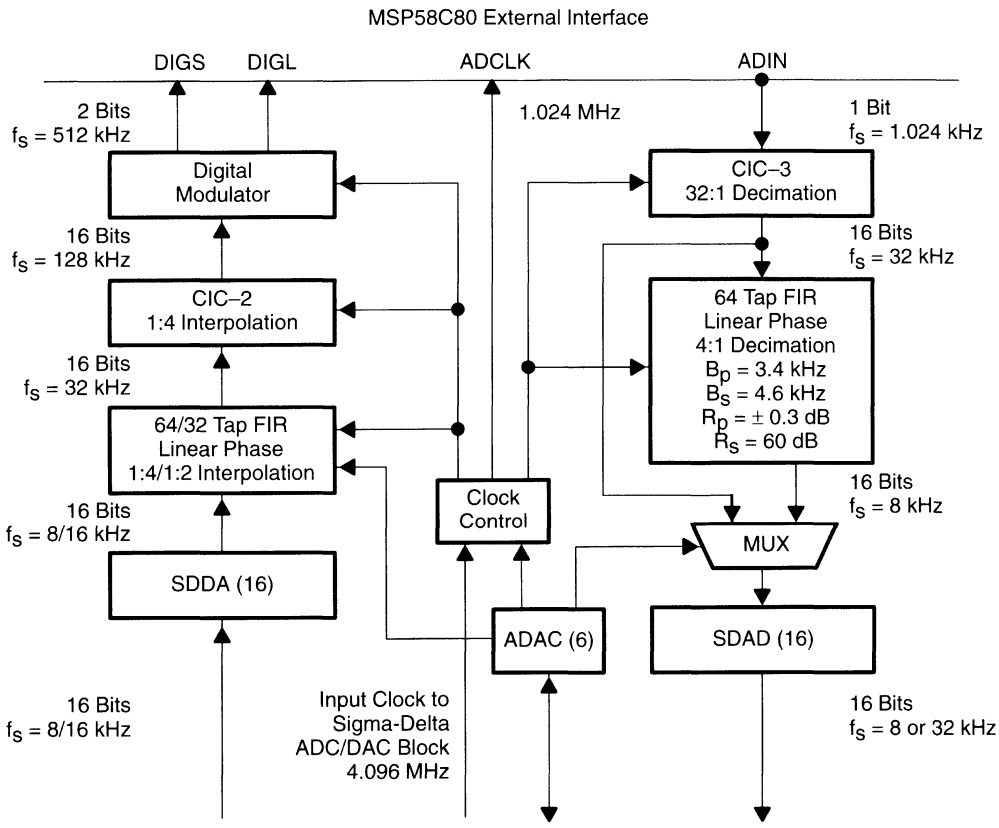
The MSP58C80 and an external companion device, the MSP58C20, provide a voice-band ADC and DAC (see Figure 2–19 and Appendix E). The analog section of the sigma-delta converters is isolated in the MSP58C20, which communicates with the MSP58C80 using a minimum of four interface lines (see Table 2–18). All decimation and interpolation is performed by dedicated digital filters on the MSP58C80, placing no software overhead on the DSP core or its memory.

The DAC can synchronously operate in full duplex with the ADC. The DAC can be set to operate at the nominal or double the nominal sample rate by way of the ADAC memory-mapped register (see subsection 2.1.3.15, *ADAC — Sigma-Delta ADC/DAC Control Register*). The ADC can also be set by way of the ADAC register to run either at the nominal sample rate or at four times the nominal sample rate. When the ADC runs at the nominal sample rate, a 64-tap mask-programmed FIR low-pass filter is used. When the ADC runs at four times the nominal sample rate, the FIR filter is bypassed and a custom digital filter can be implemented in software.

A single interrupt, SDINT, is provided for both the sigma-delta ADC and DAC since the sample rates are synchronous. When the interrupt is generated, the ADC and DAC status bits in the ADAC memory-mapped register indicate whether one or both of the converters have processed a sample. The nominal sample rate is determined by the PLL-clock setting and the sigma-delta predivider, which are selected with the FREQ memory-mapped register (see subsection 2.1.3.7, *FREQ — Frequency Control Register*).

A successive-approximation ADC is included on the MSP58C80 for systems needing to monitor low frequency or dc signals. The successive-approximation ADC is designed to offer medium performance at relatively slow conversion times. The successive-approximation ADC value is buffered in the lower 8 bits of the SAAD memory-mapped register with V_{DD} represented by FFh and V_{SS} represented by 00h. No interrupts are generated, but a new conversion is initiated after each read from the SAAD register. With a 4.096-MHz external crystal, the SAAD should not be accessed more rapidly than 16 kHz. Before reading the first data value from the SAAD after a reset, this register must be primed with a dummy read. One of three analog signals may be switched to the successive-approximation ADC through an on-chip analog multiplexer.

Figure 2–19. Oversampling Digital Filters in the MSP58C80



Note: f_S = sampling frequency
B_p = band-pass cutoff frequency
Bs = band-stop cutoff frequency
R_p = pass-band ripple
R_S = stop-band ripple

Table 2–18. MSP58C80/MSP58C20 Connections

MSP58C20 Terminal	MSP58C80 Terminal
DIGL	DIGL
DIGS	DIGS
ADOUT	ADIN
ADCLK	ADCLK
PWDA	User-defined
PWAD	User-defined

2.4 Low-Power Operation

The MSP58C80 is designed to be used in a single-processor environment with both DSP and general controller tasks. Many control functions require much less processing power than do DSP functions. Unfortunately, most processors have a fixed master oscillator with a speed dictated by the peak requirements of the DSP algorithm. This leads to excessive average power consumption, which is undesirable for many consumer applications.

The MSP58C80 uses a flexible clock control system to solve this peak-performance/average-power dilemma. Through software control, the PLL clock can be adjusted, making it possible to dynamically manage power consumption. In addition, the processor clock has a control that is independent of other functional blocks. Since the majority of the power is consumed by the DSP core, reducing the processor clock allows the power to be scaled down proportionally. The processor-clock rate can be modified between instruction cycles to yield the desired performance or power consumption by changing the processor-clock predivider in the FREQ memory-mapped register (see subsection 2.1.3.7, *FREQ — Frequency Control Register*). Since this control is independent of the sigma-delta sample rate and the external memory controller, changing the processor-clock predivider does not require interruptions in the operation of those blocks.

Control bits are provided in the ADAC memory-mapped register to independently enable and disable the sigma-delta ADC and DAC circuitry located on the MSP58C80. When disabled, each block is placed in an initialized, low-power state. The external MSP58C20 companion device can be placed into low-power mode by control of the PWAD and PWDA signals by way of general I/O terminal(s) on the MSP58C80. A logic high level on the PWAD signal places the ADC portion of the MSP58C20 in low-power mode, and a logic high level on the PWDA signal places the DAC portion of the MSP58C20 in low-power mode.

Reducing the speed of system clocks and disabling the sigma-delta circuitry are the primary means of decreasing power consumption. Executing the IDLE instruction results in a small additional decrease in power consumption. In order to realize the full benefits of decreasing power consumption, B-port and D-port signals need to be programmed either as outputs that have fixed logic levels or as inputs that are driven to V_{SS} or V_{DD}.

Chapter 3

Applications

The following sections give information on using the MSP58C80 in an application.

Topic **Page**

3.1 Software Package

A software package is included with the MSP58C80 to facilitate the implementation of a digital telephone answering device. The software includes 4.875 kbps and 7.4 kbps modified code-excited linear prediction (MCELP) speech analysis and synthesis routines which can record and play back outgoing and incoming messages. There is also a linear predictive coding (LPC) synthesis routine that allows day/time stamp and voice-menuing functions to be implemented with prerecorded, edited speech. The other routines included in the software package are voice activity detection (VAD), dual-tone multifrequency (DTMF) or touch-tone detection and generation, call progress tone (CPT) detection, autodisconnect, ring detection, real-time clock (RTC) handling, near-end echo cancellation, and miscellaneous control code, which includes ARAM error management.

Table 3–1 indicates the processor requirements of the TI-supplied software package. In the worst case, 14.32 MIPS are required by the software package, leaving approximately 2 MIPS for customer-specific code. There are 9928 words of ROM available and 134 words of RAM available for the customer code in the worst case.

Table 3–1. Processor Requirements of TI-Supplied Software

Function	MIPS	ROM Words	RAM Words
MCELP analysis + VAD	10.3†	4518†‡	837†
MCELP synthesis + VAD	4		574
LPC synthesis	3	1294†	183
DTMF generation	<1.4	182†	5
DTMF detection	<1.2†	277†	41†
CPT detection	<1.5†	361†	45†
Ring detection	<0.3	72†	5
RTC handling	<0.02†	77†	3†
Miscellaneous control code	<1.2†	3771†	252†
Total value	16.384	20480	1312
Worst case†	14.32	10552	1178
Left for control functions in worst case	2.064	9928	134

† Items that are used to calculate worst case.
‡ This number combines both the MCELP analysis + VAD and the MCELP synthesis + VAD.

3.2 Reset Circuit and Interrupts

After the MSP58C80 has recognized a high level on $\overline{RS}$, any negative-going transition on the $\overline{INT0}$ line that meets the minimum $\overline{INT0}$ pulse width is recognized as an interrupt request. Negative-going transitions that do not meet the minimum $\overline{INT0}$ pulse width also have a possibility of being recognized. Therefore, when an interrupt pending condition is not desired, care must be used to ensure that $\overline{INT0}$ does not glitch low (even if the glitch is of a short duration) after $\overline{RS}$ goes high.

3.3 Programming the FREQ and MEMTYPE Memory-Mapped Registers

When refresh is enabled, there are several MSP58C80 constraints that need to be observed in programming the FREQ and MEMTYPE memory-mapped registers:

- ❑ The time required to perform one refresh must be less than the requested refresh period (see Section C.1, *PLLFG and NWS Settings To Avoid If Refresh Is Enabled*).
- ❑ The requested refresh period must be greater than the instruction period (see Section C.2, *PLLFG and PCPD Settings To Avoid If Refresh Is Enabled*).
- ❑ The time required to perform one DRAM write must be less than the requested refresh period (see Section C.3, *Settings To Avoid If Dynamic Memory Reads and Writes are Performed and Refresh Is Enabled*).
- ❑ The time required to perform one SRAM/ROM/EPROM write must be less than the requested refresh period (see Section C.4, *Settings To Avoid If Standard Memory Reads and Writes are Performed and Refresh Is Enabled*).
- ❑ The timing requirements for any external memory devices must be compatible with the MSP58C80 for each PLLFG setting and NWS setting used by the application (see Appendix B).

An increase in wait-state value should precede an increase of the PLL clock rate, and a decrease in wait-state value should follow a decrease of the PLL clock rate. This results in a temporary, transitional setting as shown in item 2 of the following sequence:

1) High PLL speed, large NWS setting
2) Low PLL speed, large NWS setting
3) Low PLL speed, small NWS setting

In applications which change PLL clock rate, it is important to verify the transitional setting as well as the high-speed and low-speed settings. If the transitional setting violates any timing requirements, an intermediate-speed setting must be chosen that avoids this problem:

- ❑ High PLL speed, large NWS setting
- ❑ Intermediate PLL speed, large NWS setting
- ❑ Wait until frequency reaches intermediate-speed setting (can take a few instruction cycles)

- ❑ Intermediate PLL speed, small NWS setting
- ❑ Low PLL speed, small NWS setting

To change from high-speed to low-speed operation, this sequence is followed. To change from low-speed to high-speed operation, the reverse of this sequence is followed.

Chapter 4

MSP58C80 Development Tools

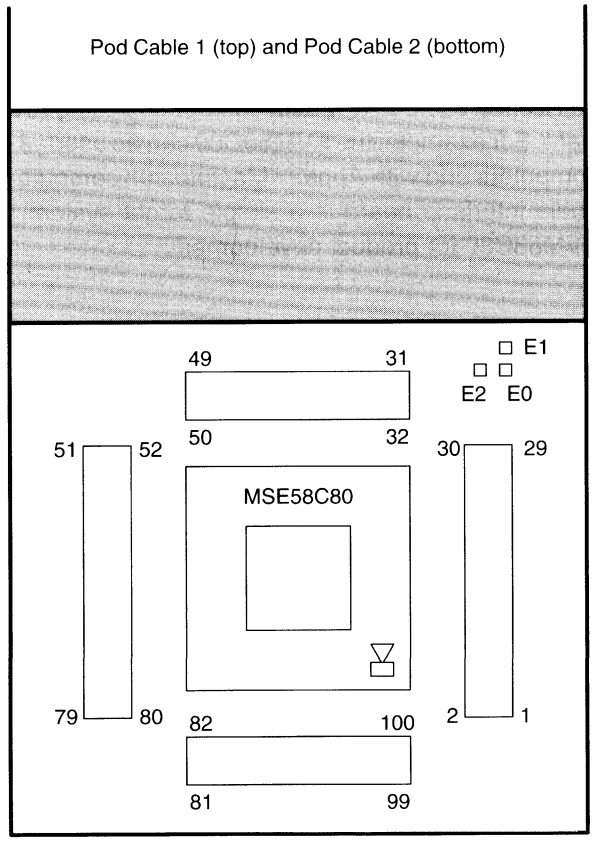
The EVM58C80 is a desktop development tool that is capable of emulating the MSP58C80 at full speed (65.536 MHz). The EVM58C80 can be connected by an RS-232 cable to any host computer running ANSI terminal emulation software. For example, the EVM58C80 can be used with an IBM PC or compatible computers and with Apple™ Macintosh™ computers. The optional AIB58C80 audio interface board is a simple target system that can be used with the EVM58C80 for product development.

Topic **Page**

4.1 EVM58C80 In-Circuit Emulator

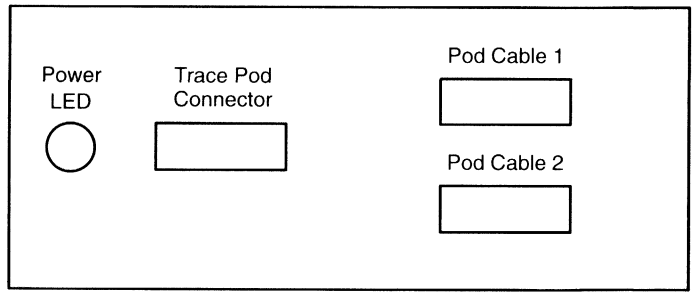
The EVM58C80 SE pod contains a MSE58C80 chip (a MSP58C80 emulator for development use only), four emulation connectors with connector pins numbered 1–100, and two 40-pin cables (see Figure 4–1).

Figure 4–1. EVM58C80 SE Pod



The SE pod is connected to your target board using the four emulation connectors. The emulation connector pins allow emulation of MSP58C80 signal functions. The emulation connector pin numbers correspond to the MSP58C80 terminal numbers described in Table 1–1. If your target board has a PQFP footprint but cannot accommodate the emulation connectors, an adapter from Emulation Technology can be used (part numbers EPP–100–QF06SA and MX–100–QF06–SUB). The SE pod is connected to the EVM58C80 box using two cables, pod cable 1, and pod cable 2 (see Figure 4–1 and Figure 4–2).

Figure 4–2. EVM58C80 Box (Front View)



Your target board must do the following (these functions are all performed by the AIB58C80 shown in Figure 4–4):

- ❑ Provide a 4.096-MHz oscillator circuit (see Figure 1–3)
- ❑ Provide a PLL filter (see Figure 1–3)
- ❑ Drive all input signals
- ❑ Supply power to appropriate pins (see the following description)

In order to supply power to the MSE58C80, a jumper must be placed either between E0 and E1 or between E0 and E2 on the SE pod (see Figure 4–1). When the jumper is placed between E0 and E1, all MSE58C80 V_{DD} terminals (except $ADCV_{DD}$ and $PLLV_{DD}$) are powered by the EVM58C80. When the jumper is placed between E0 and E2, all MSE58C80 V_{DD} terminals (except ADC V_{DD} and PLL V_{DD}) are powered by the emulation connector V_{DD} terminals. In this case, your target board powers the MSE58C80 V_{DD} terminals. The following connector terminals must have power separately supplied, *regardless of the choice of jumper position*:

- ❑ ADC V_{DD} (terminal 1),
- ❑ ADC V_{SS} (terminal 100),
- ❑ PLL V_{DD} (terminal 51),
- ❑ PLL V_{SS} (terminal 50).

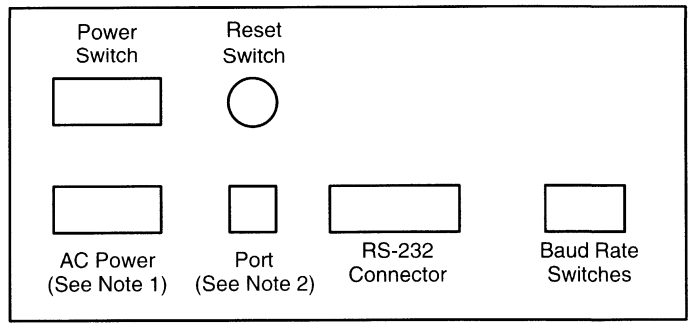
All MSE58C80 V_{SS} terminals are common with all emulation connector V_{SS} terminals (except ADC V_{SS} and PLL V_{SS}).

$\overline{RS}$ (terminal 39) is processed by the EVM58C80 before it is presented to the MSE58C80. As a result, either a pullup resistor or a reset circuit with a switch (see Figure 1–4) can be connected to $\overline{RS}$. When a pullup resistor is used, the MSE58C80 can only be reset by the EVM58C80. When a reset circuit with a switch is used, the MSE58C80 can be reset by the switch or by the EVM58C80.

The EVM58C80 has a trace pod that allows up to 16 digital lines to be monitored every instruction cycle. The trace pod has a total of 18 lines, one of which supplies the ground reference (see Figure 4–5).

The EVM58C80 must be connected to an ac power source that fits the following specification; 90 V to132 V at 47 Hz to 63 Hz (see Figure 4–3).

Figure 4–3. EVM58C80 Box (Rear View)



Note 1) AC power can be 90 V to 132 V at 47 Hz to 63 Hz. However, if *230 V/50 Hz* is printed on the back panel, ac power can also be 175 V to 264 V at 47 Hz to 63 Hz.
2) This is reserved for future expansion.

If *230 V/50 Hz* is printed on the back panel of the EVM58C80, it can also be used with a 175 V to 264 V at 47 Hz to 63 Hz power source. The EVM58C80 must be connected to the host computer serial port using the RS-232 cable supplied with the EVM58C80 (see Figure 4–3). The following software is used to create an object file and a symbol file that can be uploaded to the EVM58C80:

- ❑ DSPA†
- ❑ DSPLNK†
- ❑ DSPHEX†
- ❑ SYM1 or SYM2 (supplied with the EVM58C80)

† This software can be obtained by ordering a TMS320C2x/C5x C compiler (TI part number TMDS3242855–02).

The assembler (DSPA), linker (DSPLNK), and object format converter (DSPHEX) are used to create a TI-tagged object file that can be uploaded to the EVM58C80. The symbol table generator (SYM1 or SYM2) creates a symbol table file that also can be uploaded to the EVM58C80.

The PROCOMM PLUS™ software package (for IBM PC or compatible computers) is supplied with the EVM58C80. To use the EVM58C80 at its fastest communications speed, set the baud rate switches on the back of the EVM58C80 for 38400 baud (see Figure 4–3 and Figure 4–6). You should then launch PROCOMM PLUS, and select the following communications parameters: 38400 baud, no parity, eight data bits, one stop bit.

A program can be uploaded to the EVM58C80 with the DP (transmit from disk to program memory) command, and a symbol table can be uploaded to the

EVM58C80 with the DT (transmit from disk to symbol table) command. The program can then be executed by either using the G (go) command or the SS (single step) command. The G command executes until an H (halt) command is issued or until a breakpoint is reached. The SS command causes a single instruction to be executed.

When a program is executed with either the G or SS command, a debug display appears with the following information:

- The name of the next instruction to be executed. The instruction hexadecimal and mnemonic form are listed with the argument listed as a symbolic name if the symbol table has an entry for the argument.
- Up to eight previous instructions (listed in hexadecimal and mnemonic form) and the 16-line digital trace value for each instruction
- A user-selectable portion of the data space
- MSE58C80 registers (accumulator, product register, temporary register, status registers, auxiliary registers, and stack)

A trace buffer can be displayed with the LT (list trace) command. This display can be used to view a history of the instructions that have been executed and the digital trace values recorded along with those instructions. The trace buffer can hold 4000 single-word instructions.

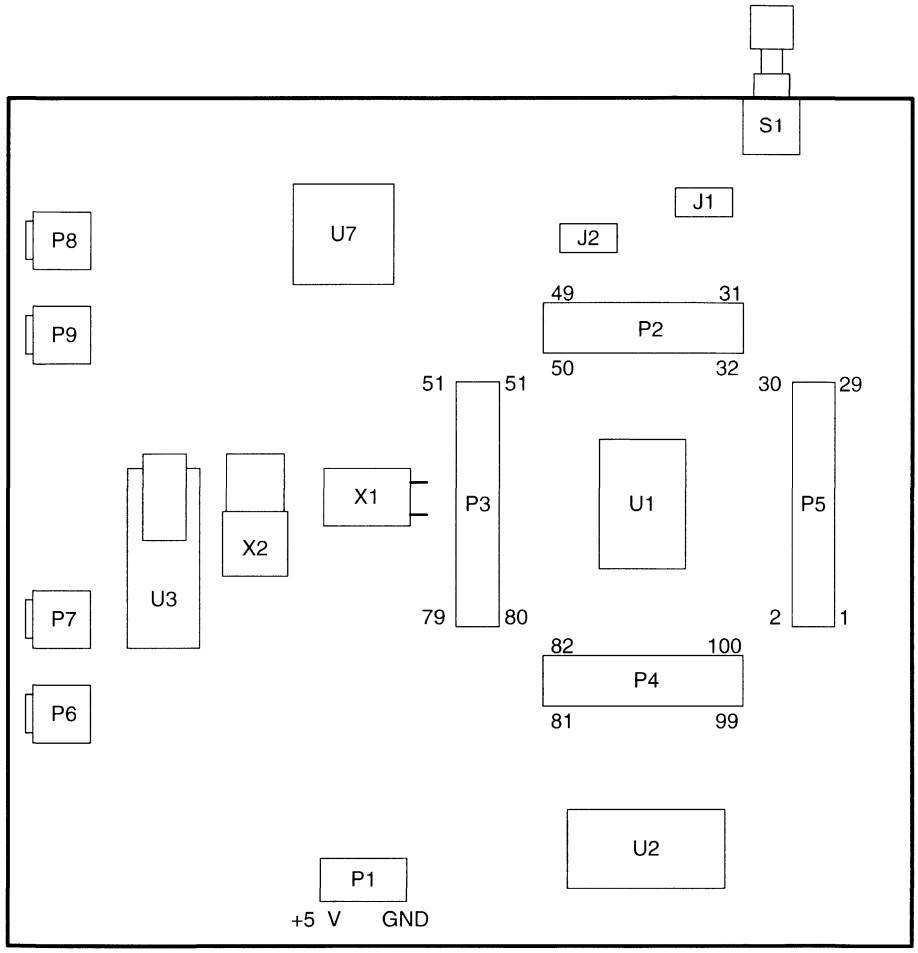
You can perform many other functions with the EVM58C80, such as editing program or data memory, modifying MSE58C80 registers, and moving program or data memory. The EVM58C80 commands are described in Table 4–1 and Table 4–2. For a step-by-step description of how to connect hardware to the EVM58C80 and how to assemble and execute software see Appendix A.

Note:

Please note that the following MSE58C80 resources are reserved for use by the EVM58C80:

- Four RAM locations (7C–7Fh)
- One stack level

Figure 4–4. AIB58C80



Note: J1 = MP/$\overline{\text{MC}}$ control (driven low when jumper is present)
J2 = $\overline{\text{INT0}}$ control (driven high when jumper is present)
P1 = 5-V dc power plug
P2–P5 = Connectors for emulation of MSP58C80
P6 = MSP58C20 audio in†
P7 = MSP58C20 audio out†
P8 = TLC320AC01 audio in†
P9 = TLC320AC01 audio out†
S1 = Reset switch
U1 = Footprint for MSP58C80
U2 = 4M×1 DRAM/ARAM
U3 = MSP58C20
U7 = TLC320AC01
X1 = Crystal for MSP58C80/MSE58C80
X2 = Oscillator for TLC320AC01

† All audio levels are line-level 0.775 Vrms

Figure 4–5. EVM58C80 Trace Pod

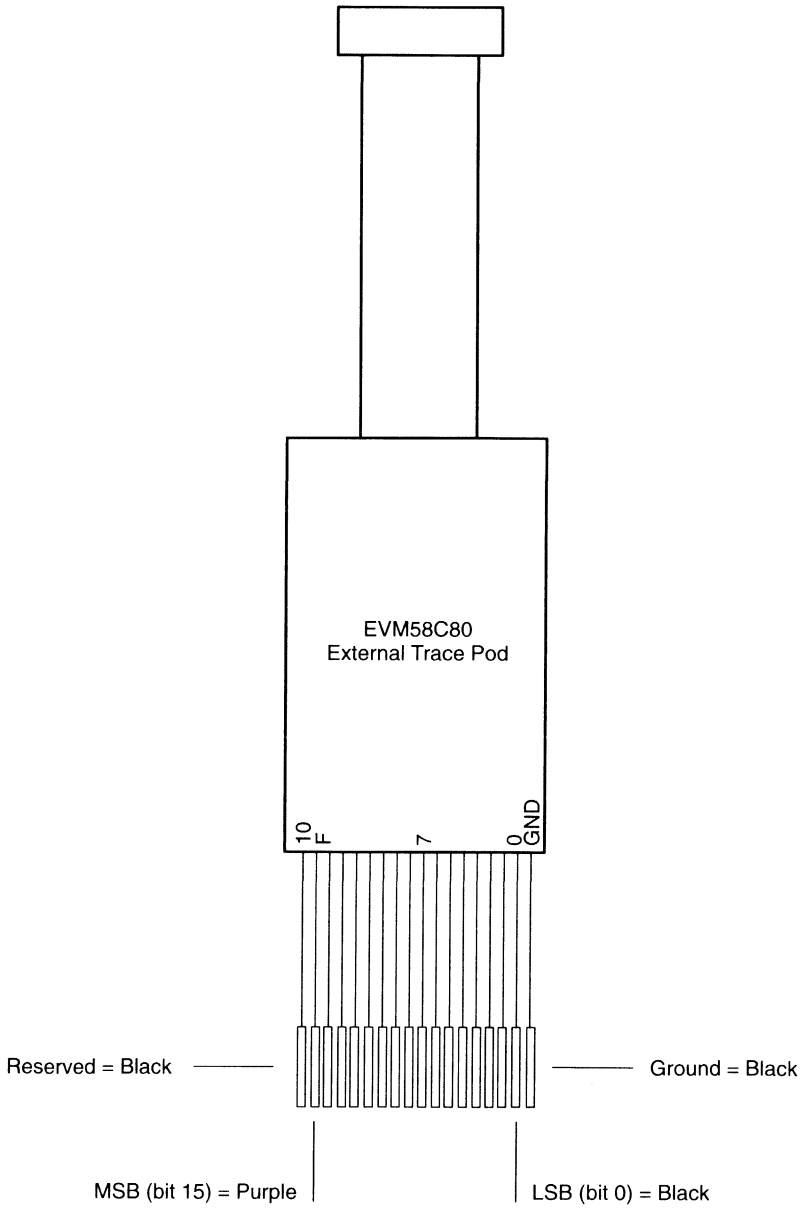
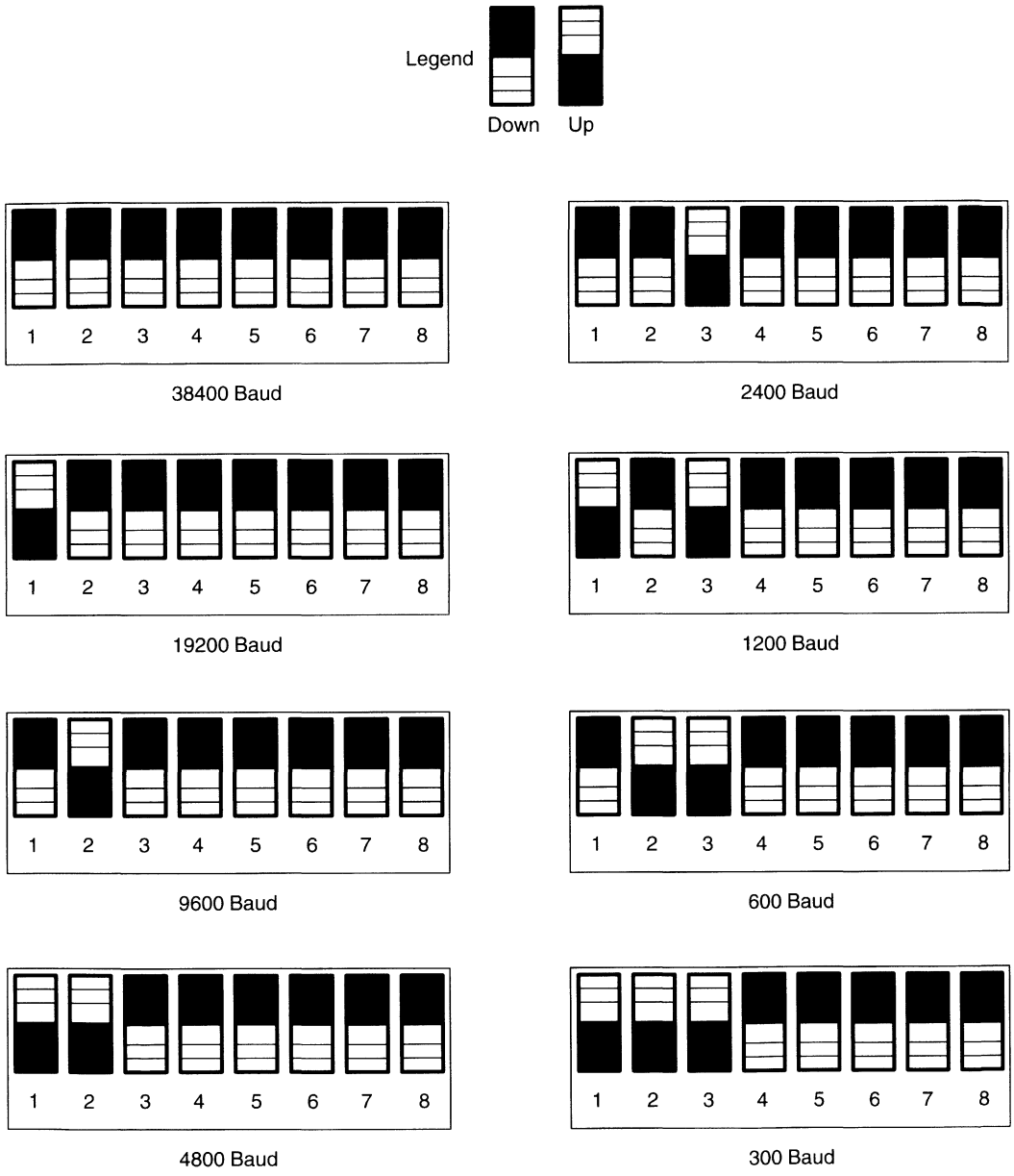


Figure 4–6. EVM58C80 Baud Rate Switch Settings



4.2 EVM58C80 Monitor-Mode Commands

This section gives the commands, syntax, arguments, and command descriptions for the EVM58C80 development tool when in the default monitor mode. The following list contains the descriptions for the monitor-mode command arguments:

- [xxx] = optional parameter
- aaaaaaaaaaaaaaaa = alphanumeric label (16 characters)
- dddd = destination address (4 hexadecimal characters)
- eeee = end address (4 hexadecimal characters)
- ee:eeee = end address (6 hexadecimal characters, separated by a colon)
- n = single-digit value
- nnnn = address offset value (4 hexadecimal characters)
- ssss = start address (4 hexadecimal characters)
- ss:ssss = start address (6 hexadecimal characters, separated by a colon)
- vvvv = value (4 hexadecimal characters)

Special Characters

The following is a list of the special characters accepted by the EVM58C80:

- . — Exit edit mode
- * — Abort execution of a command
- <space> — Pause/continue a listing

Table 4–1. EVM58C80 Monitor-Mode Commands†

Command Syntax	Description
AD n	Examine successive-approximation ADC MUX n (. = exit)
AH [vvvv]	Examine/edit high accumulator
AL [vvvv]	Examine/edit low accumulator
AM [ss:ssss]	View alternate memory, starting at address ss:ssss
AR [n] [vvvv]	Examine/edit auxiliary register n
ARB [n]	Examine/edit auxiliary register pointer buffer (ARB)
ARP [n]	Examine/edit auxiliary register pointer (ARP)
BD n	Disable (clear) breakpoint n (BC n can also be used)
BK	Halt program execution and show the debug display

† The monitor mode is the default mode when EVM58C80 powers up with DIP switch 7 set in the down position.

