TEXAS INSTRUMENTS

TMS370 Microcontroller Family

User's Guide

1996

8-Bit Microcontroller Family

Texas Instruments

Printed in U.S.A., March 1996
1604927-9761 revision C

SPNU127

User's Guide

TMS370 Microcontroller Family

1996

TMS370 Microcontroller Family User's Guide

Literature Number: SPNU127A
Manufacturing Part Number: 1604927-9761 revision C
February 1997

Printed on Recycled Paper

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.

Preface

Read This First

About This Manual

This user's guide describes the TMS370 family of microcontroller devices. These devices have robust features that enhance performance and enable new application technologies. The specifications and descriptions included in this user's guide apply to the TMS370CxxxA devices; all differences for the TMS370Cxxx devices are described in Appendix A.

The objective of this user's guide is to provide the information you need to implement a microcontroller design using a TMS370 device. The user's guide contains the following chapters:

Chapter 1 **Introduction to the TMS370 Family Devices.** Discusses the key features and the major components of the TMS370 family devices. Also includes block diagrams for each device category.

Chapter 2 **TMS370 Family Pinouts and Pin Descriptions.** Provides pinouts and pin descriptions for the TMS370 family device categories.

Chapters 3–15 Describe the operation and programming of each major function in the TMS370 architecture.

Chapter 16 **Assembly Language Instruction Set.** Describes the TMS370 addressing modes and each of the 73 instructions, including samples and examples.

Chapter 17 **Development Support.** Describes the hardware and software development tools available for the TMS370 devices.

Chapter 18 **Electrical Specifications and Timings.** Gives timing diagrams and electrical specifications for each of the device categories.

Chapter 19 **Customer Information.** Describes mask-ROM prototyping, TMS370 physical characteristics, and parts ordering.

Appendix A **Differences Among the TMS370CxxxA, TMS370CxxxB, and TMS370Cxxx Devices (Contact Options).** Points out the differences between the TMS370CxxxA and TMS370CxxxB devices as described in this manual and the TMS370Cxxx devices.

Appendix B **Peripheral File Memory Map.** Gives reference tables for the TMS370 control bits and registers.

Appendix C **Block Diagrams.** Gives reference block diagrams of the major circuits.

Appendix D **ASCII Character Set.** Lists the ASCII character set that the TMS370 assembler recognizes.

Appendix E **Opcode/Instruction Cross-Reference.** Gives an opcode-to-instruction cross-reference of all 73 mnemonics and 274 opcodes of the TMS370 instruction set.

Appendix F **Instruction/Opcode Cross-Reference and Bus Activity Table.** Gives an instruction-to-opcode cross-reference of all 73 mnemonics and 274 opcodes of the TMS370 instruction set and provides a cycle-by-cycle bus activity table.

Appendix G **Device Pinouts.** Provides pinouts for the individual device categories.

Appendix H **PLCC-to-PGA Pinouts.** Shows the pinouts for the standard PLCC-to-PGA sockets that are commonly used in prototype and production applications. You can use these pinouts when you wirewrap your breadboard with a socket.

Appendix I **PACT.H.** Gives PACT.H macros used with PACT example programs.

Appendix J **Glossary.** Defines acronyms and key terms used in this book.

Style and Symbol Conventions

This document uses the following conventions.

Symbol or Term	Example	Description
(xxxxxx.n)	SPICTL.4	Bit location convention used in text and figures, where 'xxxxxx' is the name of the register (e.g., SPICTL) containing the bit and 'n' is the bit number (7 = MSB, 0 = LSB).
h	1ABCh	Designates a number in the hexadecimal number system.
P0n	P012	Hexadecimal Peripheral File (PF) address used in instructions accessing the PF. (i.e., P012 = P18)
Pn	P18	Decimal Peripheral File (PF) address used in instructions accessing the PF. (i.e., P18 = P012).
R0n	R010	Hexadecimal Register File (RF) address used in instructions accessing the RF. (i.e., R010 = R16)
Rn	R16	Decimal Register File (RF) address used in instructions accessing the RF. (i.e., R16 = R010)
set		When used in reference to bits, means to write a logic 1 to the bit.
clear		When used in reference to bits, means to write a logic 0 to the bit.
MSbyte		Most significant byte
MSB		Most significant bit
LSbyte		Least significant byte
LSB		Least significant bit
bps		bits per second

- Program listings, program examples, interactive displays, filenames, and symbol names are shown in a special typeface similar to a typewriter's.

  Here is a sample program listing:

```
LABEL  SUB  R19,B        ;(B) minus (R19) is
                         ;stored in B

       SUB  #076h,A      ;(A) minus 076h is stored
                         ;in A

       SUB  R4,R9        ;(R9) minus (R4) is stored
                         ;in R9
```

- In syntax descriptions, the instruction, command, or directive is in a **bold typeface** font, and parameters are in an *italic typeface*. Portions of a syntax that are in **bold** should be entered as shown; portions of a syntax that are in *italics* describe the type of information that should be entered. Here is an example of a directive syntax:

  MOV *s,d*

  MOV is the instruction. This instruction has two parameters, indicated by *s* and *d*.

- Braces ({ and }) indicate a list. The symbol **|** (read as *or*) separates items within the list. Here's an example of a command that has a list:

  TST {A|B}

  This provides two *choices*: **TST A** or **TST B**.

  Unless the list is enclosed in square brackets, you must choose one item from the list.

Information About Cautions and Warnings

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.

This is an example of a warning statement.

A warning statement describes a situation that could potentially cause harm to you.

The information in a caution or a warning is provided for your protection; please read each carefully.

Related Documentation From Texas Instruments

The following books describe the TMS370 family devices and related support tools. To obtain a copy of any of these TI documents, call the Texas Instruments Literature Response Center at (800) 477–8924. When ordering, please identify the book by its title and literature number.

TMS370 Microcontroller Family Applications Book (literature number SPNA017) provides examples of hardware setups and accompanying software routines for various applications of the TMS370 microcontroller.

TMS370Cxxx 8-Bit Microcontrollers Data Sheets describe the features of the device and provide pinouts, electrical specifications, and timings for the following microprocessors:

- ***TMS370Cx0x*** (literature number SPNS029)
- ***TMS370Cx1x*** (literature number SPNS012)
- ***TMS370Cx2x*** (literature number SPNS018)
- ***TMS370Cx32*** (literature number SPNS015)
- ***TMS370Cx36*** (literature number SPNS039)
- ***TMS370Cx4x*** (literature number SPNS016)
- ***TMS370Cx5x*** (literature number SPNS010)
- ***TMS370Cx6x*** (literature number SPNS033)
- ***TMS370Cx7x*** (literature number SPNS034)
- ***TMS370Cx8x*** (literature number SPNS035)
- ***TMS370Cx9x*** (literature number SPNS036)
- ***TMS370CxAx*** (literature number SPNS037)
- ***TMS370CxBx*** (literature number SPNS038)
- ***TMS370CxCx*** (literature number SPNS040)

TMS370 8-Bit Microcontrollers Data Book (literature number SPND003) is a collection of the TMS370Cxxx microcontroller data sheets into a single book with additional chapters that discuss ordering information, an overview of development tools, a selection guide with a cross–reference of device functions, and quality and reliability.

TMS370 and TMS370C8 8-Bit Microcontroller Family Assembly Language Tools User's Guide (literature number SPNU010) describes the assembly language tools (assembler, linker, and other tools used to develop assembly code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS370/C8 8-bit family of devices.

TMS370 and TMS370C8 8-Bit Microcontroller Family Optimizing C Compiler User's Guide (literature number SPNU022) describes the TMS370/C8 8-bit C compiler. This C compiler accepts ANSI standard C source code and produces assembly language source code for the TMS370/C8 8-bit family of devices.

TMS370 Family C Source Debugger User's Guide (literature number SPNU028) tells you how to invoke the '370 XDS/22 emulator and application board versions of the C source debugger interface. This book discusses various aspects of the debugger interface, including window management, command entry, code execution, data management, and breakpoints, and includes a tutorial that introduces basic debugger functionality. It also includes an advanced tutorial that introduces the breakpoint, trace, and timing features.

Trademarks

If You Need Assistance. . .

If you want to. . .	**Do this. . .**
Request more information about Texas Instruments microcontroller 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**
Ask questions about product operation or report suspected problems	Call the Microcontroller Hotline: **(713) 274–2370** **FAX: (713) 274–4203**
Product information	Web address: **http://www.ti.com/sc/micro** Product Information Center: **(214) 644-5580**
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

Contents

Figures

Tables

Examples

Chapter 1

Introduction to the TMS370 Family Devices

The TMS370 family consists of VLSI, 8-bit, CMOS microcontrollers with on-chip EEPROM storage and peripheral support functions. These devices offer superior performance in complex, realtime control applications in demanding environments. They are available with mask-programmable ROM and OTP/EPROM. This chapter covers the following topics:

Topic **Page**

Texas Instruments has added new, more robust features to the TMS370 family of devices. These features are designed to enhance performance and enable new application technologies. The improved features include new watchdog modes, low-power modes, clock module, ROM security, standby RAM, Analog-to-Digital Converters, Serial Communication Interface 2, Timer 2B, and I/O ports. All family members are software compatible, so you can run many existing applications on the improved devices without having to modify your software. (Refer to Appendix A for more information about compatibility.)

There are 14 device families (see Table 1–2 on page 1-4 for a full listing). This chapter describes the key features and major components of each of the TMS370 microcontroller families, and includes block diagrams to illustrate controllers from each device family.

1.1 Typical Applications

In expanding its powerful TMS370 family of microcontrollers, TI offers many new configurable devices for specific applications. As microcontrollers have evolved, TI has added multiple peripheral functions to chips that originally had only a CPU, memory, and I/O blocks. Now, with the high-performance, software-compatible TMS370 microcontrollers, you can choose from over 78 standard products. Alternatively, you can use as many as 27 function modules to configure your new device quickly, easily, and cost effectively for your application.

The TMS370 family of devices is the ideal choice for the applications shown in Table 1–1.

Table 1–1. Typical Applications for TMS370 Family Microcontroller Devices

Application Area	Applications	
Automotive	Climate control systems Cruise control Entertainment systems Instrumentation	Navigational systems Engine control Antilock braking Body controllers
Computer	Keyboards Peripheral interface control Disk controllers	Terminals Tape Drives
Industrial	Motor control Temperature controllers Process control	Meter control Medical instrumentation Security systems
Telecommunications	Modems Intelligent phones Intelligent line card control	Telecopiers Debit cards

1.2 Available Development Support Products

The TMS370 family is fully supported by development tools that facilitate simplified software development for bringing new products to market more quickly. These tools include an assembler, a linker, an optimizing C compiler, a C source debugger, a design kit, a starter kit (with simulator), an in-circuit emulator, and an OTP/EPROM programmer. All of these tools work together using an IBM-compatible personal computer (PC) as the host and central control element. This allows you to select the host computer method of text management, and editing tools according to your system requirements.

The TMS370 in-circuit emulator (XDS—eXtended Development Support and the CDT370 Compact Development Tool) allows you to immediately begin designing, testing, and debugging your system. This is because the emulator is modular and configurable, eliminating the need to produce a complete new emulator for each TMS370 configuration.

More detailed information on development support products is provided in Chapter17, *Development Support*.

1.3 Device Families

TMS370 devices are divided into 14 families as shown Table 1–2. All the families are supported by a full complement of development tools.

Table 1–2. TMS370 Families and Their Corresponding Devices

Family	Devices Included		
TMS370Cx0x	TMS370C002A SE370C702†	TMS370C302A	TMS370C702
TMS370Cx1x	TMS370C010A TMS370C311A TMS370C712B	TMS370C012A TMS370C312A SE370C712A†	TMS370C310A TMS370C712A SE370C712B†
TMS370Cx2x	TMS370C020A TMS370C322A	TMS370C022A TMS370C722	TMS370C320A SE370C722†
TMS370Cx32	TMS370C032A SE370C732†	TMS370C332A	TMS370C732A
TMS370Cx36	TMS370C036A	TMS370C736A	SE370C736A†
TMS370Cx4x	TMS370C040A TMS370C042A	TMS370C340A TMS370C342A	TMS370C742A SE370C742A†
TMS370Cx5x	TMS370C050A TMS370C058A TMS370C156A TMS370C350A TMS370C356A TMS370C456A TMS370C758B SE370C758A†	TMS370C052A TMS370C059A TMS370C250A TMS370C352A TMS370C358A TMS370C756A TMS370C759A SE370C758B†	TMS370C056A TMS370C150A TMS370C256A TMS370C353A TMS370C452A TMS370C758A SE370C756A† SE370C759A†
TMS370Cx6x	TMS370C067A TMS370C768A SE370C769A†	TMS370C068A TMS370C769A	TMS370C069A SE370C768A†
TMS370Cx7x	TMS370C077A	TMS370C777A	SE370C777A†
TMS370Cx8x	TMS370C080‡ SE370C686A†	TMS370C380A	TMS370C686A
TMS370Cx9x	TMS370C090A	TMS370C792	SE370C792†
TMS370CxAx	TMS370C3A7A‡		
TMS370CxBx	TMS370C0B6A§		
TMS370CxCx	TMS370C3C0A	TMS370C6C2A	SE370C6C2A†

† These devices are system evaluators and are used only in a prototype environment. Their reliability has not been characterized.
‡ These devices use the recommended TMS370C758 (with converter socket) for prototyping.
§ This device uses the recommended TMS370C758 for prototyping. Refer to the TMS370CxBx Data Sheet (SPNS038) for pin-to-pin compatibility.

1.4 Key Features of the TMS370 Family

Each TMS370 device family has the following key features (not all features are available for all devices):

- **Compatibility** for supporting software migration between current and future microcontrollers
- **CMOS EPROM technology** for providing reprogrammable EPROM and OTP (one-time programmable) program memory for prototypes and for small-volume or quick-turn production
- **CMOS EEPROM technology** for providing EEPROM programming with a single 5-V supply
- **ADC technology** for converting analog signals to digital values
- **Static RAM/registers** for offering numerous memory options
- **Standby RAM** that offers data protection during a power-off condition
- **Flexible operating features:**
  - Power-reduction standby and halt modes
  - Operating temperature options:
    - 0°C to 70°C (L range)
    - –40°C to 85°C (A range)
    - –40°C to 105°C (T range)
  - Temperature range by device:
    - ROM devices: A, L, and T
    - EPROM and ROMless: T only
  - Input clock frequency options:
    - Divide by 4 (0.5 to 5 MHz SYSCLK) standard oscillator
    - Divide by 1 (2 to 5 MHz SYSCLK) phase-locked loop
  - Operating voltage range: 5 V ±10%
- **Flexible interrupt handling** for design flexibility:
  - Two programmable interrupt levels
  - Programmable rising- or falling-edge detect
- **System integrity features** that increase flexibility during the software development phase:
  - Oscillator fault detection
  - Privileged mode lockout
  - Watchdog timer
  - Memory security (ROM)

- **Memory-mapped ports** for easy addressing
- **An optimizing C compiler** that translates ANSI C programs into TMS370 assembly language source
- **A high-level language debugger** that lets you refine and correct code
- **A modular library** for quickly changing the device configuration.
- **18 addressing modes** that use eight formats, including the following:
  - Implied
  - Register-to-register arithmetic
  - Indirect addressing
  - Indexed and indirect branches and calls
  - PC relative
- **250-mA typical latch-up immunity at 25°C**
- **ESD** (electrostatic discharge) protection that exceeds 2000 V per MIL-STD-883C method 3015

1.5 Major Components of the TMS370 Architecture

In addition to the features listed in Section 1.4, the TMS370 families have the following architectural features. Table 1–4 on page 1-13 summarizes the features described below.

CPU

The TMS370 8-bit CPU has a status register, a program counter register, and a stack pointer. The CPU uses another feature, the register file, as working registers, accessed on the internal bus in one bus cycle. The 8-bit internal bus also allows access to memory and to the peripheral interfaces. The TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, and TMS370CxBx devices allow external memory expansion through ports A, B, C, and D.

Refer to Chapter 3, *CPU and Memory Organization*, for more complete information about the CPU.

Register File

The register file is located at the beginning of the TMS370 memory map. Register-access instructions in the TMS370 instruction set allow access to any of the first 256 registers (if available) in one bus cycle. The register file is used as general-purpose RAM and contains the stack.

Chapter 3, *CPU and Memory Organization*, provides additional information about the register file.

RAM

RAM modules other than those contained in the register file are mapped after the register file. The TMS370 accesses this RAM in two cycles.

Memory is described in full in Chapter 3, *CPU and Memory Organization.*

Data EEPROM

With the exception of the TMS370CxAx and TMS370CxCx families, all the devices in each product family use EEPROM.

The data EEPROM modules provide in-circuit programmability and data retention in power-off mode. The modules contain 256 or 512 bytes of EEPROM. This memory is useful for constants and infrequently changed variables required by the application program. The EEPROM can be

programmed and erased by using available EEPROM programmers or by the TMS370 itself under program control.

The data EEPROM modules are described in Chapter 6, *EPROM and EEPROM Modules.*

Program Memory

The program memory provides alternatives to meet the needs of your application. The program memory modules presently contain 2K, 4K, 8K, 16K, 24K, 32K, or 48K bytes of memory.

The program memory in TMS370C6xx, TMS370C7xx, SE370C6xx, and SE370C7xx devices is EPROM. EPROM, in a windowed ceramic package, can be programmed, erased, and reprogrammed for prototyping. EPROM devices in a non-windowed plastic package are one-time programmable (OTP) devices, used for small production runs. In TMS370C0xx, TMS370C3xx, and TMS370C4xx devices, the program memory is mask ROM that is programmed at the factory. ROM devices are appropriate for larger volume production.

Program memory is discussed in Chapter 6, *EPROM and EEPROM Modules.*

Input/Output Ports

The TMS370 family of devices have varying numbers of I/O ports, and of various port widths. Table 1–3 lists the widths, in the number of bits, for each of the ports of the different TMS370 families.

I/O ports are described in greater detail in Chapter 4, *System and Digital I/O Configuration.*

Table 1–3. Bits Per Port for TMS370 Devices

	Bits for Ports A – H					
Families†	**A**	**B**	**C**	**D**	**G**	**H**
TMS370Cx0x, TMS370Cx1x	8			5		
TMS370Cx2x	8	8	1	5		
TMS370Cx32	8			4		
TMS370Cx36	8			5		
TMS370Cx4x	8	3		5		
TMS370Cx5x, (64 pin)	8‡	8‡	8‡	6‡		
TMS370Cx5x, (68 pin)	8‡	8‡	8‡	8‡		
TMS370Cx6x	8‡	8‡	8‡	5‡		
TMS370Cx7x (64 pin)	8	8	8	6	6	
TMS370Cx7x (68 pin)	8	8	8	8	6	
TMS370Cx8x (40 pin)	8	8	6	5		
TMS370Cx8x (44 pin)	8	8	8	5		
TMS370Cx9x	8			5		
TMS370CxAx	8	8	1	5		
TMS370CxBx (64 pin)	8	8	8	6	8	1
TMS370CxBx (68 pin)	8	8	8	8	8	1
TMS370CxCx	8			4		

† For all families, the ports for these microcontrollers can be programmed, bit by bit, to function as either digital input or digital output.

‡ These ports can be configured by the software as the data bus, control bus, and address bus for external memory. Any bits not needed for external memory can be programmed to be either digital input or digital output.

Timer 1

Timer 1 is a 16-bit timer that can be configured in the following ways:

- ❑ A programmable 8-bit prescaler (providing a 24-bit realtime timer) that determines the independent clock sources for the general-purpose timer and the watchdog timer
- ❑ A 16-bit event timer to keep a cumulative total of the transitions
- ❑ A 16-bit pulse accumulator to measure the pulse input width
- ❑ A 16-bit input-capture function that latches the counter value on the occurrence of an external input

- Two 16-bit compare registers that trigger when the counter matches the contents of a compare register
- A self-contained PWM (pulse-width modulated) output control function

The operation of the timer can generate an interrupt to the CPU, set flag bits, reset the timer counter, toggle an I/O line, or generate PWM outputs. The timer can provide as much as 200 ns of resolution with a system clock (SYSCLK) speed of 5 MHz.

Timer 1 is described fully in Chapter 7.

Timer 2n

Timers 2A and 2B are 16-bit timers that can be configured in the following ways:

- Four independent clock sources for the general-purpose timer
- A 16-bit event timer to keep a cumulative total of the transitions
- A 16-bit pulse accumulator to measure the input pulse duration
- Two 16-bit input-capture devices which change a counter value to match the contents of a compare register
- Two 16-bit compare registers which trigger when a counter matches the contents of a compare register
- A self-contained PWM (pulse-width modulated) output controller

The operation of timers 2A and 2B can generate an interrupt to the CPU, set flag bits, reset the timer counter, toggle an I/O line, or generate PWM outputs. These timers provide as many as 200ns of resolution with a SYSCLK speed of 5 MHz.

Timers 2A and 2B are described fully in Chapter 8.

Watchdog Timer

The watchdog timer helps ensure system integrity. It can be programmed to generate a hardware reset when it times out. This function provides a hardware monitor over the software to avoid losing a program. If it is not needed as a watchdog, this timer can be used as a general-purpose timer.

For more information about the watchdog timer, refer to Section 7.7, page 7-21.

PACT (Programmable Acquisition and Control Timer)

The PACT module in the TMS370Cx3x family is a programmable timing module that uses some of the on-chip RAM to store its commands as well as the timer values. Only the TMS370Cx36 device offers 256-bytes of standby RAM, which protects data stored there against power failures. The PACT module offers the following:

- Input capture on as many as six pins, four of which can have a programmable prescaler
- One input capture pin that can drive an 8-bit event counter
- As many as eight timer-driven outputs
- A timer capability of as many as 20 bits
- Interaction between event counter and timer activity
- 18 independent interrupt vectors to allow the better servicing of events
- A watchdog timer with a selectable time-out period
- A mini-SCI (serial communications interface), which works as a full-duplex UART (universal asynchronous receiver transmitter)

Once set up, the PACT requires no CPU overhead except to service interrupts. The PACT module is described fully in Chapter 15, *Programmable Acquisition and Control Timer (PACT).*

SCI (Serial Communications Interfaces)

The SCI1 and SCI2 modules are built-in serial interfaces. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration. The SCI1 model is programmable to be isosynchronous up to 2.5 Mbps, and both SCIs share the following features:

- Programmable to be asynchronous (up to 156 kbps)
- Full-duplex, double-buffered receive (Rx) and transmit (Tx)
- Programmable format with error-checking capabilities

The SCI1 and SCI2 modules are described fully in Chapter 9, *Serial Communications Interface (SCI1) Module,* and Chapter 10*, Serial Communications Interface (SCI2) Module.*

SPI (Serial Peripheral Interface)

The SPI module is a built-in serial interface that facilitates communication between the network master, slave CPUs, and external peripheral devices. This module provides synchronous data transmission up to 2.5 Mbps. Like SCI1 and SCI2, the SPI is set up by software. After that, the CPU takes no part in timing, data format, or protocol. Also, like the SCI, the CPU reads and writes to memory-mapped registers to receive and transmit data. A SPI interrupt alerts the CPU when received data is ready.

The SPI module is described fully in Chapter 11, *Serial Peripheral Interface (SPI) Module.*

ADC (Analog-to-Digital Converter) Modules

The 8-bit ADC converter modules perform successive approximated conversion. The term ADC is a general term used for ADC1, ADC2, and ADC3 modules. The ADCs have the following input channels for each of the microcontroller families shown.

- ADC1 has the following:
  - Four channels in the 40-pin TMS370Cx4x family
  - Eight channels in the 44-pin TMS370Cx4x, TMS370Cx32, and TMS370Cx36 families
  - Eight channels in the 64- and 68-pin TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, and TMS370CxBx families
- ADC2 has four channels in the 28-pin TMS370CxCx family
- ADC3 has 15 channels in the 40- and 44-pin TMS370Cx9x families

The reference source and the input channels are selectable. You can program the conversion result to be the ratio of the input voltage to the reference voltage or the ratio of one analog input to another. Input lines that are not required for analog to digital conversion can be programmed as digital input lines.

The ADC converter modules are described in Chapters 12, 13, and 14.

1.6 Summary of Components by Device

The major components of the TMS370 device family (described in Section 1.5) are summarized by device in Table 1–4 beginning on page 1-13.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx0x	TMS370C002A	8K	—	256	256	—	SCI1	T1	—	22	÷4 or ÷1	28PLCC
	TMS370C302A	8K	—	—	256	—	SCI1	T1	—	22	÷4 or ÷1	28PLCC
	TMS370C702#	—	8K	256	256	—	SCI1	T1	—	22	÷4	28PLCC
	SE370C702Δ	—	8K	256	256	—	SCI1	T1	—	22	÷4	28CLCC
'Cx1x	TMS370C010A	4K	—	256	128	—	SPI	T1	—	22	÷4 or ÷1	28PDIP/PLCC
	TMS370C012A	8K	—	256	256	—	SPI	T1	—	22	÷4 or ÷1	28PDIP/PLCC
	TMS370C310A	4K	—	—	128	—	SPI	T1	—	22	÷4 or ÷1	28PDIP/PLCC
	TMS370C311A	2K	—	—	128	—	SPI	T1	—	22	÷4 or ÷1	28PDIP/PLCC
	TMS370C312A	8K	—	—	128	—	SPI	T1	—	22	÷4 or ÷1	28PDIP/PLCC
	TMS370C712A#	—	8K	256	256	—	SPI	T1	—	22	÷4	28PDIP/PLCC
	TMS370C712B#	—	8K	256	256	—	SPI	T1	—	22	÷1	28PDIP/PLCC
	SE370C712AΔ	—	8K	256	256	—	SPI	T1	—	22	÷4	28CDIP/CLCC
	SE370C712BΔ	—	8K	256	256	—	SPI	T1	—	22	÷1	28CDIP/CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx2x	TMS370C020A	4K	—	256	256	—	SCI1/SPI	T1	—	34	÷4 or ÷1	40PDIP/PSDIP/44PLCC
	TMS370C022A	8K	—	256	256	—	SCI1/SPI	T1	—	34	÷4 or ÷1	40PDIP/PSDIP/44PLCC
	TMS370C320A	4K	—	—	256	—	SCI1/SPI	T1	—	34	÷4 or ÷1	40PDIP/PSDIP/44PLCC
	TMS370C322A	8K	—	—	256	—	SCI1/SPI	T1	—	34	÷4 or ÷1	40PDIP/PSDIP/44PLCC
	TMS370C722#	—	8K	256	256	—	SCI1/SPI	T1	—	34	÷4	40PDIP/PSDIP/44PLCC
	SE370C722Δ	—	8K	256	256	—	SCI1/SPI	T1	—	34	÷1	40CDIP/CSDIP/44CLCC
'Cx32	TMS370C032A	8K	—	256	256	—	PACT–SCI	PACT	ADC1/8	23	÷4 or ÷1	44PLCC
	TMS370C332A	8K	—	—	256	—	PACT–SCI	PACT	ADC1/8	23	÷4 or ÷1	44PLCC
	TMS370C732A#	—	8K	256	256	—	PACT–SCI	PACT	ADC1/8	23	÷4	44PLCC
	SE370C732AΔ	—	8K	256	256	—	PACT–SCI	PACT	ADC1/8	23	÷4	44CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@ Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx36	TMS370C036A	16K	—	256	512◊	—	SPI/ PACT–SCI	PACT	ADC1/8	25	÷4 or ÷1	44PLCC
	TMS370C736A#	—	16K	256	512◊	—	SPI/ PACT–SCI	PACT	ADC1/8	25	÷4	44PLCC
	SE370C736Δ	—	16K	256	512◊	—	SPI/ PACT–SCI	PACT	ADC1/8	25	÷4	44CLCC
'Cx4x	TMS370C040A	4K	—	256	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4 or ÷1	40PDIP/PSDIP 44PLCC
	TMS370C042A	8K	—	256	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4 or ÷1	40PDIP/PSDIP 44PLCC
	TMS370C340A	4K	—	—	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4 or ÷1	40PDIP/PSDIP 44PLCC
	TMS370C342A	8K	—	—	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4 or ÷1	40PDIP/PSDIP 44PLCC
	TMS370C742A#	—	8K	256	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4	40PDIP/PSDIP 44PLCC
	SE370C742AΔ	—	8K	256	256	—	SCI1	T1/T2A	ADC1/4, 8§	32/36	÷4	40CDIP/CSDIP 44CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx5x	TMS370C050A	4K	—	256	256	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP 68PLCC
	TMS370C052A	8K	—	256	256	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP 68PLCC
	TMS370C056A	16K	—	512	512	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP 68PLCC
	TMS370C058A	32K	—	256	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP 68PLCC
	TMS370C059A☆	48K	—	256	3584	20K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C150A¶	—	—	—	256	56K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68PLCC
	TMS370C156A¶	—	—	—	512	56K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68PLCC
	TMS370C250A¶	—	—	256	256	56K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68PLCC
	TMS370C256A¶	—	—	512	512	56K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68PLCC
	TMS370C350A	4K	—	—	256	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC
	TMS370C352A	8K	—	—	256	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC
	TMS370C353A	12K	—	—	1536	112K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C356A	16K	—	—	512	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC
	TMS370C358A	32K	—	—	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx5x	TMS370C452A	8K□	—	256	256	112K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4 or ÷1	68PLCC
(Cont.)	TMS370C456A	16K□	—	512	512	112K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C756A#	—	16K	512	512	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4	64PSDIP/ 68PLCC
	TMS370C758A#	—	32K	256	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4	64PSDIP/ 68PLCC
	TMS370C758B#	—	32K	256	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷1	64PSDIP/ 68PLCC
	TMS370C759A☆#	—	48K	256	3584	20K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68PLCC
	SE370C756AΔ	—	16K	512	512	112K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4	64CSDIP/ 68CLCC
	SE370C758AΔ	—	32K	256	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷4	64CSDIP/ 68CLCC
	SE370C758BΔ	—	32K	256	1024	64K	SCI1/SPI	T1/T2A	ADC1/8	53/55	÷1	64CSDIP/ 68CLCC
	SE370C759A☆Δ	—	48K	256	3584	20K	SCI1/SPI	T1/T2A	ADC1/8	55	÷4	68CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'Cx6x	TMS370C067A	24K	—	256	1024	24K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C068A	32K	—	256	1024	24K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C069A☆	48K	—	256	3584	8K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4 or ÷1	68PLCC
	TMS370C768A#	—	32K	256	1024	24K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4	68PLCC
	TMS370C769A#☆	—	48K	256	3584	8K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4	68PLCC
	SE370C768AΔ	—	32K	256	1024	24K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4	68CLCC
	SE370C769AΔ☆	—	48K	256	3584	8K	SCI1/SPI	T1/T2A/ T2B	ADC1/8	55	÷4	68CLCC
'Cx7x	TMS370C077A	24K	—	256	512	—	—	T1/T2A	ADC1/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC
	TMS370C777A#	—	24K	256	512	—	—	T1/T2A	ADC1/8	53/55	÷4	64PSDIP/ 68PLCC
	SE370C777AΔ	—	24K	256	512	—	—	T1/T2A	ADC1/8	53/55	÷4	64CSDIP/ 68CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
’Cx8x	TMS370C080ϕ	4K	—	256	128	—	—	T1	—	33	÷4	40PLCC
	TMS370C380A	4K	—	—	128	—	—	T1	—	35	÷4 or ÷1	44PLCC
	TMS370C686A#	—	16K	—	256	—	—	T1	—	35	÷4	44PLCC
	SE370C686AΔ	—	16K	—	256	—	—	T1	—	35	÷4	44CLCC
’Cx9x	TMS370C090A	4K	—	256	128	—	—	T1	ADC3/15	25	÷4 or ÷1	40PSDIP/ 44PLCC
	TMS370C792#	—	8K	256	128	—	—	T1	ADC3/15	25	÷4	40PSDIP/ 44PLCC
	SE370C792Δ	—	8K	256	128	—	—	T1	ADC3/15	25	÷4	40CSDIP/ 44CLCC
’CxAx	TMS370C3A7Aϕ	24K	—	—	512	—	SCI1	T1/T2A	—	34	÷4 or ÷1	40PDIP
’CxBx	TMS370C0B6Aδ	16K	—	256	384	—	—	T1	ADC3/8	53/55	÷4 or ÷1	64PSDIP/ 68PLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

Table 1–4. TMS370 Family Architecture, Memory, and Module Summary (Continued)

Device Family	Device	Program Memory (bytes)		Data Memory (bytes)		Off-Chip Mem. Exp. (bytes)	Serial Interface Modules†	Timer Modules‡	ADC Channels	I/O Pins	Clock Generator	No. of Pins/ Package@
		ROM	EPROM	EEPROM	RAM							
'CxCx	TMS370C3C0A	4K	—	—	128	—	SCI2	T1	ADC2/4	22	÷4 or ÷1	28PLCC
	TMS370C6C2A#	—	8K	—	128	—	SCI2	T1	ADC2/4	22	÷4	28PLCC
	SE370C6C2AΔ	—	8K	—	128	—	SCI2	T1	ADC2/4	22	÷4	28CLCC

† The PACT module has a mini–SCI. SCI1 has a 3-pin configuration and SCI2 has a 2-pin configuration.
‡ The timer 1 module includes a watchdog timer that you can program to serve as a general-purpose 16-bit timer. The PACT module includes a watchdog timer.
§ 8 channels for the ADC1 module in the 44-pin package, and 4 channels for the ADC1 module in the 40-pin package
¶ In ROMless (microprocessor) mode, all address, data, and control lines are fixed as to their functions.
# For OTP (PLCC) availability information, contact your local TI sales office or distributor
Δ System evaluator (reprogrammable EPROM) for use in a prototype environment only.
☆ TMS370Cx59 and TMS370Cx69 can only operate up to 3 MHz SYSCLK.
□ TMS370C45x has ROM security, which inhibits the reading of data using any programmer.
◊ 256-byte standby RAM is general-purpose memory and is powered separately by the V_{CCSTBY} pin. The data stored in this memory is protected against power failure on the main V_{CC1} pins.
@Refer to Table 19–1 on page 19-7 for package type acronyms.
ϕ Uses the recommended TMS370C758 EPROM devices (requiring converter socket) for prototyping.
δ Uses the recommended TMS370C758 EPROM devices for prototyping. Refer to the TMS370CxBx Data Sheet (P/N SPNS038) for pin-to-pin compatibility, which requires software modification.

Note: Temperature ranges: ROM: A (–40 °C to 85 °C), L (0 °C to 70 °C), and T (–40 °C to 105 °C); EPROM and ROMless: T only.

1.7 Device Block Diagrams

This section contains functional block diagrams for each of the microcontroller device families. Each of these diagrams show the basic internal connections among the major architectural features summarized in Table 1–4. Refer to Chapter 2, *TMS370 Family Pinouts and Pin Descriptions*, for descriptions of the external connection names.

Figure 1–1. TMS370Cx0x Block Diagram

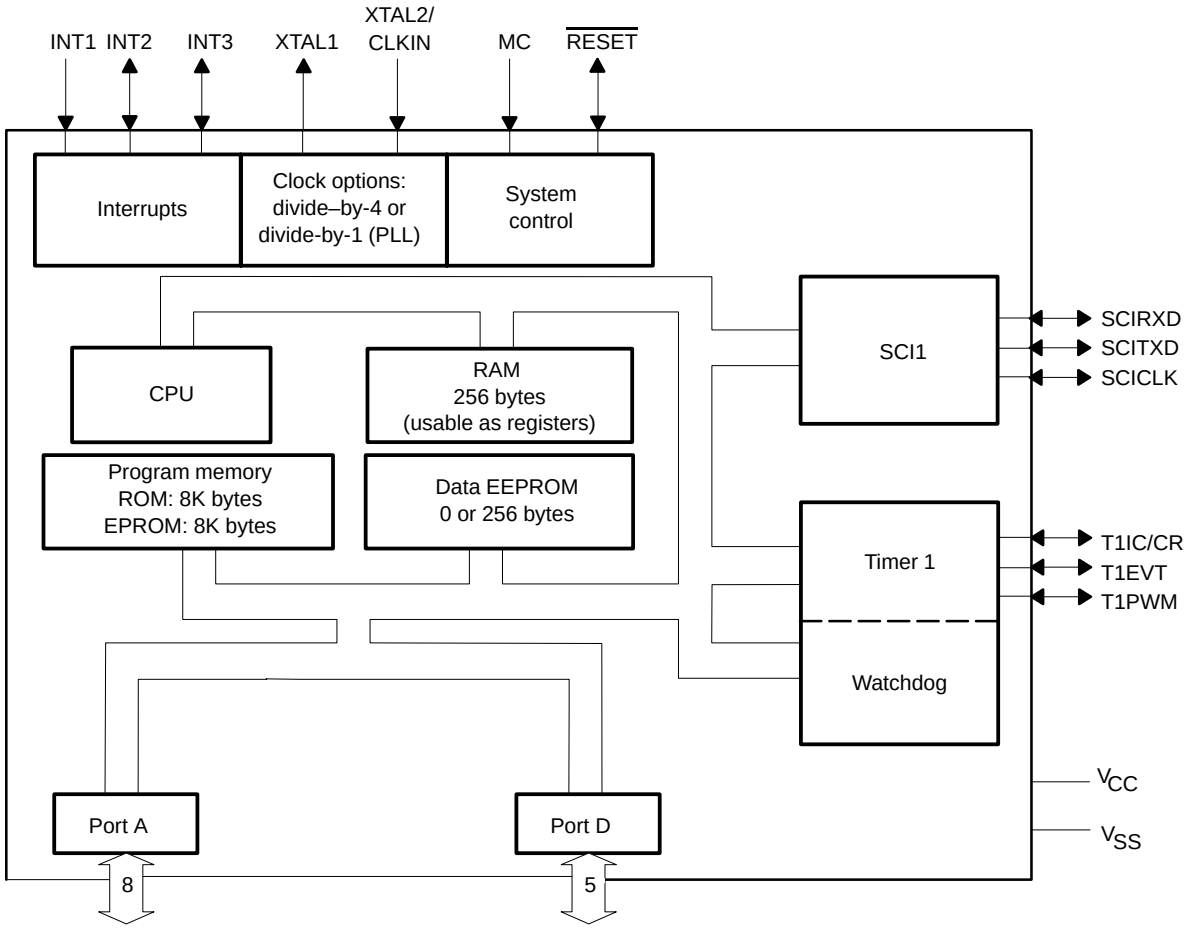


Figure 1–2. TMS370Cx1x Block Diagram

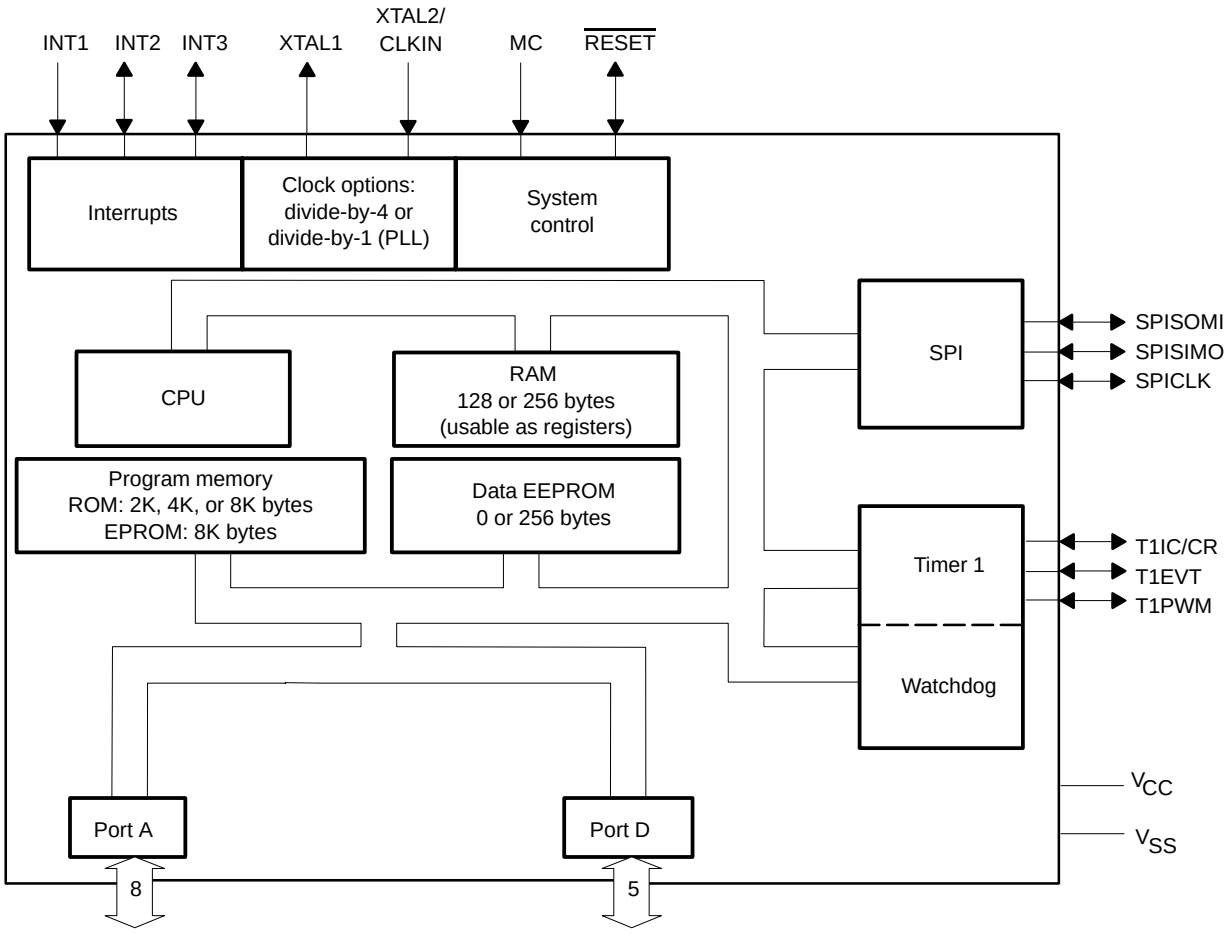


Figure 1–3. TMS370Cx2x Block Diagram

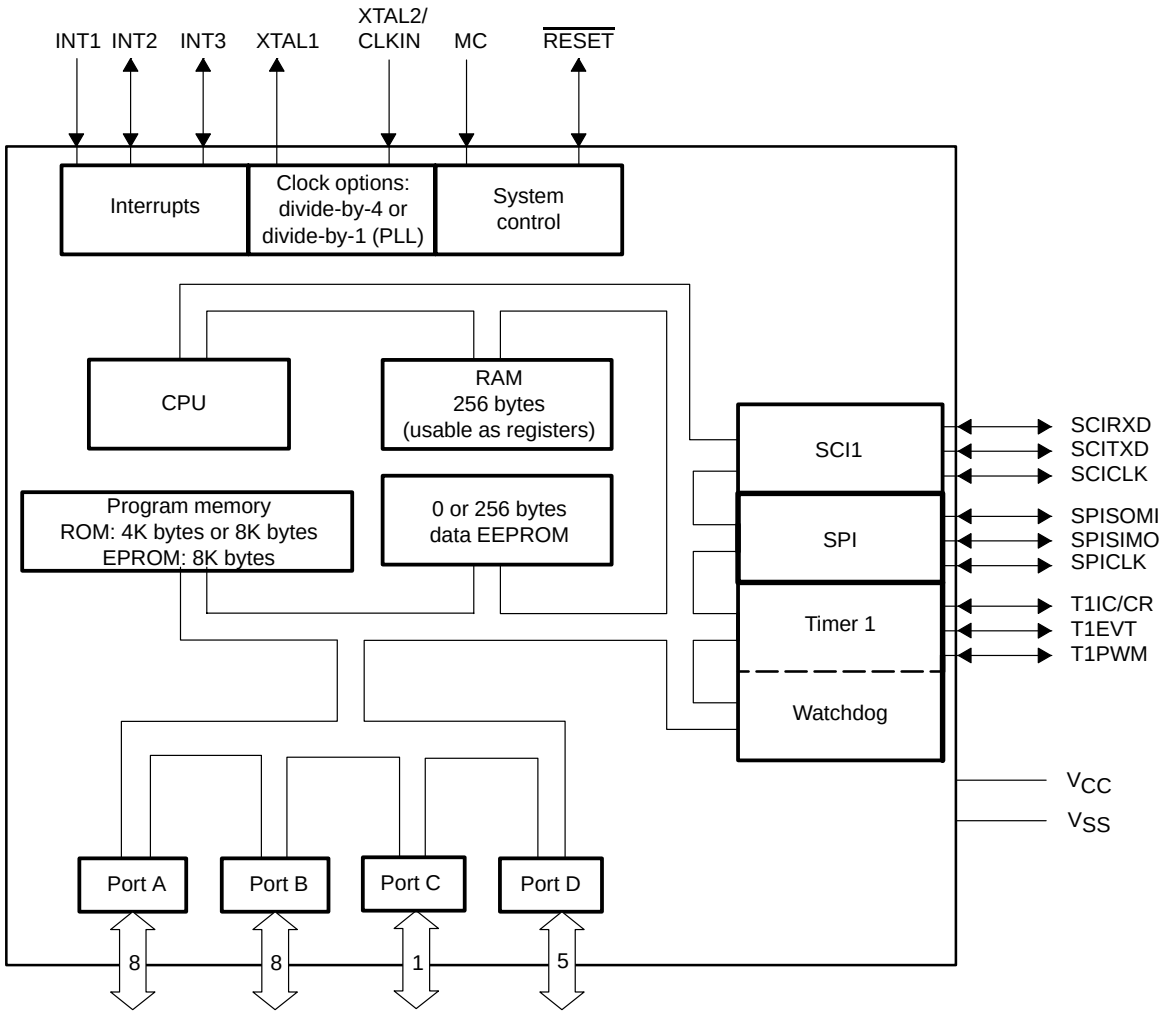
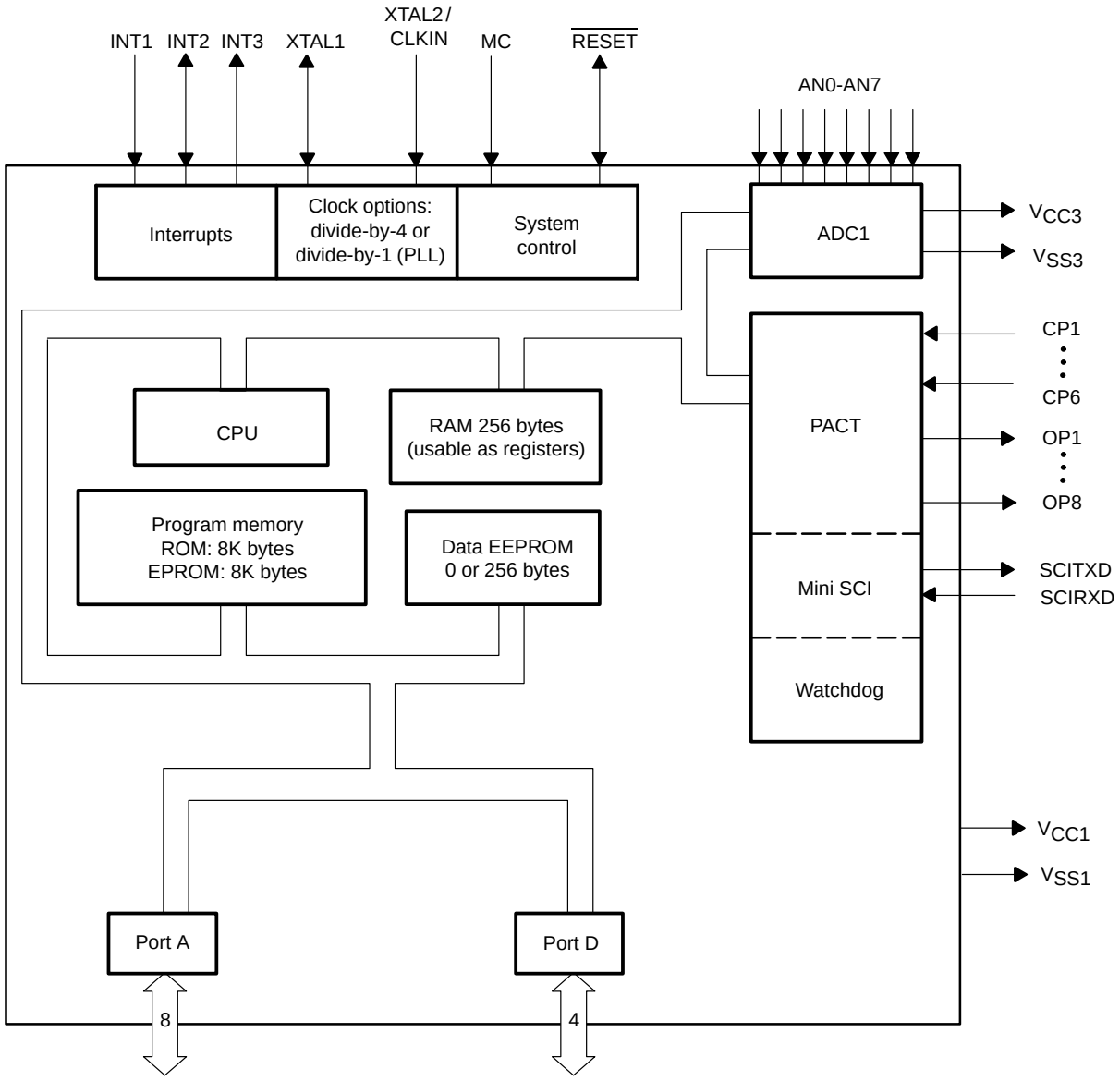
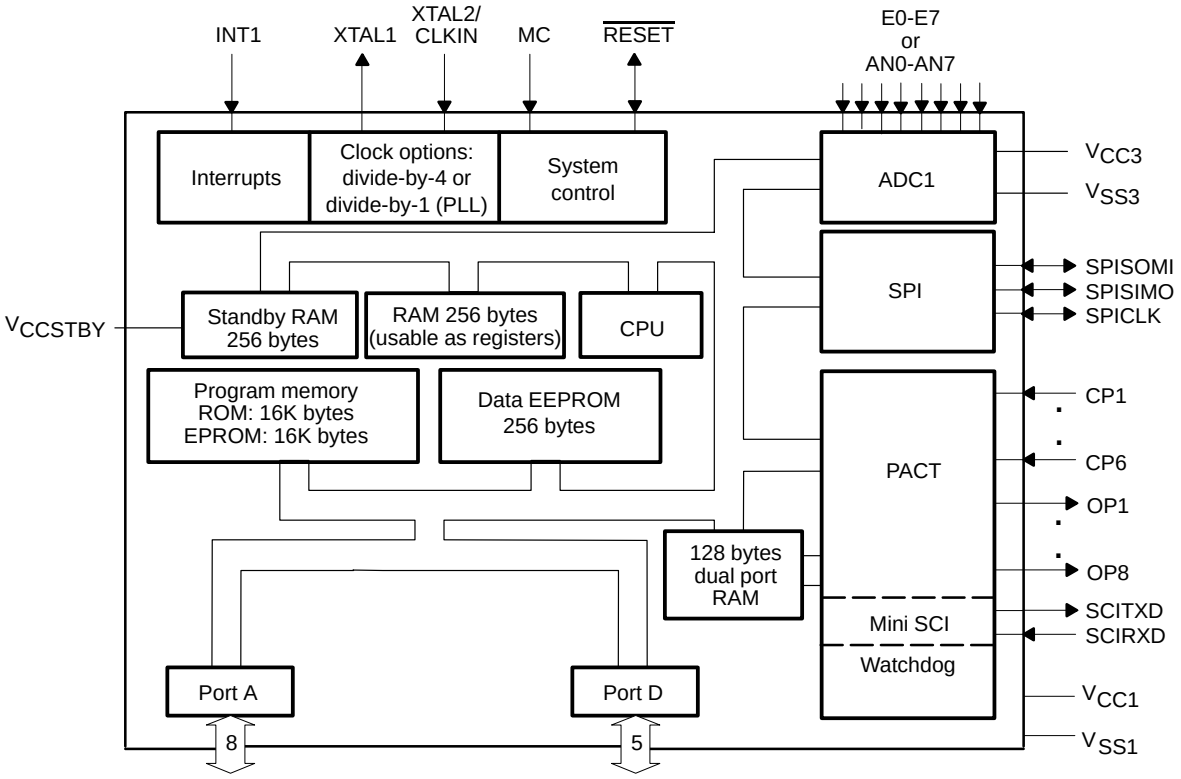


Figure 1–4. TMS370Cx32 Block Diagram



NOTE: Three of port D's four I/O buffers (D4, D6, and D7) are internally connected to three of the PACT module's inputs (CP3, CP4, and CP5). The actual pins are D4/CP3, D6/CP4, and D7/CP5.

Figure 1–5. TMS370Cx36 Block Diagram



Note: Four of port D's five I/O buffers (D4, D5, D6, and D7) are respectively connected internally to four of the PACT module's inputs (CP4, CP1, CP6, and CP5). The actual pins are D4/CP4, D5/CP1, D6/CP6, and D7/CP5.

Figure 1–6. TMS370Cx4x Block Diagram

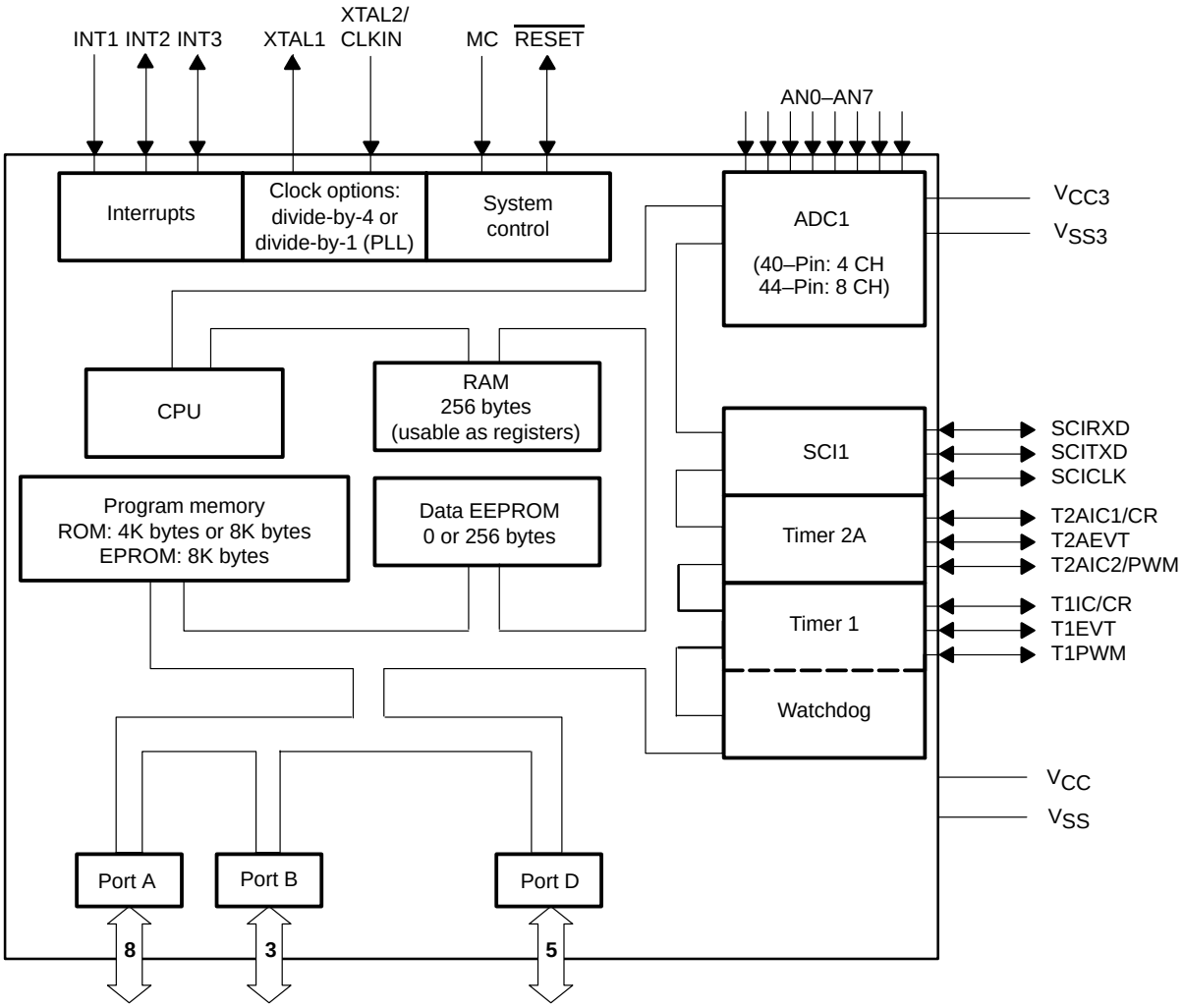
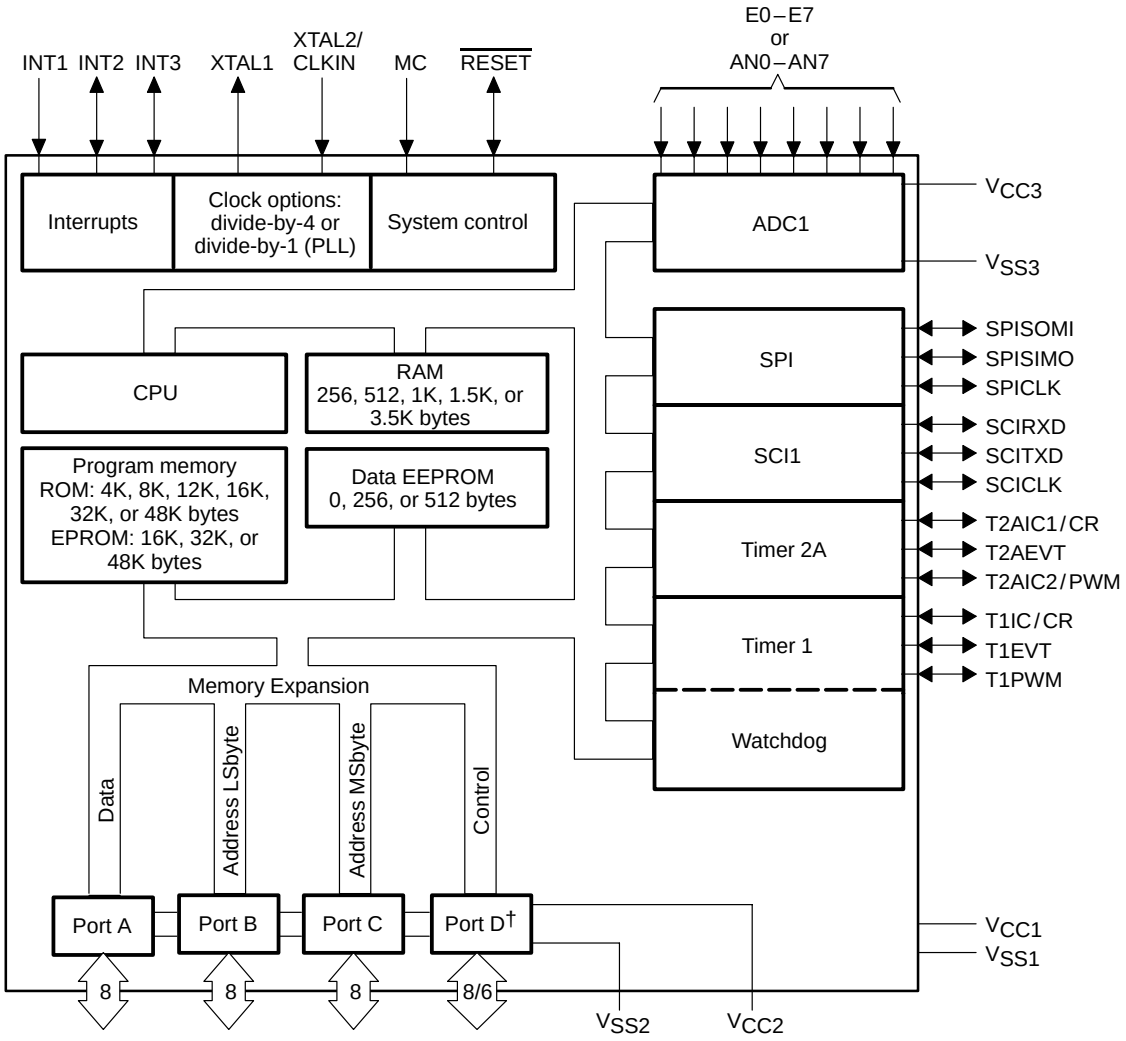


Figure 1–7. TMS370Cx5x Block Diagram



† For the 64-pin devices, there are only six pins for port D.

Figure 1–8. TMS370Cx6x Block Diagram

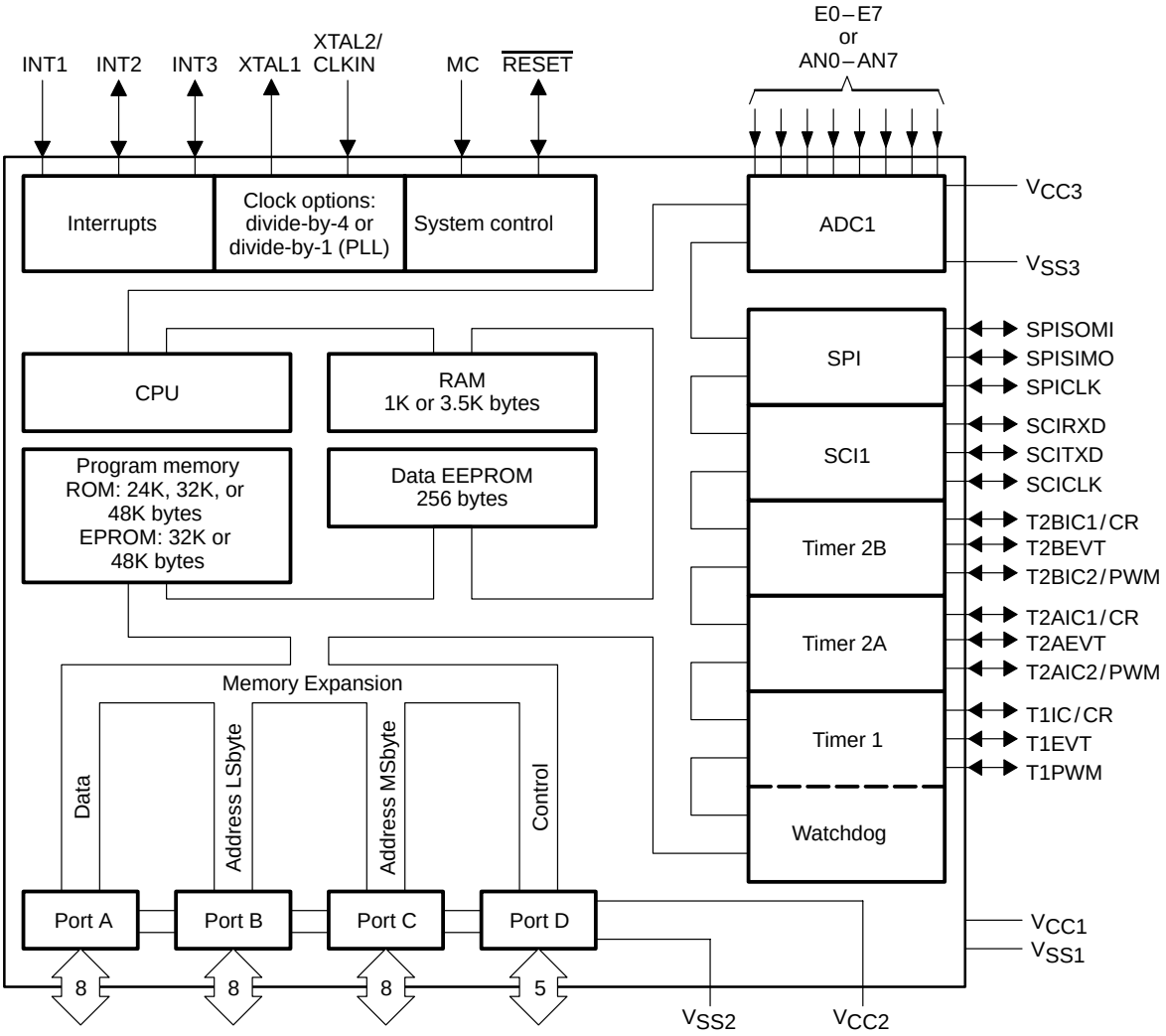
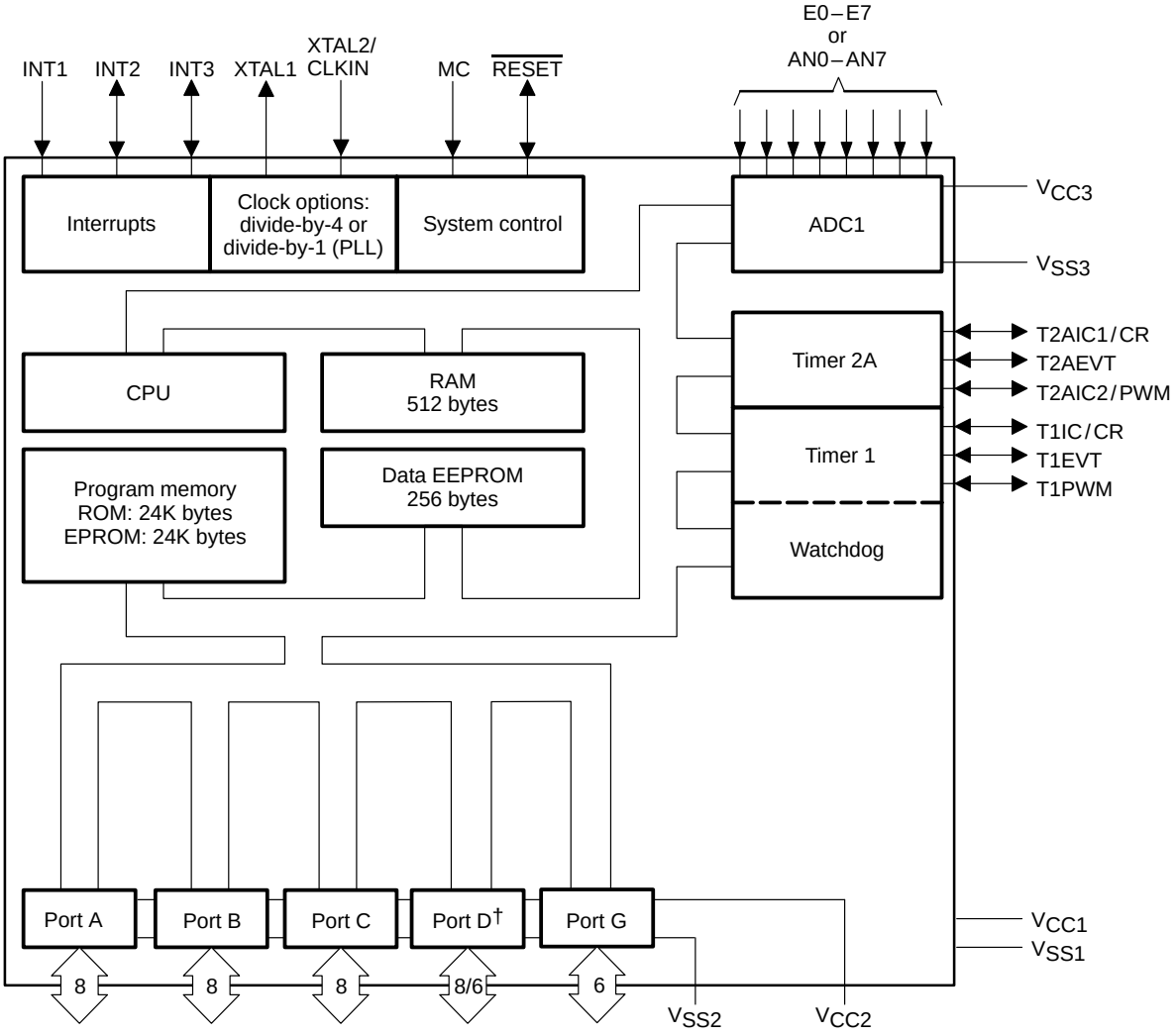
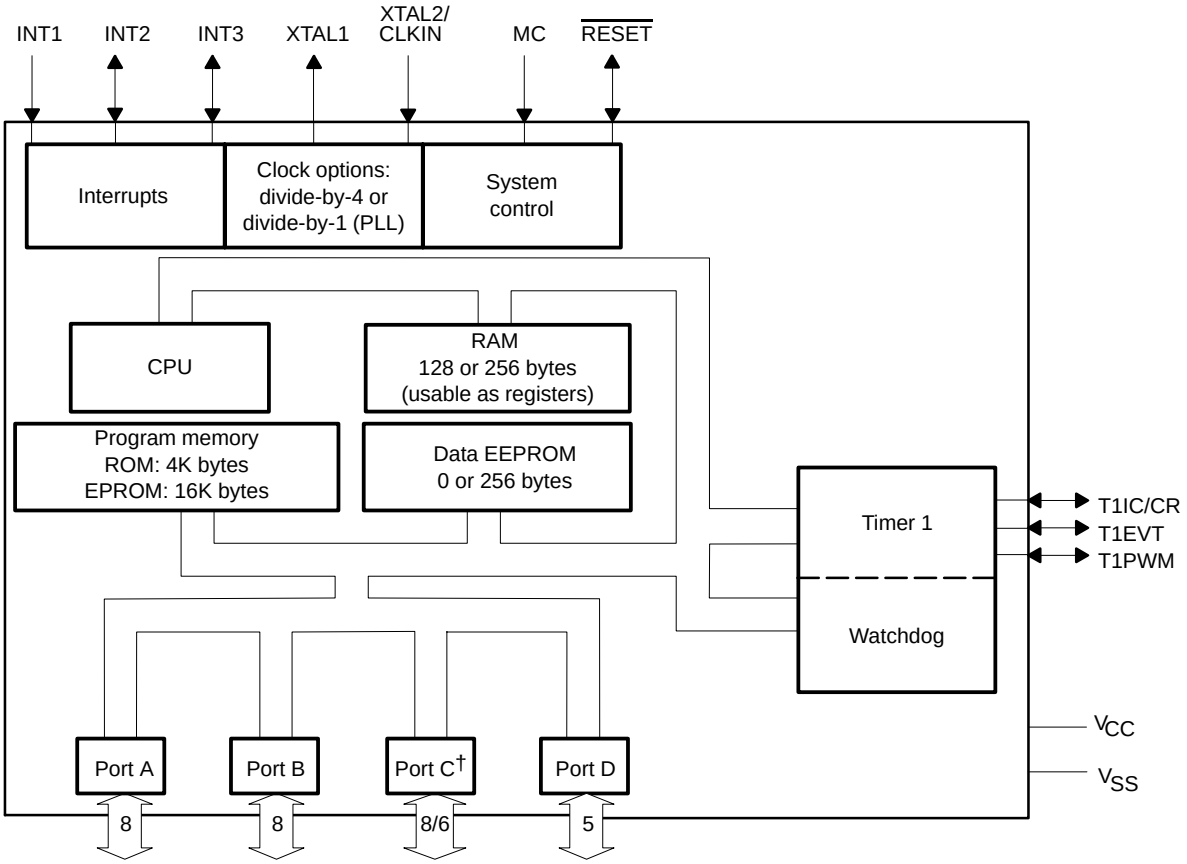


Figure 1–9. TMS370Cx7x Block Diagram



† For the 64-pin devices, there are only six pins for port D.

Figure 1–10. TMS370Cx8x Block Diagram



† For the 40-pin devices, there are only six pins for port C.

Figure 1–11. TMS370Cx9x Block Diagram

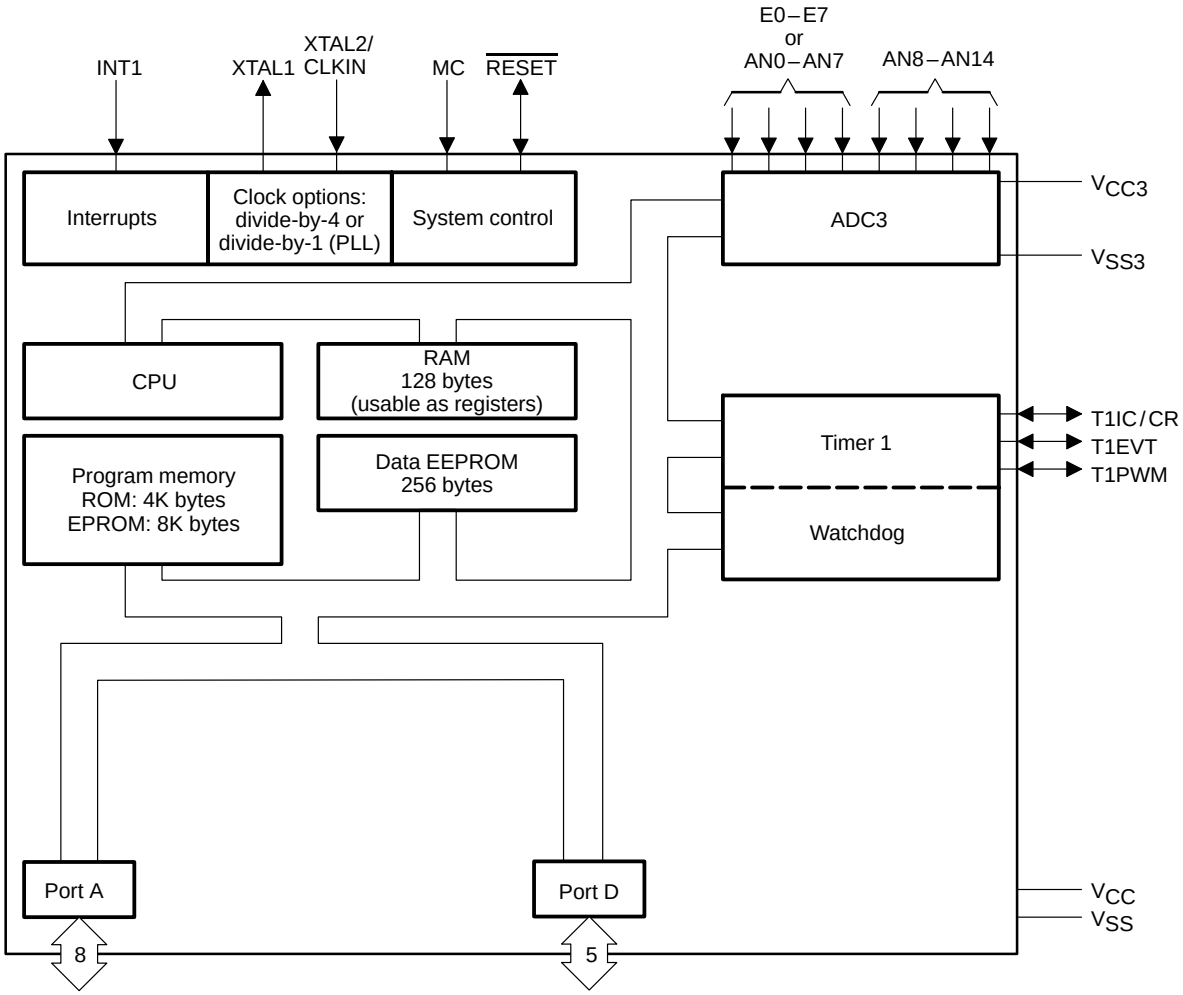


Figure 1–12. TMS370CxAx Block Diagram

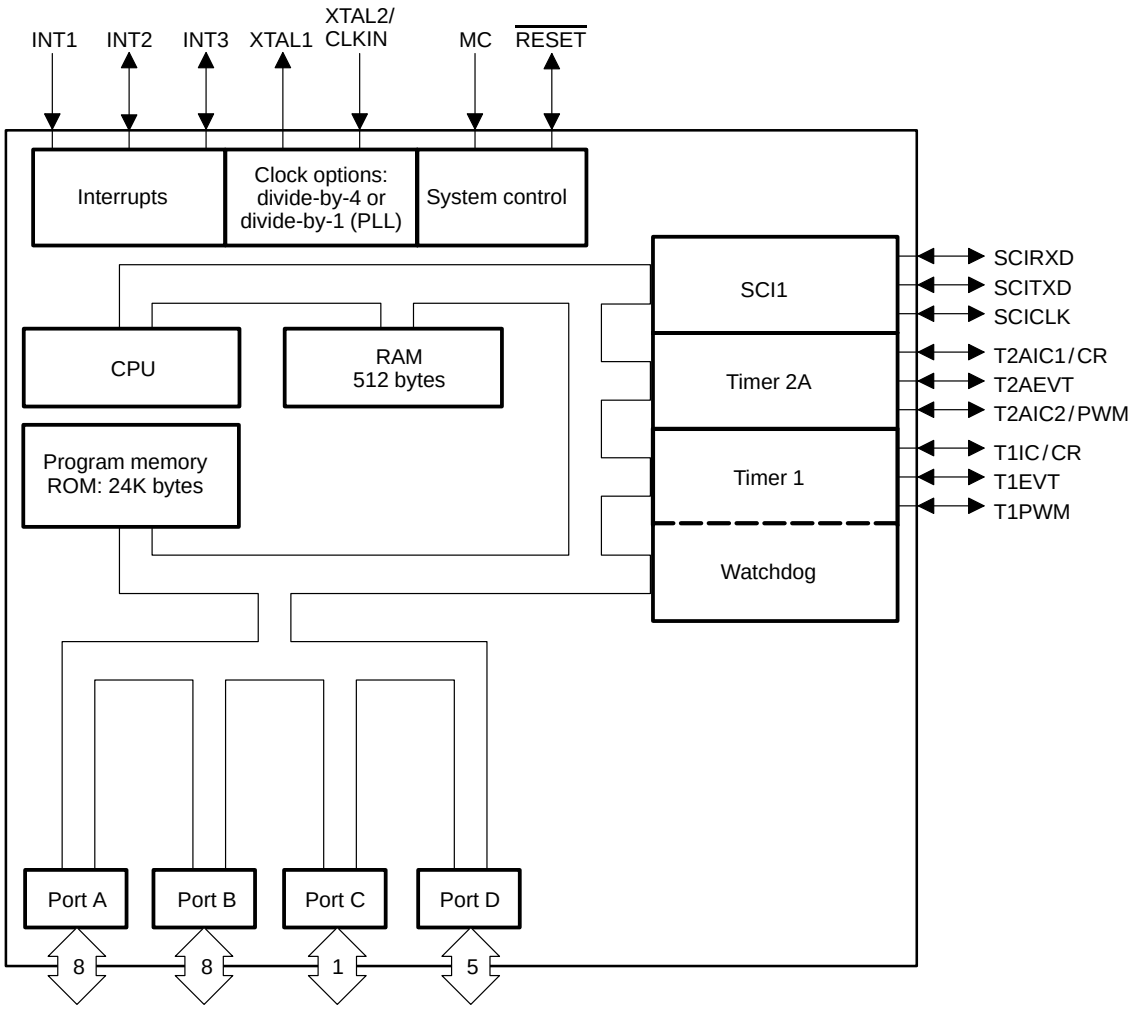
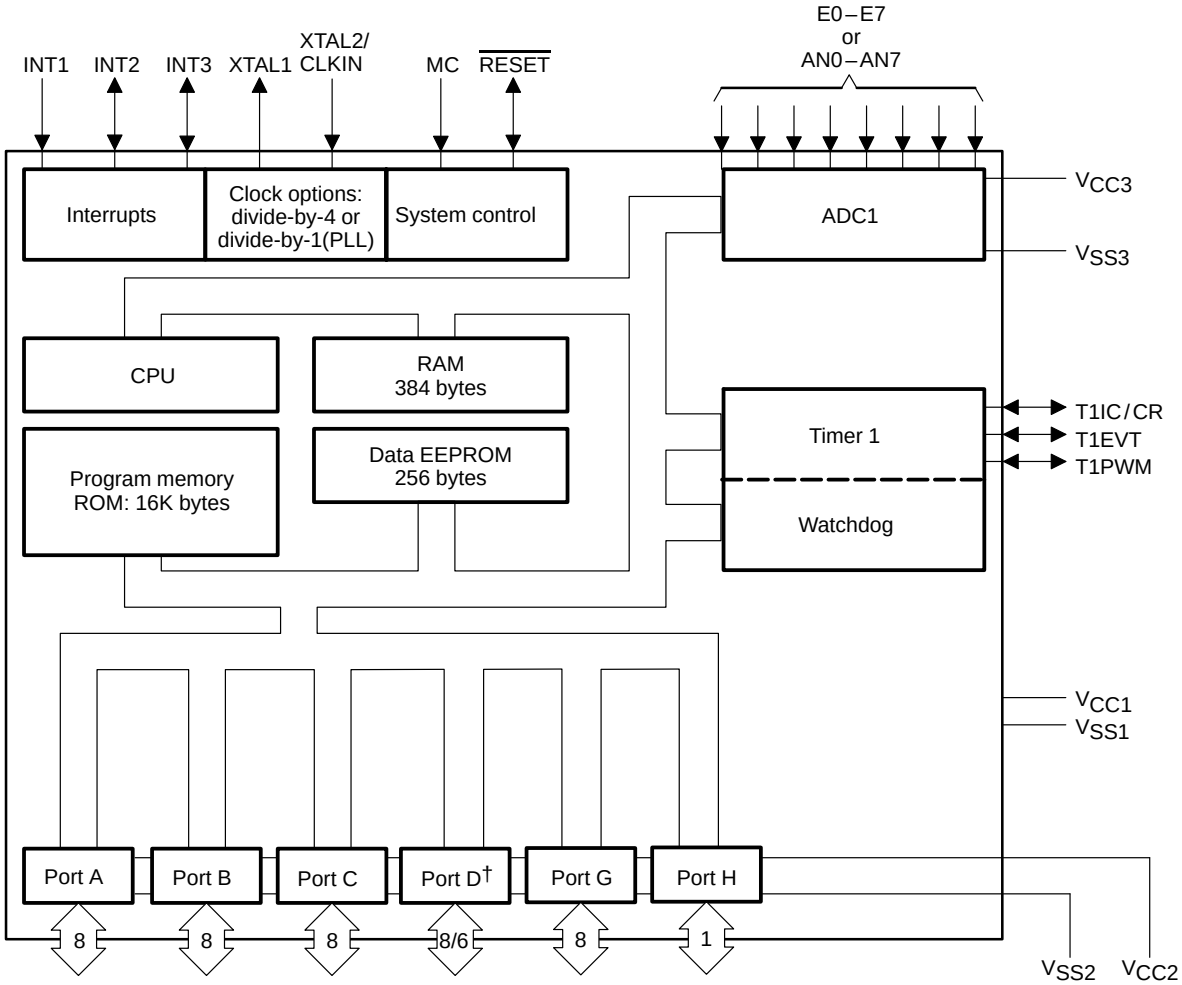
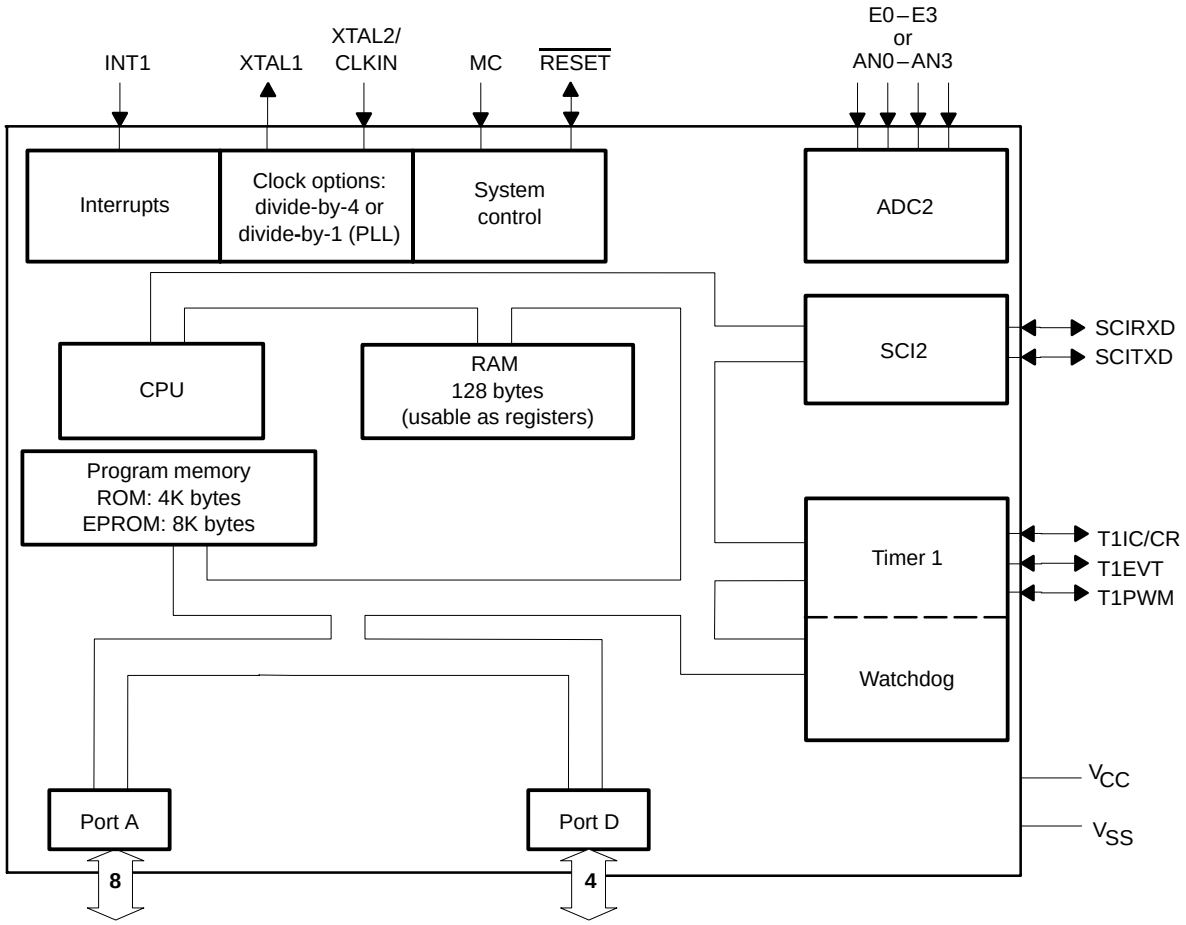


Figure 1–13. TMS370CxBx Block Diagram



† For the 64-pin devices, there are only six pins for port D.

Figure 1–14. TMS370CxCx Block Diagram



Chapter 2

TMS370 Family Pinouts and Pin Descriptions

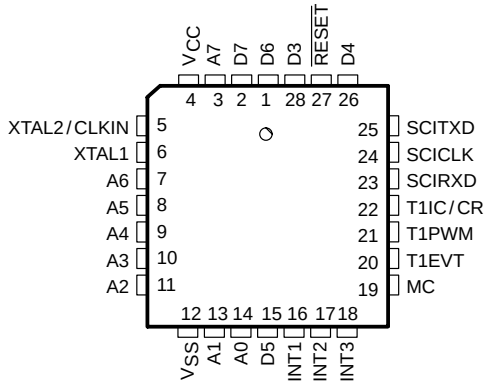
This chapter provides pinouts and pin descriptions for the individual device categories.

Topic **Page**

2.1 TMS370Cx0x Pinouts and Pin Descriptions

The pinouts and pin descriptions for the TMS370Cx0x devices are shown in Figure 2–1 and in Table 2–1, respectively.

Figure 2–1. Pinouts for TMS370Cx0x



A. 28-Pin PLCC (FN)
B. 28-Pin CLCC (FZ)

Table 2–1. TMS370Cx0x Pin Descriptions

28-Pin LCC		I/O†	Description
Name	**No.**		
A0	14	I/O	Port A is a general-purpose bidirectional I/O port.
A1	13	I/O	
A2	11	I/O	
A3	10	I/O	
A4	9	I/O	
A5	8	I/O	
A6	7	I/O	
A7	3	I/O	
D3	28	I/O	Port D is a general-purpose bidirectional I/O port. D3 is also configurable as SYSCLK.
D4	26	I/O	
D5	15	I/O	
D6	1	I/O	
D7	2	I/O	
INT1	16	I	External (non-maskable or maskable) interrupt/general-purpose input pin.
INT2	17	I/O	External maskable interrupt input/general-purpose bidirectional pin.
INT3	18	I/O	External maskable interrupt input/general-purpose bidirectional pin.
T1IC/CR	22	I/O	Timer1 input capture/counter reset input pin/general-purpose bidirectional pin.
T1PWM	21	I/O	Timer1 PWM output pin/general-purpose bidirectional pin.
T1EVT	20	I/O	Timer1 external event input pin/general-purpose bidirectional pin.
SCITXD	25	I/O	SCI transmit data output pin, general-purpose bidirectional pin (see Note).
SCIRXD	23	I/O	SCI receive data input pin/general-purpose bidirectional pin.
SCICLK	24	I/O	SCI bidirectional serial clock pin/general-purpose bidirectional pin.
$\overline{\text{RESET}}$	27	I/O	System reset bidirectional pin; as input pin, $\overline{\text{RESET}}$ initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by watchdog or oscillator fault circuit.
MC	19	I	Mode control input pin; enables EEPROM write protection override (WPO) mode, also supplies EPROM V_{PP}.
XTAL2/CLKIN	5	I	Internal oscillator crystal input/External clock source input.
XTAL1	6	O	Internal oscillator output for crystal.
V_{CC}	4		Positive supply voltage
V_{SS}	12		Ground reference

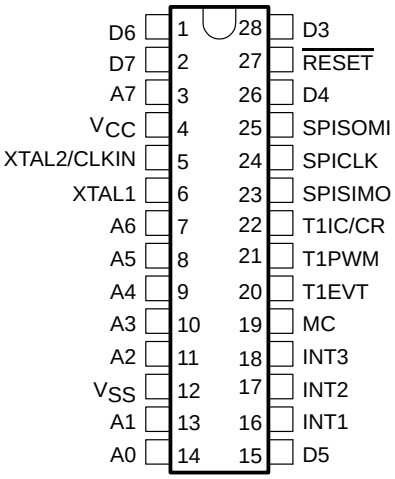
† I = input, O = output

Note: The three–pin SCI configuration is referred to as SCI1.

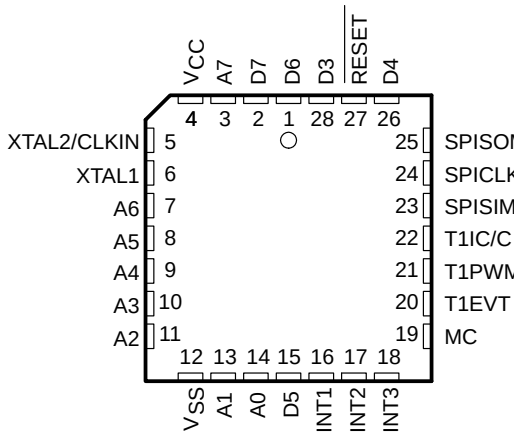
2.2 TMS370Cx1x Pinouts and Pin Descriptions

The pinouts and pin descriptions for the TMS370Cx1x devices are shown in Figure 2–2 and in Table 2–2, respectively.

Figure 2–2. Pinouts for TMS370Cx1x



A. 28-Pin PDIP (N)
B. 28-Pin CDIP (JD)



C. 28-Pin PLCC (FN)
D. 28-Pin CLCC (FZ)

Table 2–2. TMS370Cx1x Pin Descriptions

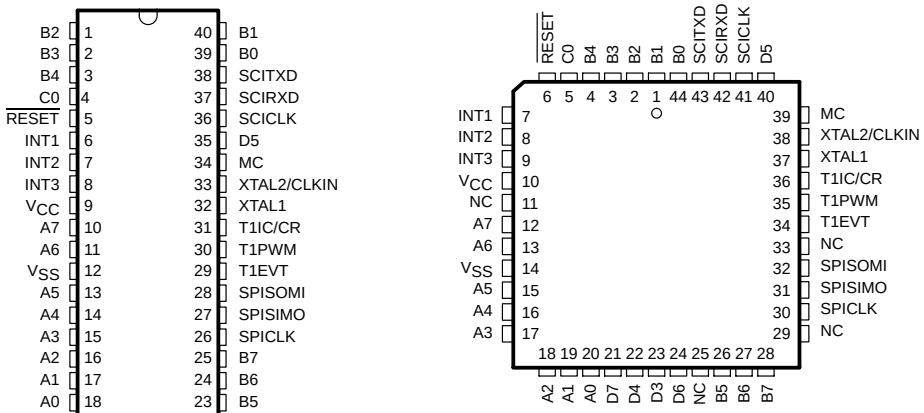
28–Pin DIP/LCC			
Name	**No.**	**I/O†**	**Description**
A0 A1 A2 A3 A4 A5 A6 A7	14 13 11 10 9 8 7 3	I/O I/O I/O I/O I/O I/O I/O I/O	Port A is a general-purpose bidirectional I/O port
D3 D4 D5 D6 D7	28 26 15 1 2	I/O I/O I/O I/O I/O	Port D is a general-purpose bidirectional I/O port; D3 is also configurable as SYSCLK
INT1 INT2 INT3	16 17 18	I I/O I/O	External (nonmaskable or maskable) interrupt/general-purpose input pin External maskable interrupt input/general-purpose bidirectional pin External maskable interrupt input/general-purpose bidirectional pin
T1IC/CR T1PWM T1EVT	22 21 20	I/O I/O I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin Timer 1 PWM output pin/general-purpose bidirectional pin Timer 1 external event input pin/general-purpose bidirectional pin
SPISOMI SPISIMO SPICLK	25 23 24	I/O I/O I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin SPI slave input pin, master output pin/general-purpose bidirectional pin SPI bidirectional serial clock pin/general-purpose bidirectional pin
$\overline{\text{RESET}}$	27	I/O	System reset bidirectional pin; as an input, $\overline{\text{RESET}}$ initializes the microcontroller; as an open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by the watchdog or oscillator fault circuit
MC	19	I	Mode control input pin; enables EEPROM write protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL2/CLKIN XTAL1	5 6	I O	Internal oscillator crystal input/external clock source input Internal oscillator output for crystal
V_{CC}	4		Positive supply voltage
V_{SS}	12		Ground reference

† I = input, O = output

2.3 TMS370Cx2x Pinouts and Pin Descriptions

The pinouts and pin descriptions for the TMS370Cx2x devices are shown in Figure 2–3 and Table 2–3, respectively.

Figure 2–3. Pinouts for TMS370Cx2x



A. 40-Pin PDIP (N)
B. 40-Pin CDIP (JD)
C. 40-Pin PSDIP (NJ)†
D. 40-Pin CSDIP (JC)
E. 44-Pin PLCC (FN)
F. 44-Pin CLCC (FZ)

† The NJ designator for the 40–pin plastic shrink DIP package was formerly known as N2. The mechanical drawing of the NJ is identical to the N2 package and did not need to be requalified.

Table 2–3. TMS370Cx2x Pin Descriptions

Pin				
	Pin No. DIP / LCC			
Name	**40**	**44**	**I/O†**	**Description**
A0 A1 A2 A3 A4 A5 A6 A7	18 17 16 15 14 13 11 10	20 19 18 17 16 15 13 12	I/O I/O I/O I/O I/O I/O I/O I/O	Port A is a general-purpose bidirectional I/O port
B0 B1 B2 B3 B4 B5 B6 B7	39 40 1 2 3 23 24 25	44 1 2 3 4 26 27 28	I/O I/O I/O I/O I/O I/O I/O I/O	Port B is a general-purpose bidirectional I/O port
C0	4	5	I/O	Port C is a general-purpose bidirectional I/O port
D3 D4 D5 D6 D7	21 20 35 22 19	23 22 40 24 21	I/O I/O I/O I/O I/O	Port D is a general-purpose bidirectional I/O port; D3 is also configurable as a SYSCLK
SCITXD SCIRXD SCICLK	38 37 36	43 42 41	I/O I/O I/O	SCI transmit data output pin/general-purpose bidirectional pin (see Note) SCI receive data input pin/general-purpose bidirectional pin SCI bidirectional serial clock pin/general-purpose bidirectional pin
INT1 INT2 INT3	6 7 8	7 8 9	I I/O I/O	External (nonmaskable or maskable) interrupt/general-purpose input pin External maskable interrupt input/general-purpose bidirectional pin External maskable interrupt input/general-purpose bidirectional pin
T1IC/CR T1PWM T1EVT	31 30 29	36 35 34	I/O I/O I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin Timer 1 PWM output pin/general-purpose bidirectional pin Timer 1 external event input pin/general-purpose bidirectional pin
SPISOMI SPISIMO SPICLK	28 27 26	32 31 30	I/O I/O I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin SPI slave input pin, master output pin/general-purpose bidirectional pin SPI bidirectional serial clock pin/general-purpose bidirectional pin
$\overline{\text{RESET}}$	5	6	I/O	System reset bidirectional pin; as an input, $\overline{\text{RESET}}$ initializes the microcontroller; as an open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit
MC	34	39	I	Mode control input pin; enables the EEPROM write-protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL1 XTAL2/CLKIN	32 33	37 38	O I	Internal oscillator output for crystal Internal oscillator crystal input/external clock source input
NC	– – – –	11 25 29 33		No connections
V_{CC} V_{SS}	9 12	10 14		Positive supply voltage Ground reference

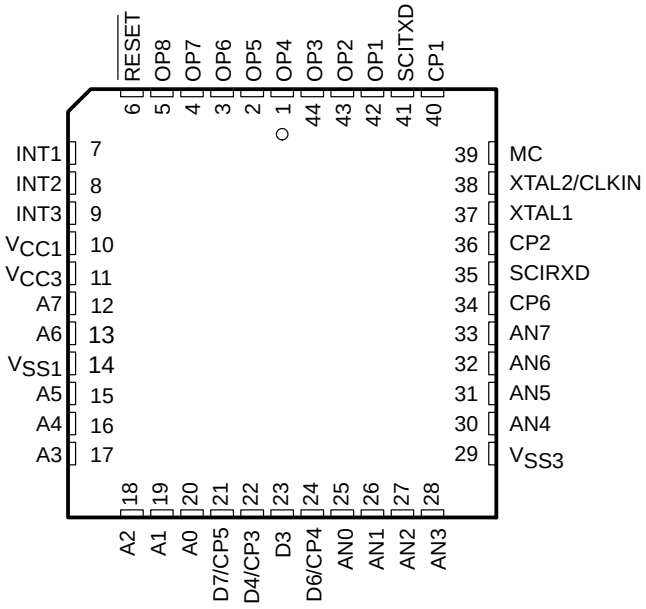
† I = input, O = output

Note: The three–pin SCI configuration is referred to as SCI1.

2.4 TMS370Cx32 Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx32 devices are shown in Figure 2–4 and Table 2–4, respectively.

Figure 2–4. Pinout for TMS370Cx32



A. 44-Pin PLCC (FN)
B. 44-Pin CLCC (FZ)

Table 2–4. TMS370x32 Pin Descriptions

44–Pin LCC			
Name	**No.**	**I/O‡**	**Description**
A0	20	I/O	Port A is a general-purpose bidirectional port
A1	19	I/O	
A2	18	I/O	
A3	17	I/O	
A4	16	I/O	
A5	15	I/O	
A6	13	I/O	
A7	12	I/O	
			Port D is a general-purpose bidirectional port
D3	23	I/O	I/O pin: Also configurable as SYSCLK
D4/CP3	22	I/O	I/O pin: PACT input capture 3.†
D6/CP4	24	I/O	I/O pin: PACT input capture 4.†
D7/CP5	21	I/O	I/O pin: PACT input capture 5.†
INT1	7	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
INT2	8	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	9	I/O	External maskable interrupt input/general-purpose bidirectional pin
CP1	40	I	PACT input capture pin 1
CP2	36	I	PACT input capture pin 2
CP6	34	I	PACT input capture pin 6; external event input pin (for event counter)
SCITXD	41	O	PACT mini SCI transmit output pin
SCIRXD	35	I	PACT mini SCI receive input pin
OP1	42	O	PACT output pin 1
OP2	43	O	PACT output pin 2
OP3	44	O	PACT output pin 3
OP4	1	O	PACT output pin 4
OP5	2	O	PACT output pin 5
OP6	3	O	PACT output pin 6
OP7	4	O	PACT output pin 7
OP8	5	O	PACT output pin 8
AN0	25	I	ADC1 analog input (AN0–AN7) or positive reference pins (AN1–AN7)
AN1	26	I	
AN2	27	I	
AN3	28	I	
AN4	30	I	The analog port can be individually programmed as general-purpose input pins if it is
AN5	31	I	not used as ADC1 converter analog input or positive reference input
AN6	32	I	
AN7	33	I	
$\overline{\text{RESET}}$	6	I/O	System reset bidirectional pin; as input, $\overline{\text{RESET}}$ initializes the microcontroller; as an open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by the watchdog or oscillator fault circuit
MC	39	I	Mode control input pin; enables EEPROM write-protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL2/CLKIN	38	I	Internal oscillator crystal input/external clock source input
XTAL1	37	O	Internal oscillator output for crystal

† Some of port D's digital I/O buffers are internally connected to some of the PACT module's input capture pins. This allows the microcontroller to read the level on the input capture pin or, if the port D pin is configured as an output, to generate a capture. Be careful to leave the port D pin configured as an input if the corresponding input capture pin is being driven by external circuitry.

‡ I = input, O = output

Table 2–4. TMS370x32 Pin Descriptions (Continued)

44–Pin LCC		I/O‡	Description
Name	**No.**		
V_{CC1}	10		Positive supply voltage for digital logic and digital I/O pins
V_{SS1}	14		Ground reference for digital logic and digital I/O pins
V_{CC3}	11		ADC1 converter positive supply voltage and optional positive reference input
V_{SS3}	29		ADC1 converter ground supply and low reference input pin

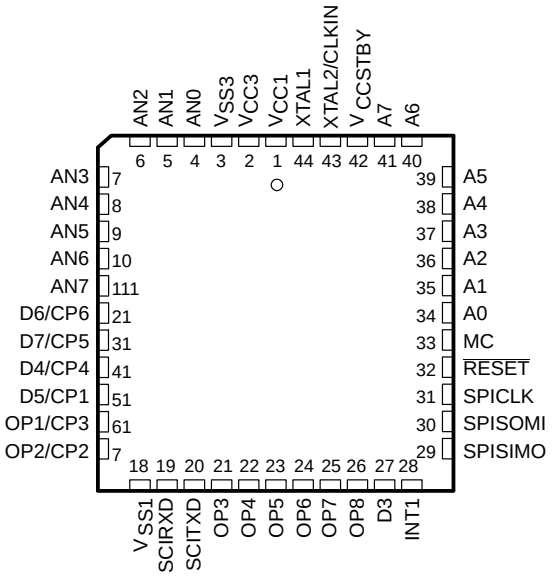
† Some of port D's digital I/O buffers are internally connected to some of the PACT module's input capture pins. This allows the microcontroller to read the level on the input capture pin or, if the port D pin is configured as an output, to generate a capture. Be careful to leave the port D pin configured as an input if the corresponding input capture pin is being driven by external circuitry.

‡ I = input, O = output

2.5 TMS370Cx36 Pinout and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx36 device family are shown in Figure 2–5 and Table 2–5, respectively.

Figure 2–5. Pinout for TMS370Cx36



A. 44-Pin PLCC (FN)
B. 44-Pin CLCC (FZ)

Table 2–5. TMS370Cx36 Pin Descriptions

44–Pin LCC			
Name	**No.**	**I/O†**	**Description**
A0	34	I/O	Port A is a general-purpose bidirectional I/O port.
A1	35	I/O	
A2	36	I/O	
A3	37	I/O	
A4	38	I/O	
A5	39	I/O	
A6	40	I/O	
A7	41	I/O	
D3	27	I/O	Port D is a general-purpose bidirectional port. Also configurable as SYSCLK (Note 1).
D4/CP4	14	I/O	PACT input capture 4 (Note 2).
D5/CP1	15	I/O	PACT input capture 1 (Note 2).
D6/CP6	12	I/O	PACT input capture 6 (Note 2).
D7/CP5	13	I/O	PACT input capture 5 (Note 2).
AN0/E0	4	I	ADC1 analog input pins (AN0–AN7)/port E digital input pins (E0–E7).
AN1/E1	5	I	
AN2/E2	6	I	Port E can be programmed individually as a general-purpose digital input pin if it is not
AN3/E3	7	I	used as ADC1 analog input or positive reference input.
AN4/E4	8	I	
AN5/E5	9	I	
AN6/E6	10	I	
AN7/E7	11	I	
INT1	28	I	External (non-maskable or maskable) interrupt/general-purpose input pin
OP1/CP3	16	O	PACT PWM output 1/input capture 3 (Note 3).
OP2/CP2	17	O	PACT output pin 2/input capture 2 (Note 3).
OP3	21	O	PACT PWM output 3
OP4	22	O	PACT PWM output 4
OP5	23	O	PACT PWM output 5
OP6	24	O	PACT PWM output 6
OP7	25	O	PACT PWM output 7
OP8	26	O	PACT PWM output 8
SCIRXD	19	I	PACT mini SCI data receive input pin
SCITXD	20	O	PACT mini SCI data transmit output pin
SPISOMI	30	I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin
SPISIMO	29		SPI slave input pin, master output pin/general-purpose bidirectional pin
SPICLK	31		SPI bidirectional serial clock pin/general-purpose bidirectional pin

† I = input, O = output

NOTES: 1. D3 can be configured as SYSCLK by appropriately programming the DPORT1 and DPORT2 registers.
2. These digital I/O buffers are connected internally to some of the PACT module's input capture pins. This allows the microcontroller to read the level on the input capture pin, or if the port D pin is configured as an output, to generate a capture. Be careful to leave the port D pin configured as an input if the corresponding input capture pin is being driven by external circuitry.
3. CP2 and CP3 are connected internally to OP2 and OP1. CP2 and CP3 can be used only to capture, respectively, OP2 and OP1, and not as external capture inputs.

Table 2–5 TMS370Cx36 Pin Descriptions (Continued)

44–Pin LCC		I/O†	Description
Name	**No.**		
RESET	32	I/O	System reset bidirectional pin; as input pin, $\overline{\text{RESET}}$ initializes the microcontroller; as open–drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by the watchdog or oscillator fault circuit.
MC	33	I	Mode control input pin; enables EEPROM write protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL2/CLKIN XTAL1	43 44	I O	Internal oscillator crystal output/external clock source input Internal oscillator output for crystal
V_{CC1} V_{SS1} V_{CC3} V_{SS3} V_{CCSTBY}	1 18 2 3 42		Positive supply voltage for digital logic and digital I/O pins Ground reference for digital logic and digital I/O pins ADC1 positive supply voltage and optional positive reference input ADC1 ground supply and low reference input pin Positive supply voltage pin for standby RAM

† I = input, O = output

NOTES: 4. D3 can be configured as SYSCLK by appropriately programming the DPORT1 and DPORT2 registers.

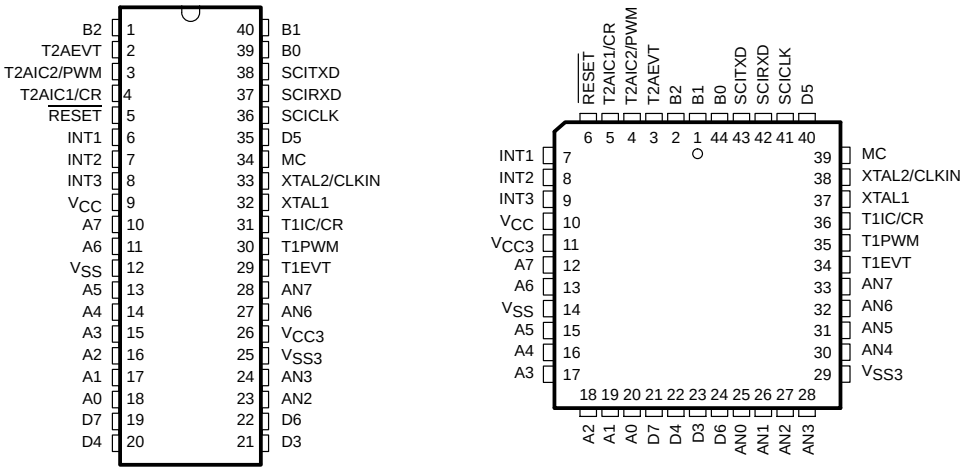
5. These digital I/O buffers are connected internally to some of the PACT module's input capture pins. This allows the microcontroller to read the level on the input capture pin, or if the port D pin is configured as an output, to generate a capture. Be careful to leave the port D pin configured as an input if the corresponding input capture pin is being driven by external circuitry.

6. CP2 and CP3 are connected internally to OP2 and OP1. CP2 and CP3 can be used only to capture, respectively, OP2 and OP1, and not as external capture inputs.

2.6 TMS370Cx4x Pinouts and Pin Descriptions

The pinouts and pin descriptions for the TMS370Cx4x devices are shown in Figure 2–6 and Table 2–6, respectively.

Figure 2–6. Pinouts for TMS370Cx4x



A. 40-Pin PDIP (N)
B. 40-Pin CDIP (JD)
C. 40-Pin PSDIP (NJ)†
D. 40-Pin CSDIP (JC)

E. 44-Pin PLCC (FN)
F. 44-Pin CLCC (FZ)

† The NJ designator for the 40–pin plastic shrink DIP package was formerly known as N2. The mechanical drawing of the NJ is identical to the N2 package and did not need to be requalified.

Table 2–6. TMS370Cx4x Pin Descriptions

Pin			I/O†	Description
Name	**DIP (40)**	**LCC (44)**		
A0	18	20	I/O	Port A is a general-purpose bidirectional I/O port
A1	17	19	I/O	
A2	16	18	I/O	
A3	15	17	I/O	
A4	14	16	I/O	
A5	13	15	I/O	
A6	11	13	I/O	
A7	10	12	I/O	
B0	39	44	I/O	Port B is a general-purpose bidirectional I/O port
B1	40	1	I/O	
B2	1	2	I/O	
D3	21	23	I/O	Port D is a general-purpose bidirectional I/O port; D3 is also configurable as a SYSCLK
D4	20	22	I/O	
D5	35	40	I/O	
D6	22	24	I/O	
D7	19	21	I/O	
AN0/E0	—	25	I	ADC1 analog input channels or positive reference pins; any ADC1 channel can be programmed as general-purpose input pins (E port) if not used as an analog input or reference channel
AN1/E1	—	26	I	
AN2/E2	23	27	I	
AN3/E3	24	28	I	
AN4/E4	—	30	I	
AN5/E5	—	31	I	
AN6/E6	27	32	I	
AN7/E7	28	33	I	
V_{CC3}	26	11		ADC1 converter positive supply voltage and optional positive reference input pin
V_{SS3}	25	29		ADC1 converter ground supply and low reference input pin
INT1	6	7	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
INT2	7	8	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	8	9	I/O	External maskable interrupt input/general-purpose bidirectional pin
T1IC/CR	31	36	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	30	35	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin
T1EVT	29	34	I/O	Timer 1 external event input pin/general-purpose bidirectional pin
T2AIC1/CR	4	5	I/O	Timer 2A input capture/counter reset input pin/general-purpose bidirectional pin
T2AIC2/PWM	3	4	I/O	Timer 2A input capture 2/PWM output pin/general-purpose bidirectional pin
T2AEVT	2	3	I/O	Timer 2A external event Input pin/general-purpose bidirectional pin
SCITXD	38	43	I/O	SCI transmit data output pin/general-purpose bidirectional pin (see Note)
SCIRXD	37	42	I/O	SCI receive data input pin/general-purpose bidirectional pin
SCICLK	36	41	I/O	SCI bidirectional serial clock pin/general-purpose bidirectional pin
$\overline{\text{RESET}}$	5	6	I/O	System reset bidirectional pin; as input, $\overline{\text{RESET}}$ initializes the microcontroller; as an open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by the watchdog or oscillator fault circuit
MC	34	39	I	Mode control input pin; enables the EEPROM write protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL1	32	37	I	Internal oscillator output for crystal
XTAL2/CLKIN	33	38	O	Internal oscillator crystal input/external clock source input
V_{CC}	9	10		Positive supply voltage
V_{SS}	12	14		Ground reference

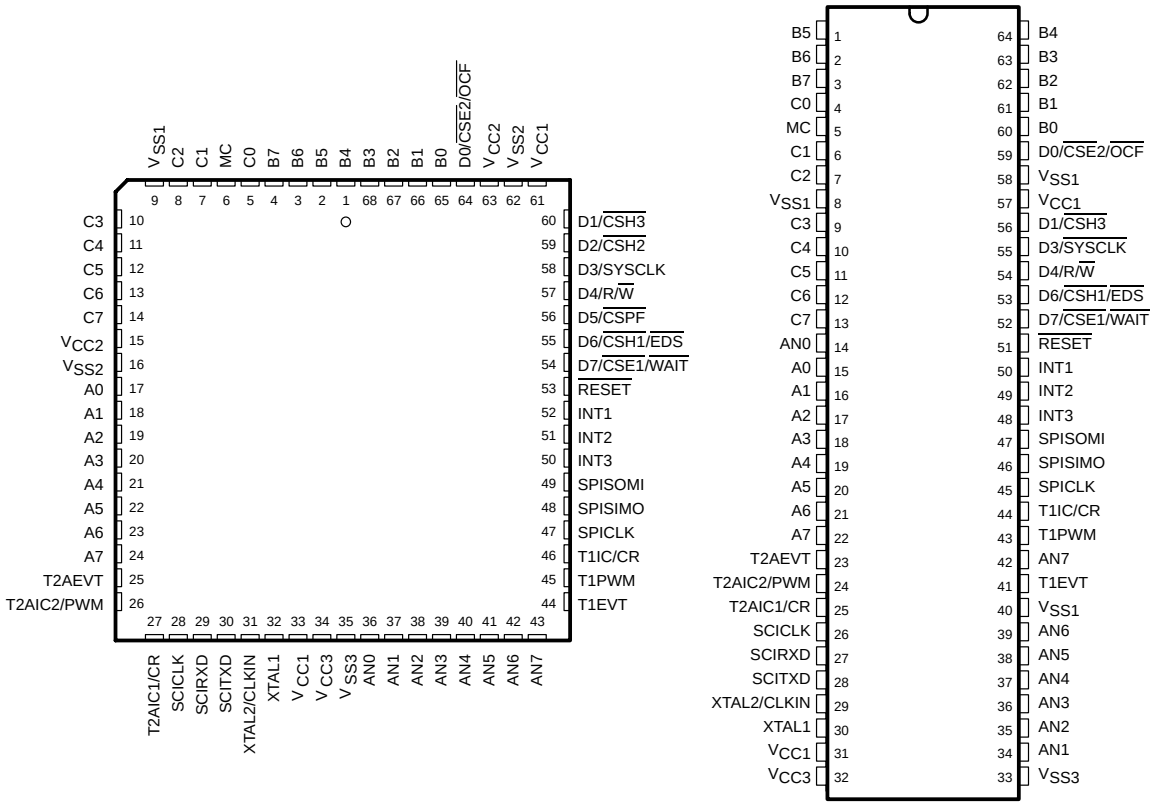
† I = input, O = output

Note: The three–pin SCI configuration is referred to as SCI1.

2.7 TMS370Cx5x Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx5x devices are shown in Figure 2–7 and Table 2–7, respectively.

Figure 2–7. Pinouts for TMS370Cx5x



A. 68-Pin PLCC (FN)
B. 68-Pin CLCC (FZ)
C. 64-Pin PSDIP (NM)
D. 64-Pin CSDIP (JN)

Table 2–7. TMS370Cx5x Pin Descriptions

Pin					
Name	**Alternate Function**	**DIP (64)**	**LCC (68)**	**I/O†**	**Description**
A0	DATA0 (LSB)	15	17	I/O	Single-chip mode: port A is a general-purpose bidirectional port Expansion mode: port A can be individually programmed as the external bidirectional data bus (DATA0–DATA7)
A1	DATA1	16	18	I/O	
A2	DATA2	17	19	I/O	
A3	DATA3	18	20	I/O	
A4	DATA4	19	21	I/O	
A5	DATA5	20	22	I/O	
A6	DATA6	21	23	I/O	
A7	DATA7 (MSB)	22	24	I/O	
B0	ADDR0	60	65	I/O	Single-chip mode: port B is a general-purpose bidirectional I/O port Expansion modes: port B can be individually programmed as the low-order address output bus (ADDR0–ADDR7)
B1	ADDR1	61	66	I/O	
B2	ADDR2	62	67	I/O	
B3	ADDR3	63	68	I/O	
B4	ADDR4	64	1	I/O	
B5	ADDR5	1	2	I/O	
B6	ADDR6	2	3	I/O	
B7	ADDR7	3	4	I/O	
C0	ADDR8	4	5	I/O	Single-chip mode: port C is a general-purpose bidirectional I/O port Expansion mode: port C can be individually programmed as the high-order address output bus (ADDR8–ADDR15)
C1	ADDR9	6	7	I/O	
C2	ADDR10	7	8	I/O	
C3	ADDR11	9	10	I/O	
C4	ADDR12	10	11	I/O	
C5	ADDR13	11	12	I/O	
C6	ADDR14	12	13	I/O	
C7	ADDR15	13	14	I/O	
INT1	NMI	50	52	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
INT2	–	49	51	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	–	48	50	I/O	External maskable interrupt input/general-purpose bidirectional pin

† I = input, O = output

Table 2–7. TMS370Cx5x Pin Descriptions (Continued)

Pin					I/O†	Description
Name	Alternate Function		DIP (64)	LCC (68)		
	Function					Single-chip mode: port D is a general-purpose bidirectional I/O port. Each of the port D pins can be individually configured as a general-purpose I/O pin, primary memory control signal (function A), or secondary memory control signal (function B). All chip selects are independent and can be used for memory bank switching.
	A	B				
D0	$\overline{\text{CSE2}}$	$\overline{\text{OCF}}$	59	64	I/O	I/O pin/A: chip-select eighth output 2 goes low during memory accesses. I/O pin/B: Opcode fetch goes low during the opcode fetch memory cycle
D1	$\overline{\text{CSH3}}$	—	56	60	I/O	I/O pin/A: chip-select half output 3 goes low during memory accesses. I/O pin/B: reserved.
D2	$\overline{\text{CSH2}}$	—	—	59	I/O	I/O pin/A: chip-select half output 2 goes low during memory accesses. I/O pin/B: reserved.
D3‡	SYSCLK	SYSCLK	55	58	I/O	I/O pin/A, B: internal clock signal is 1/1 (PLL) or 1/4 XTAL2/CLKIN frequency.
D4	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$	54	57	I/O	I/O pin/A, B: read/write output pin
D5	$\overline{\text{CSPF}}$	—	—	56	I/O	I/O pin/A: chip-select peripheral output for peripheral file goes low during memory accesses. I/O pin/B: reserved.
D6	$\overline{\text{CSH1}}$	$\overline{\text{EDS}}$	53	55	I/O	I/O pin/A: chip-select half output 1 goes low during memory accesses. I/O pin/B: external data strobe output goes low during memory accesses from external memory and has the same timings as the five chip selects.
D7	$\overline{\text{CSE1}}$	$\overline{\text{WAIT}}$	52	54	I/O	I/O pin/A: chip-select eighth output goes low during memory accesses. I/O pin/B: wait input pin extends bus signals
T1IC/CR	T1IO1		44	46	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	T1IO2		43	45	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin
T1EVT	T1IO3		41	44	I/O	Timer 1 external event input pin/general-purpose bidirectional pin
T2AIC1/CR	T2IO1		25	27	I/O	Timer 2A input capture 1/counter reset input pin/general-purpose bidirectional pin
T2AIC2/PWM	T2IO2		24	26	I/O	Timer 2A input capture 2/PWM output pin/general-purpose bidirectional pin
T2AEVT	T2IO3		23	25	I/O	Timer 2A external event input pin/general-purpose bidirectional pin
SPISOMI	SPIIO1		47	49	I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin
SPISIMO	SPIIO2		46	48	I/O	SPI slave input pin, master output pin/general-purpose bidirectional pin
SPICLK	SPIIO3		45	47	I/O	SPI bidirectional serial clock pin/general-purpose bidirectional pin

† I = input, O = output
‡ Port D3 can be configured as SYSCLK

Table 2–7. TMS370Cx5x Pin Descriptions (Continued)

Pin				I/O†	Description
Name	**Alternate Function**	**DIP (64)6**	**LCC (68)**		
SCITXD	SCIIO1	28	30	I/O	SCI transmit data output pin/general-purpose bidirectional pin (see Note)
SCIRXD	SCIIO2	27	29	I/O	SCI receive data input pin/general-purpose bidirectional pin
SCICLK	SCIIO3	26	28	I/O	SCI bidirectional serial clock pin/general-purpose bidirectional pin
AN0	E0	14	36	I	ADC1 analog input (AN0–AN7) or positive reference pins (AN1–AN7) Port E can be individually programmed as general-purpose input pins if not used as ADC1 analog input or positive reference input
AN1	E1	34	37	I	
AN2	E2	35	38	I	
AN3	E3	36	39	I	
AN4	E4	37	40	I	
AN5	E5	38	41	I	
AN6	E6	39	42	I	
AN7	E7	42	43	I	
V_{CC3}		32	34		ADC1 positive supply voltage and optional positive reference input pin
V_{SS3}		33	35		ADC1 ground supply and low reference input pin
$\overline{\text{RESET}}$		51	53	I/O	System reset bidirectional pin; as an input, $\overline{\text{RESET}}$ initializes microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit
MC		5	6	I	Mode control pin; enables EEPROM write protection override (WPO) mode, and also supplies EPROM Vpp.
XTAL2/CLKIN		29	31	I	Internal oscillator crystal input/external clock source input
XTAL1		30	32	O	Internal oscillator output for crystal
V_{CC1}		31, 57	33, 61		Positive supply voltage
V_{CC2}		—	15, 63		Positive supply voltage for digital I/O logic
V_{SS1}		8, 58	9		Ground reference for digital logic
V_{SS2}		—	16, 62		Ground reference for digital I/O pins

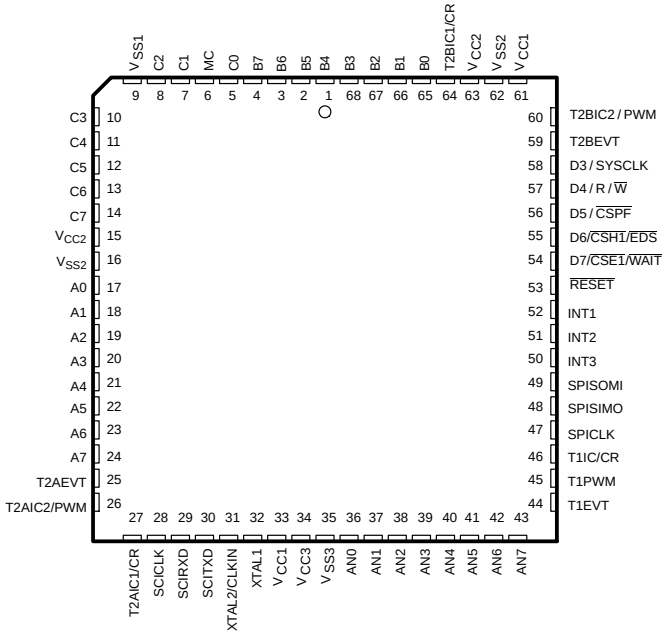
† I = input, O = output

Note: The three–pin SCI configuration is referred to as SCI1.

2.8 TMS370Cx6x Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx6x devices are shown in Figure 2–8 and Table 2–8, respectively.

Figure 2–8. Pinouts for TMS370Cx6x



A. 68-Pin PLCC (FN)
B. 68-Pin CLCC (FZ)

Table 2–8. TMS370Cx6x Pin Descriptions

Pin			I/O†	Description‡
Name	**Alternate Function**	**LCC (68)**		
A0 A1 A2 A3 A4 A5 A6 A7	DATA0 DATA1 DATA2 DATA3 DATA4 DATA5 DATA6 DATA7	17 18 19 20 21 22 23 24	I/O I/O I/O I/O I/O I/O I/O I/O	Single-chip mode: Port A is a general-purpose bidirectional I/O port. Expansion mode: Port A can be individually programmed as the external bidirectional data bus (DATA0–DATA7).
B0 B1 B2 B3 B4 B5 B6 B7	ADDR0 ADDR1 ADDR2 ADDR3 ADDR4 ADDR5 ADDR6 ADDR7	65 66 67 68 1 2 3 4	I/O I/O I/O I/O I/O I/O I/O I/O	Single-chip mode: Port B is a general-purpose bidirectional I/O port. Expansion mode: Port B can be individually programmed as the low-order address output bus (ADDR0–ADDR7).
C0 C1 C2 C3 C4 C5 C6 C7	ADDR8 ADDR9 ADDR10 ADDR11 ADDR12 ADDR13 ADDR14 ADDR15	5 7 8 10 11 12 13 14	I/O I/O I/O I/O I/O I/O I/O I/O	Single-chip mode: Port C is a general-purpose bidirectional I/O port. Expansion mode: Port C can be individually programmed as the high-order address output bus (ADDR8–ADDR15).
INT1 INT2 INT3	NMI — —	52 51 50	I I/O I/O	External (nonmaskable or maskable) interrupt/general-purpose input pin External maskable interrupt input/general-purpose bidirectional pin External maskable interrupt input/general-purpose bidirectional pin
AN0 AN1 AN2 AN3 AN4 AN5 AN6 AN7	E0 E1 E2 E3 E4 E5 E6 E7	36 37 38 39 40 41 42 43	I I I I I I I I	ADC1 analog input (AN0–AN7) or positive reference pins (AN1–AN7) Port E can be individually programmed as general-purpose input pins if not used as ADC1 converter analog input or positive reference input.
V_{CC3} V_{SS3}		34 35		ADC1 positive-supply voltage and optional positive-reference input pin ADC1 ground reference pin
$\overline{\text{RESET}}$		53	I/O	System reset bidirectional pin. $\overline{\text{RESET}}$, as an input, initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit.

† I = input, O = output
‡ Ports A, B, C, and D can be configured only as general-purpose I/O pins. Also, port D3 can be configured as SYSCLK.
Note: The three–pin SCI configuration is referred to as SCI1.

Table 2–8. TMS370Cx6x Pin Descriptions (Continued)

Pin				I/O†	Description‡
Name	**Alternate Function**		**LCC (68)**		
XTAL2/CLKIN XTAL1			31 32	I O	Internal oscillator crystal input/external clock source input Internal oscillator output for crystal
MC			6	I	Mode control (MC) pin. MC enables EEPROM write-protection override (WPO) mode, and also supplies EPROM V_{PP}.
V_{CC1}			33,61		Positive supply voltage
V_{CC2}			15,63		Positive supply voltage for digital I/O logic
V_{SS1}			9		Ground reference for digital logic
V_{SS2}			16,62		Ground reference for digital I/O logic
	Function				Single-chip mode: Port D is a general-purpose bidirectional I/O port. Each of the port D pins can be individually configured as a general-purpose I/O pin, primary memory control signal (function A), or secondary memory control signal (function B). All chip selects are independent and can be used for memory bank switching.
	A	B			
D3	SYSCLK	SYSCLK	58	I/O	I/O pin/A, B: Internal clock signal is 1/1 (PLL) or 1/4 XTAL2/CLKIN frequency.
D4	R/$\overline{W}$	R/$\overline{W}$	57	I/O	I/O pin/A, B: Read/write output pin.
D5	$\overline{CSPF}$	—	56	I/O	I/O pin/A: Chip select peripheral output for peripheral file goes low during memory accesses. I/O pin/B: Reserved.
D6	$\overline{CSH1}$	$\overline{EDS}$	55	I/O	I/O pin/A: Chip select half output 1 goes low during memory accesses. I/O pin/B: External data strobe output goes low during memory accesses from external memory and has the same timings as the five chip selects.
D7	$\overline{CSE1}$	$\overline{WAIT}$	54	I/O	I/O pin/A: Chip select eighth output goes low during memory accesses. I/O pin/B: Wait input pin extends bus signals.
SCITXD	SCIIO1		30	I/O	SCI transmit data output pin/general-purpose bidirectional pin (see Note)
SCIRXD	SCIIO2		29	I/O	SCI receive data input pin/general-purpose bidirectional pin
SCICLK	SCIIO3		28	I/O	SCI bidirectional serial clock pin/general-purpose bidirectional pin
T1IC/CR	T1IO1		46	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	T1IO2		45	I/O	Timer 1 pulse width modulation (PWM) output pin/general-purpose bidirectional pin
T1EVT	T1IO3		44	I/O	Timer 1 external event input pin/general-purpose bidirectional pin

† I = input, O = output
‡ Ports A, B, C, and D can be configured only as general-purpose I/O pins. Also, port D3 can be configured as SYSCLK.
Note: The three–pin SCI configuration is referred to as SCI1.

Table 2–8. TMS370Cx6x Pin Descriptions (Continued)

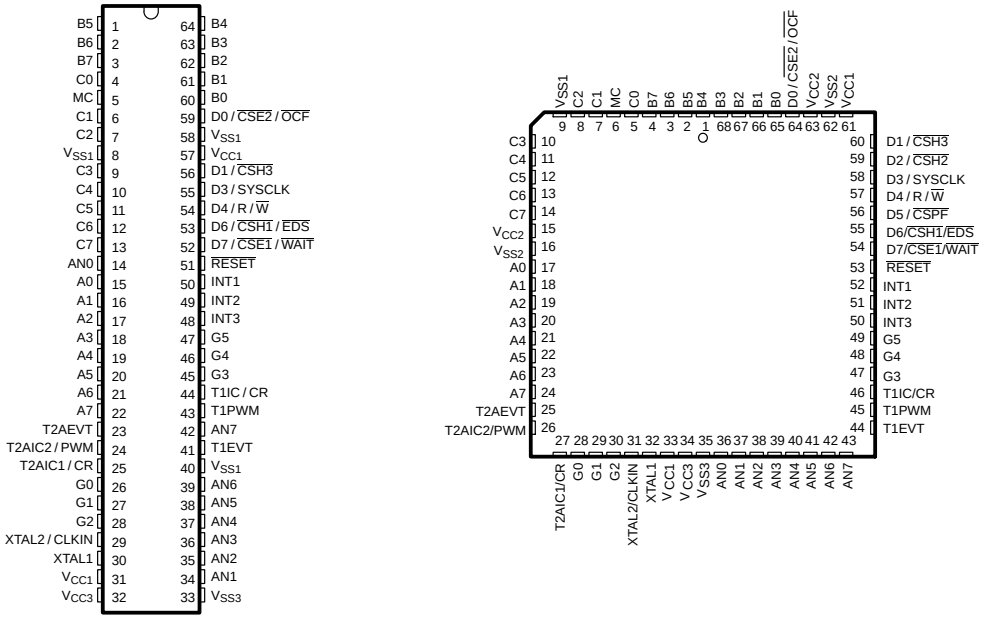
Pin			I/O†	Description‡
Name	**Alternate Function**	**LCC (68)**		
T2AIC1/CR	T2AIO1	27	I/O	Timer 2A input capture 1/counter-reset input pin/general-purpose bidirectional pin
T2AIC2/PWM	T2AIO2	26	I/O	Timer 2A input capture 2/PWM output pin/general-purpose bidirectional pin
T2AEVT	T2AIO3	25	I/O	Timer 2A external event input pin/general-purpose bidirectional pin
T2BIC1/CR	T2BIO1	64	I/O	Timer 2B input capture 1/counter reset input pin/general–purpose bidirectional pin
T2BIC2/PWM	T2BIO2	60	I/O	Timer 2B input capture 2/PWM output pin/general–purpose bidirectional pin
T2BEVT	T2BIO3	59	I/O	Timer 2B external event input pin/general–purpose bidirectional pin
SPISOMI	SPIIO1	49	I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin
SPISIMO	SPIIO2	48	I/O	SPI slave input pin, master output pin/general-purpose bidirectional pin
SPICLK	SPIIO3	47	I/O	SPI bidirectional serial clock pin/general-purpose bidirectional pin

† I = input, O = output
‡ Ports A, B, C, and D can be configured only as general-purpose I/O pins. Also, port D3 can be configured as SYSCLK.
Note: The three–pin SCI configuration is referred to as SCI1.

2.9 TMS370Cx7x Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx7x devices are shown in Figure 2–9 and Table 2–9, respectively.

Figure 2–9. Pinouts for TMS370Cx7x



A. 64-Pin PSDIP (NM)
B. 64-Pin CSDIP (FZ)

C. 68-Pin PLCC (FN)
D. 68-Pin CLCC (FZ)

Table 2–9. TMS370Cx7x Pin Descriptions

Pin				
Name	**DIP (64)**	**LCC (68)**	**I/O†**	**Description**
A0	15	17	I/O	Port A is a general-purpose bidirectional I/O port.
A1	16	18	I/O	
A2	17	19	I/O	
A3	18	20	I/O	
A4	19	21	I/O	
A5	20	22	I/O	
A6	21	23	I/O	
A7	22	24	I/O	
B0	60	65	I/O	Port B is a general-purpose bidirectional I/O port.
B1	61	66	I/O	
B2	62	67	I/O	
B3	63	68	I/O	
B4	64	1	I/O	
B5	1	2	I/O	
B6	2	3	I/O	
B7	3	4	I/O	
C0	4	5	I/O	Port C is a general-purpose bidirectional I/O port.
C1	6	7	I/O	
C2	7	8	I/O	
C3	9	10	I/O	
C4	10	11	I/O	
C5	11	12	I/O	
C6	12	13	I/O	
C7	13	14	I/O	
INT1/NMI	50	52	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
INT2	49	51	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	48	50	I/O	External maskable interrupt input/general-purpose bidirectional pin
AN0/E0	14	36	I	ADC1 analog input (AN0–AN7) or positive reference pins (AN1–AN7)
AN1/E1	34	37	I	
AN2/E2	35	38	I	
AN3/E3	36	39	I	Port E can be individually programmed as general-purpose input pins if not used as ADC1 analog input or positive reference input.
AN4/E4	37	40	I	
AN5/E5	38	41	I	
AN6/E6	39	42	I	
AN7/E7	42	43	I	
V_{CC3}	32	34		ADC1 positive-supply voltage and optional positive-reference input pin
V_{SS3}	33	35		ADC1 ground reference pin

† I = input, O = output

Table 2–9. TMS370Cx7x Pin Descriptions (Continued)

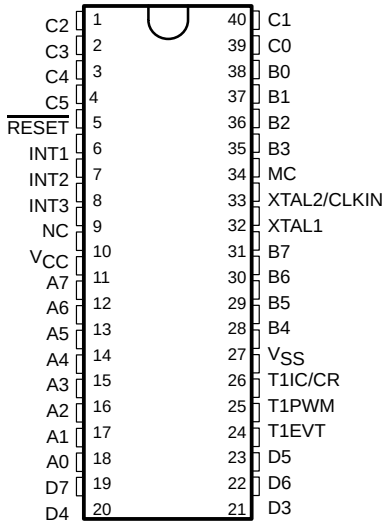
Pin				
Name	**DIP (64)**	**LCC (68)**	**I/O†**	**Description**
$\overline{\text{RESET}}$	51	53	I/O	System reset bidirectional pin. $\overline{\text{RESET}}$, as an input, initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit.
MC	5	6	I	Mode control (MC) pin. MC enables EEPROM write-protection override (WPO) mode, also supplies EPROM V_{PP}.
XTAL2/CLKIN	29	31	I	Internal oscillator crystal input/external clock source input
XTAL1	30	32	O	Internal oscillator output for crystal
V_{CC1}	31, 57	33, 61		Positive supply voltage
V_{CC2}	—	15,63		Positive supply voltage for digital I/O logic
V_{SS1}	8, 40, 58	9		Ground reference for digital logic
V_{SS2}		16,62		Ground reference for digital I/O logic
D0	59	64	I/O	Port D is a general-purpose bidirectional I/O port that can be individually configured. D3 is also configurable as SYSCLK.
D1	56	60	I/O	
D2	–	59	I/O	
D3	55	58	I/O	
D4	54	57	I/O	
D5	–	56	I/O	
D6	53	55	I/O	
D7	52	54	I/O	
G0	26	28	I/O	Port G is a general-purpose bidirectional port.
G1	27	29	I/O	
G2	28	30	I/O	
G3	45	47	I/O	
G4	46	48	I/O	
G5	47	49	I/O	
T1IC/CR	44	46	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	43	45	I/O	Timer 1 pulse-width-modulation (PWM) output pin/general-purpose bidirectional pin
T1EVT	41	44	I/O	Timer 1 external event input pin/general-purpose bidirectional pin
T2AIC1/CR	25	27	I/O	Timer 2A input capture 1/counter reset input pin/general-purpose bidirectional pin
T2AIC2/PWM	24	26	I/O	Timer 2A input capture 2/PWM output pin/general-purpose bidirectional pin
T2AEVT	23	25	I/O	Timer 2A external event input pin/general-purpose bidirectional pin

† I = input, O = output

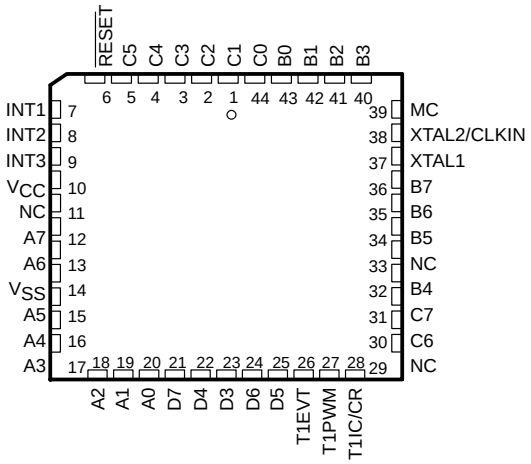
2.10 TMS370Cx8x Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx8x devices are shown in Figure 2–10 and Table 2–10, respectively.