Table 4–1. EVM58C80 Monitor-Mode Commands (Continued)†

Command Syntax and Attributes	Description
BL	List breakpoints
BS n [ssss] [eeee]	Set breakpoint n at ssss through eeee (i.e., define addresses ssss–eeee to be a breakpoint range)
C [n]	Examine/edit carry bit (C)
CA	Disable (clear) all breakpoints
CB	Change baud rate (if power is cycled on the EVM58C80, the baud rate reverts to what is selected by the baud rate switches)
CNF [n]	Examine/edit on-chip RAM configuration control bit (CNF)
DA [ss:ssss]	Transmit from disk to alternate memory, starting at address ss:ssss
DB	Show the debug display
DD [ssss]	Transmit from disk to data memory, starting at address ssss
DM [ssss]	View data memory, starting at address ssss
DP [ssss]	Transmit from disk to program memory, starting at address ssss
DPP [nnn]	Examine/edit data memory pointer (DP)
DS	Transmit from disk an emulator operating system
DT	Transmit from disk a symbol table
EA [ss:]ssss	Edit alternate memory starting at ss:ssss
ED ssss	Edit data memory starting at ssss
EP ssss	Edit program memory starting at ssss
ES	Enter new symbol
ET [aaaaaaaaaaaaaaaa]	Edit symbol table starting with aaaaaaaaaaaaaaaa
FA [ss:]ssss eeee vvvv	Fill alternate memory with vvvv from ss:ssss through ss:eeee if eeee ≥ ssss; otherwise, fill alternate memory with vvvv from ss:ssss through ss+1:eeee
FD ssss eeee vvvv	Fill data memory with vvvv from ssss through eeee
FO [n]	Examine/edit format bit (FO)
FP ssss eeee vvvv	Fill program memory with vvvv from ssss through eeee
FSM [n]	Examine/edit frame synchronization mode bit (FSM)
G [ssss]	Go from current PC or ssss

† The monitor mode is the default mode when EVM58C80 powers up with DIP switch 7 set in the down position.

Table 4–1. EVM58C80 Monitor-Mode Commands (Continued)†

Command Syntax and Attributes	Description
GT eeee	Go from current PC and set breakpoint 8 at eeee
HA	Halt program execution without showing the debug display
HM [n]	Examine/edit hold mode bit (HM)
IN n	Input from I/O port n and display
INTM [n]	Examine/edit interrupt mode bit (INTM)
LA [ss:ssss] [nnnn]	List nnnn words of alternate memory starting from ss:ssss
LD [ssss] [nnnn]	List nnnn words of data memory starting from ssss
LP [ssss] [nnnn]	List nnnn words of program memory starting from ssss
LS [aaaaaaaaaaaaaaaa]	List symbols starting with aaaaaaaaaaaaaaaa
LT [ssss]	List trace memory starting at trace buffer location ssss
MA [ss:]ssss eeee [dd:]dddd	Move alternate memory from ss:ssss through eeee to addresses starting at [dd:]dddd
MD ssss eeee dddd	Move data memory from ssss through eeee to addresses starting at dddd
MP ssss eeee dddd	Move program memory from ssss through eeee to addresses starting at dddd
NA [ss:]ssss eeee vvvv	Search alternate memory from ss:ssss through ss:eeee for value not = vvvv
ND [ssss] [eeee] vvvv	Search data memory from ssss through eeee for value not = vvvv
NP [ssss] [eeee] vvvv	Search program memory from ssss through eeee for value not = vvvv
OU n vvvv	Output to I/O port n value vvvv
OV [n]	Examine/edit overflow flag bit (OV)
OVM [n]	Examine/edit overflow mode bit (OVM)
PC [vvvv]	Examine/edit program counter
PH [vvvv]	Examine/edit product register high
PL [vvvv]	Examine/edit product register low
PM [ssss]	View program memory, starting at address ssss

† The monitor mode is the default mode when EVM58C80 powers up with DIP switch 7 set in the down position.

Table 4–1. EVM58C80 Monitor-Mode Commands (Continued)†

Command Syntax and Attributes	Description
PSM [n]	Examine/edit product shift mode (PM)
QU	Quiet mode enable/disable. When quiet mode is enabled (QU 1), typing G or SS does not invoke the debug mode. When quiet mode is disabled (QU 0), typing G or SS does invoke the debug mode.
RA ss:ssss	Receive (load) alternate memory binary file
RD ssss	Receive (load) data memory binary file
RE	Reset target MSE58C80
RP ssss	Receive (load) program memory binary file
SA [ss:]ssss eeee vvvv	Search alternate memory from ss:ssss through ss:eeee for value = vvvv
SD ssss eeee vvvv	Search data memory from ssss through eeee for value = vvvv
SK [n] [vvvv]	Examine/edit stack location n
SP ssss eeee vvvv	Search program memory from ssss through eeee for value = vvvv
SR [n] [vvvv]	Examine/edit status register n
SS [ssss]	Single-step starting at either address ssss or current PC value
ST vvv	Search trace for PC = vvvv
SXM [n]	Examine/edit sign extension mode bit (SXM)
T [vvvv]	Examine/edit temporary register
TC [n]	Examine/edit test/control flag bit (TC)
TR ssss eeee dddd	Table read. Move program memory ssss through eeee to data memory starting at dddd
TW ssss eeee dddd	Table write. Move data memory ssss through eeee to program memory starting at dddd
TXM [n]	Examine/edit transmit mode bit (TXM)
VA ssss	Viewport address; set the beginning address of data memory to view in the debug display and show debug display
WA [ss:]ssss eeee	Display Intel hex format representation of alternate memory ss:ssss through ss:eeee on screen
WD ssss eeee	Display Intel hex format representation of data memory ssss through eeee on screen

† The monitor mode is the default mode when EVM58C80 powers up with DIP switch 7 set in the down position.

Table 4–1. EVM58C80 Monitor-Mode Commands (Continued)†

Command Syntax and Attributes	Description
WP ssss eeee	Display Intel hex format representation of program memory ssss through eeee on screen
XA ss:ssss nnnn	Transmit nnnn binary words of alternate memory starting from ss:ssss
XD ssss nnnn	Transmit nnnn binary words of data memory
XF [n]	Examine/edit external flag output status bit (XF)
XP ssss nnnn	Transmit nnnn binary words of program memory
XR	Transmit register values in binary format

† The monitor mode is the default mode when EVM58C80 powers up with DIP switch 7 set in the down position.

4.3 EVM58C80 Debug-Mode Commands

The debug mode enables the programmer to stop execution of the program and perform certain functions in order to observe or improve the execution of that program. The debug mode is entered with execution of one of the following monitor-mode commands: BK, DB, G, SS, or VA. However, when quiet mode is engaged, neither G nor SS cause debug mode to be entered. Table 4–2 lists the debug-mode commands.

Table 4–2. EVM58C80 Debug Mode Commands†

Command	Description
G	Go. Go from current address in PC
H	Halt. Halt program execution
J	Jump. Jump around call, halting upon return from subroutine call
M	Monitor. Return to monitor mode
R	Repeat. Go from current address in PC and halt upon return to current address in PC
S	Step. Single step
T	Trace. Show trace display
U	Update. Update debug display and continue execution

† Debug mode is entered with the BK, DB, G, SS, or VA commands.

Chapter 5

Frequently Asked Questions

This chapter contains a list of frequently asked questions with answers.

1) What is the maximum data rate of the speech recording and playback routines?

   The data rate is 4.875 kbps without silence management. A lower data rate is expected with silence suppression, possibly as low as 3.3 kbps. This rate cannot be ensured because the amount of silence for a given speaker cannot be predicted.

2) How many samples are there in each frame of compressed speech?

   There are 192 samples per frame. The sampling rate is 8 kHz. Therefore, each speech frame has a duration of 24 ms. A standard frame is compressed to 117 bits and a silent frame is compressed to 26 bits when using the 4.875 kbps vocoding rate.

3) Can the MSP58C80 directly support 4M $\times$ 4 (16M-bit) DRAM/ARAM?

   Yes, and without requiring glue logic. The DRAM must use the same number of address terminals for the row addresses as for the column addresses.

4) Is it possible to back up ARAM for 100 hours after main power failure?

   With the MSP58C80 slowed down to the minimum speed and only performing refresh operations, the MSP58C80 and ARAM typically draw about 5 mA to 10 mA. According to the Eveready manual (see *Related Documentation* on page v), four AA alkaline batteries fall to 1.3 V after 138 hours of operation at 8 mA. If there is no regulator to drop the voltage, 100 hours of battery back up is possible.

5) What is the allowable tolerance of the 4.096-MHz crystal?

   It is anticipated that crystals with a tolerance less than or equal to 200 ppm are used by our customers. The accuracy that is desired for the crystal is determined principally by the accuracy desired for the real-time clock function and DTMF detection/generation. Long-term drift with temperature, a typical crystal characteristic, causes only slight distortion of speed and pitch on playback. This is not considered a noticeable problem.

Appendix A

Assembly To Emulation With a Simple Program

This appendix provides a listing of a simple MSP58C80 assembly program and its linker command file. The appendix also lists the steps that a user must perform in order to run the program on the EVM58C80.

Topic **Page**

A.1 Simple Program

The following sample program increments the accumulator and stores the incremented value in data memory.

```
; ************************************************************************
; test.asm is a simple MSP58C80 program that increments the accumulator
; and stores the incremented value in data memory
; ************************************************************************
; RAM locations
data    .usect     "block2",2
; ************************************************************************
       .sect     "vectors"     ; interrupt vectors and reserved ROM
reset  b start                  ; reset vector
       b int0                   ; int0 interrupt vector
       b sdint                  ; sdint interrupt vector
       b cint                   ; cint interrupt vector
       .bes 16*16               ; skip reserved ROM
       b tint                   ; tint interrupt vector
       b rint                   ; rint interrupt vector
       b xint                   ; xint interrupt vector
       b trap                   ; trap interrupt vector
; ************************************************************************
       .sect     "code"
start  zac                      ; accumulator = 0
       ldpk 0                   ; set up for direct addressing
loop   addk 1                   ; increment accumulator
       sach data                ; store result
       sacl data+1              ;   in 2-word location
       b loop                   ; loop forever
int0
sdint
cint
tint
rint
xint
trap
       nop
       ret
       .end
```

A.2 Simple Linker Command File

The simple linker command file provides memory allocation information for the linker.

```
/************************************************************************/
/*                LINKER COMMAND FILE FOR THE MSP58C80                  */
/*                        FILE NAME: TEST.C80                           */
/*                             DATE: 2/93                               */
/************************************************************************/
MEMORY
{
/* PROGRAM MEMORY SPACE */
PAGE 0:   VECS:     org = 0x0000 len = 0x0020 /* INTERRUPT VECTORS */
          IROM:     org = 0x0020 len = 0x4F90 /* INTERNAL ROM */
          XROM:     org = 0x8000 len = 0x8000 /* EXTERNAL ROM */
/* DATA MEMORY SPACE */
PAGE 1:   MMR:      org = 0x0000 len = 0x0014 /* MEMORY-MAPPED REGISTERS */
          BLKB2: org = 0x0060 len = 0x001C /* RAM BLOCK 2 (EXCEPT RESERVED RAM)*/
          BLKB0:    org = 0x0200 len = 0x0200 /* RAM BLOCK 0 */
          BLKB1:    org = 0x0400 len = 0x0300 /* RAM BLOCK 1 */
          RAM:      org = 0x0800 len = 0xF800 /* EXTERNAL RAM */
}
SECTIONS
{
          vectors:  load = VECS               /* INTERRUPT VECTORS */
          code:     load = IROM               /* PROGRAM CODE */
          block2:   load = BLKB2              /* INTERNAL RAM */
}
```

A.3 MS-DOS Commands for Assembling and Linking

In the following list, the boldfaced text lists the MS-DOS command to be executed and the normal text gives a description of the command.

1) **dspa –s –l test.asm**
   This command assembles the file test.asm and generates the list file test.lst and the COFF object file test.obj.

2) **dsplnk test.obj test.c80 –o test.out –m test.map**
   This command links the file test.obj and generates the output map file test.map and the executable COFF object file test.out.

3) **dsphex –t test.out**
   This command converts the executable COFF object file test.out to the TI-tagged object file test.t0.

4) **sym1 test**
   This command reads the executable COFF object file test.out to generate the symbol file test.sym (a log file, test.log, and an intermediate file, test.srt, are also generated).

A.4 Connecting the Hardware

This list explains how to set up the EVM58C80 and AIB58C80 to execute a program.

1) Connect Pod Cable 1 and Pod Cable 2 (40-wire ribbon cables) from the front of the EVM58C80 box to the EVM58C80 SE pod (see Figure 4–1 and Figure 4–2).

2) Connect the EVM58C80 SE pod to the AIB58C80 (audio interface board for the EVM58C80, see Figure 4–1 and Figure 4–4). Verify that the emulation connector pins on the SE pod are aligned with the emulation connectors on the AIB58C80 and verify that pin 1 of the SE pod is inserted into pin 1 of the AIB58C80. Also, verify that the SE pod is fully seated in the AIB58C80.

3) Verify that there is a jumper between jumper post E0 and jumper post E2 on the EVM58C80 SE pod (see Figure 4–1).

4) Connect the RS-232 cable (25-wire ribbon cable) from the back of the EVM58C80 box to a serial port on the host computer (see Figure 4–3).

5) Set the baud rate switches on the back of the EVM58C80 to 38400 baud (see Figure 4–3 and Figure 4–6).

6) Connect a 90 V to132 V, 47 Hz to 63 Hz ac power source to the ac power plug on the back of the EVM58C80 (Figure 4–3). If *230 V/50 Hz* is printed on the back panel of the EVM, a 175 V to 264 V, 47 Hz to 63 Hz ac power source can also be used.

7) Connect a 5-V dc power source to connector P1 on the AIB58C80 (see Figure 4–4).

8) Turn on power to the AIB58C80 and the EVM58C80 at the same time.

A.5 Running the Program

This list explains how to set up PROCOMM PLUS and how to execute a simple program from the EVM58C80.

1) Follow the directions in the PROCOMM PLUS documentation to install PROCOMM PLUS. When prompted for terminal emulation type, select **VT/ANSI** at the main menu and **ANSI** at the submenu. When prompted for file download and upload protocols, select **ASCII**.

2) Type **PCPLUS** at the MS-DOS prompt to execute PROCOMM PLUS.

3) Type any key to enter the terminal mode of PROCOMM PLUS (the EVM58C80 prompt, <, should be visible). If the prompt is not visible, try pressing the carriage return key or typing an asterisk (*).

4) Type **Alt–P** to bring up a *Current Settings* menu.

5) Type **7** to select 38400 baud (provided your computer serial port is capable of operating at this speed).

6) Type **Alt–N** to select no parity, eight data bits, and one stop bit.

7) Type **F1** to select the COM1 port (or type the appropriate key to select the COM port that the EVM58C80 is connected to).

8) Type **Alt–S** to exit the *Current Settings* menu.

9) Type **Alt–S** to bring up a *PROCOMM PLUS Setup Utility* menu.

10) Select **Protocol Options**; then select **ASCII Protocol Options**.

11) Type **D**, then **0**, and press carriage return to set *Character Pacing* to zero.

12) Type **E**, then **0**, and press carriage return to set *Line Pacing* to zero.

13) Type **ESC** to exit the *PROCOMM PLUS Setup Utility* menu.

14) Type **DP** at the EVM58C80 prompt.

15) Type **Page Up** to select a *PROCOMM PLUS upload.*

16) Type **A** to select *ASCII upload*

17) Type **test.t0** to upload the object file to the EVM58C80 (include the path if test.t0 is not in the default directory; e.g., **c:\sample\test.t0**).

18) When the prompt returns, type **DT** at the EVM58C80 prompt.

19) Type **Page Up** to select a *PROCOMM PLUS upload.*

20) Type **A** to select *ASCII upload.*

21) Type **test.sym** to upload the symbol file to the EVM58C80 (include the path if test.sym is not in the default directory; e.g., **c:\sample\test.sym**).

22) When the prompt returns, type **RE** to reset the MSE58C80.

23) Type **VA 60** at the EVM58C80 prompt. This allows you to view the RAM locations 60h–7Fh.

24) Type **G** to execute the program.

25) Type **U** to update the screen and allow program execution to continue. With this program you should see RAM locations 60h and 61h and the accumulator incrementing.

26) Type **H** to halt program execution.

27) Instead of typing **H** to halt program execution, a breakpoint can be set from the monitor mode by using the BS command. In this way, program execution can be halted once an instruction at a specific address or an instruction in a specific address range has been executed. To display breakpoints, use the BL command.

28) Type **Alt–X** to exit PROCOMM PLUS.

A.6 Upgrading the EVM System Software

Before you can upgrade your EVM system software, you must determine whether the EVM58C80 is a RAM-based model or an EPROM-based model. Set switch 7 on the back of the EVM58C80 to the up position (Figure 4–6). Then turn on the EVM58C80. If a text greeting followed by a prompt (<) appears, the EVM58C80 is EPROM-based. If only the prompt (<) appears, the EVM58C80 is RAM-based.

RAM-based EVM58C80:

1) Type **DS** at the EVM prompt.
2) Type **Page Up** to select a *PROCOMM PLUS upload.*
3) Type **A** to select *ASCII upload.*
4) Type **msplehe.tag** to upload the system software to the EVM58C80 (include the path if msplehe.tag is not in the default directory; e.g., **c:\sample\msplehe.tag**).
5) Set switch 7 on the back of the EVM58C80 to the down position and push the reset button on the back of the EVM58C80 (Figure 4–6).

EPROM-based EVM58C80:

1) Turn off the EVM58C80 and disconnect from external circuits.
2) Orient the EVM58C80 so the rear panel faces you.
3) Remove two screws that attach the top cover of the EVM58C80 to the rear panel.
4) Remove the top cover by sliding it toward the rear panel.
5) Remove the two screws on the right side of the rear panel.
6) Remove the two screws from the support rail on which the power supply is mounted.
7) Loosen the five remaining screws on the right side of the EVM58C80 about 1/2 inch.
8) Disconnect the dc power cable from the motherboard connector J5.
9) Gently pull the right side panel out until the support rail and power supply can be rotated counter-clockwise.

10) Rotate the support rail and power supply until sockets U3 and U4 become visible. Remove the EPROMs from these sockets.

11) Program U4 with the least significant byte of the system software (currently called msplehe.u4).

12) Program U3 with the most significant byte of the system software (currently called msplehe.u3).

13) Reinstall the EPROMs and reverse the above steps to reassemble the EVM58C80.

Appendix B

Electrical Specifications and Timings

This appendix contains electrical and timing information for the MSP58C80/C81/C82.

Topic **Page**

B.1 Absolute Maximum Ratings Over Operating Free-Air Temperature Range†‡

Supply voltage, V_{DD} (see Note 1) . . . . . −0.3 V to 7 V
Input voltage range, V_I . . . . . −0.3 V to 7 V
Continuous power dissipation . . . . . 1 W
Operating free-air temperature range, T_A . . . . . 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. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

‡ This table applies to the MSP58C80, MSP58C81, and MSP58C82 only.

NOTE 1: All voltage values are with respect to V_{SS}.

Stresses beyond those listed here may cause permanent damage to the device. This is a stress rating only.

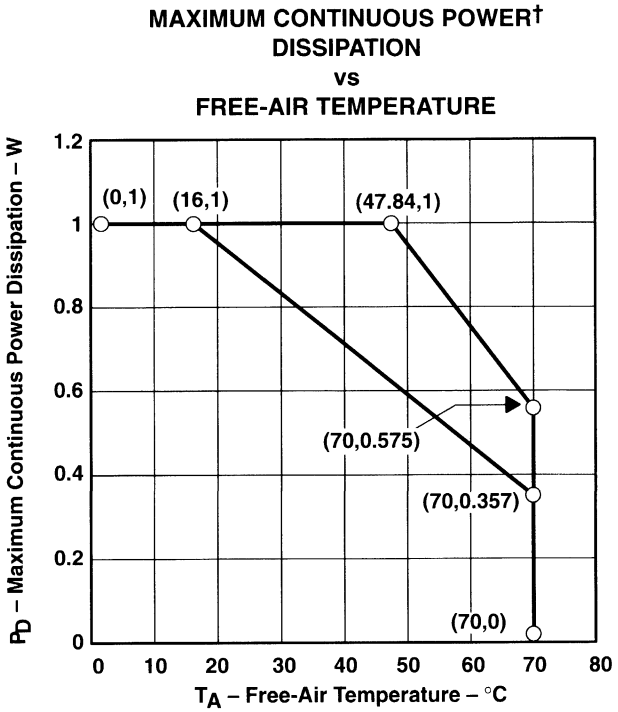
B.2 Recommended Operating Conditions

Table B–1 contains recommended operating conditions for the MSP58C80/C81/C82. Figure B–1 is a graph charting maximum power dissipation versus free-air temperature.

Table B–1.Recommended Operating Conditions

		MSP58C80/C81/C82			UNIT
		MIN	NOM	MAX	
Supply voltage, V_{DD}		4.75	5	5.25	V
Supply voltage, V_{SS}			0		V
High-level input voltage, V_{IH}	All inputs except X2/CLKIN	2.4		V_{DD} + 0.3	V
	X2/CLKIN	3.15		V_{DD} + 0.3	V
Low-level input voltage, V_{IL}		–0.3		0.8	V
High-level output current, I_{OH}				300	µA
Low-level output current, I_{OL}				2	mA
Operating free-air temperature, T_A		0		70	°C

Figure B–1. Maximum Power Dissipation Versus Free-Air Temperature



† Power dissipation should not exceed 1 W.
Free-air temperature should not exceed 70°C.
Junction temperature, as given by the following equation, should not exceed 100°C

$T_J = T_A + (\theta ja)(P_D)$

where

T_J = junction temperature
T_A = free-air temperature
θja = 84°C/Watt for alloy-42 leadframe
52.16°C/Watt for copper leadframe
Please consult with the MSP58C80 product engineer to see which value applies
P_D = power dissipation

B.3 Electrical Characteristics

Table B–2 gives electrical characteristics for the MSP58C80/C81/C82.

Table B–2.Electrical Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MSP58C80/C81/C82			UNIT
			MIN	TYP†	MAX	
V_{OH}	High-level output voltage	V_{DD} = MIN, I_{OH} = MAX, T_A = 25°C	2.4	3		V
V_{OL}	Low-level output voltage	V_{DD} = MIN, I_{OL} = MAX, T_A = 25°C		0.3	0.6	V
I_Z	Three-state current	V_{DD} = MAX	–20		20	µA
I_I	Input current	V_I = V_{SS} to V_{DD}	–10		10	µA
I_{DD}	Supply current (see Note 2)	Using only internal program and data, V_{DD} = MAX, $f_{(PLL)}$ = MAX, T_A = 25°C		100	130	mA
C_i	Input capacitance			15		pF
C_o	Output capacitance			15		pF

† Typical values are at V_{DD} = 5 V and T_A = 25°C unless otherwise noted.

NOTE 2: The measurement of I_{DD} assumes all inputs are tied to either V_{SS} or V_{DD} and all outputs are open circuits.

B.4 Clock and Timing Characteristics

This section contains recommended clock operating conditions for the MSP58C80/C81/C82, dynamic, standard memory timing information for the MSP58C80, $\overline{RS}$ and $\overline{INT0}$ timing for the MSP58C80/C81/C82, and serial port timing for the MSP58C80/C81/C82.

B.4.1 Recommended Clock Operating Conditions

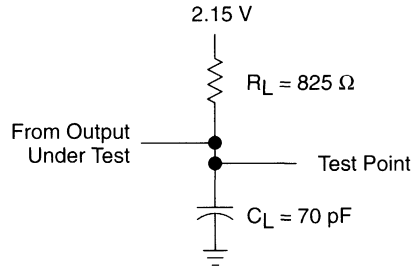
The MSP58C80/C81/C82 can use either its internal oscillator or an external frequency source for a clock. The internal oscillator is enabled by connecting a crystal across X1 and X2/CLKIN. Table B–3 contains the recommended operating conditions for MSP58C80/C81/C82 clocks.

Table B–3. Recommended Clock Operating Conditions, T_A = 0°C to 70°C

PARAMETER			MSP58C80/C81/C82			UNIT
			MIN	TYP	MAX	
f_x	Input clock frequency	MSP58C80/C81	3891	4096	4301	kHz
		MSP58C82	3891	5120	5120	
$f_{(PLL)}$	Internal PLL frequency	MSP58C80/C81	128		65536	kHz
		MSP58C82	128		81920	
f_{sx}	Serial-port frequency	MSP58C80/C81/C82	0†		5120	kHz

† The serial port is not production tested down to 0 kHz. However, the serial port is fully static and should function properly down to 0 kHz.

Figure B–2. Test-Load Circuit



B.4.2 DRAM Timing Requirements for the MSP58C80

The specifications listed in Table B–4 give the requirements of the DRAMS that are used with the MSP58C80. The nominal values in this table indicate design targets. The maximum values indicate the DRAM performance required after accounting for the MSP58C80 input and output buffer delays.