Figure 2–10. Pinouts for TMS370Cx8x



A. 40–pin PDIP (N)



B. 44–Pin PLCC (FN)
C. 44–Pin CLCC (FZ)

Table 2–10. TMS370Cx8x Pin Descriptions

Pins			I/O†	Description
Name	**PDIP (40)**	**LCC (44)**		
A0	18	20	I/O	Port A is a general-purpose bidirectional I/O port.
A1	17	19	I/O	
A2	16	18	I/O	
A3	15	17	I/O	
A4	14	16	I/O	
A5	13	15	I/O	
A6	12	13	I/O	
A7	11	12	I/O	
B0	38	43	I/O	Port B is a general-purpose bidirectional I/O port.
B1	37	42	I/O	
B2	36	41	I/O	
B3	35	40	I/O	
B4	28	32	I/O	
B5	29	34	I/O	
B6	30	35	I/O	
B7	31	36	I/O	
C0	39	44	I/O	Port C is a general-purpose bidirectional I/O port.
C1	40	1	I/O	
C2	1	2	I/O	
C3	2	3	I/O	
C4	3	4	I/O	
C5	4	5	I/O	
C6	—	30	I/O	
C7	—	31	I/O	
D3	21	23	I/O	Port D is a general-purpose bidirectional I/O port. D3 is also configurable as SYSCLK.
D4	20	22	I/O	
D5	23	25	I/O	
D6	22	24	I/O	
D7	19	21	I/O	
INT1	6	7	I	External (non-maskable or maskable) interrupt/general-purpose input pin
INT2	7	8	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	8	9	I/O	External maskable interrupt input/general-purpose bidirectional pin
T1IC/CR	26	28	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	25	27	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin
T1EVT	24	26	I/O	Timer 1 external event input pin/general-purpose bidirectional pin
$\overline{\text{RESET}}$	5	6	I/O	System reset bidirectional pin: as input pin, $\overline{\text{RESET}}$ initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by the watchdog or oscillator fault circuit.
MC	34	39	I	Mode control pin; enables EEPROM write-protection override (WPO) mode, and also supplies EPROM V_{PP}
XTAL2/CLKIN	33	38	I	Internal oscillator crystal input/external clock source input
XTAL1	32	37	O	Internal oscillator output for crystal

† I = input, O = output

Table 2-10. TMS370Cx8x Pin Descriptions (Continued)

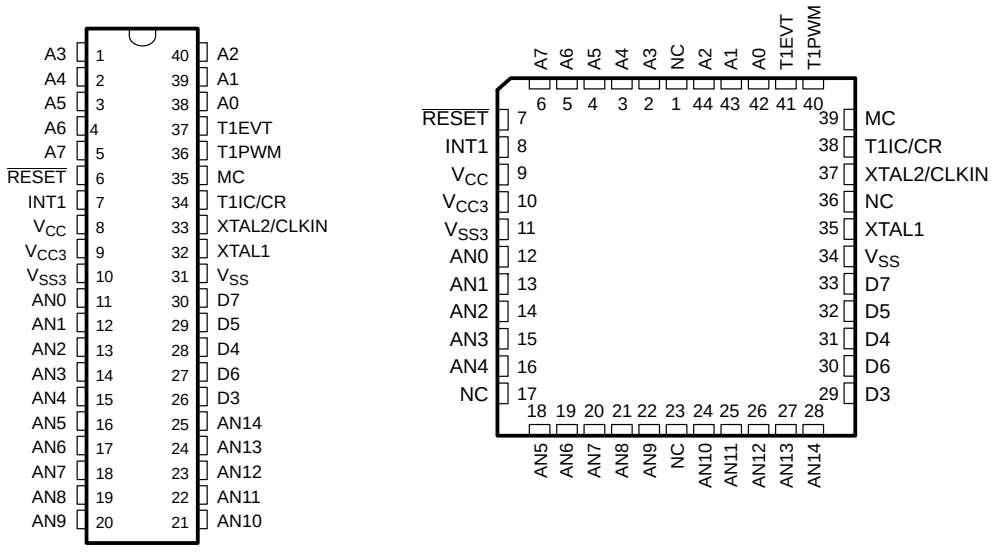
Pins			I/O†	Description
Name	**PDIP (40)**	**LCC (44)**		
V_{CC}	10	10		Positive supply voltage
V_{SS}	27	14		Ground reference for digital logic
NC	9	11, 29, 33		These pins have no connection to the internal die.

† I = input, O = output

2.11 TMS370Cx9x Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370Cx9x devices are shown in Figure 2–11 and Table 2–11, respectively.

Figure 2–11. Pinouts for TMS370Cx9x



A. 40-Pin PSDIP (NJ)†
B. 40-Pin CSDIP (JC)
C. 44-Pin PLCC (FN)
D. 44-Pin CLCC (FZ)

† The NJ designator for the 40–pin plastic shrink DIP package was formerly known as N2. The mechanical drawing of the NJ is identical to the N2 package and did not need to be requalified.

Table 2–11. TMS370Cx9x Pin Descriptions

Pin			I/O†	Description
Name	**SDIP (40)**	**LCC (44)**		
A0	38	42	I/O	Port A is a general-purpose bidirectional I/O port.
A1	39	43	I/O	
A2	40	44	I/O	
A3	1	2	I/O	
A4	2	3	I/O	
A5	3	4	I/O	
A6	4	5	I/O	
A7	5	6	I/O	
D3/SYSCLK	26	29	I/O	Port D is a general-purpose bidirectional I/O port. D3 is also configurable as SYSCLK.
D4	28	31	I/O	
D5	29	32	I/O	
D6	27	30	I/O	
D7	30	33	I/O	
AN0/E0	11	12	I	ADC3 analog input (AN0–AN7) or positive reference pins (AN6–AN7). Port E can be individually programmed as general-purpose input pins if not used as ADC3 analog input. Only AN6 and AN7 can be software-configured as positive reference input.
AN1/E1	12	13	I	
AN2/E2	13	14	I	
AN3/E3	14	15	I	
AN4/E4	15	16	I	
AN5/E5	16	18	I	
AN6/E6	17	19	I	
AN7/E7	18	20	I	
AN8	19	21	I	ADC3 analog input pins.
AN9	20	22	I	
AN10	21	24	I	
AN11	22	25	I	
AN12	23	26	I	
AN13	24	27	I	
AN14	25	28	I	
INT1	7	8	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
T1IC/CR	34	38	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin.
T1PWM	36	40	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin.
T1EVT	37	41	I/O	Timer 1 external event input pin/general-purpose bidirectional pin.
$\overline{\text{RESET}}$	6	7	I/O	System-reset bidirectional pin. $\overline{\text{RESET}}$, as an input, initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit.
MC	35	39	I	Mode control (MC) pin. MC enables EEPROM write-protection override (WPO) mode, and also supplies EPROM V_{PP}.
XTAL2/CLKIN	33	37	I	Internal oscillator crystal input / external clock source input
XTAL1	32	35	O	Internal oscillator output for crystal
V_{CC}	8	9		Positive supply voltage for digital logic.
V_{SS}	31	34		Ground reference for digital logic.

† I = input, O = output

Table 2–11. TMS370Cx9x Pin Descriptions (Continued)

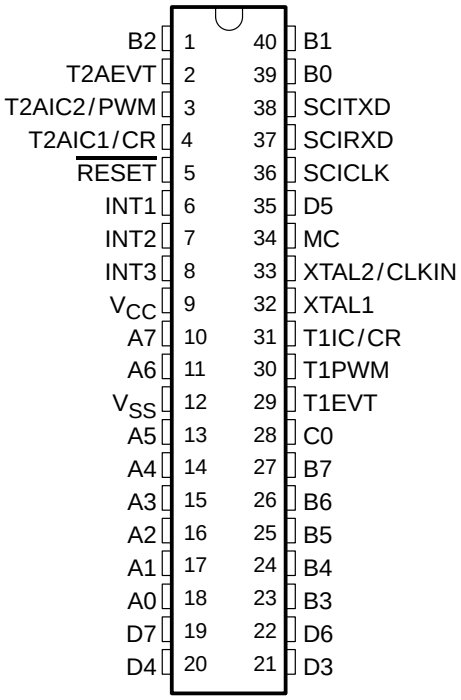
Pin			I/O†	Description
Name	**SDIP (40)**	**LCC (44)**		
V_{CC3}	9	10		Positive supply voltage for ADC3
V_{SS3}	10	11		Ground reference for ADC3
NC	—	1, 17, 23, 36		These pins have no connection to the internal die.

† I = input, O = output

2.12 TMS370CxAx Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370CxAx devices are shown in Figure 2–12 and Table 2–12, respectively.

Figure 2–12. Pinouts for TMS370CxAx



A. 40–Pin PDIP (N)

Table 2–12. TMS370CxAx Pin Descriptions

40–Pin PDIP		I/O†	Description
Name	**No.**		
A0	18	I/O	Port A is a general-purpose bidirectional I/O port.
A1	17	I/O	
A2	16	I/O	
A3	15	I/O	
A4	14	I/O	
A5	13	I/O	
A6	11	I/O	
A7	10	I/O	
B0	39	I/O	Port B is a general-purpose bidirectional I/O port.
B1	40	I/O	
B2	1	I/O	
B3	23	I/O	
B4	24	I/O	
B5	25	I/O	
B6	26	I/O	
B7	27	I/O	
C0	28	I/O	Port C is a general-purpose bidirectional I/O port.
D3	21	I/O	Port D is a general-purpose bidirectional I/O port. D3 is also configurable as SYSCLK.
D4	20	I/O	
D5	35	I/O	
D6	22	I/O	
D7	19	I/O	
INT1	6	I	External (non-maskable or maskable) interrupt/general-purpose input pin.
INT2	7	I/O	External maskable interrupt input/general-purpose bidirectional pin.
INT3	8	I/O	External maskable interrupt input/general-purpose bidirectional pin.
T1IC/CR	31	I/O	Timer 1 input capture/counter reset input pin /general-purpose bidirectional pin.
T1PWM	30		Timer 1 PWM output pin/general-purpose bidirectional pin.
T1EVT	29		Timer 1 external event input pin/general-purpose bidirectional pin.
SCITXD	38	I/O	SCI transmit data output pin, general-purpose bidirectional pin (see Note).
SCIRXD	37		SCI receive data input pin/general-purpose bidirectional pin.
SCICLK	36		SCI bidirectional serial clock pin/general-purpose bidirectional pin.
$\overline{\text{RESET}}$	5	I/O	System reset bidirectional pin; as input pin, $\overline{\text{RESET}}$ initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by watchdog or oscillator fault circuit.
T2AIC1/CR	4	I/O	Timer 2A input capture 1/counter reset input pin/general-purpose bidirectional pin.
T2AIC2/PWM	3		Timer 2A input capture 2/PWM output pin/general-purpose bidirectional pin.
T2AEVT	2		Timer 2A external event input pin/general-purpose bidirectional pin.
MC	34	I	Mode control input pin.
XTAL2/CLKIN	33	I	Internal oscillator crystal input/external clock source input.
XTAL1	32	O	Internal oscillator output for crystal.
V_{CC}	9		Positive supply voltage
V_{SS}	12		Ground reference

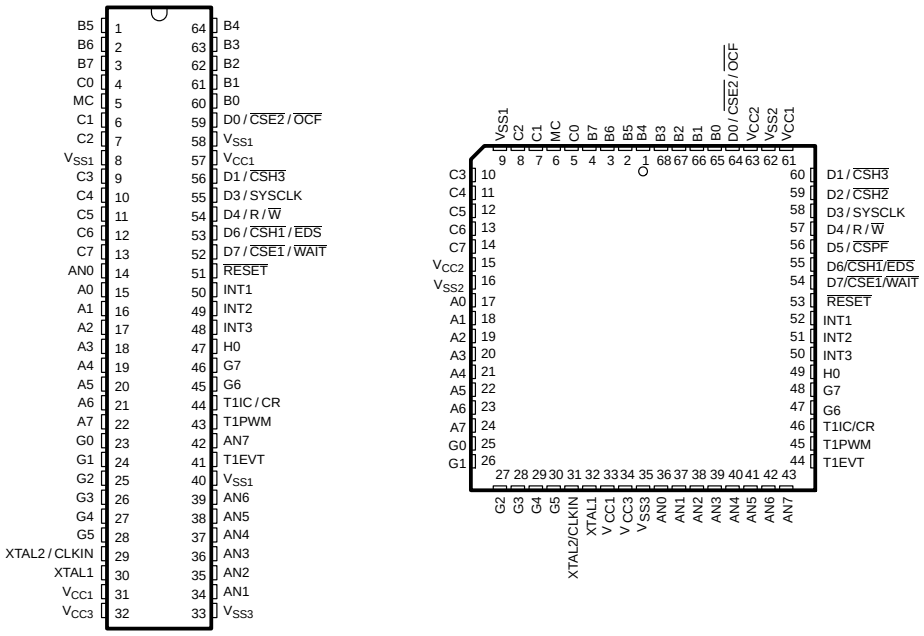
† I = input, O = output

Note: The two SCI configuration pins are referred to as SCI2.The three–pin SCI configuration is referred to as SCI1.

2.13 TMS370CxBx Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370CxBx devices are shown in Figure 2–13 and Table 2–13, respectively.

Figure 2–13. Pinouts for TMS370CxBx



A. 64–Pin PSDIP (NM)

B. 68–Pin PLCC (FN)

Table 2–13. TMS370CxBx Pin Descriptions

Pin			I/O†	Description
Name	PSDIP (64)	LCC (68)		
A0	15	17	I/O	Port A is a general-purpose bidirectional I/O port.
A1	16	18	I/O	
A2	17	19	I/O	
A3	18	20	I/O	
A4	19	21	I/O	
A5	20	22	I/O	
A6	21	23	I/O	
A7	22	24	I/O	
B0	60	65	I/O	Port B is a general-purpose bidirectional I/O port.
B1	61	66	I/O	
B2	62	67	I/O	
B3	63	68	I/O	
B4	64	1	I/O	
B5	1	2	I/O	
B6	2	3	I/O	
B7	3	4	I/O	
C0	4	5	I/O	Port C is a general-purpose bidirectional I/O port.
C1	6	7	I/O	
C2	7	8	I/O	
C3	9	10	I/O	
C4	10	11	I/O	
C5	11	12	I/O	
C6	12	13	I/O	
C7	13	14	I/O	
INT1/NMI	50	52	I	External (nonmaskable or maskable) interrupt/general-purpose input pin
INT2	49	51	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	48	50	I/O	External maskable interrupt input/general-purpose bidirectional pin
AN0/E0	14	36	I	ADC1 analog input (AN0 – AN7) or positive reference pins (AN1 – AN7)
AN1/E1	34	37	I	
AN2/E2	35	38	I	Port E can be individually programmed as general-purpose input pins
AN3/E3	36	39	I	if not used as ADC1 analog input or positive reference input.
AN4/E4	37	40	I	
AN5/E5	38	41	I	
AN6/E6	39	42	I	
AN7/E7	42	43	I	
V_{CC3}	32	34		ADC1 positive-supply voltage and optional positive-reference input pin
V_{SS3}	33	35		ADC1 ground reference pin
$\overline{\text{RESET}}$	51	53	I/O	System reset bidirectional pin. $\overline{\text{RESET}}$, as an input, initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates an internal failure was detected by the watchdog or oscillator fault circuit.

† I = input, O = output

Table 2–13. TMS370CxBx Pin Descriptions (Continued)

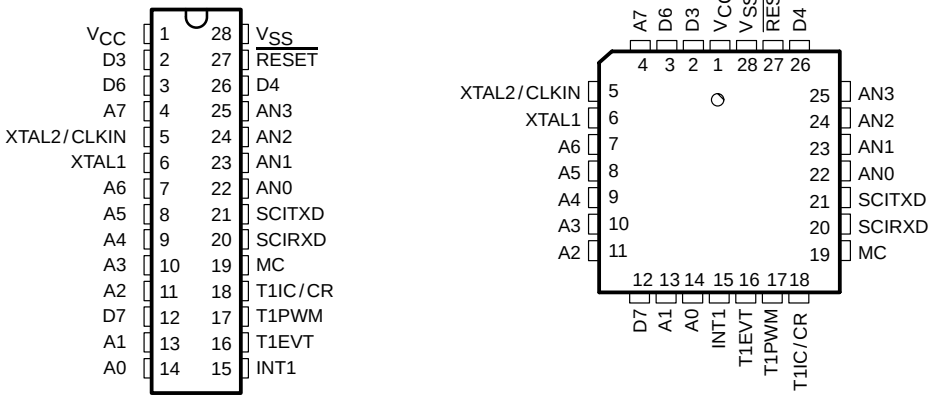
Pin			I/O†	Description
Name	**PSDIP (64)**	**LCC (68)**		
MC	5	6	I	Mode control (MC) pin. MC enables EEPROM write-protection override (WPO) mode.
XTAL2/CLKIN XTAL1	29 30	31 32	I O	Internal oscillator crystal input / external clock source input Internal oscillator output for crystal
V_{CC1}	31, 57	33,61		Positive supply voltage
V_{CC2}	—	15,63		Positive supply voltage for digital I/O logic
V_{SS1}	8, 58, 40	9		Ground reference for digital logic
V_{SS2}	—	16,62		Ground reference for digital I/O logic
D0 D1 D2 D3 D4 D5 D6 D7	59 56 – 55 54 – 53 52	64 60 59 58 57 56 55 54	I/O I/O I/O I/O I/O I/O I/O I/O	Port D is a general-purpose bidirectional I / O port. D3 is also configurable as SYSCLK.
G0 G1 G2 G3 G4 G5 G6 G7	23 24 25 26 27 28 45 46	25 26 27 28 29 30 47 48	I/O I/O I/O I/O I/O I/O I/O I/O	Port G is a general-purpose bidirectional port.
H0	47	49	I/O	Port H is a general-purpose bidirectional port.
T1IC / CR	44	46	I/O	I/O timer 1 input capture / counter reset input pin / general-purpose bidirectional pin
T1PWM	43	45	I/O	I/O timer 1 pulse-width-modulation (PWM) output pin / general-purpose bidirectional pin
T1EVT	41	44	I/O	I/O timer 1 external event input pin / general-purpose bidirectional pin

† I = input, O = output

2.14 TMS370CxCx Pinouts and Pin Descriptions

The pinout and pin descriptions for the TMS370CxCx devices are shown in Figure 2–14 and Table 2–14, respectively.

Figure 2–14. Pinouts for TMS370CxCx



A. 28–Pin PDIP (N)
B. 28–Pin CDIP (JD)
C. 28–Pin PLCC (FN)
D. 28–Pin CLCC (FZ)

Table 2–14. TMS370CxC Pin Descriptions

28-Pin DIP/LCC		I/O†	Description
Name	**No.**		
A0	14	I/O	Port A is a general-purpose bidirectional I/O port.
A1	13	I/O	
A2	11	I/O	
A3	10	I/O	
A4	9	I/O	
A5	8	I/O	
A6	7	I/O	
A7	4	I/O	
D3	2	I/O	Port D is a general-purpose bidirectional I/O port. D3 is also configurable as SYSCLK.
D4	26	I/O	
D6	3	I/O	
D7	12	I/O	
INT1	15	I	External (non-maskable or maskable) interrupt/general-purpose input pin.
AN0/E0	22	I	ADC2 module analog input (AN0–AN3) or positive reference pins (AN1–AN3).
AN1/E1	23	I	
AN2/E2	24	I	Port E can be individually programmed as general-purpose input pins if not used as
AN3/E3	25	I	ADC2 analog input.
T1IC/CR	18	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin.
T1PWM	17	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin.
T1EVT	16	I/O	Timer 1 external event input pin/general-purpose bidirectional pin.
SCITXD	21	I/O	SCI module transmit data output/general-purpose bidirectional pin. (See Note)
SCIRXD	20	I/O	SCI module receive data input pin/general-purpose bidirectional pin.
$\overline{\text{RESET}}$	27	I/O	System reset bidirectional pin; as input pin, $\overline{\text{RESET}}$ initializes the microcontroller; as open-drain output, $\overline{\text{RESET}}$ indicates that an internal failure was detected by watchdog or oscillator fault circuit.
MC	19	I	Mode control input pin; and also supplies EPROM V_{PP}.
XTAL2/CLKIN	5	I	Internal oscillator crystal input/External clock source input.
XTAL1	6	O	Internal oscillator output for crystal.
V_{CC}	1		Positive supply voltage
V_{SS}	28		Ground reference

† I = input, O = output

Note: The two SCI configuration pins are referred to as SCI2.

Chapter 3

CPU and Memory Organization

This chapter describes the CPU registers and memory organization. In the TMS370 register-to-register architecture, the CPU and up to the first 256 bytes of RAM act as a single unit with the program counter, stack pointer, and status register.

This chapter covers the following topics:

Topic **Page**

3.1 CPU/Register File Interaction

The TMS370 architecture provides the following components:

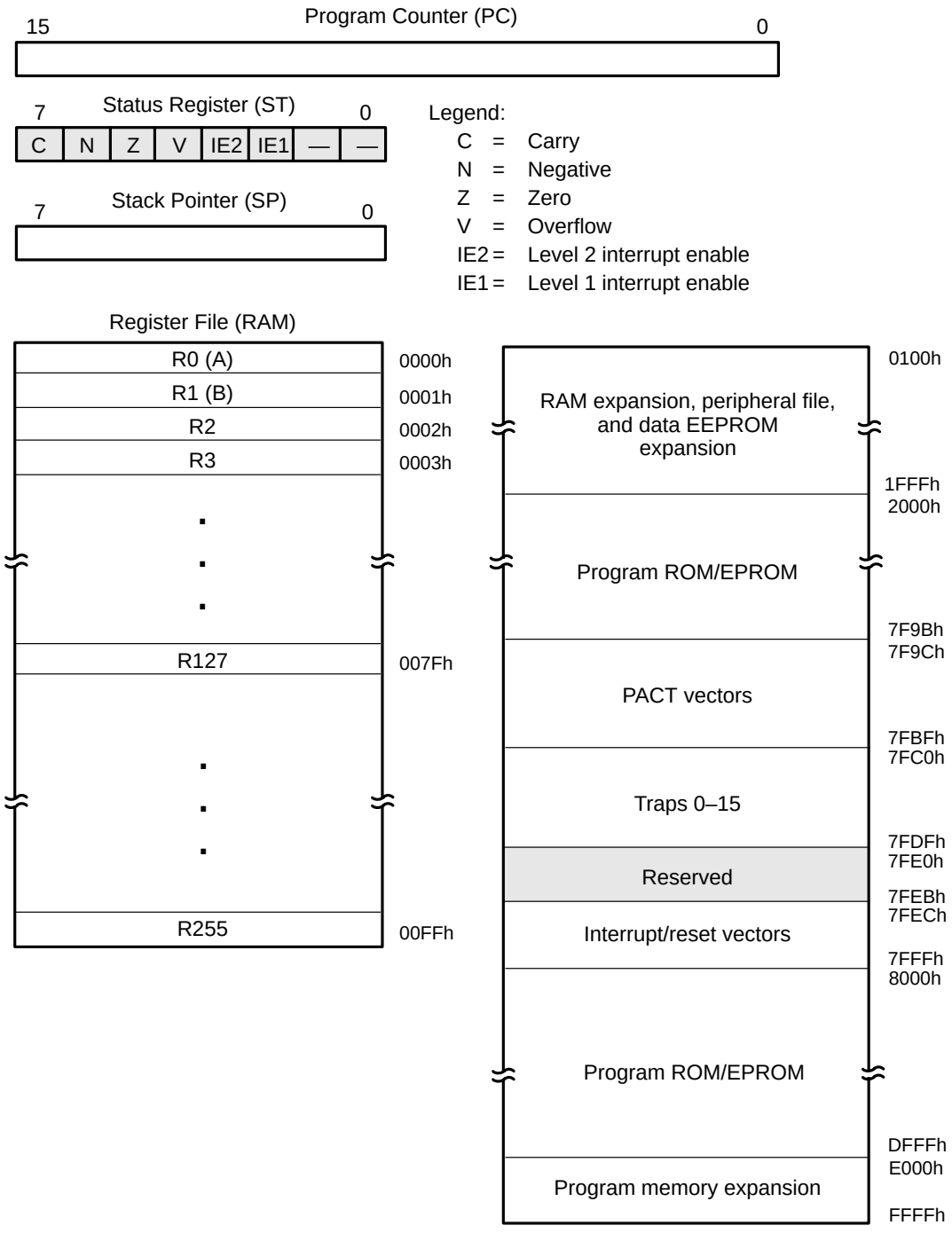
- CPU registers
  - A **stack pointer**, which points to the last entry in the memory stack
  - A **status register**, which monitors the operation of the instructions and contains the global interrupt bits
  - A **program counter**, which points to the memory location of the next instruction to be executed
- Memory map
  - A **register file** that can be accessed as general-purpose registers, data memory storage, program instructions, or part of the stack
  - A **peripheral file** that provides access to all internal peripheral modules, system-wide control functions, and EEPROM/EPROM programming control
  - **Data EEPROM modules**, which provide in-circuit programmability and data retention in power-off mode
  - **Program memory** that provides alternatives to meet the needs of your application

Figure 3–1 illustrates the CPU registers and memory blocks.

Note: Device Block Diagrams

Section 1.7 on page 1-21 contains a collection of block diagrams of each device family. These diagrams show each element of the TMS370 architecture (for example, TMS370Cx4x in Figure 1–6 on page 1-26.)

Figure 3–1. Programmer's Model



3.2 CPU Registers

The CPU contains three registers to control the status and direction of the program. These are the stack pointer, status register, and program counter. These registers and their use are described in the following subsections.

3.2.1 Stack Pointer (SP)

The stack operates as a last-in, first-out, read/write memory. The stack is typically used to store the return address on subroutine calls and the status register contents during interrupts.

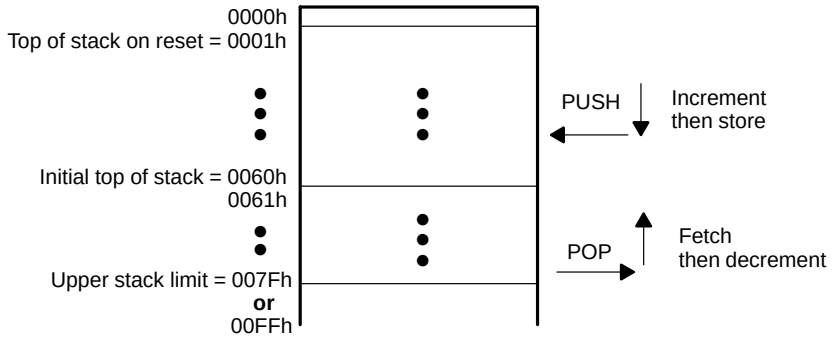
The stack pointer (SP) is an 8-bit CPU register that points to the last entry or top of the stack. The SP is automatically incremented *before* data is pushed onto the stack and decremented *after* data is popped from the stack.

The stack can be placed anywhere in the register file. During reset, the SP is loaded with a value of 01h. To control the area occupied by the stack, the application program must set the stack pointer and include code to monitor the stack size.

The SP is loaded from register B (R1) by the assembly language instruction LDSP (load stack pointer). The LDSP instruction allows the stack to be located anywhere in the register file space. The SP can be read into register B by the STSP (store stack pointer) command. Figure 3–2 illustrates an example SP initialization and stack operation.

```
INIT   MOV #60h,B          ;Load register B with the value
                           ;60h.
       LDSP                ;Load the stack pointer with the
                           ;contents of register B.
```

Figure 3–2. Stack Example



For devices with 256 (or more) bytes of RAM, if the stack is pushed beyond its limit of 00FFh, the SP register wraps around from 00FFh to 0000h without an error indication. For devices with only 128 bytes of RAM, the stack is not implemented beyond 007Fh; data pushed beyond this limit is lost. Your application program must guard against a stack overflow.

3.2.2 Status Register (ST)

The status register (ST) monitors the operation of the instructions and contains the global interrupt enable bits.

The ST includes four status bits (condition flags – C, N, Z, and V), two interrupt enable bits (IE2 and IE1), and two reserved bits.

The four status bits indicate the outcome of the previous instruction; conditional instructions (for example, the conditional jump instructions) use these status bits to determine the program flow.

Status Register (ST)

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
	C	N	Z	V	IE2	IE1	—	—
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0		

R = Read, W = Write, *-n* = Value of the bit after the register is reset

The ST, status bit notation, and status bit definitions are as follows:

Bit 7 **C**. Carry.

This status bit is set by arithmetic instructions as a carry bit or as a no-borrow bit. It is also affected by the rotate instructions. See each instruction in Chapter 16 for a detailed description of how the carry bit is used.

Bit 6 **N**. Negative.

The CPU sets this bit to the value of the most significant bit (sign bit) of the result of the previous operation.

Bit 5 **Z**. Zero.

This bit is set by the CPU if the result of the previous operation was 0; cleared otherwise.

Bit 4 **V**. Overflow.

This bit is set by the CPU if a signed arithmetic overflow condition is detected during the previous instruction. The value of this flag is significant at the completion of the following instructions: ADC, ADD, INC, INCW, CMP, DEC, SUB, SBB, and DIV.

Instruction	V (Bit 4) Equals a 1 If
ADC, ADD, INC, INCW	(C XOR N) AND (bit 7{s} XNOR bit 7{d})
CMP, DEC, SUB, SBB	(C XOR N) AND (bit 7{s} XOR bit 7{d})
DIV (Rs, A)	Rs ≤ A, which means quotient > 255

Bit 3 **IE2**. Level 2 Interrupt Enable, (daisy-chain order).

This bit controls interrupt level 2 (lowest priority).

0 = Disables interrupt requests from priority level 2.
1 = Enables interrupt requests from priority level 2.

Bit 2 **IE1**. Level 1 Interrupt Enable.

This bit controls interrupt level 1 (highest priority).

0 = Disables interrupt requests from priority level 1.
1 = Enables interrupt requests from priority level 1.

Bits 1–0 **Reserved.** Read data is indeterminate.

When the CPU acknowledges an interrupt, the contents of the ST are automatically pushed onto the stack; then the ST is cleared (for more information about interrupt effects on the ST, see subsection 5.1.1, page 5-3). The RTI instruction implements a normal exit from an interrupt service routine. When the CPU executes the RTI instruction, it automatically restores the contents of the ST with a stack-pop operation.

The four condition flags (C, N, Z, and V) are updated every time an instruction is executed that manipulates or moves data. As a result, conditional branches should be performed immediately after data is manipulated. The instructions that *do not* affect the contents of these flags are as follows:

TRAP 0 to TRAP 15	DJNZ	IDLE	STSP
CALL	JMP	NOP	JMPL
CALLR	Jcnd (jump on condition instructions)	PUSH ST	LDSP
BR		RTS	

The LDST instruction allows a program to change all bits in the status register. The byte following this instruction is loaded directly into the status register. The

assembly language instructions DINT, EINT, EINTH, and EINTL enable specific interrupts. These instructions are converted to an LDST #iop8 opcode by the assembler so that #iop8 is the appropriate value to set or clear the specific interrupt (see Chapter 16 for more information on the LDST instruction).

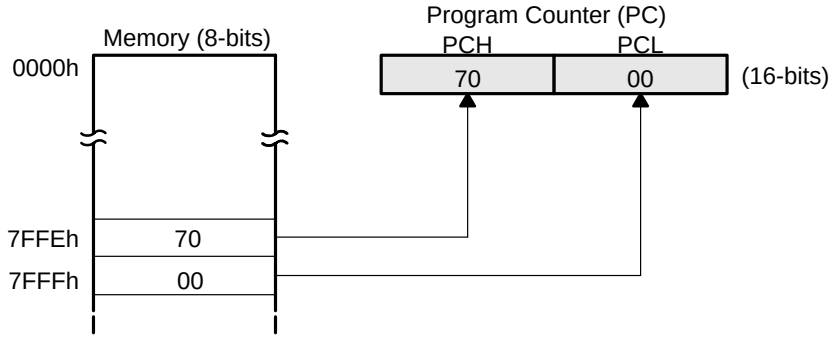
The carry (C) bit can be set with the SETC opcode and cleared with the CLRC opcode.

3.2.3 Program Counter (PC)

The contents of the program counter (PC) point to the memory location of the next instruction to be executed. The PC consists of two 8-bit registers in the CPU: the program counter high (PCH) and program counter low (PCL). These registers contain the MSbyte and LSbyte of a 16-bit address.

During reset, the PCH (PC's MSbyte) is loaded with the contents of memory location 7FFEh, and the PCL (PC's LSbyte) is loaded with the contents of memory location 7FFFh. Figure 3–3 illustrates this operation using an example value of 7000h as the contents of memory locations 7FFEh and 7FFFh (reset vector).

Figure 3–3. Program Counter After Reset



3.3 Memory Map

Figure 3–4 shows the memory map of the TMS370 family members. The partitioning of memory and physical location of memory (that is, the on- or off-chip) depends on the device used and the memory mode (μC or μP — microcontroller or microprocessor, discussed in Section 3.4, page 3-16).

Each device that has memory expansion can be programmed up to sixteen address bits. This allows access to up to 56K bytes of memory. In addition, memory expansion features allow using up to 112K bytes of external memory. (The expansion features are described further in subsection 3.4.2 page 3-19.)

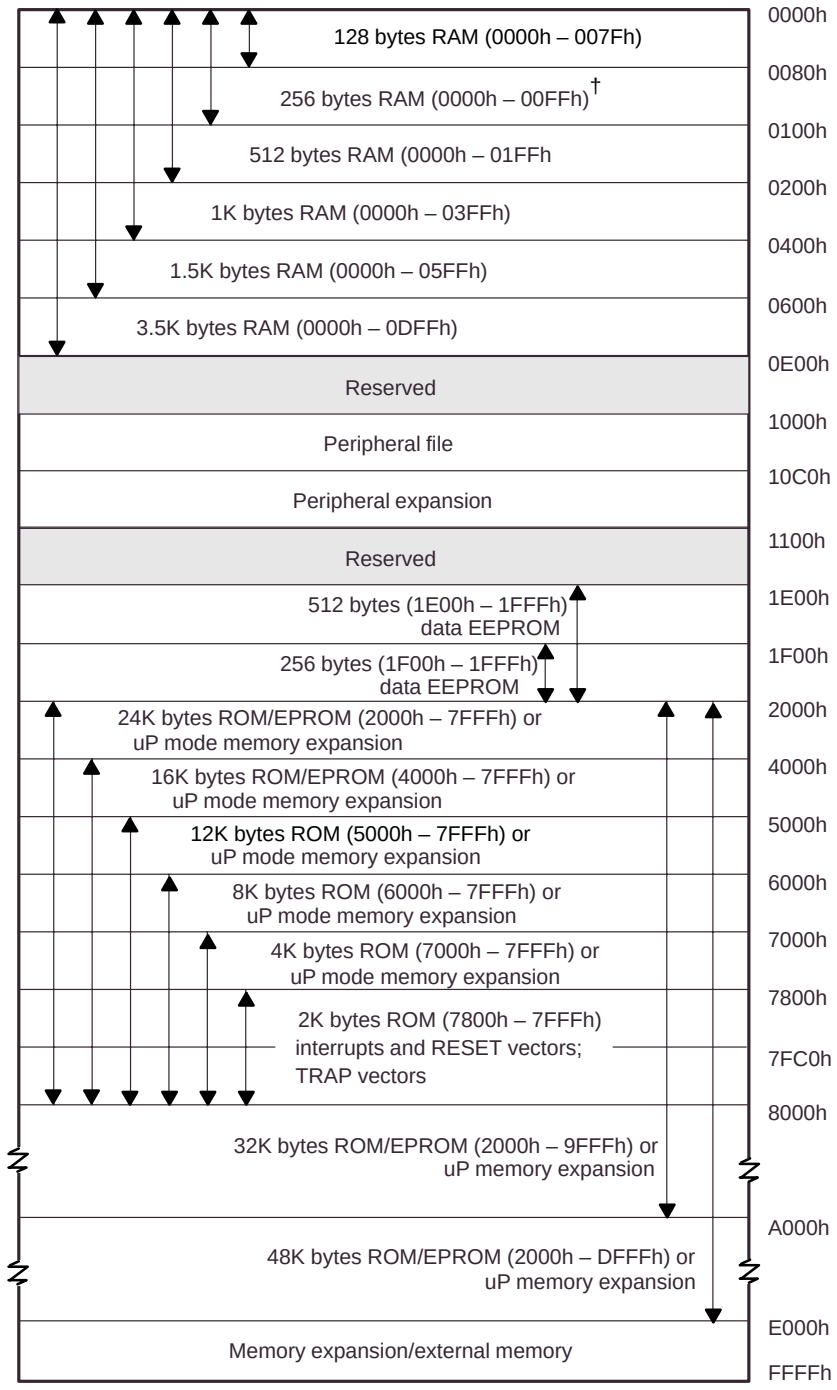
3.3.1 Register File

The register file (RF) is located in on-chip RAM at memory addresses 0000h–00FFh.

- In devices with 128 bytes of RAM, the RF has 128 memory locations treated as registers R0 through R127.
- In devices with 256 bytes of RAM, the RF has 256 memory locations treated as registers R0 through R255.
- If the device incorporates the PACT module:
  - 128 bytes of dual-port RAM are mapped into memory locations 0080h–00FFh for ’x32 devices or 0180–01FFh for ’x36 devices. For the ’x32 device, any of this RAM not used by the PACT module can be used as registers or stack. For the ’x36 device, any of this RAM not used by the PACT can be used only as data memory storage.
  - 256 bytes of standby RAM (only available in ’x36 devices) are mapped into memory locations 0200h–02FFh. The standby RAM is used as general-purpose memory, and cannot be used as RF or stack. The standby RAM is powered by pin V_{CCSTBY}; therefore, data is saved if the main V_{CC1} power fails.

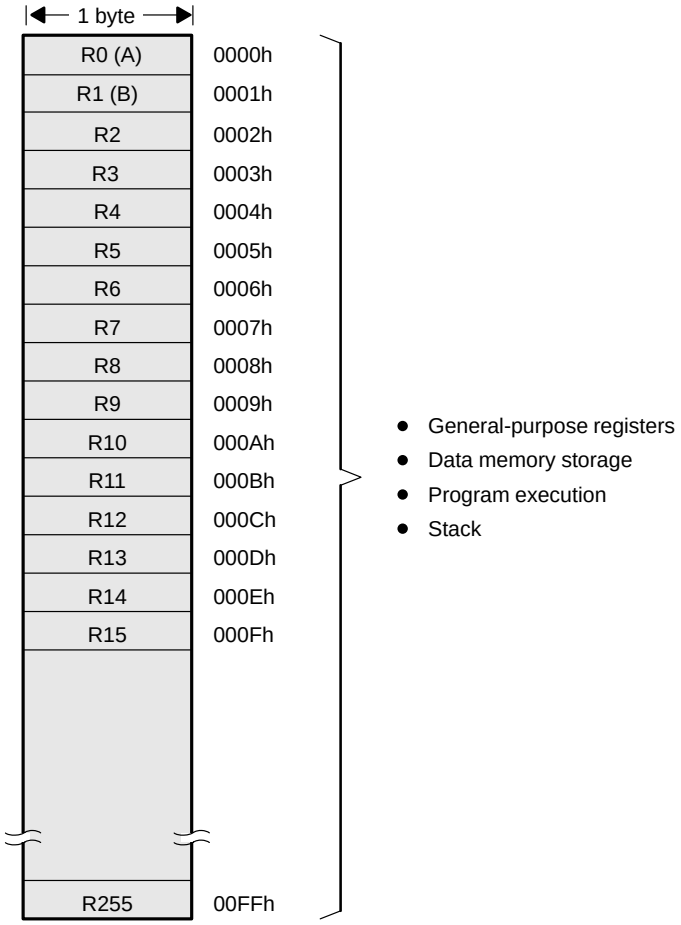
The memory addresses of the registers in the RF are shown in Figure 3–5 on page 3-10.

Figure 3–4. TMS370 Memory Map



† In devices with more than 256 bytes of RAM, only the first 256–byte block can be used as registers or stack.

Figure 3–5. Register File Addresses



The first two registers, R0 and R1, are also called registers A and B, respectively. Some instructions imply the use of register A or B. For example, the LDSP instruction assumes that the value to be loaded into the stack pointer is contained in register B.

The partitioning of the RF is determined by the value loaded into the stack pointer and by the program's use of the RF. Locations within the RF address space can serve as either the CPU register file or general-purpose read/write memory. Program instructions can reside in and be executed from any location in the address space without restriction. The stack also occupies a portion of the RF. The multiple uses of the RF give you the flexibility to use the RF however you wish.

Any location in the RF can be accessed in one of three ways:

- Register access using the register number as shown in the following example code:

```
MOV    A,R6        ;Move the contents of Register A to
                   ;Register R6.
MOV    R12,R200    ;Move the contents of Register 12 to
                   ;Register R200.
```

- Stack access using the stack pointer as shown in the following example code:

```
MOV    #5,B        ;Move the value 5 into Register B.
LDSP               ;Move the contents of Register B to
                   ;the stack pointer.
PUSH   A           ;Increment stack pointer to 6.
                   ;Move contents of Register A to 0006h.
```

- Normal memory access using 16-bit addresses as shown in the following example code:

```
MOV    A,&0006     ;Move the contents of Register A to
                   ;memory location 0006h.
```

When working with the RF, you must keep the following in mind:

- **Access time.** When the RF is used as a general-purpose register, the access time is a single-system clock cycle. Access to the RF for any other purpose takes two clock cycles.
- **Reset operations.** A reset operation has no effect on the contents of any memory location within the RF except for locations 0000h (register A) and 0001h (register B). Registers A and B are cleared in the beginning of the reset process.
- **Halt, idle, and standby states.** The halt, idle, and standby states have no effect on the contents of the RF or RAM.
- **RAM outside of the RF.** RAM that is not within the first 256 bytes (0000h–00FFh) is general-purpose RAM and is not considered part of the RF. Access to this RAM takes two clock cycles.

3.3.2 Peripheral File

The peripheral file (PF) is a set of memory-mapped registers that provide access to all internal peripheral modules, system-wide control functions, and EEPROM/EPROM programming control.

The PF includes 256 addresses in the memory map from 1000h–10FFh. The PF is divided into sixteen frames of 16 bytes each. Each peripheral frame is

allocated its own set of control registers. In addition, some frames are dedicated to specific functions.

The instruction set includes some instructions which access the peripheral file directly. These instructions designate the register by the number of the file register relative to 1000h, preceded by P0 for a hexadecimal designator or P for a decimal designator. For example, the system configuration control register 0 is located at address 1010h; its peripheral file hexadecimal designator is P010, and its decimal designator is P16.

Table 3–1 shows the address map of the peripheral file for the different device families. The following describe frames 0 – 15 of the file:

- Frame 0 of the peripheral file (memory addresses 1000h–100Fh) is reserved for factory testing. The results of access to this frame are unpredictable.
- Frame 1 (1010h–101Fh) contains system configuration and control functions. It also contains registers for controlling EEPROM/EPROM programming. EEPROM/EPROM module control registers are described in Chapter 6, *EPROM and EEPROM Modules*.
- Frame 2 (1020h–102Fh) contains the digital I/O pin configuration and control registers. The individual functions controlled by these registers are described in Section 4.4, page 4-18.
- Frames 3 through 8 are used by the internal peripherals. These peripherals and their control registers are described in the following chapters:
  - Timer 1 registers — Chapter 7
  - Timer 2A registers — Chapter 8
  - Timer 2B registers — Chapter 8
  - SCI1 registers — Chapter 9
  - SCI2 registers — Chapter 10
  - SPI registers — Chapter 11
  - ADC1 registers — Chapter 12
  - ADC2 registers — Chapter 13
  - ADC3 registers — Chapter 14
  - PACT registers — Chapter 15
- Frames 9 through 11 are reserved.
- Frames 12 through 15 are available for the external expansion of the peripheral file on devices that have a memory expansion capability. These frames are located in external memory and accessed by the external address and data buses.

Table 3–1. Peripheral File Address Map

Frame No.	Address	Description	TMS370 Device Families													
			x0x	x1x	x2x	x32	x36	x4x	x5x	x6x	x7x	x8x	x9x	xAx	xBx	xCx
0	1000h	Reserved for Factory Test	—	—	—	—	—	—	—	—	—	—	—	—	—	—
1	1010h	System and EEPROM/EPROM Control Registers	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	NA	Yes	Yes
2	1020h	Digital I/O Port Control Registers	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3	1030h	SPI Registers	NA	Yes	Yes	NA	Yes	NA	Yes	Yes	NA	NA	NA	NA	NA	NA
4	1040h	Timer 1 Registers	Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
		PACT Registers	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	1050h	SCI Registers†	Yes	NA	Yes	NA	NA	Yes	Yes	Yes	NA	NA	NA	Yes	NA	Yes
6	1060h	Timer 2A Registers	NA	NA	NA	NA	NA	Yes	Yes	Yes	Yes	NA	NA	Yes	NA	NA
7	1070h	ADC Registers‡	NA	NA	NA	Yes	Yes	Yes	Yes	Yes	Yes	NA	Yes	NA	Yes	Yes
8	1080h	Timer 2B registers	NA	NA	NA	NA	NA	NA	NA	Yes	NA	NA	NA	NA	NA	NA
9	1090h	Reserved	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	10A0h	Reserved	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	10B0h	Reserved	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	10C0h	External Peripheral Control	NA	NA	NA	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA
13	10D0h	External Peripheral Control	NA	NA	NA	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA
14	10E0h	External Peripheral Control	NA	NA	NA	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA
15	10F0h	External Peripheral Control	NA	NA	NA	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA

NA = Not Available
† SCI refers to SCI1 and SCI2
‡ ADC refers to ADC1, ADC2, and ADC3

3.3.3 Data EEPROM Modules

Data EEPROM modules are provided on all TMS370 family devices, except TMS370CxAx and TMS370CxCx device families.

The data EEPROM modules are 256- and 512-byte arrays. The 256-byte array is located at memory addresses 1F00h through 1FFFh, with the WPR (write protection register) at 1F00h. The 512-byte array is located at memory addresses 1E00h through 1FFFh, with WPRs at 1E00h and 1F00h. Larger arrays will continue to grow toward the smaller memory addresses with WPRs located in the first byte of every 256-byte boundary.

Each set of 256 bytes is configured into eight blocks of 32 bytes and has an associated WPR. Each block can be individually write protected by setting the

appropriate bit in the WPR. This module can be programmed on the basis of an entire array, a byte, or a single bit. The read-access time for the EEPROM module is two system clock cycles.

Programming of the data EEPROM array is controlled by the data EEPROM control register (DEECTL) at memory address 101Ah (P01A) and the corresponding WPRs. See subsections 6.2.1 (page 6-3) and 6.2.2 (page 6-4) for more details on the WPR and DEECTL registers.

3.3.4 Program Memory

The program memory options available for the TMS370 family allow a wide selection of memory types: ROM or EPROM, ranging in size from 2K to 48K bytes. The program memory is arranged as individually addressable bytes in the memory map. Data can be read or code can be executed directly from these locations.

Memory addresses 7F9Ch through 7FBFh and 7FECh through 7FFFh are reserved for interrupt and reset vectors. Trap vectors, used with TRAP0 through TRAP15 instructions, are at addresses 7FC0h through 7FDFh. Table 3–2 gives the memory map for the reserved vector locations and describes the differences among the TMS370 family members.

Table 3–2. Vector Address Map

Memory Address	Description	Available on TMS370 Family Members														No. Bytes
		x0x	x1x	x2x	x32	x36	x4x	x5x	x6x	x7x	x8x	x9x	xAx	xBx	xCx	
7F9Ch	PACT INT 1–18	NA	NA	NA	Yes	Yes	NA	NA	NA	NA	NA	NA	NA	NA	NA	36
7FBEh	Timer 2B	NA	NA	NA	NA	NA	NA	NA	Yes	NA	NA	NA	NA	NA	NA	2
7FC0h	Trap 0–15	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	32
7FE0h	Reserved	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	12
7FECh	ADC†	NA	NA	NA	Yes	Yes	Yes	Yes	Yes	Yes	NA	Yes	NA	Yes	Yes	2
7FEEh	Timer 2A	NA	NA	NA	NA	NA	Yes	Yes	Yes	Yes	NA	NA	Yes	NA	NA	2
7FF0h	SCI TX‡	Yes	NA	Yes	NA	NA	Yes	Yes	Yes	NA	NA	NA	Yes	NA	Yes	2
7FF2h	SCI RX‡	Yes	NA	Yes	NA	NA	Yes	Yes	Yes	NA	NA	NA	Yes	NA	Yes	2
7FF4h	Timer 1	Yes	Yes	Yes	NA	NA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2
7FF6h	SPI	NA	Yes	Yes	NA	Yes	NA	Yes	Yes	NA	NA	NA	NA	NA	NA	2
7FF8h	Interrupt 3	Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes	Yes	NA	Yes	Yes	NA	2
7FFAh	Interrupt 2	Yes	Yes	Yes	Yes	NA	Yes	Yes	Yes	Yes	Yes	NA	Yes	Yes	NA	2
7FFCh	Interrupt 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2
7FFEh	Reset	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	2

NA = Not Available
† ADC refers to ADC1, ADC2, or ADC3.
‡ SCI refers to SCI1 or SCI2.

3.3.4.1 Program ROM Module (TMS370C0xx, TMS370C3xx, and TMS370C4xx Devices Only)

The program ROM module consists of read-only memory, which is programmed at the time of device fabrication. The present ROM module sizes are 2K, 4K, 8K, 12K, 16K, 24K, 32K, and 48K. All accesses to the ROM module require two system clock cycles. ROM security is a feature of the TMS370C45x devices which inhibits the reading of data using any programmer (secured memory feature).

Note:

All TMS370 family devices contain mask-ROM space reserved for TI use only. This space includes locations 7FE0h through 7FEBh. This reserved area should not be used in your software algorithm, nor should it be used during mask-ROM or firmware development.

The contents of the reserved locations are changed by TI only.

3.3.4.2 ROM-less Devices (TMS370C1xx and TMS370C2xx Devices Only)

Program memory for ROM-less devices must be off-chip. The TMS370 must be in the microprocessor mode to operate.

3.3.4.3 Program EPROM Modules (TMS370C6xx and TMS370C7xx Devices Only)

The program EPROM modules replace the program ROM for systems in prototype or small production runs. The modules presently consist of 8K, 16K, 24K, 32K, and 48K bytes of EPROM and the necessary programming control logic.

Read access to the program EPROM is performed as normal memory read cycles. Write cycles require a special sequence of events. See subsection 6.4.3, page 6-13, for a detailed discussion of programming the EPROM modules.

The EPROM can be written to only when V_{PP} is applied to the MC pin and the VPPS bit (EPCTLx.6) is set. When V_{PP} is applied to the MC pin, all on-chip EEPROM is in write protect override (WPO) mode, regardless of the state of the VPPS bit. This allows the EPROM to be protected while the EEPROM is in WPO.

3.4 Memory Operating Modes

Devices that have the memory expansion can operate in one of two major memory modes.

- **Microcomputer (μC)** modes
  - Microcomputer single-chip mode
  - Microcomputer with external expansion
- **Microprocessor (μP)** modes
  - Microprocessor without internal program memory
  - Microprocessor with internal program memory

Devices that do not have the memory expansion can operate only in the microcomputer single-chip mode. Table 3–3 shows the devices and the modes that they can operate in.

Table 3–3. Memory Modes Available Per Device

Mode	Member of TMS370C Family			Described in	
	x0x, x1x, x2x, x32, x36, x4x, x7x, x8x, x9x, xAx, xBx, xCx	15x, 25x	05x, 35x, 45x, 75x, x6x	Subsection	Page
Microcomputer Single Chip	Yes	No	Yes	3.4.1	3-17
Microcomputer External Expansion	No	No	Yes	3.4.2	3-19
Microprocessor Without Internal Program Memory	No	Yes	Yes	3.4.3	3-24
Microprocessor With Internal Program Memory	No	No	Yes	3.4.4	3-26

For devices that have the memory expansion, the basic microcomputer and microprocessor operating modes are selected by the voltage level applied to the dedicated MC pin when the $\overline{\text{RESET}}$ pin goes inactive (high).

- If the MC pin is low when the $\overline{\text{RESET}}$ signal goes high, then the processor enters the microcomputer mode.
- If the MC pin is high when the $\overline{\text{RESET}}$ signal goes high, then the processor enters the microprocessor mode.

Changing the MC pin alone does not change the memory mode. To change the memory operating mode, change the MC pin and then reset the device. Applying 12 volts to the MC pin *after* the $\overline{\text{RESET}}$ signal goes inactive high forces the device to enter the WPO mode.

Note: Consideration for MC-Pin Voltage Application

If 12 V is applied to the MC pin during the low-to-high transition of the $\overline{\text{RESET}}$ pin, the results are unpredictable.

If the processor resets into a microcomputer mode, the software can change the internal system configuration registers to select the desired memory expansion configuration. Part of this configuration setup involves digital I/O port D. Each pin of port D can be programmed to serve one of three purposes: digital I/O, function A signal, or function B signal. Function A includes chip-select signals, which can be used in the microcomputer mode with external memory expansion. Function B includes signals used in either the microcomputer or the microprocessor modes to access external memory chips.

3.4.1 Microcomputer Single-Chip Mode

In the microcomputer single-chip mode, a TMS370 device functions as a self-contained microcomputer with all memory and peripherals on the chip. This mode has no external address or data memory and allows more pins (used for the external buses in other modes) to be programmed as input/output pins. The single-chip mode maximizes the general-purpose I/O capability for real-time control applications. Figure 3–6 on page 3-18 shows a memory map for the microcomputer single-chip mode.

During reset, the MC pin must remain at a low level to successfully enter the microcomputer mode. While the device is operating in the single-chip mode, external circuitry can place 12 volts on the MC pin to enter the WPO mode to alter protected EEPROM.

Use the following steps to set a TMS370 device to the microcomputer single-chip mode:

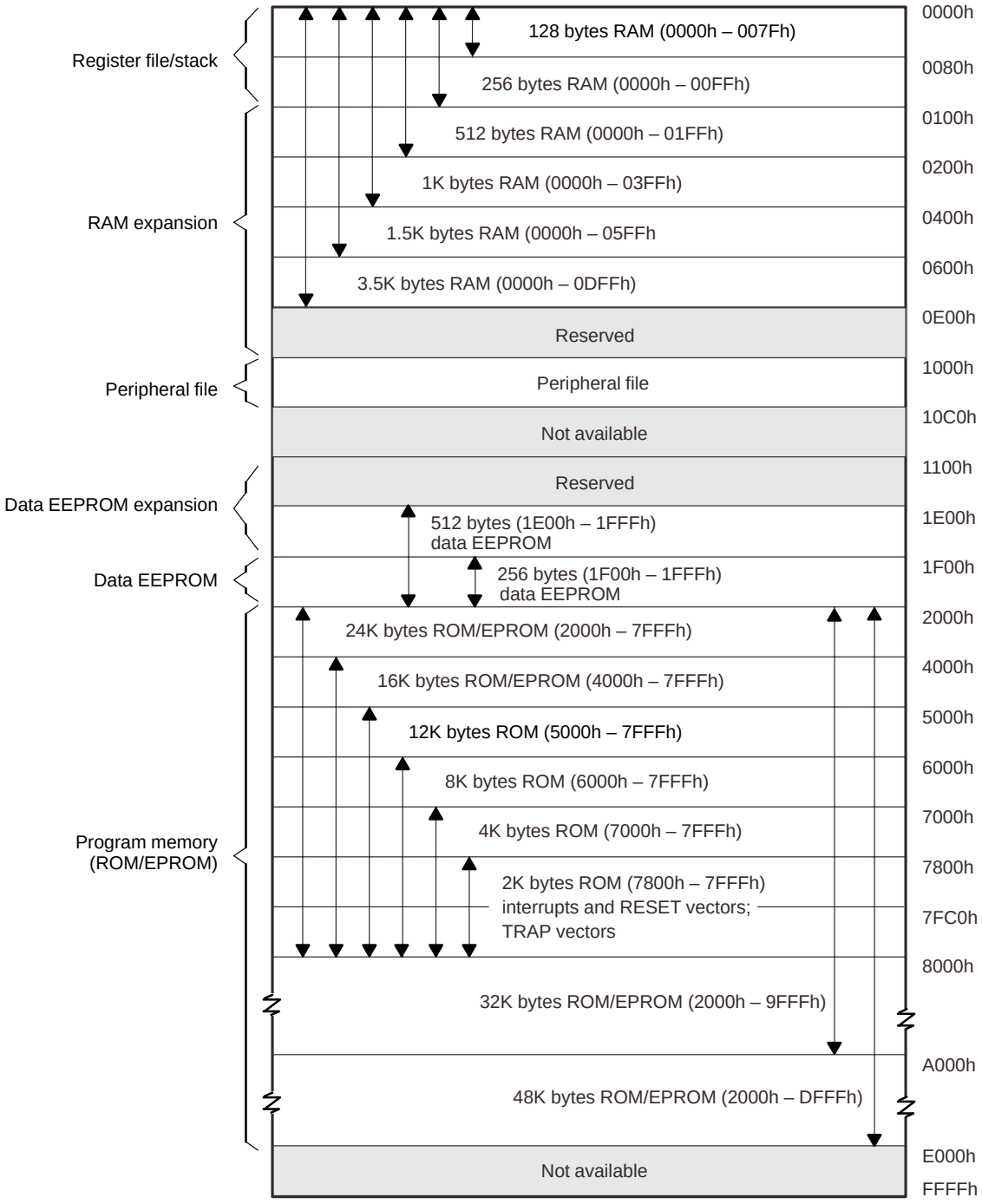
1) Place a low logic level on the MC pin.

2) Bring the $\overline{\text{RESET}}$ pin to active low, then return $\overline{\text{RESET}}$ to its inactive high state.

Note:

The preceding procedure must be followed for devices that do not have the memory expansion, even though they operate only in the microcomputer single-chip mode.

Figure 3–6. Microcomputer Single-Chip Mode



† In devices with more than 256 bytes of RAM, only the first 256-byte block can be used as registers or stacks.

3.4.2 Microcomputer Mode With External Expansion (All Devices With Memory Expansion and Internal Program Memory)

The microcomputer mode also supports memory expansion to external memory or peripherals, while all on-chip memory (register file, ROM, and EPROM) remains active. Digital I/O ports, under the control of their associated port control registers, become external memory as follows:

- Port A: 8-bit data memory
- Ports B and C: 16-bit address memory
- Port D: 8-bit control memory

Each pin that is not used for address, data, or control memory can be individually programmed as a general-purpose input/output pin. These bits are programmed by setting the digital I/O control registers in the peripheral file (see Section 4.4, page 4-18, for further information on programming I/O pins).

The address memory and data memory are nonmultiplexed, eliminating the requirement for an external address/data latch and thereby lowering system cost. External interface decode logic can be reduced further by using the precoded chip-select outputs.

The port D outputs can be programmed on a pin-by-pin basis to provide direct memory/peripheral chip-select or chip-enable functions. Each port D pin can be individually set to function A, function B, or general-purpose I/O.

Function A

When port D is set up to drive the chip-selection signals (function A), memory accesses to certain ranges of memory activate pins (refer to Table 3–4 on page 3-20, and Figure 3–7 on page 3-22):

Applications that use more than one chip-select signal for the same address should set the unused chip selects (i.e., chip selects not currently used to select memory banks) to general-purpose high-level outputs. For example, an application that uses both $\overline{\text{CSE1}}$ and $\overline{\text{CSE2}}$ can set one of these signals as the active chip-select function and set the other as a general-purpose high-level output. The $\overline{\text{CSH1}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSH3}}$ signals can be used as memory bank select signals under software control. The $\overline{\text{CSPF}}$ signal can be used as a chip select for the external expansion of the peripheral file. These chip-select control lines allow access to more than 112K bytes of external memory.

Table 3–4. Memory-Enabled Pins Activated when Memory Accessed

Amount of Internal Memory in Device	Memory Areas Accessed	Pins Activated†
4K, 8K, 12K, or 16K bytes	10C0h – 10FFh	Pin $\overline{\text{CSPF}}$ for 68-pin x50, x52, x53, and x56 devices if the pin is enabled
	2000h – 3FFFh	Pins $\overline{\text{CSE1}}$ and $\overline{\text{CSE2}}$ for 64- and 68-pin x50, x52, x53, and x56 devices
	8000h – FFFFh	Pins $\overline{\text{CSH1}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSE3}}$ for 68-pin x50, x52, x53, and x56 devices
		Pins $\overline{\text{CSH1}}$ and $\overline{\text{CSE3}}$ for 64-pin x50, x52, and x56 devices
24K or 32K bytes	10C0h – 10FFh	Pin $\overline{\text{CSPF}}$ for 68-pin x58, x67, and x68 devices if the pin is enabled
	A000h – BFFFh	Pins $\overline{\text{CSE1}}$ and $\overline{\text{CSE2}}$ for 64- and 68-pin x58 devices
		Pin $\overline{\text{CSE1}}$ for 68-pin x67 and x68 devices
	C000h – FFFFh	Pins $\overline{\text{CSH1}}$ and $\overline{\text{CSH3}}$ for 64-pin x58 devices
		Pins $\overline{\text{CSH1}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSH3}}$ for 68-pin x58 devices
		Pin $\overline{\text{CSH1}}$ for 68-pin x67 and x68 devices
48K bytes	10C0h – 10FFh	Pin $\overline{\text{CSPF}}$ for 68-pin x59 and x69 devices if the pin is enabled
	E000h – EFFFh	Pins $\overline{\text{CSE1}}$ and $\overline{\text{CSE2}}$ for 68-pin x59 devices
		Pin $\overline{\text{CSE1}}$ for 68-pin x69 devices
	F000h – FFFFh	Pins $\overline{\text{CSH1}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSH3}}$ for 68-pin x59 devices
		Pin $\overline{\text{CSH1}}$ for 68-pin x69 devices

† These pins are activated providing that the pins are enabled by the appropriate port control register(s).

See Section 4.4 (page 4-18) for a description of the digital I/O port control registers and how the chip-select signals are enabled. Table 3–5 on page 3-21 shows a memory map for the microcomputer mode with function A expansion.

Note: Areas Not Available for External Access

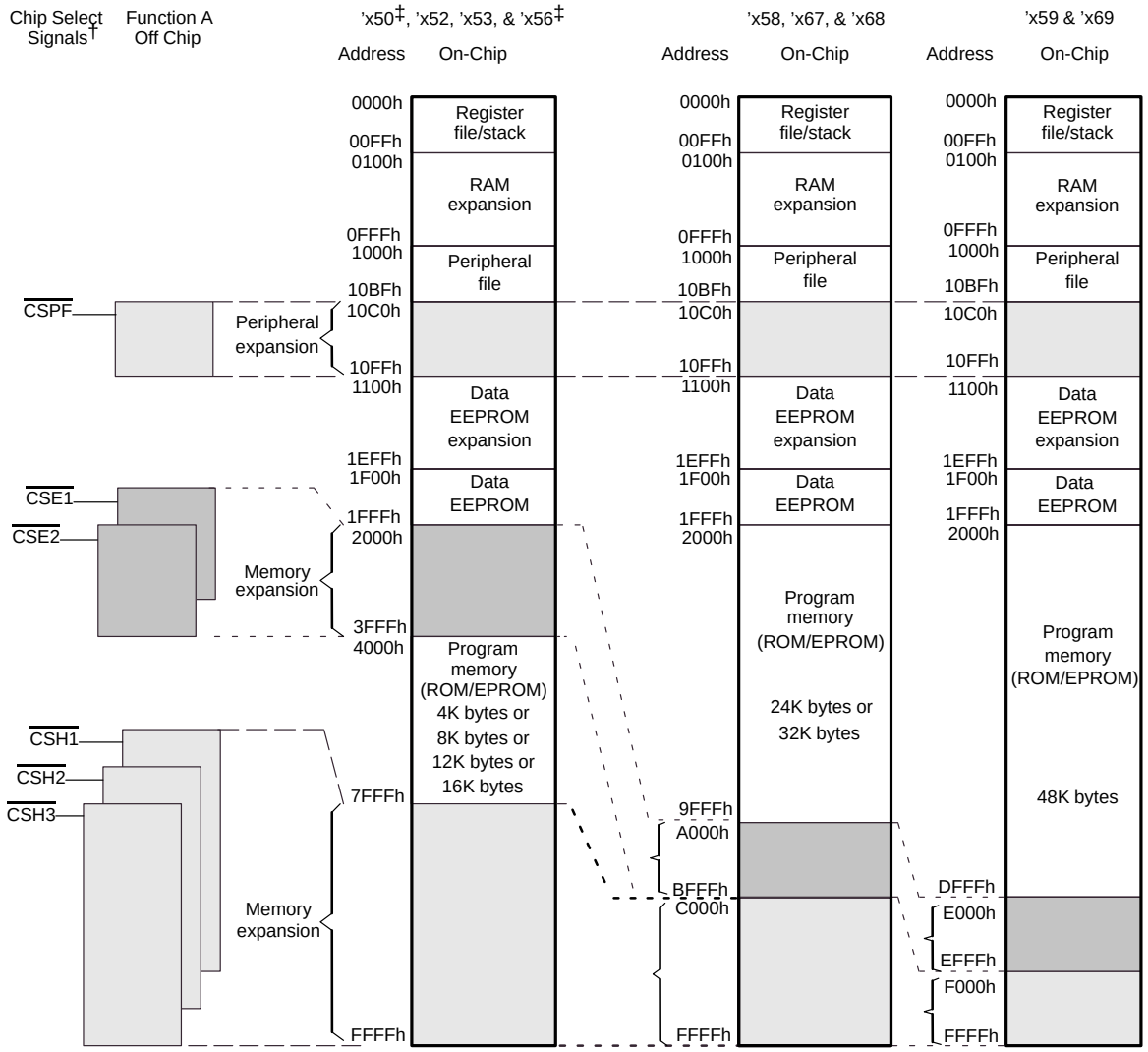
The RAM, data EEPROM, and internal memory expansion areas are not available for external access.