Table B–4.DRAM Timing Requirements†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{AA}	Access time from column-address		4Q + 4QW	4Q + 4QW – 45	s
t_{CAC}	Access time from $\overline{CAS}$ low		2Q + 4QW	2Q + 4QW – 45	s
t_{CPA}	Access time from column precharge		4Q + 4QW	4Q + 4QW – 45	s
t_{RAC}	Access time from $\overline{RAS}$ low		6Q + 8QW	6Q + 8QW – 45	s
t_{OFF}	Output disable time after $\overline{CAS}$ high			4Q + 4QW – 5	s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

B.4.3 Dynamic-Memory Switching Characteristics for the MSP58C80

The specifications listed in Table B–5 give the characteristics of signals produced by the MSP58C80. The typical values indicate design targets. The minimum and maximum values indicate the range of the MSP58C80 signals that may be produced.

Table B–5.DRAM Switching Characteristics†‡

PARAMETER			MSP58C80			UNIT
			MIN	NOM	MAX	
t_{RC}	Random read or write cycle time ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		12Q + 12QW – 5			s
t_{PC}	Page-mode read or write cycle time		4Q + 4QW – 5	4Q + 4QW		s
t_{RASP}	Page-mode pulse duration	Narrow	66Q + 68QW – 5	66Q + 68QW	66Q + 68QW + 5	s
		Wide	18Q + 20 QW – 5	18Q + 20 QW	18Q + 20 QW + 5	s
t_{RAS}	Nonpage-mode pulse duration, $\overline{RAS}$ low ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		6Q + 8QW – 5	6Q + 8QW	6Q + 8QW + 5	s
t_{CAS}	Pulse duration, $\overline{CAS}$ low		2Q + 4QW– 5	2Q + 4QW	2Q + 4QW + 5	s
t_{CP}	Pulse duration, $\overline{CAS}$ high		2Q – 5	2Q		s
t_{RP}	Pulse duration, $\overline{RAS}$ high (precharge)		6Q + 4QW – 5			s
t_{WP}	Write pulse duration	Narrow	72Q + 72QW – 5	72Q + 72QW		s
		Wide	24Q + 24QW – 5	24Q + 24QW		s
t_{ASC}	Column-address setup time, before $\overline{CAS}$ low		2Q – 5	2Q		s
t_{ASR}	Row-address setup time before $\overline{RAS}$ low		2Q – 5	2Q		s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–5. DRAM Switching Characteristics (Continued)†‡

PARAMETER			MSP58C80 MIN	MSP58C80 NOM	MSP58C80 MAX	UNIT
t_{DS}	Data setup time		2Q – 5	2Q		s
t_{RCS}	Read setup time before $\overline{CAS}$ low		6Q + 4QW – 5			s
t_{CWL}	R/$\overline{W}$ low setup time before $\overline{CAS}$ high		8Q + 8QW – 5	8Q + 8QW		s
t_{RWL}	R/$\overline{W}$ low setup time before $\overline{RAS}$ high	Narrow	68Q + 68QW – 5	68Q + 68QW		s
		Wide	20Q + 20QW – 5	20Q + 20QW		s
t_{WCS}	R/$\overline{W}$ low setup time before $\overline{CAS}$ low (early write operation only)		6Q + 4QW – 5	6Q + 4QW		s
t_{WSR}	R/$\overline{W}$ high setup time ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		2Q – 5			s
t_{CAH}	Column-address hold time after $\overline{CAS}$ low		2Q + 4QW – 5	2Q + 4QW		s
t_{DHR}	Data hold time after $\overline{RAS}$ low		6Q + 8QW – 5	6Q + 8QW		s
t_{DH}	Data hold time		2Q + 4QW – 5	2Q + 4QW		s
t_{AR}	Column-address hold time after $\overline{RAS}$ low		6Q + 8QW – 5	6Q + 8QW		s
t_{RAH}	Row-address hold time after $\overline{RAS}$ low		2Q + 4QW – 5	2Q + 4QW		s
t_{RCH}	Read hold time after $\overline{CAS}$ high		4Q + 4QW – 5	4Q + 4QW		s
t_{RRH}	Read hold time after $\overline{RAS}$ high		4Q + 4QW – 5	4Q + 4QW		s
t_{WCH}	R/$\overline{W}$ hold time after $\overline{CAS}$ low (early write operation only)		6Q + 8QW – 5	6Q + 8QW		s
t_{WCR}	Write hold time after $\overline{RAS}$ low	Narrow	70Q + 72QW – 5	70Q + 72QW		s
		Wide	22Q + 24QW – 5	22Q + 24QW		s
t_{WHR}	R/$\overline{W}$ high hold time ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		10Q + 12QW – 5			s
t_{CHR}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		6Q + 8QW – 5	6Q + 8QW		s
t_{CRP}	Delay time, $\overline{CAS}$ high to $\overline{RAS}$ low		6Q + 4QW – 5			s
t_{CSH}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ high		6Q + 8QW – 5	6Q + 8QW		s
t_{CSR}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ low ($\overline{CAS}$-before-$\overline{RAS}$ refresh only)		2Q – 5	2Q		s
t_{RAD}	Delay time, $\overline{RAS}$ low to column address		2Q + 4QW – 5	2Q + 4QW		s
t_{RAL}	Delay time, column address to $\overline{RAS}$ high		4Q + 4QW – 5	4Q + 4QW		s
t_{CAL}	Delay time, column address to $\overline{CAS}$ high		4Q + 4QW – 5	4Q + 4QW		s
t_{RCD}	Delay time, $\overline{RAS}$ low to $\overline{CAS}$ low		4Q + 4QW – 5	4Q + 4QW		s
t_{RPC}	Delay time, $\overline{RAS}$ high to $\overline{CAS}$ low		4Q + 4QW – 5			s
t_{RSH}	Delay time, $\overline{CAS}$ low to $\overline{RAS}$ high		2Q + 4QW – 5	2Q + 4QW		s
t_{REF}	Refresh time interval (1024 cycle)	RD = 32		8		ms
		RD = 64		16		ms

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

B.4.4 SRAM Switching Characteristics for the MSP58C80

Table B–6 through Table B–13 give the switching characteristics for the MSP58C80 when interfacing with standard memory (SRAM, EPROM, or ROM).

Table B–6. Standard Wide-Memory Switching Characteristics for Read Cycle (Program, Data, or I/O Space) (Figure 2–6)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CR1}	Read cycle time	4Q + 4QW – 5	4Q + 4QW	4Q + 4QW + 5	s
$t_{a(A)1}$	Access time from address	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(S)1}$	Access time from $\overline{STRB}$ low	2Q + 4QW – 45	2Q + 4QW		s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–7. Standard Wide-Memory Switching Characteristics for Write Cycle (Program, Data, or I/O Space) (Figure 2–7)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CW1}	Write cycle time	4Q + 4QW – 5	4Q + 4QW	4Q + 4QW + 5	s
t_{W1}	Write-low pulse width	4Q + 4QW – 5	4Q + 4QW	4Q + 4QW + 5	s
$t_{su(A)1}$	Address setup time before $\overline{STRB}$ low	1Q – 5	1Q	1Q + 5	s
$t_{su(D)1}$	Data setup time before $\overline{STRB}$ high	3Q + 4QW – 5	3Q + 4QW	3Q + 4QW + 5	s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–8. Standard Narrow-Memory Switching Characteristics for Read Cycle (Program, Data, or I/O Space) (Figure 2–8)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CR2}	Read cycle time	8Q + 8QW – 5	8Q + 8QW	8Q + 8QW + 5	s
$t_{a(A)2}$	Low byte access time from address	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(A)3}$	High byte access time from address	7Q+ 8QW – 45	7Q+ 8QW		s
$t_{a(ALATCH)1}$	Low byte access time from $\overline{ALATCH}$ ($\overline{ALATCH}$ provides the lsb of the address)	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(ALATCH)2}$	High byte access time from $\overline{ALATCH}$ ($\overline{ALATCH}$ provides the lsb of the address)	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(S)2}$	Access time from $\overline{STRB}$ low	2Q + 4QW – 45	2Q + 4QW		s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–9. Standard Narrow-Memory Switching Characteristics for Write Cycle (Program, Data, or I/O Space) (Figure 2–9)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CW2}	Write cycle time	8Q + 8QW – 5	8Q + 8QW	8Q + 8QW + 5	s
t_{W2}	Write-low pulse width	8Q + 8QW – 5	8Q + 8QW	8Q + 8QW + 5	s
$t_{su(A)2}$	Low byte setup time, address to $\overline{STRB}$ low	1Q – 5	1Q	1Q + 5	s
$t_{su(A)3}$	High byte setup time, address to $\overline{STRB}$ low	5Q + 4QW – 5	5Q + 4QW	5Q + 4QW + 5	
$t_{su(ALATCH)1}$	Low byte setup time, $\overline{ALATCH}$ low to $\overline{STRB}$ low ($\overline{ALATCH}$ provides lsb of address)	1Q – 5	1Q	1Q + 5	s
$t_{su(ALATCH)2}$	High byte setup time, $\overline{ALATCH}$ high to $\overline{STRB}$ low ($\overline{ALATCH}$ provides lsb of address)	1Q – 5	1Q	1Q + 5	
$t_{su(D)2}$	Data setup time before $\overline{STRB}$ high	3Q + 4QW – 5	3Q + 4QW	3Q + 4QW + 5	s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–10. Standard Wide-Memory Switching Characteristics for Read Cycle (Alternate Data Space) (Figure 2–10)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CR3}	Read cycle time	8Q + 4QW – 5	8Q + 4QW	8Q + 4QW + 5	s
$t_{a(LA)1}$	Access time from low address	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(S)3}$	Access time from $\overline{STRB}$ low	2Q + 4QW – 45	2Q + 4QW		s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–11. Standard Wide-Memory Switching Characteristics for Write Cycle (Alternate Data Space) (Figure 2–11)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CW3}	Write cycle time	8Q + 4QW – 5	8Q + 4QW	8Q + 4QW + 5	s
t_{W3}	Write-low pulse width time	8Q + 4QW – 5	8Q + 4QW	8Q + 4QW + 5	s
$t_{su(HA)1}$	High address setup time before $\overline{ALATCH}$ low ($\overline{ALATCH}$ is provided to latch high address)	2Q – 5	2Q	2Q + 5	s
$t_{su(LA)1}$	Low address setup time before $\overline{STRB}$ low	1Q – 5	1Q	1Q + 5	s
$t_{su(D)3}$	Data setup time from $\overline{STRB}$ high	7Q + 4QW – 5	7Q + 4QW	7Q + 4QW + 5	s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–12. Standard Narrow-Memory Switching Characteristics for Read Cycle (Alternate Data Space) (Figure 2–12)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CR4}	Read cycle time	12Q + 8QW – 5	12Q + 8QW	12Q + 8QW + 5	s
$t_{a(ALATCH)3}$	Low byte access time from $\overline{ALATCH}$ ($\overline{ALATCH}$ provides lsb of address)	5Q + 4QW – 45	5Q + 4QW		s
$t_{a(ALATCH)4}$	High byte access time from $\overline{ALATCH}$ ($\overline{ALATCH}$ provides lsb of address)	3Q + 4QW – 45	3Q + 4QW		
$t_{a(LA)2}$	Low byte access time from low address	3Q + 4QW – 45	3Q + 4QW		s
$t_{a(LA)3}$	High byte access time from low address	7Q + 8QW – 45	7Q + 8QW		
$t_{a(S)4}$	Access time from $\overline{STRB}$ low	2Q + 4QW – 45	2Q + 4QW		s

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

Table B–13. Standard Narrow-Memory Switching Characteristics for Write Cycle (Alternate Data Space) (Figure 2–13)†‡

PARAMETER		MSP58C80			UNIT
		MIN	NOM	MAX	
t_{CW4}	Write cycle time	12Q + 8QW – 5	12Q + 8QW	12Q + 8QW + 5	s
t_{W4}	Write-low pulse width	12Q + 8QW – 5	12Q + 8QW	12Q + 8QW + 5	s
$t_{su(HA)2}$	High-address setup time before $\overline{ALATCH}$ low ($\overline{ALATCH}$ is provided to latch high address)	2Q – 5	2Q	2Q + 5	s
$t_{su(LA)2}$	Low-byte setup time, low address to $\overline{STRB}$ low	1Q – 5	1Q	1Q + 5	s
$t_{su(LA)3}$	High-byte setup time, low address to $\overline{STRB}$ low	5Q + 4QW – 5	5Q + 4QW	5Q + 4QW + 5	
$t_{su(ALATCH)3}$	Low-byte setup time, $\overline{ALATCH}$ to $\overline{STRB}$ low ($\overline{ALATCH}$ provides lsb of address)	3Q – 5	3Q	3Q + 5	s
$t_{su(ALATCH)4}$	High-byte setup time, $\overline{ALATCH}$ to $\overline{STRB}$ low ($\overline{ALATCH}$ provides lsb of address)	1Q – 5	1Q	1Q + 5	
$t_{su(D)4}$	Low-byte setup time, data to $\overline{STRB}$ high	7Q + 4QW – 5	7Q + 4QW	7Q + 4QW + 5	s
$t_{su(D)5}$	High-byte setup time, data to $\overline{STRB}$ high	3Q + 4QW – 5	3Q + 4QW	3Q + 4QW + 5	

† These values assume a transition time of less than 5 ns for input signals.
‡ Q = 1/PLL clock rate; W = wait states programmed

B.4.5 $\overline{RS}$ and $\overline{INT0}$ Timing

Table B–14 gives timing requirements for the MSP58C80/C81/C82 $\overline{RS}$ and $\overline{INT0}$ signals.

Table B–14. $\overline{RS}$ and $\overline{INT0}$ Timing Requirements Over Recommended Operating Conditions†

PARAMETER		MSP58C80/C81/C82			UNIT
		MIN	NOM	MAX	
$t_{w(IN)}$	$\overline{INT0}$ low pulse duration	$t_{c(C)}$			ns
$t_{w(RS)}$	$\overline{RS}$ low pulse duration	$3t_{c(C)}$			ns

† $t_{c(C)} = 4Q$

B.4.6 Serial-Port Timing

Table B–15 gives switching characteristics and Table B–16 gives timing requirements for the MSP58C80/C81/C82 serial port signals.

Table B–15. Serial Port Switching Characteristics Over Recommended Operating Conditions

PARAMETER		MSP58C80/C81/C82			UNIT
		MIN	TYP	MAX	
$t_{d(CH-DX)}$	DX valid after CLKX rising edge†			75	ns
$t_{d(FL-DX)}$	DX valid after FSX falling edge (TXM = 0)†			40	ns
$t_{d(CH-FS)}$	FSX valid after CLKX rising edge (TXM = 1)			40	ns

† The last occurrence of FSX falling and CLKX rising.

Table B–16. Serial Port Timing Requirements Over Recommended Operating Conditions

PARAMETER		MSP58C80/C81/C82			UNIT
		MIN	NOM	MAX	
$t_{c(SCK)}$	Serial-port clock (CLKX/CLKR) cycle time†	200			ns
$t_{f(SCK)}$	Serial-port clock (CLKX/CLKR) fall time			25‡	ns
$t_{r(SCK)}$	Serial-port clock (CLKX/CLKR) rise time			25‡	ns
$t_{w(SCK)}$	Serial-port clock (CLKX/CLKR) low pulse duration§	80			ns
$t_{w(SCK)}$	Serial-port clock (CLKX/CLKR) high pulse duration§	80			ns
$t_{su(FS)}$	FSX/FSR setup time before CLKX/CLKR falling edge (TXM = 0)	18			ns
$t_{h(FS)}$	FSX/FSR hold time after CLKX/CLKR falling edge (TXM = 0)	20			ns
$t_{su(DR)}$	DR setup time before CLKR falling edge	10			ns
$t_{h(DR)}$	DR hold time after CLKR falling edge	20			ns

† The serial port is not production tested down to 0 kHz. However, the serial port is fully static and should function properly down to 0 kHz.

‡ Value derived from characterization data and not tested

§ The duty cycle of the serial-port clock must be within 40–60%.

Appendix C

Memory-Mapped Register Constraints

This chapter contains very important information about the number-of-wait-states (NWS) bits of the MEMTYPE memory-mapped register, the settings of the PLLFG and PCPD bits of the FREQ memory-mapped register, and how together they can alter the operation of the MSP58C80 to the point of being unstable.

Topic **Page**

C.1 PLLFG and NWS Settings To Avoid If Refresh Is Enabled

Some phase-locked loop frequency gain (PLLFG) settings and alternate data space number of wait states (NWS) settings must be avoided if refresh is enabled. Otherwise, refreshes do not occur at the proper rate and the processor can become locked up.

The PLLFG and the alternate data space NWS setting determine the time required to perform a refresh operation. Note that the alternate data space NWS setting determines the number of wait states that are used for refresh timing regardless of whether the dynamic memory is located in the alternate data space. The refresh divider setting (RD) determines the refresh period.

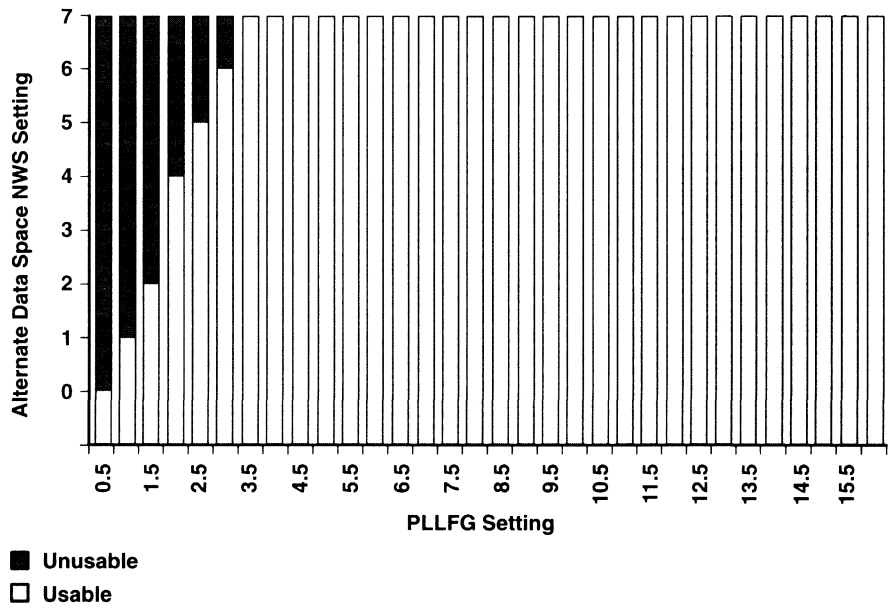
$$\text{Refresh Time} = \frac{(3 \times \text{NWS} + 3) \times 4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Refresh Period} = \frac{\text{RD}}{\text{Reference Oscillator Rate}}$$

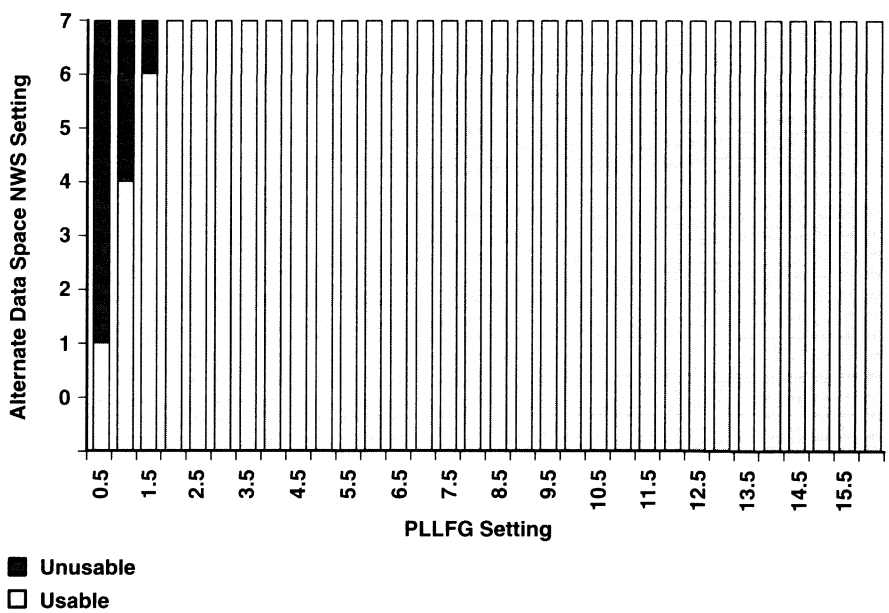
Figure C–1 labels as unusable those settings that cause the refresh time to be greater than or equal to the requested refresh period. In Figure C–1(a) RD equals 32, and in Figure C–1(b) RD equals 64. If one of the unusable settings is chosen, reading from or writing to external memory can cause program execution to be halted. This happens because the MSP58C80 constantly has a refresh request pending, which prevents reads and writes from being performed. Once program execution is halted, a reset signal must be supplied to allow the chip to recover.

Figure C–1.Alternate Data Space NWS Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64



C.2 PLLFG and PCPD Settings To Avoid If Refresh Is Enabled

Some PLL settings (PLLFG) and processor clock predivider settings (PCPD) must be avoided if refresh is enabled. Otherwise, refreshes do not occur at the proper rate.

The PLLFG and the PCPD determine the instruction period. The refresh divider setting (RD) determines the refresh period (i.e., the time period from one refresh operation to the next refresh operation).

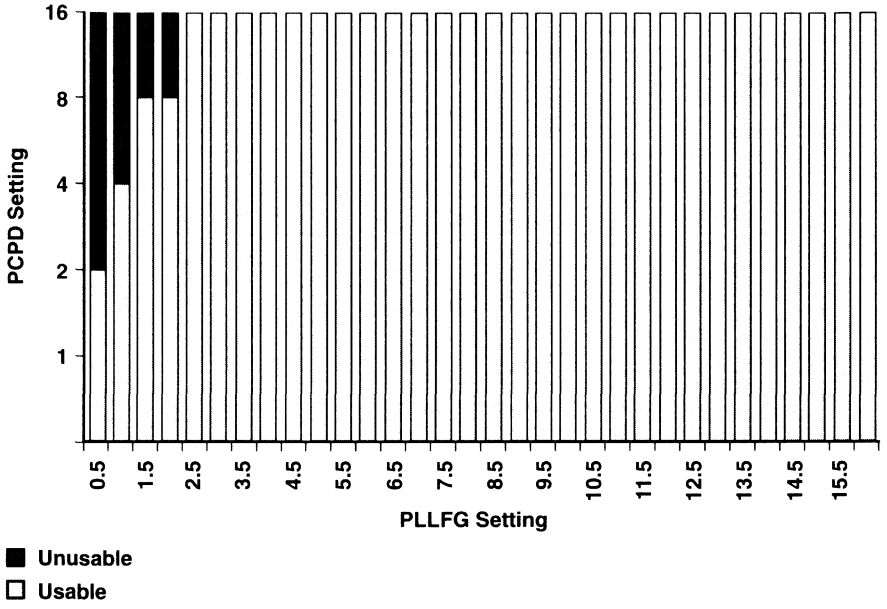
$$\text{Instruction Period} = \frac{4 \times \text{PCPD}}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Refresh Period} = \frac{\text{RD}}{\text{Reference Oscillator Rate}}$$

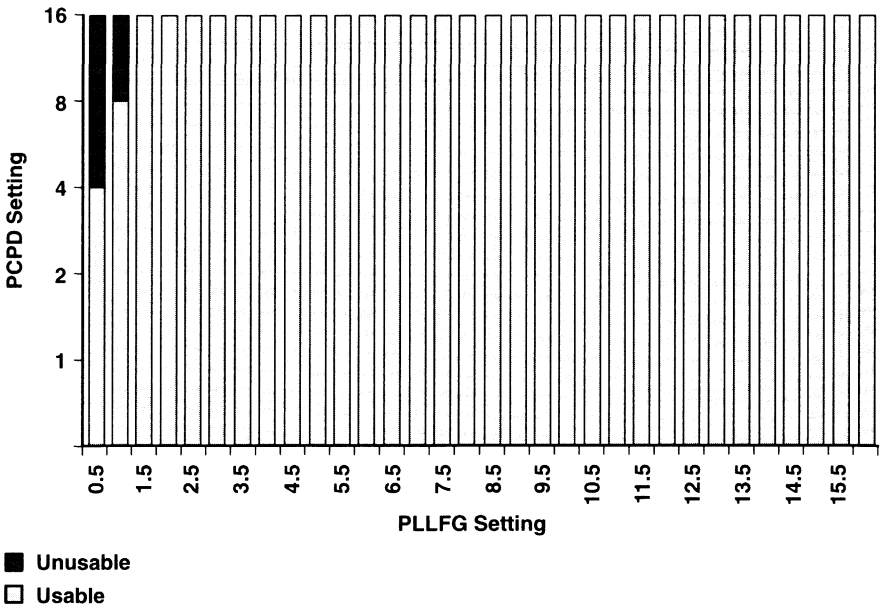
The period between two successive refreshes produced by the MSP58C80 is a multiple of the instruction period. Figure C–2 labels as unusable those settings for which the refresh period is less than or equal to the instruction period. In Figure C–2(a) RD equals 32, and in Figure C–2(b) RD equals 64.

Figure C–2. PCPD Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64



C.3 Settings To Avoid If Dynamic Memory Reads or Writes Are Performed and Refresh Is Enabled

Some PLL settings (PLLFG), processor clock predivider settings (PCPD), and NWS settings for dynamic memory (DRAM) must be avoided if reads from or writes to the dynamic memory are performed. Otherwise, refreshes do not occur at the proper rate.

The PLLFG, the PCPD, and the NWS setting of the memory space that contains dynamic memory determine the time required to write a word to dynamic memory. The refresh divider setting (RD) determines the refresh period.

$$\text{Dynamic Narrow Memory Write Time} = (18 \times \text{NWS} + 18 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

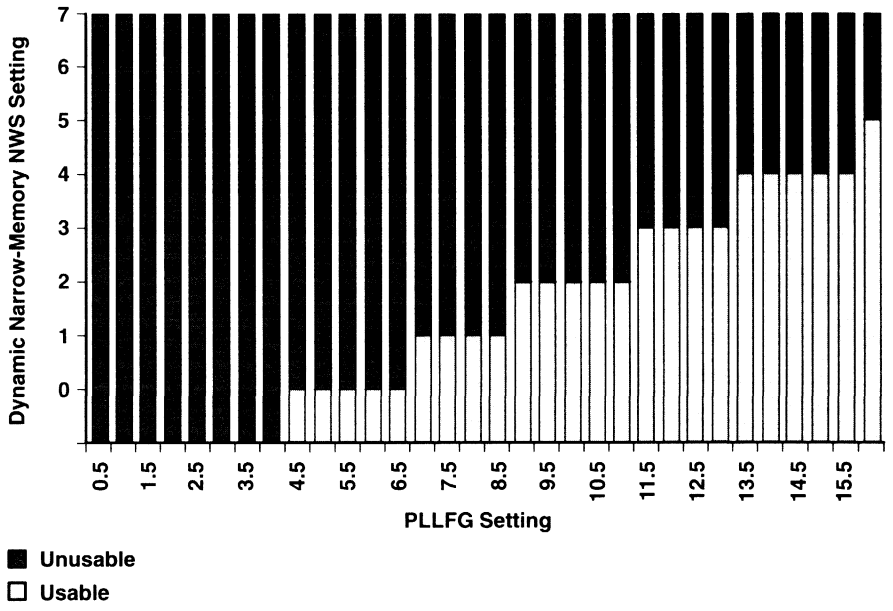
$$\text{Dynamic Wide Memory Write Time} = (6 \times \text{NWS} + 6 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Refresh Period} = \frac{\text{RD}}{\text{Reference Oscillator Rate}}$$

Figure C–3 and Figure C–4 label as unusable those settings that cause the write time to be greater than or equal to the requested refresh period. Figure C–3(a) shows dynamic narrow-memory constraints when RD equals 32, Figure C–3(b) shows dynamic narrow-memory constraints when RD equals 64, Figure C–4(a) shows dynamic wide-memory constraints when RD equals 32, and Figure C–4(b) shows dynamic wide-memory constraints when RD equals 64. The figures assume PCPD equals 16. If PCPD is less than 16, some of the settings labeled as unusable are usable.

Figure C–3.Dynamic Narrow-Memory NWS Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64

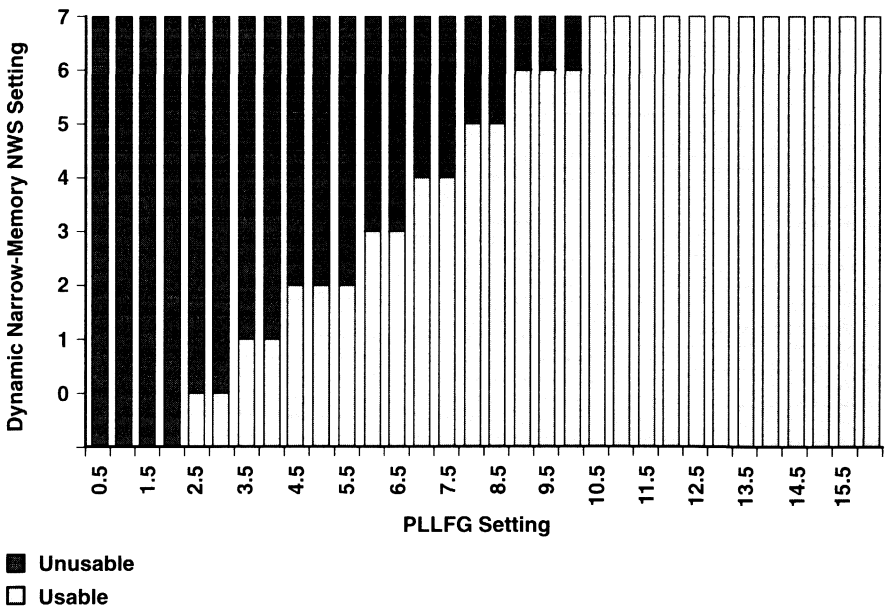
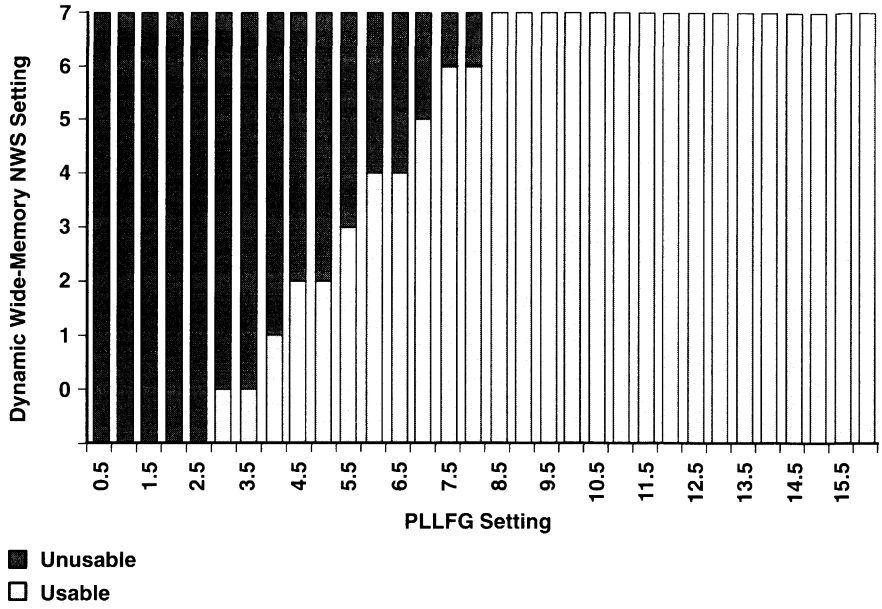
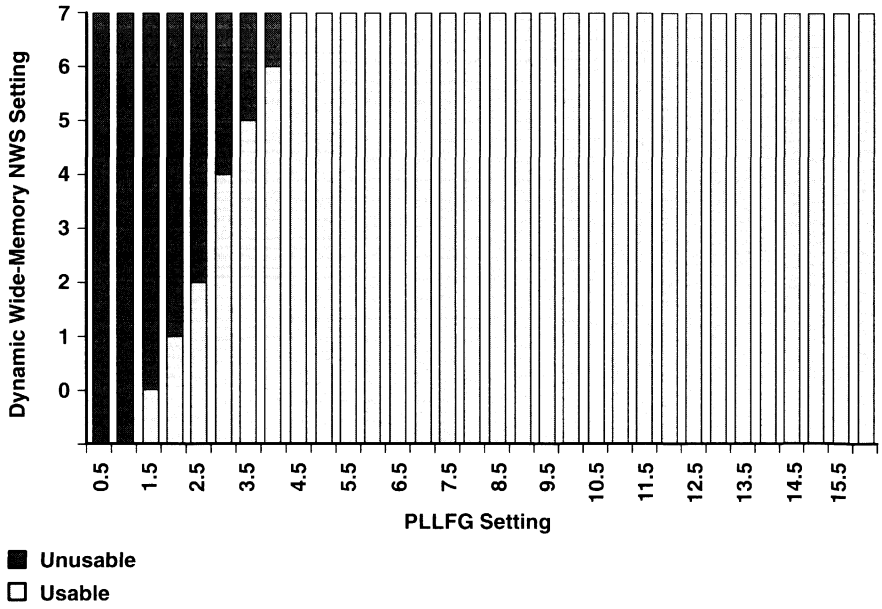


Figure C–4. Dynamic Wide-Memory NWS Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64



C.4 Settings To Avoid If Standard Memory Reads or Writes Are Performed and Refresh Is Enabled

Some PLL settings (PLLFG), processor clock predivider settings (PCPD), and NWS settings for standard memory (SRAM/ROM/EPROM) must be avoided if reads from or writes to the standard memory are performed and refresh is enabled. Otherwise, refreshes do not occur at the proper rate.

The PLLFG, the PCPD, and the NWS setting of the memory space that contains standard memory determine the time required to write a word to standard memory. The refresh divider setting (RD) determines the refresh period.

$$\text{Standard Narrow Memory Write Time (Alternate Data Space)} = (2 \times \text{NWS} + 3 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Standard Narrow Memory Write Time (Program, Data, I/O Space)} = (2 \times \text{NWS} + 2 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Standard Wide Memory Write Time (Alternate Data Space)} = (\text{NWS} + 2 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

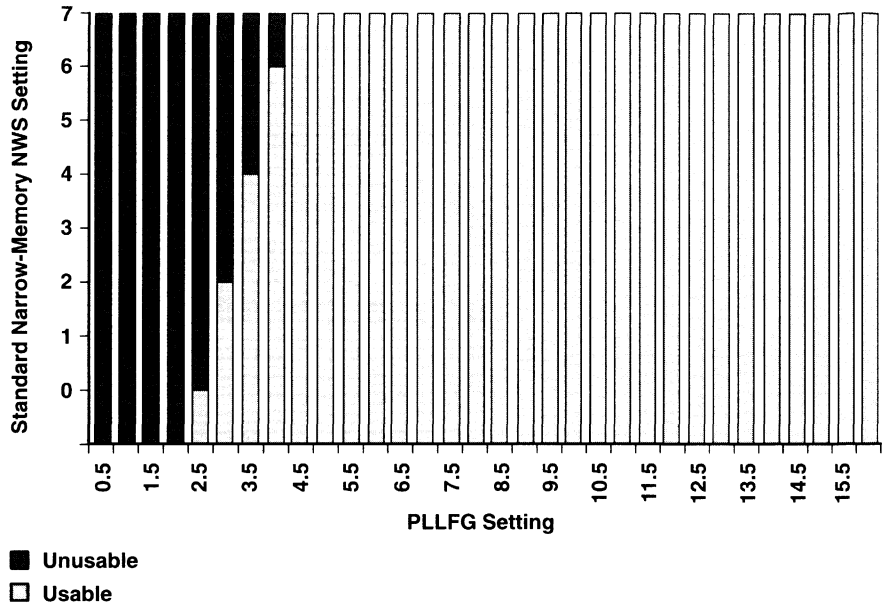
$$\text{Standard Wide Memory Write Time (Program, Data, I/O Space)} = (\text{NWS} + 1 + \text{PCPD}) \times \frac{4}{\text{Reference Oscillator Rate} \times \text{PLLFG}}$$

$$\text{Refresh Period} = \frac{\text{RD}}{\text{Reference Oscillator Rate}}$$

Figure C–5 and Figure C–6 label as unusable those settings that cause the write time to be greater than or equal to the requested refresh period. Figure C–5(a) shows standard narrow-memory constraints when RD equals 32, Figure C–5(b) shows standard narrow-memory constraints when RD equals 64, Figure C–6(a) shows standard wide-memory constraints when RD equals 32, and Figure C–6(b) shows standard wide-memory constraints when RD equals 64. The figures assume the PCPD equals 16. However, if PCPD is less than 16, some of the settings labeled as unusable are usable. The figures assume the alternate data space is being used for the standard memory. If the program, data, or I/O space is being used for the standard memory, some of the settings labeled as unusable are usable.

Figure C–5. Standard Narrow-Memory NWS Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64

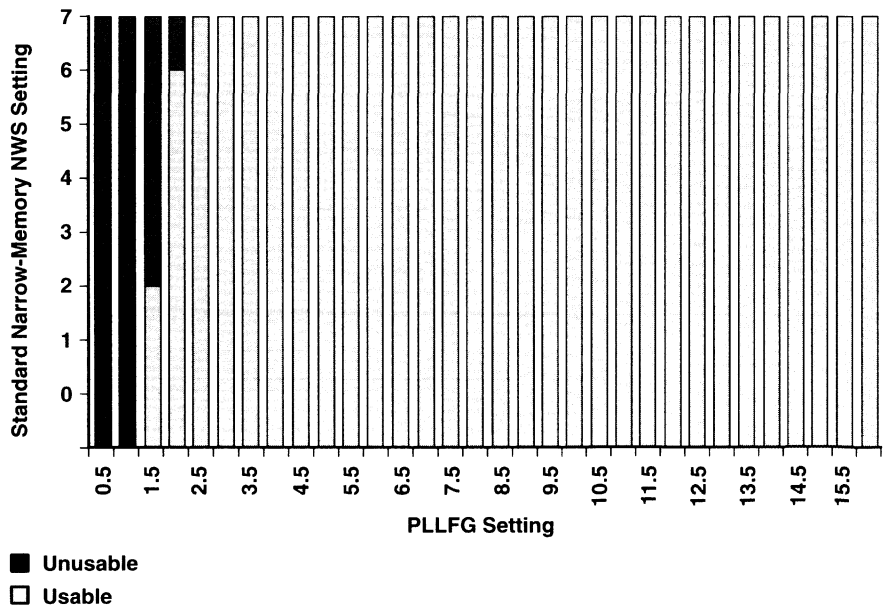
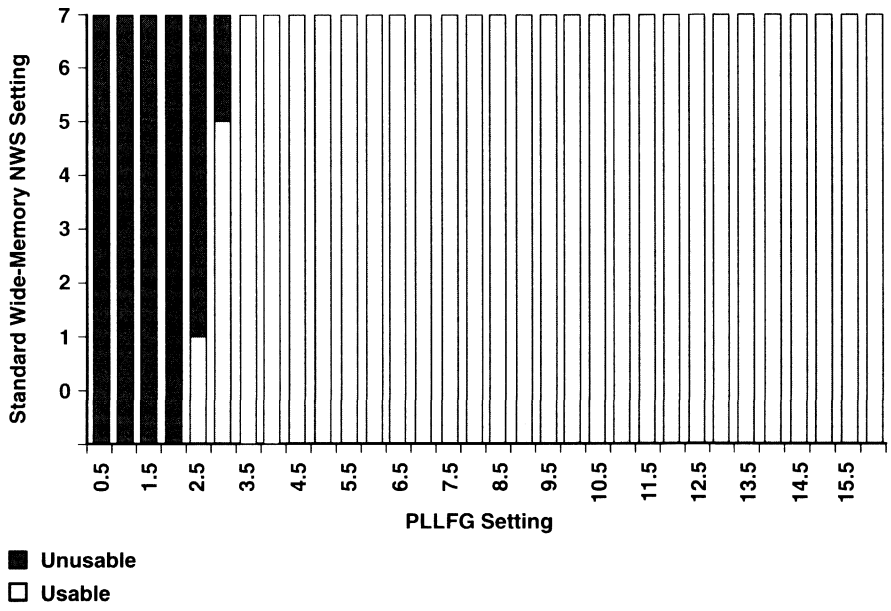
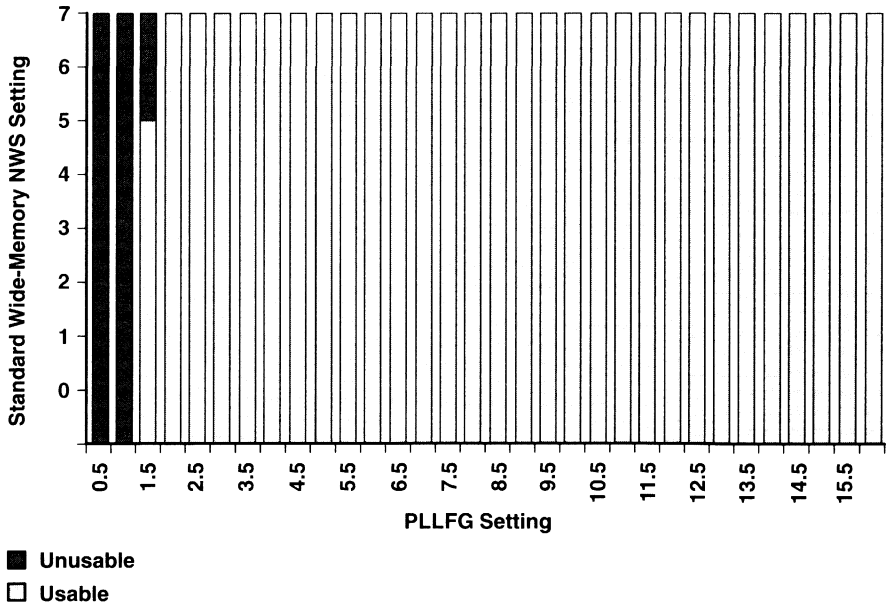


Figure C–6. Standard Wide-Memory NWS Setting Versus PLLFG Setting

(a) With RD = 32



(b) With RD = 64



Appendix D

Customer Information

This appendix includes an example ordering form and mechanical information.

Topic **Page**

D.1 New Product Release Form for the MSP58C80

The new product release form is used to track and document all the steps involved in implementing a new speech code onto one of the parent speech devices. A blank form is provided in this section (note that the addresses on these forms are subject to change). Copy the new product release forms (NPRF) provided or get one from your TI field sales office to initiate the implementation process. As seen on the blank form, Section 1 allows you to choose your own customer part number used for ordering your parts. If no customer part number is indicated, then TI defaults to the MSP58Cxxx part number for ordering purposes. You are required to fill in the Company, Project, Purchase Order #, and Contact fields. After completing Section 1, you must submit the NPRF (along with your speech code) to the speech products group by way of your local TI field sales office.

Once the speech products group receives the speech code and the NPRF, you have completed the initial steps involved in implementing this code onto production devices. Since the MSP58C80 is mask programmable, the speech code must first be converted into a format that the speech products mask vendor can use to generate this new mask. This format is called a PG output. During the creation of the PG output file, the speech code is echoed to a verification file. This file is sent back to you for recheck. This recheck ensures that the PG output file used the correct data. Along with the verification file, the NPRF is also returned to you with Section 2 completed by TI. In this section, TI assigns your own MSP58Cxxx part number and, in the case of packaged devices, TI also proposes a symbol format to you. If you wish to deviate from the suggested symbol format, you must consult TI for requested changes.

After you verify the reconstructed speech code and accept the proposed symbol format, you are required to sign Section 3 as authorization for TI to generate the mask, prototypes, and risk units in accordance with the pertinent purchase order. You then need to send or fax the NPRF to the speech products group by way of the local TI field sales office. TI should have the prototypes shipped to you approximately six weeks after receiving the NPRF with Section 3 signed. Once you receive these prototypes, verify the functionality of the prototypes, sign Section 4, and send the NPRF (with Section 4 signed) back to TI. At this point, you can start ordering production units.

New Product Release Form

SECTION 1. ORDER INFORMATION

This section is to be completed by the customer and sent to TI along with the code.

Division: _____ Company: _____

Project Name: _____

Purchase Order #: _____

Management Contact: _____ Phone: (_____) _____

Technical Contact: _____ Phone: (_____) _____

Customer Part Number: _____

****

SECTION 2A. ASSIGNMENT OF TI PRODUCTION PART NUMBER

This section to be completed by TI.

TI Part Number: _____

SECTION 2B. ASSIGNMENT OF SYMBOLIZATION FORMAT

This section to be completed by TI and approved by the customer (customer approval in Section 3).

TOP SIDE SYMBOLIZATION:

TI		
LOGO YYWW QQQQ	YYWW =	DATE CODE
MSP58CXXXPJM	QQQQ =	LOT NUMBER
{OPTIONAL 12 CHARACTER)		(LAST 4 DIGITS)
	MSP58CXXX =	TI PART NUMBER

****

SECTION 3. AUTHORIZATION TO GENERATE MASKS, PROTOTYPES, AND RISK UNITS

This section is to be completed by the customer and sent to TI after the following criteria have been met:

1) The customer has verified that the TI–generated data matches the original data.

2) The customer approves the symbolization format in Section 2B.

I hereby certify that the TI-generated verification data has been checked and found to be correct, and I authorize TI to generate masks, prototypes, and risk units in accordance with the purchase order in Section 1 above. In addition, I also authorize TI to use the symbolization format illustrated in 2B on all devices with the part number indicated in 2A.

By:_____ Title:_____

Date: _____

****

SECTION 4. APPROVAL OF PROTOTYPES AND AUTHORIZATION TO START PRODUCTION

This section is to be completed by the customer after prototype devices have been received and tested.

I hereby certify that the prototype devices have been received and tested and found to be acceptable, and I authorize TI to start normal production in accordance with purchase order # _____.

By: _____ Title: _____

Date: _____

****

Return to: Texas Instruments Incorporated
Attn: Speech Product Engineer for MSP58C80
P.O. Box 655303, MS 8211
Dallas, TX 75265

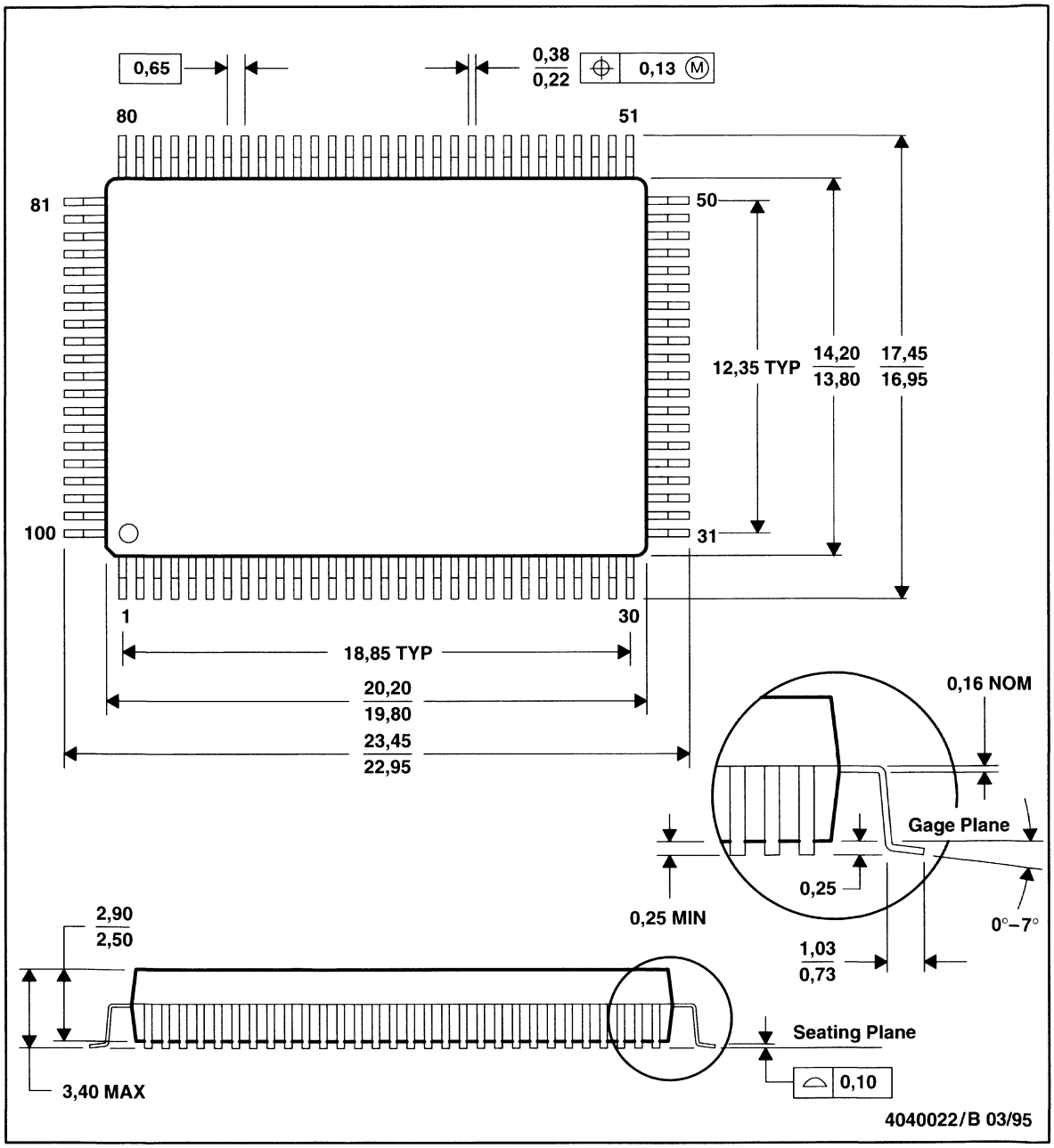
or FAX to: (214) 997–3471
Attn: Speech Product Engineer for MSP58C80

D.2 Mechanical Information

The MSP58C80/C81/C82 is available in a PJM/R plastic quad flatpack package. Figure D–1 gives the mechanical information for the 100-lead plastic quad flatpack package.

Figure D–1.MSP58C80/C81/C82 100-Lead Plastic Quad Flatpack

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



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-022

Appendix E

MSP58C20 Audio-Band Converter Data Sheet

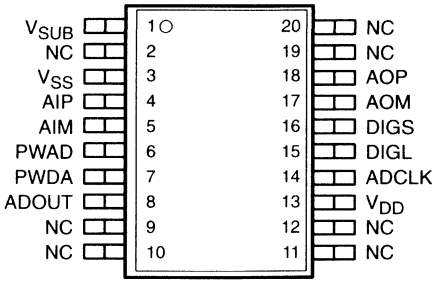
This appendix contains data sheet information for the MSP58C20 audio-band converter.

MSP58C20
AUDIO-BAND CONVERTER

SPSS015B – DECEMBER 1993 – REVISED JULY 1996

- **Analog Portion of ADC and DAC for Audio-Band Signal-Processing Applications**
- **5-V Supply Voltage**
- **Oversampling Second-Order Sigma-Delta Modulator**
- **1.024-MHz Master Clock Frequency**
- **On-Chip Continuous-Time Antialiasing and Smoothing Filters**
- **High-Performance Fully Differential and Symmetrical Analog Data Paths**
- **Internal Reference Voltage and Common-Mode Bias Voltage Generation**
- **Very Low Power Consumption Mode**

DW PACKAGE
(TOP VIEW)



NC – No internal connection

description

The MSP58C20 is the analog portion of an audio-band sigma-delta analog-to-digital and digital-to-analog converter and is a companion part to the MSP58C80. The MSP58C20 is designed to operate only with the MSP58C80, which contains the digital portion of the audio-band converter. The circuit consists of three main blocks: the analog-to-digital converter (ADC), the digital-to-analog converter (DAC), and internal reference and bias voltages.

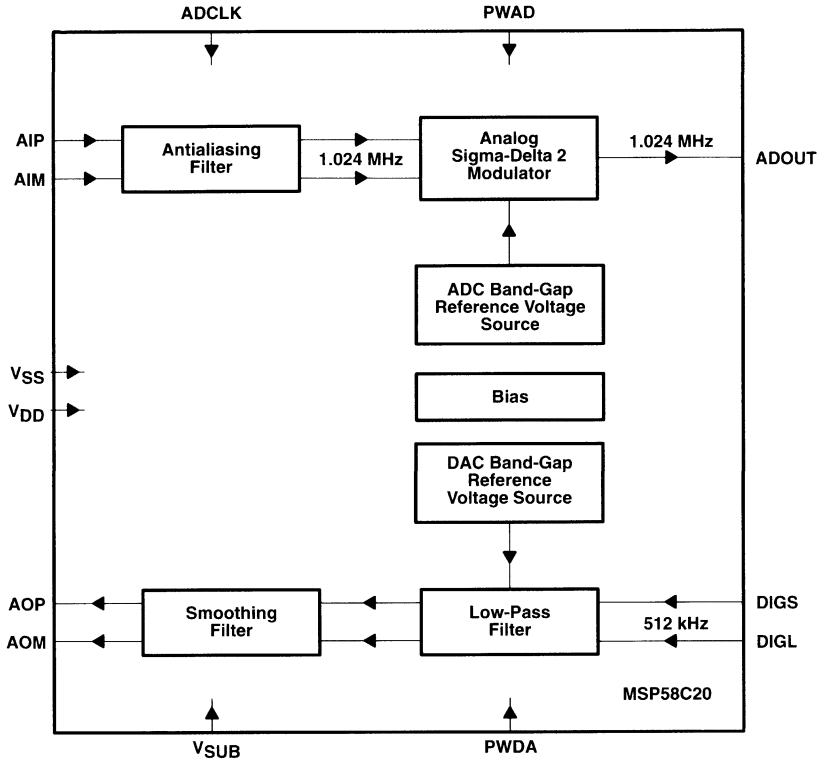
The analog-to-digital conversion chain consists of a continuous-time antialiasing stage, an analog oversampled modulator, and the modulator bias voltage. The antialiasing stage is a second-order low-pass filter with a cutoff frequency of typically 190 kHz. The modulator is a sigma-delta feedback loop, which oversamples the signal at 1.024 MHz and provides second-order noise shaping. It performs the conversion of the differential analog input signal to a pulse-density-modulated single-bit digital output (ADOUT). When a maximum positive differential input voltage (i.e., a maximum positive voltage difference of AIP – AIM) is applied at the AIP and AIM inputs, the resulting code at the ADOUT output is all ones.

The digital-to-analog conversion chain consists of a fast DAC, an analog low-pass filter, and the filter's bias voltage. The two input bits (DIGS and DIGL), sampled at 0.512 MHz from a digital modulator on the MSP58C80, are the inputs of the DAC conversion chain. Based on the values for DIGS (the sign bit) and DIGL (the level bit), the following table shows the DAC voltage steps that are produced.