Table 3–5. Microcomputer Mode with Function A Expansion

Internal Program Memory (R=ROM E=EPROM)§	Maximum External Memory§	Function A: Chip-Select Signals (Showing Memory Areas)§					
		$\overline{\text{CSE1}}$ 64/68 pin	$\overline{\text{CSE2}}$ 64/68 pin	$\overline{\text{CSH1}}$ 64/68 pin	$\overline{\text{CSH2}}$ 68 pin	$\overline{\text{CSH3}}$ 64/68 pin	$\overline{\text{CSPF}}$ 68 pin
4KB R ('x50‡)	112KB	2000h–3FFFh (8KB)	2000h–3FFFh (8KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	10C0h–10FFh (64 B)
8KB R ('x52)	112KB	2000h–3FFFh (8KB)	2000h–3FFFh (8KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	10C0h–10FFh (64 B)
12KB R ('x53)	112KB	2000h–3FFFh (8KB)	2000h–3FFFh (8KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	10C0h–10FFh (64 B)
16KB R/E ('x56‡)	112KB	2000h–3FFFh (8KB)	2000h–3FFFh (8KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	8000h–FFFFh (32KB)	10C0h–10FFh (64 B)
†24KB R/E ('x67)	24KB/'x67 64KB/'x77	A000h–BFFFh (8KB)	A000h–BFFFh (8KB)	C000h–FFFFh (16KB)	C000h–FFFFh (16KB)	C000h–FFFFh (16KB)	10C0h–10FFh (64 B)
†32KB R/E ('x58, 'x68)	64KB/'x58 24KB/'x68	A000h–BFFFh (8KB)	A000h–BFFFh (8KB)	C000h–FFFFh (16KB)	C000h–FFFFh (16KB)	C000h–FFFFh (16KB)	10C0h–10FFh (64 B)
†48KB R/E ('x59, 'x69)	20KB/'x59 8KB/'x69	E000h–EFFFh (4KB)	E000h–EFFFh (4KB)	F000h–FFFFh (4KB)	F000h–FFFFh (4KB)	F000h–FFFFh (4KB)	10C0h–10FFh (64 B)

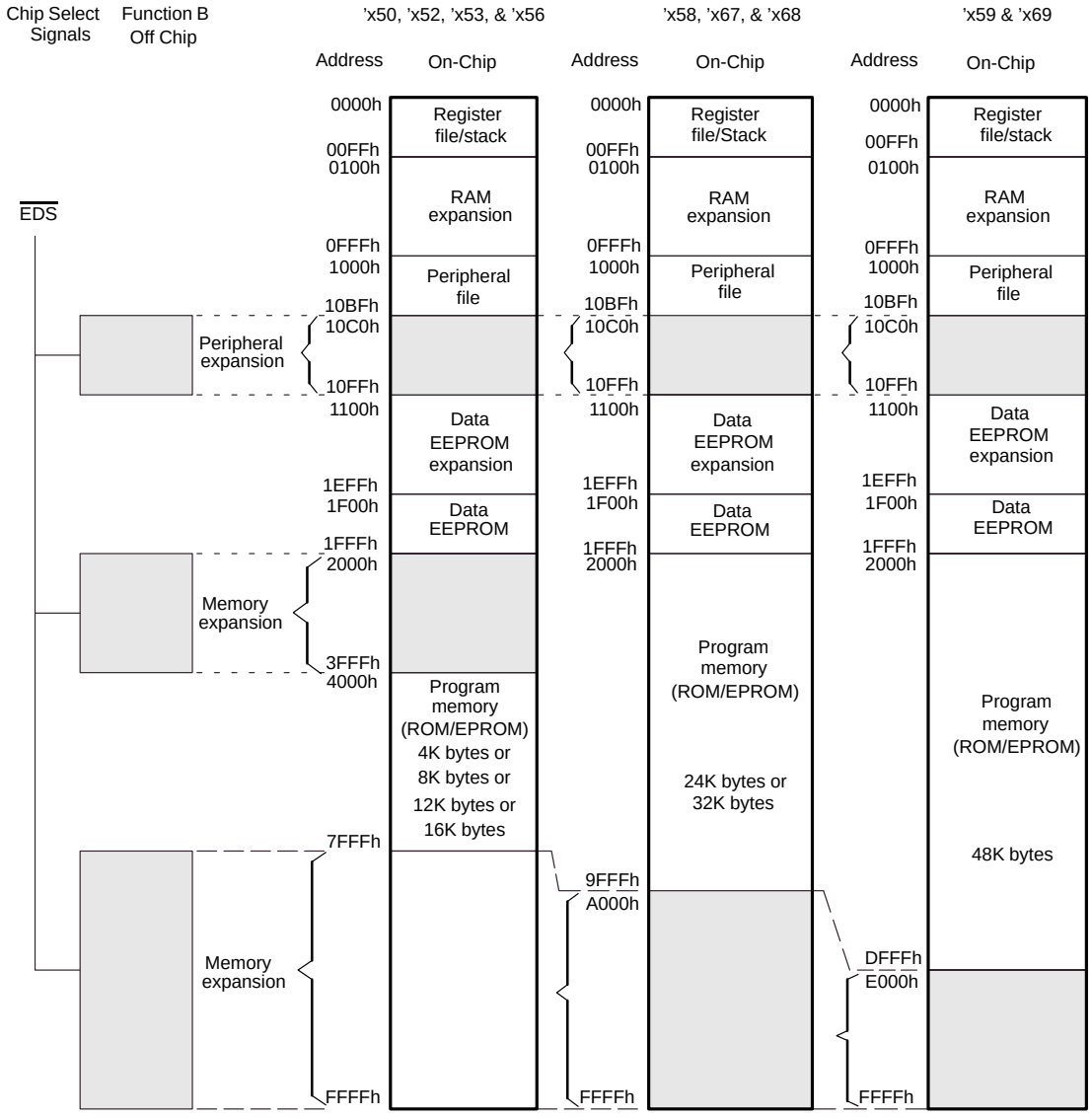
† Chip-select pins $\overline{\text{CSE2}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSH3}}$ are not available for the 'x6x device family.
‡ Devices '150, '250, '156, and '256 are ROMless. These devices cannot operate in microcomputer mode function A.
§ KB = kilobytes, where $1\text{KB} = 1024_{10}$.

Figure 3–7. Microcomputer Mode With Function A Expansion



† 1) Chip select signals $\overline{\text{CSH2}}$ and $\overline{\text{CSPF}}$ are not available in 64-pin devices.
2) Chip select signals $\overline{\text{CSE1}}$, $\overline{\text{CSH2}}$, and $\overline{\text{CSH3}}$ are not available for x6x devices.
‡ '150, '250, '156, and '256 are ROMless devices. These devices cannot operate in Microcomputer Mode Function A.

Figure 3–8. Microcomputer Mode With Function B Expansion



Function B

All predecoded chip selects have the same timing as the external data strobe ($\overline{\text{EDS}}$) signal (see Chapter 18, *Electrical Specifications*). $\overline{\text{EDS}}$ is a function B (microprocessor mode) signal that goes low whenever an access to external memory is made. Figure 3–8 shows a memory map for the microcomputer mode with function B expansion.

Table 3–6. Microcomputer with Function B Expansion

Internal Program Memory (R=ROM, E=EPROM)	Function B: Chip-Select Signal	Maximum External Memory
	$\overline{\text{EDS}}$ 64/68 Pins	
4K bytes R ('x50)	10C0h to 10FFh (64 bytes) 2000h to 3FFFh (8K bytes) 8000h to FFFFh (32K bytes)	40K bytes
8K bytes R ('x52)	10C0h to 10FFh (64 bytes) 2000h to 3FFFh (8K bytes) 8000h to FFFFh (32K bytes)	40K bytes
12K bytes R ('x53)	10C0h to 10FFh (64 bytes) 2000h to 3FFFh (8K bytes) 8000h to FFFFh (32K bytes)	40K bytes
16K bytes R/E ('x56)	10C0h to 10FFh (64 bytess) 2000h to 3FFFh (8K bytes) 8000h to FFFFh (32K bytes)	40K bytes
24K bytes R/E ('x67)	10C0h to 10FFh (64 bytes) A000h to FFFFh (24K bytes)	24K bytes
32K bytes R/E ('x58, 'x68)	10C0h to 10FFh (64 bytes) A000h to FFFFh (24K bytes)	24K bytes
48K bytes R/E ('x59, 'x69)	10C0h to 10FFh (64 bytes) E000h to EFFFh (4K bytes)	4K bytes

Table 3–6 shows the memory addresses for the function B chip-select signal. See Section 4.4 on page 4-18 for a description of the digital I/O port control registers and how the chip-select signals are enabled.

To place a device in to the microcomputer mode with external expansion (the device *must* have bus expansion), execute the following steps:

1) Place a low logic level on the MC pin.

2) Pull the $\overline{\text{RESET}}$ pin active low, then return $\overline{\text{RESET}}$ to its inactive high state.

3) Program the digital I/O registers to select the chip-select or control signals needed (function A or function B).

3.4.3 Microprocessor Mode Without Internal Memory (Memory Expansion Devices Only)

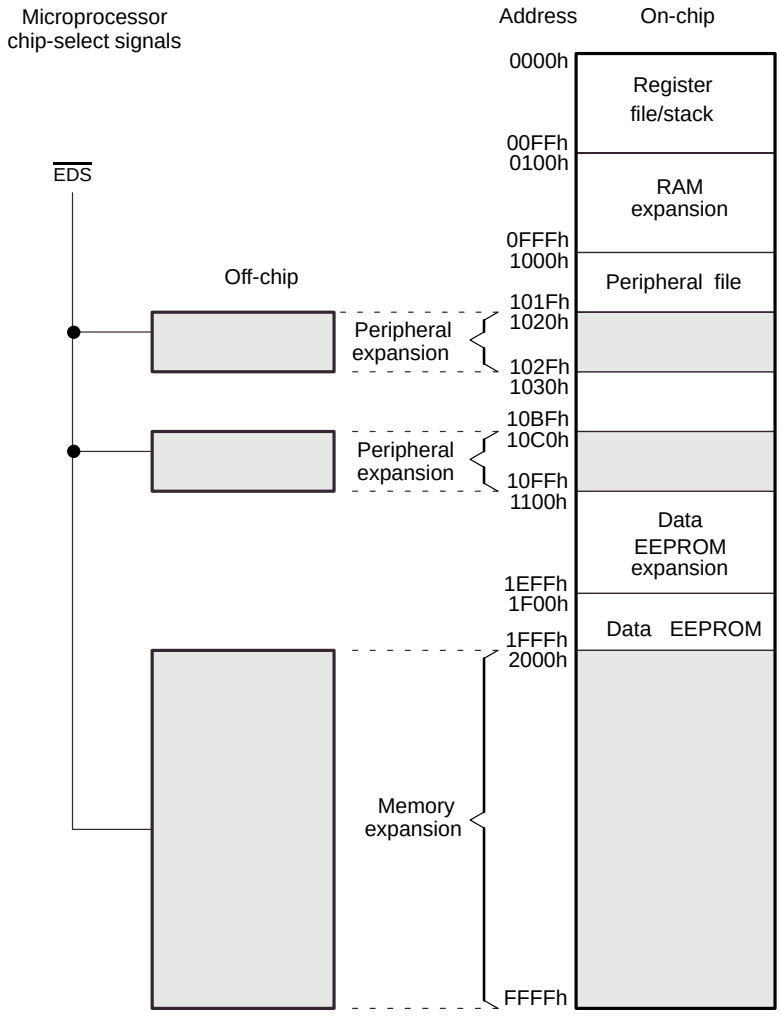
When a device is activated in the microprocessor mode, the register file and data EEPROM remain active, but the on-chip program ROM or EPROM is disabled. The $\overline{\text{EDS}}$ signal goes low when a memory access is made to addresses 1020h–102Fh, 10C0h–10FFh, and 2000h–FFFFh. The program area, the re-

set vector, interrupt vectors, and trap vectors must be located in off-chip memory locations.

When a device is reset into the microprocessor mode, the digital I/O port D registers are set to function B expansion memory control signals. The chip-select signals are not available in function B. Ports B and C are set up as the external address bus, and port A is set up to be the external data bus. Software cannot change the digital I/O configuration.

Figure 3–9 shows a memory map for the microprocessor mode without internal memory.

Figure 3–9. Microprocessor Mode Without Internal Memory



To place a device in the microprocessor mode without internal memory, do the following:

1) Place a high logic level on the MC pin.

2) Pull the $\overline{\text{RESET}}$ pin to active low, then return $\overline{\text{RESET}}$ to its inactive high state.

3.4.4 Microprocessor Mode With Internal Program Memory (Memory Expansion Devices Only)

Once the device is in microprocessor mode, the internal program memory can be re-enabled by clearing the MEMORY DISABLE bit (SCCR1.2). This mode is exactly the same as the microprocessor mode without internal memory except that when the internal memory is enabled, the $\overline{\text{EDS}}$ signal is no longer active during memory accesses to addresses 1020h–102Fh, and addresses that internal memory occupies. For example, when the MEMORY DISABLE bit (SCCR1.2) is cleared, memory accesses to addresses 1020h–102Fh and 5000h–7FFFh (x53 device) addresses access internal program memory; while $\overline{\text{EDS}}$ maintains the ability to access external memory locations 2000h–4FFFh and 8000h–FFFFh. This is shown in Table 3–7 and in Figure 3–10 (page 3-28), together with other TMS370 memory expansion devices. The actual amount of program memory available depends on the device.

In this mode, accesses to memory addresses 1020h through 102Fh are not valid for external memory or for the internal port control registers. This peripheral frame should not be used in this mode.

To use this mode, external memory must be implemented at memory addresses 7FFEh and 7FFFh to contain the reset vector. This memory can be switched in and out with the internal memory by clearing and setting the memory disable bit.

Figure 3–10 (page 3-28) shows a memory map for the microprocessor mode with internal program memory.

Table 3–7. Microprocessor Mode with Internal Program Memory

Device	Internal Memory Disabled (SCCR1.2=1)		Internal Memory Enabled (SCCR1.2=0)	
	Active Internal Memory	**Active External Memory**	**Active Internal Memory**	**Active External Memory**
TMS370Cx50	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 7000h – 7FFFh	10C0h – 10FFh 2000h – 6FFFh 8000h – FFFFh
TMS370Cx52	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 6000h – 7FFFh	10C0h – 10FFh 2000h – 5FFFh 8000h – FFFFh
TMS370Cx53	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 5000h – 7FFFh	10C0h – 10FFh 2000h – 4FFFh 8000h – FFFFh
TMS370Cx56	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 4000h – 7FFFh	10C0h – 10FFh 2000h – 3FFFh 8000h – FFFFh
TMS370Cx67	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 2000h – 7FFFh	10C0h – 10FFh 2000h – 1FFFh 8000h – FFFFh
TMS370Cx58, TMS370Cx68	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 2000h – 9FFFh	10C0h – 10FFh 2000h – 1FFFh A000h – FFFFh
TMS370Cx59, TMS370Cx69	—	10C0h – 10FFh 2000h – FFFFh	1020h – 102Fh 2000h – DFFFh	10C0h – 10FFh 2000h – 1FFFh E000h – FFFFh

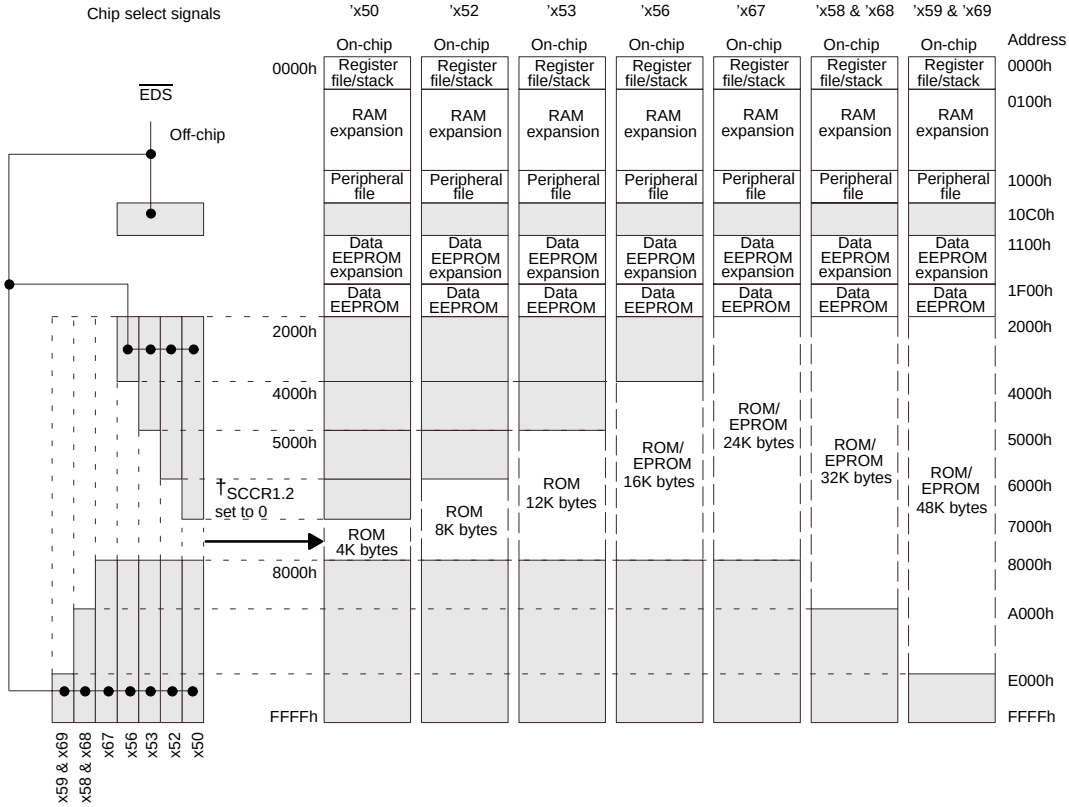
To place a device in the microprocessor mode with internal program memory, use the following steps:

1) Place a high logic level on the MC pin.

2) Take the $\overline{\text{RESET}}$ pin active low, then return $\overline{\text{RESET}}$ to its inactive high state.

3) The CPU reads the reset vectors from external memory (7FFEh/7FFFh). The program pointed to by the vectors must include code to clear the MEMORY DISABLE bit (SCCR1.2) to enable the internal memory. The internal program memory becomes available. (The SCCR1 register is described in subsection 4.3.2 on page 4-15).

Figure 3–10. Microprocessor Mode With Internal Program Memory



† After the μP is reset, the CPU reads the reset vectors from external memory (7FFEh/7FFFh) until the MEMORY DISABLE bit (SCCR1.2) is cleared by software, then the internal memory is enabled.

3.4.5 Memory Mode Summary

Table 3–8 summarizes the features of each memory mode and outlines the procedure for putting the TMS370 device into each mode.

Table 3–8. Operating Mode Summary

Feature	μComputer Single-Chip Mode	μComputer With Expanded Memory Mode	μProcessor with Internal Memory Mode	μProcessor Mode
Devices	All TMS370s with internal program memory	Devices with memory expansion and internal program memory	Devices with memory expansion and internal program memory	Devices with memory expansion
Program Memory	Internal	Internal	Internal and external	External
Ports A, B,C,D	Digital I/O	Digital I/O Function A† Function B‡	Function B‡	Function B‡
Predecoded CS (chip selects)	No	Optional	No	No
Procedure to enter the mode	1) Place logic 0 on the MC pin. 2) Set the $\overline{\text{RESET}}$ pin to active low, then release $\overline{\text{RESET}}$.	1) Place logic 0 on the MC pin. 2) Set the $\overline{\text{RESET}}$ pin to active low, then release $\overline{\text{RESET}}$. 3) Set digital I/O registers to function A†/B‡.	1) Place logic 1 on the MC pin. 2) Take the $\overline{\text{RESET}}$ pin active low, then release $\overline{\text{RESET}}$. 3) Enable internal memory (clear SCCR1.2).	1) Place logic 1 on the MC pin. 2) Take the $\overline{\text{RESET}}$ pin active low, then release $\overline{\text{RESET}}$.

† Function A: Port D = Chip-select signals for 68-pin $\overline{\text{CSE1}}$, $\overline{\text{CSE2}}$, $\overline{\text{CSH1}}$, $\overline{\text{CSH2}}$, $\overline{\text{CSH3}}$, and $\overline{\text{CSPF}}$ (see Section 4.4, pp. 4-18).
= Chip-select signals for 64 pin $\overline{\text{CSE1}}$, $\overline{\text{CSE2}}$, $\overline{\text{CSH1}}$, and $\overline{\text{CSH3}}$ (see Section 4.4).
‡ Function B: Port D = Expansion memory control signals $\overline{\text{OCF}}$, $\overline{\text{EDS}}$, and $\overline{\text{WAIT}}$ (see Section 4.4).

Chapter 4

System and Digital I/O Configuration

This chapter discusses the system and I/O configuration. Features and options are described, as well as the registers that control the configuration. Examples of how to set specific configurations are also given. This chapter covers the following topics:

Topic **Page**

4.1 System Configuration

The system configuration is controlled and monitored by the first three registers of peripheral file frame 1. These registers' names, designations, addresses, and peripheral file (PF) register numbers are shown below and in Figure 4–1. The PF numbers are used by peripheral file instructions; for example, MOV #00h,P010 clears PF P010 at address 1010h.

Name	Designation	Address	PF
System control and configuration register 0	SCCR0	1010h	P010
System control and configuration register 1	SCCR1	1011h	P011
System control and configuration register 2	SCCR2	1012h	P012

Figure 4–1. Peripheral File Frame 1: System Configuration and Control Registers

			Bit Names/Functions							
Register Name	Addr	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCCR0	1010h	P010	COLD START (RW–*)	OSC POWER (RP–0)	PF AUTO-WAIT (RP–0)	OSC FLT FLAG (RW–0)	MC PIN WPO (R–0)	MC PIN-DATA (R–*)	—	µP/µC MODE (R–*)
SCCR1	1011h	P011	—	—	—	AUTO-WAIT DISABLE (RP–0)	—	MEMORY DISABLE (RP–*)	—	—
SCCR2	1012h	P012	HALT/ STAND-BY (RP–0)	PWR-DWN/ IDLE (RP–0)	—	BUS STEST (RP–0)	CPU STEST (RP–1)	—	INT1 NMI (RP–0)	PRIVI-LEGE DISABLE (RS–0)

Note: Shaded boxes denote privileged mode bits that can be written to only in the privileged mode.

4.1.1 Privilege Mode

The TMS370 architecture allows you to configure the system and peripherals by software to meet the requirements of a variety of applications. The privilege mode of operation ensures the integrity of the system configuration, once it is defined for an application.

The shaded boxes in Figure 4–1 denote privilege mode bits; that is, you can write to these bits only in the privilege mode.

Following a hardware reset, the processor operates in the privilege mode. In this mode, peripheral file registers have unrestricted read/write access. The application program can configure the system during the initialization sequence following a reset. As the last step of a system initialization, set the PRIVILEGE DISABLE bit (SCCR2.0) to enter the nonprivilege mode and prevent changes to specific control bits within the peripheral file.

Table 4–1 shows the system configuration bits that are write-protected during the nonprivilege mode. These bits should be configured by software before exiting the privilege mode.

Table 4–1. Privilege-Mode Configuration Bits

Register	Bit	Bit Name
SCCR0	PF AUTOWAIT OSC POWER	Peripheral File Automatic Wait Cycle Oscillator Power
SCCR1	MEMORY DISABLE AUTOWAIT DISABLE	Memory Disable Automatic Wait-State Disable
SCCR2	PRIVILEGE DISABLE INT NMI CPU STEST BUS STEST PWRDWN/IDLE HALT/STANDBY	Privilege Mode Disable Interrupt 1, Nonmaskable Interrupt CPU Factory Test BUS Factory Test Powerdown/Idle Halt/Standby
SPIPRI	SPI STEST SPI PRIORITY SPI ESPEN	SPI Factory Test SPI Interrupt Priority Select SPI emulator suspend enable
SCIPRI	SCI STEST SCITX PRIORITY SCIRX PRIORITY SCI ESPEN	SCI Factory Test SCI Transmitter Interrupt Priority Select SCI Receiver Interrupt Priority Select SCI emulator suspend enable
T1PRI	T1 STEST T1 PRIORITY	Timer 1 Factory Test Timer 1 Interrupt Priority Select
T2APRI	T2A STEST T2A PRIORITY	Timer 2A Factory Test Timer 2A Interrupt Priority Select
T2BPRI	T2B STEST T2B PRIORITY	Timer 2B Factory Test Timer 2B Interrupt Priority Select
ADPRI	AD STEST AD PRIORITY AD ESPEN	AD Factory Test AD Interrupt Priority Select AD emulator suspend enable
PACTPRI	PACT WD PRESCALE SELECT 0 PACT WD PRESCALE SELECT 1 PACT MODE SELECT PACT GROUP 3 PRIORITY PACT GROUP 2 PRIORITY PACT GROUP 1 PRIORITY PACT STEST	PACT Watchdog Prescale Select 0 PACT Watchdog Prescale Select 1 PACT Mode Select PACT Group 3 Priority Select PACT Group 2 Priority Select PACT Group 1 Priority Select PACT Stest
PACTSCR	PACT PRESCALE SELECT 0 PACT PRESCALE SELECT 1 PACT PRESCALE SELECT 2 PACT PRESCALE SELECT 3 FAST MODE SELECT	PACT Prescale Select 0 PACT Prescale Select 1 PACT Prescale Select 2 PACT Prescale Select 3 Fast Mode Select

The only way to change the privilege bits after leaving the privilege mode is to reset the processor and program the control registers. The write protection override (WPO) used for the EEPROM has no effect on the privilege bits.

Note that privilege mode has no effect on timer 1 (T1) watchdog (WD) bits. These bits are protected in a separate manner.

4.1.2 Clock Options

The TMS370 family provides two clock options which are referred to as divide-by-1 (phase-locked loop) and divide-by-4 (standard oscillator). Both, the divide-by-1 and divide-by-4 options are configurable during the manufacturing process of a TMS370 microcontroller. The ROM-masked devices offer both options to meet system engineering requirements. Only one of the two clock options is allowed on each ROM device.

The divide-by-1 clock module option provides the capability for reduced electromagnetic interference (EMI) with no added cost.

The divide-by-1 option provides a one-to-one match of the external resonator frequency (CLKIN) to the internal system clock (SYSCLK) frequency, whereas the divide-by-4 option produces a SYSCLK which is one-fourth the frequency of the external resonator. Inside the divide-by-1 module, the frequency of the external resonator is multiplied by four, and the clock module then divides the resulting signal by four to provide the four internal system clock signals. The resulting SYSCLK is equal to the resonator frequency. These are formulated as follows:

$$\text{Divide-by-4 option:}\quad \text{SYSCLK} = \frac{\text{external resonator frequency}}{4} = \frac{\text{CLKIN}}{4}$$

$$\text{Divide-by-1 option:}\quad \text{SYSCLK} = \frac{\text{external resonator frequency x 4}}{4} = \text{CLKIN}$$

The main advantage of choosing a divide-by-1 oscillator is the improved EMI performance. The harmonics of low-speed resonators extend through less of the emissions spectrum than the harmonics of faster resonators. The divide-by-1 option also reduces the resonator speed by 4x, resulting in a steeper decay of emissions produced by the oscillator.

4.1.3 Oscillator Fault

The processor contains a system of circuits to monitor the oscillator operation and to detect and contain major oscillator problems. This enhances processor and system reliability and aids in system recovery from a crash that is caused by a temporary fault.

The circuit stops the processor whenever circuitry detects an out-of-range oscillator operation. The oscillator fault detection circuitry consists of:

- An amplitude detector to detect whether the oscillator signal has a proper voltage level.
- A frequency detector to sense when the oscillator frequency goes too low. The oscillator fault detection circuit will always trigger below 20 kHz and never above 500 kHz.

The oscillator circuitry is designed to delay operation of the device until a stable clock signal is received. This protects against slow crystal startup times coming out of a halt mode or after an oscillator fault when the input clock cannot be operating at the correct voltage range. Device operation remains suspended until the input clock signal is within the required voltage range.

Whenever the oscillator fault detection circuitry detects a major oscillator problem, the processor generates a reset by pulling the $\overline{\text{RESET}}$ pin low for at least eight cycles; this causes external devices to reset with the processor. After a reset, the program checks the oscillator fault flag (OSC FLT FLAG, SCCR0.4), the cold start flag (COLD START, SCCR0.7), and the watchdog reset key (WDRST) to determine the source of the reset. A reset does not clear these flags.

4.1.4 Automatic Wait States

If an application system uses peripherals or expansion memory with access times slower than those of the TMS370 processor, wait states are required. Other microprocessors require complex additional circuitry, but the TMS370 series provides for the automatic addition of wait states that can slow the processor's access time to a compatible period.

The TMS370 series has a $\overline{\text{WAIT}}$ pin that can hold the processor in a wait state indefinitely. The following two bits control the insertion of the automatic wait state:

- The PF AUTOWAIT bit (SCCR0.5) that controls the external frames of the peripheral file so that these frames can access off-chip peripherals.
- The AUTOWAIT DISABLE bit (SCCR1.4) that controls all other external memory.

When the AUTOWAIT DISABLE bit equals 1, any access to *external* memory (excluding the PF file) takes two system clock cycles to complete. When AUTOWAIT DISABLE bit equals 0, the access takes three system clock (SYSCLK) cycles. The reset value of this bit selects the slower 3-cycle access.

When the PF AUTOWAIT bit equals 1, memory access to the *external* peripheral files takes four SYSCLK cycles. This bit does not affect the accesses to the internal registers. When the PF AUTOWAIT bit equals 0, the memory is treated like any external memory, and the AUTOWAIT DISABLE bit selects the number of SYSCLK cycles per access as either two or three cycles.

Table 4–2 summarizes the effects of the wait-state control bits.

Table 4–2. Wait-State Control Bits

Wait State Control Bits		No. of Clock Cycles per Access	
PF AUTOWAIT Bit (SCCR0.5)	**AUTOWAIT DISABLE Bit (SCCR1.4)**	**PF File**	**External Memory**
0	0	3	3
0	1	2	2
1	0	4	3
1	1	4	2

An external device can pull the $\overline{\text{WAIT}}$ input pin low and cause the processor to wait an indefinite number of clock cycles for its data. When the wait line is released, the processor resynchronizes with the rising edge of the SYSCLK signal and continues with the program. The $\overline{\text{WAIT}}$ pin is sampled only during *external* memory cycles.

Note: Tie Un-needed $\overline{\text{WAIT}}$ Line to V_{CC}

When constructing an application circuit with expansion memory, do not forget to connect any unneeded $\overline{\text{WAIT}}$ line to V_{CC}.

4.2 Low-Power and Idle Modes

The OTP, mask-ROM, and reprogrammable EPROM devices have two low-power (powerdown) modes and an idle mode. For mask-ROM devices, low-power modes can be disabled permanently through a programmable contact at the time when the mask is manufactured (refer to Chapter 19, *Customer Information,* for order information about mask-ROM devices).

Note: Low-Power Mode Difference

Low-power modes operate differently for TMS370Cxxx, TMS370CxxxA, and TMS370CxxxB devices. Refer to Section A.3, page A-4.

The low-power modes reduce the operating power by reducing or stopping the activity of various modules. The processor has the following two types of low-power modes: the halt mode and the standby mode (see Table 4–3). Bits 6 and 7 of register SCCR2 select the halt, standby, or idle modes.

- The **standby** mode stops the internal clock in every module except the T1 module. The T1 module continues to run and can bring the processor out of the standby mode. In devices with the PACT module, only the default timer and the first command are active in standby mode.
- The **halt** mode stops the internal clock. This stops processing in all of the modules, resulting in the lowest amount of power consumption.
- The **idle** mode (which is not a low-power mode) is a state that waits for the next interrupt.

Executing an IDLE instruction causes the processor to enter one of the two low-power modes or the simple idle mode, depending on SCCR2.6 and SCCR2.7. The low-power and idle-mode selection bits are summarized in Table 4–3.

Table 4–3. Powerdown/Idle Control Bits

Powerdown Control Bits		
PWRDWN/IDLE (SCCR2.6)	**HALT/STANDBY (SCCR2.7)**	**Mode Selected**
1	0	Standby
1	1	Halt
0	Don’t care	Idle

When low-power modes are disabled through a programmable contact in the mask-ROM devices, writing to the SCCR2.6–7 bits is ignored. In addition, if you execute an IDLE instruction when low-power modes are disabled through a programmable contact, the device *always* enters the idle mode.

To provide a method of exiting low-power modes for mask-ROM devices, INT1 is automatically enabled as a nonmaskable interrupt (NMI) during low-power modes when the hard WD mode is selected. This means that the NMI is always generated, regardless of the status of the interrupt enable flags and the values of the status bits (INT1 PRIORITY bit [INT1.1], INT1 ENABLE bit [INT1.0], INT1 NMI bit [SCCR2.1]), and global interrupt enable flags of the status register (IE1 and IE2). See subsection 7.7.2.1, page 7-26, for more information. The low-power modes and the methods of exiting these modes are discussed further in subsections 4.2.1 and 4.2.2 on page 4-9.

In the standby and halt modes, the digital output ports remain active. In addition, the following information is retained:

- The CPU registers:
  - Program counter (PC)
  - Status (ST)
  - Stack pointer (SP)
- The contents of the RAM
- The digital output data registers
- Control and status registers of all the modules, including the timer contents and the WD counter

If the serial peripheral interface (SPI) or serial communications interface (SCI—SCI1 or SCI2) is in the process of receiving or transmitting data when a low-power mode is entered, the data received or transmitted can be lost. The results of an analog-to-digital (AD) conversion (ADC1, ADC2, or ADC3) or an EEPROM write process is invalid when a low-power mode is entered.

Use caution when using low-power modes in conjunction with the WD mode; the WD stops counting in both low-power modes. In the standard WD option, if the program executes an IDLE instruction without the interrupts enabled (described in subsections 4.2.1 and 4.2.2, page 4-9), then only a reset can start the processor running again.

For additional information regarding WD operation during low-power modes, see subsection 7.7.2.1 (page 7-26) and Section 7.8 (page 7-29).

4.2.1 Standby Mode

The standby mode uses less power than the normal operating mode but more than the halt mode. The standby mode stops the clocks to every module except the T1 module or the PACT module. These modules can bring the processor out of this low-power mode if the interrupts are enabled. To enter the standby mode, use the following steps:

- ❑ Set the PWRDWN/IDLE bit (SCCR2.6)
- ❑ Clear the HALT/STANDBY bit (SCCR2.7).
- ❑ The next execution of an IDLE instruction causes the processor to enter the standby mode.

You can cause the processor to exit the standby mode by any one of the following methods:

- ❑ A reset
- ❑ An external interrupt 1, 2, or 3 (if enabled)
- ❑ A low level on the SCIRXD pin if the SCI RX interrupt and receiver are enabled (described in Chapter 9, *Serial Communications Interface 1 (SCI1) Module* — subsections 9.8.2 (page 9-24) and 9.8.5 (page 9-29), and Chapter 10, *Serial Communications Interface 2 (SCI2) Module* — subsections 10.8.2 (page 10-20) and 10.8.5 (page 10-24)).

Note:

When using the SCIRXD pin as a method of exiting the standby mode, the first character received will wake–up the device; therefore, the first character will be misinterrupted and not be treated as valid data. Subsequent data transmissions are valid.

- ❑ A T1 or PACT's first command/definition entry interrupt, if enabled

For additional standby mode power savings, see subsection 4.2.3 on page 4-10.

4.2.2 Halt Mode

The halt mode stops all internal operations and clocks (including T1 and the PACT counter) and uses the least power of the low-power modes. T1 cannot bring the processor out of this low-power mode. To select the halt mode, use the following steps:

- ❑ Set the PWRDWN/IDLE bit (SCCR2.6)
- ❑ Set HALT/STANDBY bit (SCCR2.7)
- ❑ Execute an IDLE instruction.

Note:

The OSC power bit (SCCR0.6) must be cleared before entering the HALT mode in order to exit the HALT mode properly.

You can cause the processor to exit the halt mode by any one of the following methods:

- A reset
- An external Interrupt 1, 2, or 3 if enabled
- A low level on the SCIRXD pin if the SCI RX interrupt and receiver are enabled (described in Chapter 9, *Serial Communications Interface 1 (SCI1) Module*, and Chapter 10, *Serial Communications Interface 2 (SCI2) Module*.

Note:

When using the SCIRXD pin as a method of exiting the halt mode, the first character received will wake–up the device; therefore, the first character will be misinterrupted and not be treated as valid data. Subsequent data transmissions are valid.

4.2.3 Using Interrupts to Exit From the Halt Mode

You must be aware of the following items when using an interrupt to exit the halt mode:

- Interrupts enabled during the halt mode are level-sensitive and not edge-sensitive.
- The interrupt must be at the inactive level when the device enters the halt mode.
- The processor exits the halt mode when the interrupt goes from the inactive level to the active level.
- Bit 2 in the interrupt control register determines the active and inactive levels.
  - When the INT2 POLARITY bit (INT2.2) is selected to be triggered on the rising edge, the active level is high.
  - When the INT2 POLARITY bit is selected to be triggered on the falling edge, the active level is low.
- The interrupt should be at an active level until the oscillator is stable enough to generate an interrupt.

- If the halt mode is entered with the interrupt at an active level, the following actions occur:
  - For devices with a divide–by–4 clock, the processor exits the halt mode and enters the idle mode.
  - For devices with a PLL clock, an oscillator fault flag (SCCR0.4) is set causing a system reset to occur.
- When the selected interrupt edge is detected, the program continues.

Refer to Figure 4–2 and Figure 4–3 for the differences between the correct and incorrect methods for entering the halt mode.

Figure 4–2. Correct Method to Enter Halt Mode

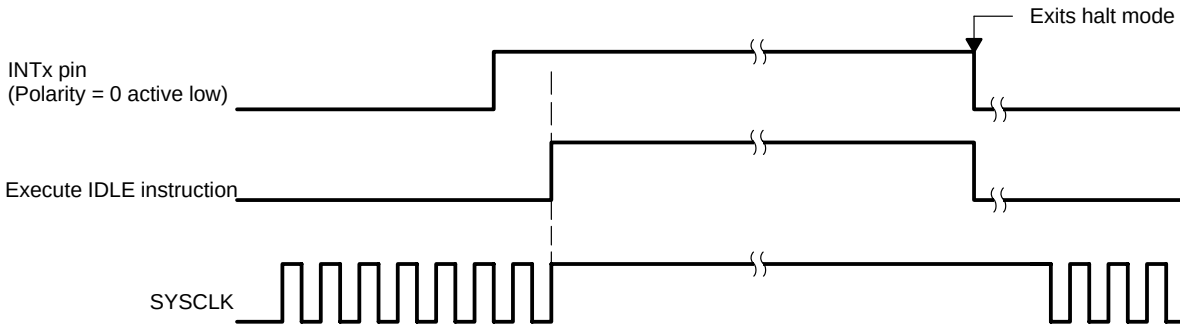
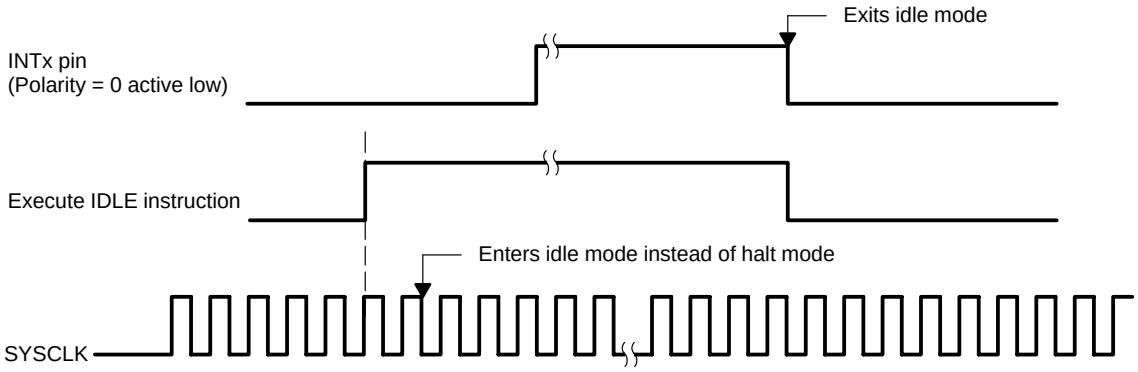


Figure 4–3. Improper Method to Enter Halt Mode



If this halt-mode condition exists and the device uses the WD, then the WD can reset while the program is waiting and unable to service the WD in the normal power idle mode.

The same considerations apply when you use the SCIRXD pin to exit the halt mode. The processor exits halt mode anytime an enabled SCI receiver and pin detect a low level on SCIRXD.

4.2.4 Oscillator Power Bit

The OSC POWER bit (SCCR0.6) allows additional standby mode power savings. When in effect, this feature reduces the oscillator drive current and disables the oscillator fault detection circuitry. The OSC POWER bit can be used effectively with a frequency of up to 3-MHz SYSCLK. If the SYSCLK frequency is greater than 3 MHz, this bit must be cleared. For power reduction specifications, see Chapter 18, *Electrical Specifications*.

4.3 System Control Registers

Each system control register is summarized in the following charts with definitions of each control bit.

4.3.1 System Control and Configuration Register 0 (SCCR0)

System Control and Configuration Register 0 (SCCR0)
[Memory Address 1010h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P010	COLD START	OSC POWER	PF AUTOWAIT	OSC FLT FLAG	MC PIN WPO	MC PIN DATA	—	µP/µC MODE
	RW-*	RP-0	RP-0	RW-0	R-0	R-*		R-*

R = Read, W = Write, P= Privilege write only, C = Clear only,
-n = Value of the bit after the register is reset, -* = See bit description

Bit 7 **COLD START.** Cold Start Flag.

This bit indicates whether the microcontroller is coming out of a powerup reset. This bit does not change during a reset under normal power.

0 = No full-power cycle occurred since last writing a 0 to this bit. This setting is used to determine the source of a reset.
1 = A full-power cycle has occurred since last writing a 0 to this bit. If the application does not zero this bit, the bit has no meaning.

Only writing a 0 to this bit can clear the COLD START flag. This bit is set to 1 only after the V_{CC} is off for several hundred milliseconds. As a result, you cannot use this bit to detect short V_{CC} glitches or brownout conditions.

Note: COLD START Bit Considerations

The COLD START bit is not designed to detect short V_{CC} glitches or brownout conditions.

Bit 6 **OSC POWER.** Oscillator Power.

This bit controls an oscillator power reduction feature. When this feature is in effect, the oscillator drive current is reduced, and the oscillator fault detection circuitry is powered down. Current reduction is most useful in the standby mode. However, when this bit is set during normal operation, the operating mode power consumption can be slightly reduced. Before entering the HALT mode, the OSC POWER bit (SCCR0.6) must be cleared in order to exit the HALT mode properly. This feature is effective up to a 3-MHz maximum SYSCLK frequency. If the SYSCLK frequency is greater than 3 MHz, this bit must be cleared. For power reduction specifications, see Chapter 18, *Electrical Specifications.*

0 = No oscillator drive current reduction.
1 = Oscillator drive current reduction.

Bit 5 **PF AUTOWAIT.** Peripheral File Automatic Wait Cycle.

0 = Any access to the peripheral file will take two system clock cycles with no system auto wait (bit 4 of SCCR1=1), or three system clock cycles with the system autowait on (bit 4 of SCCR1=0). (See subsection 4.1.4, page 4-5.)
1 = Any access to the upper four frames of the peripheral file (address 10C0h to 10FFh) will take four system clock cycles to complete. This eases interface requirements for peripheral devices slower than the TMS370 processor. Normal full-speed operation consists of two system clock cycles per access.

Bit 4 **OSC FLT FLAG.** Oscillator Fault Flag.

This flag is reset upon an initial power-up reset. A reset under power does not affect this flag. Therefore, this bit can be polled to determine the source of a reset.

0 = No oscillator fault found.
1 = Oscillator fault found. Oscillator period is now or was out of the correct operating range. The oscillator fault detect circuit always triggers below 20 KHz and never above 500 kHz.

Bit 3 **MC PIN WPO.** Mode Control Pin Write Protect Override Status.

This bit indicates whether the voltage on the MC pin is adequate for WPO functions. (If this bit is set, then bit 2 is also set.)

0 = Voltage on the MC pin is not enough to override write protection.
1 = Voltage on the MC pin is enough for write-protect override. Protected bits in data EEPROM and program EPROM can now be written to. Override voltage is nominally 12 volts.

Bit 2 **MC PIN DATA.** Mode Control Pin Data.

This bit shows the current status of the MC pin.

0 = Voltage on the MC pin is a logic 0 level.
1 = Voltage on the MC pin is a logic 1 level.

Bit 1 **Reserved.** Read data is indeterminate.

Bit 0 **μP/μC MODE.** Microprocessor/Microcomputer Mode.

This bit indicates the current operating mode (as described in subsection 3.4, page 3-16).

0 = Currently operating in microcomputer mode.
1 = Currently operating in microprocessor mode.

4.3.2 System Control and Configuration Register 1 (SCCR1)

System Control and Configuration Register 1 (SCCR1)
[Memory Address 1011h]

Bit #	7	6	5	4	3	2	1	0
P011	—	—	—	AUTO-WAIT DISABLE	—	MEMORY DISABLE	—	—
				RP-0		RP-*		

R = Read, P= Privilege write only, *-n* = Value of the bit after the register is reset, -* = See bit description

Bits 7–5 **Reserved.** Read data is indeterminate.

Bit 4 **AUTOWAIT DISABLE.** Automatic Wait State Disable.

This bit is cleared after a reset, and causes an extra cycle to be added to all external bus accesses to accommodate slower memory.

0 = Enables the autowait feature and makes the external bus access three system clock cycles long.
1 = Disables the autowait feature and makes the external bus access two system clock cycles long.

Changes to this bit can occur only in the privilege mode. If the PF AUTOWAIT bit in SCCR0 is set, external peripheral file access takes four SYSCLK cycles, regardless of the AUTOWAIT DISABLE bit.

Bit 3 **Reserved.** Read data is indeterminate.

Bit 2 **MEMORY DISABLE.**

This bit enables or disables the internal program memory (memory addresses affected are device dependent — refer to Table 3–6 on page 3-24). This bit does not affect data EEPROM or internal RAM. A reset initializes this bit to the state of the MC pin. Changes to this bit can occur only in the privilege mode.

0 = Enables internal program memory and accesses internal memory locations. The $\overline{\text{EDS}}$ memory signal will not appear during accesses to internal ROM/EPROM programming memory and 1020h–102Fh. These ranges are accessed as on-chip memory.
1 = Disables internal program memory and accesses external memory (see Table 3–6 on page 3-24 for the external memory locations). An operation on these locations generates an external memory bus cycle with the $\overline{\text{EDS}}$ memory signal validating the access. This bit disables the program EPROM control register, EPCTL (described in Section 6.4, on page 6-10), if applicable, and disables 1020h–102Fh. These ranges are accessed as off-chip memory.

Bits 1–0 **Reserved.** Read data is indeterminate.

4.3.3 System Control and Configuration Register 2 (SCCR2)

System Control and Configuration Register 2 (SCCR2)
[Memory Address 1012h]

Bit #	7	6	5	4	3	2	1	0
P012	HALT/ STANDBY	PWRDWN/ IDLE	—	BUS STEST	CPU STEST	—	INT1 NMI	PRIVILEGE DISABLE
	RP-0	RP-0		RP-0	RP-1		RP-0	RS-0

R = Read, P= Privilege write only, S = Set only, *-n* = Value of the bit after the register is reset

Bit 7 **HALT/STANDBY.**

The following descriptions apply only if the PWRDWN/IDLE bit is set; otherwise, the HALT/STANDBY bit has no effect. See subsection 4.2, on page 4-7 for a description of the halt and standby modes. Changes to this bit can occur only in the privilege mode.

0 = When an IDLE instruction executes, the processor enters the standby mode, which stops the program's execution and disables the SYSCLK to all nonessential peripherals. The SYSCLK to the T1 continues to run, and the timer generates an interrupt to bring the processor out of the standby mode.

1 = When an IDLE instruction executes, the processor enters the halt mode, which stops the internal oscillator and suspends the system and peripheral operations. This mode provides the lowest power consumption.

Bit 6 **PWRDWN/IDLE.** Powerdown/Idle.

This bit determines the mode entered by the CPU when an IDLE instruction is executed. Changes to this bit can occur only in the privilege mode.

0 = The processor enters an idle mode when the program executes an IDLE instruction. The processor waits at the IDLE instruction until any enabled interrupt occurs. The processor then enters the interrupt routine and returns to the instruction after the IDLE instruction. The idle is not a low-power mode.

1 = The processor enters a low-power mode when the program executes an IDLE instruction. The HALT/STANDBY bit determines the type of low-power mode.

Bit 5 **Reserved.** For TMS370CxxxA or 'CxxxB devices, read data is indeterminate and writing to this bit has no effect, and an oscillator fault generates a system reset, regardless of this bit's status.

Note: Bit 5 Operation Depends on Device

Bit 5 operates differently for TMS370Cxxx devices than for TMS370CxxxA or TMS370CxxxB devices. Refer to Section A.5, page A-5.

Bit 4 **BUS STEST.** BUS STEST bit.

This bit must be cleared (0) to ensure a proper operation.

Bit 3 **CPU STEST.** CPU STEST bit.

This bit is used only during a factory test and has no effect in normal operating modes.

Bit 2 **Reserved.** For TMS370CxxxA or 'CxxxB devices, read data is indeterminate and writing to this bit has no effect.

> **Note: Bit 2 Operation Depends on Device**
>
> Bit 2 operates differently for TMS370Cxxx devices than for TMS370CxxxA or TMS370CxxxB devices. Refer to Section A.5, page A-5.

Bit 1 **INT1 NMI.** Interrupt 1, Nonmaskable Interrupt.

This bit determines whether interrupt 1 is maskable or nonmaskable. When interrupt 1 is nonmaskable, it is the second highest priority interrupt (reset is highest) and is unaffected by the interrupt mask described in subsection 5.1.2, on page 5-8. The NMI mode disables the enable and priority select bits of the interrupt 1 control register. The program can change this bit only in the privilege mode.

When programming an EEPROM, you must ensure that nonmaskable interrupt routines do not access the EEPROM between an EEPROM write instruction and the point when the EXE bit (DEECTL.0) is set to 1, or data will be corrupted.

0 = Interrupt 1 is maskable.
1 = Interrupt 1 is nonmaskable (NMI).

Bit 0 **PRIVILEGE DISABLE.** Privilege Mode Disable.

Many bits controlling the system configuration can be changed only while in the privilege mode. After setting the system configuration bits, write a 1 to the PRIVILEGE DISABLE bit to disable the privilege mode and lock out any changes to the privilege protected bits. Only a reset can clear this bit.

0 = System is operating in privilege mode.
1 = System is not operating in privilege mode.

4.4 Digital I/O Configuration

On TMS370 devices, the power, reset, MC, and crystal pins are dedicated to one function. The remaining pins can be programmed to be general-purpose input and/or output pins, or special function pins. Some of these pins are associated with the functions of the peripheral modules. The pins are briefly described below and are summarized in Table 4–4.

- On TMS370C**x0x** devices, 13 of a possible 22 I/O pins are dedicated to ports A and D. Port A contains eight pins, and port D contains five pins.
- On TMS370C**x1x** devices, 13 of a possible 22 I/O pins are dedicated to ports A and D. Port A contains eight pins, and port D contains five pins.
- On TMS370C**x2x** devices, 22 of a possible 34 I/O pins are dedicated to ports A, B, C, and D. Ports A and B each have eight pins. Port C contains one pin, and port D contains five pins.
- On TMS370C**x32** devices, 12 of a possible 23 I/O pins are dedicated to ports A and D. Port A contains eight pins. Port D contains four pins.
- On TMS370C**x36** devices, 13 of a possible 25 I/O pins are dedicated to ports A and D. Port A contains eight pins. Port D contains five pins.
- On TMS370C**x4x** devices:
  - On 40-pin devices, 16 of a possible 32 I/O pins are dedicated to ports A, B, and D. Port A contains eight pins, port B contains three pins, and port D contains five pins.
  - On 44-pin devices, 16 of a possible 36 I/O pins are dedicated to ports A, B, and D. Port A contains eight pins, port B contains three pins, and port D contains five pins.
- On TMS370C**x5x** devices:
  - On 68-pin devices, 32 of a possible 55 I/O pins are dedicated to ports A, B, C, and D; each port has eight pins.
  - On 64-pin devices, 30 of a possible 53 I/O pins are dedicated to ports A, B, C, and D; ports A, B, and C each have eight pins, and port D has six.
- On TMS370C**x6x** devices, 29 of a possible 55 I/O pins are dedicated to ports A, B, C, and D; ports A, B, and C each have eight pins, and port D has five pins.

- On TMS370C**x7x** devices:
  - On 68-pin devices, 38 of a possible 55 I/O pins are dedicated to ports A, B, C, D, and G. Ports A, B, C, and D each have eight pins, and port D has six pins.
  - On 64-pin devices, 36 of a possible 53 I/O pins are dedicated to ports A, B, C, D, and G. Ports A, B, and C each have eight pins, and ports D and G each have six pins.
- On TMS370C**x8x** devices:
  - On 44-pin devices, 29 of a possible 35 I/O pins are dedicated to ports A, B, C, and D; ports A, B, and C each have eight pins, and port D has five pins.
  - On 40-pin devices, 27 of a possible 33 I/O pins are dedicated to ports A, B, C, and D; ports A and B each have eight pins, port C has six pins, and port D has five pins.
- On TMS370C**x9x** devices, 13 of a possible 25 I/O pins are dedicated to ports A and D. Port A contains eight pins, and port D contains five pins.
- On TMS370C**xAx** devices, 22 of a possible 34 I/O pins are dedicated to ports A, B, C, and D; ports A and B each have eight pins, port C has one pin, and port D has five pins.
- On TMS370C**xBx** devices:
  - On 68-pin devices, 41 of a possible 55 I/O pins are dedicated to ports A, B, C, D, G, and H; ports A, B, C, D, and G each have eight pins, and port H has one pin.
  - On 64-pin devices, 39 of a possible 53 I/O pins are dedicated to ports A, B, C, D, G, and H; ports A, B, C, and G each have eight pins, port D has six pins, and port H has one pin.
- On TMS370C**xCx** devices, 12 of a possible 22 I/O pins are dedicated to ports A and D. Port A contains eight pins, and port D contains four pins.

Frames 2 and 3 of the peripheral file (memory addresses 1020h to103Bh) contain the control registers for reading, writing, and configuring ports A, B, C, D, G, and H. These registers are shown in Figure 4–4 on page 4-21.

Table 4–4. Digital I/O Pins by Device Family

	Maximum Digital General Purpose I/O	
Device Family	**Bidirectional**	**Input Only**
TMS370Cx0x	21	1
TMS370Cx1x	21	1
TMS370Cx2x	33	1
TMS370Cx32	14	9
TMS370Cx36	16	9
TMS370Cx4x	27	9/5†
TMS370Cx5x	46/44‡	9
TMS370Cx6x	46	9
TMS370Cx7x	46/44‡	9
TMS370Cx8x	34/32§	1
TMS370Cx9x	16	9
TMS370CxAx	33	1
TMS370CxBx	46/44‡	9
TMS370CxCx	17	5

† 5 input pins for 40-pin devices; 9 input pins for 44-pin devices
‡ 46 bidirectional pins for 68-pin devices; 44 bidirectional pins for 64-pin devices
§ 34 bidirectional pins for 44-pin devices; 32 bidirectional pins for 40-pin devices

Figure 4–4. Peripheral File Frames 2 and 3: Digital Port Control Registers

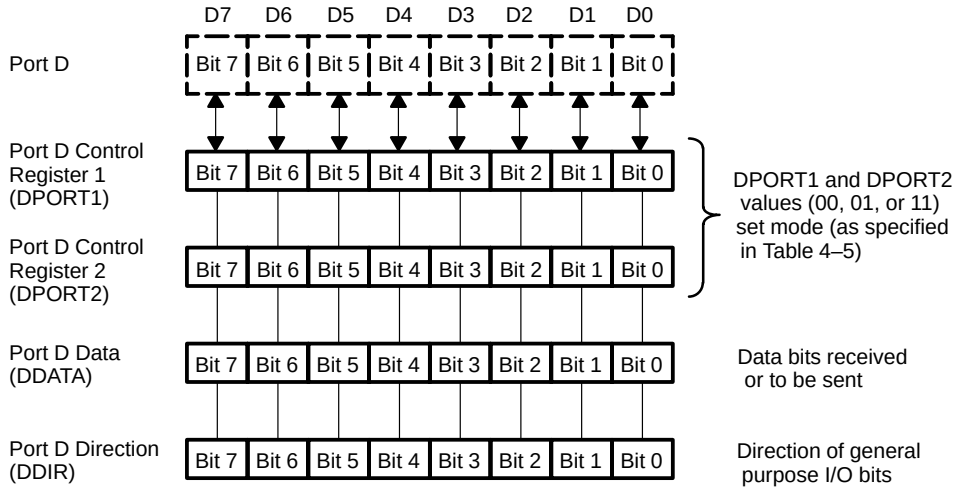
Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
APORT1	1020h	P020	Reserved							
APORT2	1021h	P021	Port A Control Register 2							
ADATA	1022h	P022	Port A Data							
ADIR	1023h	P023	Port A Direction							
BPORT1	1024h	P024	Reserved							
BPORT2	1025h	P025	Port B Control Register 2							
BDATA	1026h	P026	Port B Data							
BDIR	1027h	P027	Port B Direction							
CPORT1	1028h	P028	Reserved							
CPORT2	1029h	P029	Port C Control Register 2							
CDATA	102Ah	P02A	Port C Data							
CDIR	102Bh	P02B	Port C Direction							
DPORT1	102Ch	P02C	Port D Control Register 1							
DPORT2	102Dh	P02D	Port D Control Register 2							
DDATA	102Eh	P02E	Port D Data							
DDIR	102Fh	P02F	Port D Direction							
	1030h to 1035h	P030 to P035	Reserved							
GDATA	1036h	P036	Port G Data							
GDIR	1037h	P037	Port G Direction							
	1038h	P038	Reserved							
	1039h	P039	Reserved							
HDATA	103Ah	P03A	Port H Data							
HDIR	103Bh	P03B	Port H Direction							

Each port has as many as four associated control registers, as follows:

- ❏ Port x control register 1 (xPORT1) — found only as DPORT1 in port D)
- ❏ Port x control register 2 (xPORT2) — found in APORT2, BPORT2, CPORT2, and DPORT2
- ❏ Port x data (xDATA) — found in ADATA, BDATA, CDATA, DDATA, GDATA, and HDATA
- ❏ Port x direction (xDIR) — found in ADIR, BDIR, CDIR, DDIR, GDIR, and HDIR

The same bit position of each of these registers affects the corresponding bit in the port. For example, bit 0 of registers DPORT1, DPORT2, DDATA, and DDIR control port D, bit 0, as illustrated in Figure 4–5.

Figure 4–5. Typical Control-Register Operation Using Port D



Bits from the DPORT1 and xPORT2 registers determine the function of the corresponding port pin to be an I/O, data, address, or control signal, depending on the port. The same bit from the xDIR register determines the direction (input or output) if the pin has been defined as an I/O pin. The same bit from the xDATA register is the bit to write to or read from if the pin has been defined as an I/O pin.

Table 4–5 shows the function that each pin can serve, depending on which port contains the pin. Memory expansion signals for function A and function B are defined in Table 4–6 on page 4-24.

Table 4–5. Port Configuration Registers Setup

		When $\overline{\text{RESET}}$ Goes High				
		MC Pin Low				MC Pin High
		General-Purpose I/O†		Microcomputer Mode†		
Port	Pin	‡DPORT1 =0 xPORT2 = 0 xDATA = Data In xDIR = 0 = Input Data In Mode☆	‡DPORT1 = 0 xPORT2 = 0 xDATA = Data Out xDIR = 1 = Output Data Out Mode□	‡DPORT1 = 0 xPORT2 = 1 xDATA (not used) xDIR (not used) Function A◊	‡DPORT1 = 1 xPORT2 = 1 xDATA (not used) xDIR (not used) Function B◊	Micro-processor Mode
A	0–7	Data In = y	Data Out = q	DATA BUS	DATA BUS	DATA BUS
B	0–7	Data In = y	Data Out = q	LOW ADDR	LOW ADDR	LOW ADDR
C	0–7	Data In = y	Data Out = q	HI ADDR	HI ADDR	HI ADDR
D	0	Data In = y	Data Out = q	$\overline{\text{CSE2}}$	$\overline{\text{OCF}}$	$\overline{\text{OCF}}$
D	1	Data In = y	Data Out = q	$\overline{\text{CSH3}}$	§	¶
D	2	Data In = y	Data Out = q	$\overline{\text{CSH2}}$	§	¶
D	3	Data In = y	Data Out = q	SYSCLK	SYSCLK	SYSCLK
D	4	Data In = y	Data Out = q	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$
D	5	Data In = y	Data Out = q	$\overline{\text{CSPF}}$	§	¶
D	6	Data In = y	Data Out = q	$\overline{\text{CSH1}}$	$\overline{\text{EDS}}$	$\overline{\text{EDS}}$
D	7	Data In = y	Data Out = q	$\overline{\text{CSE1}}$	$\overline{\text{WAIT}}$	$\overline{\text{WAIT}}$
G	0–7	Data In = y	Data Out = q	§	§	¶
H	0	Data In = y	Data Out = q	§	§	¶

† Registers DPORT1 and xPORT2 determine whether the port is configured as I/O, data bus, address bus, or control signal. If DPORT1 = 1 and xPORT2 = 0, the function is not valid. The variable x represents port A, B, C, D, G, and H.
‡ DPORT1 exists for port D only.
§ These pins can be configured only as general-purpose I/O.
¶ Pins D1, D2, D5, G0 – G7, and H0 are not available in microprocessor mode.
# Ports vary for each device. See the applicable device pin descriptions in Chapter 2 for ports available on each device.
◊ Function A and B signals are defined in Table 4–6 on page 4-24.
☆ y is the value read from the xDATA register.
□ q is the value written to the xDATA register.

Table 4–6. Function A and B Signal Definitions

Signal	Definition
$\overline{\text{CSE1}}$	Chip-select eighth 1. This signal selects a bank of external memory. It has the same timing as $\overline{\text{EDS}}$. Setting this pin to a high-level general-purpose output disables the external memory bank connected to $\overline{\text{CSE1}}$. Table 4–7 lists $\overline{\text{CSE1}}$ external memory accesses for the entire line of TMS370 devices.
$\overline{\text{CSE2}}$	Chip-select eighth 2. This signal selects a bank of external memory. It has the same timing as $\overline{\text{EDS}}$. Setting this pin to a high-level general-purpose output disables the external memory bank connected to $\overline{\text{CSE2}}$. Table 4–7 lists $\overline{\text{CSE2}}$ external memory accesses for the entire line of TMS370 devices.
$\overline{\text{CSPF}}$	Chip-select peripheral file. This signal has the same timing as $\overline{\text{EDS}}$, but it goes active only during an access to the external frames of the peripheral file (locations 10C0h–10FFh).
$\overline{\text{CSH1}}$	Chip-select half 1. This signal selects a bank of external memory. It has the same timing as $\overline{\text{EDS}}$. Setting this pin to a high-level general-purpose output disables the external memory bank connected to $\overline{\text{CSH1}}$. Table 4–7 lists $\overline{\text{CSH1}}$ external memory accesses for the entire line of TMS370 devices.
$\overline{\text{CSH2}}$	Chip-select half 2. This signal selects a bank of external memory. It has the same timing as $\overline{\text{EDS}}$. Setting this pin to a high-level general-purpose output disables the external memory bank connected to $\overline{\text{CSH2}}$. Table 4–7 lists $\overline{\text{CSH2}}$ external memory accesses for the entire line of 68-pin TMS370 devices.
$\overline{\text{CSH3}}$	Chip-select half 3. This signal selects a bank of external memory. It has the same timing as $\overline{\text{EDS}}$. Setting this pin to a high-level general-purpose output disables the external memory bank connected to $\overline{\text{CSH3}}$. Table 4–7 lists $\overline{\text{CSH3}}$ external memory accesses for the entire line of TMS370 devices.
SYSCLK	System clock. This signal synchronizes external peripherals. It outputs the SYSCLK signal.
DATA BUS	External data bus. Input and output.
$\overline{\text{EDS}}$	External data strobe. This signal goes low during external memory operations. The rising edge of $\overline{\text{EDS}}$ validates the read input data; the write data is available after the falling edge of $\overline{\text{EDS}}$. Table 4–8 lists $\overline{\text{EDS}}$ external memory accesses for the entire line of TMS370 devices.
LOW ADDR/ HI ADDR	External memory address bus. Output only.
R/$\overline{\text{W}}$	Read or write operation. Goes high at the beginning of read operations and low during write operations. This line is active during both internal and external accesses.

Table 4–6. Function A and B Signal Definitions (Continued)

Signal	Definition
$\overline{\text{OCF}}$	Opcode fetch. Goes low at the beginning of a memory read operation that fetches the first byte of an instruction. It then resumes its high level at the end of the opcode fetch(es).
$\overline{\text{WAIT}}$	Wait input. An external, low signal applied to this pin, when sampled, causes the processor to hold the information on the expansion bus for one or more extra clockout cycles. This pin is sampled during the rising edge of SYSCLK after $\overline{\text{EDS}}$ goes active.

Table 4–7. TMS370 Family and Internal Program Memory (64- and 68-pin packages)

	Microcomputer-Mode Function-A Chip-Select Signals (64- and 68-pin devices)						
Signal	**'x50† (4K)**	**'x52 (8K)**	**'x53 (12K)**	**'x56† (16K)**	**'x67 (24K)**	**'x58/'x68 (32K)**	**'x59/'x69 (48K)**
$\overline{\text{CSE1}}$	2000h–3FFFh	2000h–3FFFh	2000h–3FFFh	2000h–3FFFh	A000h–BFFFh	A000h–BFFFh	E000h–EFFFh
$\overline{\text{CSE2}}$	2000h–3FFFh	2000h–3FFFh	2000h–3FFFh	2000h–3FFFh	A000h–BFFFh	A000h–BFFFh	E000h–EFFFh
$\overline{\text{CSH1}}$	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	C000h–FFFFh	C000h–FFFFh	F000h–FFFFh
$\overline{\text{CSH2}}$	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	C000h–FFFFh	C000h–FFFFh	F000h–FFFFh
$\overline{\text{CSH3}}$	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	8000h–FFFFh	C000h–FFFFh	C000h–FFFFh	F000h–FFFFh

† The TMS370C150, TMS370C156, TMS370C250, and TMS370C256 devices are ROMless. These devices cannot operate in microcomputer mode function A.

Table 4–8. TMS370 Family $\overline{EDS}$ External Memory Accesses

TMS370 Device	Microcomputer Mode, Function B $\overline{EDS}$	Microprocessor Mode With Internal Program Memory‡		Microcprocessor Mode Without Internal Memory $\overline{EDS}$
		Bit SCCR1.2 = 0 $\overline{EDS}$	Bit SCCR1.2 = 1 $\overline{EDS}$	
TMS370Cx50†	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh 2000h–6FFFh 8000h–FFFFh	10C0h–10FFh 2000h–FFFFh	—
TMS370Cx52	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh 2000h–5FFFh 8000h–FFFFh	10C0h–10FFh 2000h–FFFFh	—
TMS370Cx56†	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh 2000h–FFFFh	—
TMS370Cx67	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh 8000h–FFFFh	10C0h–10FFh 2000h–FFFFh	—
TMS370Cx58 TMS370Cx68	10C0h–10FFh 2000h–3FFFh 8000h–FFFFh	10C0h–10FFh A000h–5FFFh	10C0h–10FFh 2000h–FFFFh	—
TMS370C150 TMS370C156 TMS370C250 TMS370C256	—	—	—	1020h–102Fh 10C0h–10FFh 2000h–FFFFh

† The TMS370C150, TMS370C250, TMS370C156, and TMS370C256 devices are ROMless. These devices cannot operate in microcomputer mode function B or microprocessor mode with internal program memory.

‡ When clearing the MEMORY DISABLE bit (SCCR1.2 = 0) in the microprocessor mode with internal program memory, the $\overline{EDS}$ signal cannot access off-chip memory that has the same locations as internal program memory.

4.4.1 Configuration Example

The predecoded chip selects ($\overline{\text{CSE1}}$, $\overline{\text{CSE2}}$, and $\overline{\text{CSH1}}$ to $\overline{\text{CSH3}}$) allow the TMS370 to access external addresses with a minimum of external logic. In many cases, no external logic is necessary between the TMS370 and the peripheral device, because of the predecoded chip selects, autowait features, and the nonmultiplexed bus. Chip selects also make it easy to do memory bank selection. Without bank selection, the $\overline{\text{CSH1}}$, $\overline{\text{CSE1}}$, and $\overline{\text{CSPF}}$ signals can easily access about 40K bytes of memory in the three different areas. With bank selection, the processor can access as many as 112K bytes of memory.

To illustrate configuring the digital ports, assume that a TMS370C050 is to operate in an expanded microcomputer mode, and that 2K bytes of memory are needed at 2000h to 27FFh. The top half of Example 4–1 shows the desired port configuration.

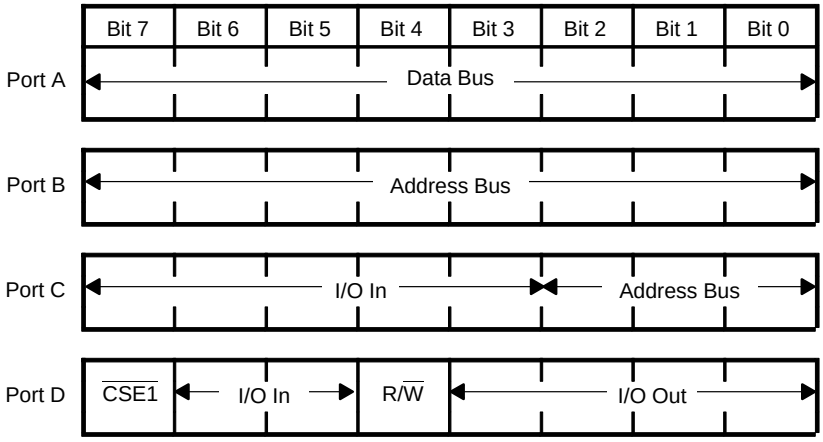
- Port A is set as the external data bus.
- Port B contains the low-order address bits of the 11 bits necessary to access 2K bytes of memory.
- Bits 4 through 7 of port C are set as I/O input.
- In port D, bit 7 is the chip-select signal to access 2000h to 3FFFh, and bit 4 is for external memory control signal R/$\overline{\text{W}}$. The remaining bits of port D are used as I/O output.

The bottom half of Example 4–1 shows the port control registers set up to establish the configuration shown in the top half. To determine the bits needed to set the registers, use Table 4–5 on page 4-23. For example, to set port A as the data bus in microcomputer mode, function A, find Port A at the left-hand column of Table 4–5. Look across the row to find the data bus, then follow the column up to the bit settings for the desired mode (microcomputer or microprocessor) and function in the column heading:

DPORT1 = 0
xPORT2 = 1
xDATA (not used)
xDIR (not used)

The assembly language instructions in the right column of Example 4–1 show one method of setting up the registers to the left. The Pxxx operand indicates peripheral file access (see Chapter 16, *Assembly Language Instruction Set,* for more information on peripheral file instructions).

Example 4–1. Digital Port Setup



Port A Control Registers

APORT2	1021h	1	1	1	1	1	1	1	1	MOV #0FFh, P021
ADATA	1022h	dc	dc	dc	dc	dc	dc	dc	dc	
ADIR	1023h	dc	dc	dc	dc	dc	dc	dc	dc	

Port B Control Registers

BPORT2	1025h	1	1	1	1	1	1	1	1	MOV #0FFh, P025
BDATA	1026h	dc	dc	dc	dc	dc	dc	dc	dc	
BDIR	1027h	dc	dc	dc	dc	dc	dc	dc	dc	

Port C Control Registers

CPORT2	1029h	0	0	0	0	0	1	1	1	MOV #007h, P029
CDATA	102Ah	y	y	y	y	y	dc	dc	dc	
CDIR	102Bh	0	0	0	0	0	dc	dc	dc	MOV #0, P02B

Port D Control Registers

DPORT1	102Ch	0	0	0	0	0	0	0	0	MOV #0, P02C
DPORT2	102Dh	1	0	0	1	0	0	0	0	MOV #090h, P02D
DDATA	102Eh	dc	y	y	dc	q	q	q	q	
DDIR	102Fh	dc	0	0	dc	1	1	1	1	MOV #00Fh, P02F

Legend: dc = don't care
y = data value read
q = data value written

When the device operates with internal program memory disabled, any access to the port peripheral frame, 1020h–102Fh, is decoded as an external

address. Memory accesses to this frame can control external hardware that emulates the digital I/O functions.

4.4.2 Microprocessor Mode

Initializing a device with bus expansion to the microprocessor mode forces ports A, B, C, and D to function B as shown in Table 4–5, page 4-23. Port A is the data bus, port B is the low-order address bus, and port C is the high-order address bus in this mode. Devices that are not defined for operation in a memory expansion mode must be powered up in the microcomputer single-chip mode.

4.4.3 Microcomputer Mode

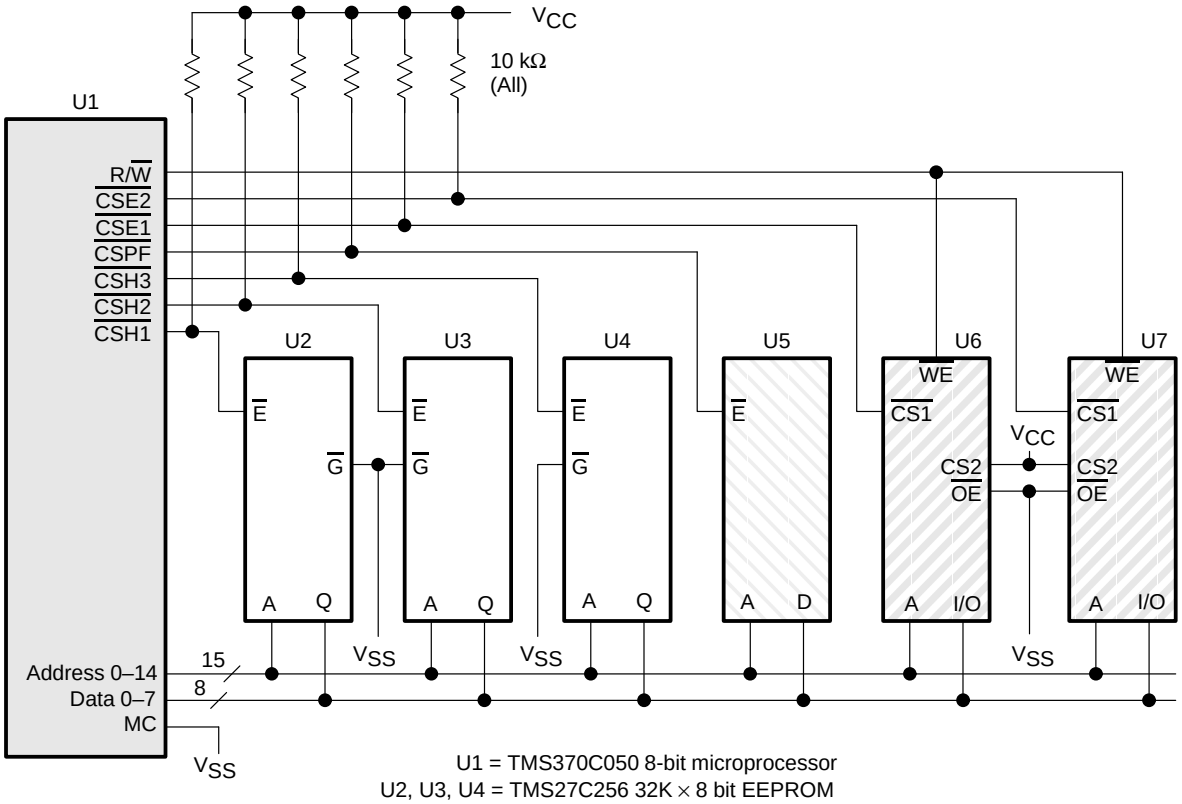
Initializing the device to the microcomputer mode forces ports A, B, C, and D to general-purpose high-impedance inputs. The program can set the control bits to change the function of the port pins to one of four functions: general-purpose output, general-purpose input, function A, or function B.

When you change a pin from a general-purpose input pin to an output pin, write to the data register first to set up the data; then, set the data direction register. This prevents unknown data on the pin from interfering with the external circuitry.

The TMS370 in the microcomputer mode can individually reconfigure any address, data, or control signal to use only the necessary signals and leave the other signals on the port for general-purpose I/O operations.

Figure 4–6 shows an example of the TMS370C050 interfaced to 112K bytes of external memory (see the *TMS370 8-bit Microcontroller Applications Manual* [SPNA019] for detailed examples). The function A chip-select signals enable one of three banks of EPROM, an external peripheral device, and one of two banks of static RAM. In this example, all eight bits of port A are the data bus, all eight bits of port B are the address LSbyte, and seven port C bits complete the 15-bit address bus.

Figure 4–6. System Interface Example



U1 = TMS370C050 8-bit microprocessor
U2, U3, U4 = TMS27C256 32K × 8 bit EEPROM
U5 = Unspecified 64-byte peripheral
U6, U7 = 8K × 8 bit static RAM

Chapter 5

Interrupts and System Reset

This chapter discusses the internal and external interrupts of the TMS370. Device reset methods are also discussed. This chapter covers the following topics:

Topic **Page**

5.1 Interrupts

The TMS370 programmable interrupt structure allows flexible on-chip and external interrupt configurations to meet real-time interrupt-driven application requirements.

Whenever an internal or external circuit requests an enabled interrupt, the processor finishes the current instruction and then fetches the address of the appropriate interrupt service routine from the interrupt table. The processor then pushes the contents of the program counter and status register onto the stack and begins execution at the interrupt service routine address found in the interrupt table. When the interrupt service routine completes its execution, the program executes an RTI (return from interrupt) instruction, which pops the previous status register and program counter contents from the stack. The processor resumes execution from the point of interruption.

Table 5–1 shows the interrupt and reset vectors by device category.

Table 5–1. Interrupts and Reset Vectors

Device Category	Interrupts/Reset		
	External†	Vectors Total	Sources Total
TMS370Cx0x	4	7	15
TMS370Cx1x	4	6	13
TMS370Cx2x	4	8	16
TMS370Cx32	4	23	25
TMS370Cx36	2	22	24
TMS370Cx4x	4	9	22
TMS370Cx5x	4	10	23
TMS370Cx6x	4	11	29
TMS370Cx7x	4	7	19
TMS370Cx8x	4	5	12
TMS370Cx9x	2	4	11
TMS370CxAx	4	8	21
TMS370CxBx	4	6	13
TMS370CxCx	2	6	14

† Three external interrputs and a reset for all devices except ’x36, ’x9x, and ’xCx which have one external interrupt and a reset.

5.1.1 Interrupt Operation

The hardware interrupt structure includes two selectable priority levels as shown in Figure 5–1. Interrupt level 1 has a higher priority than interrupt level 2. The two priority levels can be independently masked by clearing the global interrupt enable bits (IE1 and IE2) of the status register (described in subsection 3.2.2, page 3-5.

Note:

All the peripheral modules are connected in a daisy-chain order, shown running horizontally in the center of Figure 5–1. For the same level (1 or 2), the module connected closest to the CPU (right side of figure) has a higher priority than the modules that are further away. For example, the order of priority shown in the figure is INT1, followed by INT2, INT3, SPI, TIMER 1, SCI, TIMER 2A, PACT, ADC, and TIMER 2B. The devices and their modules are listed in Table 5–2 on page 5-5.

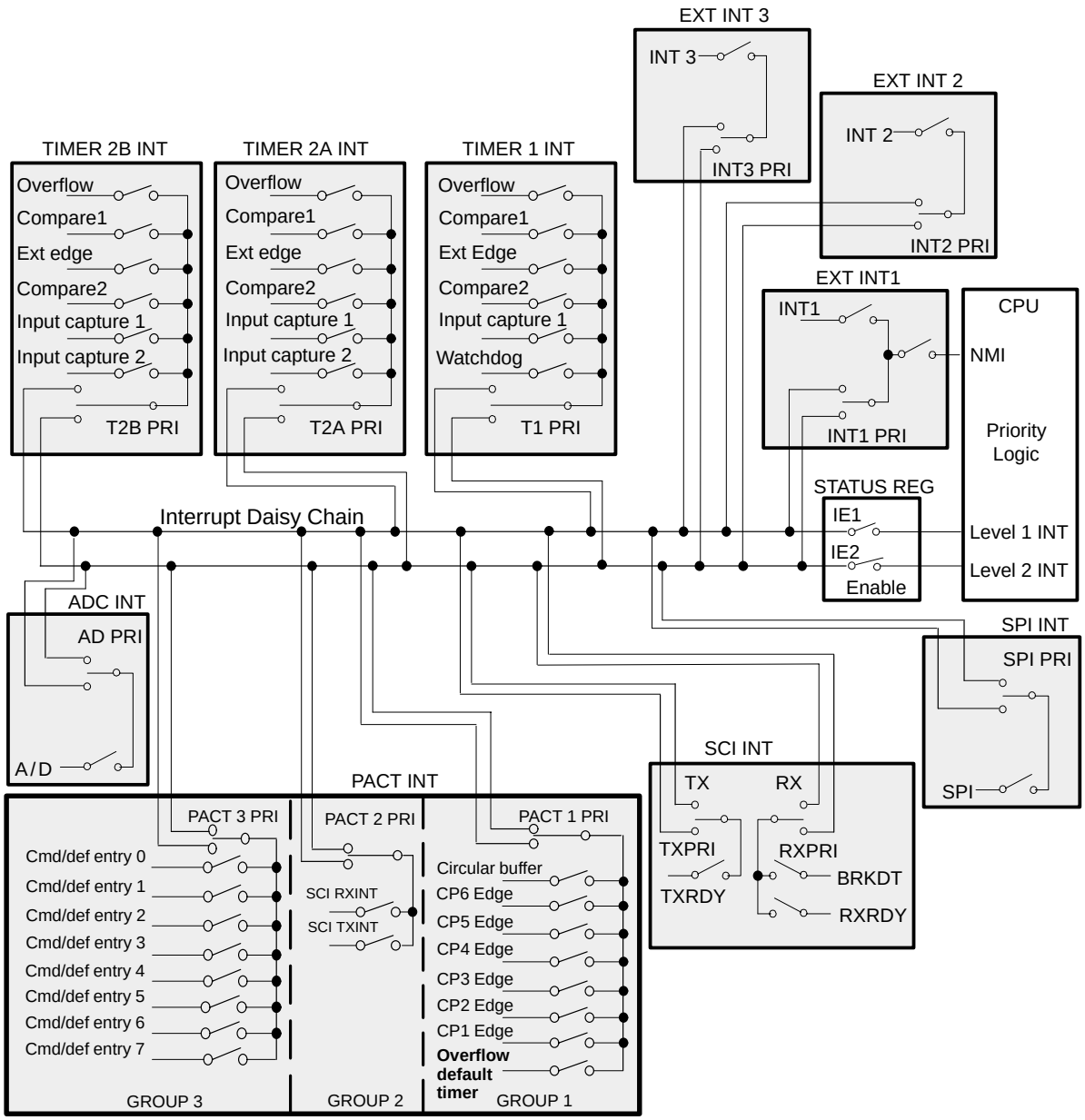
During system initialization, the application program can assign system interrupts to either the high or low priority level. The program can reassign priority levels at any time, except for those priority levels that are protected by the privileged mode. Within each level, the hardware determines the interrupt priority.

The processor services the pending interrupts after executing the current instructions, depending on the interrupt mask and priority conditions. The processor services all enabled level 1 interrupts before servicing any level 2 interrupts. Within each level, the processor services the highest priority interrupts first. Table 5–2 (page 5-5) shows the hardware priorities (starting at 1 for highest) of the devices at the time of this printing.

The TMS370 architecture allows up to 128 independent interrupt vectors, located within the memory addresses 7F00h to 7FFFh. This memory space also contains the trap tables, and 12 bytes are reserved for Texas Instruments' use. If the device does not define this memory for an interrupt vector, it can be used for program memory. Table 5–3 shows the interrupt vector source(s) and corresponding address(es). Note that a system interrupt can have multiple interrupt sources.

The application program can individually enable or disable all of the interrupt sources via local interrupt enable control bits in the associated peripheral file. Also, the software can read each interrupt source's flag bit to determine which interrupt source generated the system interrupt.

Figure 5–1. Interrupt Control



Note: Figure 5–1 shows the daisy chain order of all peripheral modules. The modules connected closest to the CPU have a higher priority than those further away. For instance, the following are in priority order as shown above: INT1, INT2, INT3, SPI, TIMER 1, SCI (SCI1 or SCI2), TIMER 2A, PACT, ADC (ADC1, ADC2, or ADC3), TIMER 2B. Not all peripheral modules are available on any given device. Refer to Table 5–2 for device module availability and priority for each family.

The processor acknowledges an interrupt if its flag bit equals 1 and the interrupt is enabled. To avoid immediately re-entering the same interrupt service routine, the interrupt service routine must clear all appropriate flag bits before leaving the routine. For example, clear the INT1 flag bit as shown in Figure 5–2 on page 5-9.

Table 5–2. Module Interrupt Priority in Lowest-to-Highest Order

Vector Start Address	Module†		Device Module Priority														Module Vector Bytes
			x0x	x1x	x2x	x32	x36	x4x	x5x	x6x	x7x	x8x	x9x	xAx	xBx	xCx	
7FBEh	Timer 2B		NA	NA	NA	NA	NA	NA	NA	11	NA	NA	NA	NA	NA	NA	2
7FECh	ADC‖		NA	NA	NA	8	7	9	10	10	7	NA	4	NA	6	6	2
7F9Ch	P§	Grp3	NA	NA	NA	7‡	6‡	NA	NA	NA	NA	NA	NA	NA	NA	NA	36
		Grp2	NA	NA	NA	6‡	5‡	NA	NA	NA	NA	NA	NA	NA	NA	NA	
		Grp1	NA	NA	NA	5‡	4‡	NA	NA	NA	NA	NA	NA	NA	NA	NA	
7FEEh	Timer 2A		NA	NA	NA	NA	NA	8	9	9	6	NA	NA	8	NA	NA	2
7FF0h	SCI☆	TX¶	7	NA	8	NA	NA	7	8	8	NA	NA	NA	7	NA	5	4
		RX#	6	NA	7	NA	NA	6	7	7	NA	NA	NA	6	NA	4	
7FF4h	Timer 1		5	6	6	NA	NA	5	6	6	5	5	3	5	5	3	2
7FF6h	SPI		NA	5	5	NA	3	NA	5	5	NA	NA	NA	NA	NA	NA	2
7FF8h	External INT3		4	4	4	4	NA	4	4	4	4	4	NA	4	4	NA	2
7FFAh	External INT2		3	3	3	3	NA	3	3	3	3	3	NA	3	3	NA	2
7FFCh	External INT1		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
7FFEh	RESET		1	1	1	1	1	1	1	1	1	1	1	1	1	1	2

† Modules are listed in order of lowest priority at the top, highest priority (1) on the bottom.
‡ The in-circuit emulator with PACT (TMS370Cx32 and TMS370x36) has the PACT module as the lowest priority interrupt, while the rest of the device family has (depending on the resident modules) timer 2B as the lowest priority, followed by the ADC, timer 2A, and so forth. The priority interrupts are shown in the table from the lowest to the highest priority.
§ PACT module
¶ Transmit
# Receive
‖ ADC refers to ADC1, ADC2, or ADC3
☆ SCI refers to SCI1 or SCI2

Table 5–3. Interrupt Vector Sources

Module	Vector Address	Interrupt Source	Interrupt Flag	System Interrupt	Priority in Group†
Timer 2B	7FBEh, 7FBFh	Timer 2B Overflow	T2B OVRFL INT FLAG	T2BINT	1
		Timer 2B compare 1	T2BC1 INT FLAG		
		Timer 2B compare 2	T2BC2 INT FLAG		
		Timer 2B external edge	T2BEDGE INT FLAG		
		Timer 2B input capture 1	T2BIC1 INT FLAG		
		Timer 2B input capture 2	T2BIC2 INT FLAG		
ADC‡	7FECh, 7FEDh	ADC conversion complete	AD INT FLAG	ADINT	1
PACT (Group 3)	7FA0h, 7FA1h	PACT cmd/def entry 0	CMD/DEF INT 0 FLAG	CDINT0	1
	7FA2h, 7FA3h	PACT cmd/def entry 1	CMD/DEF INT 1 FLAG	CDINT1	2
	7FA4h, 7FA5h	PACT cmd/def entry 2	CMD/DEF INT 2 FLAG	CDINT2	3
	7FA6h, 7FA7h	PACT cmd/def entry 3	CMD/DEF INT 3 FLAG	CDINT3	4
	7FA8h, 7FA9h	PACT cmd/def entry 4	CMD/DEF INT 4 FLAG	CDINT4	5
	7FAAh, 7FABh	PACT cmd/def entry 5	CMD/DEF INT 5 FLAG	CDINT5	6
	7FACh, 7FADh	PACT cmd/def entry 6	CMD/DEF INT 6 FLAG	CDINT6	7
	7FAEh, 7FAFh	PACT cmd/def entry 7	CMD/DEF INT 7 FLAG	CDINT7	8
PACT (Group 2)	7F9Eh, 7F9Fh	PACT SCI RXINT	PACT RXRDY	PRXINT	1
	7F9Ch, 7F9Dh	PACT SCI TXINT	PACT TXRDY	PTXINT	2
PACT (Group 1)	7FB0h, 7FB1h	PACT circular buffer (half/full)	BUFFER HALF/FULL INT FLAG	BUFINT	1
	7FB2h, 7FB3h	PACT CP6 edge	CP6 INT FLAG	CP6INT	2
	7FB4h, 7FB5h	PACT CP5 edge	CP5 INT FLAG	CP5INT	3
	7FB6h, 7FB7h	PACT CP4 edge	CP4 INT FLAG	CP4INT	4
	7FB8h, 7FB9h	PACT CP3 edge	CP3 INT FLAG	CP3INT	5
	7FBAh, 7FBBh	PACT CP2 edge	CP2 INT FLAG	CP2INT	6
	7FBCh, 7FBDh	PACT CP1 edge	CP1 INT FLAG	CP1INT	7
	7FBEh, 7FBFh	PACT default timer overflow	DEFTIM OVRFL INT FLAG	POVRFL INT	8

† 1 is the highest priority. The table, top to bottom, is in lowest to highest priority order.
‡ ADC refers to ADC1, ADC2, or ADC3.

Table 5–3. Interrupt Vector Sources (Continued)

Module	Vector Address	Interrupt Source	Interrupt Flag	System Interrupt	Priority in Group†
Timer 2A	7FEEh, 7FEFh	Timer 2A overflow	T2A OVRFL INT FLAG	T2AINT	1
		Timer 2A compare 1	T2AC1 INT FLAG		
		Timer 2A compare 2	T2AC2 INT FLAG		
		Timer 2A external edge	T2AEDGE INT FLAG		
		Timer 2A input capture 1	T2AIC1 INT FLAG		
		Timer 2A input capture 2	T2AIC2 INT FLAG		
SCI TX‡	7FF0h, 7FF1h	SCI TX data register empty	TXRDY FLAG	TXINT	1
SCI RX‡	7FF2h, 7FF3h	SCI RX data register full	RXRDY FLAG	RXINT	1
		SCI RX break detect	BRKDT FLAG		
Timer 1	7FF4h, 7FF5h	Timer 1 overflow	T1 OVRFL INT FLAG	T1INT	1
		Timer 1 compare 1	T1C1 INT FLAG		
		Timer 1 compare 2	T1C2 INT FLAG		
		Timer 1 external edge	T1EDGE INT FLAG		
		Timer 1 input capture 1	T1IC1 INT FLAG		
		Watchdog overflow	WD OVRFL INT FLAG		
SPI	7FF6h, 7FF7h	SPI RX/TX complete	SPI INT FLAG	SPIINT	1
External INT	7FF8h, 7FF9h	External INT3	INT3 FLAG	INT3	1
	7FFAh, 7FFBh	External INT2	INT2 FLAG	INT2	1
	7FFCh, 7FFDh	External INT1	INT1 FLAG	INT1	1
$\overline{\text{RESET}}$	7FFEh, 7FFFh	External $\overline{\text{RESET}}$	COLD START	$\overline{\text{RESET}}$	1
		Watchdog overflow	WD OVRFL INT FLAG		
		Oscillator fault detect	OSC FLT FLAG		

† 1 is the highest priority. The table, top to bottom, is in lowest to highest priority order.
‡ SCI refers to SCI1 or SCI2.

Interrupts are sampled and arbitrated by the CPU during every opcode fetch. If one or more requests are pending (and the appropriate enable bits are set in the status register for maskable interrupts), then at the normal completion of the opcode fetch, the interrupt context switch begins. The new opcode is discarded, and the program counter is rewound to point to the discarded instruction. Moreover, at the completion of the interrupt service routine, the discarded instruction is fetched again.

The context switch routine proceeds as follows:

1) Increments the stack pointer (SP) and stores the contents of the status register (ST) at the location pointed to by the SP.

2) Resets the ST to 00h (disables further interrupt recognition).

3) Obtains the identity of the interrupting peripheral.

4) Rewinds the program counter (PC) to point to the aborted opcode.

5) Increments SP and stores the original PC high byte (PCH) at the location pointed to by the SP.

6) Gets the interrupt-service-routine address (low byte) and stores it in the PC low byte (PCL).

7) Increments SP and stores the original PCL at the location pointed to by SP.

8) Gets the address (high byte) of the interrupt service routine and stores it in the PCH.

9) Resumes instruction execution with the new PC contents.

A minimum of 15 cycles is required from the time that an interrupt is triggered until the first instruction of the interrupt service routine is read. The moment at which the interrupt is asserted, and the place in the instruction at which the interrupt is asserted, depend on the instruction in progress. The worst case happens if the interrupt occurs near the start of a divide instruction; the processor may require up to 78 clock cycles to enter the interrupt service routine. If wait states are needed, the appropriate number of cycles must be added. Also, an external interrupt (INT1, INT2, or INT3) requires two extra clock cycles to synchronize before the processor can detect it.

5.1.2 External Interrupts

External pins INT1, INT2, and INT3 allow external devices to interrupt the program and enter a specific interrupt service routine. The INT1, INT2, and INT3 control registers in peripheral file frame 1 govern the software configuration of the external interrupts. Figure 5–2 shows these registers.

Figure 5–2. Peripheral File Frame 1: External Interrupt Control Registers

Desig-nation	Address	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT1	1017h	P017	INT1 FLAG (RC–0)	INT1 PIN DATA (R–0)	—	—	—	INT1 POLARITY (RW–0)	INT1 PRIORITY (RW–0)	INT1 ENABLE (RW–0)
INT2	1018h	P018	INT2 FLAG (RC–0)	INT2 PIN DATA (R–0)	—	INT2 DATA DIR (RW–0)	INT2 DATA OUT (RW–0)	INT2 POLARITY (RW–0)	INT2 PRIORITY (RW–0)	INT2 ENABLE (RW–0)
INT3	1019h	P019	INT3 FLAG (RC–0)	INT3 PIN DATA (R–0)	—	INT3 DATA DIR (RW–0)	INT3 DATA OUT (RW–0)	INT3 POLARITY (RW–0)	INT3 PRIORITY (RW–0)	INT3 ENABLE (RW–0)

The software uses the interrupt polarity bits to individually configure each external interrupt to trigger on either a rising edge or a falling edge. If the interrupt function is not required, the software can configure INT1 to be an input pin, and INT2 and INT3 to be general-purpose input/output pins.

INT1 can be programmed as a maskable or nonmaskable interrupt. When INT1 is nonmaskable, it cannot be masked by the individual or global mask bits. Remember that the INT1 NMI bit (SCCR2.1) is protected during nonprivileged operations and should be configured during the system initialization sequence following a reset (see the INT1 NMI bit description in subsection 4.3.3, page 4-17).

The nonmaskable interrupt is used for events that you want to respond to immediately. For example, this event could be derived from monitoring the power supply and saving important data to EEPROM whenever a brownout/power loss condition occurs.

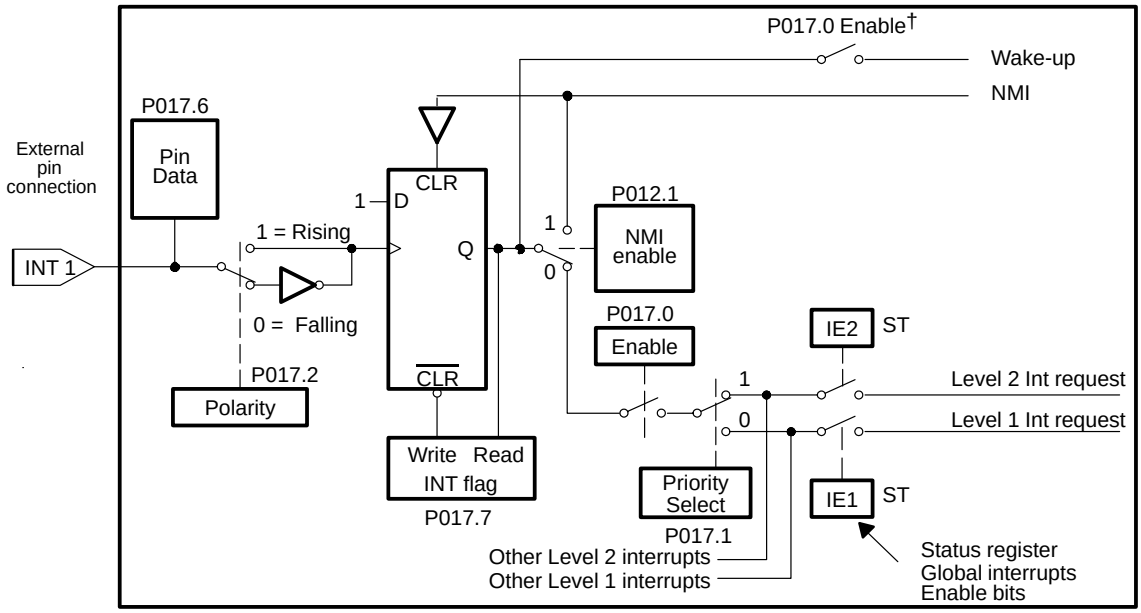
Notes:

- When a nonmaskable interrupt (NMI) is used, the interrupt is taken on every active edge of the INT1 pin. This pin should be debounced to avoid multiple interrupts that could cause the stack to overflow. Once the stack overflows, the program may not operate as expected.
- To provide a method of exiting low-power modes, INT1 is automatically enabled as an NMI during low-power modes when the hard watchdog mode is selected. This means that the NMI is always generated regardless of the interrupt enable flags and the values of the status bits (that is, INT1 PRIORITY bit [INT1.1], INT1 ENABLE bit [INT1.0], INT1 NMI bit [SCCR2.1], and the global interrupt enable flags of the status register [IE1 and IE2]). See subsection 7.7.2.1, page 7-26, for more information.

The application program must configure the following bits for each interrupt to function correctly (refer to Figure 5–3 and Figure 5–4 on page 5-10).

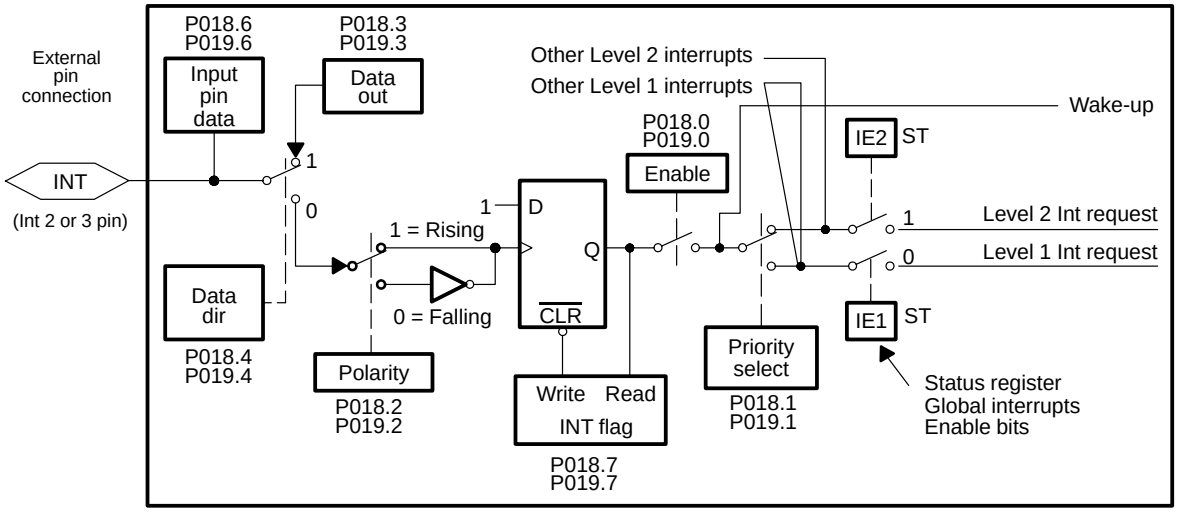
- The INTx PRIORITY bit configures the interrupt as either a level 1 or a level 2 interrupt.
- The INTx POLARITY bit selects the trigger as either a falling edge or a rising edge.
- The INTx ENABLE bit allows the request to be transmitted to the CPU if either the IE1 or IE2 enable bit, whichever is appropriate, is enabled.

Figure 5–3. Interrupt 1 Block Diagram



† This bit is ignored if you are using the hard watchdog option.

Figure 5–4. Interrupts 2 and 3 Block Diagram



- The INTx FLAG indicates that the selected edge (rising or falling) has occurred. If the enables are set, an interrupt is requested. This bit remains a 1 until the software or a RESET clears it. The INTx FLAG bit is useful for programs that poll the interrupt flag instead of generating a system interrupt.
- The INTx PIN DATA bit shows the condition presently on the interrupt pin.
- On interrupts 2 and 3, the INTx DATA DIR determines whether the pin functions as a general-purpose output pin or as an input/interrupt pin.
- If you select the general-purpose output function for a pin, then the value written by software to the INTx DATA OUT bit determines the value of the output.

All external interrupts can bring the processor out of both the halt and the standby low-power modes if the interrupt enable and the interrupt level mask are enabled. Note that in halt mode, the interrupt is detected on the level and not the edge. For further information, refer to subsection 4.2.3, page 4-10.

5.2 Interrupt Control Registers

The interrupt control registers control the configuration of the external interrupts.

5.2.1 Interrupt 1 Control Register (INT1)

The INT1 register controls the interrupt configuration for the INT1 pin. INT1 is available for all TMS370 devices.

Interrupt 1 Control Register (INT1)
[Memory Address 1017h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P017	INT1 FLAG	INT1 PIN DATA	—	—	—	INT1 POLARITY	INT1 PRIORITY	INT1 ENABLE
	RC-0	R-0				RW-0	RW-0	RW-0

R = Read, W = Write, C = Clear only, *-n* = Value of the bit after the register is reset

Bit 7 **INT1 FLAG.** Interrupt 1 Flag.

When set, this bit indicates that the selected transition on INT1 has occurred. An interrupt can occur as long as this bit remains set; as a result, the application program must clear this bit during the interrupt handling routine. This bit is set, even if the INT1 ENABLE bit is cleared. This flag is not set if INT1 is configured as an NMI.

1 = Transition detected
0 = No transition

Bit 6 **INT1 PIN DATA.** Interrupt 1 Pin Data.

This bit displays the current condition of the INT1 pin.

1 = High-level input voltage (V_{IH}) at the INT1 pin
0 = Low-level input voltage (V_{IL}) at the INT1 pin

Bits 5,4,3 **Reserved.** Read data is indeterminate.

Bit 2 **INT1 POLARITY.** Interrupt 1 Polarity.

This bit determines whether INT1 triggers on a rising edge or on a falling edge.

1 = Triggers on a rising edge (low-to-high transition)
0 = Triggers on a falling edge (high-to-low transition)

Bit 1 **INT1 PRIORITY.** Interrupt 1 Priority.

This bit determines the interrupt level of the INT1 pin—either a high, level-1 interrupt or a low, level-2 interrupt. This bit is ignored if INT1 NMI = 1.

1 = Level 2 interrupt (low level)
0 = Level 1 interrupt (high level)

Bit 0 **INT1 ENABLE.** Interrupt 1 Enable.

When set, this bit enables the interrupts for the INT1 pin. This bit is ignored if INT1 NMI = 1.

1 = Enables INT1 interrupts
0 = Disables INT1 interrupts

5.2.2 Interrupt 2 Control Register (INT2)

The INT2 register controls the interrupt configuration for the INT2 pin. INT2 is available for all TMS370 devices except ’x36, ’x9x, and ’xCx devices.

Interrupt 2 Control Register (INT2)
[Memory Address 1018h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P018	INT2 FLAG	INT2 PIN DATA	—	INT2 DATA DIR	INT2 DATA OUT	INT2 POLARITY	INT2 PRIORITY	INT2 ENABLE
	RC-0	R-0		RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, C = Clear only, *-n* = Value of the bit after the register is reset

Bit 7 **INT2 FLAG.** Interrupt 2 Flag.

This bit indicates that the selected transition on INT2 has occurred. An interrupt can occur as long as this bit remains set; as a result, the program must clear this bit during the interrupt handling routine. This bit is set, even if the INT2 ENABLE bit is cleared.

1 = Transition detected
0 = No transition

Bit 6 **INT2 PIN DATA.** Interrupt 2 Pin Data.

This bit displays the current value of the INT2 pin.

1 = High-level input voltage (V_{IH}) at the INT2 pin
0 = Low-level input voltage (V_{IL}) at the INT2 pin

Bit 5 **Reserved.** Read data is indeterminate.

Bit 4 **INT2 DATA DIR.** Interrupt 2 Data Direction.

The INT2 pin can be configured as either an output pin or as an input/interrupt pin.

1 = INT2 pin is an output pin
0 = INT2 pin is an input/interrupt pin

Bit 3 **INT2 DATA OUT.** Interrupt 2 Data Out.

If the software configures the INT2 pin as an output pin (INT2 DATA DIR = 1), then the value that the software writes to the INT2 DATA OUT bit determines the value of that output pin.

Bit 2 **INT2 POLARITY.** Interrupt 2 Polarity.

This bit determines whether INT2 triggers on a rising edge or on a falling edge.

1 = Triggers on a rising edge (low-to-high transition)
0 = Triggers on a falling edge (high-to-low transition)

Bit 1 **INT2 PRIORITY.** Interrupt 2 Priority.

This bit determines the interrupt level of the INT2 pin—either a high, level-1 interrupt or a low, level-2 interrupt.

1 = Level 2 interrupt (low level)
0 = Level 1 interrupt (high level)

Bit 0 **INT2 ENABLE.** Interrupt 2 Enable.

When set, this bit enables the interrupts for the INT2 pin.

1 = Enables INT2 interrupts
0 = Disables INT2 interrupts

5.2.3 Interrupt 3 Control Register (INT3)

The INT3 register controls the interrupt configuration for the INT3 pin. INT3 is available for all TMS370 devices except ’x36, ’x9x, and ’xCx devices.

Interrupt 3 Control Register (INT3)
[Memory Address 1019h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P019	INT3 FLAG	INT3 PIN DATA	—	INT3 DATA DIR	INT3 DATA OUT	INT3 POLARITY	INT3 PRIORITY	INT3 ENABLE
	RC-0	R-0		RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, C = Clear only, *-n* = Value of the bit after the register is reset

Bit 7 **INT3 FLAG**. Interrupt 3 Flag.

This bit indicates that the selected transition on INT3 has occurred. An interrupt can occur as long as this bit remains set; as a result, the program must clear this bit during the interrupt handling routine. This bit will be set, even if the INT3 ENABLE bit is cleared.

1 = Transition detected
0 = No transition

Bit 6 **INT3 PIN DATA**. Interrupt 3 Pin Data.

This bit displays the current condition of the INT3 pin.

1 = High-level input voltage (V_{IH}) at the INT3 pin
0 = Low-level input voltage (V_{IL}) at the INT3 pin

Bit 5 Reserved. Read data is indeterminate.

Bit 4 **INT3 DATA DIR**. Interrupt 3 Data Direction.

The INT3 pin can be configured as either an output pin or as an input/interrupt pin.

1 = INT3 pin is an output pin
0 = INT3 pin is an input/interrupt pin

Bit 3 **INT3 DATA OUT**. Interrupt 3 Data Out.

If software configures the INT3 pin as an output pin (INT3 DATA DIR=1), then the value that the software writes to the INT3 DATA OUT bit determines the value of that output pin.

Bit 2 **INT3 POLARITY**. Interrupt 3 Polarity.

This bit determines whether INT3 triggers on a rising edge or on a falling edge.

1 = Triggers on a rising edge (low-to-high transition)
0 = Triggers on a falling edge (high-to-low transition)

Bit 1 **INT3 PRIORITY**. Interrupt 3 Priority.

This bit determines the interrupt level of the INT1 pin—either a high, level-1 interrupt or a low, level-2 interrupt.

1 = Level-2 interrupt (low level)
0 = Level-1 interrupt (high level)

Bit 0 **INT3 ENABLE**. Interrupt 3 Enable.

This bit enables the interrupts for the INT3 pin.

1 = Enables INT3 interrupts
0 = Disables INT3 interrupts

5.3 Multiple Interrupt Servicing

When servicing an interrupt, the processor automatically clears the global interrupt enable bits IE1 and IE2 in the status register. This prevents all other interrupts from being recognized during the execution of the interrupt service routine. Once the service routine is completed by executing the RTI (return from interrupt) instruction, the old status register contents are popped from the stack. This returns bits IE1 and IE2 to their original conditions and allows any pending interrupts to be recognized.

An interrupt service routine allows nested interrupts by executing the EINT, EINTL, or EINTH instructions to set the global interrupt enable bits in the status register. This permits other interrupts to be recognized during the service routine execution. When a nested interrupt service routine completes, it returns to the previous interrupt service routine when the RTI instruction executes. Too many nested interrupts could overflow the stack, and cause a program failure.

5.4 Resets

The TMS370 has three possible reset sources:

- A low input to the $\overline{\text{RESET}}$ pin
- A programmable watchdog timer timeout (described in Section 7.7 and Section 15.2)
- A programmable oscillator fault failure (described in subsection 4.1.3)

After a reset, the program can interrogate the status bits (shown in Table 5–4) to determine the source of the reset in order to take appropriate action. If none of the sources shown in Table 5–4 caused the reset, then the $\overline{\text{RESET}}$ pin was pulled low by external hardware or the PACT module watchdog.

The $\overline{\text{RESET}}$ pin starts the hardware initialization and ensures an orderly software startup. The $\overline{\text{RESET}}$ pin is an input/output pin. A low-level pulse initiates the reset sequence. The processor may detect short reset pulses of a few nanoseconds, but a low level (active) of one SYSCLK cycle is necessary to guarantee that the device sees the reset signal. The microcontroller is held in reset until the $\overline{\text{RESET}}$ pin goes inactive (high). If the reset input signal remains low for less than eight system clock cycles (SYSCLK), the processor holds the external $\overline{\text{RESET}}$ pin low for eight system clock cycles to reset the external system components.

The basic operating mode (microcomputer or microprocessor) is determined by the voltage level applied to the MC pin when the $\overline{\text{RESET}}$ pin goes inactive (high). The $\overline{\text{RESET}}$ pin can be pulled low at any time during its operation to start the reset sequence.

Table 5–4. Reset Sources

Register	Address	PF	Bit No.	Control Bit	Source of Reset
SCCR0	1010h	P010	7	COLD START	Cold or warm start reset
SCCR0	1010h	P010	4	OSC FLT FLAG	Oscillator out of range
T1CTL2	104Ah	P04A	5	WD OVRFL INT FLAG	Watchdog timer timeout

The sequence of events during reset is as follows:

1) CPU registers are initialized (ST = 00h and SP = 01h).

2) Registers A and B are initialized to 00h (no other RAM is changed).

3) Contents of 7FFFh are read and stored in the PCL (PC low).

4) Contents of 7FFEh are read and stored in the PCH (PC high).

5) Program execution is started with an opcode fetch from the address pointed to by the PC.

The reset sequence takes 20 SYSCLK cycles in the microcomputer mode (22 SYSCLK cycles in the microprocessor mode) from the time the reset pulse is released until the first opcode fetch begins.

When a watchdog overflow or an oscillator fault detection circuit generates a reset, the $\overline{\text{RESET}}$ pin is pulled low so that it resets other external components in the system.

During a reset, RAM contents (except for register A and register B) remain unchanged, and the majority of the peripheral file bits are cleared to 0, with the exception of the control bits shown in Table 5–5.

5.4.1 Simple Reset Circuitry

An application must activate the $\overline{\text{RESET}}$ pin at power-up with an external input to $\overline{\text{RESET}}$ or an RC power-up circuit. The $\overline{\text{RESET}}$ pin must be held low until the clock signal is valid and V_{CC} is within operating range. Figure 5–5 shows a simple reset circuit that holds $\overline{\text{RESET}}$ low during the power-up.

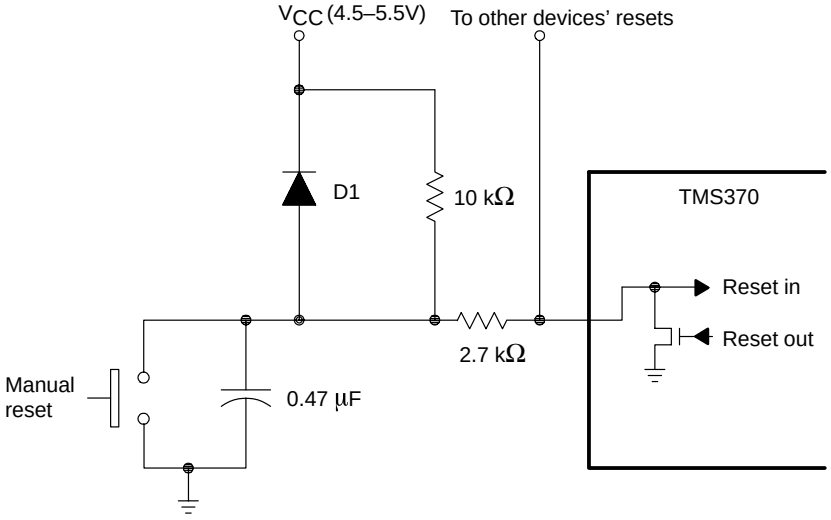
The simple reset circuit shown in Figure 5–5 cannot handle short brownouts or power supply glitches. Voltage glitches can occur when a power switch is closed or power wires are connected. If your design is subject to these conditions, refer to the Reset Circuitry with Low–Voltage Detection (subsection 5.4.2, page 5-22). The TMS370 devices are not guaranteed to operate properly when V_{CC} is outside of the recommended voltage range (4.5 to 5.5 volts).

Table 5–5. Control-Bit States Following a Reset

Register	Control Bit	Power-up Microcomputer	Warm Reset	
			Microcomputer	Microprocessor
SCCR0	µP/µC Mode	0	0	1
	MC PIN DATA	0	0	1
	COLD START	1	See Note 1	See Note 1
	OSC FLT FLAG	0	See Note 1	See Note 1
xPORT1 (See Note 2)	All 8 bits	0	0	1
xPORT2 (See Note 2)	All 8 bits	0	0	1
T1CTL2	WD OVRFL FLAG	0	See Note 1	See Note 1
TXCTL	TX EMPTY	1	1	1
	TXRDY	1	1	1
ADSTAT	AD READY	1	1	1
PACT	PACT TXRDY	1	1	1

Notes: 1) State determined by cause of reset. See bit descriptions in Section 5.2.
2) Refers to port control registers with x = A, B, C, or D.

Figure 5–5. Simple Reset Circuit



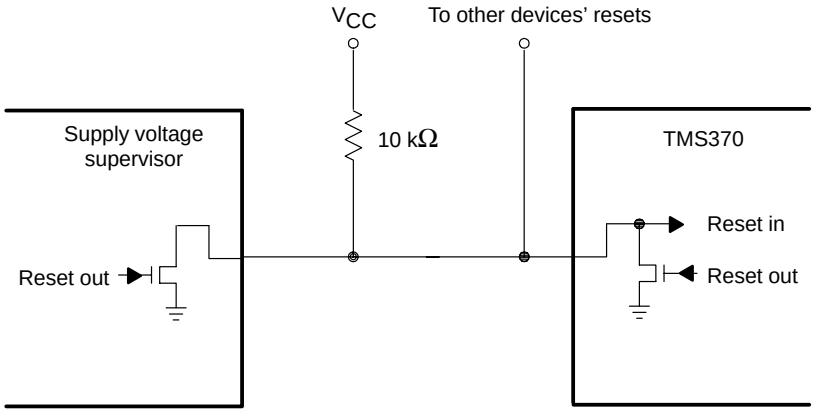
- The RC network of 10 kΩ and 0.47μF provides a power-up rise time. If this power-up rise time is not long enough (depending on the rise time of the power supply you are using), you can use a larger capacitor. However, replacing the 10 kΩ resistor with a larger resistor can cause the voltage at the $\overline{\text{RESET}}$ pin to be less than V_{IH}.
- The 2.7-kΩ resistor protects the $\overline{\text{RESET}}$ pin from the capacitor discharging directly into the pin when the pin is pulled low internally.
- The diode allows the capacitor to discharge quickly during a brownout or power-off situation.

Capacitors should not discharge directly into the $\overline{\text{RESET}}$ pin. Protect this pin from damage by using a resistor such as the 2.7 kΩ resistor shown in Figure 5–5.

5.4.2 Reset Circuitry With Low-Voltage Detection

It is recommended to have an asserted $\overline{\text{RESET}}$ during low-power or brownout conditions. In these instances, an active reset circuit with a low-voltage detection feature can be connected to the $\overline{\text{RESET}}$ pin. Figure 5–6 shows a typical circuit for using a supply voltage supervisor to assert $\overline{\text{RESET}}$.

Figure 5–6. Typical Reset Circuit Using a Supply Voltage Supervisor



The supply voltage supervisor must **not** cause a drive conflict with the TMS370 $\overline{\text{RESET}}$ pin. Most importantly, the supply voltage supervisor should not drive

$\overline{\text{RESET}}$ high since the TMS370 can drive the pin low. However, a pull-up resistor is needed.

To ensure the integrity of the contents of volatile memory (EEPROM, RAM), devices incorporating such memory require that the external $\overline{\text{RESET}}$ pin is active (low) while V_{CC} is below its minimum specified operating level. Active reset circuitry prevents the EEPROM contents from being corrupted by improper instruction execution due to an insufficient V_{CC} supply voltage and ensures that the EEPROM write control register (DEECTL) powers up in the correct state when V_{CC} returns to its specified operating range.

To guarantee the retention of RAM data when power is at 3.0 V to 4.5 V, $\overline{\text{RESET}}$ must be externally asserted and released only while V_{CC} is within the recommended operating range of 4.5 V to 5.5 V.

Chapter 6

EPROM and EEPROM Modules

This chapter discusses the architecture and programming of the following:

- Data EEPROM modules of the TMS370 family
- Program EPROM modules of the TMS370C6xx and TMS370C7xx devices.

Additional information about these modules is included in Chapter 18, *Electrical Specifications and Timings*, and in the *TMS370 Family Applications Book (SPNA017)*. This chapter covers the following topics:

Topic **Page**

6.1 Data EEPROM Module

The entire TMS370 family (except TMS370CxAx and the TMS370xCx) contains data EEPROM modules.

The TMS370 data EEPROM module contains a 256-byte array configured into eight 32-byte blocks. Devices can have multiple 256-byte arrays. Each additional array is also configured with eight 32-byte blocks. The first byte of each 256-byte array is the write protection register (WPR) for that array. This module also contains a voltage generator that provides a special precise programming voltage to the EEPROM array. This special voltage helps increase the reliability of the EEPROM and allows the TMS370 to program the EEPROM with a single V_{CC} source.

Reading the EEPROM module is identical to reading other internal memory and takes two system clock cycles. The CPU can fetch data and execute instructions from the EEPROM arrays. The data EEPROM module can be programmed on an array, a byte, or a single-bit basis. The memory can also be protected from inadvertent writing with a write-protect feature.

The data EEPROM control register (DEECTL) and the WPR control the data EEPROM. The DEECTL register contains the bits needed to initiate and monitor EEPROM programming. The WPR of the given array contains the write protection bits for each 32-byte block of that data EEPROM array.

6.2 Data EEPROM Control Registers

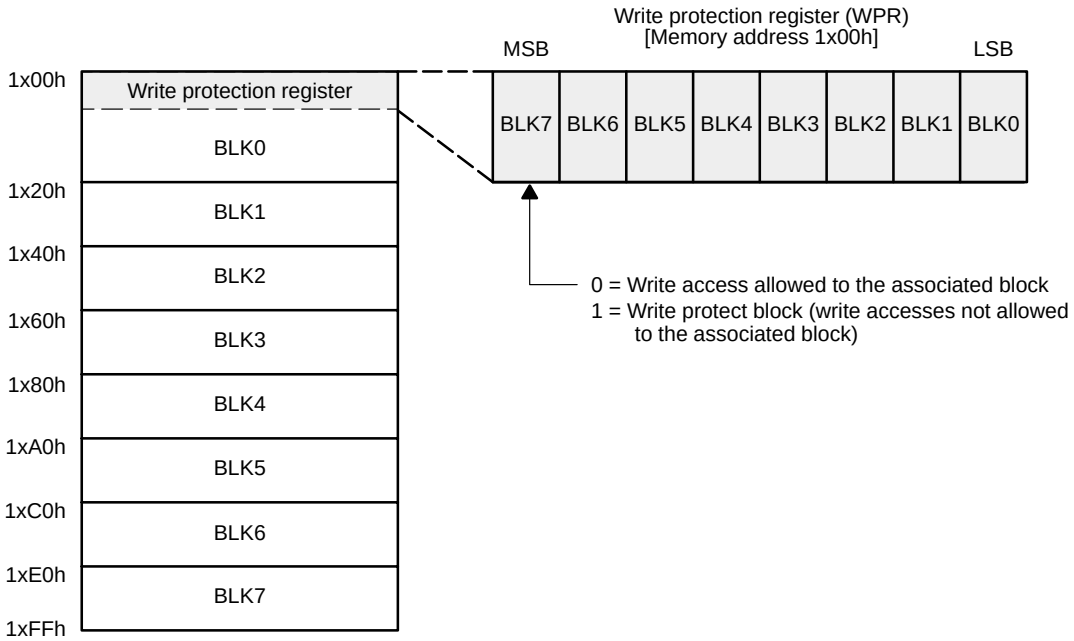
The data EEPROM can be write-protected, block by block (32 bytes), with the WPR(s). The DEECTL register determines the programming mode and when programming is initiated.

6.2.1 Write Protection Register (WPR)

The WPR(s) provide write protection for the data EEPROM contents. The WPR is the first byte of each 256-byte data EEPROM array and is located in BLK0 of this array, generally at address 1x00h (where x is either E or F). This implies that the WPR, for the 256-byte array, is itself protected whenever the BLK0 bit of that array is protected.

There are eight blocks of equal size in the data EEPROM array. Each bit in the WPR corresponds to one of the blocks. Setting a bit to a 1 in this register protects the corresponding block. Figure 6–1 shows the block protected by each bit.

Figure 6–1. Write Protection Bits in an EEPROM Array



Once block 0 is protected, the write-protect configuration cannot be altered unless write protection is overridden by placing the microcomputer into the write-protection override mode. (To enter the WPO mode, apply 12 volts to the MC pin while the $\overline{\text{RESET}}$ pin is a logic 1.) There is no write protection during a write-protection override, and the WPR is considered a normal data location within the data EEPROM array during this time.

Example 6–1 illustrates one way to program the WPR. In this example, the program protects blocks 0 and 2. Also, assume that the WPR contains the value 00h before the example begins.

Example 6–1. Write Protection Register Programming

```
        DINT                ;Disable interrupt
        MOV     #05,A       ;Protect bits for BLK0 and
        MOV     A,&1F00h    ;BLK2
        MOV     #3,P01A     ;Set DEECTL to program 1's
                            ;Set W1W0 and EXE bits
        EINT                ;Enable interrupt
        MOVW    #2778,R011  ;10 ms delay loop
DELAY   INCW    #–1,R011
        JC      DELAY
        MOV     #0,P01A     ;Clear W1W0 and EXE bits
        .
        .
        .
```

See *TMS370 Family Applications Book (SPNA017)* for more examples of programming the EEPROM module.

6.2.2 Data EEPROM Control Register (DEECTL)

The DEECTL register is located in the peripheral file at address P01A (101Ah), and controls the data EEPROM programming.

Data EEPROM Control Register (DEECTL)
[Memory Address 101Ah]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P01A	BUSY	—	—	—	—	AP	W1W0	EXE
	R-*					RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset (-* = see the individual bit description)

Bit 7 **BUSY**.

This bit is set during data EEPROM programming to indicate that an operation is in progress. Reading any location of the EEPROM during programming returns the data being programmed. In order to let the EEPROM voltages stabilize, the BUSY bit is set for 128 SYSCLK cycles:

- ❑ After a reset,
- ❑ After an exit from a low-power mode, and
- ❑ After programming the EEPROM.

If an attempt is made to access the EEPROM during this 128-cycle period, the data EEPROM holds the execution of the processor by asserting the WAIT signal until the 128 SYSCLK cycles are complete.

0 = EEPROM array is ready for access
1 = EEPROM array is not ready for access

Bit 6–3 **Reserved**. Read data is indeterminate.

Bit 2 **AP**. Array Program.

> **Note:**
>
> This bit operates differently for TMS370Cxxx devices (vs. ’370CxxxA or ’370CxxxB devices). See Section A.6, AP Bit in the DEECTL Register (DEECTL.2) on page A-6.

The following applies to devices with a single **or** a multiple 256-byte array:

In a single programming cycle, this bit programs the entire array space with the value specified by the W1W0 bit. However, the device must be in the WPO mode for the array to be programmed. Moreover, there is no write protection during WPO mode; the WPR is considered a normal data location within the data EEPROM array during this time.

If the device is not in the WPO mode, the AP bit has no effect on the programming operation, and a single byte is programmed.

0 = Disables array programming
1 = Enables array programming

Bit 1 **W1W0**. Write1/Write0.

This bit determines whether to use the ones or zeroes programming mode (see Section 6.3, Programming the Data EEPROM, on page 6-6). This bit is write protected whenever the EXE bit is set.

0 = Write zeros
1 = Write ones

Bit 0 **EXE.** Execute.

This bit initiates the write operation defined by the remaining control register bits. When cleared, this bit terminates a programming operation in progress. If the application program reads a data EEPROM location while the EXE bit is set, the processor reads the data being programmed into the EEPROM. If software attempts a write to the EEPROM while the EXE bit is set, the data byte is ignored.

0 = Inactive
1 = Active

6.3 Programming the Data EEPROM

The DEECTL (P01A) register and the associated array's WPR (1x00h) register control the programming of the data EEPROM. Individual bits are programmed to a 1 or 0 under the control of the W1W0 bit and the EXE bit in the DEECTL register.

- When the W1W0 bit is set, bit positions set to 1 in the data byte are programmed to 1 in the EEPROM byte; zeros are not changed.

- When the W1W0 bit is cleared, bit positions cleared to 0 in the data byte are programmed to 0 in the EEPROM byte; ones are not changed.

The EXE bit initiates EEPROM programming when set and disables programming when cleared. The WPR (1x00h) registers must have the corresponding protection bit cleared or be in the WPO mode to enable a data EEPROM write operation. (To enter the WPO mode, apply 12 volts to the MC pin while the $\overline{\text{RESET}}$ pin is a logic 1.)

To load the data byte into the EEPROM module:

1) Perform a memory-write operation to the EEPROM at the desired address. The data byte is latched in the module, ready for the Execute command (EXE bit = 1).

   To prevent data corruption, ensure that nonmaskable interrupt routines do not access the EEPROM between the EEPROM write instruction and the point when the EXE bit is set to 1.

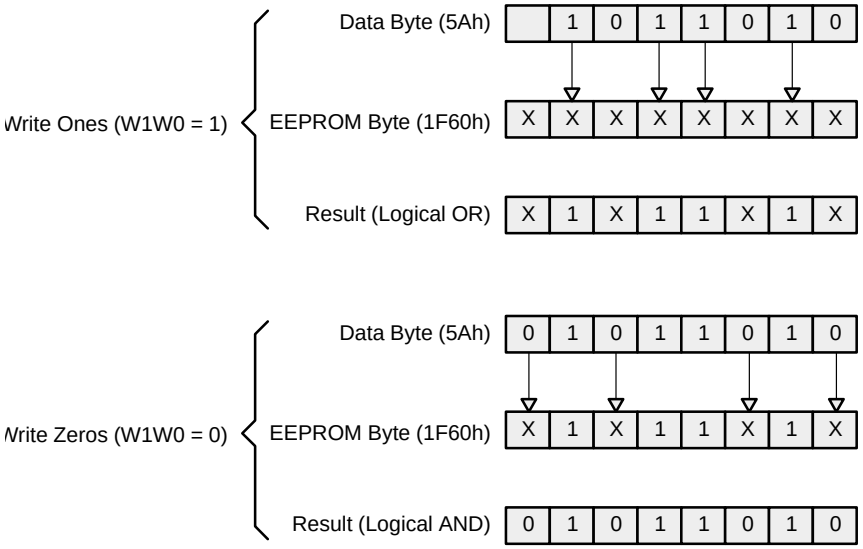
2) Following the memory cycle to the EEPROM address, write 03h (for W1W0 = 1) or 01h (for W1W0 = 0) to the DEECTL register to set the W1W0 and EXE bits. The W1W0 and the EXE bits must remain unchanged for the duration of the EEPROM timing parameter of $t_{W(PGM)B}$ to ensure proper programming.

3) When the program time has elapsed, reset the EXE bit with another write operation to the DEECTL register.

If W1W0 = 1, the data that now resides in the programmed EEPROM location is the logical OR of the previous data stored in the location and the data written to the location. If W1W0 = 0, the data that now resides in the programmed EEPROM location is the logical AND of the previous data stored in the location and the data written to the location.

If a data value cannot be programmed by writing only ones or zeros, first perform the write-ones operation and follow it with a write-zeros operation (or write zeros followed by write ones). Figure 6–2 illustrates these operations. In the programming operations, only the EEPROM bits that do not match the data bits are programmed. Therefore, there is no need to read the EEPROM value to determine which bits to program.

Figure 6–2. EEPROM Programming Example



The software should end the programming operation before entering a halt or standby state. When the microcomputer is in the halt or standby low-power mode, all operations of the data EEPROM module are stopped, and all DEECTL bits are cleared. Any EEPROM programming operation in progress is aborted when the halt is entered, and the data at the address being programmed is indeterminate.

The subroutine in Example 6–2 loads the data byte 5Ah into the data EEPROM location 1F60h. Figure 6–2 illustrates the result of this subroutine.

Example 6–2. Data EEPROM Programming

The following subroutine loads the data byte 5Ah into the data EEPROM location 1F60h.

```
(a)           DINT              ; Disable all interrupts
(b)  DATA     MOV   #5Ah,A      ; Write 5A to location 1F60h
              MOV   A,&1F60h
(c)           MOV   #03,P01A    ; Write Ones: W1W0=1, EXE=1
(d)           EINT              ; Enable all interrupts
(e)           MOVW  #2778,R017  ; Begin t(PGM)B delay (10 ms)
(f)  DELAY1   INCW  #-1,R017    ; Decrement R017
(g)           JC    DELAY1      ; Jump to DELAY1 if R017>0
(h)           MOV   #0,P01A     ; Clear DEECTL. EXE=0
(i)           DINT              ; Disable all interrupts
(j)           MOV   #5Ah,A      ; Write 5A to location 1F60h
              MOV   A,&1F60h
(k)           MOV   #01,P01A    ; Write zeros: W1W0=0 EXE=1
(l)           EINT              ; Enable all interrupts
(m)           MOVW  #2778,R017  ; Begin t(PGM)B delay (10 ms)
(n)  DELAY2   INCW  #-1,R017    ; Decrement R017
(o)           JC    DELAY2      ; Jump to DELAY2 if R017>0
(p)           MOV   #0,P01A     ; Clear DEECTL. EXE=0
              .
              .
              .
```

- Disable all interrupts. When programming the data EEPROM, you must ensure that nonmaskable interrupt routines do not access the EEPROM between an EEPROM write instruction and the point when the EXE bit is set to 1 (steps a and i in Example 6–2), or data will be corrupted.
- Load the value 5Ah into the data EEPROM address 1F60h (step b).
- Begin a write-ones programming sequence (step c) by setting the W1W0 and EXE bits in the DEECTL register to a 1.
- Re-enable all interrupts (step d).
- The programming delay parameter, $t_{W(PGM)B}$, (10 ms for this example—see Chapter 18, *Electrical Specifications and Timings*, for the required timing) is taken care of with a delay loop (steps f and g).

- The number of loops required is 2778 (step e) and is derived in the following manner:
  - The delay loop (steps f and g) requires 18 SYSCLK cycles to complete if a jump is taken.
  - An operating frequency of 5-MHz SYSCLK results in a system cycle time of 200 ns.
  - The number of loops required is calculated as follows:

    loop count = $t_{W(PGM)B}$ / (system cycle time $\times$ delay loop cycle count)

    loop count = 10 ms / (200 ns $\times$ 18) = 10 ms / 3.6 µs = 2778

    Note: As an alternative, a timer can be used for this delay.
- After the delay, clear the EXE bit (step h), re-enable all interrupts, and continue the write-zeros routine (steps i through p). The value 5Ah has now been programmed into location 1F60h of the data EEPROM.

Following an EEPROM write operation, the EEPROM voltage must stabilize before an EEPROM read operation is performed. The BUSY flag indicates the status of the EEPROM voltage. When BUSY is set, the EEPROM is not ready for a read operation. The BUSY flag is cleared to zero (0) by the EEPROM control logic when 128 system clock cycles have elapsed following the time that the EXE bit is cleared to 0.The BUSY bit remains set for 128 SYSCLK cycles:

- After a reset,
- After exit from a low-power mode, and
- After programming the EEPROM.

If an attempt is made to access the EEPROM during this 128 SYSCLK cycle period, the data EEPROM holds execution of the processor by asserting the WAIT signal until the 128 SYSCLK cycles are completed.

To prevent data corruption of the read or write EEPROM location, do not access EEPROM locations between writing data to the EEPROM address and setting the EXE bit to 1. In addition, you should disable interrupts during this time.

6.4 Program EPROM Modules

The program EPROM modules used in the TMS370 family replace the 2K- and 48K-byte program ROM within the TMS370 families for system prototypes or small production runs. The CPU can fetch data and execute instructions from these memory spaces.

These modules consist of either an 8K-byte array or a 16K-byte array of EPROM. The following describes each EPROM device by total memory size:

- 8K- or 16K-byte EPROM devices
  - Address area:
    - 8K-byte array: 6000h – 7FFFh
    - 16K-byte array: 4000h – 7FFFh
  - Arrays controlled by register EPCTLM (address 101Ch, P01C)

- 24K-byte EPROM device (one 16K- and one 8K-byte array of EPROM)
  - Address area:
    - First 16K–byte array: 2000h – 5FFFh
    - Next 8K–byte array: 6000h – 7FFFh
  - Arrays controlled by:
    - First 16K-byte array: register EPCTLL (address 101Eh, P01E)
    - Next 8K-byte array: register EPCTLM (address 101Ch, P01C)

- 32K-byte EPROM device (two 16K-byte arrays of EPROM)
  - Address area:
    - First 16K-byte array: 2000h – 5FFFh
    - Second 16K-byte array: 6000h – 9FFFh
  - Arrays controlled by:
    - First 16K-byte array: register EPCTLL (address 101Eh, P01E)
    - Second 16K-byte array: register EPCTLM (address 101Ch, P01C)

Note: EPCTLx Register Differs According to Device

The EPCTLL and EPCTLM registers operate differently for TMS370C758, TMS370C758A, and TMS370C758B. Refer to Section A.7 on page A-7.

- 48K-byte EPROM device (three 16K-byte arrays of EPROM)
  - Address area:
    - First 16K-byte array: 2000h – 5FFFh
    - Second 16K-byte array: 6000h – 9FFFh
    - Third 16K-byte array: A000h – DFFFh

- Arrays are controlled by:
  - First 16K-byte array: register EPCTLL (address 101Eh, P01E)
  - Second 16K-byte array: register EPCTLM (address 101Ch, P01C)
  - Third 16K-byte array: register EPCTLH (address 1014h, P014)

Table 6–1 is a summary of the memory addresses and corresponding control registers for each EPROM size.

The CPU accesses the arrays with normal memory read cycles. Write cycles to the program EPROM require a special sequence of events. This sequence is described in subsection 6.4.3, Programming the Program EPROM, page 6-13.

An external voltage supply is needed at the MC pin to provide the necessary V_{PP} for programming. The EPCTLL, EPCTLM, and EPCTLH registers in the peripheral file control the programming.