DIGS	DIGL	DAC VOLTAGE STEPS
L	L	$-1 \times V_{ref}$
L	H	$-2 \times V_{ref}$
H	L	$+1 \times V_{ref}$
H	H	$+2 \times V_{ref}$

When DIGS = L, the AOM analog output has a more positive voltage than AOP. When DIGL = H, the absolute value of the voltage difference between AOP and AOM is greater than when DIGL = L. A band-gap voltage source is used to produce the DAC and ADC reference voltages. These two references are different to avoid crosstalk between the two converters.

functional block diagram



Terminal Functions

TERMINAL NAME	TERMINAL NO.	A/D	I/O	DESCRIPTION
ADCLK	14	D	I	ADCLK is a 1.024-MHz clock input.
ADOUT	8	D	O	ADOUT is the 1-bit output of the ADC modulator and is sampled at 1.024 MHz.
AIM	5	A	I	AIM is a negative differential input for the ADC. AIP and AIM together form a balanced differential input. The biasing of this terminal is fixed through resistors by the internal common-mode voltage source. This terminal can be ac coupled or dc coupled. If the terminal is dc coupled, external common-mode bias should satisfy recommended operating conditions.
AIP	4	A	I	AIP is a positive differential input for the ADC. AIP and AIM together form a balanced differential input. The biasing of this terminal is fixed through resistors by the internal common-mode voltage source. This terminal can be ac coupled or dc coupled. If the terminal is dc coupled, external common-mode bias should satisfy recommended operating conditions.
AOM	17	A	O	AOM is a negative differential DAC output. AOP and AOM together form a balanced differential output. The common-mode voltage at this terminal is fixed by the internal common-mode circuitry.
AOP	18	A	O	AOP is a positive differential DAC output. AOP and AOM together form a balanced differential output. The common-mode voltage at this terminal is fixed by the internal common-mode circuitry.
DIGL	15	D	I	DIGL is the input level bit of the DAC and is sampled at 0.512 MHz.
DIGS	16	D	I	DIGS is the input sign bit of the DAC and is sampled at 0.512 MHz.
PWAD	6	D	I	When PWAD is high, it puts the ADC part of the circuit into a power-down mode. When both PWAD and PWDA are high, the MSP58C20 is in a stable low-power-consumption state.
PWDA	7	D	I	When PWDA is high, it puts the DAC part of the circuit in a power-down mode. When both PWAD and PWDA are high, the MSP58C20 is in a stable low-power-consumption state.
V_{SUB}	1	n/a	n/a	V_{SUB} and V_{SS} must be connected together to minimize substrate currents during power up, power down, and normal operation.
V_{DD}	13	n/a	n/a	V_{DD} is the 5-V power supply.
V_{SS}	3	n/a	n/a	V_{SS} is ground. The internal band-gap voltage and the common-mode bias voltages are referenced to V_{SS}.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage range, V_{DD} (see Note 1) ..... −0.3 V to 6 V
Input voltage range, V_I (any digital or analog input, see Note 1) ..... −0.3 V to V_{DD} + 0.3 V
V_{SUB}, V_{SS} voltage range, relative to each other ..... −30 mV to 30 mV
Operating free-air temperature range, T_A ..... 0°C to 70°C
Storage temperature range, T_{stg} ..... −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to V_{SS} unless otherwise noted.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{DD} (see Note 1)	4.75	5	5.25	V
High-level input voltage, digital inputs, V_{IH} (see Note 1)	2			V
Low-level input voltage, digital inputs, V_{IL} (see Note 1)			0.8	V
Maximum differential input voltage between AIP and AIM (ac or dc peak-to-peak voltage), V_{ID}	−3		3	V
Common-mode input voltage at AIP and AIM, V_{IC} (see Note 1)	$0.45 \times V_{DD}$	$0.5 \times V_{DD}$	$0.55 \times V_{DD}$	V
Input clock frequency, ADCLK		1.024		MHz
Resistive load between AOP and AOM	15			kΩ
Capacitive load at AOP and AOM (at each output versus V_{SS})			50	pF
Operating free-air temperature, T_A	0		70	°C

NOTE 1: All voltage values are with respect to V_{SS} unless otherwise noted.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature, ADCLK input frequency = 1.024 MHz, PWDA = L and PWAD = L (power-up mode) (unless otherwise noted)

supply current characteristics

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{DD}	Supply current	PWAD =H, PWDA = H, Digital inputs = V_{DD} or V_{SS}, Digital output = no load			50	µA
		PWAD = L, PWDA = L	6.5	9	16	mA

analog input characteristics

PARAMETER			TEST CONDITIONS	MIN	TYP	MAX	UNIT
	Transmit dynamic range, maximum differential input voltage (between AIP and AIM)		dc or ac voltage	±2.22	±2.36	±2.5	V
V_{IO}	Transmit differential input offset voltage		See Note 2	–150		150	mV
V_{IC}	Internal common-mode voltage at AIP and AIM			0.4 × V_{DD}	0.5 × V_{DD}	0.6 × V_{DD}	V
z_i	Input impedance	AIP	Between AIP and internal common-mode voltage source (AIM = V_{DD}/2)	15	25	35	kΩ
		AIM	Between AIM and internal common-mode voltage source (AIP = V_{DD}/2)	15	25	35	
	Input capacitance	AIP	Measured at 5 MHz between AIP and V_{SS} (AIM = V_{DD}/2)			50	pF
		AIM	Measured at 5 MHz between AIM and V_{SS} (AIP = V_{DD}/2)			50	

NOTE 2: Calculated by linear regression based on five dc measurements between –1 V and 1 V

digital output characteristics

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	Digital high-level output voltage versus V_{SS}	I_{OH} = 300 µA	2.4			V
V_{OL}	Digital low-level output voltage versus V_{SS}	I_{OL} = 1 mA			0.4	V

analog output characteristics

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OD}	Differential output voltage, dynamic range, AOP to AOM	Balanced loads, dc measurement	±2.82	±3	±3.18	V
V_{OO}	Differential output offset voltage	dc measurement	–150		150	mV
V_{OC}	Common-mode output voltage at AOP and AOM		0.4 × V_{DD}	0.5 × V_{DD}	0.6 × V_{DD}	V

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature, ADCLK input frequency = 1.024 MHz, PWDA = L and PWAD = L (power-up mode) (unless otherwise noted) (continued)

ADC transmit characteristics†

PARAMETER		TEST CONDITIONS			MIN	TYP	MAX	UNIT
	Transmit absolute gain tolerance	V_{DD} = 5 V, T_A = 25°C, Input = 1-kHz sine wave at –13 dBrl					±0.5	dB
	Transmit gain versus input level	Input = 1-kHz sine wave, Gain reference level = gain measured at input level of –13 dBrl, See Note 3	Input level = –1 dBrl to –43 dBrl				±0.25	dB
			Input level = –43 dBrl to –53 dBrl				±0.5	
			Input level = –53 dBrl to –58 dBrl				±1	
	Transmit gain versus supply voltage	V_{DD} = 4.75 V to 5.25 V, Input = 1 kHz at –13 dBrl					±0.15	dB
	Transmit idle channel in-band noise	Psophometrically-weighted output noise, Transmit channel idle				–76		dBrlp
	Transmit idle channel single-frequency noise spectrum (see Note 4)	T_A = 25°C, FFT rectangular window bandwidth = 125 Hz, Transmit channel idle, See Figure 5		f = 50 Hz		–80		dBrl
				f = 300 Hz		–82		
				f = 3.4 kHz		–82		
				f = 4 kHz		–80		
				f = 7 kHz		–72		
				f = 12 kHz		–65		
				f = 20 kHz		–64		
	Transmit single-frequency distortion	Input = one frequency in 0.7-kHz to 1.1-kHz band at –4 dBrl, Measured first two harmonics					–50	dB
	Transmit intermodulation distortion (see Note 4)	Input = two frequencies in 0.3-kHz to 3.4-kHz band, Input levels = –7 dBrl and –24 dBrl, Measured second and third intermodulation products					–40	dBrl
	Transmit-signal-to-total-noise-plus-distortion ratio (see Note 5)	V_{DD} = 5.25 V, T_A = 25°C, Input = 1-kHz sine wave, Measured psophometrically-weighted total noise plus distortion, See Figure 6	Input level = –70 dBrl		–13			dB
			Input level = –20 dBrl		50			
			Input level = –1 dBrl		50			
	Transmit gain variations versus input frequency (see Notes 4 and 6)	f = 0.1 kHz to 4 kHz, Input level = –13 dBrl					±0.6	dB
	Transmit power supply rejection	See Note 7			30			dB
I_{lkg}	Leakage current	Voltage applied to terminal is between V_{SS} and V_{DD}, PWDA = H (power-down mode)		AIP	–10		10	µA
				AIM	–10		10	
	Receive-to-transmit crosstalk	Receive input = one frequency in 0.3-kHz to 3.4-kHz band at –3 dBrl, Crosstalk measured at transmit digital output, Transmit channel idle					–70	dB

† This table contains specifications in which the power levels are expressed in dBrl; dBrl stands for dB above reference level. 0 dBrl is the ADC theoretical overload point. This overload point corresponds to a sine wave at the input of the modulator with peak amplitude equal to 2.25 V dBrlp is a psophometrically-weighted value being compared against a psophometrically-weighted reference.

NOTES: 3. Input satisfies CCITT G.714 15.3, Method 2.
4. This parameter is characterized but not tested.
5. Input satisfies CCITT G.714 14.3, Method 2.
6. Gain is relative to gain at 1 kHz.
7. The power-supply rejection measurement is made with a 50-mVrms, 0- to 20-kHz signal applied to V_{DD} and with the transmit channel idle.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature, ADCLK input frequency = 1.024 MHz, PWDA = L and PWAD = L (power-up mode) (unless otherwise noted) (continued)

DAC receive characteristics†

PARAMETER	TEST CONDITIONS		MIN	TYP	MAX	UNIT
Receive gain tolerance	V_{DD} = 5 V, T_A = 25°C, Input = 1-kHz sine wave at –28 dBrl				±0.5	dB
Receive gain versus input level	Input = 1-kHz sine wave, Gain reference level = gain measured at input level of –28 dBrl, See Note 8	Input level = –1 dBrl to –43 dBrl			±0.25	dB
		Input level = –43 dBrl to –53 dBrl			±0.5	
		Input level = –53 dBrl to –58 dBrl			±1	
Receive gain versus supply voltage	V_{DD} = 4.75 V to 5.25 V, Digital input = 1-kHz sine wave at –28 dBrl				±0.15	dB
Receive idle channel in-band noise	Receive channel idle, Psophometrically-weighted output noise			–75		dBrlp
Receive idle channel single-frequency noise spectrum (see Note 4)	T_A = 25°C, Receive channel idle, Measurement bandwidth = 125 Hz, See Figure 6	f = 100 Hz		–82		dBrl
		f = 3 kHz		–82		
		f = 10 kHz		–64		
		f = 100 kHz		–64		
Receive single-frequency distortion	Input = one frequency in 0.7-kHz to 1.1-kHz band at –6 dBrl, Measured first two harmonics				–50	dB
Receive intermodulation distortion (see Note 4)	Input = two frequencies in 0.3-kHz to 3.4-kHz band, Input levels = –7 dBrl and –24 dBrl, Measured second and third intermodulation products				–40	dBrl
Receive signal-to-total-noise-plus-distortion ratio (see Note 9)	V_{DD} = 5.25 V, T_A = 25°C, Input = 1-kHz sine wave, Measured psophometrically-weighted total noise plus distortion, See Figure 7	Input level = –70 dBrl	0			dB
		Input level = –20 dBrl	50			
		Input level = –1 dBrl	50			
Receive gain variations versus input sine wave frequency (see Note 6)	V_{DD} = 4.75 V, T_A = 25°C, Input level = –13 dBrl, See Figure 9	f = 156 Hz to 4 kHz	–0.6‡		0.6	dB
		f = 4.6875 kHz	–0.7		–0.4	
		f = 6.25 kHz	–1.75		–1.4	
		f = 7.8125 kHz	–3.35		–2.9	
		f = 9.375 kHz	–5.25		–4.8	
		f = 10.9375 kHz	–7.25		–6.8	
		f = 12.5 kHz	–9.2		–8.7	
		f = 15.625 kHz	–12.8		–12.2	
Receive power supply rejection	See Note 10		30			dB

† This table contains specifications in which the power levels are expressed in dBrl; dBrl stands for dB above reference level. 0 dBrl is the DAC overload point. Overload levels of the digital modulator (see parameter measurement information) are 32767 and –32767 peak values. The 0-dBrl level is related to maximum differential output voltage, which is typically 2.25 V.

‡ The algebraic convention, in which the less positive (more negative) limit is designated as minimum, is used in this data sheet for receive gain variations versus input sine-wave frequency.

NOTES: 4. This parameter is characterized but not tested.
6. Gain is relative to gain at 1 kHz.
8. Input satisfies CCITT G.714 15.4 Method 2.
9. Input satisfies CCITT G.714 14.4 Method 2.
10. The power supply rejection measurement is made with a 50-mVrms, 0-kHz to 20-kHz signal applied to V_{DD} and with the receive channel idle.

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature, ADCLK input frequency = 1.024 MHz, PWDA = L and PWAD = L (power-up mode) (unless otherwise noted) (continued)

DAC receive characteristics (continued)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_{lkg}	Leakage current	AOP	–10		10	µA
		AOM	–10		10	
	Output impedance, differential, between AOP and AOM (see Note 4)		30			kΩ
	Transmit-to-receive crosstalk	Transmit input = one frequency in 0.3-kHz to 3.4-kHz band at –3 dBrl, Receive channel idle, Crosstalk measured at receive analog output			–70	dB

NOTE 4. This parameter is characterized but not tested.

timing requirements over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{su1}	Transmit setup time at power up (PWAD transition from H to L)	ADCLK input frequency = 1.024 MHz, See Note 11		20		µs
t_{su2}	Receive setup time at power up (PWDA transition from H to L)	ADCLK input frequency = 1.024 MHz, See Note 12		20		µs
t_{su3}	Receive setup time, DIGS or DIGL setup before ADCLK↑	See Figure 4	50			ns
t_h	Receive hold time, DIGS or DIGL hold after ADCLK↑	See Figure 4	50			ns
t_c	Cycle time, ADCLK			1		µs
t_{w1}	Pulse duration, ADCLK high		470			ns
t_{w2}	Pulse duration, ADCLK low		470			ns
t_f	Fall time, ADCLK				20	ns
t_r	Rise time, ADCLK				20	ns

NOTES: 11. After the setup time, the transmit channel displays normal operating characteristics.
12. After the setup time, the receive channel displays normal operating characteristics.

switching characteristic over recommended ranges of supply voltage and operating free-air temperature

PARAMETER		TEST CONDITION	MIN	TYP	MAX	UNIT
t_a	Transmit access time, ADOUT after ADCLK↑ (see Note 4)	See Figure 3			100	ns

NOTE 4. This parameter is characterized but not tested.

PARAMETER MEASUREMENT INFORMATION

The receive characteristics in the electrical characteristics table are measured by activating the MSP58C20 receive path through a digital modulator. This modulator consists of two functional blocks (see Figure 1 and Figure 2) connected in series. The output of the decoder (see Figure 2) is shown in Table 1.

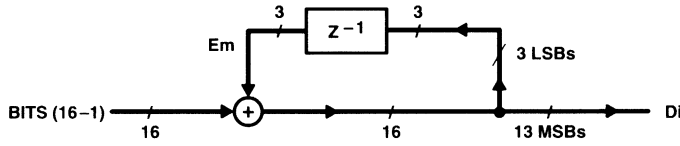


Figure 1. 16- to 13-Bit Modulator at 512-kHz Sampling Rate

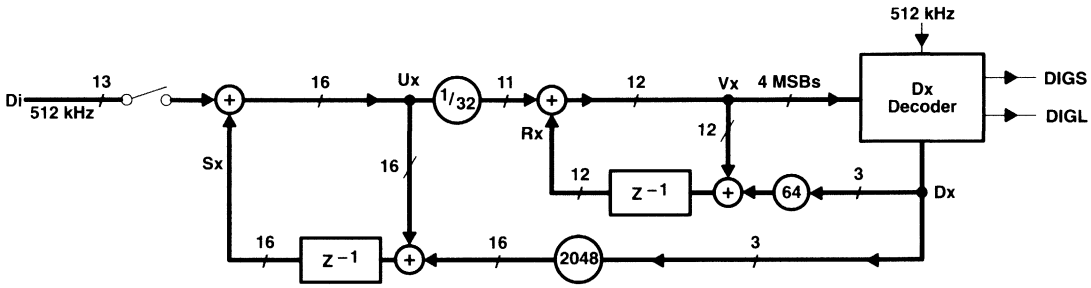


Figure 2. Sigma-Delta-2 Modulator at 512-kHz Sampling Rate

Table 1. Dx Decoder

DECODER INPUT				DECODER OUTPUT				
Vx (11)	Vx (10)	Vx (9)	Vx (8)	Dx (2)	Dx (1)	Dx (0)	DIGS	DIGL
0	1	X	X	H	H	L	L	H
0	0	1	X	H	H	L	L	H
0	0	0	1	H	H	L	L	H
0	0	0	0	H	H	H	L	L
1	1	1	1	L	L	H	H	L
1	1	1	0	L	H	L	H	H
1	1	0	X	L	H	L	H	H
1	0	X	X	L	H	L	H	H

PARAMETER MEASUREMENT INFORMATION

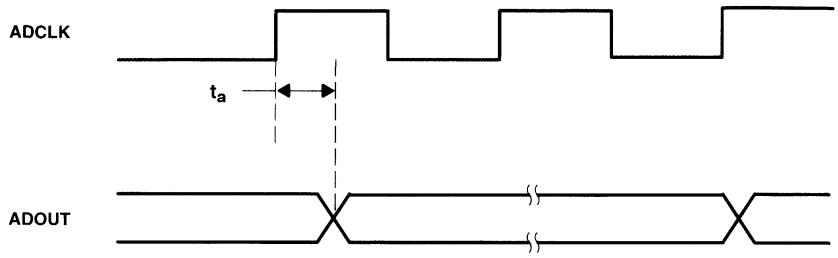


Figure 3. Transmit Access Timing Waveforms

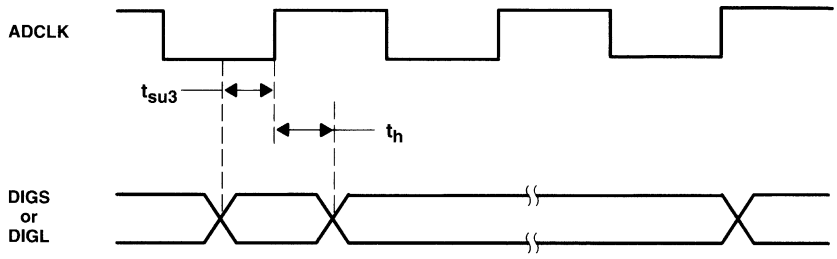


Figure 4. Receive Setup and Hold Time Waveforms

TYPICAL CHARACTERISTICS

**TRANSMIT IDLE CHANNEL
SINGLE-FREQUENCY NOISE SPECTRUM**

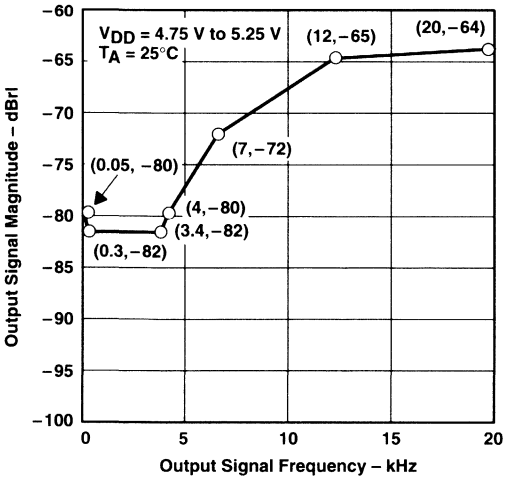
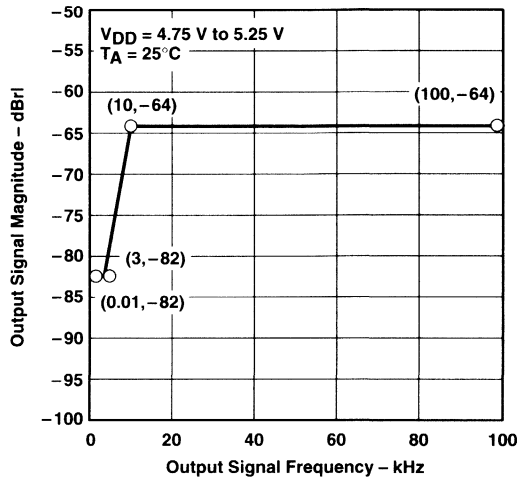


Figure 5

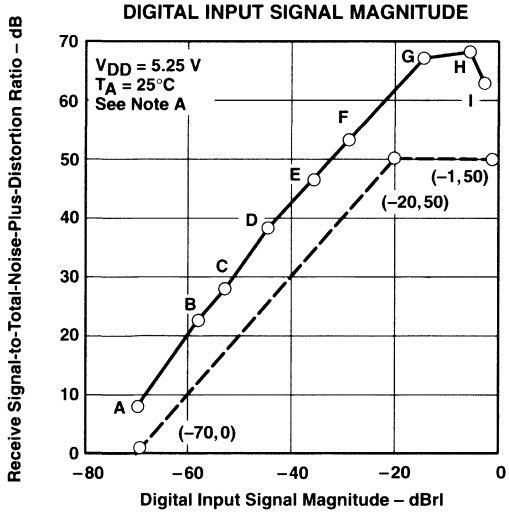
**RECEIVE IDLE CHANNEL†
SINGLE-FREQUENCY NOISE SPECTRUM**



† This parameter is characterized but not tested.

Figure 6

**RECEIVE SIGNAL-TO-TOTAL-NOISE-PLUS-DISTORTION RATIO
vs
DIGITAL INPUT SIGNAL MAGNITUDE**



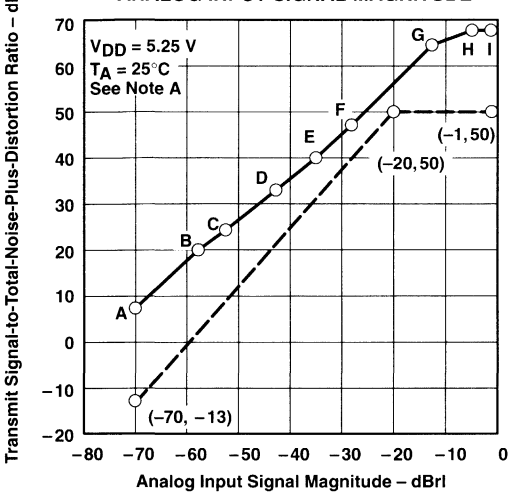
NOTE A: The three points on the dashed line are minimum qualification standards, which every MSP58C20 must pass. The curve shows empirical data from a representative lot.

SET OF POINTS	LOCATION
A	(–70,9)
B	(–58, 23)
C	(–53, 28)
D	(–43, 38)
E	(–35, 46)
F	(–28, 53)
G	(–13, 67)
H	(–5, 69)
I	(–1, 64)

Figure 7

TYPICAL CHARACTERISTICS

TRANSMIT SIGNAL-TO-TOTAL-NOISE-PLUS-DISTORTION RATIO vs ANALOG INPUT SIGNAL MAGNITUDE

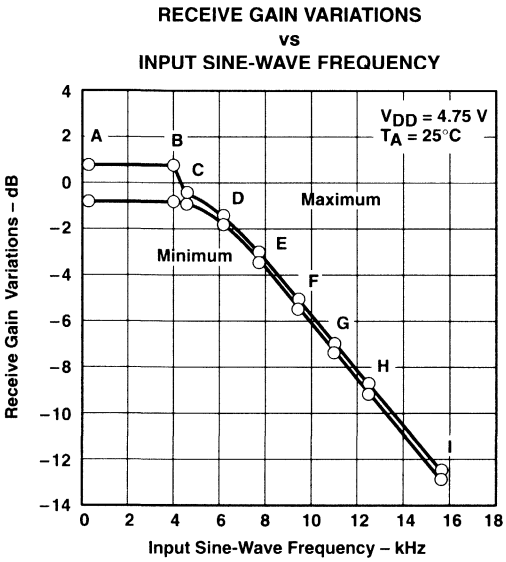


SET OF POINTS	LOCATION
A	(–70,8)
B	(–58, 20)
C	(–53, 24)
D	(–43, 32)
E	(–35, 40)
F	(–28, 48)
G	(–13, 65)
H	(–5, 69)
I	(–1, 69)

NOTE A. The three points on the dashed line are minimum qualification standards, which every MSP58C20 must pass. The curve shows empirical data from a representative lot.

Figure 8

MAXIMUM AND MINIMUM CHARACTERISTICS



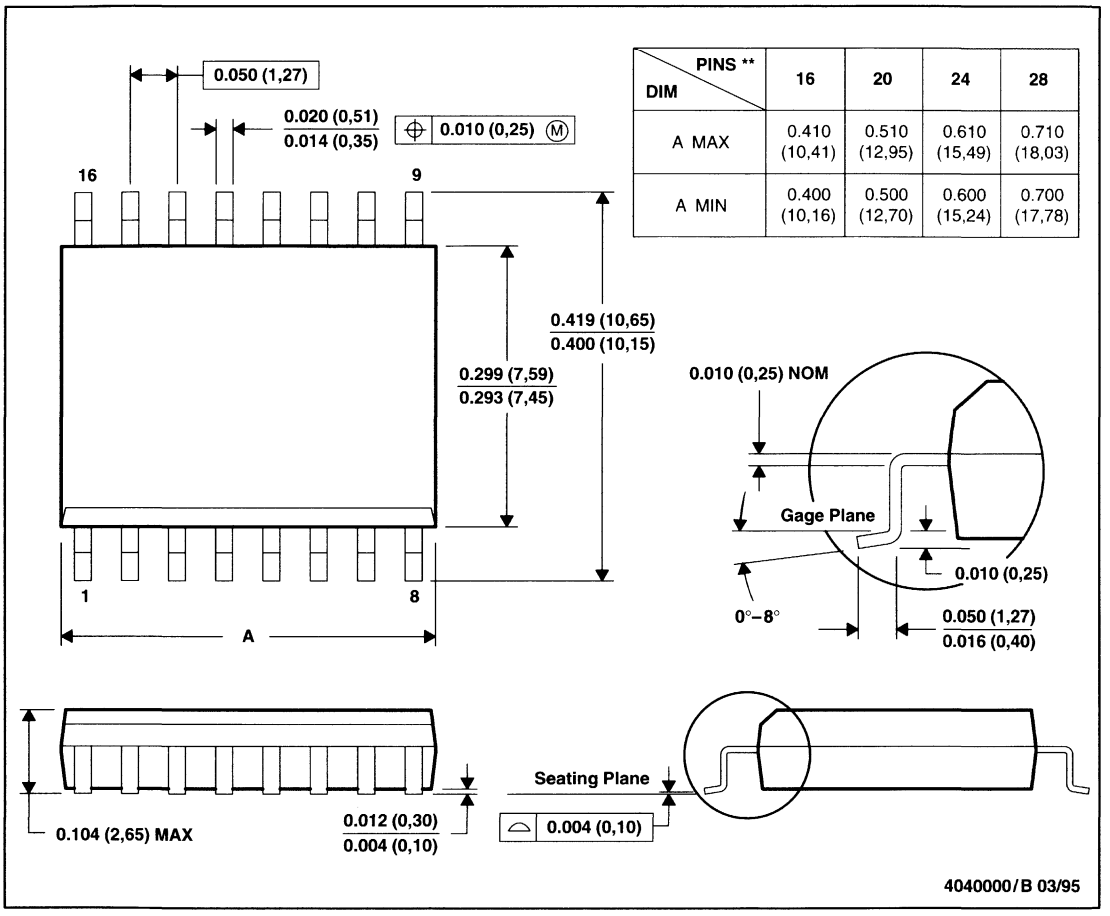
SET OF POINTS	MIN	MAX
A	(0.156, −0.6)	(0.156, 0.6)
B	(4, −0.6)	(4, 0.6)
C	(4.6875, −0.7)	(4.6875, −0.4)
D	(6.25, −1.75)	(6.25, −1.4)
E	(7.8125, −3.35)	(7.8125, −2.9)
F	(9.375, −5.25)	(9.375, −4.8)
G	(10.9375, −7.25)	(10.9375, −6.8)
H	(12.5, −9.2)	(12.5, −8.7)
I	(15.625, −12.8)	(15.625, −12.2)

Figure 9

MECHANICAL DATA

DW (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

16 PIN SHOWN



DIM \ PINS **	16	20	24	28
A MAX	0.410 (10,41)	0.510 (12,95)	0.610 (15,49)	0.710 (18,03)
A MIN	0.400 (10,16)	0.500 (12,70)	0.600 (15,24)	0.700 (17,78)

NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-013

Appendix F

MSP58P80/81 EPROM Programming

This appendix describes the MSP58P80/81 EPROM cell. The MSP58P80/81 incorporates a one-time programmable (OTP) 32K × 16-bit EPROM designed to function as a 64K × 8-bit EPROM (TMS27PC512) when used with a programmer adapter. This gives the MSP58P80/81 capabilities in the areas of prototyping, early field testing, and production.

Key features of the EPROM cell include standard programming techniques with verification capability of all bits. A programmer adapter (TI part number DPA58P80) provides the 100-pin to 28-pin conversion that is necessary when programming the MSP58P80/81. This programmer adapter was designed to work with the Data I/O™ Chiplab™ programmer. Other programmers have not yet been evaluated, but may provide the same functionality.

This appendix describes programming and verification. The major topics are as follows:

Topic **Page**

Note: MSP58P81 Product Status

The MSP58P81 is in the preliminary stage of development. All information about this product contained in this document is considered **PRODUCT PREVIEW**.

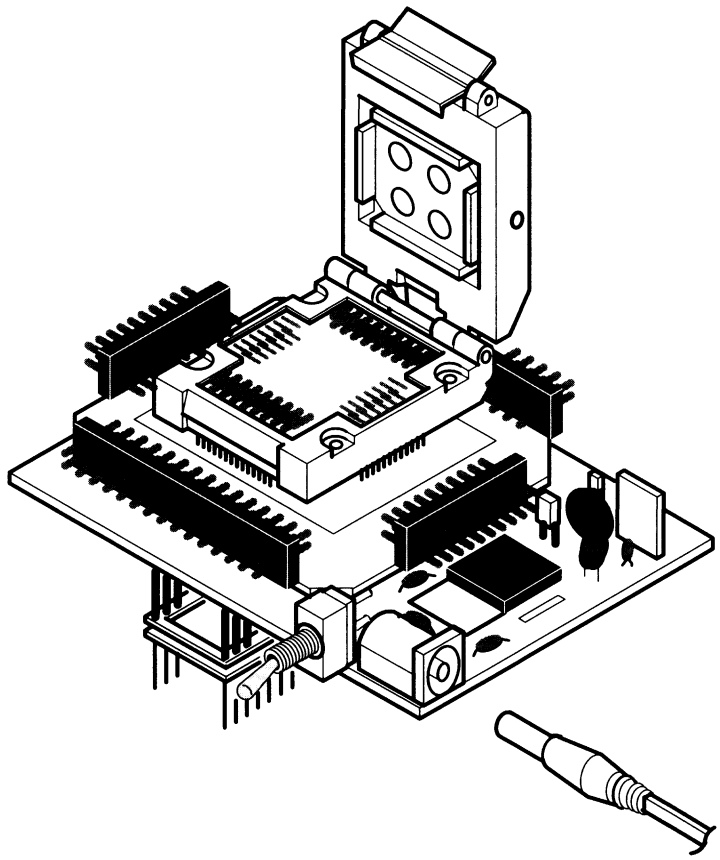
PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

F.1 Using the EPROM Programmer Adapter Socket

Most EPROM programmers have a 28-pin DIP-type socket for use with EPROM devices such as the TMS27PC512. In order to use this type of programmer to program a MSP58P80/81 100-pin QFP, you must use a special adapter that converts the programmer socket into a socket that can accept a MSP58P80/81 device.

Figure F–1 shows an example of the programmer adapter so that you can see the socket for the device and the portion that plugs into the EPROM programmer.