Table 6–1. EPROM Memory Map Summary

Parameter	Device								
	'370C7x2	**'370C7x6**	**'370C7x7**		**'370C7x8**		**'370C7x9**		
EPROM Size	8K bytes	16K bytes	24K bytes		32K bytes		48K bytes		
Memory Mapped	8K 6000h–7FFFh	16K 4000h–7FFFh	16K 2000h–5FFFh	8K 6000h–7FFFh	First 16K 2000h–5FFFh	Second 16K 6000h–9FFFh	First 16K 2000h–5FFFh	Second 16K 6000h–9FFFh	Third 16K A000h–DFFFh
Control Registers	EPCTLM P01C‡	EPCTLM P01C‡	EPCTLL P01E†	EPCTLM P01C‡	EPCTLL P01E†	EPCTLM P01C‡	EPCTLL P01E†	EPCTLM P01C‡	EPCTLH P014§

† Program EPROM control register—low array
‡ Program EPROM control register—middle array
§ Program EPROM control register—high array

6.4.1 Erasing the EPROM

Before programming (windowed versions), the EPROM module is erased by exposing the device through the transparent window to high-intensity ultraviolet (UV) light (wavelength of 2537 angstroms). The recommended minimum exposure dose (UV intensity $\times$ exposure time) is 15 watt-seconds per square centimeter. A typical 12 milliwatt-per-square-centimeter, filterless UV lamp will erase the device in 21 minutes. The lamp should be located about 2.5 centimeters above the chip during erasure. After erasure, the entire array is at logic 1 state. A programmed 0 can be erased to 1 only by exposure to ultraviolet light. Note that normal ambient light contains the correct wavelength for erasure. Therefore, when using a programmed device, you should cover the window with an opaque label. All devices are erased to a logical 1 at the factory.

CAUTION

Exposing the EPROM module to the ultraviolet light may also cause erasure in any EEPROM module. Any useful data stored in the EEPROM must be reprogrammed after exposure to UV light.

6.4.2 Program EPROM Control Register (EPCTLx)

The EPCTLL, EPCTLM, or EPCTLH (collectively referred to as “EPCTLx” in section 6.4.3) registers at addresses 101Eh, 101Ch, or 1014h, respectively, in the peripheral file control the programming of the program EPROM.

Program EPROM Control Register (EPCTLx)
[Memory Addresses 101Eh, 101Ch, or 104Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P01C, P01E or P014	BUSY	VPPS	—	—	—	—	W0	EXE
	R-0	RW-0					RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **BUSY.**

This bit reflects the value of the EXE bit.

Bit 6 **VPPS.**

This bit determines whether the programming voltage (V_{PP}) at the MC pin is connected to the EPROM module.

0 = Disables programming
1 = Enables programming

Bit 5–2 **Reserved.** Read data is indeterminate.

Bit 1 **W0.** Write 0.

This bit determines whether the programming of the zero bits (in the byte written) is enabled.

0 = Enables programming of 0 bits
1 = Disables programming of 0 bits

Bit 0 **EXE.** Execute.

This bit initiates the write operation defined by the other control register bits. When cleared, this bit terminates the operation.

0 = Inactive
1 = Active

6.4.3 Programming the Program EPROM

Programming zero (0) to the EPROM is controlled by the EPCTLx register via the EXE bit and the VPPS bit.

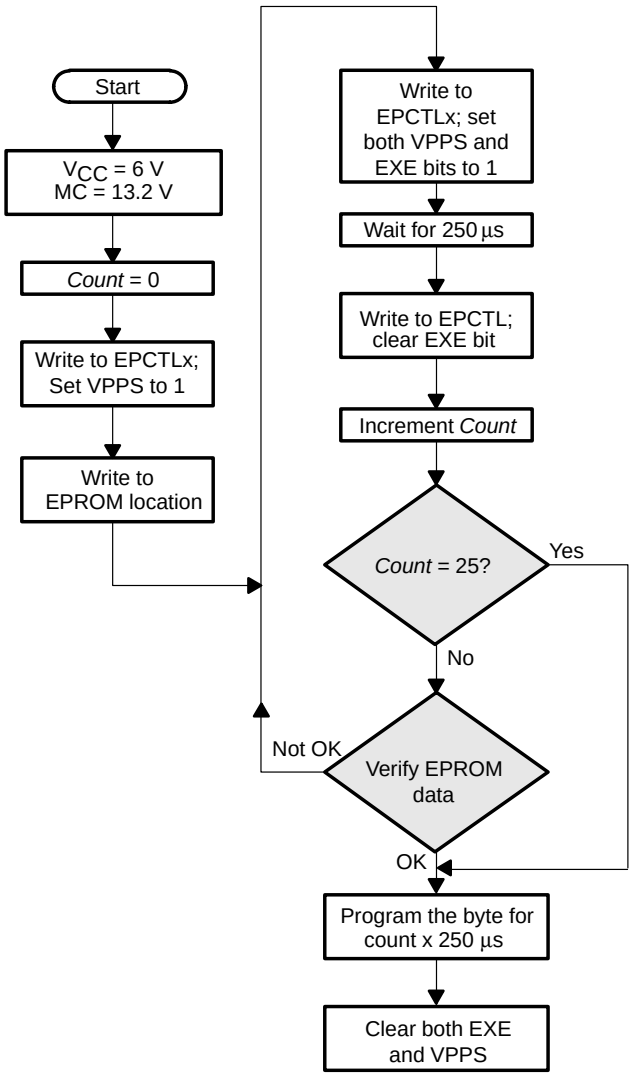
- The EXE bit initiates EPROM programming when set and disables programming when cleared.
- The VPPS bit connects the programming voltage (V_{PP}) at the MC pin to the EPROM module.

VPPS (EPCTLx.6) and EXE (EPCTLx.0) should be set separately, and the VPPS bit should be set at least two microseconds before the EXE bit is set. After programming, the application should wait for four microseconds before any read attempt is made.

Perform the programming operation (see Figure 6–3 on page 6-14) in the following sequence:

1) Supply the programming voltage to the MC pin.
2) Set (to 1) bit EPCTLx.6 (VPPS).
3) Write to the target EPROM location.
4) Set bit EPCTLx.0 (EXE). Wait at least two microseconds after step 2.
5) Wait for program time to elapse (250 microsecond).
6) Clear bit EPCTLx.0. Leave VPPS set to 1.
7) Read the byte being programmed; if the correct data is not read, repeat steps 4 through 6 for count *(a number of)* times to a maximum of 25.
8) Set bit EPCTLx.0 for final programming.
9) Wait for the program time to elapse (2 times count duration).
10) Clear bits EPCTLx.0 and EPCTLx.6.

Figure 6–3. EPROM Programming Operation



Note: VPPS and EXE are bits 6 and 0, respectively, of registers EPCTLx.

Programming operations require an external power supply at V_{PP} (13.2 V), I_{PP} (30 mA). Programming voltage (V_{PP}) is supplied via the MC pin. This also automatically puts the microcontroller in the write protection override (WPO) mode. Programming voltage can be applied via the MC pin anytime after the $\overline{\text{RESET}}$ signal goes inactive high, and the MC pin can remain at V_{PP} after programming (after the EXE bit is cleared). Applying programming voltage while $\overline{\text{RESET}}$ is active will set the microcontroller to reserved mode, where programming operations are inhibited.

6.4.4 Write Protection of the Program EPROM

To override the EPROM write protection, the V_{PP} must be applied to the MC pin, *and* the VPPS bit (EPCTL.6) must be set. This dual requirement ensures that the program EPROM is not accidentally overwritten during data EEPROM operations when V_{PP} is applied to the MC pin. Data EEPROM can be programmed when the VPPS bit is set.

Chapter 7

Timer 1 (T1) Module

This chapter discusses the architecture and programming of the T1 module and covers the following topics:

Topic **Page**

7.1 T1 Overview

The T1 module of the TMS370 family provides enhanced timer resources to perform realtime system control. This module contains a general-purpose timer and a watchdog (WD) timer. Both timers allow the program selection of input clock sources (realtime, external event, or pulse accumulate) with multiple 16-bit registers (input capture and compare) for special timer function control. Table 7–1 shows timer solutions for different system requirements.

Table 7–1. Timer-System Solutions

Requirement	Timer Solution
Realtime system control	Interval timers with interrupts
Input pulse-width measurement	Pulse accumulate or input capture functions
External event synchronization	Event count function
Timer output control	Compare function
PWM output control	PWM output function
System integrity	WD function

7.1.1 Physical Description

The T1 module, shown in Figure 7–1, has the following components:

- **A 16-bit general-purpose timer** that provides capture, compare, and event functions.
  - The capture function latches the counter value to the occurrence of an external input.
  - The event function keeps a cumulative total of the transitions on the T1EVT pin.
  - The compare function triggers when the counter matches the contents of a compare register.
- **A 16-bit WD timer** that software can reconfigure as a simple counter/timer, an event counter, or a pulse accumulator if the WD feature is not needed.
- **A prescaler/clock source** that determines the independent clock sources for the general-purpose timer and for the WD timer.
- **A selectable edge-detection circuitry** that senses active transitions on the T1IC/CR pin.

❑ **Interrupts**

The module can be programmed to issue interrupts on the occurrence of the following:

- A capture
- A compare equal
- A counter overflow
- An external edge detect

❑ **I/O pins**

The T1 module has three I/O pins that can be dedicated for counter functions or as general-purpose I/O pins. These are as follows:

- T1EVT, an input to the event counter or the external clock source
- T1IC/CR, an input to the input capture, counter reset, or PWM circuit
- T1PWM, the PWM output

Table 7–2 on page 7-4 shows the definitions of these pins, according to operating mode.

Figure 7–1. T1 Block Diagram

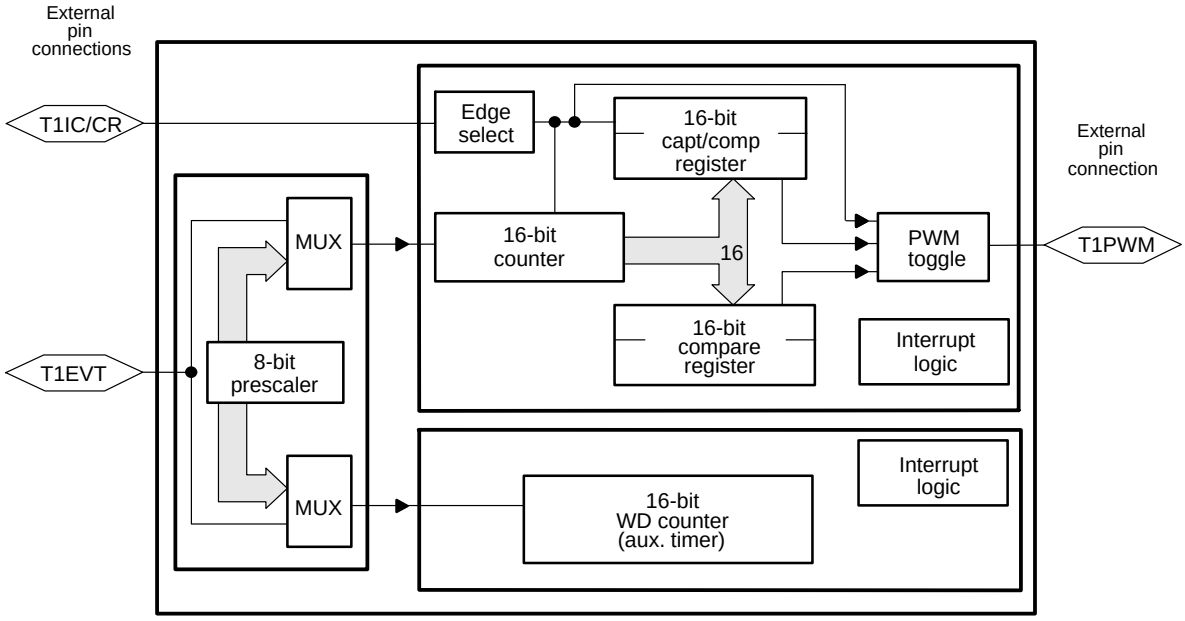


Table 7–2. T1 I/O Pin Definitions

Pin	Dual Compare Mode	Capture/Compare Mode
T1IC/CR	Counter reset input	Input capture 1 input
T1PWM	PWM output	PWM output
T1EVT	External event input or pulse accumulate input	External event input or pulse accumulate input

7.1.2 Operating Modes

The general-purpose T1 module has the following two modes of operation:

- **Dual compare mode.** The timer is configured to provide two compare registers, an external or software timer reset, an internal or external clock source, and a programmable pulse-width modulated (PWM) output. The PWM output can be configured to toggle on specified events.
- **Capture/compare mode.** The timer is configured to provide one input capture register and one compare register for use with the general-purpose timer. The compare register can be used to provide periodic interrupts to the TMS370 CPU. The capture register can be configured to capture the current timer value upon either edge of an external input.

7.1.3 Control Registers

The T1 control registers are located at addresses 1040h to 104Fh and occupy peripheral file frame 4. The function of each location is shown in Table 7–3.

Table 7–3. T1 and WD Timer Memory Map

Peripheral File Location	Symbol	Name	Description
P040 P041	T1CNTR	T1 Counter — MSbyte T1 Counter — LSbyte	16-bit resettable counter
P042 P043	T1C	Compare Register — MSbyte Compare Register — LSbyte	16-bit compare register
P044 P045	T1CC	Capture/Compare Register — MSbyte Capture/Compare Register — LSbyte	16-bit capture/compare register
P046 P047	WDCNTR	WD Counter — MSbyte WD Counter — LSbyte	16-bit WD counter
P048	WDRST	WD Reset Key	Resets the WD timer.
P049	T1CTL1	T1 Control Register 1	Controls the prescaler inputs to the WD timer and to the general-purpose timer.
P04A	T1CTL2	T1 Control Register 2	Controls the T1 and WD overflow interrupts and contains the T1 software reset bit.
P04B	T1CTL3	T1 Control Register 3	Controls the edge-detect and compare interrupts.
P04C	T1CTL4	T1 Control Register 4	Controls the mode of operation and various functions of the T1 input and output pins.
P04D	T1PC1	T1 Port Control Register 1	Controls the I/O functions of the T1 module and T1EVT pin.
P04E	T1PC2	T1 Port Control Register 2	Controls the I/O functions of the T1 module, T1IC/CR pin, and T1PWM pin.
P04F	T1PRI	T1 Interrupt Priority Control Register	Controls the level of the T1 interrupt.

7.2 General-Purpose Timer Components

The general-purpose timer uses a 16-bit counter, a compare register, and a capture/compare register to provide event, compare, and capture functions.

7.2.1 16-Bit Resettable Counter

The free-running, 16-bit counter (T1CNTR) is clocked by the output of the prescaler/clock source. The program can access the 16-bit counter at P040 (T1 counter MSbyte) and P041 (T1 counter LSbyte) in peripheral file frame 4.

- During initialization, the counter is loaded with the value 0000h and begins its count.
- If the counter is not reset before reaching FFFFh, the counter rolls over to 0000h and continues counting. Upon counter rollover, the T1 OVRFL INT FLAG bit (T1CTL2.3) is set, and a timer interrupt is generated if the T1 OVRFL INT ENA bit (T1CTL2.4) is set.
- During counting, the counter can be reset to 0000h by any of the following:
  - A 1 written to the T1 SW RESET bit (T1CTL2.0)
  - A compare equal condition from the dedicated T1 compare function
  - A system reset
  - An external pulse on the T1IC/CR pin (dual compare mode only) if the T1CR RST ENA bit (T1CTL4.1) is set to a 1.

You can select the external-transition direction on the T1IC/CR pin, low-to-high or high-to-low, to reset the counter. To do this, use the T1EDGE POLARITY bit (T1CTL4.2).

Special circuitry prevents the contents of the T1CNTR register from changing in the middle of a 16-bit read operation. See the note in Section 7.9 on page 7-30.

7.2.2 Compare Register

The compare register circuit consists of a 16-bit wide, read/write data register (T1C) and logic to compare the counter's current value with the value stored in the compare register. The program can access the 16-bit compare register at P042 (compare register MSbyte) and P043 (compare register LSbyte) in peripheral file frame 4.

When the counter's value matches the compare register value, then the circuit performs the following actions:

- Sets the T1C1 INT FLAG bit (T1CTL3.5) to 1.
- Clocks the output latch to toggle the T1PWM output pin if the T1C1 OUT ENA bit (T1CTL4.6) is set.
- Generates a T1 interrupt if the T1C1 INT ENA bit (T1CTL3.0) is set.
- Resets the counter if the T1C1 RST ENA bit (T1CTL4.4) is set (dual compare mode only).

The compare register is initialized to 0000h following a reset.

Special circuitry prevents the contents of the T1C register from changing in the middle of a 16-bit read operation. See the note in Section 7.9 on page 7-30.

Note:

If the counter is programmed to reset when its value equals the contents of the compare register, the reset occurs on the following counter clock cycle (after a prescale). However, the compare flag is set and the interrupt event occurs during the clock cycle that incremented the counter to equal the compare-equal value. As a result, there could be a delay of up to 256 system clock cycles (depending on the prescale tap in use) from the time that the event is recognized by the program until the counter actually resets to zero. If the program writes to the compare register during this interval, the counter cannot be reset during the following counter clock cycle.

The compare register value required for a specific timing application can be calculated using the following formula:

$$\text{Compare Value} = \left(\frac{t}{\text{PS} \times \frac{1}{\text{SYSCLK}}} \right) - 1$$

where:

t	=	desired timer compare period (seconds)
SYSCLK	=	CLKIN/4 for divide-by-4 (external clock frequency)
	=	CLKIN/1 for divide-by-1 clock
PS	=	1, 4, 16, 64, or 256, depending on the prescale tap selected

Table 7–4 provides some sample compare register values to achieve the various desired timings using a 5 MHz SYSCLK.

Table 7–4. T1 Compare Values: (5 MHz SYSCLK)

Time			T1 Compare Register Value (N)		Percent Error
Seconds	mSeconds	Prescale	Decimal	Hex	(See Note)
0.0005	0.5	None	2499	009C3h	0.000
0.001	1	None	4999	01387h	0.000
0.002	2	None	9999	0270Fh	0.000
0.005	5	None	24999	061A7h	0.000
0.01	10	None	49999	0C34Fh	0.000
0.02	20	/4	24999	061A7h	0.000
0.05	50	/4	62499	0F423h	0.000
0.1	100	/16	31249	07A11h	0.000
0.2	200	/16	62499	0F423h	0.000
0.5	500	/64	39062	09896h	0.000
1.0	1000	/256	19530	04C4Ah	0.001
2.0	2000	/256	39061	09895h	0.001
3.0	3000	/256	58593	0E4E1h	0.001

Note: Percent of error induced by the T1 formula. This error margin varies, depending on the desired timer compare period and the minimum timer resolution (PS $\times$ (1/SYSCLK)).

7.2.3 Capture/Compare Register

The 16-bit wide capture/compare register (T1CC) serves one of two functions, depending on the operating mode. The T1CC register is located at P044 (capture/compare register MSbyte) and P045 (capture/compare register LSbyte) in peripheral file frame 4.

Special circuitry prevents the contents of the T1CC register from changing in the middle of a 16-bit read operation. See the note in Section 7.9 on page 7-30.

7.2.3.1 Dual Compare Mode

In the dual compare mode, the T1CC register acts as a read/write compare register. It functions exactly like the compare register described in subsection 7.2.2 except that T1CC **cannot** reset the counter.

When the counter value matches the capture/compare register value, the circuit performs the following:

- ❑ Sets the T1C2 INT FLAG bit (T1CTL3.6) to 1.
- ❑ Clocks the output latch to toggle the T1PWM output pin if the T1C2 OUT ENA bit (T1CTL4.5) is set.
- ❑ Generates a T1 interrupt if the T1C2 INT ENA bit (T1CTL3.1) is set.

7.2.3.2 Capture/Compare Mode

In the capture/compare mode, the edge detection signal captures the current counter content, loads it into the T1CC register, and sets the T1EDGE INT FLAG bit (T1CTL3.7).

7.3 Operating Modes of the General-Purpose Timer

The operating mode of the T1 general-purpose timer determines whether the capture/compare register functions as a capture register in the capture/compare mode or as a compare register in the dual compare mode. The T1 MODE bit (T1CTL4.7) selects the mode as follows:

T1 MODE = 0 = dual compare mode
T1 MODE = 1 = capture/compare mode

7.3.1 Dual Compare Mode

The dual compare mode provides the following:

- A 16-bit compare register (called compare 1)
- A 16-bit capture/compare register that acts as a compare register (called compare 2)
- A 16-bit external, resettable counter
- A timer output pin

These components allow the timer to act as an interval timer, a PWM output, simple output toggle, or to perform other timer functions. The dual compare mode is shown in Figure 7–2.

The dual compare mode continuously compares the contents of the two compare registers to the current value of the 16-bit counter.

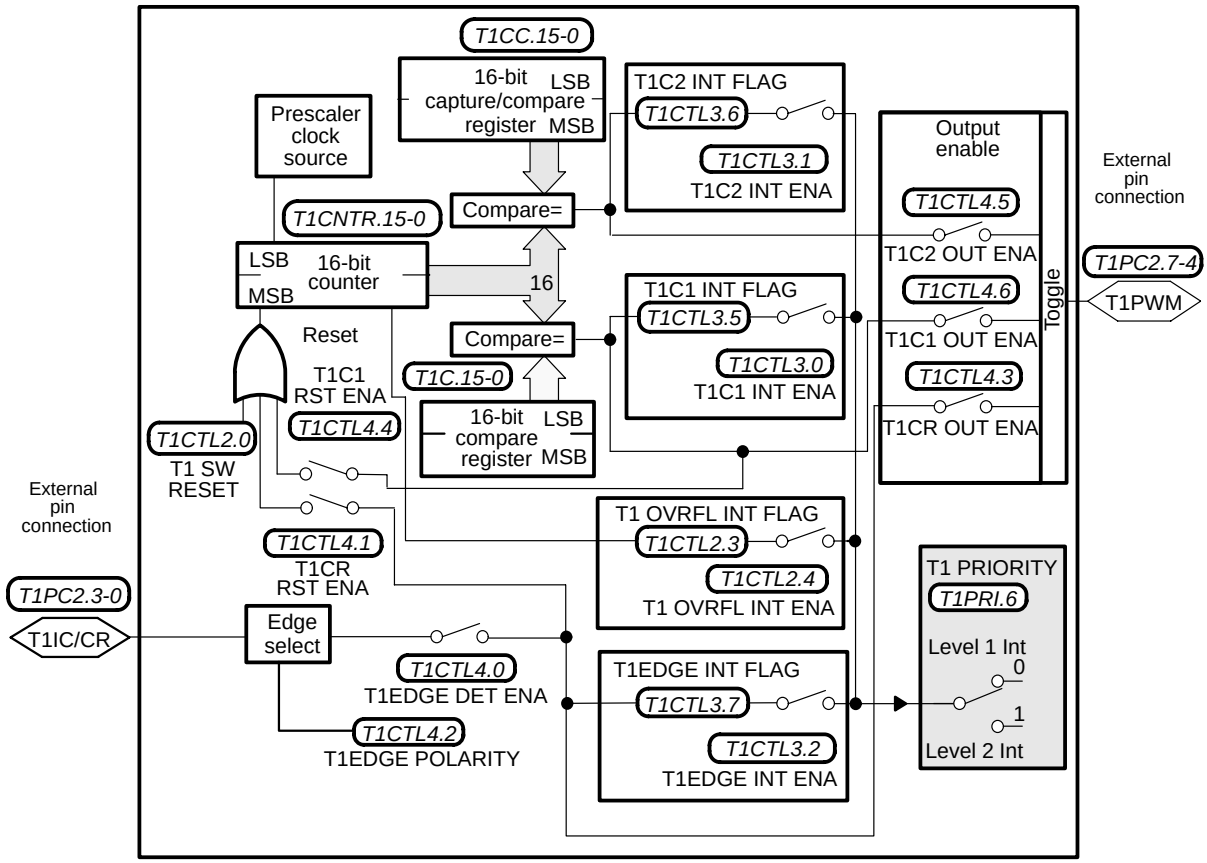
- If the compare 1 register equals the counter, the circuit conducts the following:
  - Sets the T1C1 INT FLAG bit (T1CTL3.5) to 1.
  - Clocks the output latch to toggle the T1PWM output pin if the T1C1 OUT ENA bit (T1CTL4.6) is set.
  - Generates a timer 1 interrupt if the T1C1 INT ENA bit (T1CTL3.0) is set.
  - Initiates a counter reset if the T1C1 RST ENA bit (T1CTL4.4) is set.

  Additionally, you can program an interval timer function by using the compare-equal condition to generate a system interrupt combined with the counter reset function.

- If the compare 2 register equals the counter, then the circuit conducts the following:
  - Sets the T1C2 INT FLAG bit (T1CTL3.6) to 1.
  - Clocks the output latch to toggle the T1PWM output pin if the T1C2 OUT ENA bit (T1CTL4.5) is set.
  - Generates a T1 interrupt if the T1C2 INT ENA bit (T1CTL3.1) is set.

  The compare 2 register can be used as an additional system timing function.

Figure 7–2. Dual Compare Mode



Note: The annotations on this diagram identify the register and the bit(s) in the peripheral frame. For example, the actual address of T1CTL2.0 is 104Ah, bit 0, in the T1CTL2 register.

7.3.1.1 PWM Applications

Either compare register can be used to toggle the T1PWM output pin when a compare-equal condition occurs. Using both compare registers to control the T1PWM pin allows direct PWM generation with minimal CPU software overhead.

In typical PWM applications, the compare registers are loaded as follows:

- The compare 1 register is loaded with the periodic interval and configured to allow a counter reset on a compare-equal condition.
- The compare 2 (capture/compare) register is loaded with the pulse width to be generated within that interval. The program pulse width can be changed by the application program during the timer operation to alter the PWM output. For high-speed control applications, a minimum pulse width of 200 ns and a period as low as 400 ns can be maintained when a 5-MHz SYSCLK is used.

The PWM output can be used to support time-critical control applications. In these applications, an external input (T1IC/CR) is typically used to:

- Reset the counter.
- Generate a timer interrupt.
- Toggle the T1PWM pin to start the PWM output.

The compare function then toggles the output after the programmed pulse width has elapsed.

7.3.1.2 Input Edge Detect

The input edge detect function is enabled under program control by the T1EDGE DET ENA bit (T1CTL4.0); upon the next occurrence of the selected edge transition, the following occurs:

- The T1EDGE INT FLAG bit (T1CTL3.7) is set.
- A timer interrupt is generated (if T1EDGE INT ENA = 1).
- The T1PWM output pin is toggled (if T1CR OUT ENA = 1).

The T1EDGE POLARITY bit (T1CTL4.2) selects the active input transition. In the dual compare mode, the edge detect function must be re-enabled after each valid edge detect.

7.3.1.3 Clock Input

The clock input to the 16-bit counter (T1CNTR) is either the internal system clock, with or without prescale, or the external clock (T1EVT). The clock pulse to the counter is always synchronized with the system clock.

The counter (T1CNTR) is free-running except when it receives a reset pulse from one of the following sources:

- A 1 written to the T1 SW RESET (T1CTL2.0) bit
- A compare equal condition from the dedicated T1 compare function
- A system reset
- An external pulse on the T1IC/CR pin (dual compare mode)

The counter rolls over to 0000h if it is not reset before a count of FFFFh. When this rollover occurs, the counter sets the T1 OVRFL INT FLAG (T1CTL2.3), generates an interrupt if the T1 OVRFL INT ENA bit (T1CTL2.4) is set, and continues counting.

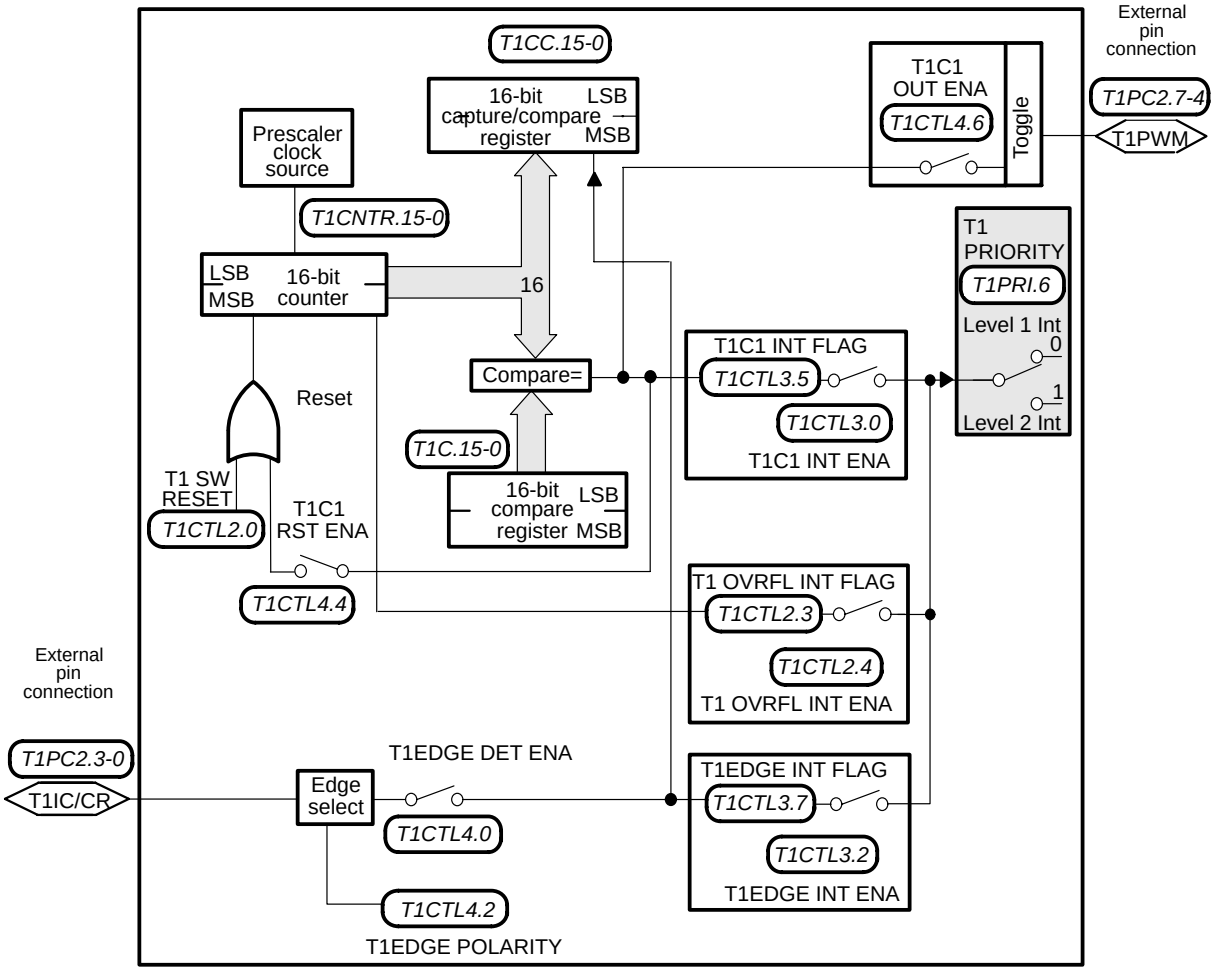
7.3.2 Capture/Compare Mode

In the capture/compare mode, T1 provides the following:

- A 16-bit input capture register for external timing and pulse-width measurement.
- A 16-bit compare register for use as a programmable interval timer. This register functions the same as compare 1 does in the dual compare mode described in subsection 7.3.1, including the ability to toggle the PWM pin.

The capture/compare mode is shown in Figure 7–3.

Figure 7–3. Capture/Compare Mode



On the occurrence of valid input on the T1IC/CR pin, the following occurs:

- The current counter value is loaded into the 16-bit input capture register.
- The T1EDGE INT FLAG bit (T1CTL3.7) is set.
- If T1EDGE INT ENA bit (T1CTL3.2) is set, a timer interrupt is generated.

The input detect function is enabled by the T1EDGE DET ENA bit (T1CTL4.0), with the T1EDGE POLARITY bit (T1CTL4.2) selecting the active input transition. In the capture/compare mode, the edge detect function, once enabled, remains enabled following a valid edge detect.

7.4 Edge-Detection Circuitry

The edge detection circuitry senses active transitions on the T1 input capture/ counter reset pin (T1IC/CR). The T1EDGE POLARITY bit (T1CTL4.2) determines whether the active transition is low-to-high or high-to-low. The module sets the T1EDGE INT FLAG (T1CTL3.7) when an active transition is detected. The program must reset this flag.

7.4.1 Dual Compare Mode

In this mode, the program must set the T1EDGE DET ENA bit (T1CTL4.0) to re-enable the circuit after each edge detection. Writing a 1 to this bit enables the detect circuit to look for the next correct level transition. After this active transition occurs, the T1EDGE DET ENA bit is cleared.

When the edge detection circuit is enabled and detects the appropriate edge transition, the T1EDGE INT FLAG bit (T1CTL3.7) is set.

When the T1CR RST ENA bit (T1CTL4.1) is set, the selected edge resets the counter. If the T1CR OUT ENA bit (T1CTL4.3) is set, the selected edge toggles the T1PWM output latch.

The T1EDGE POLARITY bit (T1CTL4.2) determines which edge polarity (rising or falling) is detected.

7.4.2 Capture/Compare Mode

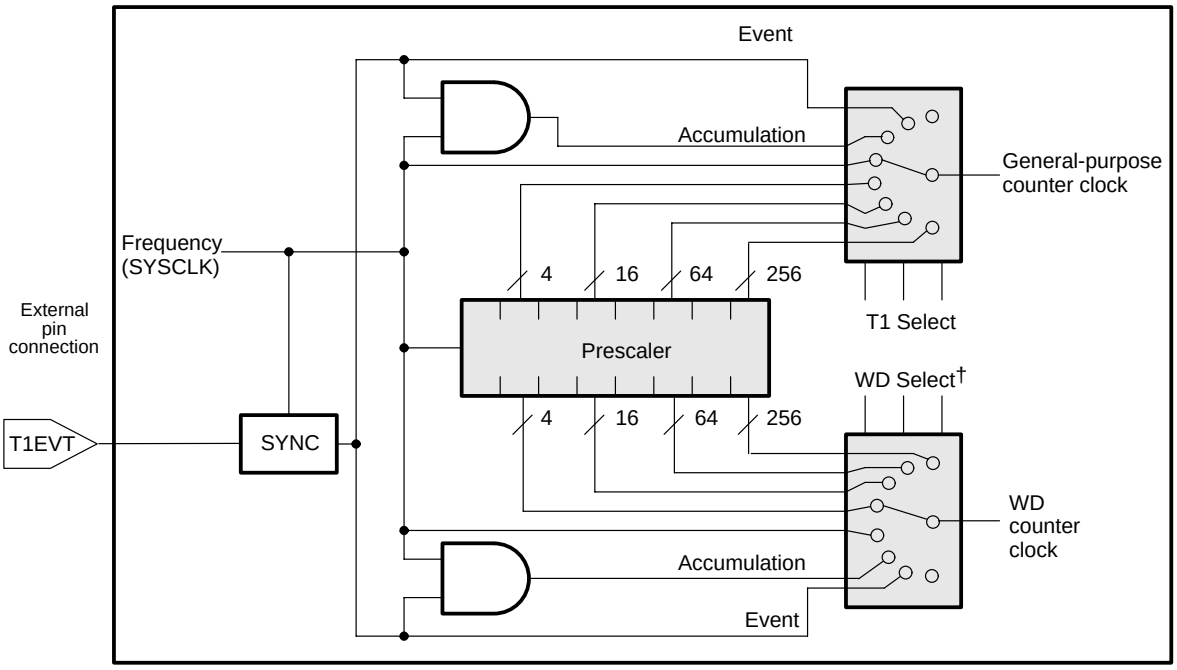
When the appropriate (rising or falling) transition is detected, the edge detection circuit signals the capture register to load the current counter value if the T1 EDGE DET ENA bit is set. The T1EDGE POLARITY bit determines which edge of the signal on the T1IC/CR pin to detect.

The input detect function is enabled by the T1EDGE DET ENA bit, with T1EDGE POLARITY selecting the active input transition. In the capture/ compare mode, the edge detect function, once enabled, remains enabled following a valid edge detect.

7.5 Clock Prescaler/External Clock Source

A prescaler is a circuit that slows the rate of a clocking source to a counter. This block, illustrated in Figure 7–4, allows the selection of the clock inputs (sources) to the general-purpose counter and to the WD counter independently. Each counter has three bits in the T1CTL1 register (see subsection 7.9.1, page 7-32) that determine whether the counter is clocked by one of the prescaled system clock values or by the external clock source (T1EVT).

Figure 7–4. T1 System Clock Prescaler



† For the hard WD configuration of the mask-ROM device, the clock source comes only from one of the four taps from the prescaler that provide a system clock divided by 4, 16, 64, or 256.

The counter clock sources can be any of the following:

- ❑ A system clock with no prescale
- ❑ No clock (the counter is stopped)
- ❑ An external source that is synchronized with the system clock (event counter operation)
- ❑ A system clock while the external input is high (pulse accumulation)
- ❑ One of four taps from the prescaler that provide a system clock divided by 4, 16, 64, or 256

The external clock input to the module (T1EVT) must not exceed SYSCLK/2. If the application does not require the external clock, the T1EVT pin can be re-configured as a digital I/O pin.

The event input is not routed through the prescaler, so the T1 module can use different taps of the prescaler for T1 and the WD timer.

The maximum counter duration when the internal clock is used is determined by the internal system clock time (SYSCLK) and the prescale tap. These relationships are shown below:

Maximum Counter Duration (seconds)	=	$2^{16} \times PS \times (1/SYSCLK)$
Counter Resolution	=	$PS \times (1/SYSCLK)$
where: SYSCLK	=	CLKIN/4 for divide-by-4 clock
	=	CLKIN/1 for divide-by-1 clock
PS	=	1 for no prescale
	=	4 for divide by 4
	=	16 for divide by 16
	=	64 for divide by 64
	=	256 for divide by 256

Table 7–5 gives the real-time counter overflow rates for various crystal and prescaler values.

Software can configure the overflow rates for the WD counter as shown in Table 7–5 or as the value shown divided by two if the WD OVRFL TAP SEL bit (T1CTL1.7) is set (see Section 7.7 on page 7-21). This bit configures the WD counter as either a 15-bit counter when set or a 16-bit counter when cleared.

Table 7–5. Counter Overflow Rates

Register T1CTL1 Bits				SYSCLK Frequency (MHz)			
				0.5§	1§	2.5	5
				System Clock Period (ns)			
SELECT2	SELECT1	SELECT0	Divide By	2000	1000	400	200
0	0	0	2^{16}	0.131†	0.066	0.026	0.013
0	0	1	(P.A.)	‡	‡	‡	‡
0	1	0	(Event)	‡	‡	‡	‡
0	1	1	(Stop)	‡	‡	‡	‡
1	0	0	2^{18}	0.524	0.262	0.105	0.052
1	0	1	2^{20}	2.10	1.05	0.419	0.210
1	1	0	2^{22}	8.39	4.19	1.68	0.839
1	1	1	2^{24}	33.6	16.8	6.71	3.355

† Time is given in seconds.
‡ Not applicable.
§ Divide-by-1 clock can operate only from a minimum of 2 MHz SYSCLK to a maximum of 5 MHz SYSCLK.

7.5.1 Event Counter Mode

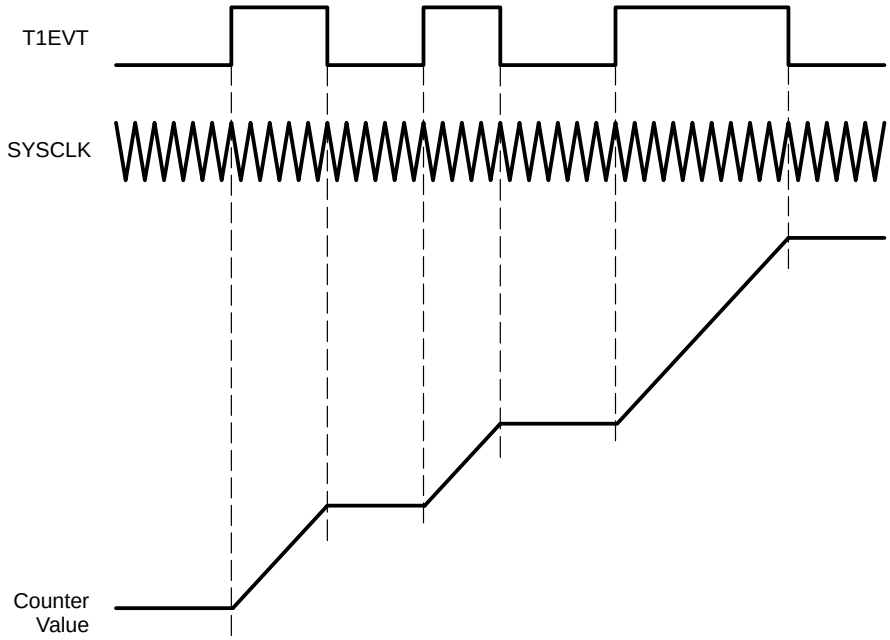
When you use the event counter clock source, the 16-bit counter is programmable as a 16-bit event counter. An external high-to-low transition on the T1EVT pin provides the clock for the internal timer.

7.5.2 Pulse Accumulator Mode

When you use the pulse accumulator clock source, the 16-bit counter is programmable as a 16-bit pulse accumulator. An external input on the T1EVT pin is used to gate the internal system clock to the internal timers. While T1EVT input is logic one (high), the timer is clocked at the system clock rate and counts system clock pulses until the T1EVT pin returns to logic zero.

The pulse accumulator mode keeps a cumulative count of SYSCLK pulses gated by the T1EVT signal as shown in Figure 7–5.

Figure 7–5. Pulse Accumulation



7.6 Interrupts

In dual compare mode, any of the following four separate events can generate an interrupt:

- Compare equal from compare register 2 if the T1C2 INT ENA bit (T1CTL3.1) is set
- Compare equal from compare register 1 if the T1C1 INT ENA bit (T1CTL3.0) is set
- Counter overflow if the T1 OVRFL INT ENA bit (T1CTL2.4) is set
- Edge detect is set if the T1EDGE INT ENA bit (T1CTL3.2) is set

In the capture/compare mode, any of the following three separate events can generate an interrupt:

- Compare equal if the T1C1 INT ENA bit (T1CTL3.0) is set
- Counter overflow if the T1 OVRFL INT ENA bit (T1CTL2.4) is set
- Input capture acknowledge if the T1EDGE INT ENA bit (T1CTL3.2) is set

Note:

All set and enabled interrupt flags must be cleared before the processor exits the T1 interrupt routine. If the flags are not reset, the processor enters the T1 interrupt routine again before continuing with the mainstream program. If the flag bits are never reset, the program continually enters the interrupt service routine.

7.7 WD Timer

The WD timer can be configured as one of the three different mask options.

- A **standard WD** (see note below):
  - For ROMless devices with revision A:
    TMS370C1xxA and TMS370C2xxA
  - For Mask-ROM devices with revision A:
    TMS370C0xxA, TMS370C3xxA, and TMS370C4xxA
  - For EPROM devices with revision A:
    TMS370C6xxA, TMS370C7xxA, SE370C6xxA, and SE370C7xxA
- A **hard WD** (see note below):
  - For Mask-ROM devices with revision A:
    TMS370C0xxA, TMS370C3xxA, and TMS370C4xxA
  - For EPROM devices with revision B:
    TMS370C7xxB and SE370C7xxB
- A **simple counter** (see note below) for mask-ROM devices with revision A: TMS370C0xxA, TMS370C3xxA, and TMS370C4xxA

Note: Hard WD and Simple Counter Advantages

The hard WD and simple counter options provide an improvement to the WD counter circuitry: the hard WD option enables the WD counter reset ability at all times and the simple counter option disables the WD counter reset ability at all times.

The additional simple counter option is available only on TMS370CxxxA devices. The hard WD option is available on both TMS370CxxxA and TMS370C7xxB devices. Refer to Section A.1, page A-2, for the differences between each TMS370Cxxx device.

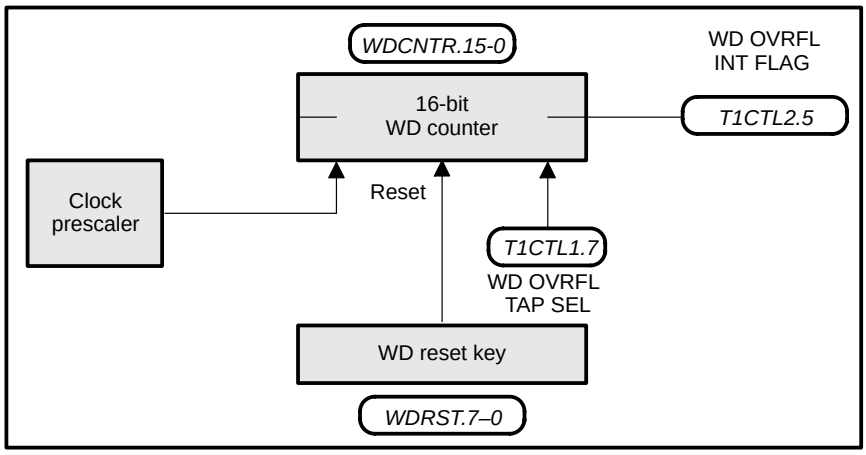
ROM devices with revision A can be configured with any of the three options listed above. All ROMless devices with revision A are configured as a standard WD. All EPROM devices with revision A are configured as a standard WD, and EPROM devices with revision B are configured as a hard WD.

The WD timer, shown in Figure 7–6, consists of the following blocks:

- A 16-bit, resettable WD/event counter that provides up to 2^{24} clock cycles between counter overflows, depending on the prescaler tap used. The program can read the contents of this counter at locations P046 (WD counter MSbyte) and P047 (WD counter LSbyte) in the peripheral file.

- A prescaled clock input selection or external clock that functions the same as in the general-purpose timer (see Section 7.2, on page 7-6).
- A WD reset key that is used to reset the WD counter (WDRST–P048).
- An overflow flag that is set whenever the WD counter overflows.

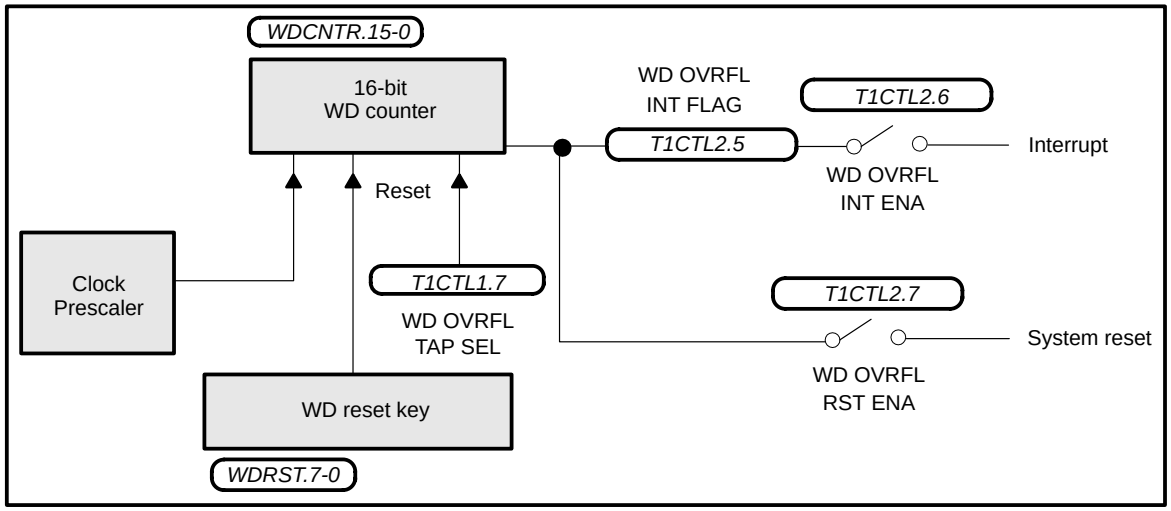
Figure 7–6. WD Timer



7.7.1 Standard WD Configuration

The standard WD can be configured as either a WD or as a simple counter through setting or clearing the WD OVRFL RST ENA bit (T1CTL2.7) in the software. Figure 7–7 illustrates the block diagram of the standard WD.

Figure 7–7. Standard WD Block Diagram



The standard WD can be configured in one of two modes: WD mode or non-WD mode.

7.7.1.1 WD Mode

In the WD mode (WD OVRFL RST ENA = 1), the WD timer generates a system reset if the counter overflows or if the WD counter is reinitialized by an incorrect value; a system reset pulls the $\overline{\text{RESET}}$ pin low for eight system clock cycles. The required reinitialization frequency is determined by the system clock frequency, the prescaler/clock source selected, and whether the WD OVRFL TAP SEL bit (T1CTL1.7) is set for 15- or 16-bit counter rollover.

The WD overflow times are the same as those given in Table 7–5, page 7-18, when the timer is configured as a 16-bit counter (WD OVRFL TAP SEL = 0). Divide the times in Table 7–5 in half when the timer is configured as a 15-bit counter (WD OVRFL TAP SEL =1).

With a 5-MHz SYSCLK, the WD-counter overflow times range from 6.55 ms to 3.35 seconds. These values are selected before the timer enters the WD mode because once the software enables the WD reset function (WD OVRFL RST ENA = 1), subsequent writes to these control bits are ignored. Writes to these WD control bits can occur only following a reset.

To reinitialize the WD counter, write a predefined value to the WD reset key (WDRST) located in the peripheral file at P048. The correct reset key alternates between 55h and AAh, beginning with 55h following the enable of the WD reset function. Writes of the correct values must occur before the timer overflow period.

A write of any value other than the correct predefined value to the WD reset key is interpreted as a lost program, and a system reset is initiated. A WD-counter overflow or incorrect reset key sets the WD OVRFL INT FLAG bit (T1CTL2.5) to 1. The program can read this flag after a reset to determine the source of the reset. WD resets are not prevented when the flag is set.

Note:

A standard WD is disabled in low-power mode (see Section 7.8 on page 7-29).

The routine in Example 7–1 initializes the WD in the standard WD mode to generate a system reset when the counter overflows. The watchdog counter is set to 16 bits in length, and the full 8-bit prescale tap is used.

Example 7–1. Standard WD Initialization

```
.
.
;Set up WD timer for a 24-bit countdown time.
;
OR    #70h,P049    ; Set the WD overflow tap to 16 bits
                   ; and select the /256 prescale value
OR    #0C0h,P04A   ; WD timer reset is enabled along
                   ; with enabling the WD timer
                   ; interrupt.
.
.
The WD timer has now been initialized to cause a system
reset if the counter is not reset before reaching FFFFh.
To reset the counter, the code must write an alternating
55h and AAh, starting with 55h, to the WD timer reset key
register (P048), such as:
.
.
MOV   #55h,P048    ; First write to WD RESET KEY
.
.
MOV   #0AAh,P048   ; Next write to WD RESET KEY
.
.
MOV   #55h,P048    ; Next write to WD RESET KEY
.
.
```

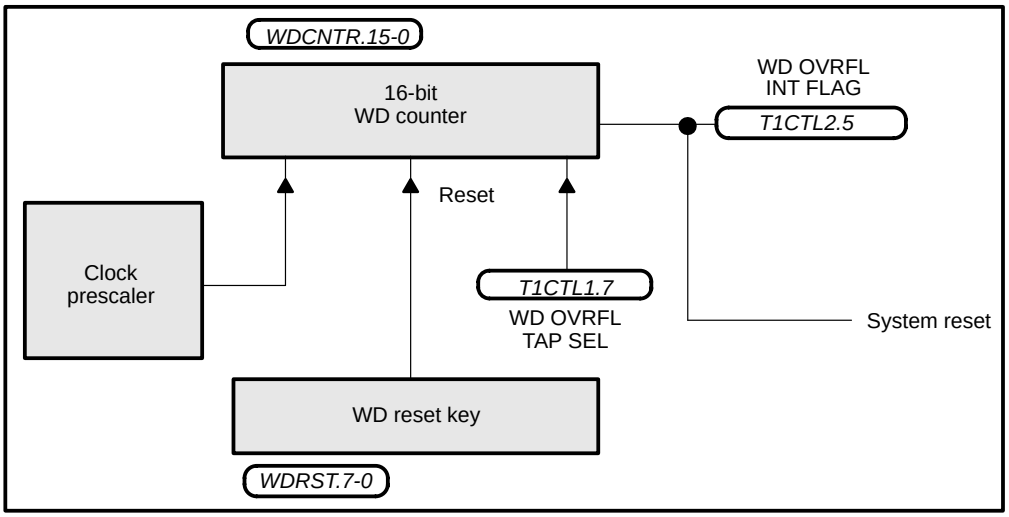
7.7.1.2 Non-WD Mode

In the non-WD mode (WD OVRFL RST ENA bit = 0), the WD counter can be used as an event counter, a pulse accumulator, or an interval timer. In this mode, the system reset function is disabled; to reinitialize the WD counter, write any value to the WD reset key (WDRST). In realtime control applications, the timer overflow rates are determined by the system clock frequency, the prescaler/clock source value selected, and the value of the WD OVRFL TAP SEL bit. If the WD counter is not reset before overflowing, the counter rolls over to either 0000h or 8000h, as determined by the WD OVRFL TAP SEL bit, and continues counting. Upon counter overflow, the WD OVRFL INT FLAG bit is set, and a timer interrupt is generated if the WD OVRFL INT ENA bit is set. Alternately, an external input on the T1EVT pin can be used with the WD timer to provide an additional 16-bit event counter or pulse accumulator.

7.7.2 Hard WD Configuration

In the hard WD configuration, you can operate the WD timer only as a WD. Upon the powerup reset, the hard WD is enabled, and the WD INPUT SELECT0–1 bits (T1CTL1.4–5) are cleared (system clock/4). Figure 7–8 is a block diagram of the hard WD.

Figure 7–8. Hard WD Block Diagram



The hard WD provides additional system integrity. If the counter overflows or if the WD timer is reinitialized by an incorrect value, the hard WD generates a system reset, which pulls the $\overline{\text{RESET}}$ pin low for eight system clock cycles. The required reinitialization frequency is determined by the system clock fre-

quency, WD INPUT SELECT0 and 1 (T1CTL1.4 and T1CTL1.5), the prescaler/clock source selected, and whether the WD OVRFL TAP SEL bit is set for 15- or 16-bit counter rollover. The WD INPUT SELECT2 bit (T1CTL1.6) is functionally interpreted as 1 at all times.

The WD INPUT SELECT0–1 bits and WD OVRFL TAP SEL bit can be modified at any time. Your program should reinitialize these bits after a reset, and periodically thereafter, to ensure a corrected counter overflow rate and to protect against any hardware or software corruptions.

The WD timer is reinitialized by writing a predefined value to the WD reset key (WDRST) located in the peripheral file at P048. The correct reset key alternates between 55h and AAh, beginning with 55h following a system reset. Writes of the correct value must occur before the timer overflow period, or the WD generates a reset.

A write to the WD reset key of any value other than the correct predefined value is interpreted as a lost program, and a system reset is initiated. A WD-counter overflow or incorrect reset key sets the WD OVRFL INT FLAG bit (T1CTL2.5) to 1. After a reset, the program can read this flag to determine the source of the reset. WD resets are not prevented when the flag is set.

Note:

A hard WD is disabled in low-power mode (see Section 7.8 on page 7-29).

7.7.2.1 INT1 Operation During an Inadvertent Low-Power Mode

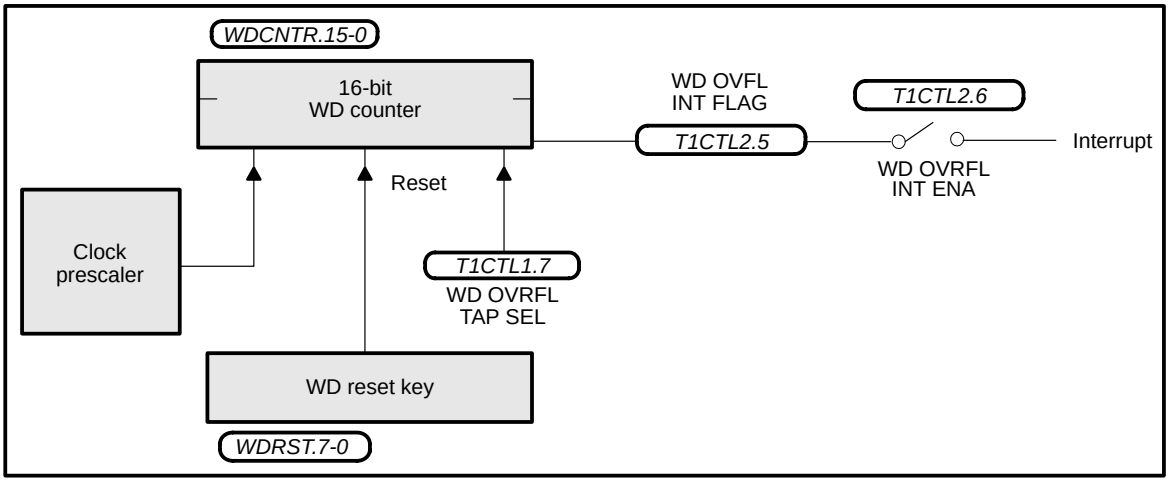
When the hard WD mask option is selected, INT1 is enabled as a non-maskable interrupt (NMI) during low-power modes. This NMI is generated regardless of the interrupt enable flags and the values of the following status bits: INT1 PRIORITY bit (INT1.1), INT1 ENABLE bit (INT1.0), INT1 NMI bit (SCCR2.1), and the global interrupt enable flags in the status register (IE1 and IE2).

INT1 is configured as an NMI in the hard WD to provide a method of exiting a low-power mode. Normally, when the halt or standby mode is entered, the WD counter clock source is disabled, which disables the ability of the WD counter to generate a reset. Note that an active edge on the NMI INT1 pin brings the device out of the low-power mode, and the WD counter is activated. Additionally, if the halt or standby mode is entered while INT1 pin is active (low if the INT1 POLARITY bit is cleared to 0 or high if the INT1 POLARITY bit is set to 1), an NMI is generated immediately.

7.7.3 Simple Counter Configuration

In the simple counter configuration, the WD timer can be used as an event counter, a pulse accumulator, or an interval timer (similar to the non-WD mode in the standard WD configuration). However, in this configuration, the system reset function of the WD timer is *disabled.* Figure 7–9 is a block diagram of a simple counter.

Figure 7–9. Simple Counter Block Diagram



To reinitialize the WD counter, write any value to the WD reset key (WDRST). The timer overflow rates are determined by the system clock frequency, the WD INPUT SELECT0–2 bits (T1CTL1.4–6), and the value of the WD OVRFL TAP SEL bit (T1CTL1.7). If the WD OVRFL RST ENA bit is set to 1, subsequent writes to WD INPUT SELECTs and WD OVRFL TAP SEL bits are ignored. Once the WD OVRFL RST ENA bit is set, these control bits can be changed only after a powerup reset.

7.7.4 Summary of WD Options

Table 7–6 summarizes the features of each watchdog option and specifies the options available to ROMless, mask-ROM, and EPROM devices.

Table 7–6. WD Option Summary

	Standard WD			
Option	**WD**	**Non-WD**	**Hard WD**	**Simple Counter**
WD OVRFL TAP SEL bit and WD INPUT SELECT0–2 bits	Once the WD OVRFL RST ENA is set, the values of these bits can be changed only after a system reset.	These bits can be changed at any time, as long as WD OVRFL RST ENA is not set.	The values of these WD bits can be changed at any time, even if WD OVRFL RST ENA bit is set. However, the WD INPUT SELECT2 bit is not available.	Once the WD OVRFL RST ENA is set, the values of these bits can be changed only after a system reset.
Generates an interrupt when the WD counter overflows?	No	Yes	No	Yes
Generates a system reset?	Yes	No	Yes	No
WD OVRFL RST ENA bit	Select to be a WD. If bit=1, WD counter does initiate a reset upon overflow. This bit is cleared by any system reset.	Select to be a non-WD. If bit = 0, WD counter does not initiate reset upon overflow.	This bit is ignored.	If bit=0, WD bits and WD OVRFL TAP SELECT are not locked. If bit=1, WD bits and WD OVRFL TAP SELECT are locked.
INT1 during low-power modes	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).	Enabled as an NMI.	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).
Available Devices	All devices	All devices	Mask-ROM and EPROM devices	Mask-ROM devices

7.8 Low–Power Modes

The T1 module supports low-power (powerdown) modes that aid in reducing power consumption during periods of inactivity. These modes are the halt and the standby modes. For more information on low-power modes, see Section 4.2, page 4-7.

7.8.1 Halt Mode

The halt mode is entered when the CPU executes an IDLE instruction while the HALT/STANDBY bit (SCCR2.7) and the PWRDWN/IDLE bit (SCCR2.6) are set (the SCCR2 register is described in detail in subsection 4.3.3, page 4-16). During the halt mode, all T1 module functions (including the WD timer) hold the prehalt status of all other storage elements.

The module holds the state of each external pin constant, regardless of whether the pins are used as timer pins or as dedicated I/O pins. That is, inputs remain inputs, output low levels remain low, and output high levels remain high.

7.8.2 Standby Mode

You can put the timer in standby mode by executing an IDLE instruction when the PWRDWN/IDLE (SCCR2.6) bit is set and the HALT/STANDBY bit (SCCR2.7) is cleared. During the standby mode, the WD counter clock input is halted while the rest of the T1 module remains fully functional.

7.9 T1 Control Registers

Seven registers control the configuration of T1 global functions, prescale values, WD timing, optional uses for the associated I/O pins, and other counter functions (refer to Figure 7–10). The bits that are shown in *shaded boxes* are privilege mode bits; that is, they can be written to only in the privilege mode.

Note: 16-bit Register Read/Write Protocol

Special circuitry prevents 16-bit registers from changing in the middle of a 16-bit read or write operation. When you read a 16-bit register, read the least significant byte (LSbyte) first to lock in the value, and then read the most significant byte (MSbyte). When you write to a 16-bit register, write the MSbyte first and then write the LSbyte. The register value does not change between reading and writing the bytes when they are done in this order. While you are reading or writing to a 16-bit register, do not read or write from a second 16-bit register within this module until the process is complete with the first register. Otherwise, the correct value for the first register's MSbyte will not be correct. The 16-bit read/write operation actually occurs when you access the LSbyte. In summary, the order of read/write operations is as follows:

Read: **LSbyte then MSbyte**
Write: **MSbyte then LSbyte**

Figure 7–10. Peripheral File Frame 4: T1 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTR	1040h	P040	Bit 15	T1 Counter MSbyte						Bit 8
T1CNTR	1041h	P041	Bit 7	T1 Counter LSbyte						Bit 0
T1C	1042h	P042	Bit 15	Compare Register MSbyte						Bit 8
T1C	1043h	P043	Bit 7	Compare Register LSbyte						Bit 0
T1CC	1044h	P044	Bit 15	Capture/Compare Register MSbyte						Bit 8
T1CC	1045h	P045	Bit 7	Capture/Compare Register LSbyte						Bit 0
WDCNTR	1046h	P046	Bit 15	WD Counter MSbyte						Bit 8
WDCNTR	1047h	P047	Bit 7	WD Counter LSbyte						Bit 0
WDRST	1048h	P048	Bit 7	WD Reset Key						Bit 0
T1CTL1	1049h	P049	WD OVRFL TAP SEL † (RP–0)	WD INPUT SELECT2† (RP–0)	WD INPUT SELECT1† (RP–0)	WD INPUT SELECT0† (RP–0)	—	T1 INPUT SELECT2 (RW–0)	T1 INPUT SELECT1 (RW–0)	T1 INPUT SELECT0 (RW–0)
T1CTL2	104Ah	P04A	WD OVRFL RST ENA † (RS–0)	WD OVRFL INT ENA (RW–0)	WD OVRFL INT FLAG (RC–*)	T1 OVRFL INT ENA (RW–0)	T1 OVRFL INT FLAG (RC–0)	—	—	T1 SW RESET (S–0)
T1CTL3	104Bh	P04B	Dual Compare Mode							
			T1EDGE INT FLAG (RC–0)	T1C2 INT FLAG (RC–0)	T1C1 INT FLAG (RC–0)	—	—	T1EDGE INT ENA (RW–0)	T1C2 INT ENA (RW–0)	T1C1 INT ENA (RW–0)
			Capture / Compare Mode							
			T1EDGE INT FLAG (RC–0)	—	T1C1 INT FLAG (RC–0)	—	—	T1EDGE INT ENA (RW–0)	—	T1C1 INT ENA (RW–0)
T1CTL4	104Ch	P04C	Dual Compare Mode							
			T1 MODE = 0 (RW–0)	T1C1 OUT ENA (RW–0)	T1C2 OUT ENA (RW–0)	T1C1 RST ENA (RW–0)	T1CR OUT ENA (RW–0)	T1EDGE POLARITY (RW–0)	T1CR RST ENA (RW–0)	T1EDGE DET ENA (RW–0)
			Capture / Compare Mode							
			T1 MODE = 1 (RW–0)	T1C1 OUT ENA (RW–0)	—	T1C1 RST ENA (RW–0)	—	T1EDGE POLARITY (RW–0)	—	T1EDGE DET ENA (RW–0)
T1PC1	104Dh	P04D	—	—	—	—	T1EVT DATA IN (R–0)	T1EVT DATA OUT (RW–0)	T1EVT FUNCTION (RW–0)	T1EVT DATA DIR (RW–0)
T1PC2	104Eh	P04E	T1PWM DATA IN (R–0)	T1PWM DATA OUT (RW–0)	T1PWM FUNCTION (RW–0)	T1PWM DATA DIR (RW–0)	T1IC/CR DATA IN (R–0)	T1IC/CR DATA OUT (RW–0)	T1IC/CR FUNCTION (RW–0)	T1IC/CR DATA DIR (RW–0)
T1PRI	104Fh	P04F	T1 STEST (RP–0)	T1 PRIORITY (RP–0)	—	—	—	—	—	—

† Once the WD OVRFL RST ENA bit is set, these bits cannot be changed until a reset occurs; this applies only to the standard WD and to the simple counter. In the hard WD, these bits can be modified at any time; the WD INPUT SELECT2 bit is ignored.

7.9.1 T1 Control Register 1 (T1CTL1)

The T1CTL1 register controls the prescaler inputs to the WD timer and the general-purpose timer.

T1 Control Register 1 (T1CTL1)
[Memory Address 1049h]

Bit #	7	6	5	4	3	2	1	0
P049	WD OVRFL TAP SEL	WD INPUT SELECT2	WD INPUT SELECT1	WD INPUT SELECT0	—	T1 INPUT SELECT2	T1 INPUT SELECT1	T1 INPUT SELECT0
	RP-0	RP-0	RP-0	RP-0		RW-0	RW-0	RW-0

R = Read, W = Write, P = Write protected when WD OVRFL RST ENA=1 (only in standard WD and simple counter configurations), -*n* = Value of the bit after the register is reset

Bit 7 **WD OVRFL TAP SEL.** WD Overflow Tap Select

This bit determines whether the WD counter operates as a 15-bit or a 16-bit counter in the standard WD, hard WD, and simple counter options. The default is the full 16 bits of the counter. If a shorter WD counter overflow rate is needed, then the most significant bit of the counter can be forced to remain at 1. This, in effect, changes the WD counter to a 15-bit counter with an overflow period half that of a 16-bit counter. This tap select feature, combined with the clock prescaler, allows WD overflow rates from 2^{15} to 2^{24} system clock cycles. Once the WD RST ENA bit is set, this bit can be changed only after a reset (for the non-WD mode of the standard WD and simple counter). In the hard WD, this bit can be changed at any time.

0 = 16-bit WD counter overflow
1 = 15-bit WD counter overflow

Bits 6–4 **WD INPUT SELECT2–0.** WD Input Select 2–0

Standard WD and simple counter: These three bits select one of eight possible clock sources. Once the WD OVRFL RST ENA bit is set, the values of these three bits can be changed only after a reset; a write to this bit has no effect when the WD OVRFL RST ENA bit is set.

Hard WD: The WD INPUT SELECT0 and WD INPUT SELECT1 bits are used to select one of the four possible clock sources. The clock sources come from one of the four taps from the 8-bit prescaler, which provides the system clock divided by 4, 16, 64, or 256. Note that the WD INPUT SELECT2 bit is functionally interpreted as 1.

The combinations are shown in Table 7–7.