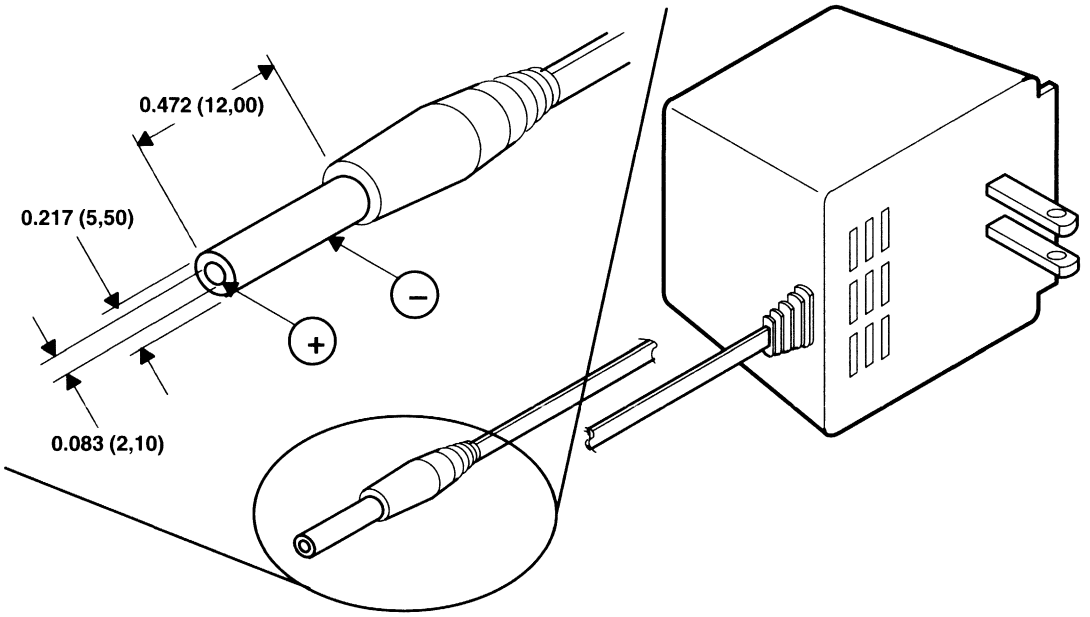
Figure F–1. EPROM Programmer Adapter



F.1.1 Supplying External Power

In most cases, the EPROM programmer cannot supply the V_{CC} power needs of the MSP58P80/81 device, so it becomes necessary to supply external V_{CC}. The programmer adapter is shipped from the factory with an ac adapter. The ac adapter that is provided operates on 120 Vac, 60 Hz power and provides 12 Vdc, 800 mA to the programmer adapter. Figure F–2 shows the size and polarity of the dc connector.

Figure F–2. Size and Polarity of DC Connector on AC Adapter



Note: All linear dimensions are in inches (millimeters).

Due to the MSP58P80/81 uses dynamic logic for much of its internal circuitry, the I_{CC} requirements for the V_{CC} are significantly greater than a typical TMS27PC512-type EPROM. As a result, many EPROM programmers sense this condition and erroneously indicate that the chip is plugged in backwards. To prevent this from occurring, the adapter provides the proper V_{CC} from an ac adapter and on-board voltage regulator. This effectively bypasses the EPROM programmer I_{CC} test and allows the device to be programmed.

Note: Pin Continuity Test

It may be necessary to disable the pin continuity test before programming an MSP58P80/81.

Some EPROM programmers do a pin continuity test by connecting a current-limited power source to each pin before V_{CC} is applied. A voltage measurement is made to determine if an open circuit condition exists on that pin. This pin continuity test may fail because the MSP58P80/81 V_{CC} is provided by an external source and the input pins have a high input impedance when V_{CC} is applied. Therefore, it may be necessary to disable the pin continuity test before programming an MSP58P80/81.

F.2 Programming and Verification

The MSP58P80/81 EPROM cell is similar to the TMS27PC512 64K × 8-bit EPROM. Their memories can be electrically programmed by using the same family and device codes. The MSP58P80/81, like the TMS27PC512, requires a high-voltage source for programming. The programmer should apply the standard 13-V source to the $\overline{G}/V_{PP}$ terminal. However, the programmer adapter reduces this voltage by two diode drops. This satisfies the requirements of the MSP58P80/81 to have 11.5 V at the V_{PP} terminal during programming. All other programming signals are TTL level. Locations may be systematically or randomly programmed as a singular or blocked address. Figure F–3 shows the EPROM programming data format.

Figure F–3. EPROM Programming Data Format

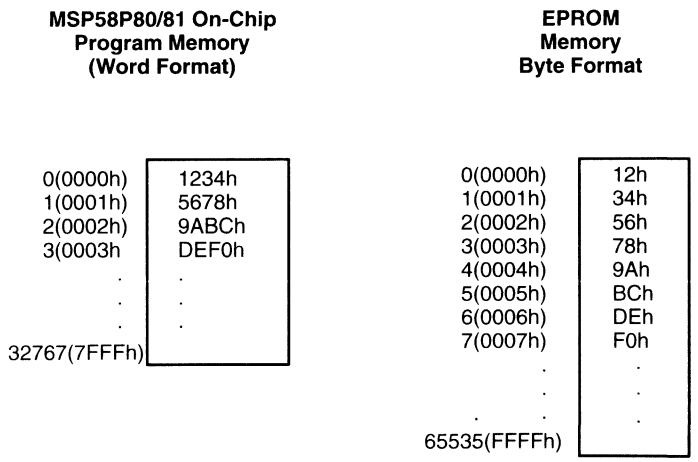


Table F–1 contains the nomenclature and description of each terminal. Figure F–4 shows the wiring diagram of the MSP58P80/81 programmer adapter.

Table F–1. Terminal Nomenclature for the MSP58P80/81 and TMS27PC512

Signals			I/O	Definition
MSP58P80	**MSP58P81**	**TMS27PC512**		
A15 (MSB)–A0 (LSB)	A15 (MSB)–A0 (LSB)	A15 (MSB)–A0 (LSB)	I	EPROM programming address lines
MP/$\overline{MC}$	MP/$\overline{MC}$	$\overline{E}$	I	EPROM chip select
V_{SS} (terminal 49 only)	V_{SS} (terminal 49 only)	$\overline{G}/V_{PP}$	I	EPROM output enable/programming power supply
V_{SS} (all others)	V_{SS} (all others)	GND	I	Ground
V_{DD}	V_{DD}	V_{CC}	I	EPROM power supply
D7 (MSB)–D0 (LSB)	C7 (MSB)–C0 (LSB)	DQ7 (MSB) – DQ0 (LSB)	I/O	Data inputs (programming)/outputs

Figure F–4. MSP58P80/81 EPROM Conversion to TMS27PC512 EPROM Pinout

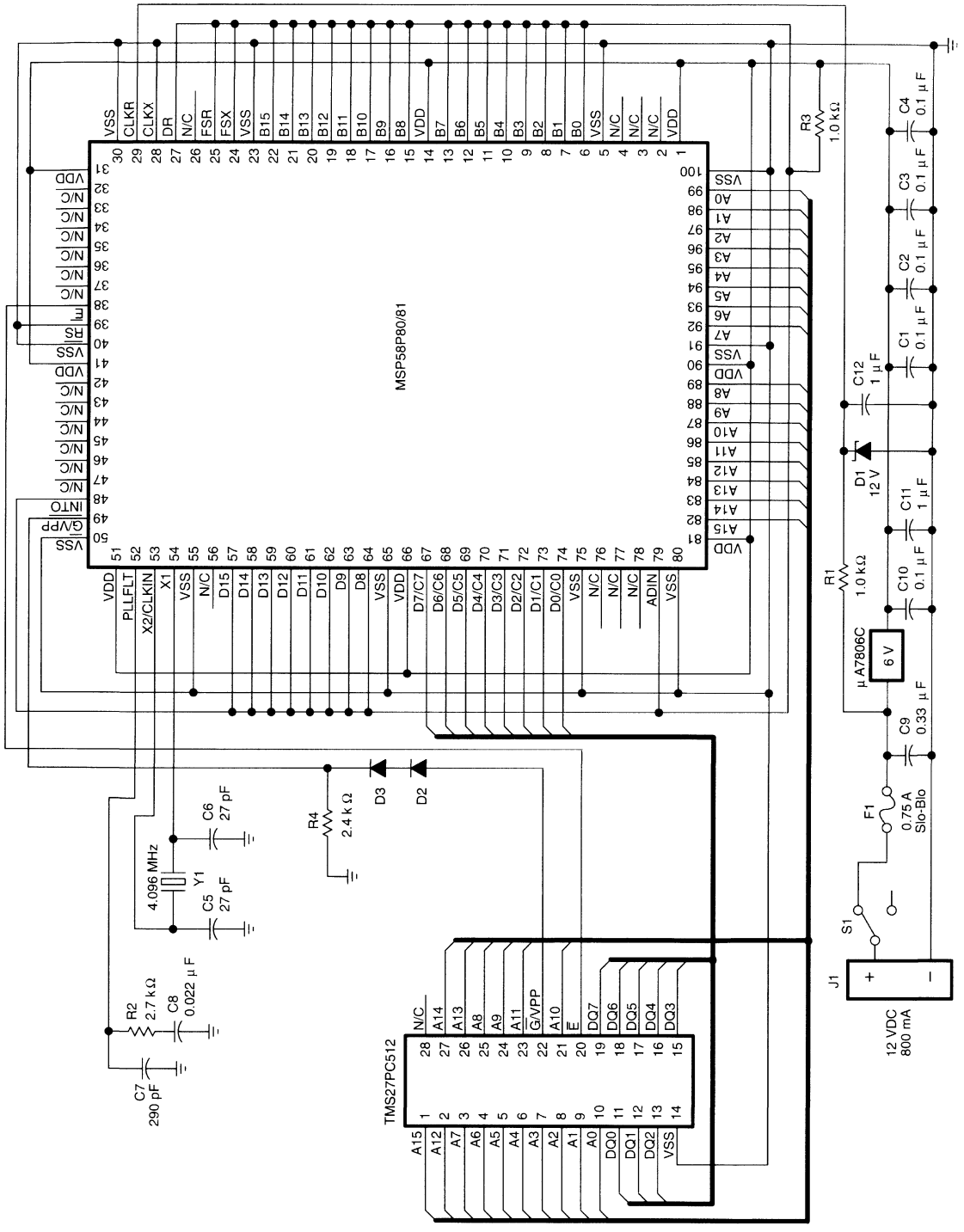


Table F–2 shows the seven modes of operation and the MSP58P80/81-to-TMS27PC512 pinout conversion. All inputs are TTL level except for $\overline{G}/V_{PP}$ during programming and A9 during signature mode. Following the table are individual descriptions of each mode.

Table F–2. MSP58P80/81 Programming Mode Levels

Signal Name†	MSP58P80/81 Terminal	TMS27PC512 Terminal	Mode							
			Read	Output Disable	Stand-by	Program	Verify	Program Inhibit	Signature	
A0	99	10	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	V_{IL}	V_{IH}
A8–A1	89, 92–98	25, 3–9	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	V_{IL}	V_{IL}
A9	88	24	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	VH‡	VH‡
A15 – A10	82–87	1,27,26 2,23,21	ADDR	ADDR	ADDR	ADDR	ADDR	ADDR	V_{IL}	V_{IL}
DQ7 – DQ0	67–74	19–15,13–11	Q_{OUT}	Hi-Z	Hi-Z	D_{IN}	Q_{OUT}	Hi-Z	MFG	Device
									97	85
$\overline{E}$	38	20	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{IL}
$\overline{G}/V_{PP}$	49	22	V_{IL}	V_{IH}	X	V_{PP}	V_{IL}	V_{PP}	V_{IL}	V_{IL}
V_{CC}	1,14,31,41 51,66,81,90	28	V_{CC}	V_{CC}	V_{CC}	$V_{CC}+1$	$V_{CC}+1$	$V_{CC}+1$	$V_{CC}+1$	$V_{CC}+1$

LEGEND:

† = MSP58P80/81 EPROM programming mode produces these TMS27PC512 signals.
‡ = VH is defined as 12 V ± 0.5 V
ADDR = byte address bit
D_{IN} = byte to be programmed at ADDR
Q_{OUT} = byte stored at ADDR
V_{CC} = 5 ± 0.25 V. This is standard power supply voltage.
V_{CC} + 1 = 6 ± 0.25 V. This is provided by the programmer adapter on-board regulator and ac adapter.
V_{IH} = TTL high level
V_{IL} = TTL low level
V_{PP} = 13 ± 0.25 V at the TMS27PC512 terminal, but the programmer adapter reduces this voltage to 11.5 ± 0.25 V for the MSP58P80/81.
X = do not care

F.2.1 SNAP! Pulse Programming

The EPROM can be programmed by using the TI SNAP! pulse programming algorithm as illustrated in the flowchart of Figure F–5, programming time is greatly reduced to a nominal duration of ten seconds. Actual programming time varies as a function of the programmer that is being used. Data is presented in parallel (eight bits) on terminals DQ7 – DQ0. Once addresses and data are stable, $\overline{E}$ is pulsed.

The SNAP! pulse programming algorithm uses pulses of 100 microseconds, followed by a byte verification to determine if the addressed byte has been successfully programmed. Up to ten 100-microsecond pulses per byte are verified before a failure is recognized.

The programming mode is achieved when V_{PP} = 13.0 V, V_{CC} = 6 V, $\overline{G} = V_{IH}$, and $\overline{E} = V_{IL}$. Locations may be programmed in any order. When the SNAP! pulse programming routine has been completed, all bits are verified.

F.2.2 Program Verify

Programmed bits may be verified with $\overline{G}/V_{PP} = V_{IL}$ and $\overline{E} = V_{IL}$. Figure F–6 shows the timing of the program and verification operations for SNAP! pulse programming.

Figure F–5. SNAP! Pulse Programming Flowchart

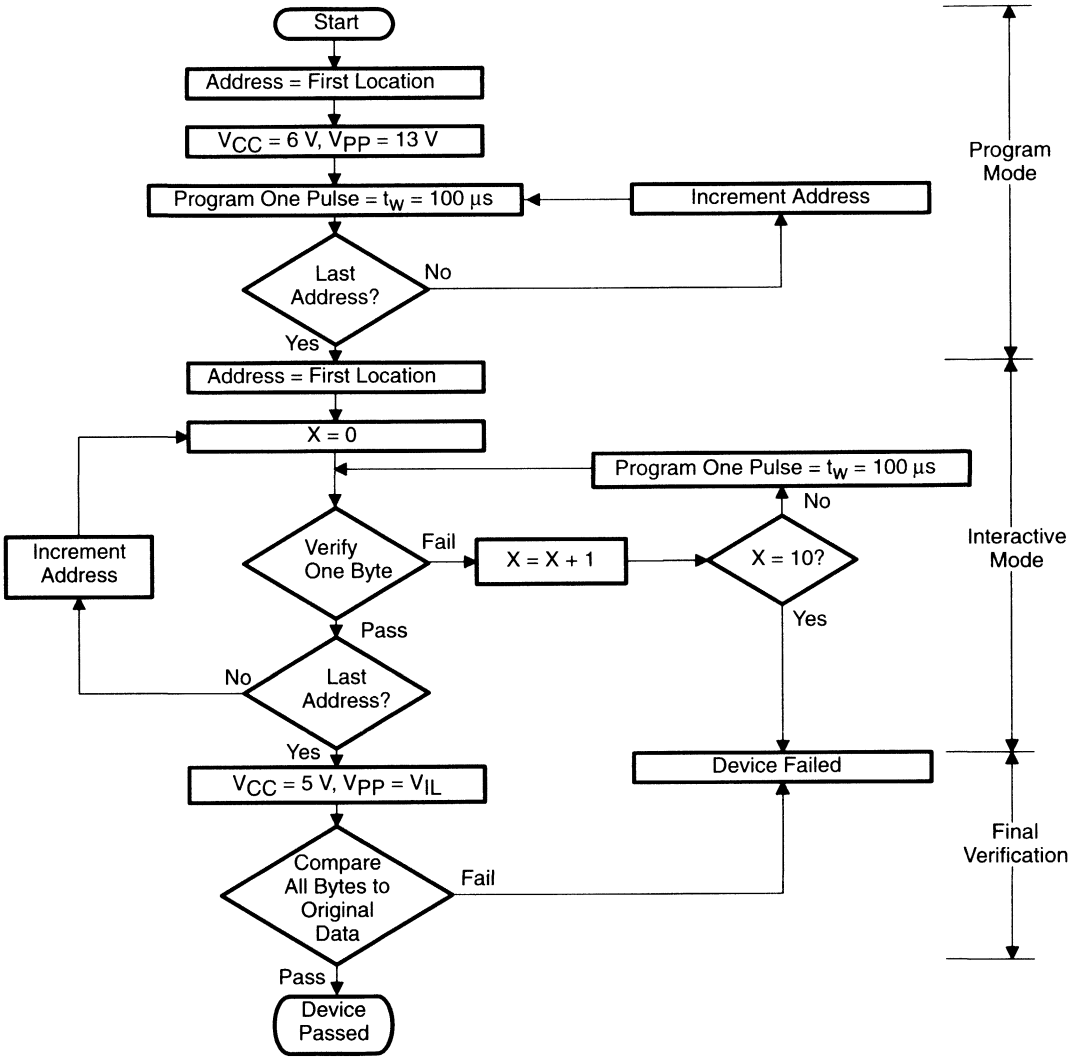
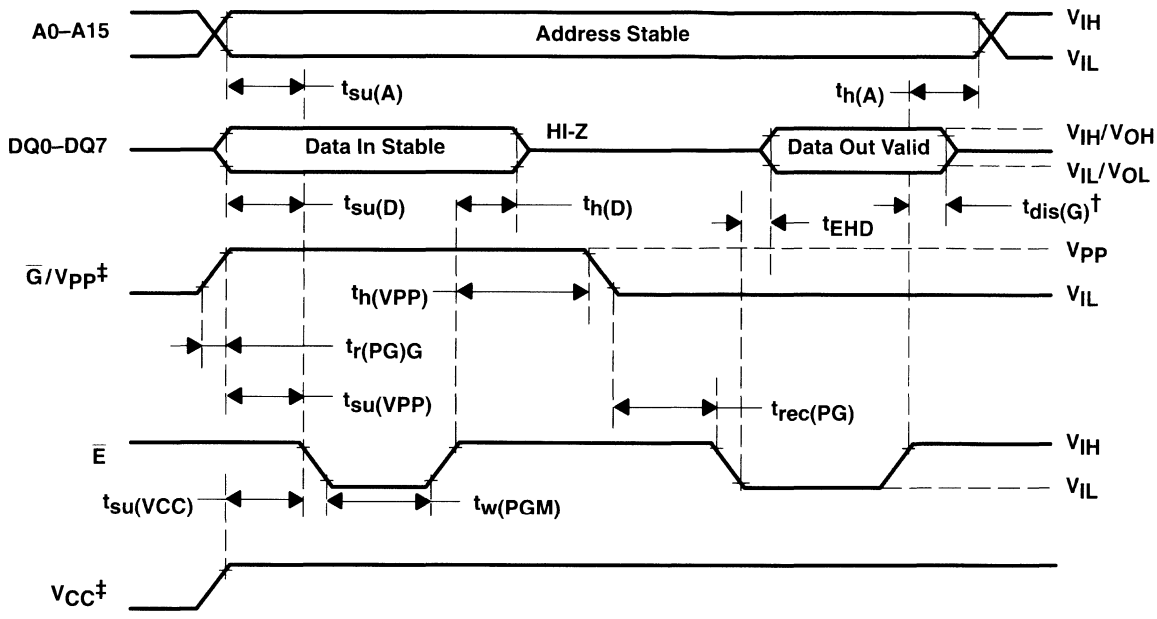


Figure F–6. Programming Timing



† $t_{dis(G)}$ is a characteristic of the device but must be accommodated by the programmer.
‡ 13-V at $\overline{G}/V_{PP}$ and 6-V at V_{CC} for SNAP! Pulse programming.

F.2.3 Program Inhibit

Programming can be inhibited by maintaining a high-level input on the $\overline{E}$ terminal.

F.2.4 Read/Output Disable

When the outputs of two or more devices are connected in parallel on the same bus, the output of any particular device in the circuit can be read with no interference from competing outputs of the other devices. To read the output of a single device, a low-level signal is applied to the $\overline{E}$ and $\overline{G}/V_{PP}$ terminals. All other devices in the circuit should have their outputs disabled by applying a high-level signal to one of the $\overline{E}$ and $\overline{G}/V_{PP}$ terminals. Output data is accessed at terminals DQ0 through DQ7.

F.2.5 Standby

Active I_{CC} supply current can be reduced by applying a high TTL/CMOS signal to the $\overline{E}$ terminal. In this mode all outputs are in the high-impedance state.

F.2.6 Signature Mode

The signature mode provides access to a binary code identifying the manufacturer and type. This mode is activated when A9 is forced to 12 V ± 0.5 V. Two identifier bytes are accessed by A0 (i.e., A0 = V_{IL} accesses the manufacturer code, which is output on DQ0 – DQ7, A0 = V_{IH} accesses the device code, which is output on DQ0 – DQ7). All other addresses must be held at V_{IL}. The manufacturer code for these devices is 97h and the device code is 85h.

F.3 Step-By-Step Procedure

The following procedure shows you how to load a file, set up the Chiplab programmer, and program a MSP58P80/81 with the DPA58P80 programmer adapter.

Step 1: Create a file named 58P80.cmd and edit this file to include the following data:

```
/*****************************************************************/
/*      DSPHEX Command File for use with the MSP58P80/81         */
/*****************************************************************/
Filename.out      /* Input COFF file           */
-i                /* Intel hex format          */
-byte             /* 8-bit programming         */
-romwidth 16      /* 16-bit data width         */
-image            /* Fill all memory addresses */
-fill 0x0000      /* Specify a fill of zeros   */

ROMS              /* Specify 8000 hex words to be programmed */
{
    P80:            origin=0x0000,length=0x8000
}
```

Step 2: Type the command and file name **DSPHEX 58P80.cmd** at the DOS prompt to convert Filename.out to Filename.i0. Verify Filename.out and 58P80.cmd are located in the current directory and that DSPHEX.exe is located in a directory included in the path statement.

Step 3: Install Data I/O Chiplab programmer according to the instructions provided by Data I/O. Do not install the device programmer adapter (DPA58P80) until the Chiplab programmer has completed the power-up
sequence.

Step 4: Click the mouse on **Device**, then on **Select**. The device type selection menu is presented. Select **TI** in the manufacturer block. Select **27PC512** in the Device block. Click the mouse on **Select**.

Step 5: Click the mouse on **File**, then on **Open**. Enter the source file name. Change the File Format to **Intel Hex** if necessary. Choose **Specific** in the Automatic RAM Fill block, and enter **00** as the fill data. Click the mouse on **Read**.

Step 6: Click the mouse on **Data**, then on **Data editor**, then on **Edit User Data**. Verify that the data is loaded correctly. Click the mouse on **Exit**.

Step 7: Click the mouse on **Config**, then on **General Parameters**. Disable **Continuity Checking**. Click the mouse on **Close**.

Step 8: Install the DPA58P80 into the Chiplab programmer as if it were a 28-terminal EPROM. Connect the 12-VDC power source to the DPA58P80. Making sure the DPA58P80 power switch is in the Off position, plug the power source into a 110 VAC 60 Hz outlet.

Step 9: Place the MSP58P80/81 into the socket. Verify the device is inserted correctly. It should be rotated so that terminal 1 is in the corner labeled pin #1 on the PWB of the DPA58P80. Close the socket and toggle the DPA58P80 power switch to on.

Step 10: Click the mouse on **Device** and then on **Program Device**. Verify that the User Data Size is correct. Click the mouse on **Program**.

Step 11: Toggle the DPA58P80 power switch to off and remove the MSP58P80/81.

Appendix G

MSP58C81/C82 Mixed-Signal Processor

The purpose of this chapter is to supply information about the MSP58C81/C82 that is not given in or that is different from the description of the MSP58C80 in chapters 1 – 5 of this document. This chapter contains only a small fraction of the information needed to use the MSP58C81/C82.

Topic **Page**

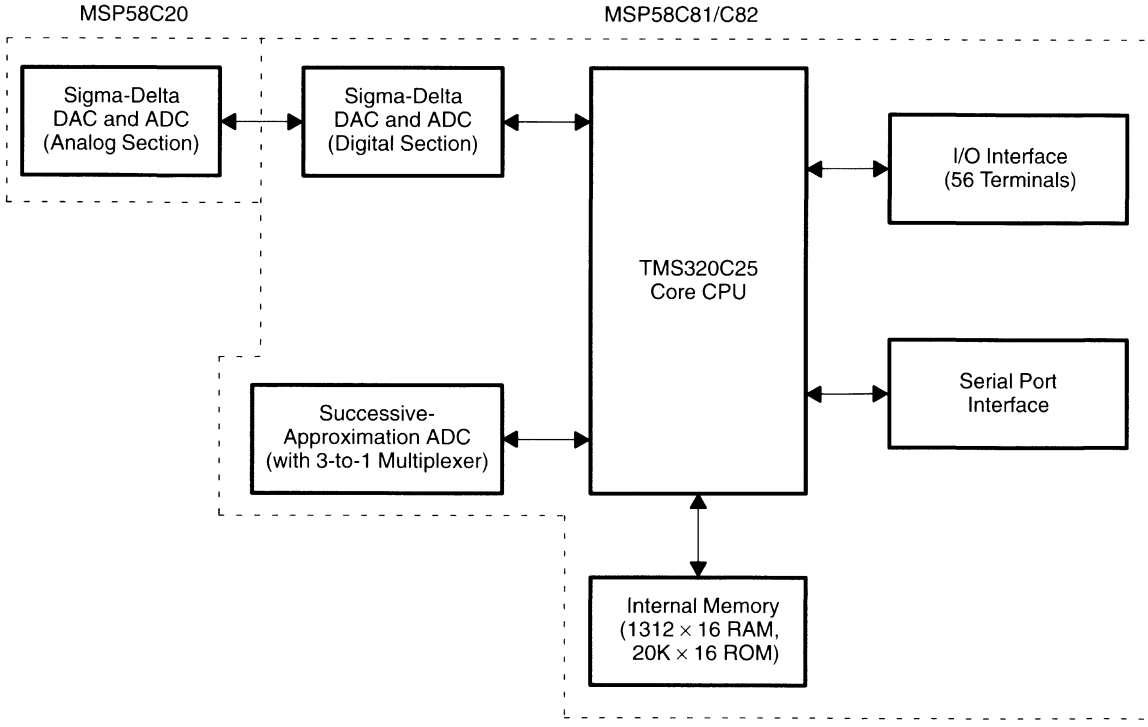
G.1 Introduction

The MSP58C81 and MSP58C82 are derivatives of the MSP58C80 mixed-signal processor. These two derivatives differ only in their timing specifications (see Section B.4, *Clock and Timing Characteristics*). The differences between the MSP58C81/C82 and the MSP58C80 are listed in the following bulleted list:

- The MSP58C81/C82 I/O interface has more terminals than the MSP58C80. These provide the extra I/O terminals necessary to implement a software-driven external memory interface, without losing general-purpose I/O capability.
- The MSP58C81/C82 does not have a hardware-driven external memory interface.
- The MSP58C81/C82 serial port can supply hardware-generated CLKX and FSX signals.

Just as with the MSP58C80, a MSP58C20 can be used with the MSP58C81/C82. A simplified block diagram of the MSP58C81/C82 used with a MSP58C20 is given in Figure G–1.

Figure G–1. MSP58C81/C82 and MSP58C20 Simplified Functional Block Diagram



G.2 Features

The following is a list of features of the MSP58C81/C82.

TMS320C25 Core CPU

- ❑ MSP58C81 has a 16.384 MIPS TMS320C25 core (61-ns instruction cycle)
- ❑ MSP58C82 has a 20.48 MIPS TMS320C25 core (49-ns instruction cycle)
- ❑ 128K-word total data/program memory space
- ❑ 32-bit ALU/accumulator
- ❑ 16-bit parallel shifter between data bus and ALU
- ❑ Eight auxiliary registers with independent arithmetic unit
- ❑ 16-bit x 16-bit parallel multiplier with 32-bit product
- ❑ Eight-level hardware stack
- ❑ 133-instruction TMS320C25 instruction set (see the TMS320C2x User's Guide)
  - ■ Single-cycle multiply/accumulate instructions
  - ■ Repeat instructions for efficient use of program space and enhanced execution
  - ■ Block moves for data/program management
  - ■ Instructions that support adaptive filtering, fast Fourier transform (FFT), and extended-precision arithmetic functions
  - ■ Bit-reversed addressing mode for radix-2 FFT
- ❑ On-chip timer for control operations
- ❑ Serial port for direct codec interface

Expanded Internal Memory

- ❑ 20K-word on-chip program ROM
- ❑ 1312-word on-chip RAM

Expanded I/O Capability

- ❑ 24 general-purpose input/output terminals
- ❑ 16 general-purpose input terminals
- ❑ 16 general-purpose output terminals

Enhanced Clock Control

- ❑ The MSP58C81 is used with a 4.096 MHz crystal. Its phase-locked loop (PLL) generates a clock that is software controllable between 2.048 MHz and 65.536 MHz
- ❑ The MSP58C82 uses a crystal between 4.096 MHz and 5.120 MHz. When using a 5.120 MHz crystal, the MSP58C82 PLL generates a clock that is software controllable between 2.560 MHz and 81.920 MHz.
- ❑ The MSP58C81/C82 has a variable-speed processor clock for power consumption control.
- ❑ The MSP58C81/C82 uses a low-frequency external crystal, ceramic resonator, or clock reference source.
- ❑ The MSP58C81/C82 internal "real-time" counter is tied to reference source

Analog-to-Digital and Digital-to-Analog Converters

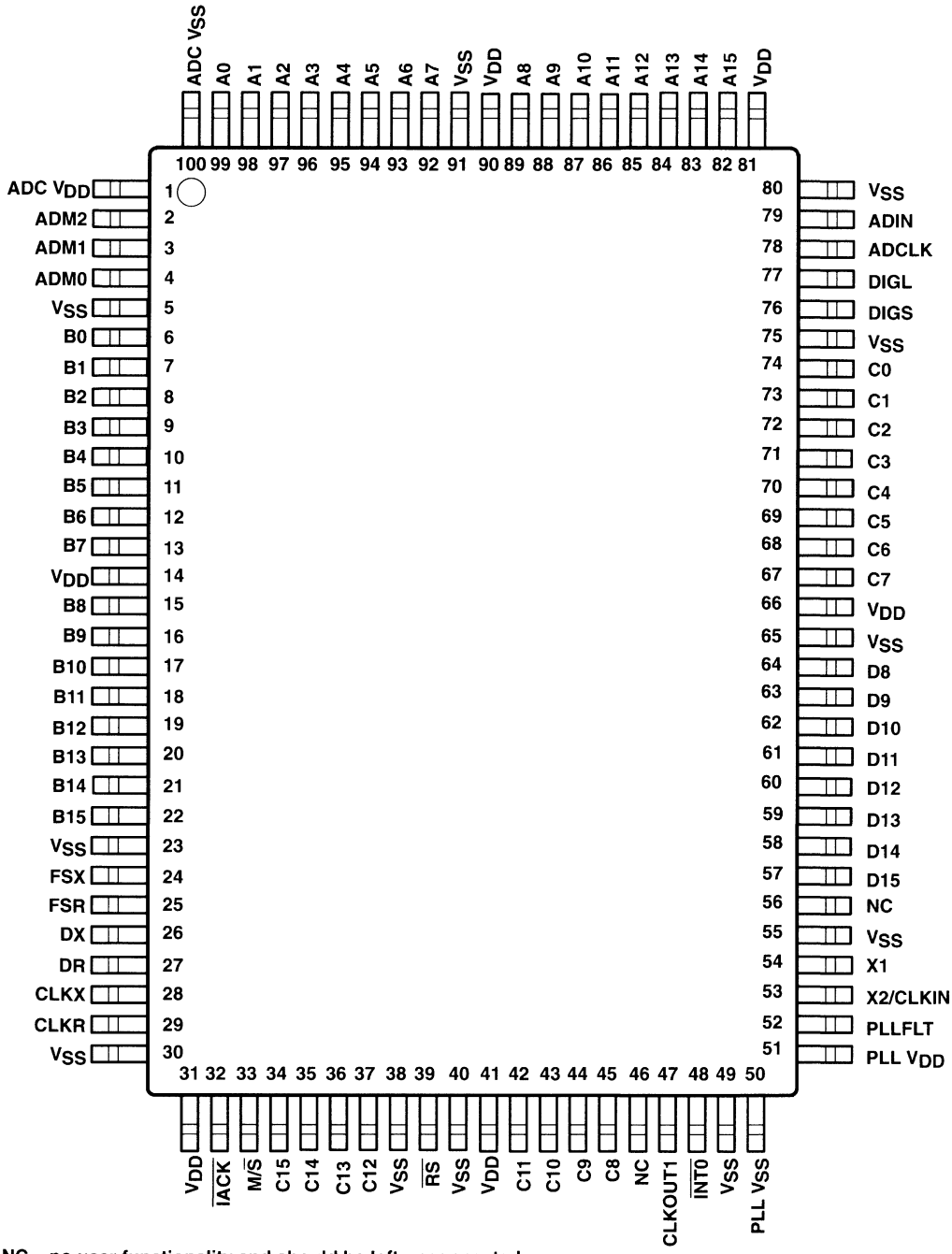
- ❑ 16-bit resolution voice-band analog-to-digital converter (ADC)
- ❑ 16-bit resolution voice-band digital-to-analog converter (DAC)
- ❑ 8-bit resolution successive-approximation low-frequency analog-to-digital converter with on-chip 3-to-1 analog multiplexer

G.3 Pinout and Signal Descriptions

Figure G–2 shows the signal assignments for the MSP58C81/C82, Figure G–3 illustrates the recommended clock and PLL circuits, Figure G–4 shows the recommended reset circuit, and Table G–1 provides signal descriptions.

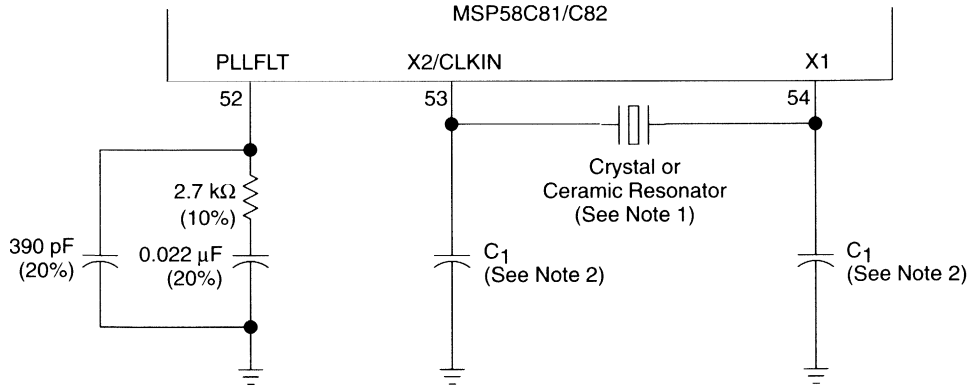
Because the MSP58C81/C82 supports a larger number of I/O terminals than the MSP58C80, several MSP58C80 terminal functions are not supported on the MSP58C81/C82, they include $\overline{PS}$, $\overline{DS}$, $\overline{IS}$, $\overline{AS}$, CLKOUT2, DD0, $\overline{RAS}$, $\overline{CAS}$, $\overline{ALATCH}$, $\overline{STRB}$, R/$\overline{W}$, MP/$\overline{MC}$, and D0 – D7.

Figure G–2. MSP58C81/C82 Terminal Assignments



NC – no user functionality and should be left unconnected.

Figure G–3. External Clock and PLL Filter



Notes 1) The MSP58C81 uses a 4.096 MHz crystal. The MSP58C82 uses a crystal between 4.096 MHz and 5.120 MHz.
2) Capacitance is specified by crystal or ceramic resonator manufacturer.

Figure G–4. Reset Circuit

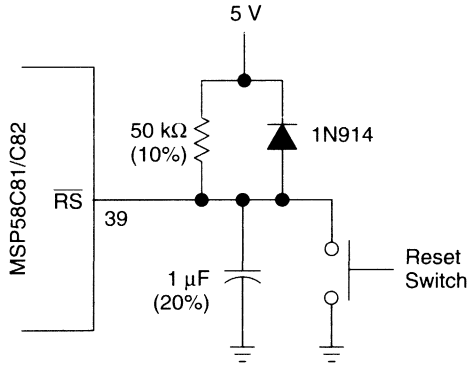


Table G–1. MSP58C81/C82 Signal Descriptions

Signal Name	Terminal No.	I/O/Z	Description
			Memory and I/O Lines
A0 – A7 A8 – A15	99 – 92, 89 – 82	O	A port. A port is a parallel output bus comprising A15 (MSB) through A0 (LSB). The A port provides 16 general purpose output terminals. This bus is I/O mapped. Performing an OUT instruction to I/O address 0h modifies the values driven on this bus. The values driven to the A port upon reset are undetermined.
C0 – C7 C8 – C11 C12 – C15	74 – 67, 45 – 42, 37 – 34	I	C port. C port is a parallel input bus comprising C15 (MSB) through C0 (LSB). The C port provides 16 general purpose input terminals. This bus is I/O mapped. Performing an IN instruction from I/O address 0h reads the values on this bus.
			Serial Interface Signals
M/$\overline{S}$	33	I	Serial port master/slave select input. When M/$\overline{S}$ is high, the serial port is placed in master mode, causing CLKX and FSX to be outputs. When M/$\overline{S}$ is low, the serial port is placed in slave mode, causing CLKX and FSX to be inputs.
CLKX	28	I/O	Transmit clock input/output. CLKX is the clock signal for clocking data from the XSR (serial port transmit shift register) to the DX terminal. When operating the serial port in master mode (M/$\overline{S}$ is high), CLKX is an output terminal and has a frequency given by the reference oscillator rate divided by 2 (CLKX is 2.048 MHz for a 4.096 MHz reference oscillator). When operating the serial port in slave mode (M/$\overline{S}$ is low), CLKX is an input terminal and must be provided by an external source to shift data out the DX terminal.
FSX	24	I/O	Frame synchronization input/output pulse for transmit. The falling edge of the FSX pulse initiates the data transmit process, beginning the clocking of the XSR (serial port transmit shift register). When operating the serial port in master mode (M/$\overline{S}$ is high), FSX is an output terminal and has a frequency given by the CLKX rate divided by 256 (FSX is 8 kHz for a 4.096 MHz reference oscillator). When operating the serial port in slave mode (M/$\overline{S}$ is low), FSX is an input terminal. Whether the serial port is operating in master mode or slave mode, the TXM bit of status register ST1 must always be cleared. If it is not cleared, the emulation part and the production part will have different behavior.
			Miscellaneous Signals
V_{SS}	38	I	This terminal must be tied to the other V_{SS} terminals.
NC	46, 56		These terminals must be left unconnected.

G.4 Memory Organization

The MSP58C81/C82 provides three separate address spaces that are mapped internally: program space, data space, and I/O space.

The MSP58C81/C82 has 1312 words of on-chip RAM that are divided into three blocks (B0, B1, and B2). Of these three blocks, block B0 (512 words) is configurable as either data or program memory, and blocks B1 and B2 (total of 800 words) are always configured as data memory. The MSP58C81/C82 also has 20K-words of on-chip ROM, which is configured as program memory.

Program/data RAM block B0 (512 words) resides in pages 4 – 7 of the data space when configured as data memory and in the highest 512 words of the program space when configured as program memory. Data RAM block B1 (768 words) resides in pages 8 – 13 of the data space, and data RAM block B2 resides in the upper 32 words of data page 0. The remainder of page 0 of the data space is composed of memory-mapped registers and internally reserved locations, and pages 1 – 3 and 14 – 15 consist of internally reserved locations. The internally reserved locations cannot be used for storage, and their contents are undefined when read.

Block B0 is mapped into either the data or program space, based on the value of the CNF bit of status register ST1. The CNFD command clears the CNF bit, causing block B0 to be configured as data memory. The CNFP command sets the CNF bit, causing block B0 to be configured as program memory. A reset configures block B0 as data memory.

Within the 64K-word program memory space: a) addresses 0000h – 001Fh consist of interrupt vectors and internally reserved locations, b) addresses 0020h – 4FAFh are mapped to program ROM, c) addresses 4FB0h – 4FFFh are reserved for factory test code, d) addresses 5000h – FFFFh are reserved locations and their contents are undefined when read (with the exception that RAM block B0 can still be mapped as internal program memory at addresses FE00h – FFFFh with the CNFP command).

The MSP58C81/C82 is capable of addressing 16 locations in the I/O space. These locations consist of one I/O-mapped register and internally reserved locations.

Figure G–5. System Memory Maps After a CNFD Instruction (CNF Bit is Cleared)

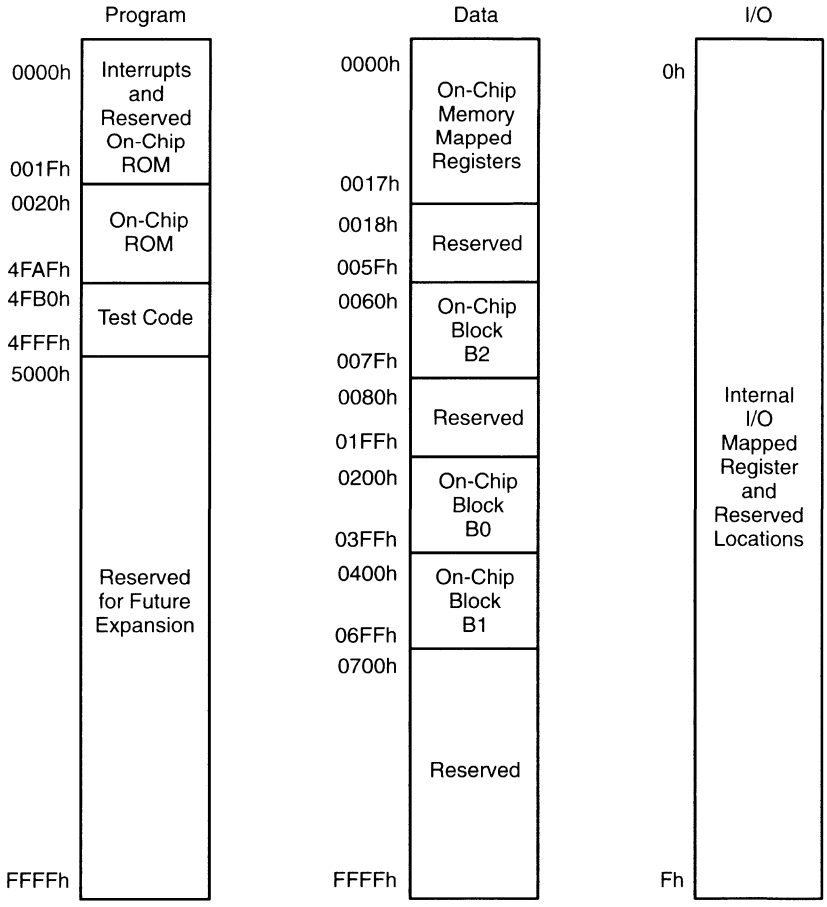
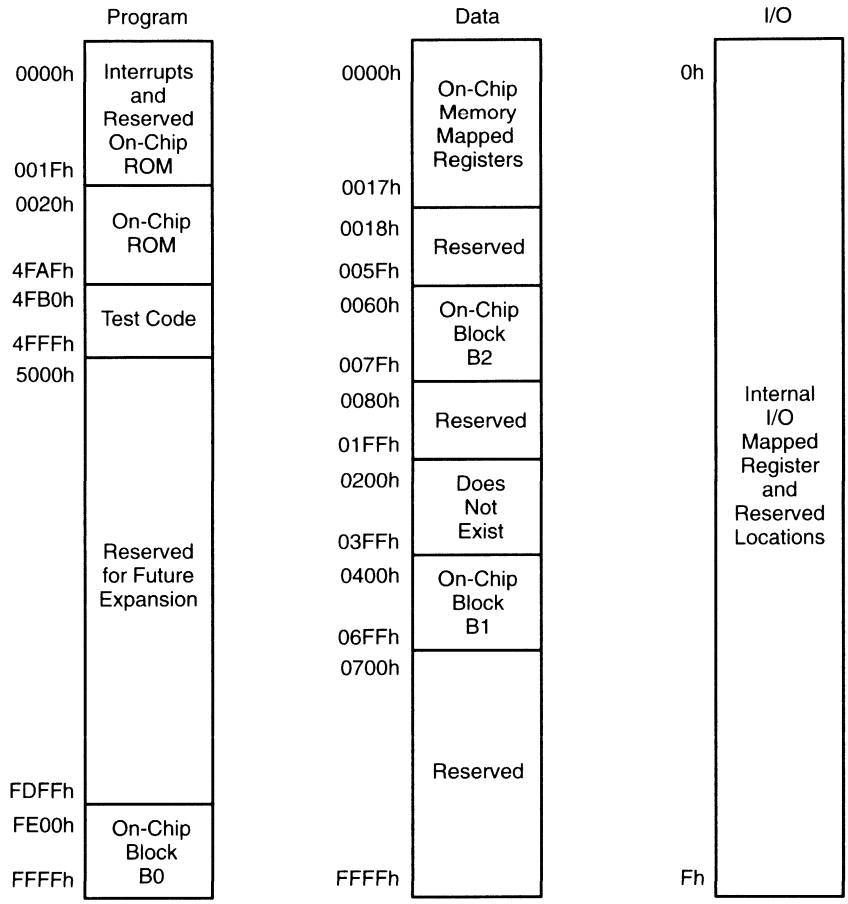


Figure G–6. System Memory Maps After a CNFP Instruction (CNF Bit is Set)



G.4.1 Memory Mapped Registers

The lower 24 words of the MSP58C81/C82 data space consist of 18 memory-mapped registers and 6 reserved locations (see Table 2–1). The reserved locations cannot be used for storage, and their contents are undefined when read. The following sections provide descriptions of memory-mapped registers that behave differently on the MSP58C81/C82 than on the MSP58C80. For any memory-mapped register not found in the following sections, please refer to Section 2.1.3, *Memory-Mapped Registers*.

G.4.1.1 MEMTYPE — External-Memory Interface Register

The external-memory interface register (MEMTYPE) is a 16-bit read/write register that specifies the following settings for each of the external-memory spaces:

- Standard (SRAM/ROM) or dynamic (DRAM) memory
- Wide or narrow data format
- Number of wait states

MEMTYPE also contains a global external-memory-refresh enable bit (see Table G–2).

Table G–2. External-Memory Interface Register (MEMTYPE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RF	D/S	N/W	NWS			D/S	N/W	NWS			D/S	N/W	NWS		
Global	Alternate Data Space					Data Space					Program and I/O Spaces				

RF: Global refresh enable, 1 = enable, 0 = disable
D/S: Dynamic = 1, Standard = 0
N/W: Narrow = 1, Wide = 0
NWS: Number of wait states, ranging from 0 (000) through 7 (111)

The MSP58C81/C82 does not provide external refresh signals, nor does it provide external terminals for accessing the program space, data space, I/O space, or alternate data space.

I/O address 0h accesses the A port and C port. Performing an IN instruction from I/O address 0h causes the values on the C port terminals to be copied to the RAM location specified by the IN instruction. Performing an OUT instruction from I/O address 0h copies the value from the RAM location specified by the OUT instruction to the A port latches.

Note: I/O Space Configuration for Latching

In order for the latching to work properly, the I/O space must be configured as standard, narrow, and with two wait states (i.e., the lower 5 bits of MEMTYPE must be 01010). All remaining bits in MEMTYPE should be cleared.

MEMTYPE defaults to 0000h on reset. Before accessing A port or C port, 000Ah must be written to MEMTYPE.

G.4.1.2 DI — D-Port Input Register

The D-port input register (DI) is a 16-bit read-only register that contains data being read from the D-port lines. The MSP58C81/C82 only has eight D-port lines (D8 – D15). The value of these lines can be read on the eight most significant bits of DI. The contents of the eight least significant bits of DI are undefined when read.

G.4.1.3 DO — D-Port Output Register

The D-port output register (DO) is a 16-bit read/write register. This register contains data being written to the D-port lines that are configured as outputs. The MSP58C81/C82 only has eight D-port lines (D8 – D15), and the value driven on these lines is controlled by writing to the eight most significant bits of DO.

Note: D-Port Default Setting

All D-port lines default to input on reset.

G.4.1.4 DDIR — D-Port Direction Register

The D-port direction register (DDIR) is a 16-bit read/write register that allows each individual line of the D port to be specified as either an input (the selected DDIR bit is cleared) or an output (the selected DDIR bit is set). DDIR defaults to 0000h upon reset. The upper eight bits of DDIR program the eight D-port terminals of the MSP58C81/C82 as either inputs or outputs.

Note: DDIR Bit Settings

The lower eight bits of DDIR must always be cleared.

G.4.2 I/O-Mapped Registers

The MSP58C81/C82 can address 16 locations in the I/O space using the IN and OUT instructions. The I/O registers are numbered in Table G–3 from 0h through Fh, with the number of each register indicating its I/O address.

Table G–3. I/O-Mapped Registers

Address	Abbreviation	Name
0h	ACPORT	A-port and C-port control register
1h – Fh		Reserved

ACPORT — A-Port and C-Port Control Register

The A-port and C-port control register (ACPORT) is a 16-bit register. When performing an OUT instruction to any of the I/O addresses between 0h and 7h, the ACPORT writes to the A-port terminals. However, since only address 0h is tested, only OUT instructions to I/O address 0h should be used. When performing an IN instruction from any of the I/O addresses between 0h and 7h, ACPORT reads from the C-port terminals. However, since only address 0h is tested, only IN instructions from I/O address 0h should be used or by using an MSP58P81.

G.5 Development Tools

Emulation of the MSP58C81 can be performed by inserting an EAB58C81 (emulation adaptor board) between the SE pod of the EVM58C80 and the customer target board. Prototyping of the MSP58C81 can be performed by inserting an EAB58C81 between a MSP58P80 and the customer target board or by using an MSP58P81.

Appendix H

Glossary

A

A15–A0: The parallel memory address bus A15 (MSB) through A0 (LSB). It addresses external I/O, data, program, and alternate data spaces.

AAH: Alternate Address High (mapped to I/O location Ah). The alternate address high register (AAH) is an 8-bit register.

AAL: Alternate Address Low (mapped to I/O location 9h). The alternate address low register (AAL) is a 16 bit register. Since writing to the AAL does not initiate a prefetch-read, this register is generally used when preparing for a postwrite operation.

AALP: Alternate Address Low for Prefetch-Read (mapped to I/O location Dh). The alternate address low register for prefetch-read (AALP) is a 16-bit register.

ACC: Accumulator (MSP58C80 hardware). This is an internal register of the MSP58C80.

ADAC: Sigma-Delta ADC/DAC Control Register (mapped to data location 0010h). The sigma-delta ADC/DAC control register is a 6-bit register that allows the sigma-delta ADC and DAC blocks on the MSP58C80 to be placed in a low-power state. It indicates the status of ADC and DAC processing, allows the ADC FIR filter to be bypassed, and sets the DAC sample rate to be the nominal rate or twice the nominal rate.

ADB: Alternate Data Buffer for Read without Prefetch (mapped to I/O location 8h). The alternate data buffer for read without prefetch (ADB) is a 16-bit register.

ADBP: Alternate Data Buffer for Prefetch-Read/Postwrite (mapped to I/O location Ch). The alternate data buffer for prefetch-read/postwrite (ADBP) is a 16-bit register.

ADC: Analog-to-Digital Converter. The MSP58C80 has two different ADCs. The first is a sigma-delta ADC that requires the MSP58C20 to function. The second is a successive-approximation low-frequency ADC.

ADCLK: Sigma-Delta ADC/DAC Modulator Clock Input (MSP58C20 terminal). ADLCK is a clock input signal for connection to the MSP58C80 ADCLK signal output.

ADCLK: Sigma-Delta ADC/DAC Modulator Clock Output (MSP58C80 terminal). ADCLK is a clock output signal for connection to the MSP58C20 ADCLK signal input.

ADDK: Add to Accumulator Short Immediate (TMS320C25 instruction). ADDK adds an 8-bit immediate value to the accumulator with the result replacing the accumulator contents.

ADIN: Sigma-Delta ADC Data Input (MSP58C80 terminal). ADIN is a digital input signal for connection to MSP58C20 ADOUT signal.

ADM0: Successive-Approximation low-frequency ADC MUX Input 0 (MSP58C80 terminal). ADM0 is one of three analog input signals, any one of which may be connected to the successive-approximation ADC by way of an on-chip analog multiplexer.

ADM1: Successive-Approximation ADC MUX Input 1 (MSP58C80 terminal). ADM1 is one of three analog input signals, any one of which may be connected to the successive-approximation ADC via an on-chip analog multiplexer.

ADM2: Successive-Approximation ADC MUX Input 2 (MSP58C80 terminal). ADM2 is one of three analog input signals, any one of which may be connected to the successive-approximation ADC via an on-chip analog multiplexer.

ADOUT: Sigma-Delta ADC Data Output (MSP58C20 terminal). ADOUT is a digital output signal for connection to MSP58C80 ADIN signal.

AIB58C80: The AIB58C80 is part of the EVM58C80 development system and acts as the user's target board. It provides a 4.096-MHz oscillator circuit, a PLL filter, and supplies power to approximate terminals.

AIM: Sigma-Delta Analog Input Minus (MSP58C20 terminal). AIM is an analog interface input terminal for the MSP58C20.

AIP: Sigma-Delta Analog Input Positive (MSP58C20 terminal). AIP is an analog interface input terminal for the MSP58C20.

$\overline{\text{ALATCH}}$: Standard Memory Address Demultiplex Strobe (MSP58C80 terminal). $\overline{\text{ALATCH}}$ serves as a demultiplexing latch signal for 24-bit multiplexed addresses and as a time multiplexing signal for accessing lower/upper byte from byte-wide memories.

ALU: Arithmetic Logic Unit (MSP58C80 hardware). The ALU is an internal hardware block of the MSP58C80 used for 32-bit arithmetic operations.

AOM: Sigma-Delta Analog Output Minus (MSP58C20 terminal). AOM is an analog interface output terminal for the MSP58C20.

AOP: Sigma-Delta Analog Output Positive (MSP58C20 terminal). AOP is an analog interface output terminal for the MSP58C20.

ARx: Auxiliary Registers 0 – 7 (MSP58C80 register). These registers can be used for indirect addressing of data memory or temporary data storage.

ARAM: Audio-Grade Dynamic Random Access Memory. ARAMs are DRAM chips that meet a reduced specification.

ARAU: Auxiliary Register Arithmetic Unit (MSP58C80 hardware). The ARAU can simultaneously perform one operation while the ALU is busy performing another operation.

ARB: Auxiliary Register Pointer Buffer. The ARB comprises three bits of status register ST1. When the ARP is loaded, the old value of the ARP is loaded into the ARB except during an LST or LST1 instruction.

ARP: Auxiliary Register Pointer. The ARP comprises three bits of status register ST0. This field selects the AR to be used in indirect addressing.

$\overline{\text{AS}}$: Alternate Space Select Signal (MSP58C80 terminal). $\overline{\text{AS}}$ is a signal for selecting alternate data space memory.

B

B15–B0: The B Port. B port is a parallel data bus B15 (MSB) through B0 (LSB) and provides 16 general I/O signals.

BDIR: B Port Direction Register (mapped to data location 000Eh). The B port direction register is a 16-bit, read/write register that allows individual lines of the B port to be specified as either input (logic 0) or output (logic 1) lines.

BI: B Port Input Register (mapped to data location 000Ch). The B-port input register (BI) is a 16-bit read-only register that contains data being read from the B-port lines.

$\overline{\text{BIO}}$: Branch Control Input (TMS320C25 terminal). $\overline{\text{BIO}}$ controls the branching of the BIOZ instruction. When $\overline{\text{BIO}}$ is low, the branch is executed.

BIOZ: Branch on I/O Status = 0 (TMS320C25 instruction). When the $\overline{\text{BIO}}$ signal is low then BIOZ causes a branch to be executed.

BLKD: Block Move from Data Memory to Data Memory (TMS320C25 instruction). BLKD copies consecutive memory words from a source data memory block to a destination data memory block.

BO: B-Port Output Register (mapped to data location 000Dh). The B-port output register (BO) is a 16-bit read/write register that contains data being written to B-port lines that are configured as outputs (all B-port lines default to being inputs on reset).

Bp: Band-pass cutoff frequency. Bp is the band-pass cutoff frequency for the oversampling digital filter of the MSP58C80.

bps: Bits per second

Bs: Band-stop cutoff frequency. Bs is the band-stop cutoff frequency for the oversampling digital filter of the MSP58C80.

C

C: Carry Bit. C is one bit of the status register ST1. C is set if an addition instruction produces a carry and cleared if a subtraction instruction produces a borrow.

$\overline{\text{CAS}}$: DRAM Column Address Strobe (MSP58C80 terminal). $\overline{\text{CAS}}$ is asserted low after the column address has been set up for each of the columns accessed during a read or write operation.

CINT Interrupt: Real-Time Counter Interrupt (MSP58C80 maskable interrupt). A CINT interrupt is generated when the RTC register decrements to zero.

CINT Mask Bit: Real-Time Counter Interrupt Mask Bit. CINT is one bit of the IMR register. When a zero is set in CINT, the real-time counter interrupt (CINT) is masked.

CLKOUT1: Instruction Cycle Clock Output (MSP58C80 terminal). CLKOUT1 is a clock output with a period equal to the instruction cycle period.

CLKOUT2: Second Instruction Cycle Clock Output (MSP58C80 terminal). CLKOUT2 is a second clock output with a period equal to the instruction cycle period.

CLKR: Receive Clock Input (MSP58C80 terminal). CLKR is an external clock signal for clocking data from the DR signal into the RSR (serial port receive shift register).

CLKX: Transmit Clock Input (MSP58C80 terminal). CLKX is an external clock signal for clocking data from the XSR (serial port transmit shift register) to the DX signal.

CNF: On-Chip RAM Configuration Control Bit. The CNF is one bit of the status register ST1. This bit causes memory block B0 to be mapped to either data memory or program space memory.

CNFD: Configure Block as Data Memory (TMS320C25 instruction). CNFD clears the CNF bit of status register ST1 so that memory block B0 is mapped to data memory.