Table 7–7. Counter Clock Sources for the WD Input Select 0–2

WD INPUT SELECT2	WD INPUT SELECT1	WD INPUT SELECT0	Counter Clock Source
0	0	0	System clock†
0	0	1	Pulse accumulation†
0	1	0	Event input†
0	1	1	No clock input†
1	0	0	System clock/4
1	0	1	System clock/16
1	1	0	System clock/64
1	1	1	System clock/256

† These options are not available for the hard WD

Bit 3 **Reserved.** Read data is indeterminate

Bit 2–0 **T1 INPUT SELECT2–0.** T1 Input Select 2–0

These three bits select one of eight possible clock sources for the T1 general-purpose counter. These sources are as follows:

- The system clock with no prescale (system clock)
- The system clock when the external input T1EVT is high (pulse accumulation)
- An external source synchronized with the system clock (event input)
- No system clock source (no clock input)
- One of four taps from the 8-bit prescaler, which provides the system clock divided by 4, 16, 64, or 256

The combinations are shown in Table 7–8:

Table 7–8. Clock Sources for the T1 General Purpose Counter

T1 INPUT SELECT2	T1 INPUT SELECT1	T1 INPUT SELECT0	Counter Clock Source
0	0	0	System clock
0	0	1	Pulse accumulation
0	1	0	Event input
0	1	1	No clock input
1	0	0	System clock/4
1	0	1	System clock/16
1	1	0	System clock/64
1	1	1	System clock/256

7.9.2 T1 Control Register 2 (T1CTL2)

The T1CTL2 register controls the T1 and WD overflow interrupts and contains the T1 software reset bit.

T1 Control Register 2 (T1CTL2)
[Memory Address 104Ah]

Bit #	7	6	5	4	3	2	1	0
P04A	WD OVRFL RST ENA	WD OVRFL INT ENA	WD OVRFL INT FLAG	T1 OVRFL INT ENA	T1 OVRFL INT FLAG	—	—	T1 SW RESET
	RS-0	RW-0	RC-*	RW-0	RC-0			S-0

R = Read, S = Set only, W = Write, C = Clear only, -*n* = Value of the bit after the register is reset, -* = see bit description

Bit 7 **WD OVRFL RST ENA.** WD Overflow Reset Enable

Note:

This bit operates differently for TMS370Cxxx devices than TMS370CxxxA and TMS370C7xxB devices. Refer to Section A.4, page A-5.

Standard WD: This bit controls the ability of a WD timer to generate a reset. The WD timer is a simple counter pulse accumulator when cleared. Once set, this bit can be cleared only by any system reset and locks the values of other WD bits so that they can be changed only after a reset.

0 = WD counter does *not* initiate a reset upon overflow.
1 = WD counter *does* initiate a reset upon overflow.

Simple counter: This bit protects the WD INPUT SELECT and WD OVRFL TAP SEL bits. Once set, subsequent writes to these control bits are ignored; they can be changed only after reset.

0 = Other WD bits are not protected
1 = Locks the value of other WD bits

Hard WD: This bit is ignored.

Bit 6 **WD OVRFL INT ENA.** Watchdog Overflow Interrupt Enable

This bit controls the WD overflow interrupting capability.

0 = Disables WD interrupt
1 = Enables WD interrupt

Bit 5 **WD OVRFL INT FLAG.** Watchdog Overflow Interrupt Flag

Note:

This bit operates differently for TMS370Cxxx devices than TMS370CxxxA and TMS370CxxB devices. Refer to Section A.4, page A-5.

This bit is set if the last reset is initiated by the WD counter. Setting this bit will not prevent WD resets. This bit is cleared by writing a zero to it or by any system reset that is not initiated by the WD counter.

0 = WD interrupt is inactive.
1 = WD counter has overflowed or the incorrect value is written to the WD reset key register.

Bit 4 **T1 OVRFL INT ENA.** T1 Overflow Interrupt Enable.

This bit controls the T1 overflow interrupting capability.

0 = Disables interrupt
1 = Enables interrupt

Bit 3 **T1 OVRFL INT FLAG.** T1 Overflow Interrupt Flag

This bit indicates the status of the T1 overflow interrupt.

0 = General-purpose overflow interrupt is inactive.
1 = General-purpose overflow interrupt is pending.

Bits 2–1 **Reserved.** Read values are indeterminate.

Bit 0 **T1 SW RESET.** T1 Software Reset

This bit is always read as a 0; however, when a 1 is written to this bit, the counter resets to 0000h on the next system clock cycle.

Note:

Be careful using the AND, OR, XOR, CMPBIT, SBIT0, or SBIT1 instructions to modify this register. The read/modify/write nature of these instructions can inadvertently clear an interrupt flag that was set between the read and the write cycles. If the state of the interrupt enable bits is known, the MOV #iop8, Pd instruction can be used. If the state of the interrupt enable bits is not known, a sequence similar to the example shown below should be used.

```
;clearing the T1 OVRFL INT FLAG:
      MOV   P04A,A
      OR    #028H,A
      AND   #0F7H,A
      MOV   A,P04A
```

7.9.3 T1 Control Register 3 (T1CTL3)

The T1CTL3 register controls the edge-detect and compare interrupts. The six active bits in this register serve different functions for each mode, as shown below:

T1 Control Register 3 (T1CTL3)
[Memory Address 104Bh]

Mode: Dual Compare

Bit #	7	6	5	4	3	2	1	0
P04B	T1EDGE INT FLAG	T1C2 INT FLAG	T1C1 INT FLAG	—	—	T1EDGE INT ENA	T1C2 INT ENA	T1C1 INT ENA
	RC-0	RC-0	RC-0			RW-0	RW-0	RW-0

Mode: Compare/Capture

Bit #	7	6	5	4	3	2	1	0
P04B	T1EDGE INT FLAG	—	T1C1 INT FLAG	—	—	T1EDGE INT ENA	—	T1C1 INT ENA
	RC-0		RC-0			RW-0		RW-0

R = Read, W = Write, C = Clear only, -*n* = Value of the bit after the register is reset

Bit 7 **T1EDGE INT FLAG.** T1 Edge Interrupt Flag

This bit indicates when an external pulse transition of the correct polarity is detected on the T1 input capture/counter reset (T1IC/CR) pin. This bit also indicates an input capture in the capture/compare mode.

0 = No transition
1 = Transition detected

Bit 6 **T1C2 INT FLAG.** T1 Compare 2 Interrupt Flag

Dual compare mode: This bit is set when the capture/compare register first matches the counter value.

0 = Interrupt inactive
1 = Interrupt pending

Capture/compare mode: Reserved. Read data is indeterminate.

Bit 5 **T1C1 INT FLAG.** T1 Compare 1 Interrupt Flag

This bit is set when the compare register first matches the counter value.

0 = Interrupt inactive
1 = Interrupt pending

Bit 4–3 **Reserved.** Read data is indeterminate.

Bit 2 **T1EDGE INT ENA.** T1 Edge Interrupt Enable

This bit determines whether or not the active edge input to the T1IC/CR pin generates an interrupt. The T1EDGE DET ENA bit (T1CTL4.0) must be set before an edge can be detected.

0 = Disables interrupt
1 = Enables interrupt

Bit 1 **T1C2 INT ENA.** T1 Compare 2 Interrupt Enable

Dual compare mode only: This bit determines whether or not the capture/compare register flag can generate an interrupt.

0 = Disables interrupt
1 = Enables interrupt

Capture/compare mode: Reserved. Read data is indeterminate.

Bit 0 **T1C1 INT ENA.** T1 Compare 1 Interrupt Enable

This bit determines whether or not the compare register flag can generate an interrupt.

0 = Disables interrupt
1 = Enables interrupt

Note:

Be careful using the AND, OR, XOR, CMPBIT, SBIT0, or SBIT1 instructions to modify this register. The read/modify/write nature of these instructions can inadvertently clear an interrupt flag that was set between the read and the write cycles. If the state of the interrupt enable bits is known, the MOV #iop8, Pd, instruction can be used. If the state of the interrupt enable bits is not known, a sequence similar to the example shown below should be used.

```
;Clearing the T1C1 INT FLAG
      MOV   P04B,A
      OR    #0E0h,A
      AND   #0DFh,A
      MOV   A,P04B
```

7.9.4 T1 Control Register 4 (T1CTL4)

The T1CTL4 register controls the mode of operation and various functions of the T1 input and output pins. The bits in this register serve different functions, depending on the mode.

T1 Control Register 4 (T1CTL4)
[Memory Address 104Ch]

Mode: Dual Compare

Bit #	7	6	5	4	3	2	1	0
P04C	T1 MODE=0	T1C1 OUT ENA	T1C2 OUT ENA	T1C1 RST ENA	T1CR OUT ENA	T1EDGE POLARITY	T1CR RST ENA	T1EDGE DET ENA
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Mode: Compare/Capture

Bit #	7	6	5	4	3	2	1	0
P04C	T1 MODE=1	T1C1 OUT ENA	—	T1C1 RST ENA	—	T1EDGE POLARITY	—	T1EDGE DET ENA
	RW-0	RW-0		RW-0		RW-0		RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **T1 MODE.** T1 Mode Select

This bit selects the general-purpose counter mode.

0 = Dual compare mode
1 = Capture/compare mode

Bit 6 **T1C1 OUT ENA.** T1 Output-Compare Output Enable 1

When this bit is set and the compare register 1 is equal to the counter, the T1PWM pin toggles (when configured as a PWM pin).

0 = Disables pulse-to-toggle output
1 = Enables pulse-to-toggle output

Bit 5 **T1C2 OUT ENA.** T1 Output-Compare Output Enable 2

Dual Compare Mode: When this bit is set and compare register 2 is equal to the counter, the T1PWM pin toggles (when configured as a PWM pin).

0 = Disables pulse-to-toggle output
1 = Enables pulse-to-toggle output

Capture/compare mode: Reserved. Read data is indeterminate.

Bit 4 **T1C1 RST ENA.** T1 Compare 1 Reset Enable

When this bit is set and compare register 1 is equal to the counter, the counter will reset on the next counter increment.

0 = Disables counter reset upon compare equal
1 = Enables counter reset upon compare equal

Bit 3 **T1CR OUT ENA.** T1 External Edge Output Enable

Dual compare mode: This bit determines whether the input signal on the T1IC/CR pin can toggle the output signal on the T1PWM pin.

0 = Disables pulse-to-toggle output
1 = Enables pulse-to-toggle output

Capture/compare mode: Reserved. Read data is indeterminate.

Bit 2 **T1EDGE POLARITY.** T1 Edge Polarity

This bit determines the transition direction on the T1IC/CR pin to trigger a capture or counter reset, depending on the counter mode selected.

0 = Triggers on a high-to-low transition
1 = Triggers on a low-to-high transition

Bit 1 **T1CR RST ENA.** T1 External Reset Enable

Dual compare mode: This bit determines whether an external signal can reset the counter.

0 = Disables external reset of the counter
1 = Enables external reset of the counter on the next valid edge detect

Capture/compare mode: Reserved. Read data is indeterminate.

Bit 0 **T1EDGE DET ENA.** T1 Edge Detect Enable

Dual compare mode: This bit enables the edge detection circuit to sense the next level transition on the T1IC/CR pin. This bit is cleared after the selected transition is detected and during a reset.

0 = Disables edge detection
1 = Enables edge detection

Capture/compare mode: This bit enables the input capture circuit to capture the current counter value upon the next level transition on the counter reset/input capture pin, as determined by the T1EDGE POLARITY bit. This bit remains unchanged after the selected transition is detected.

0 = Disables input capture
1 = Enables input capture

7.9.5 T1 Port Control Registers (T1PC1 and T1PC2)

Port control registers (PCRs) T1PC1 and T1PC2 are organized to allow all functions for a pin to be programmed in one write cycle. Each module pin is controlled by a nibble in one of the PCRs.

7.9.5.1 T1 Port Control Register 1 (T1PC1)

The T1PC1 register controls the I/O functions of the T1 module, T1EVT pin.

T1 Port Control Register 1 (T1PC1)
[Memory Address 104Dh]

Bit #	7	6	5	4	3	2	1	0
P04D	—	—	—	—	T1EVT DATA IN	T1EVT DATA OUT	T1EVT FUNCTION	T1EVT DATA DIR
					R-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bits 7–4 **Reserved.** Read data is indeterminate.

Bit 3 **T1EVT DATA IN.** T1EVT Pin Data In

This bit contains the data present on the T1EVT pin. A write operation to this bit has no effect.

Bit 2 **T1EVT DATA OUT.** T1EVT Pin Data Out

This bit contains the data to be output on the T1EVT pin if the following conditions are met:

a. Bit TIEVT DATA DIR = 1
b. Bit T1EVT FUNCTION = 0

Bit 1 **T1EVT FUNCTION.** T1EVT Pin Function Select

This bit determines the function of the T1EVT pin.

0 = The T1EVT is a general-purpose digital I/O pin.
1 = The T1EVT is the event-input pin.

Bit 0 **T1EVT DATA DIR.** T1 Event-Pin Data Direction

This bit selects the T1EVT pin as an input or output if the T1EVT FUNCTION bit = 0.

0 = Enables T1EVT pin as data input
1 = Enables T1EVT pin as data output

7.9.5.2 T1 Port Control Register 2 (T1PC2)

The T1PC2 register controls the I/O functions of the T1IC/CR and T1PWM pins.

T1 Port Control Register 2 (T1PC2)
[Memory Address 104Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P04E	T1PWM DATA IN	T1PWM DATA OUT	T1PWM FUNCTION	T1PWM DATA DIR	T1IC/CR DATA IN	T1IC/CR DATA OUT	T1IC/CR FUNCTION	T1IC/CR DATA DIR
	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **T1PWM DATA IN.** T1PWM Pin Data In 1

This bit contains the data input on pin T1PWM. A write operation to this bit has no effect.

Bit 6 **T1PWM DATA OUT.** T1PWM Pin Data Out

This bit contains the data to be output on the T1PWM pin if the following conditions are met:

a. Bit T1PWM DATA DIR = 1
b. Bit T1PWM FUNCTION = 0

Bit 5 **T1PWM FUNCTION.** T1PWM Pin Function Select

This bit determines the function of the T1PWM pin.

0 = The T1PWM pin is a general-purpose digital I/O pin.
1 = The T1PWM pin is the PWM output.

Bit 4 **T1PWM DATA DIR.** T1PWM Pin Data Direction

This bit selects the T1PWM pin as an input or output if the T1PWM FUNCTION bit = 0.

0 = Enables T1PWM pin data input
1 = Enables T1PWM pin data output

Bit 3 **T1IC/CR DATA IN.** T1IC/CR Pin Data In

This pin contains the data input on pin T1IC/CR. A write operation to this bit has no effect.

Bit 2 **T1IC/CR DATA OUT.** T1IC/CR Pin Data Out

This bit contains the data output on pin T1IC/CR if the following conditions are met:

a. Bit T1IC/CR DATA DIR = 1
b. Bit T1IC/CR FUNCTION = 0

Bit 1 **T1IC/CR FUNCTION.** T1IC/CR Pin Function Select

This bit determines the function of the T1IC/CR pin.

0 = The T1IC/CR pin is a general-purpose digital I/O pin.
1 = The T1IC/CR pin is the input capture/counter reset pin.

Bit 0 **T1IC/CR DATA DIR.** T1IC/CR Pin Data Direction

This bit selects the T1IC/CR pin as an input or output if the T1IC/CR FUNCTION bit = 0.

0 = Enables T1IC/CR pin data input
1 = Enables T1IC/CR pin data output

7.9.6 T1 Interrupt Priority Control Register (T1PRI)

The T1PRI register controls the level of the T1 interrupt. You can write to this register only in the privilege mode. During normal operation, this is a read-only register.

T1 Interrupt Priority Control Register (T1PRI)
[Memory Address 104Fh]

Bit #	7	6	5	4	3	2	1	0
P04F	T1 STEST	T1 PRIORITY	—	—	—	—	—	—
	RP-0	RP-0						

R = Read, P = Privilege write only, *-n* = Value of the bit after register is reset

Bit 7 **T1 STEST.** T1 STEST

This bit must be cleared (0) to ensure proper operation.

Bit 6 **T1 PRIORITY.** T1 Interrupt Priority Select

This bit determines the level of the interrupt generated by T1.

0 = Interrupts are level 1 (high priority) requests.
1 = Interrupts are level 2 (low priority) requests.

Bits 5–0 **Reserved.** Read data is indeterminate.

Chapter 8

Timer 2A (T2A) and Timer 2B (T2B) Modules

This chapter discusses the architecture and programming of the T2A and T2B modules and covers the following topics:

Topic **Page**

8.1 T2n Overview

The T2n module (T2A or T2B) is a 16-bit general-purpose timer. Depending on the TMS370 device, there may be 0, 1, or 2 on-chip T2 modules. For devices with one T2 module, the timer is referred to as T2A. Devices (’x6x) with two T2 modules are referred to as T2A and T2B. The term T2n is used to refer to the T2A or T2B timer modules throughout this chapter. The T2n module is composed of a 16-bit resettable counter, a 16-bit compare register with associated compare logic, a 16-bit capture register, and a 16-bit register that functions as a capture register in one mode and as a compare register in the other mode. The T2n module adds an additional timer that provides event count, input capture, and compare functions. The T2n solutions to different system requirements are shown in Table 8–1.

Table 8–1. System-Requirement Solutions Using T2n

Requirement	Timer Solution
Realtime system control	Interval timers with interrupts
Input pulse-width measurement	Pulse accumulate or input capture functions
External event synchronization	Event count function
Timer output control	Compare function
PWM output control	PWM output function

8.1.1 Physical Description

The T2n module has the following features. The T2n module is shown in Figure 8–1 on page 8-3:

- ❑ A 16-bit resettable counter
- ❑ A 16-bit compare register with associated compare logic
- ❑ A 16-bit capture register
- ❑ A 16-bit capture/compare register
- ❑ Selectable edge-detection circuitry
- ❑ Interrupts

  The T2n module has maskable interrupts for the following:

  - Two input captures
  - Two output compares
  - Counter overflow
  - External edge detect

❑ I/O Pins

The T2n module has three I/O pins that can be dedicated as timer functions or used as general-purpose I/O pins. The following list describes each I/O pin:

- T2nEVT, which provides for input to the event counter or the external clock source
- T2nIC1/CR, which provides for input to the counter reset, input capture, or pulse-width modulation (PWM) circuit
- T2nIC2/PWM, which provides for PWM output or a second input capture

The definitions of these pins are contained in the two port control registers located at addresses P06E and P06D of peripheral file frame 6 for T2A, and P08E and P08D of peripheral file frame 8 for T2B. Table 8–2 on page 8-4 defines the functions of the three T2n I/O pins for both operating modes.

Figure 8–1. T2n Block Diagram

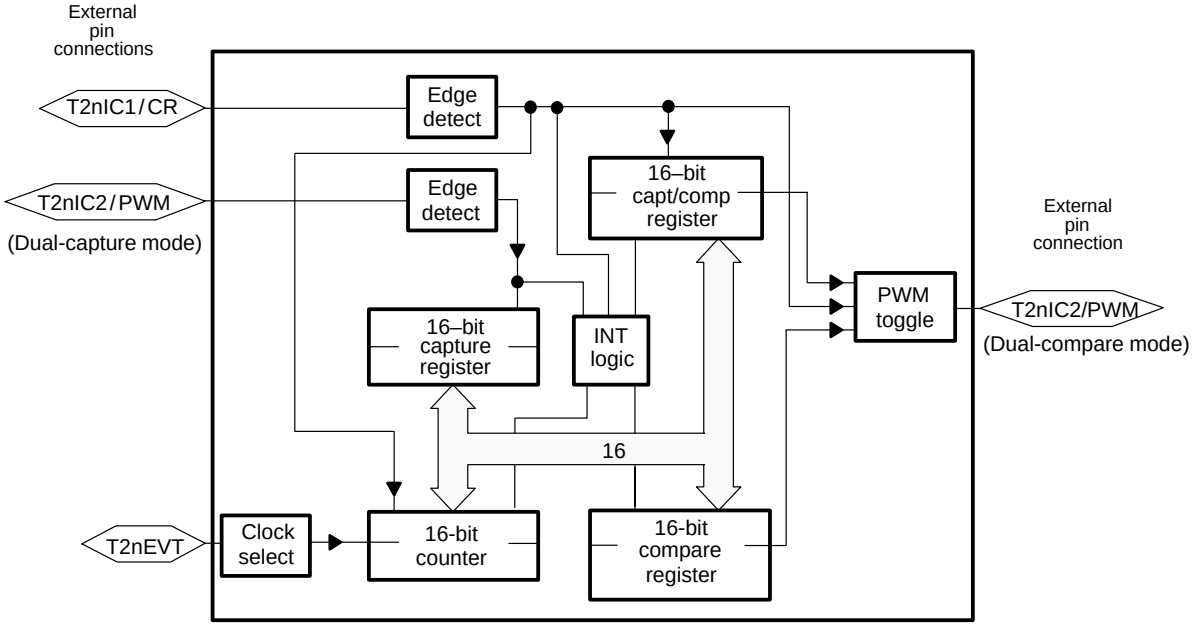


Table 8–2. T2n I/O Pin Definitions

Pin	Dual Compare Mode	Dual Capture Mode
T2nIC1/CR	Counter reset input	Input capture 1 input
T2nIC2/PWM	PWM output	Input capture 2 input
T2nEVT	External event input or pulse accumulate input	External event input or pulse accumulate input

8.1.2 Operating Modes

- **Dual compare mode.** The timer is configured to provide dual compare registers, an external or software reset of the counter, an internal or external clock source, and a programmable PWM output. The T2nIC2/PWM pin can also be configured to toggle upon an external input edge. The external clock source can be selected for use as an event counter or pulse accumulator.
- **Dual capture mode.** The timer is configured to provide dual input capture registers and one compare register for use as a general-purpose timer. The compare register can provide periodic interrupts to the rest of the microcomputer. Each capture register can be configured to capture the current counter value upon either edge of an external input.

8.1.3 Control Registers

The T2A control registers are located at addresses 1060h to 106Fh, with locations 1068h and 1069h reserved; T2B control registers are located at 1080h to 108Fh, with locations 1088h and 1089h reserved. The functions of these locations are shown in Table 8–3 on page 8-5.

Table 8–3. T2n Memory Map

Peripheral File 6 Location T2A	Peripheral File 8 Location T2B	Symbol	Name	Description
P060	P080	T2nCNTR	T2n Counter — MSbyte	16-bit resettable counter
P061	P081		T2n Counter — LSbyte	
P062	P082	T2nC	Compare Register — MSbyte	16-bit compare register
P063	P083		Compare Register — LSbyte	
P064	P084	T2nCC	Capture/Compare Register — MSbyte	16-bit capture/compare register
P065	P085		Capture/Compare Register — LSbyte	
P066	P086	T2nIC	Capture Register — MSbyte	16-bit capture register
P067	P087		Capture Register — LSbyte	
P068	P088		Reserved	
P069	P089		Reserved	
P06A	P08A	T2nCTL1	T2n Control Register 1	Controls the clock input selection, counter overflow interrupts, and counter software reset.
P06B	P08B	T2nCTL2	T2n Control Register 2	Contains interrupt flags and controls the module's capability to issue interrupts.
P06C	P08C	T2nCTL3	T2n Control Register 3	Controls the mode of operation, outputs, active transition polarity, and counter reset.
P06D	P08D	T2nPC1	T2n Port Control Register 1	Assigns the I/O function of the T2nEVT pin as either a general-purpose digital I/O or external event input of the module.
P06E	P08E	T2nPC2	T2n Port Control Register 2	Assigns the I/O functions of the T2nIC1/CR and T2nIC2/PWM pins as either general-purpose digital I/O pins or the input capture/counter reset and PWM output pins, respectively.
P06F	P08F	T2nPRI	T2n Interrupt Priority Control Register	Assigns the priority level of interrupts generated by the T2n module.

8.2 T2n Components

The T2n module uses a 16-bit counter, a compare register, a capture register, and a capture/compare register to provide event, compare, and capture functions.

8.2.1 16-Bit Resettable Counter

The 16-bit free-running, read-only counter (T2nCNTR) is clocked by the system clock, the external event, or the system clock during the occurrence of an active external event (pulse accumulate).

- ❏ During initialization, the counter is loaded with 0000h and begins its count.
- ❏ If the counter is not reset before reaching FFFFh, the counter rolls over to 0000h and continues counting. When the counter rolls over, the T2n OVRFL INT FLAG bit (T2nCTL1.3) is set; a timer interrupt is generated if the T2n OVRFL INT ENA bit (T2nCTL1.4) is set.
- ❏ The counter can be reset to 0000h during counting by any of the following:
  - A 1 written to the T2n SW RESET bit (T2CTL1.0)
  - A compare equal condition from the dedicated T2n compare function
  - System reset
  - An external pulse on the T2nIC1/CR pin (dual compare mode only) if the T2nC1 RST ENA bit (T2nCTL3.4) is set.

To reset the counter, you can select the external transition on the T2nIC1/CR pin to be either low-to-high or high-to-low. To do this, use the T2nEDGE1 POLARITY bit (T2nCTL3.2).

Special circuitry prevents the contents of the T2nCNTR register from changing in the middle of a 16-bit read operation. See the note in Section 8.8 on page 8-17.

8.2.2 Compare Register

The compare register circuit consists of a 16-bit wide, read/write data register (T2nC) and logic to compare the counter's current value with the value stored in the compare register.

When the counter value matches the compare register value, the circuit performs the following:

- ❏ Sets the T2nC1 INT FLAG bit (T2CTL2.5) to 1.
- ❏ Generates a T2n interrupt if the T2nC1 INT ENA bit (T2nCTL2.0) is set.
- ❏ Resets the counter if the T2nC1 RST ENA bit (T2nCTL3.4) is set.
- ❏ Toggles the PWM output pin if the T2nC1 OUT ENA bit (T2nCTL3.6) is set (dual compare mode only).

Once the T2nC1 INT FLAG bit is set by a compare-equal condition and then cleared, it will not be set again if the same compare-equal condition still exists (that is, the same compare-equal condition can set the T2nC1 INT FLAG bit only once). This flag causes various events to occur, depending on the mode of operation and on which enable bits are set.

Special circuitry prevents the T2nC register from changing in the middle of a 16-bit read or write operation. See the note in Section 8.8.

The compare register value required for a specific timing application can be calculated using the following formula:

$$Compare\ Value = \left(\frac{t}{\frac{1}{SYSCLK}} \right) - 1$$

where:

t = desired timer compare period (seconds)
SYSCLK = CLKIN/4 for divide-by-4 clock (external clock frequency)
CLKIN/1 for divide-by-1 clock

Table 8–4 provides some sample compare register values to achieve various desired timings with a 5-MHz SYSCLK.

Table 8–4. T2n Compare Values: (5-MHz SYSCLK)

Time		T2n Compare Register		% Error (See Note)
Seconds	**m Seconds**	**Decimal**	**Hex**	
0.0005	0.5	2499	009C3h	0.0000
0.001	1	4999	01387h	0.0000
0.002	2	9999	0270Fh	0.0000
0.005	5	24999	061A7h	0.0000
0.010	10	49999	0C34Fh	0.0000
0.013	13	64999	0FDE7h	0.0000

Note: The percent of error induced by the T2n formula varies, depending on the desired timer compare period and the minimum timer resolution (1/SYSCLK).

8.2.3 Capture Register (Dual Capture Mode Only)

The 16-bit capture register (T2nIC) is a read-only data register. This register captures the counter values when an input capture pulse (pin T2nIC2/PWM) is received. The capture register can be read at the addresses shown in Table 8–5. Writes to this register are ignored, so the capture register retains

the last counter value captured until another input capture pulse loads a new value in the register.

Table 8–5. T2n Capture Register MSbyte and LSbyte Addresses

Timer	MSbyte	LSbyte
T2AIC	P066	P067
T2BIC	P086	P087

On receipt of a capture pulse, the circuit conducts the following:

- Loads the value of the 16-bit counter into the capture register.
- Sets the T2nEDGE2 INT FLAG bit (T2nCTL2.6) to indicate that the capture register has latched the current counter value.
- If the T2nC2 INT ENA bit (T2nCTL2.1) is set, generates an interrupt.

Special circuitry prevents the T2nIC register from changing in the middle of a 16-bit read or write operation. See the note in Section 8.8 on page 8-17.

8.2.4 Capture/Compare Register

The 16-bit capture/compare register (T2nCC) can serve one of two functions, depending on the operating mode. Table 8–6 shows the T2nCC locations.

Table 8–6. T2n Capture/Compare Register MSbyte and LSbyte Addresses

Timer	MSbyte	LSbyte
T2ACC	P064	P065
T2BCC	P084	P085

Special circuitry prevents the T2nCC register from changing in the middle of a 16-bit read or write operation. See the note in Section 8.8 on page 8-17.

8.2.4.1 Dual Compare Mode

In the dual compare mode, the T2nCC register becomes a read/write compare register. It functions exactly as the one described in subsection 8.2.2, on page 8-6, except that T2nCC **cannot** reset the counter.

When the 16-bit counter value matches the capture/compare register value, the circuit conducts the following:

- Sets the T2nC2 INT FLAG bit (T2nCTL2.6) to 1.

- Toggles the PWM output pin if the T2nC2 OUT ENA bit (T2nCTL3.5) is set.
- Generates a T2n interrupt if the T2nC2 INT ENA bit (T2nCTL2.1) is set.

8.2.4.2 Dual Capture Mode

In the dual capture mode, the capture/compare register becomes a read-only capture register. When an external pulse appears on pin T2nIC1/CR, the following events occur if the T2nEDGE1 DET ENA bit (T2nCTL3.0) is set:

- The current counter value is latched into the capture/compare register.
- The T2nEDGE1 INT FLAG bit (T2nCTL2.7) is set.
- Generates an interrupt if the T2nEDGE1 INT ENA bit (T2nCTL2.2) is set.

8.3 Operating Modes

The T2n operating mode is determined by the T2n MODE bit (T2nCTL3.7).

T2n MODE = 0 = dual compare mode
T2n MODE = 1 = dual capture mode

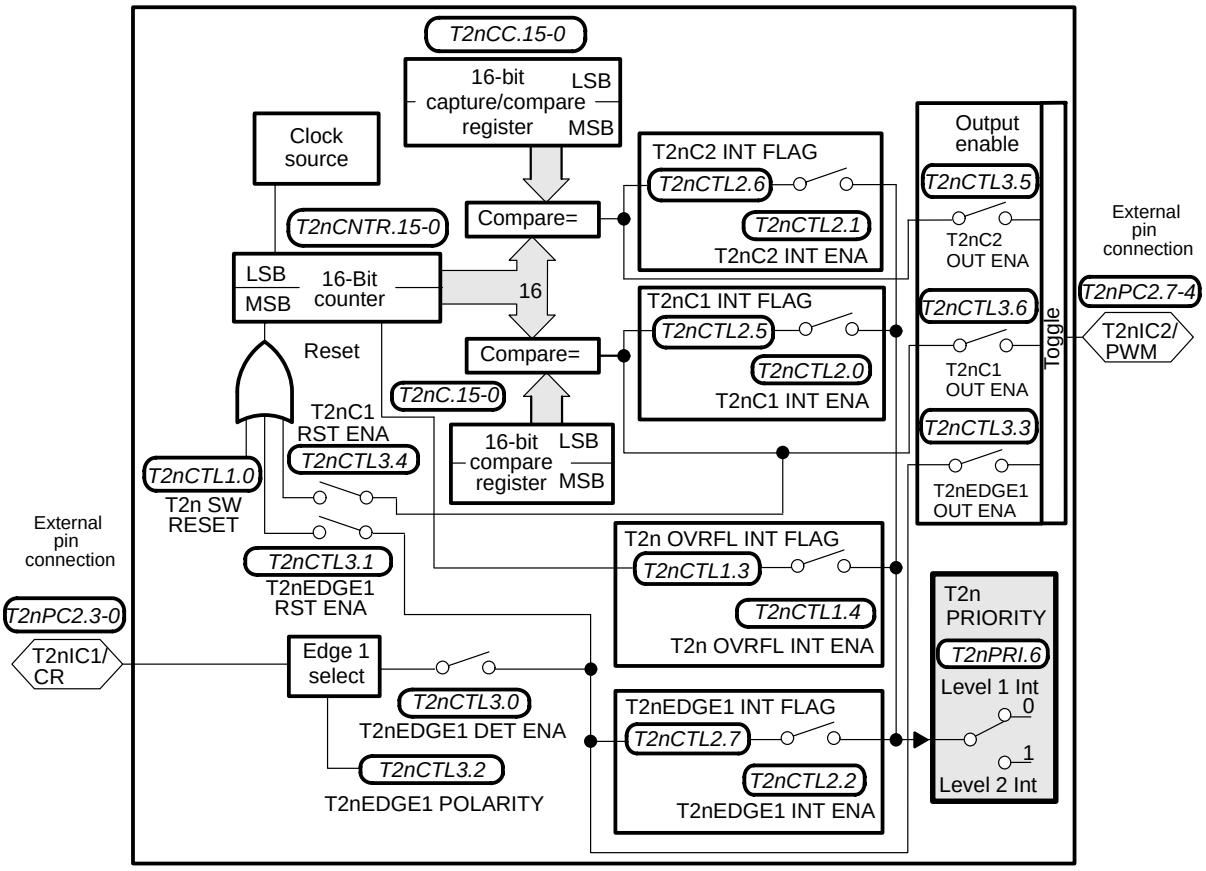
8.3.1 Dual Compare Mode

The dual compare mode provides the following:

- A 16-bit compare register (called compare 1)
- A 16-bit capture/compare register that acts as a compare register (called compare 2)
- A 16-bit externally resettable counter
- A timer output pin

These components allow T2n to act as an interval timer, a PWM output, simple output toggle, or many other timer functions. In the dual compare mode, the operation of the T2n module is identical to that of the T1 module, with the exception of the clock sources. The dual compare mode is shown in Figure 8–2.

Figure 8–2. Dual Compare Mode



Note: The annotations on the diagram identify the register and the bit (s) in the peripheral frame. For example, the actual address of T2nCTL2.0 is 106Bh (n=A) or 108Bh (n=B), bit 0, in the T2nCTL2 register.

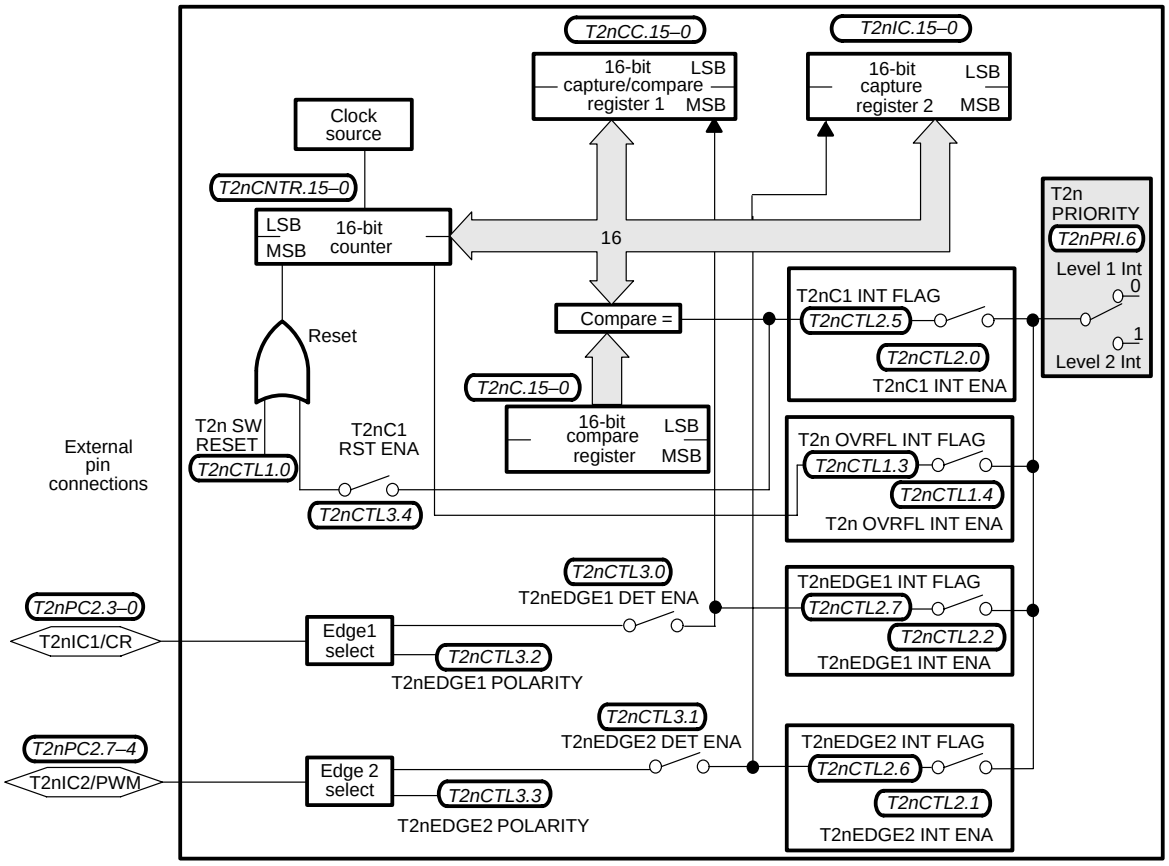
8.3.2 Dual Capture Mode

In the dual capture mode, T2n provides the following:

- ❑ A 16-bit compare register for use as a programmable interval timer
- ❑ A 16-bit capture register 2 for external input time and pulse width measurement
- ❑ A 16-bit capture/compare register 1 that acts as a capture register for external input timing and pulse width measurement

The dual capture mode is shown in Figure 8–3 on page 8-12.

Figure 8–3. Dual Capture Mode



Note: The annotations on the diagram identify the register and the bit (s) in the peripheral frame. For example, the actual address of T2nCTL2.0 is 106Bh (n=A) or 108Bh (n=B), bit 0, in the T2nCTL2 register.

Each input capture pin (T2nIC1/CR and T2nIC2/PWM) has an input edge detect function enabled by the associated DET ENA control bit, with the associated POLARITY bit selecting the active input transition.

On the occurrence of a valid input on the T2nIC1/CR or T2nIC2/PWM pin, the current counter value is loaded into the 16-bit capture/compare register or 16-bit input capture register, respectively. In addition, the respective input capture INT FLAG bit is set, and a timer interrupt is generated if the respective INT ENA bit is set.

8.4 Edge-Detection Circuitry

This edge detection circuitry senses an active pulse transition on the input pins and provides appropriate output transitions to the rest of the module.

8.4.1 Dual Compare Mode

In this mode, the edge detection circuitry is connected to the module's T2nIC1/CR pin. The program must set the T2nEDGE1 DET ENA bit (T2nCTL3.0) to re-enable the T2n module after each edge detection.

When the T2n module detects an active transition (while enabled), then the module performs the following:

- ❏ Clears the T2nEDGE1 DET ENA bit.
- ❏ Sets the T2nEDGE1 INT FLAG bit (T2nCTL2.7).
- ❏ Resets the counter if T2nEDGE1 RST ENA bit (T2nCTL3.1) is set.
- ❏ Toggles the output flip-flop if the T2nEDGE1 OUT ENA bit (T2nCTL3.3) is set.

In the dual compare mode, the T2nEDGE1 POLARITY bit (T2nCTL3.2) determines whether the active transition is low-to-high or high-to-low.

8.4.2 Dual Capture Mode

In this mode, the edge detection circuitry is connected to both the T2nIC1/CR pin and the T2nIC2/PWM pin.

When the edge 1 detect circuit detects an active edge transition on the T2nIC1/CR pin, the T2n module conducts the following:

- ❏ Loads the capture/compare register with the current counter value.
- ❏ Sets the T2nEDGE1 INT FLAG bit (T2nCTL2.7).

When the edge 2 detect circuit detects an active edge transition on the T2nIC2/PWM pin, the module performs the following:

- ❏ Loads the capture register with the current counter value.
- ❏ Sets the T2nEDGE2 INT FLAG bit (T2nCTL2.6).

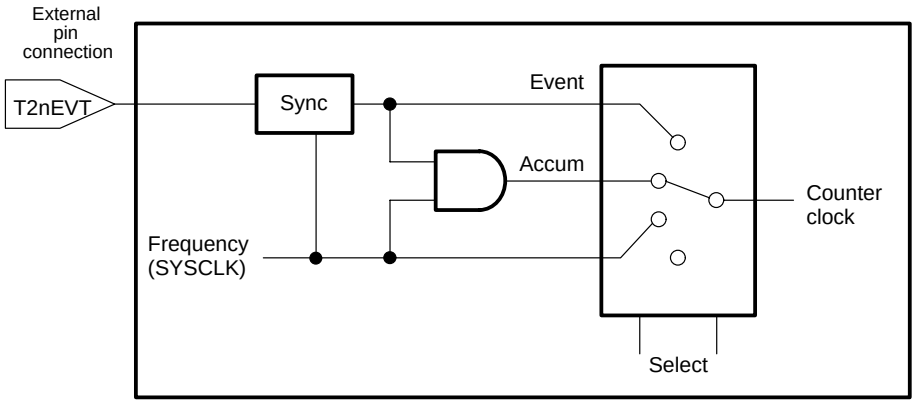
The T2nEDGE1 POLARITY bit (T2nCTL3.2) and the T2nEDGE2 POLARITY bit (T2nCTL3.3) determine the transition (rising or falling) to be detected.

8.5 Clock Sources

The T2n clock sources are shown in Figure 8–4 and can be any of the following:

- System clock
- No clock (the counter is stopped)
- External clock synchronized to the system clock (event counter)
- System clock while external input is high (pulse accumulation)

Figure 8–4. T2n Clock Sources



The T2n INPUT SELECT0 bit (T2nCTL1.1) and the T2n INPUT SELECT1 bit (T2nCTL1.2) select one of four clock sources (refer to subsection 8.8.1 on page 8-19).

The maximum counter duration with an internal clock is based on the internal system clock time (SYSCLK) as follows:

$$\text{Maximum Counter Duration} = 2^{16} \times (1/\text{SYSCLK})$$
$$\text{Counter Resolution} = 1/\text{SYSCLK}$$

where:

SYSCLK = CLKIN/4 for divide-by-4 clock
= CLKIN/1 for divide-by-1 clock

The external event frequency input to the module cannot exceed SYSCLK/2. All external event inputs are synchronized with the system clock.

When the timer is using the system clock input, the 16-bit timer generates an overflow rate of 13.1 ms with 200-ns resolution (5-MHz SYSCLK).

8.5.1 Event Counter Mode

When you use the event counter clock source, the 16-bit counter is programmable as a 16-bit event counter. An external low-to-high transition on the T2nEVT pin provides the clock for the internal timer. The T2nEVT external clock frequency cannot exceed the system clock frequency divided by 2.

8.5.2 Pulse Accumulator Mode

When you use the pulse accumulator clock source, the 16-bit counter is programmable as a 16-bit pulse accumulator. An external input on the T2nEVT pin is used to gate the internal system clock to the internal timers. While Tn2EVT input is logic one (high), the timer is clocked at the system clock rate and counts system clock pulses until the T2nEVT pin returns to logic zero.

8.6 Interrupts

Interrupts can be enabled to occur upon an output compare equal, counter overflow, and/or an external edge detect (input capture).

In dual compare mode, the following four separate events can generate an interrupt:

- A compare-equal for the dedicated compare register if the T2nC1 INT ENA bit (T2nCTL2.0) is set
- A compare-equal for the capture/compare register if the T2nC2 INT ENA bit (T2nCTL2.1) is set
- A counter overflow if the T2n OVERFL INT ENA bit (T2nCTL1.4) is set
- An external edge detect if the T2nEDGE1 DET ENA and T2nEDGE1 INT ENA bits are set (T2nCTL3.0 and T2nCTL2.2, respectively)

In dual capture mode, four separate events can generate an interrupt. These events are as follows:

- A compare-equal for the dedicated compare register if the T2nC1 INT ENA bit (T2nCTL2.0) is set
- A counter overflow if the T2n OVERFL INT ENA bit (T2nCTL1.4) is set
- An external edge 1 detect if the T2nEDGE1 DET ENA and T2nEDGE1 INT ENA bits are set (T2nCTL3.0 and T2nCTL2.2)
- An external edge 2 detect if the T2nEDGE2 DET ENA and T2nEDGE2 INT ENA bits are set (T2nCTL3.1 and T2nCTL2.1)

Note:

All set and enabled interrupt flags must be cleared before the processor exits the T2n interrupt routine. If the flags are not reset, the processor will enter the T2n interrupt routine again instead of continuing the mainstream program. If the flag bits are never cleared, the program will continually enter the interrupt service routine.

8.7 Low-Power Modes

The T2n module supports low-power (powerdown) modes that aid in reducing power consumption during periods of inactivity. These modes are the halt and the standby modes. In both the halt and standby modes, no clocks or external inputs are recognized.

If the PWRDWN/IDLE bit (SCCR2.6) is set, the low-power modes are entered when an IDLE instruction is executed by the CPU. During the low-power mode, the T2n module holds the pre-idle status of all storage elements. All external pins are held constant, regardless of the pin function: inputs remain inputs, output low levels remain low, and output high levels remain high. When the idle state is exited, the I/O timer module continues from where it entered the idle state.

8.8 T2n Control Registers

Six registers control the T2n module operating mode selection, interrupt enable, status flags, and output configuration. These registers are shown in Figure 8–5. The bits that are shown in *shaded boxes* are privilege mode bits; that is, they can be written to only in the privilege mode.

Note:

Special circuitry prevents 16-bit registers from changing in the middle of a 16-bit read or write operation. When you read a 16-bit register, read the least significant byte (LSbyte) first to lock in the value, and then read the most significant byte (MSbyte). When you write to a 16-bit register, write the MSbyte first and then write the LSbyte. The register value does not change between reading and writing the bytes when they are done in this order. While you access a 16-bit register, do not read or write from a second 16-bit register within this module; if you do, the value for the first register's MSbyte will not be correct. The 16-bit read/write operation actually occurs when you access the LSbyte. In summary, the read/write operation should be conducted in the following order:

Read: **LSbyte then MSbyte**
Write: **MSbyte then LSbyte**

Figure 8–5. Peripheral File Frames 6 (T2A) and 8 (T2B): T2n Control Registers

Designation	ADDR T2A/T2B	PF T2A/T2B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2nCNTR	1060h/1080h	P060/P080	Bit 15	T2n Counter MSbyte						Bit 8
T2nCNTR	1061h/1081h	P061/P081	Bit 7	T2n Counter LSbyte						Bit 0
T2nC	1062h/1082h	P062/P082	Bit 15	Compare Register MSbyte						Bit 8
T2nC	1063h/1083h	P063/P083	Bit 7	Compare Register LSbyte						Bit 0
T2nCC	1064h/1084h	P064/P084	Bit 15	Capture/Compare Register MSbyte						Bit 8
T2nCC	1065h/1085h	P065/P085	Bit 7	Capture/Compare Register LSbyte						Bit 0
T2nIC	1066h/1086h	P066/P086	Bit 15	Capture Register 2 MSbyte						Bit 8
T2nIC	1067h/1087h	P067/P087	Bit 7	Capture Register 2 LSbyte						Bit 0
	1068h/1088h	P068/P088	Bit 15	Reserved						Bit 8
	1069h/1089h	P069/P089	Bit 7	Reserved						Bit 0
T2nCTL1	106Ah/108Ah	P06A/P08A	—	—	—	T2n OVRFL INT ENA (RW–0)	T2n OVRFL INT FLAG (RC–0)	T2n INPUT SELECT1 (RW–0)	T2n INPUT SELECT0 (RW–0)	T2n SW RESET (S–0)
T2nCTL2	106Bh/108Bh	P06B/P08B	In Dual Compare Mode							
			T2nEDGE1 INT FLAG (RC–0)	T2nC2 INT FLAG (RC–0)	T2nC1 INT FLAG (RC–0)	—	—	T2nEDGE1 INT ENA (RW–0)	T2nC2 INT ENA (RW–0)	T2nC1 INT ENA (RW–0)
			In Dual Capture Mode							
			T2EDGE1 INT FLAG (RC–0)	T2EDGE2 INT FLAG (RC–0)	T2nC1 INT FLAG (RC–0)	—	—	T2nEDGE1 INT ENA (RW–0)	T2nEDGE2 INT ENA (RW–0)	T2nC1 INT ENA (RW–0)
T2nCTL3	106Ch/108Ch	P06C/P08C	In Dual Compare Mode							
			T2n MODE= 0 (RW–0)	T2nC1 OUT ENA (RW–0)	T2nC2 OUT ENA (RW–0)	T2nC1 RST ENA (RW–0)	T2nEDGE1 OUT ENA (RW–0)	T2nEDGE1 POLARITY (RW–0)	T2nEDGE1 RST ENA (RW–0)	T2nEDGE1 DET ENA (RW–0)
			In Dual Capture Mode							
			T2n MODE= 1 (RW–0)	—	—	T2nC1 RST ENA (RW–0)	T2nEDG2 POLARITY (RW–0)	T2nEDGE1 POLARITY (RW–0)	T2nEDGE2 DET ENA (RW–0)	T2nEDGE1 DET ENA (RW–0)
			In Dual Compare and Dual Capture Mode							
T2nPC1	106Dh/108Dh	P06D/P08D	—	—	—	—	T2nEVT DATA IN (R–0)	T2nEVT DATA OUT (RW–0)	T2nEVT FUNCTION (RW–0)	T2nEVT DATA DIR (RW–0)
T2nPC2	106Eh/108Eh	P06E/P08E	T2nIC2/ PWM DATA IN (R–0)	T2nIC2/PWM DATA OUT (RW–0)	T2nIC2/PWM FUNCTION (RW–0)	T2nIC2/PWM DATA DIR (RW–0)	T2nIC1/CR DATA IN (R–0)	T2nIC1/CR DATA OUT (RW–0)	T2nIC1/CR FUNCTION (RW–0)	T2nIC1/CR DATA DIR (RW–0)
T2nPRI	106Fh/108Fh	P06F/P08F	T2n STEST (RP–0)	T2n PRIORITY (RP–0)	—	—	—	—	—	—

8.8.1 T2n Control Register 1 (T2nCTL1)

The T2nCTL1 register controls the clock input selection, counter overflow interrupts, and counter software reset.

T2n Control Register 1 (T2nCTL1)
[Memory Address 106Ah (T2A) or 108Ah (T2B)]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06A or P08A	—	—	—	T2n OVRFL INT ENA	T2n OVRFL INT FLAG	T2n INPUT SELECT1	T2n INPUT SELECT0	T2n SW RESET
				RW-0	RC-0	RW-0	RW-0	S-0

R = Read, W = Write, S = Set only, C = Clear only, *-n* = Value of the bit after the register is reset

Bits 7–5 **Reserved.** Read data is indeterminate.

Bit 4 **T2n OVRFL INT ENA.** T2n Overflow Interrupt Enable

This bit controls the T2n overflow interrupting capability.

0 = Disables interrupt
1 = Enables interrupt from overflow

Bit 3 **T2n OVRFL INT FLAG.** T2n Overflow Interrupt Flag

This bit is the T2n counter overflow bit.

0 = Overflow interrupt is inactive.
1 = Overflow interrupt is pending.

Bits 2–1 **T2n INPUT SELECT1–0.** T2n Input Select

These two bits select one of four clock sources as an input to the counter. The four options are:

- System clock with no prescale
- System clock when external input is high (pulse accumulation)
- External source synchronized with system clock (event input)
- No clock

The combinations are shown below:

T2n INPUT SELECT2	T2n INPUT SELECT1	Counter Clock Source
0	0	System clock
0	1	Pulse accumulation
1	0	Event input
1	1	No clock input

Bit 0 **T2n SW RESET.** T2n Software Reset

When a 1 is written to this bit, the counter will reset to 0000h on the next system clock cycle; however, this bit is always read as a zero.

8.8.2 T2n Control Register 2 (T2nCTL2)

The T2nCTL2 register contains interrupt flags and controls the capability of the module to issue interrupts. Each of these registers is paired (P06B/P08B and P06C/P08C) with a T2n control register 3 (described in subsection 8.8.3 on page 8-23). The registers are used in the dual compare mode or in the dual capture mode depending on the mode set in the T2n control register 3, bit 7.

T2n Control Register 2 (T2nCTL2)
[Memory Address 106Bh (T2A) or 108Bh (T2B)]

Mode: Dual Compare

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06B or P08B	T2nEDGE1 INT FLAG	T2nC2 INT FLAG	T2nC1 INT FLAG	—	—	T2nEDGE1 INT ENA	T2nC2 INT ENA	T2nC1 INT ENA
	RC-0	RC-0	RC-0			RW-0	RW-0	RW-0

Mode: Dual Capture

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06B or P08B	T2nEDGE1 INT FLAG	T2nEDGE2 INT FLAG	T2nC1 INT FLAG	—	—	T2nEDGE1 INT ENA	T2nEDGE2 INT ENA	T2nC1 INT ENA
	RC-0	RC-0	RC-0			RW-0	RW-0	RW-0

R = Read, W = Write, C = Clear only, *-n* = Value of the bit after the register is reset

Bit 7 **T2nEDGE1 INT FLAG.** T2n External Edge 1 Interrupt Flag

This bit is set when the appropriate edge is detected on the T2nIC1/CR pin.

0 = Interrupt inactive
1 = Interrupt pending from edge 1 detect circuitry

Note:

Be careful using the AND, OR, XOR, CMPBIT, SBIT0, or SBIT1 instruction to modify this register. The read/modify/write nature of these instructions can inadvertently clear an interrupt flag that was set between the read and the write cycles. If the state of the interrupt enable bits is known, the MOV #iop8, Pd instruction can be used. If the state of the interrupt enable bits is not known, a sequence similar to the example shown below should be used.

```
; Clearing the T2AC1 INT FLAG
      MOV    P06B,A
      OR     #0E0h,A
      AND    #0DFh,A
      MOV    A,P06B
```

Bit 6 *Dual Compare Mode:*

T2nC2 INT FLAG. T2n Output Compare 2 Interrupt Flag

This bit is set when the capture/compare register first matches the counter value.

0 = Interrupt inactive
1 = Interrupt pending from compare 2

Dual Capture Mode:

T2nEDGE2 INT FLAG. T2n Edge 2 Interrupt Flag

This bit is set when the appropriate edge is detected on T2nIC2/PWM and indicates that the capture register was loaded.

0 = Interrupt inactive
1 = Interrupt pending from edge 2 detect

Bit 5 **T2nC1 INT FLAG.** T2n Output Compare 1 Interrupt Flag

This bit is set when the output compare register first matches the counter value.

0 = Interrupt inactive
1 = Interrupt pending from compare 1

Bits 4–3 **Reserved.** Read data is indeterminate.

Bit 2 **T2nEDGE1 INT ENA.** T2n External Edge 1 Interrupt Enable

This bit determines whether or not the active edge input to the T2nIC1/CR pin generates an interrupt.

0 = Disables interrupt
1 = Enables interrupt

Bit 1 *Dual Compare Mode:*

T2nC2 INT ENA. T2n Output Compare 2 Interrupt Enable

This bit controls the interrupting capability of the compare 2 register.

0 = Disables interrupt
1 = Enables interrupt from compare 2 register

Dual Capture Mode:

T2nEDGE2 INT ENA. T2n External Edge 2 Interrupt Enable

This bit determines whether or not the active edge input to the T2nIC2/PWM pin generates an interrupt.

0 = Disables interrupt
1 = Enables interrupt

Bit 0 **T2nC1 INT ENA.** T2n Compare 1 Interrupt Enable

This bit controls the interrupting capability of the compare 1 register.

0 = Disables interrupt
1 = Enables interrupt from compare 1 register

8.8.3 T2n Control Register 3 (T2nCTL3)

The T2nCTL3 register controls the T2n module mode of operation (dual compare or dual capture as set by bit 7), outputs, active transition polarity, and counter reset. Each of these registers is paired with a T2n control register 2 (P06B/P08B and P06C/P08C) described in subsection 8.8.2 on page 8-20.

T2n Control Register 3 (T2nCTL3)
[Memory Address 106Ch (T2A) or 108Ch (T2B)]

Mode: Dual Compare

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06C or P08C	T2n MODE=0	T2nC1 OUT ENA	T2nC2 OUT ENA	T2nC1 RST ENA	T2nEDGE1 OUT ENA	T2nEDGE1 POLARITY	T2nEDGE1 RST ENA	T2nEDGE1 DET ENA
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Mode: Dual Capture

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06C or P08C	T2n MODE=1	—	—	T2nC1 RST ENA	T2nEDGE2 POLARITY	T2nEDGE1 POLARITY	T2nEDGE2 DET ENA	T2nEDGE1 DET ENA
	RW-0			RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **T2n MODE.** T2n Mode Select

This bit selects the operating mode for the counter.

0 = Dual compare mode
1 = Dual capture mode

Bit 6 *Dual Compare Mode:*

T2nC1 OUT ENA. T2n Output Compare 1 Enable

This bit controls whether or not the compare equal pulse from the compare register toggles T2nIC2/PWM pin.

0 = Disables pulse from toggling output
1 = Enables pulse to toggle output

Dual Capture Mode:

Reserved. Read data is indeterminate.

Bit 5 *Dual Compare Mode:*

T2nC2 OUT ENA. T2n Output Compare 2 Enable

This bit controls whether or not the output compare equal pulse from the capture/compare register toggles the T2nIC2/PWM output pin.

0 = Disables pulse to toggle output

1 = Enables pulse to toggle output

Dual Capture Mode:

Reserved. Read data is indeterminate.

Bit 4 **T2nC1 RST ENA.** T2n Output Compare 1 Reset Enable

This bit controls whether or not the compare equal pulse from the compare register resets the counter on the next counter increment.

0 = Disables reset upon compare equal

1 = Enables reset upon compare equal

Bit 3 *Dual Compare Mode:*

T2nEDGE1 OUT ENA. T2n Edge 1 Detect Output Enable

This bit controls whether or not the pulse indicating an external edge detect toggles the module's output pin.

0 = Disables pulse to toggle output

1 = Enables pulse to toggle output

Dual Capture Mode:

T2nEDGE2 POLARITY. T2n Edge 2 Polarity Select

This bit controls which transition level on the T2nIC2/PWM pin is active.

0 = Triggers on high-to-low transition

1 = Triggers on low-to-high transition

Bit 2 **T2nEDGE1 POLARITY.** T2n Edge 1 Polarity Select

This bit controls which transition level on the T2nIC1/CR pin is active.

0 = Triggers on high-to-low transition

1 = Triggers on low-to-high transition

Bit 1 *Dual Compare Mode:*

T2nEDGE1 RST ENA. T2n Edge 1 Detect Reset Enable

This bit controls whether or not an external signal can reset the counter.

0 = Disables external reset of the counter

1 = Enables external reset of the counter

Dual Capture Mode:

T2nEDGE2 DET ENA. T2n External Edge 2 Detect Enable

This bit enables the edge detection circuit to sense the next active level transition on the T2nIC2/PWM pin. This bit remains unchanged after the selected transition is detected and during reset.

0 = Disables edge detect

1 = Enables edge detect

Bit 0 *Dual Compare Mode:*

T2nEDGE1 DET ENA. T2n Edge 1 Detect Enable

This bit enables the edge detection circuit to sense the next active level transition on the T2nIC1/CR pin. This bit is cleared after the selected transition is detected and during reset.

0 = Disables edge 1 detect

1 = Enables edge 1 detect

Dual Capture Mode:

T2nEDGE1 DET ENA. T2n Edge 1 Detect Enable

This bit enables the edge detection circuit to sense the next active level transition on the T2nIC1/CR pin. This bit remains unchanged after the selected transition is detected and during reset.

0 = Disables input capture.

1 = Enables input capture.

8.8.4 T2n Port Control Registers (T2nPC1 and T2nPC2)

The port control registers (PCRs) control the functions of the I/O pins. Each module pin is controlled by a nibble in one of the PCRs.

8.8.4.1 T2n Port Control Register 1 (T2nPC1)

The T2nPC1 register assigns the I/O function of the T2nEVT pin as either a general-purpose digital I/O or external event input of the module.

T2n Port Control Register 1 (T2nPC1)
[Memory Address 106Dh (T2A) or 108Dh (T2B)]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06D or P08D	—	—	—	—	T2nEVT DATA IN	T2nEVT DATA OUT	T2nEVT FUNCTION	T2nEVT DATA DIR
					R-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bits 7–4 **Reserved.** Read data is indeterminate.

Bit 3 **T2nEVT DATA IN.** T2n Event Pin Data In

This bit contains the data to be input from the T2nEVT pin. A write to this bit has no effect.

Bit 2 **T2nEVT DATA OUT.** T2n Event Pin Data Out

This bit contains the data to be output on the T2nEVT pin if the following conditions are met:

a. Bit T2nEVT DATA DIR = 1
b. Bit T2nEVT FUNCTION = 0

Bit 1 **T2nEVT FUNCTION.** T2n Event Pin Function Select

This bit selects the function of the T2EVT pin.

0 = T2nEVT is a general-purpose digital I/O pin.
1 = T2nEVT is the event input pin.

Bit 0 **T2nEVT DATA DIR.** T2n Event Pin Data Direction

This bit determines the data direction on the T2nEVT pin if the T2nEVT FUNCTION bit = 0.

0 = T2nEVT is configured as input.
1 = T2nEVT is configured as output.

8.8.4.2 T2n Port Control Register 2 (T2nPC2)

The T2nPC2 register assigns the I/O functions of the T2nIC1/CR and T2nIC2/PWM pins as either general-purpose digital I/O pins or the input capture/counter reset and PWM output pins, respectively.

T2n Port Control Register 2 (T2nPC2)
[Memory Address 106Eh (T2A) or 108Eh (T2B)]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06E or P08E	T2nIC2/ PWM DATA IN	T2nIC2/ PWM DATA OUT	T2nIC2/ PWM FUNCTION	T2nIC2/ PWM DATA DIR	T2nIC1/CR DATA IN	T2nIC1/CR DATA OUT	T2nIC1/CR FUNCTION	T2nIC1/CR DATA DIR
	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **T2nIC2/PWM DATA IN.** T2n IC2/PWM Data In

This bit contains the data input on the T2nIC2/PWM pin. A write to this bit has no effect.

Bit 6 **T2nIC2/PWM DATA OUT.** T2n IC2/PWM Data Out

This bit contains the data output on the T2nIC2/PWM pin if the following conditions are true:

a. Bit T2nIC2/PWM DATA DIR = 1
b. Bit T2nIC2/PWM FUNCTION = 0

Bit 5 **T2nIC2/PWM FUNCTION.** T2n IC2/PWM Function Select

This bit determines the function of the T2nIC2/PWM pin.

0 = T2nIC2/PWM is a general-purpose digital I/O pin.
1 = T2nIC2/PWM is the input capture/PWM output pin.

Bit 4 **T2nIC2/PWM DATA DIR.** T2n IC2/PWM Data Direction

This bit determines the direction of data on the T2nIC2/PWM pin if the T2nIC2/PWM FUNCTION bit = 0.

0 = T2nIC1/PWM is an input.
1 = T2nIC2/PWM is an output.

Bit 3 **T2nIC1/CR DATA IN.** T2n IC1/CR Data In

This bit contains the data input on the T2nIC1/CR pin. A write to this bit has no effect.

Bit 2 **T2nIC1/CR DATA OUT.** T2n IC1/CR Data Out

This bit contains the data output on the T2nIC1/CR pin if the following conditions are true:

a. Bit T2nIC1/CR DATA DIR = 1
b. Bit T2nIC1/CR FUNCTION = 0

Bit 1 **T2nIC1/CR FUNCTION.** T2n IC1/CR Function Select

This bit determines the function of the T2nIC1/CR pin.

0 = T2nIC1/CR is a general-purpose digital I/O pin.
1 = T2nIC1/CR is the input capture/counter reset pin.

Bit 0 **T2nIC1/CR DATA DIR.** T2n IC1/CR Data Direction

This bit determines the direction of data on the T2nIC1/CR pin if the T2nIC1/CR FUNCTION bit = 0.

0 = T2nIC1/CR is an input.
1 = T2nIC1/CR is an output.

8.8.5 T2n Interrupt Priority Control Register (T2nPRI)

The T2nPRI register assigns the priority level of interrupts generated by the T2n module. You can write to this register only in the privilege mode. During normal operation, this is a read-only register.

T2n Priority Control Register (T2nPRI)
[Memory Address 106Fh (T2A) or 108Fh (T2B)]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P06F or P08F	T2n STEST	T2n PRIORITY	—	—	—	—	—	—
	RP-0	RP-0						

R = Read, P = Privilege write only, *-n* = Value of the bit after the register is reset

Bit 7 **T2n STEST.** T2n STEST

This bit must be cleared to ensure proper operation.

Bit 6 **T2n PRIORITY.** T2n Interrupt Priority Select

This bit determines the level of T2n interrupts.

0 = Interrupts are level 1 (high priority) requests.
1 = Interrupts are level 2 (low priority) requests.

Bits 5-0 **Reserved.** Read data is indeterminate.

Chapter 9

Serial Communications Interface 1 (SCI1) Module

The SCI is described in Chapter 9 (*SCI1*: 3 I/O pins, asynchronous *and* isosynchronous modes) and Chapter 10 (*SCI2*: 2 I/O pins, asynchronous mode). This chapter describes SCI1 and covers the following topics:

Topic **Page**

9.1 SCI1 Overview

The serial communications interface 1 (SCI1) module is a programmable I/O port that facilitates digital communications between the TMS370 device and other asynchronous peripherals and uses the standard NRZ (nonreturn to zero) format. The SCI1 transmits and receives serial data, one bit at a time, at a programmable bit rate. Both the SCI1 receiver and transmitter are double-buffered and have their own separate enable and interrupt bits. They can be operated independently or simultaneously in the full duplex mode. The SCI1 module is available in the following families: TMS370Cx0x, TMS370Cx2x, TMS370Cx4x, TMS370Cx5x, TMS370Cx6x, and TMS370CxAx.

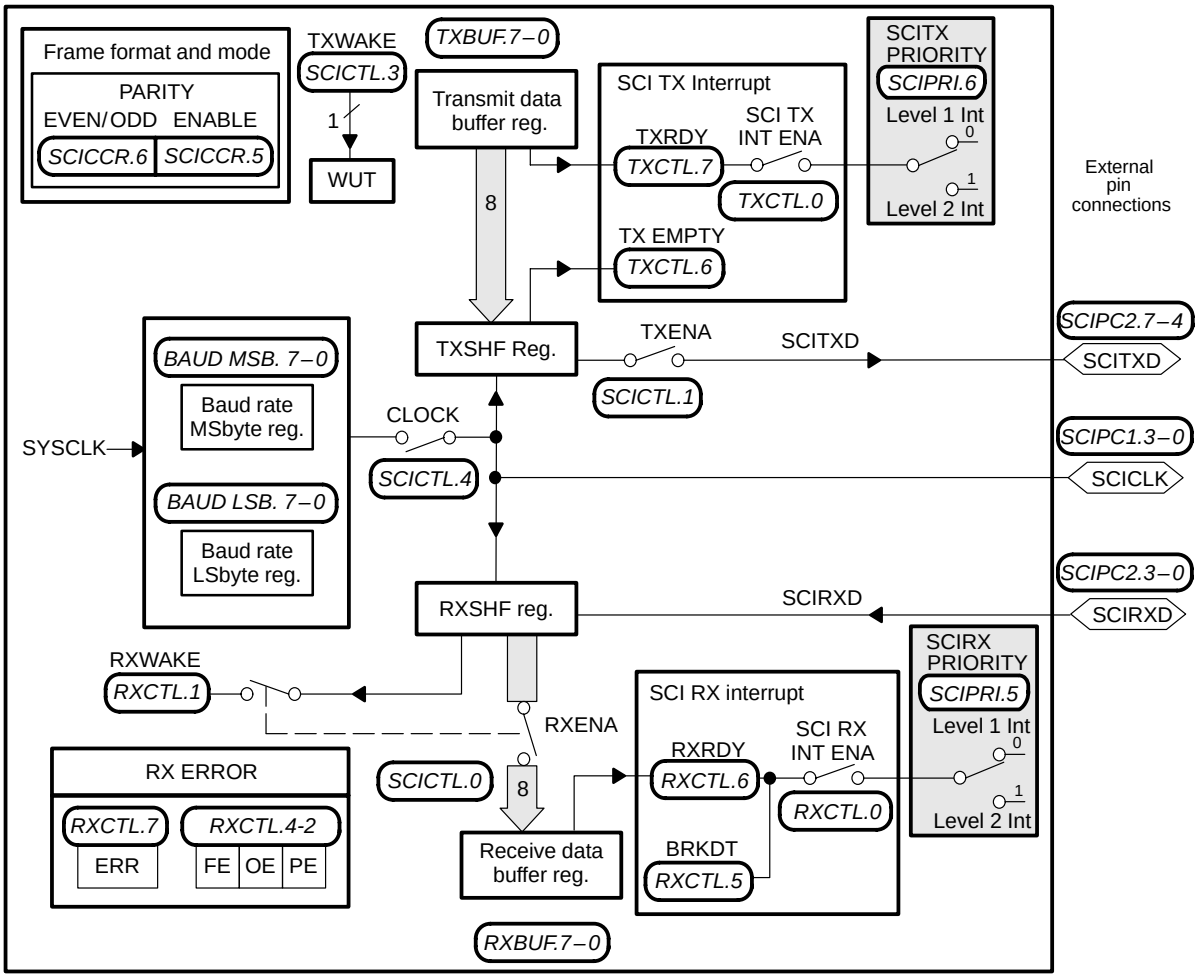
9.1.1 Physical Description

The three-pin SCI1 module, shown in Figure 9–1, key features:

- Three I/O pins:
  - SCIRXD (SCI1 receive data input)
  - SCITXD (SCI1 transmit data output)
  - SCICLK (SCI1 bidirectional serial clock)
- Two communications formats:
  - Asynchronous
  - Isosynchronous
- Programmable bit rates to over 65,000 different speeds through a 16-bit baud select register
  - Asynchronous:
    - Range at 5 MHz SYSCLK—3 bits/s to 156 Kbits/s
    - Number of bit rates—64K
  - Isosynchronous:
    - Range at 5 MHz SYSCLK—39 bits/s to 2.5 Mbits/s
    - Number of bit rates—64K
- Programmable data word length from 1 to 8 bits
- Programmable stop bits of either 1 or 2 bits in length
- Error detection flags that ensure data integrity:
  - Parity error
  - Overrun error
  - Framing error
  - Break detect

- Two wake-up multiprocessor modes that can be used with either communications format:
  - Idle line wake-up
  - Address bit wake-up
- Full duplex operation
- Separate transmitter and receiver interrupts for polled or interrupt-driven operation
- Double-buffered receive and transmit functions
- Separate enable bits for the transmitter and receiver
- NRZ (nonreturn to zero) format

Figure 9–1. SCI1 Block Diagram - Three Pin Configuration



Note: SCI1 registers are described in detail in Section 9.8.1 beginning on page 9-22.