CNFP: Configure Block as Program Memory (TMS320C25 instruction). CNFP sets the CNF bit of status register ST1 so that memory block B0 is mapped to program space memory.

CPT: Call Progress Tone.

D

D15–D0: The D Port. D port is a parallel data bus D15 (MSB) through D0 (LSB) that transfers data between MSP58C80 and external I/O, data, program, and alternate data spaces.

DAC: Digital-to-Analog Converter. The MSP58C80 contains one internal sigma-delta DAC that requires the MSP58C20 to function.

DD0: DRAM Data Terminal for ×1 DRAM Interface (MSP58C80 terminal). DD0 transfers data between MSP58C80 and 1-bit wide DRAM.

DDIR: D-Port Direction Register (mapped to data location 000Bh). DDIR is a 16-bit read/write register that allows individual lines of the D port to be specified as either input (logic 0) or output (logic 1) lines.

DI: D-Port Input Register (mapped to data location 0009h). DI is a 16-bit read-only register that contains data being read from the D-port lines.

DIGL Input Terminal: Sigma-Delta DAC Level Input (MSP58C20 terminal). DIGL is a digital input signal for connection to the MSP58C80 DIGL output signal.

DIGL Output Terminal: Sigma-Delta DAC Level Output (MSP58C80 terminal). DIGL is a digital input signal for connection to the MSP58C20 DIGL input signal.

DIGS Input Terminal: Sigma-Delta DAC Sign Input (MSP58C20 terminal). DIGS is a digital input signal for connection to the MSP58C80 DIGS output signal.

DIGS Output Terminal: Sigma-Delta DAC Sign Output (MSP58C80 terminal). DIGS is a digital output signal for connection to the MSP58C20 DIGS input signal.

DMOV: Data Move in Data Memory (TMS320C25 instruction). DMOV copies the contents of a specified data memory address to the next higher address.

DO: D-Port Output Register (mapped to data location 000Ah). DO is a 16-bit read/write register that contains data being written to D-port lines that are configured as outputs (all D-port lines default to being inputs on reset).

DP: Data Memory Page Pointer. DP comprises nine bits of status register ST0. The 9-bit DP pointer is concatenated with the seven LSBs of an instruction word to form a direct-memory address of 16-bits.

DR: Serial Data Receive Input (MSP58C80 terminal). Serial data is received into the RSR (serial port receive shift register) using the DR terminal.

DRAM: Dynamic Random Access Memory

DRR: Serial Port Data Receive Register (mapped to data location 0000h). DRR is a 16-bit read/write register that holds the data received by the serial port, and it can be operated in 8-bit byte or 16-bit word mode.

D/S: Dynamic Memory/Standard Memory Selection. The D/S occupies three bits in the MEMTYPE register. There is one bit each for selecting dynamic or standard memory for alternate data space, data space, and program and I/O space.

$\overline{\text{DS}}$: Data Space Select Signal (MSP58C80 terminal). $\overline{\text{DS}}$ is used for selecting data space memory.

DSP: Digital Signal Processor

DSPA: TMS320C25 Assembler (software). DSPA is software used to assemble TMS320C25 source code.

DSPHEX: TMS320C25 Object Format Converter (software). DSPHEX is the software used to create a TI-tagged object file that can be uploaded to the EVM58C80.

DSPLNK: TMS320C25 Linker (software). DSPLNK is software used to link TMS320C25 object code.

DSPSYM: EVM58C80 Symbol Table Generator (software). DSPSYM is software used to create a symbol table file that can be uploaded to the EVM58C80.

DTAD: Digital Telephone Answering Device. A DTAD is a possible application for the MSP58C80.

DTMF: Dual Tone Multifrequency. A method of coding signals used in telephone applications in which two nonharmonically related frequencies are added together to represent the information. DTMF tones are also referred to as touch tones.

DX: Serial Data Transmit Output (MSP58C80 terminal). Serial data is transmitted from the XSR (serial port transmit shift register) using the DX terminal.

DXR: Serial Port Data Transmit Register (mapped to data location 0001h). DXR is a 16-bit read/write register that holds the data to be transmitted by the serial port and can be operated in 8-bit byte or 16-bit word mode or can be used as a general-purpose register.

E

EINT: Enable Interrupt (TMS320C25 instruction). The EINT instruction enables unmasked interrupts.

EVM58C80: MSP58C80 development tool. The EVM58C80 is a desktop development tool that is capable of emulating the MSP58C80 at full speed (65.536 MHz). It can be connected by an RS-232 cable to any host computer running ANSI terminal emulation software.

F

FFT: Fast Fourier Transform

FO: Format Bit. FO is one bit of status register ST1. FO controls whether the serial port registers are configured to receive and transmit 8-bit bytes or 16-bit words.

FREQ: Frequency Control Register (mapped to data location 0007h). FREQ is a 13-bit read/write register that allows four MSP58C80 clock rates to be adjusted. These clocks are the PLL clock, the refresh clock, the sigma-delta clock, and the processor clock. The PLL clock and the refresh clock are defined relative to the reference oscillator, which is either provided externally or controlled by an external crystal or ceramic resonator.

fs: Sampling Frequency. The sampling frequency for the oversampling digital filters of the MSP58C80 is represented by the symbol fs.

FSM: Frame Synchronization Mode Bit. This is one bit of status register ST1.

FSR: Frame Synchronization Input Pulse for Receive (MSP58C80 terminal). The failing edge of the FSR pulse initiates the data receive process, beginning the clocking of the RSR (serial port receive shift register).

FSX: Frame Synchronization Input/Output Pulse for Transmit (MSP58C80 terminal). The falling edge of the FSX pulse initiates the data transmit process, beginning the clocking of the XSR (serial port transmit shift register).

H

HM: Hold Mode Bit. HM is one bit of status register ST1 and is unused by the MSP58C80.

$\overline{\text{HOLD}}$: Hold Input (TMS320C25 terminal). $\overline{\text{HOLD}}$, when high, places the data, address, and control lines of the TMS320C25 in a high-impedance state (hold mode).

$\overline{\text{HOLDA}}$: Hold Acknowledge Signal (TMS320C25 terminal). $\overline{\text{HOLDA}}$ indicates that the TMS320C25 has gone into hold mode.

I

$\overline{\text{IACK}}$: Interrupt Acknowledge (MSP58C80 terminal). $\overline{\text{IACK}}$ indicates receipt of an interrupt and that the program is branching to the interrupt vector.

IFR: Interrupt Flag Register (MSP58C80 register). IFR is a 6-bit register used to indicate whether any of the interrupts ($\overline{\text{INT0}}$, SDINT, CINT, TINT, RINT, and XINT) are enabled or disabled.

IMR: Interrupt Mask Register (mapped to data location 0004h). IMR is a 6-bit read/write register that allows $\overline{\text{INT0}}$, SDINT, CINT, TINT, RINT, and XINT interrupts to be enabled and disabled.

IN: Input Data from Port (TMS320C25 instruction). IN reads a 16-bit value from one of the I/O-mapped registers or one of the external I/O ports into the specified data memory location.

$\overline{\text{INT0}}$: External User Interrupt (MSP58C80 maskable interrupt). The $\overline{\text{INT0}}$ interrupt input is maskable by the interrupt mask register and the interrupt mode bit.

$\overline{\text{INT1}}$: External User Interrupt #1 (TMS320C25 maskable interrupt).

$\overline{\text{INT2}}$: External User Interrupt #2 (TMS320C25 maskable interrupt).

INTM: Interrupt Mode Bit (part of status register ST0). INTM is one bit of status register ST0. It enables or disables all maskable interrupts.

Ips: Instructions Per Second

$\overline{\text{IS}}$: I/O space select signal (MSP58C80 terminal). $\overline{\text{IS}}$ is used for selecting I/O space memory.

J

JEDEC: Joint Electronic Device Engineering Council

L

LACK: Load Accumulator Immediate Short (TMS320C25 instruction). LACK loads an 8-bit instruction into the accumulator right-justified.

LALK: Load Accumulator Long Immediate with Shift (TMS320C25 instruction). LALK loads a left-shifted 16-bit value into the accumulator and either the value is sign-extended or the high-order bits of the accumulator (past the shift) are set to zero. This depends on the value of SXM.

LARK: Load Auxiliary Register Immediate Short (TMS320C25 instruction). LARK loads an 8-bit constant into auxiliary register AR0.

LARP: Load Auxiliary Register Pointer (TMS320C25 instruction). LARP loads the contents of ARP (a 3-bit field in status register ST0) into ARB (a 3-bit field in status register ST1) and then loads a constant into ARP.

LDPK: Load Data Memory Page Pointer Immediate (TMS320C25 instruction). LDPK loads a constant into the DP bits of status register ST0.

linker: A linker is a software tool that combines object files to form an object module.

LPC: Linear Predictive Coding. LPC uses a mathematical model of a human vocal tract to enable efficient digital storage and recreation of realistic speech.

LSB: Least Significant Bit

LST1: Load Status Register ST1 (TMS320C25 instruction). LST1 loads a data memory value into status register ST1. ARB is then copied to ARP of status register ST0.

M

MCELP: Modified Code-Excited Linear Prediction

MEMTYPE: External Memory Interface Register (mapped to data location 0008h). The MEMTYPE register configures each type of external space (program, I/O, data, and alternate data) with the type of memory in use (standard narrow, standard wide, dynamic narrow, or dynamic wide).

MP/$\overline{\text{MC}}$: Microprocessor/Microcomputer Mode Select (MSP58C80 terminal). In microcomputer mode (low), MP/$\overline{\text{MC}}$ causes the lower 32K words of program memory to be mapped internally. In microprocessor mode (high), the lower 32K words of program memory are mapped externally.

MSB: Most Significant Bit

MSE58C80: The MSE58C80 is used for development, testing, and demonstration purposes in place of a MSP58C80 and is sometimes referred to as the system emulation (SE) chip. It does not contain a mask-programmed ROM like the MSP58C80. To replace the mask-programmed ROM, the MSE58C80 has extra terminals that can be attached to an external source. This chip can be used with the EVM58C80 to download the program from a computer. The MSE58C80 can also be used with the SEB58C80; in this situation the program is burned into EPROMs. The MSE58C80 comes in a 144-terminal PGA package.

MSP58C20: The MSP58C20 contains the analog section of the sigma-delta ADC and the sigma-delta DAC. When connected to the MSP58C80 with a minimum of four interface lines, the devices act as a voice-band sigma-delta ADC and as a voice-band sigma-delta DAC.

$\overline{\text{MSC}}$: Microstate Complete Signal (TMS320C25 terminal).

MUX: Multiplexer. A device for selecting one of a number of inputs.

MX0: Multiplexer Selection Bit #0 (part of SAAD memory-mapped register). MX0 is one of two bits of the SAAD register used to select which analog input (ADM0, ADM1, or ADM2) is sent to the ADC.

MX1: Multiplexer Selection Bit #1 (part of SAAD memory-mapped register). MX1 is one of two bits of the SAAD register used to select which analog input (ADM0, ADM1, or ADM2) is sent to the ADC.

N

N/W: Narrow Memory/Wide Memory Selection. The N/W occupies three bits in the MEMTYPE register. There is one bit each for selecting wide or narrow memory for alternate data space, data space, and program and I/O space.

NWS: Number of Wait States. The NWS occupies three 3-bit sections in the MEMTYPE register. There is one section each for setting the number of wait states for alternate data space, data space, and program and I/O space.

O

OUT: Output Data to Port (TMS320C25 instruction). OUT writes a 16-bit value from a data memory location to the specified I/O-mapped register or external I/O port.

OV: Overflow Flag Bit. OV is one bit of status register ST0 and indicates when an overflow has occurred in the ALU.

OVM: Overflow Mode Bit. OVM is one bit of status register ST0 and results in either the overflow going to the accumulator or setting the accumulator with its most negative or most positive value.

P

PCPD: Processor Clock Predivider. PCPD is three bits of the FREQ memory-mapped register and is used to set the speed of the processor clock relative to the PLL clock.

PLL: Phase-Locked Loop. A circuit for synchronizing a variable local oscillator with the phase of an input signal.

PLLFG: Phase-Locked Loop Frequency Gain. PLLFG comprises five bits of the FREQ memory-mapped register and is used to set the frequency of the phase-locked loop clock.

PLLFLT: PLL filter Connection (MSP58C80 terminal). PLLFLT is used for connecting the external PLL filter.

PM: Product Shift Mode. PM comprises two bits of status register ST1. It determines how the results of the multiplier are loaded into the ALU. This is done directly with no shift, shifted left one place with the LSB zero filled, shifted left four places with the LSBs zero filled, or shifted right six places and sign-extended.

ppm: Parts Per Million

PRD: Period Register (mapped to data location 0003h). PRD is a 16-bit read/write register that is initialized by a reset to the maximum value of FFFFh.

$\overline{PS}$: Program space select signal

PWAD: Power-Down Sigma-Delta ADC Input (MSP58C20 terminal). PWAD places the sigma-delta ADC on the MSP58C20 into low-power mode. PWAD can be controlled by the MSP58C80 using a general I/O terminal. A logic high level on the PWAD signal places the ADC portion of the MSP58C20 into low-power mode.

PWDA: Power-Down Sigma-Delta DAC Input (MSP58C20 terminal). PWDA places the sigma-delta DAC on the MSP58C20 into low-power mode. PWDA can be controlled by the MSP58C80 using a general I/O terminal. A logic high level on the PWDA signal places the DAC portion of the MSP58C20 into low-power mode.

Q

QFP: Quad Flat-Pack. QFP is the alphanumeric designator for the MSP58C80 package.

R

R/$\overline{W}$: Read/Write Signal (MSP58C80 terminal). R/$\overline{W}$ controls the direction of data transfer when communicating with an external device.

$\overline{RAS}$: DRAM Row Address Strobe (MSP58C80 terminal). $\overline{RAS}$ is asserted low after the row address has been set up for a read or write operation.

RD: Refresh Divider. RD is one bit of the FREQ memory-mapped register and is used to set the frequency of the refresh clock.

READY: Data Ready Input (TMS320C25 terminal). READY indicates that an external device is prepared for the bus transaction to be completed.

RET: Return from Subroutine (TMS320C25 instruction). RET is used to return to the calling program from a subroutine or subprogram.

RF: Global Refresh Enable Bit. RF is one bit of the MEMTYPE memory-mapped register and is used to enable global external memory refresh.

RHM: Reset Hold Mode (TMS320C25 instruction). RHM clears the HM status bit. In this mode, internal execution continues during assertion of the $\overline{\text{HOLD}}$ signal.

RINT Interrupt: Serial Port Receive Interrupt (MSP58C80 maskable interrupt). RINT is generated when the serial port receives either a byte or a word (as specified by the format bit FO).

RINT Mask Bit: Serial Port Receive Interrupt Mask Bit. RINT is one bit of the IMR register. When a zero is set in RINT, the serial port receive interrupt (RINT) is masked.

ROM: Read-Only Memory

RP: Pass-band Ripple. Rp is the band-pass ripple for the oversampling digital filter of the MSP58C80.

RPT: Repeat Instruction as Specified by Data Memory Value (TMS320C25 instruction). RPT loads the eight LSBs of the addressed memory value into the repeat counter (RPTC). This causes the following instruction to be executed one time more than the number loaded into the RPTC (provided that it is a repeatable instruction).

RPTC: Repeat Counter (MSP58C80 register). The RPTC is an 8-bit counter, which when loaded with N, causes the next single instruction to be executed N+1 times.

RPTK: Repeat Instruction as Specified by Immediate Value (TMS320C25 instruction). RPTK loads an 8-bit immediate value into the RPTC. This causes the next instruction to be executed one time more than the number loaded into the RPTC (provided that it is a repeatable instruction).

$\overline{\text{RS}}$: Reset Input (MSP58C80 terminal). $\overline{\text{RS}}$ low causes the MSP58C80 to terminate program execution and forces the program counter to zero.

Rs: Stop-band Ripple. Rs is the stop-band ripple for the oversampling digital filter of the MSP58C80.

RSR: Serial Port Receive Shift Register (MSP58C80 register). RSR shifts in serial data using the DR terminal. The RSR contents are sent to the DRR after a serial transfer is completed. RSR is not software addressable.

RTC: Real-Time Counter Register (mapped to data location 0006h). RTC is a 16-bit read/write register that is initialized by a reset of 0000h.

S

SAAD: Successive-Approximation ADC Register (mapped to data location 0013h). SAAD is a 10-bit register that contains the result of the successive-approximation ADC conversion. It also contains control bits to select which analog input is used.

SACH: Store High Accumulator With Shift (TMS320C25 instruction). SACH copies the entire accumulator into a shifter where it left shifts the entire 32-bit number anywhere from 0 to 7 bits. It then copies the upper 16 bits of the shifted value to data memory.

SACL: Store Low Accumulator (TMS320C25 instruction). SACL shifts the low-order bits of the accumulator left 0 to 7 bits, as specified by the shift code, and stores the result in data memory. The low-order bits are zero filled and the high-order bits are lost.

SDAD: Sigma-Delta ADC Input Register (mapped to data location 0011h). SDAD is a 16-bit read-only register that contains the input value read by the sigma-delta ADC.

SDDA: Sigma-Delta DAC Output Register (mapped to data location 0012h). SDDA is a 16-bit read/write register that contains the output value to be written to the sigma-delta DAC.

SDINT Interrupt: Sigma-Delta Interrupt (MSP58C80 maskable interrupt). A SDINT interrupt is generated whenever an ADC conversion or a DAC conversion (or both) has been completed.

SDINT Mask Bit: Sigma-Delta Interrupt Mask Bit. SDINT is one bit of the IMR register. When a zero is set in SDINT, the serial port receive interrupt (SDINT) is masked.

SDPD: Sigma-Delta Predivider. SDPD comprises four bits of the FREQ memory-mapped register and is used to set the speed of the sigma-delta clock relative to the PLL clock.

SE Pod: Part of the EVM58C80 development system, the SE pod contains a MSE58C80 chip, four emulation connectors with connector terminals numbered 1–100, and two 40-terminal cables. The SE pod is connected to the user's target board by way of the four emulation connectors.

SHM: Set Hold Mode (TMS320C25 instruction). SHM sets the HM status bit. In this mode, internal execution is halted during assertion of the $\overline{\text{HOLD}}$ signal.

ST0: Status Register #0 (MSP58C80 register). ST0 is one of two status registers that contain the status of various conditions and modes. ST0 contains the auxiliary register pointer (ARP), the overflow flag bit (OV), the overflow mode bit (OVM), the interrupt mode bit (INTM), and the data memory page pointer (DP).

ST1: Status Register #1 (MSP58C80 register). ST1 is one of two status registers that contain the status of various conditions and modes. ST1 contains the auxiliary register pointer buffer (ARB), the on-chip RAM configuration control bit (CNF), the test/control flag bit (TC), the sign-extension control bit (SXM), the carry bit (C), the hold mode bit (HM), the frame synchronization mode bit (FSM), the XF terminal status bit (XF), the formal bit (FO), the transmit mode bit (TXM), and the product shift mode bits (PM).

$\overline{\text{STRB}}$: Standard Memory Strobe Signal (MSP58C80 terminal). $\overline{\text{STRB}}$ provides a timing signal for standard memory data transfer.

SXM: Sign-Extension Mode Bit. SXM is one bit of status register ST1. When set, it produces sign extension on data as it is passed into the accumulator through the scaling shifter. Otherwise, it suppresses sign extension.

$\overline{\text{SYNC}}$: Synchronization Input (TMS320C25 terminal). $\overline{\text{SYNC}}$ allows clock synchronization between two or more TMS320C25s.

T

TC: Test/Control Flag Bit. It is set when a bit test instruction returns a 1, a compare instruction returns a true condition, or an exclusive-OR of the two MSBs of the accumulator returns a true when tested with the NORM instruction.

TIM: Timer Register (mapped to data location 0002h). TIM is a 16-bit read/write register that is initialized by a reset to the maximum value of FFFFh.

TINT Interrupt: Internal Timer Interrupt (MSP58C80 maskable interrupt). A TINT interrupt is generated when TIM decrements to zero.

TINT Mask Bit: Internal Timer Interrupt Mask Bit. TINT is one bit of the IMR register. When a zero is set in TINT, the internal time interrupt (TINT) is masked.

TRAP: Software Interrupt (MSP58C80 software interrupt). TRAP is a software interrupt that transfers program control to program memory location 30 and pushes the program counter plus one onto the hardware stack.

TXM: Transmit Mode Bit. TXM is one bit of status register ST1. It configures the FSX terminal as either an input or an output.

V

VAD: Voice Activity Detection

V_{DD}: Positive Supply Voltage (MSP58C80 terminals)

V_{SS}: Supply Ground (MSP58C80 terminals)

X

X1: Crystal Connection (MSP58C80 terminal). X1 is an output signal from the internal reference oscillator to the crystal or ceramic resonator.

X2/CLKIN: Crystal/Clock Connection (MSP58C80 terminal). X2/CLKIN is an input signal to the internal reference oscillator from the crystal or ceramic resonator.

XF Output Terminal: External Flag Output (TMS320C25 terminal). XF is used for signaling other processors in a multiprocessor environment or as a general output terminal.

XF Status Bit: External Flag Output Status Bit. XF is one bit of status register ST1. When terminal B15 of the MSP58C80 is configured as an output, the level placed on terminal B15 is the logical OR of the XF bit and the MSB of the BO register.

XINT Interrupt: Serial Port Transmit Interrupt (MSP58C80 maskable interrupt). An XINT interrupt is generated when all the bits of a serial transmit have been sent.

XINT Mask Bit: Serial Port Transmit Interrupt Mask Bit. XINT is one bit of the IMR register. When a zero is set in XINT, the serial port transmit interrupt (XINT) is masked.

XSR: Serial Port Transmit Shift Register (MSP58C80 register). The XSR register shifts in data from the DXR and sends it serially to the DX terminal.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

NOTES

NOTES

NOTES

TI Worldwide Sales and Representative Offices

AUSTRALIA / NEW ZEALAND: Texas Instruments Australia Ltd.: Melbourne [61] 3-9696-1211, Fax 3-9696-4446; **Sydney** 2-910-3100, Fax 2-805-1186.
BELGIUM: Texas Instruments Belgium S.A./N.V.: Brussels [32] (02) 726 75 80, Fax (02) 726 72 76.
BRAZIL: Texas Instrumentos Electronicos do Brasil Ltda.: Sao Paulo [55] 11-535-5133.
CANADA: Texas Instruments Canada Ltd.: Montreal (514) 421-2750; **Ottawa** (613) 726-3201; Fax 726-6363; **Toronto** (905) 884-9181; Fax 884-0062.
DENMARK: Texas Instruments A/S: Ballerup [45] (44) 68 74 00.
FRANCE/MIDDLE EAST/AFRICA: Texas Instruments France: Velizy-Villacoublay [33] (1) 30 70 10 01, Fax (1) 30 70 10 54.
GERMANY: Texas Instruments Deutschland GmbH.: Freising [49] (08161) 800, Fax (08161) 80 45 16; **Hannover** (0511) 90 49 60, Fax (0511) 64 90 331; **Ostfildern** (0711) 340 30, Fax (0711) 340 32 57.
HONG KONG: Texas Instruments Hong Kong Ltd.: Kowloon [852] 2956-7288, Fax 2956-2200.
HUNGARY: Texas Instruments Representation: Budapest [36] (1) 319 2748/2767/2768, Fax (1) 319 2814.
IRELAND: Texas Instruments Ireland Ltd.: Dublin [353] (01) 475 52 33, Fax (01) 478 14 63.
ITALY: Texas Instruments Italia S.p.A.: Agrate Brianza [39] (039) 684 21, Fax (039) 684 29 12; **Rome** (06) 657 26 51.
JAPAN: Texas Instruments Japan Ltd.: Kanazawa [81] 0762-23-5471, Fax 0762-23-1583; **Kita Kanto** 0485-22-2440, Fax 0485-23-5787; **Kyoto** 075-341-7713, Fax 075-341-7724; **Kyushu** 0977-73-1557, Fax 0977-73-1583; **Matsumoto** 0263-33-1060, Fax 0263-35-1025; **Nagoya** 052-232-5601, Fax 052-232-7888; **Osaka** 06-204-1881, Fax 06-204-1895; **Tachikawa** 0425-27-6760, Fax 0425-27-6426; **Tokyo** 03-3769-8700, Fax 03-3457-6777; **Yokohama** 045-338-1220, Fax 045-338-1255.
KOREA: Texas Instruments Korea Ltd.: Seoul [82] 2-551-2804, Fax 2-551-2828.
MAINLAND CHINA: Texas Instruments China Inc.: Beijing [86] 10-500-2255, Ext. 3750/3751/3752, Fax 10-500-2705.
MALAYSIA: Texas Instruments Malaysia Sdn Bhd: Kuala Lumpur [60] 3-208-6001, Fax 3-230-6605.
MEXICO: Texas Instruments de Mexico S.A. de C.V.: Colonia del Valle [52] 5-639-9740.
NORWAY: Texas Instruments Norge A/S: Oslo [47] (02) 264 75 70.
PHILIPPINES: Texas Instruments Asia Ltd.: Metro Manila [63] 2-636-0980, Fax 2-631-7702.
SINGAPORE (& INDIA, INDONESIA, THAILAND): Texas Instruments Singapore (PTE) Ltd.: Singapore [65] 390-7100, Fax 390-7062.
SPAIN/PORTUGAL: Texas Instruments España S.A.: Madrid [34] (1) 372 80 51, Fax (1) 307 68 64.
SUOMI/FINLAND: Texas Instruments/OY: Espoo [358] (0) 43 54 20 33, Fax (0) 46 73 23.
SWEDEN: Texas Instruments International Trade Corporation (Sverigefilialen): Kista [46] (08) 752 58 00, Fax (08) 751 97 15.
SWITZERLAND: Texas Instruments Switzerland AG: Dietikon [41] 886-2-3771450.
TAIWAN: Texas Instruments Taiwan Limited: Taipei [886] 2-378-6800, Fax 2-377-2718.
THE NETHERLANDS: Texas Instruments Holland, B.V. Amsterdam [31] (020) 546 98 00, Fax (020) 646 31 36.
UNITED KINGDOM: Texas Instruments Ltd.: Northampton [44] (01604) 66 30 00, Fax (01604) 66 30 01.

UNITED STATES: Texas Instruments Incorporated: ALABAMA: Huntsville (205) 430-0114; **ARIZONA: Phoenix** (602) 224-7800; **CALIFORNIA: Irvine** (714) 660-1200; **Los Angeles** (818) 704-8100; **San Diego** (619) 278-9600; **San Jose** (408) 894-9000; **COLORADO: Denver** (303) 488-9300; **CONNECTICUT: Wallingford** (203) 269-0074; **FLORIDA: Fort Lauderdale** (214) 644-5580 **Orlando** (407) 667-5308; **Tampa** (813) 573-0331; **GEORGIA: Atlanta** (770) 662-7967; **ILLINOIS: Chicago** (708) 517-4500; **INDIANA: Indianapolis** (317) 573-6400; **KANSAS: Kansas City** (913) 451-4511; **MARYLAND: Baltimore** (410) 312-7900; **MASSACHUSETTS: Boston** (617) 895-9100; **MICHIGAN: Detroit** (810) 305-5700; **MINNESOTA: Minneapolis** (612) 828-9300; **NEW JERSEY: Edison** (908) 906-0033; **NEW MEXICO: Albuquerque** (505) 345-2555; **NEW YORK: Long Island** (516) 454-6600; **Poughkeepsie** (914) 897-2900; **Rochester** (716) 385-6770; **NORTH CAROLINA: Charlotte** (704) 522-5487; **Raleigh** (919) 876-2725; **OHIO: Cleveland** (216) 328-2149; **Dayton** (513) 427-6200; **OREGON: Portland** (503) 643-6758; **PENNSYLVANIA: Philadelphia** (610) 825-9500; **PUERTO RICO: Hato Rey** (809) 753-8700; **TEXAS: Austin** (512) 250-6769; **Dallas** (214) 917-1264; **Houston** (713) 778-6592; **Midland** (915)561-6521; **WISCONSIN: Milwaukee** (414) 798-5021.

North American Authorized Distributors

COMMERCIAL	
Almac / Arrow	800-426-1410 / 800-452-9185 Oregon only
Anthem Electronics	800-826-8436
Arrow / Schweber	800-777-2776
Future Electronics (Canada)	800-388-8731
Hamilton Hallmark	800-332-8638
Marshall Industries	800-522-0084 or www.marshall.com
Wyle	800-414-4144
OBSOLETE PRODUCTS	
Rochester Electronics	508-462-9332
MILITARY	
Alliance Electronics Inc	800-608-9494
Future Electronics (Canada)	800-388-8731
Hamilton Hallmark	800-332-8638
Zeus, An Arrow Company	800-524-4735
TI DIE PROCESSORS	
Chip Supply	(407) 298-7100
Elmo Semiconductor	(818) 768-7400
Minco Technology Labs	(512) 834-2022
CATALOG	
Allied Electronics	800-433-5700
Arrow Advantage	800-777-2776
Newark Electonics	800-367-3573

For Distributors outside North America, contact your local Sales Office.

A040896

TEXAS INSTRUMENTS

TEXAS INSTRUMENTS

Printed in U.S.A.,
7/96 – 5K

SPSU005B