9.1.2 Architecture

The major elements of the full-duplex SCI1 are shown in Figure 9–1 and include:

- A transmitter (SCITX)
  - TXBUF — the transmitter buffer register that contains data, written by the CPU, to be transmitted
  - TXSHF — the transmitter shift register that is loaded from TXBUF and shifts data onto the SCITXD pin, one bit at a time
- A receiver (SCIRX)
  - RXSHF — the receiver shift register that shifts data in from the SCIRXD pin, one bit at a time
  - RXBUF — the receiver buffer register that contains data that is to be read by the CPU and that is received from the remote processor and loaded from RXSHF
- A programmable baud generator
- Memory-mapped control and status registers

The SCI1 receiver and transmitter can operate independently and simultaneously. A third port line, SCICLK, is available for an optional synchronizing clock line in the isosynchronous mode.

9.1.3 Communications Modes and Multiprocessing Modes

The SCI1 offers the following universal asynchronous receiver/transmitter (UART) communications modes for interfacing with many popular peripherals:

- Asynchronous mode (discussed in subsection 9.4.1 on page 9-12) requires two lines to interface with many standard devices such as terminals and printers that use RS-232-C formats.
- Isosynchronous mode (discussed in subsection 9.4.2 on page 9-13) permits high transmission rates and requires a synchronizing clock signal between the receiver and transmitter.

Data transmission characteristics include:

- 1 start bit
- 1 to 8 data bits (SCICCR.0-2)
- An even/odd parity bit or no parity bit (SCICCR.6)
- 1 or 2 stop bits (SCICCR.7)

The SCI1 also has two multiprocessor modes: the idle line multiprocessor mode (see subsection 9.3.1 on page 9-9) and the address bit multiprocessor mode (see subsection 9.3.2 on page 9-10). These modes allow efficient data transfer between multiple processors and can be used with either the isosynchronous or standard asynchronous formats.

9.1.4 Control Registers

The SCI1 control registers are located at addresses 1050h to 105Fh and occupy peripheral file frame 5. The function of each location is shown in Table 9–1.

Table 9–1. SCI1 Memory Map

Peripheral File Location	Symbol	Name	Description	See Page
P050	SCICCR	SCI Communication Control Register	Defines the character format, protocol, and communications mode used by the SCI1.	9-22
P051	SCICTL	SCI Control Register	Controls the RX/TX enable, TXWAKE and SLEEP functions, internal clock enable, and the SCI1 software reset.	9-24
P052	BAUD MSB	Baud Select MSbyte Register	Stores the data required to generate the bit rate.	9-27
P053	BAUD LSB	Baud Select LSbyte Register		
P054	TXCTL	SCI Transmitter Interrupt Control and Status Register	Contains the transmitter interrupt enable, the transmitter ready flag, and the transmitter empty flag.	9-28
P055	RXCTL	SCI Receiver Interrupt Control and Status Register	Contains one interrupt enable bit and seven receiver status flags.	9-29
P056		Reserved		
P057	RXBUF	SCI Receiver Data Buffer	Contains the current data from the receiver shift register.	9-31
P058		Reserved		
P059	TXBUF	SCI Transmit Data Buffer	Stores data bits to be transmitted by the SCITX.	9-31
P05A–P05C		Reserved		
P05D	SCIPC1	SCI Port Control Register 1	Controls the SCICLK pin functions.	9-32
P05E	SCIPC2	SCI Port Control Register 2	Controls the SCIRXD and SCITXD pin functions.	9-33
P05F	SCIPRI	SCI Interrupt Priority Control Register	Contains the receiver and transmitter interrupt priority select bits.	9-35

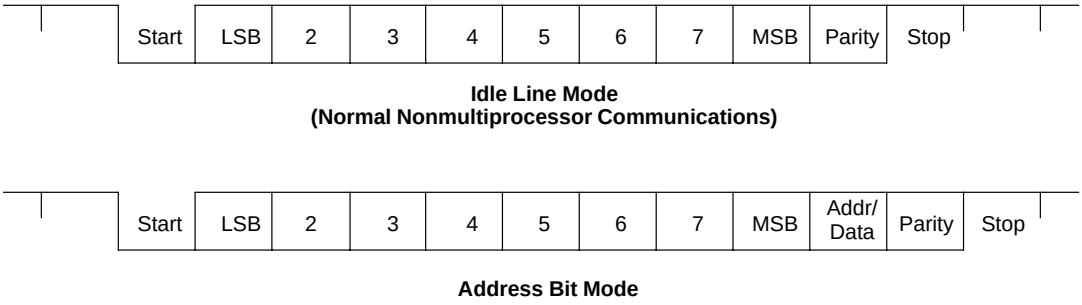
9.2 SCI1 Programmable Data Format

SCI1 data, both receive and transmit, is in NRZ (nonreturn to zero) format. The NRZ data format is illustrated in Figure 9–2 and consists of:

- 1 start bit
- 1 to 8 data bits
- An even/odd parity bit (optional)
- 1 or 2 stop bits
- An optional extra bit to distinguish addresses from data (address bit mode only).

The basic unit of data is called a character and is 1 to 8 bits in length. Each character of data is formatted with a start bit, 1 or 2 stop bits, and optional parity and address bits. A character of data with its formatting information is called a frame and is shown in Figure 9–2.

Figure 9–2. SCI1 Data Formats



To program the data format, use the SCICCR register (described in subsection 9.8.1 on page 9-22). The bits that you use to program the data format are shown in Table 9–2.

Table 9–2. Programming the Data Format Using SCICCR

Bit Name	Designation	Function
SCI CHAR0–2	SCICCR.0–2	Character length selection. Selects the character (data) length (1 to 8 bits). Refer to the bit listings on page 9-22 for additional information.
PARITY ENABLE	SCICCR.5	Enables the parity function if set to 1 or disables the parity function if cleared to 0.
EVEN/ODD PARITY	SCICCR.6	If parity is enabled, selects odd parity if cleared to 0 or even parity if set to 1.
STOP BITS	SCICCR.7	Determines the number of stop bits transmitted—one stop bit if cleared to 0 or two stop bits if set to 1.

9.3 Multiprocessor Communications

The multiprocessor communication format allows one processor to efficiently send blocks of data to other processors on the same serial link. You can have only one talker on a serial line at a time.

The first byte of a block of information that the talker sends contains an address byte that is read by all listeners. Only listeners with the correct address can be interrupted by the data bytes that follow the address byte. The listeners with an incorrect address remain uninterrupted until the next address byte.

All processors on the serial link set their SLEEP bit (SCICTL.2) to 1 so that they are interrupted only when the address byte is detected. When a processor reads a block address that corresponds to the CPU's device address as set by software, your program must clear the SLEEP bit to enable the SCI1 to generate an interrupt on receipt of each data byte.

Although the receiver still operates when the SLEEP bit is 1, it does not set RXRDY, RXINT, or the error status bits to 1 unless the address byte is detected and the address bit in the received frame is a 1. The SCI1 does not alter the SLEEP bit; your software must alter the SLEEP bit.

A processor recognizes an address byte according to the multiprocessor mode:

- The **idle line mode** leaves a quiet space before the address byte. This mode does not have an extra address/data bit and is more efficient than the address bit mode for handling blocks that contain more than 10 bytes of data.
- The **address bit mode** adds an extra bit into every byte to distinguish addresses from data. This mode is more efficient in handling many small blocks of data because, unlike the idle mode, it does not have to wait between blocks of data. However, at high transmit speeds, the program is not fast enough to avoid a 10-bit idle in the transmission stream.

You can select the multiprocessor mode via the ADDRESS/IDLE WUP bit (SCICCR.3). Both modes use the TXWAKE flag bit (SCICTL.3), RXWAKE flag bit (RXCTL.1), and the SLEEP flag bit (SCICTL.2) to control the SCITX and SCIRX features of these modes.

In both multiprocessor modes, the sequence is:

1) The SCI1 port wakes up (requests an interrupt) at the start of a block and reads the first frame that contains the destination address.
2) A software routine is entered through the interrupt and checks the RXWAKE flag bit. If the RXWAKE bit is a 1, the incoming byte is an address

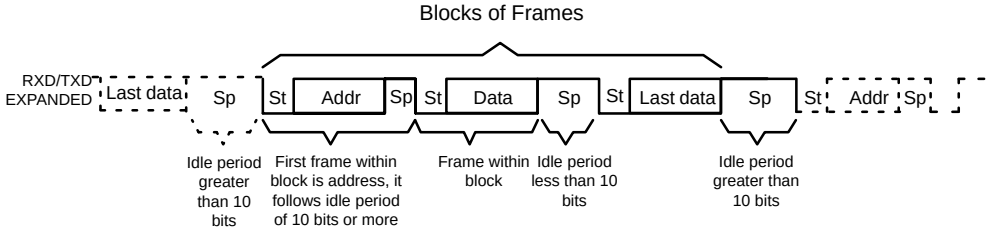
(otherwise the byte is data) and this address byte is checked against its device address byte stored in memory.

3) If the check shows that the block is addressed to the microcontroller, the CPU clears the SLEEP bit and reads the rest of the block; if not, the software routine exits with the SLEEP bit still set and does not receive SCI interrupts until the next block start.

9.3.1 Idle Line Multiprocessor Mode

In the idle line multiprocessor mode (ADDRESS/IDLE WUP bit = 0), blocks are separated by having a longer idle time between the blocks than between frames in the blocks. An idle time of 10 or more bits after a frame indicates the start of a new block. The idle line multiprocessor communication format is shown in Figure 9–3. (ADDRESS/IDLE WUP bit is SCICCR.3.)

Figure 9–3. Idle Line Multiprocessor Communication Format



Note: In the figure, “St” = start and “Sp” = stop

There are two ways to send a block start signal.

- The first method is to deliberately leave an idle time of 10 bits or more by delaying the time between the transmission of the last frame of data in the previous block and the transmission of the address frame of the new block.
- In the second method, the SCI1 port uses the TXWAKE bit (SCICTL.3) to send an idle time of exactly 11 bits. Thus, the serial communications line is not idle any longer than necessary.

Associated with the TXWAKE bit is the wake-up temporary or WUT flag bit. WUT is an internal flag, double buffered with TXWAKE. When TXSHF is loaded from TXBUF, WUT is loaded from TXWAKE, and the TXWAKE bit is cleared to 0. This arrangement is shown in Figure 9–4.

Figure 9–4. Double-Buffered WUT and TXSHF

To send out a block start signal of exactly one frame time:

1) Write a 1 to the TXWAKE bit.

2) Write a data word (don't care) to TXBUF. (The first data word written is suppressed while the block start signal is sent out, and ignored after that.)

   When TXSHF is free again, TXBUF's contents are shifted to TXSHF, the TXWAKE value is shifted to WUT, and then the TXWAKE bit is cleared. If TXWAKE bit was set to a 1, the start, data, and parity bits are replaced by an idle period of 11 bits transmitted following the last stop bit of the previous frame.

3) Write an address value to the TXBUF.

The receiver operates regardless of the SLEEP bit. The receiver does not set RXRDY, RXINT, or the error status bits until an address frame is detected.

9.3.2 Address Bit Multiprocessor Mode

In the address bit mode (ADDRESS/IDLE WUP bit = 1), frames have an extra bit, called an address bit, that immediately follows the last data bit. The address bit is set to 1 in the first frame of the block and to 0 in all other frames. The idle period timing is irrelevant (see Figure 9–5). (ADDRESS/IDLE WUP bit is SCICCR.3.)

The TXWAKE bit value is placed in the address bit. In SCITX, when the TXBUF and TXWAKE are loaded into TXSHF and WUT, TXWAKE is reset to 0, and WUT is the value of the address bit of the current frame. Thus, to send an address:

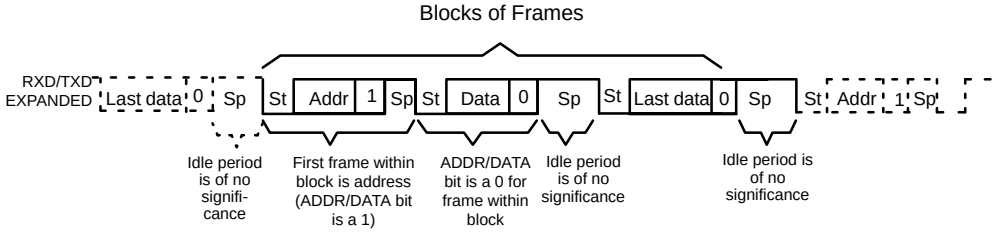
1) Set the TXWAKE bit to a 1 and write the appropriate address value to the TXBUF.

2) When this address value is transferred to TXSHF and shifted out, its address bit is sent as a 1, which flags the other processors on the serial link to read the address.

3) Since TXSHF and WUT are both double-buffered, TXBUF and TXWAKE can be written to immediately after TXSHF and WUT are loaded.

4) To transmit nonaddress frames in the block, leave the TXWAKE bit at 0.

Figure 9–5. Address Bit Multiprocessor Communication Format



Note: In the figure, “St” = start and “Sp” = stop

9.4 Communications Modes

The SCIRX/SCITX (receiver/transmitter) has two operating modes: asynchronous and isosynchronous. The ASYNC/ISOSYNC bit (SCICCR.4) determines the mode of operation. Either of these two modes can be used with either of the two forms of multiprocessor protocol: idle line and address bit.

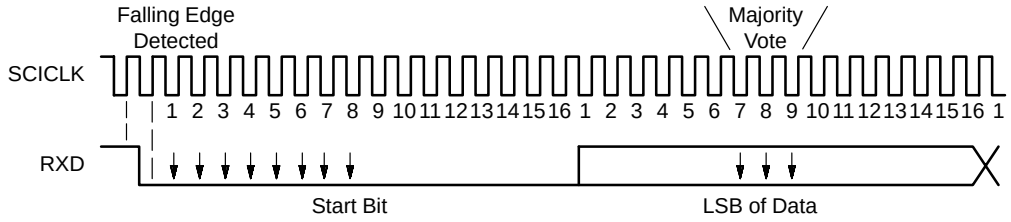
9.4.1 Asynchronous Communications Mode

The SCI1 asynchronous communication mode uses either single-line (one-way) or two-line (two-way) communications. In this mode, the frame consists of a start bit, one to eight data bits, an optional even/odd parity bit, and one or two stop bits. There are 16 SCICLK periods per data bit.

The receiver begins operation on receipt of a valid start bit. A valid start bit consists of eight consecutive zero bits. If any bit is not zero, then the processor starts over and begins looking for another start bit.

For the bits following the start bit, the processor determines the bit value by making three samples in the middle of the bits. These samples occur on the seventh, eighth, and ninth SCICLK period and are read on a majority (two out of three) basis. Figure 9–6 illustrates the asynchronous communication format, with a start bit showing how edges are found and where a majority vote is taken.

Figure 9–6. Asynchronous Communication Format



Since the receiver synchronizes itself to frames, the external transmitting and receiving devices do not have to use a synchronized serial clock; the clock can be generated locally. If the CLOCK bit (SCICTL.4) and SCICLK FUNCTION bit (SCIPC1.1) are set, then the serial clock is output continuously on the SCICLK pin.

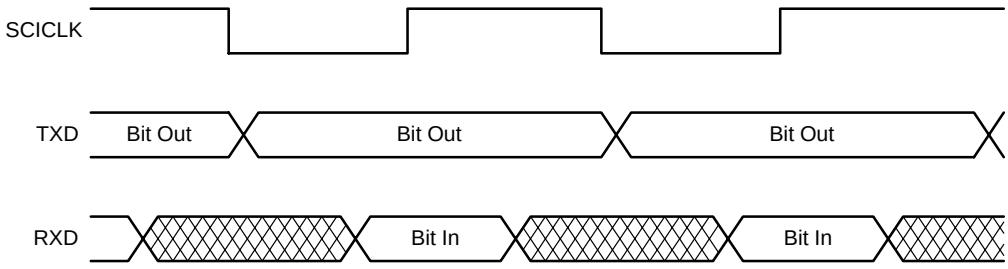
9.4.2 Isosynchronous Communications Mode

The SCI1 isosynchronous communication mode uses either two-line (one-way) or three-line (two-way) communications. The extra line (serial clock) in each case is required for data synchronization. In the isosynchronous mode, each bit of data requires only one serial clock pulse for transmission or reception. Thus, the data bit period equals the SCICLK period, and data bits are read on a single sample basis.

Since the receiver does not synchronize itself to data bits, the transmitter and receiver must be supplied with a common serial clock. If the internal serial clock is used, it must be output continuously on the SCICLK pin. The arrival of a valid start bit, which consists of a low on the RXD line at the time of a rising SCICLK edge, initiates receiver operation.

Figure 9–7 illustrates the isosynchronous communication format. A complete frame consists of a start bit, one to eight data bits, an optional even/odd parity bit, and one or two stop bits.

Figure 9–7. Isosynchronous Communication Format

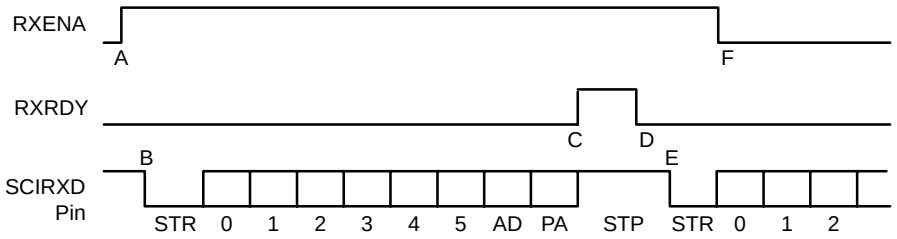


9.4.3 Receiver Signals in Communications Modes

Figure 9–8 illustrates receiver signal timing that assumes these conditions:

- Address bit wake-up mode (address bit would not appear in idle line mode)
- 6 bits per character

Figure 9–8. SCI1 RX Signals in Communications Modes



A) RX ENA goes high to enable the receiver.

B) Data arrives on the SCIRXD pin; start bit detected.

C) RXRDY goes high to signal that a new character has been received; data is shifted to RXBUF; an interrupt is requested.

D) The program reads the RXBUF register; RXRDY is automatically cleared.

E) The next byte of data arrives on the SCIRXD pin; start bit detected, then cleared.

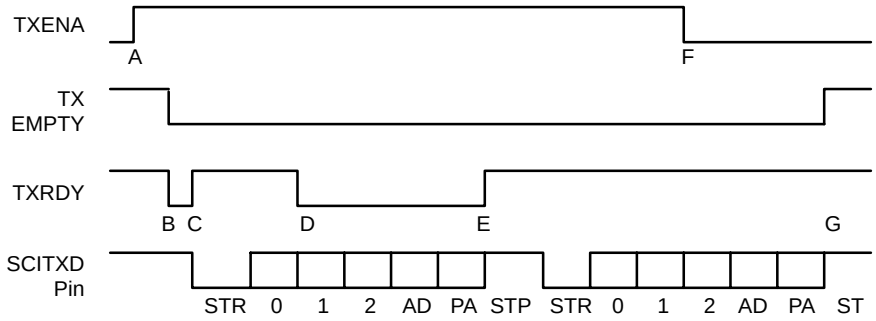
F) RX ENA goes low to disable the receiver; data continues to be assembled in the RXSHF register but is not transferred to the RXBUF register.

9.4.4 Transmitter Signals in Communications Modes

Figure 9–9 illustrates transmitter signal timing that assumes these conditions:

- Address bit wake-up mode (address bit would not appear in idle line mode)
- 3 bits per character

Figure 9–9. SCI1 TX Signals in Communications Modes



A) TX ENA goes high to enable the transmitter to send data.

B) Write to TXBUF; TXBUF is no longer empty.

C) SCI transfers data to shift register; TXBUF is ready for new character and requests an interrupt.

D) Program writes new character to TXBUF after TXRDY goes high (item C).

E) Finished transmitting first character; transfer new character to shift register.

F) TX ENA goes low to disable transmitter; SCI finishes transmitting current character.

G) Finished transmitting character; TXBUF is empty and ready for new character.

9.5 Port Interrupts

The SCI1 provides independent interrupt requests and vectors for the receiver and transmitter.

- If the SCI RX INT ENA bit (RXCTL.0) is set, the receiver interrupt is asserted when one of the following events occurs:
  - The SCI1 receives a complete frame and transfers the data in the RXSHF register to the RXBUF register. This action sets the RXRDY flag (RXCTL.6) and initiates an interrupt.
  - A break detect condition occurs (the SCIRXD is low for 10 bit periods following a stop bit). This action sets the BRKDT flag bit (RXCTL.5) and initiates an interrupt.
- If the SCI TX INT ENA bit (TXCTL.0) is set, the transmitter interrupt is asserted whenever the data in the TXBUF register is transferred to the TXSHF register, indicating that the CPU can write to the TXBUF; this action sets the TXRDY flag bit (TXCTL.7) and initiates an interrupt.

SCI1 interrupts can be programmed onto different priority levels by the SCI RX PRIORITY (SCIPRI.5) and SCI TX PRIORITY (SCIPRI.6) control bits. When both RX and TX interrupt requests are made on the same level, the receiver always has higher priority than the transmitter; this reduces the possibility of receiver overrun.

9.6 Clock Sources

The SCI1 port can be driven by an internal or external baud generator. The CLOCK bit (SCICTL.4) configures the SCI1 clock source as either an input or an output:

- If an external clock source is selected (CLOCK = 0) and the SCICLK FUNCTION bit (SCIPC1.1) is set, the SCICLK pin functions as the high-impedance serial clock input pin.
- If an internal clock source is selected (CLOCK = 1), the SCICLK pin can be used as a general-purpose I/O pin or as the serial clock output pin. If the serial clock output is selected (SCICLK FUNCTION = 0), a 50-percent duty cycle clock signal is output on the SCICLK pin that makes it a serial clock output pin.

The SCI1 receives data on rising clock edges and transmits data on falling clock edges.

The internally generated serial clock is determined by the TMS370 SYSCLK frequency and the baud select registers. The SCI1 uses the 16-bit value of the baud select registers to select one of 64K different serial clock rates for the communication modes in the following manner:

- Asynchronous Baud = SYSCLK / [(BAUD REG + 1) $\times$ 32]

  BAUD REG = [SYSCLK / (Asynchronous Baud $\times$ 32)] – 1
- Isosynchronous Baud = SYSCLK / [(BAUD REG + 1) $\times$ 2]

  BAUD REG = [SYSCLK / (Isosynchronous Baud $\times$ 2)] – 1
- SCICLK frequency = SYSCLK / [(BAUD REG + 1) $\times$ 2]

  BAUD REG = [SYSCLK / (SCICLK frequency $\times$ 2)] – 1

where

BAUD REG = The 16-bit value in the baud select registers.

Refer to Table 9–3. The baud select registers are further defined in subsection 9.8.3 on page 9-27.

Table 9–3. Asynchronous Baud Register Values for Common SCI1 Bit Rates

Baud	SYSCLK Frequency (MHz)							
	0.6144‡		1.8432‡		4.9152		5.0	
	Baud Reg†	% Error	Baud Reg†	% Error	Baud Reg†	% Error	Baud Reg†	% Error
75	255	0.00	767	0.00	2047	0.00	2082	0.02
300	63	0.00	191	0.00	511	0.00	520	-0.03
600	31	0.00	95	0.00	255	0.00	259	0.16
1200	15	0.00	47	0.00	127	0.00	129	0.16
2400	7	0.00	23	0.00	63	0.00	64	0.16
4800	3	0.00	11	0.00	31	0.00	32	-1.38
9600	1	0.00	5	0.00	15	0.00	15	1.73
19200	0	0.00	2	0.00	7	0.00	7	1.73
38400	-	-	-	-	3	-	3	1.73
156000	-	-	-	-	-	-	0	0.16

† Baud Reg = 16-bit baud register value
‡ Divide-by-1 clock can only operate from a minimum of 2 MHz SYSCLK to a maximum of 5 MHz SYSCLK.

Note:

When the device is using an externally generated SCICLK in isosynchronous mode, the maximum speed at which the SCICLK can run is limited to SYSCLK/10. This is necessary so that the internal clocks of the SCI1 have time to synchronize with the external clock. For this reason, it is recommended to use the TMS370 to drive the master serial clock in a system where maximum throughput is a major concern.

You can determine the current logic level on the SCICLK pin by reading the SCICLK DATA IN bit (SCIPC1.3).

9.7 Initialization Examples

This section contains two examples that initialize the serial port. In each example, the data is moved to and from the buffers in the interrupt routines.

1) The first example shows a typical RS-232 application that connects to a terminal.

2) The second example illustrates the address bit mode in a multiprocessor application.

In both examples, assume that the register mnemonics have been equated (EQU) with the corresponding peripheral-file location. For more examples using the TMS370 SCI, consult *Using the TMS370 SPI and SCI Modules Application Report*.

9.7.1 RS-232-C Example

This example initializes the transmitter and receiver to accept data at 9600 baud with a format of 8 data bits, 1 stop bit, and even parity.

```
B9600     .EQU    15                ;Value for counter for 9600 baud
HI        .EQU    00                ;value = (SYSCLK/32/baud) – 1 =
                                    ;(5 MHz/32/9600) – 1 = 15.27 ~ 15
                                    ;1.8 percent error
          AND     #01Fh, SCICTL     ;Make sure that SCI SW RESET bit is
                                    ;clear before writing to the SCI
                                    ;configuration registers
          MOV     #000h,SCIPRI      ;Set TX and RX to high priority
          MOV     #005h,SCIPC1      ;Set SCLK for general–purpose output
          MOV     #022h,SCIPC2      ;Set pins for RXD and TXD functions
          MOV     #HI,BAUDMSB       ;Set bit rate for 9600 (MSbyte)
          MOV     #B9600,BAUDLSB    ;Set bit rate for 9600 (LSbyte)
          MOV     #077h,SCICCR      ;1 stop bit, even parity,
                                    ;and enable 8 data bits/char
          MOV     #033h,SCICTL      ;Enable Rx, Tx, clock is internal
          MOV     #001h,TXCTL       ;Enable TX interrupt
          MOV     #001h,RXCTL       ;Enable RX interrupt
          EINT                      ;Let the interrupts begin
          MOV     #00,TXBUF         ;Start transmitter by sending null
                                    ;character
```

9.7.2 RS-232-C Multiprocessor Mode Example

This example initializes the transmitter and receiver to accept data at 9600 baud with a format of 8 data bits, 1 stop bit, and even parity. It uses the address bit wake-up mode to implement the multiprocessor protocol.

```
B9600     .EQU    15                ;Value for counter for 9600 baud
HI        .EQU    00                ;value = (SYSCLK/32/baud) – 1 =
                                    ;(5 MHz/32/9600) – 1 = 15.27 ~ 15
                                    ;1.8 percent error
          MOV     #000h,SCIPRI      ;Set TX and RX to high priority
          MOV     #005h,SCIPC1      ;Set SCLK for general–purpose output
          MOV     #022h,SCIPC2      ;Set pins for RXD and TXD functions
          MOV     #HI,BAUDMSB       ;Set bit rate for 9600 (MSbyte)
          MOV     #B9600,BAUDLSB    ;Set bit rate for 9600 (LSbyte)
          MOV     #07Fh,SCICCR      ;1 stop bit, even parity,
                                    ;and enable 8 data bits/char
          MOV     #037h,SCICTL      ;Enable Rx, Tx; RX to sleep,
                                    ;clock is internal
          MOV     #001h,TXCTL       ;Enable TX interrupt
          MOV     #001h,RXCTL       ;Enable RX interrupt
          EINT                      ;Let the interrupts begin
          EINT                      ;
                                    ;MAIN ROUTINES
                                    ;

SENDADD   OR      #8,SCICTL         ;Main line routine; set TXWAKE
                                    ;wake bit
          MOV     ADDR,TXBUF        ;Transmit address stored in ADDR
          RTS

                                    ;INTERRUPT ROUTINES
                                    ;
                                    ;The locations of the SCI transmitter and
                                    ;receiver routines, SENDATA and GETDATA,
                                    ;need to be stored in the interrupt vec
                                    ;tor table at locations 70F0h and 7FF2h,
                                    ;respectively.

                                    ;SCI1 TRANSMITTER INTERRUPT ROUTINE

SENDDATA  PUSH    A                 ;Address has already been sent by
                                    ;the SENDADD
          MOV     OUTDATA,TXBUF     ;Output character that is
                                    ;stored in DATA
          .                         ;
          .                         ;Other transmitter code
          .                         ;
          POP     A                 ;Restore and exit
          RTI
```

```
          .                         ;SCI1 RECEIVER INTERRUPT ROUTINE

GETDATA   PUSH    A                 ;Receive a new character
          BTJZ    #2,RXCTL,ISDATA   ;Is this address or data byte?
          MOV     RXBUF,A           ;Get new character and clear
                                    ;interrupt flag
          CMP     #MYADDR,A         ;Is this my address or
                                    ;another processor's address
          JNE     RXEXIT            ;Exit if another's; still
                                    ;in sleep mode
          AND     #0FBh,SCICTL      ;If my address get out of sleep mode
          JMP     RXEXIT            ;Exit and wait for data
                                    ;
ISDATA    MOV     RXBUF,INDATA      ;Put incoming data in register
          .                         ;
          .                         ;Other receiver code
          .                         ;

RXEXIT    POP     A                 ;Restore and exit
          RTI
```

9.8 SCI1 Control Registers

The SCI1 is controlled and accessed through registers in peripheral file frame 5. These registers are listed in Figure 9–10 and described in the following subsections. The bits shown in *shaded boxes* in Figure 9–10 are privilege mode bits; that is, they can only be written to in the privilege mode.

Figure 9–10. Peripheral File Frame 5: SCI1 Control Registers

Designation	ADDR	PF	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
SCICCR	1050h	P050	STOP BITS (RW–0)	EVEN/ODD PARITY (RW–0)	PARITY ENABLE (RW–0)	ASYNC/ ISOSYNC (RW–0)	ADDRESS/ IDLE WUP (RW–0)	SCI CHAR2 (RW–0)	SCI CHAR1 (RW–0)	SCI CHAR0 (RW–0)
SCICTL	1051h	P051	—	—	SCI SW RESET (RW–0)	CLOCK (RW–0)	TXWAKE (RS–0)	SLEEP (RW–0)	TXENA (RW–0)	RXENA (RW–0)
BAUD MSB	1052h	P052	BAUDF (MSB) (RW–0)	BAUDE (RW–0)	BAUDD (RW–0)	BAUDC (RW–0)	BAUDB (RW–0)	BAUDA (RW–0)	BAUD9 (RW–0)	BAUD8 (RW–0)
BAUD LSB	1053h	P053	BAUD7 (RW–0)	BAUD6 (RW–0)	BAUD5 (RW–0)	BAUD4 (RW–0)	BAUD3 (RW–0)	BAUD2 (RW–0)	BAUD1 (RW–0)	BAUD0 (LSB) (RW–0)
TXCTL	1054h	P054	TXRDY (R–1)	TX EMPTY (R–1)	—	—	—	—	—	SCI TX INT ENA (RW–0)
RXCTL	1055h	P055	RX ERROR (R–0)	RXRDY (R–0)	BRKDT (R–0)	FE (R–0)	OE (R–0)	PE (R–0)	RXWAKE (R–0)	SCI RX INT ENA (RW–0)
	1056h	P056	Reserved							
RXBUF	1057h	P057	RXDT7 (R–0)	RXDT6 (R–0)	RXDT5 (R–0)	RXDT4 (R–0)	RXDT3 (R–0)	RXDT2 (R–0)	RXDT1 (R–0)	RXDT0 (R–0)
	1058h	P058	Reserved							
TXBUF	1059h	P059	TXDT7 (RW–0)	TXDT6 (RW–0)	TXDT5 (RW–0)	TXDT4 (RW–0)	TXDT3 (RW–0)	TXDT2 (RW–0)	TXDT1 (RW–0)	TXDT0 (RW–0)
	105Ah	P05A	Reserved							
	105Bh	P05B								
	105Ch	P05C								
SCIPC1	105Dh	P05D	—	—	—	—	SCICLK DATA IN (R–0)	SCICLK DATA OUT (RW–0)	SCICLK FUNCTION (RW–0)	SCICLK DATA DIR (RW–0)
SCIPC2	105Eh	P05E	SCITXD DATA IN (R–0)	SCITXD DATA OUT (RW–0)	SCITXD FUNCTION (RW–0)	SCITXD DATA DIR (RW–0)	SCIRXD DATA IN (R–0)	SCIRXD DATA OUT (RW–0)	SCIRXD FUNCTION (RW–0)	SCIRXD DATA DIR (RW–0)
SCIPRI	105Fh	P05F	SCI STEST (RP–0)	SCITX PRIORITY (RP–0)	SCIRX PRIORITY (RP–0)	SCI ESPEN (RP–0)	—	—	—	—

Note: Shaded boxes indicate privilege mode.

9.8.1 SCI Communication Control Register (SCICCR)

The SCICCR register defines the character format, protocol, and communications modes used by the SCI1.

SCI Communication Control Register (SCICCR)
[Memory Address 1050h]

Bit #	7	6	5	4	3	2	1	0
P050	STOP BITS	EVEN/ODD PARITY	PARITY ENABLE	ASYNC/ ISOSYNC	ADDRESS/ IDLE WUP	SCI CHAR2	SCI CHAR1	SCI CHAR0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **STOP BITS.** SCI1 Number of Stop Bits.

This bit determines the number of stop bits transmitted. The receiver checks for one stop bit only.

0 = One stop bit
1 = Two stop bits

Bit 6 **EVEN/ODD PARITY.** SCI1 Parity Enable.

If the PARITY ENABLE bit is set, this bit selects odd or even parity (odd or even number of one bits in both transmitted and received characters).

0 = Sets odd parity
1 = Sets even parity

Bit 5 **PARITY ENABLE.** SCI1 Parity Enable.

This bit enables or disables the parity function. When parity is enabled during the address bit multiprocessor mode, the address bit is included in the parity calculation.

0 = Disables parity. No parity bit is generated during transmission or expected during reception.
1 = Enables parity

Bit 4 **ASYNC/ISOSYNC.** SCI1 communications Mode Control Bit.

This bit determines the SCI1 communications mode.

0 = Selects isosynchronous mode. In this mode, the bit period is equal to the SCICLK period; bits are read on a single-sample basis.
1 = Selects asynchronous mode. In this mode, the bit period is 16 times the SCICLK period; bits are read on a two-out-of-three majority basis.

Bit 3 **ADDRESS/IDLE WUP.** SCI1 Multiprocessor Mode Control Bit.

This bit selects the multiprocessor mode.

0 = Selects idle line mode
1 = Selects address bit mode

The idle line mode is usually used for normal communications because the address bit mode adds an extra bit to the frame; the idle line mode does not add this extra bit and is compatible with RS-232-type communications. Multiprocessor communication is different from the other communications modes because it uses TXWAKE and SLEEP functions.

Bits 2–0 **SCI CHAR2–0.** SCI1 Character Length Control Bits 2–0.

These bits select the SCI character (data) bit length, from 1 to 8 bits. Characters of less than 8 bits are right-justified in RXBUF and TXBUF, and are padded with leading 0s in RXBUF. TXBUF need not be padded with leading zeros.

Table 9–4. Character Bit Length

SCI CHAR2	SCI CHAR1	SCI CHAR0	Character Length
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

9.8.2 SCI Control Register (SCICTL)

The SCICTL register controls the RX/TX enable, TXWAKE and SLEEP functions, internal clock enable, and the SCI1 software reset.

SCI Control Register (SCICTL)
[Memory Address 1051h]

Bit #	7	6	5	4	3	2	1	0
P051	—	—	SCI SW RESET	CLOCK	TXWAKE	SLEEP	TXENA	RXENA
			RW-0	RW-0	RS-0	RW-0	RW-0	RW-0

R = Read, W = Write, S = Set only, *-n* = Value of the bit after the register is reset

Bits 7–6 **Reserved.** Read data is indeterminate.

Bit 5 **SCI SW RESET.** SCI1 Software Reset (Active Low).

Writing a 0 to this bit initializes the SCI state machines and operation flags to the reset condition (shown in Table 9–5). The CLOCK bit retains its state prior to the assertion of SCI SW RESET. If SCICLK is configured as an output (by bits SCIPC1.0 and1), the SCICLK resets (low level). All affected logic is held in the reset state until a 1 is written to the SCI SW RESET bit. Thus, after a system reset, you must re-enable the SCI by writing a 1 to this bit. This bit must be cleared after a receiver break detect.

SCI SW RESET affects the operating flags of the SCI1. This bit does not affect the configuration bits, nor does it put in the reset values. The flags listed in Table 9–5 are set to the values shown when SCI SW RESET is cleared. The operating flags are frozen until the SCI SW RESET bit is set again.

Table 9–5. Flags Affected by SCI SW RESET

SCI1 Flag	Designation	Value After SCI SW RESET
TXRDY	TXCTL.7	1
TXEMPTY	TXCTL.6	1
RXWAKE	RXCTL.1	0
PE	RXCTL.2	0
OE	RXCTL.3	0
FE	RXCTL.4	0
BRKDT	RXCTL.5	0
RXRDY	RXCTL.6	0
RX ERROR	RXCTL.7	0

Note: First Clear SCI SW RESET Bit

The SCI SW RESET bit must be cleared before the SCI1 configuration registers can be set up or altered. The application program should set up all configuration registers before it sets the SCI SW RESET bit.

Bit 4 **CLOCK.** SCI1 Internal Clock Enable.

This bit determines the source of the SCICLK. Clearing this bit selects an external SCICLK, which is input on the high-impedance SCICLK line and bypasses the baud generator.

- For isosynchronous transactions, one bit is transmitted or received per SCICLK period.
- For asynchronous transactions, one bit is transmitted or received per 16 SCICLK periods.

The maximum frequency for the externally sourced SCICLK is CLKIN/16. Setting the CLOCK bit selects an internal SCICLK, derived from the baud generator. This signal can be output on the SCICLK line.

0 = External SCICLK
1 = Internal SCICLK

Bit 3 **TXWAKE.** SCI1 Transmitter Wake-up.

The TXWAKE bit controls the transmit features of the multiprocessor communication modes. This bit is cleared only by system reset. The SCI hardware clears this bit, once it has been transferred to wake-up temporary (WUT).

Bit 2. **SLEEP.** SCI1 Sleep.

This bit controls the receive features of the multiprocessor communication modes. You must clear this bit to bring the SCI1 out of sleep mode.

0 = Disables sleep mode
1 = Enables sleep mode

Bit 1 **TXENA.** SCI1 Transmit Enable.

Data transmission through the SCITXD pin occurs only when this bit is set. If this bit is reset, the transmission is not halted until all the data previously written to TXBUF has been sent.

0 = Disables SCI1 transmitter
1 = Enables SCI1 transmitter

Bit 0 **RXENA.** SCI1 Receive Enable.

When this bit is set, received characters are transferred into RXBUF, and the RXRDY flag is set. When cleared, this bit prevents received characters from being transferred into the receiver buffer (RXBUF), and no receiver interrupts are generated. However, the receiver shift register continues to assemble characters. As a result, if RXENA is set during the reception of a character, the complete character is transferred into RXBUF.

0 = Disables SCI1 receiver
1 = Enables SCI1 receive

9.8.3 Baud Select Registers (BAUD MSB and BAUD LSB)

The BAUD MSB and BAUD LSB registers store the value used to generate the bit rate. The SCI1 uses the combined 16-bit value, BAUD REG, of the baud select registers to set the SCI1 clock frequency as follows:

SCICLK frequency = SYSCLK / [(BAUD Reg + 1) $\times$ 2]

or

$$\text{BAUD REG} + 1 = \frac{\text{SYSCLK}}{2(\text{SCICLK})}$$

where

BAUD REG = The 16-bit value in the baud select registers.

For example, if the SYSCLK frequency is 5 MHz, the maximum internal SCICLK frequency would be [5 MHz / 2] or 2.5 MHz.

- For asynchronous mode communication, data is transmitted and received at the rate of one bit for each 16 SCICLK periods.
- For isosynchronous mode communication, data is transmitted and received at the rate of one bit for each SCICLK period.

The asynchronous and isosynchronous bit rates are calculated as follows:

Asynchronous Baud = SYSCLK / [(BAUD REG + 1) $\times$ 32]

Isosynchronous Baud = SYSCLK / [(BAUD REG + 1) $\times$ 2]

Baud Select Register (BAUD MSB)
[Memory Address 1052h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P052	BAUDF (MSB)	BAUDE	BAUDD	BAUDC	BAUDB	BAUDA	BAUD9	BAUD8
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Baud Select Register (BAUD LSB)
[Memory Address 1053h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P053	BAUD7	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0 (LSB)
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

9.8.4 SCI Transmitter Interrupt Control and Status Register (TXCTL)

The TXCTL register contains the transmitter interrupt enable bit, the transmitter ready flag, and the transmitter empty flag. The status flags are updated each time a compete character is transmitted.

SCI Transmitter Interrupt Control and Status Register (TXCTL)
[Memory Address 1054h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P054	TXRDY	TX EMPTY	—	—	—	—	—	SCI TX INT ENA
	R-1	R-1						RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **TXRDY.** SCI1 Transmitter Ready.

The TXRDY bit is set by the transmitter to indicate that TXBUF is ready to receive another character. The bit is automatically cleared when a character is loaded into TXBUF. This flag asserts a transmitter interrupt if the interrupt enable bit SCI TX INT ENA (TXCTL.0) is set. TXRDY is a read-only flag. It is set to 1 by an SCI SW RESET (SCICTL.5) or by a system reset.

0 = TXBUF is full.
1 = TXBUF is ready to receive a character.

Bit 6 **TX EMPTY.** SCI1 Transmitter Empty.

This bit indicates the status of the transmitter-shift register and the TXBUF register. TX EMPTY is set to 1 by an SCI SW RESET or by a system reset.

0 = The CPU has written data to the TXBUF register; the data has not been completely transmitted.
1 = TXBUF and TXSHF registers are empty.

Bits 5–1 **Reserved.** Read data is indeterminate.

Bit 0 **SCI TX INT ENA.** SCI1 Transmitter Ready Interrupt Enable.

This bit controls the ability of the TXRDY bit to request an interrupt but does not prevent the TXRDY bit from being set. The SCI TX INT ENA bit (TXCTL.0) is set to 0 by a system reset.

0 = Disables SCI TXRDY interrupt
1 = Enables SCI TXRDY interrupt

9.8.5 SCI Receiver Interrupt Control and Status Register (RXCTL)

The RXCTL register contains one interrupt enable bit and seven receiver status flags (two of which can generate interrupt requests). The status flags are updated each time a complete character is transferred to the RXBUF and are cleared each time RXBUF is read.

SCI Receiver Interrupt Control and Status Register (RXCTL)
[Memory Address 1055h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P055	RX ERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	SCI RX INT ENA
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **RX ERROR.** SCI1 Receiver Error Flag.

The RX ERROR flag indicates that at least one of the error flags (bits 5–2 – BRKDT, FE, OE, PE) in the RXCTL register is set. It is a logical OR of the parity, overrun, framing error, and break detect flags. The bit can be used for fast error condition checking during the interrupt service routine because a negative value of the status register indicates that an error condition has occurred. This error flag cannot be cleared directly but is cleared when no individual error flags are set. This bit is cleared by an SCI SW RESET, by a system reset, or by reading RXBUF.

0 = No error flag set
1 = Error flag(s) set

Bit 6 **RXRDY.** SCI1 Receiver Ready.

The receiver sets this bit to indicate that RXBUF is ready with a new character and clears the bit when the character is read. A receiver interrupt is generated if the SCI RX INT ENA bit is a 1. RXRDY is reset by an SCI1 SW RESET (SCICTL.5) or by a system reset.

Bit 5 **BRKDT.** SCI1 Break Detect Flag.

The SCI1 sets this bit when a break condition occurs. A break condition occurs when the SCIRXD line remains continuously low for at least 10 bits, beginning after a missing first stop bit. The occurrence of a break causes a receiver interrupt to be generated if the SCI RX INT ENA bit is a 1, but it does not cause the receiver buffer to be loaded. A BRKDT interrupt can occur, even if the receiver SLEEP bit is set to 1.

0 = No break condition occurred
1 = Break condition occurred. Set RX ERROR bit

BRKDT is cleared by an SCI1 SW RESET or by a system reset. It is not cleared by receipt of a character after the break is detected. To receive more charac-

ters, the SCI1 must be reset by toggling the SCI SW RESET bit or by a system reset.

Bit 4 **FE.** SCI1 Framing Error Flag.

The SCI1 sets this bit when it doesn't find an expected stop bit. Only the first stop bit is checked. The missing stop bit indicates that synchronization with the start bit has been lost and that the character is incorrectly framed. It is reset by an SCI1 SW RESET, by a system reset, or by reading RXBUF.

0 = No framing error detected
1 = Framing error detected. Set RX ERROR bit

Bit 3. **OE.** SCI1 Overrun Error Flag.

The SCI1 sets this bit when a character is transferred into RXBUF before the previous character has been read out. The previous character is overwritten and lost. The OE flag is reset by an SCI1 SW RESET, by a system reset, or by reading RXBUF.

0 = No Overrun error detected
1 = Overrun error detected. Set RX ERROR bit

Bit 2 **PE.** SCI1 Parity Error Flag.

This flag bit is set when a character is received with a mismatch between the number of 1s and its parity bit (SCICCR.6 selects parity). The parity checker includes the address bit in the calculation. If parity generation and detection are not enabled, the PE flag is disabled and read as 0. The PE bit is reset by an SCI1 SW RESET, by a system reset, or by reading RXBUF.

0 = No parity error or parity is disabled.
1 = Parity error detected. Set RX ERROR bit

Bit 1 **RXWAKE.** Receiver Wake-up Detect.

The SCI1 sets this bit when a receiver wake-up condition is detected. In the address bit multiprocessor mode (SCICCR.3 = 1), RXWAKE reflects the value of the address bit for the character contained in RXBUF. In the idle line multi-processor mode, RXWAKE is set if an idle SCIRXD line is detected. RXWAKE, a read-only flag, (SCICCR.3 = 0) is cleared by transfer of the first byte after the address byte to RXBUF, by reading the address character in RXBUF, by an SCI SW RESET, or by a system reset.

Bit 0 **SCI RX INT ENA.** SCI1 Receiver Interrupt Enable.

The SCI RX INT ENA bit controls the ability of the RXRDY and the BRKDT bits to request an interrupt but does not prevent these flags from being set.

0 = Disables RXRDY/BRKDT interrupt
1 = Enables RXRDY/BRKDT interrupt

9.8.6 SCI Receiver Data Buffer Register (RXBUF)

The RXBUF register contains current data from the receiver shift register. RXBUF is cleared by a system reset.

SCI Receiver Data Buffer Register (RXBUF)
[Memory Address 1057h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P057	RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

9.8.7 SCI Transmitter Data Buffer Register (TXBUF)

The TXBUF register is a read/write register that stores data bits to be transmitted by SCITX. Data written to TXBUF are right-justified because the leftmost bits are ignored for characters less than eight bits long.

SCI Transmit Data Buffer Register (TXBUF)
[Memory Address 1059h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P059	TXDT7	TXDT6	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

9.8.8 SCI Port Control Register 1 (SCIPC1)

The SCIPC1 register controls the SCICLK pin functions.

SCI Port Control Register 1 (SCIPC1)
[Memory Address 105Dh]

Bit #	7	6	5	4	3	2	1	0
P05D	—	—	—	—	SCICLK DATA IN	SCICLK DATA OUT	SCICLK FUNCTION	SCICLK DATA DIR
					R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bits 7–4 **Reserved.** Read data is indeterminate.

Bit 3 **SCICLK DATA IN.**

The SCICLK DATA IN bit contains the current value on the SCICLK pin.

Bit 2 **SCICLK DATA OUT.**

This bit contains the data to be output on the SCICLK pin if the following conditions are met:

- Pin SCICLK is a general-purpose I/O.
- Pin SCICLK's data direction is defined as output.

Bit 1 **SCICLK FUNCTION.**

This bit defines the function of the SCICLK pin.

0 = Pin SCICLK is a general-purpose digital I/O pin.
1 = Pin SCICLK is the SCI serial clock pin.

BIT 0 **SCICLK DATA DIR.** SCICLK Data Direction.

This bit determines the data direction on the SCICLK pin if SCICLK has been configured as a general-purpose I/O pin.

0 = Pin SCICLK is a general-purpose input pin.
1 = Pin SCICLK is a general-purpose output pin.

9.8.9 SCI Port Control Register 2 (SCIPC2)

The SCIPC2 register controls the SCIRXD and SCITXD pin functions.

SCI Port Control Register 2 (SCIPC2)
[Memory Address 105Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P05E	SCITXD DATA IN	SCITXD DATA OUT	SCITXD FUNCTION	SCITXD DATA DIR	SCIRXD DATA IN	SCIRXD DATA OUT	SCIRXD FUNCTION	SCIRXD DATA DIR
	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **SCITXD DATA IN.**

This bit contains the current value on the SCITXD pin.

Bit 6 **SCITXD DATA OUT.**

This bit contains the data to be output on the SCITXD pin if the following conditions are met:

- ❏ Pin SCITXD has been defined as a general-purpose I/O pin.
- ❏ Pin SCITXD's data direction has been defined as output.

Bit 5 **SCITXD FUNCTION.**

This bit defines the function of the SCITXD pin.

0 = Pin SCITXD is a general-purpose digital I/O pin.
1 = Pin SCITXD is the SCI1 transmit pin.

Bit 4 **SCITXD DATA DIR.** SCITXD Data Direction.

This bit determines the data direction on the SCITXD pin if SCITXD has been defined as a general-purpose I/O pin.

0 = Pin SCITXD is a general-purpose input pin.
1 = Pin SCITXD is a general-purpose output pin.

Bit 3 **SCIRXD DATA IN.**

This bit contains the current value on the SCIRXD pin.

Bit 2 **SCIRXD DATA OUT.**

This bit contains the data to be output on the SCIRXD pin if the following conditions are met:

- ❏ Pin SCIRXD has been defined as a general-purpose I/O pin.
- ❏ Pin SCIRXD's data direction has been defined as output.

Bit 1 **SCIRXD FUNCTION.**

This bit defines the function of the SCIRXD pin.

0 = Pin SCIRXD is a general-purpose digital I/O pin.
1 = Pin SCIRXD is the SCI1 receiver pin.

Bit 0 **SCIRXD DATA DIR.** SCIRXD Data Direction.

This bit determines the data direction on the SCIRXD pin if SCIRXD has been defined as a general-purpose I/O pin.

0 = Pin SCIRXD is a general-purpose input pin.
1 = Pin SCIRXD is a general-purpose output pin.

9.8.10 SCI Priority Control Register (SCIPRI)

The SCIPRI register contains the receiver and transmitter interrupt priority select bits. This register is read-only during normal operation but can be written to in the privilege mode.

SCI Priority Control Register (SCIPRI)
[Memory Address 105Fh]

Bit #	7	6	5	4	3	2	1	0
P05F	SCI STEST	SCITX PRIORITY	SCIRX PRIORITY	SCI ESPEN	—	—	—	—
	RP-0	RP-0	RP-0	RP-0				

R = Read, W = Privilege write only, *-n* = Value of the bit after the register is reset

Bit 7 **SCI STEST.** SCI1 STEST.

This bit must be cleared to ensure proper operation.

Bit 6 **SCI TX PRIORITY.** SCI1 Transmitter Interrupt Priority Select.

This bit assigns the interrupt priority level of the SCI1 transmitter interrupts.

0 = Transmitter interrupts are level 1 (high-priority) requests.
1 = Transmitter interrupts are level 2 (low-priority) requests.

Bit 5 **SCI RX PRIORITY.** SCI1 Receiver Interrupt Priority Select.

This bit assigns the interrupt priority level of the SCI1 receiver interrupts.

0 = Receiver interrupts are level 1 (high-priority) requests.
1 = Receiver interrupts are level 2 (low-priority) requests.

Bit 4 **SCI ESPEN.** SCI1 Emulator Suspend Enable.

This bit has no effect except when you are using the XDS emulator to debug a program. Then, this bit determines how the SCI1 operates when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the SCI1 continues to work until the current transmit or receive sequence is complete.
1 = When the emulator is suspended, the SCI1 state machine is frozen so that the state of the SCI1 can be examined at the point that the emulator was suspended.

Bits 3–0 **Reserved.** Read data is indeterminate.

Chapter 10

Serial Communications Interface 2 (SCI2) Module

The SCI is described in Chapter 9 (*SCI1*: 3 I/O pins, asynchronous *and* isosynchronous modes) and Chapter 10 (*SCI2*: 2 I/O pins, asynchronous mode). This chapter describes SCI2 and covers the following topics:

Topic **Page**

10.1 SCI2 Overview

The serial communications interface (SCI2) module is a programmable I/O port that facilitates digital communications between the TMS370 device and other asynchronous peripherals and uses the standard NRZ (nonreturn to zero) format. The SCI2 transmits and receives serial data, one bit at a time, at a programmable bit rate. Both the SCI2 receiver and transmitter are double-buffered and have their own separate enable and interrupt bits. They can be operated independently or simultaneously in the full duplex mode. The SCI2 module is available only in the TMS370CxCx family.

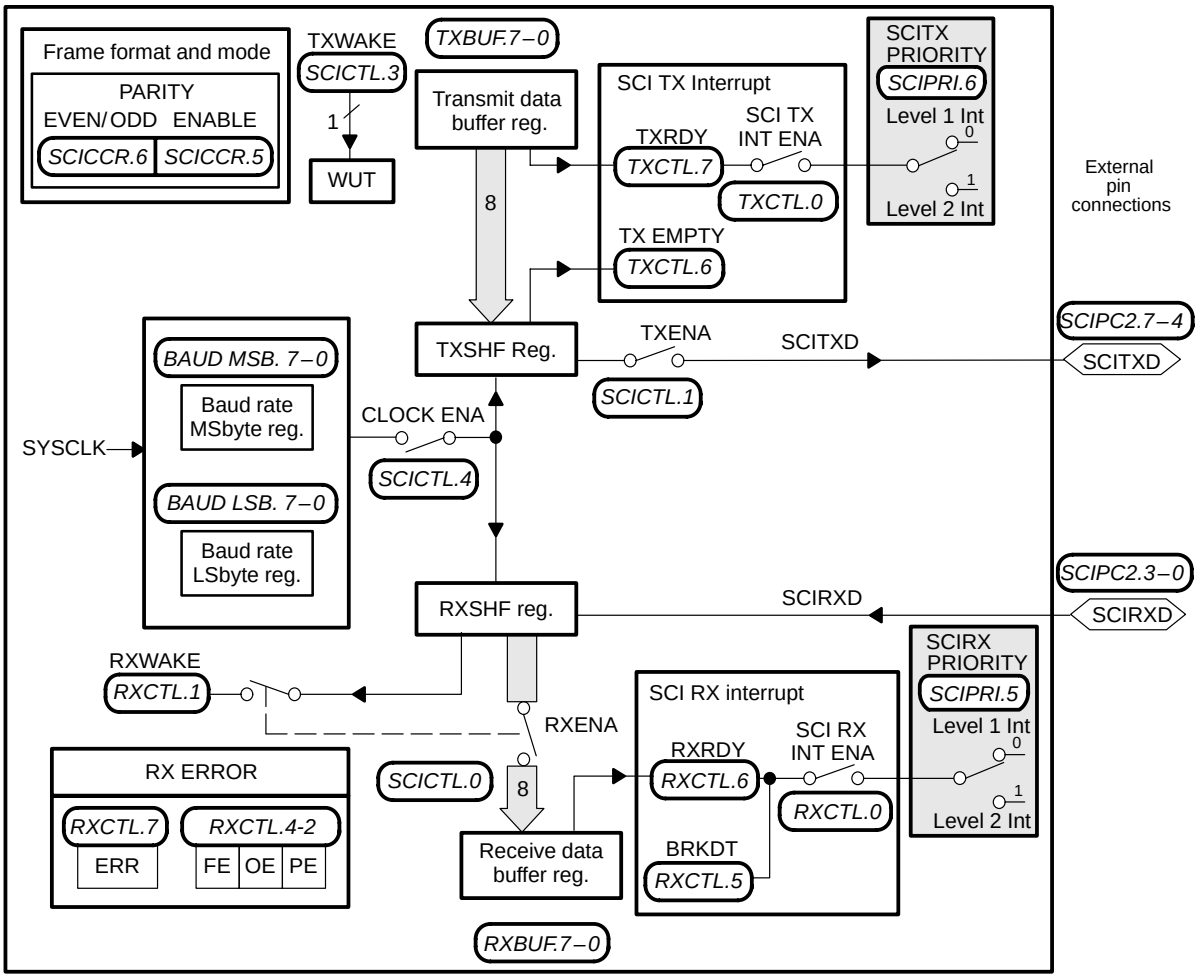
10.1.1 Physical Description

The two-pin SCI2 module, shown in Figure 10–1 key features:

- Two I/O pins:
  - SCIRXD (SCI2 receive data input)
  - SCITXD (SCI2 transmit data output)
- Asynchronous communications format with programmable bit rates to over 65,000 different speed through a 16-bit baud select register
  - Range at 5 MHz SYSCLK—3 bits/s to 156 kbits/s
  - Number of bit rates—64K
- Programmable data word length from 1 to 8 bits
- Programmable stop bits of either 1 or 2 bits in length
- Error detection flags that ensure data integrity:
  - Parity error
  - Overrun error
  - Framing error
  - Break detect
- Two wake-up multiprocessor modes that can be used with either communications format:
  - Idle line wake-up
  - Address bit wake-up
- Full duplex operation
- Separate transmitter and receiver interrupts for polled or interrupt-driven operation
- Double-buffered receive and transmit functions

- Separate enable bits for the transmitter and receiver
- NRZ (nonreturn to zero) format

Figure 10–1. SCI2 Block Diagram



10.1.2 Architecture

The major elements of the full-duplex SCI2 are shown in Figure 10–1 and include:

- A transmitter (SCITX)
  - TXBUF — the transmitter buffer register that contains data, written by the CPU, to be transmitted
  - TXSHF — the transmitter shift register that is loaded from TXBUF and shifts data onto the SCITXD pin, one bit at a time

- A receiver (SCIRX)
  - RXSHF — the receiver shift register that shifts data in from the SCIRXD pin, one bit at a time
  - RXBUF — the receiver buffer register that contains data that is to be read by the CPU and that is received from remote processor and loaded from RXSHF
- A programmable baud generator
- Memory-mapped control and status registers

The SCI2 receiver and transmitter can operate independently and simultaneously.

10.1.3 Communications Modes and Multiprocessing Modes

The SCI2 offers the following universal asynchronous receiver/transmitter (UART) communications modes for interfacing with many popular peripherals. Asynchronous mode (discussed in subsection 9.4.1 on page 9-12) requires two lines to interface with many standard devices such as terminals and printers that use RS-232-C formats.

This mode can be programmed to contain:

- 1 start bit,
- 1 to 8 data bits,
- An even/odd parity bit or no parity bit, and
- 1 or 2 stop bits.

The SCI2 also has two multiprocessor modes: the idle line multiprocessor mode (see subsection 10.3.1) and the address bit multiprocessor mode (see subsection 10.3.2). These modes allow efficient data transfer between multiple processors.

10.1.4 Control Registers

The SCI2 control registers are located at addresses 1050h to 105Fh and occupy peripheral file frame 5. The function of each location is shown in Table 12–1.

Table 10–1. SCI2 Memory Map

Peripheral File Location	Register Symbol	Name	Description	See Page
P050	SCICCR	SCI Communication Control Register	Defines the character format, protocol, and asynchronous enable used by the SCI2.	10-18
P051	SCICTL	SCI Control Register	Controls the RX/TX enable, TXWAKE and SLEEP functions, internal clock enable, and the SCI2 software reset.	10-20
P052	BAUD MSB	Baud Select MSbyte Register	Stores the data required to generate the bit rate.	10-22
P053	BAUD LSB	Baud Select LSbyte Register		
P054	TXCTL	SCI Transmitter Interrupt Control and Status Register	Contains the transmitter interrupt enable, the transmitter ready flag, and the transmitter empty flag.	10-23
P055	RXCTL	SCI Receiver Interrupt Control and Status Register	Contains one interrupt enable bit and seven receiver status flags.	10-24
P056		Reserved		
P057	RXBUF	SCI Receiver Data Buffer	Contains the current data from the receiver shift register.	10-26
P058		Reserved		
P059	TXBUF	SCI Transmit Data Buffer	Stores data bits to be transmitted by the SCITX.	10-26
P05A–P05D		Reserved		
P05E	SCIPC2	SCI Port Control Register 2	Controls the SCIRXD and SCITXD pin functions.	10-27
P05F	SCIPRI	SCI Interrupt Priority Control Register	Contains the receiver and transmitter interrupt priority select bits.	10-29

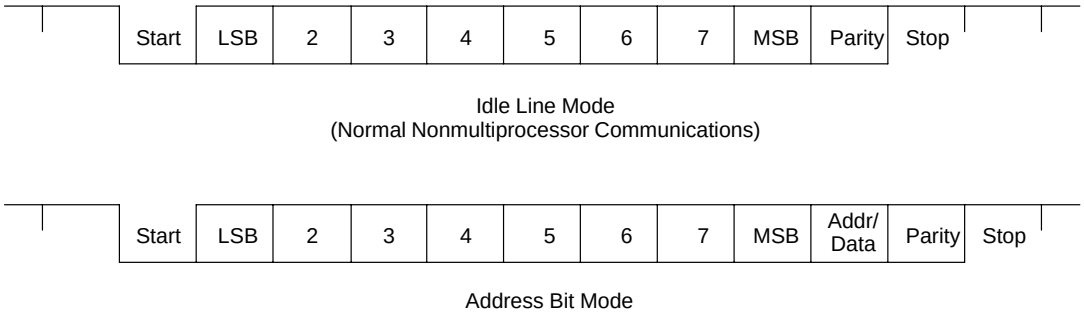
10.2 Programmable Data Format

SCI2 data, both receive and transmit, is in NRZ (nonreturn to zero) format, which means that in active state the SCIRX and SCITX lines will be held logic one. The NRZ data format is illustrated in Figure 10–2 and consists of:

- 1 start bit
- 1 to 8 data bits
- An even/odd parity bit (optional)
- 1 or 2 stop bits
- An extra bit to distinguish addresses from data (address bit mode only).

The basic unit of data is called a character and is 1 to 8 bits in length. Each character of data is formatted with a start bit, 1 or 2 stop bits, and optional parity and address bits. A character of data along with its formatting information is called a frame and is shown in Figure 10–2.

Figure 10–2. SCI2 Data Formats



To program the data format, use the SCICCR register (described in subsection 10.8.1 on page 10-18). The bits that you use to program the data format are shown in Table 10–2:

Table 10–2. Programming the Data Format Using SCICCR

Bit Name	Designation	Function
SCI CHAR0–2	SCICCR.0–2	Select the character (data) length (1 to 8 bits). Refer to the bit listings on page 10-18 and 10-19 for additional information.
PARITY ENABLE	SCICCR.5	Enables the parity function if set to 1 or disables the parity function if cleared to 0.
EVEN/ODD PARITY	SCICCR.6	If parity is enabled, selects odd parity if cleared to 0 or even parity if set to 1.
STOP BITS	SCICCR.7	Determines the number of stop bits transmitted—one stop bit if cleared to 0 or two stop bits if set to 1.

10.3 Multiprocessor Communications

The multiprocessor communication format allows one processor to efficiently send blocks of data to other processors on the same serial link. You can have only one talker on a serial line at a time.

The first byte of a block of information that the talker sends contains an address byte that is read by all listeners. Only listeners with the correct address can be interrupted by the data bytes that follow the address byte. The listeners with an incorrect address remain uninterrupted until the next address byte.

All processors on the serial link set their SLEEP bit (SCICTL.2) to 1 so that they are interrupted only when the address byte is detected. When a processor reads a block address that corresponds to the CPU's device address as set by software, your program must clear the SLEEP bit to enable the SCI2 to generate an interrupt on receipt of each data byte.

Although the receiver still operates when the SLEEP bit is 1, it does not set RXRDY, RXINT, or the error status bits to 1 unless the address byte is detected and the address bit in the received frame is a 1. The SCI2 does not alter the SLEEP bit; your software must alter the SLEEP bit.

A processor recognizes an address byte according to the multiprocessor mode:

- The **idle line mode** leaves a quiet space before the address byte. This mode does not have an extra address/data bit and is more efficient than the address bit mode in handling blocks that contain more than 10 bytes of data.
- The **address bit mode** adds an extra bit into every byte to distinguish addresses from data. This mode is more efficient in handling many small blocks of data because, unlike the idle mode, it does not have to wait between blocks of data. However, at high transmit speeds, the program is not fast enough to avoid a 10-bit idle in the transmission stream.

You can select the multiprocessor mode via the ADDRESS/IDLE WUP bit (SCICCR.3). Both modes use the TXWAKE flag bit (SCICTL.3), RXWAKE flag bit (RXCTL.1), and the SLEEP flag bit (SCICTL.2) to control the SCITX and SCIRX features of these modes.

In both multiprocessor modes, the sequence is:

1) The SCI2 port wakes up (requests an interrupt) at the start of a block and reads the first frame that contains the destination address.
2) A software routine is entered through the interrupt which checks the RXWAKE flag bit. If the RXWAKE flag bit is a 1, the incoming byte is an

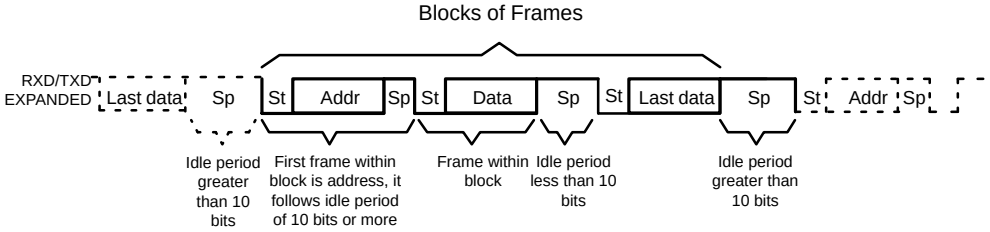
address (otherwise the byte is data), and the incoming byte is checked against the device address byte stored in memory.

3) If the check shows that the block is addressed to the microcontroller, the CPU clears the SLEEP bit and reads the rest of the block; if not, the software routine exits with the SLEEP bit still set and does not receive SCI2 interrupts until the next block start.

10.3.1 Idle Line Multiprocessor Mode

In the idle line multiprocessor mode (ADDRESS/IDLE WUP bit = 0), blocks are separated by having a longer idle time between the blocks than between frames in the blocks. An idle time of 10 or more bits after a frame indicates the start of a new block. The idle line multiprocessor communication format is shown in Figure 10–3. (The ADDRESS/IDLE WUP bit is SCICCR.3.)

Figure 10–3. Idle Line Multiprocessor Communication Format



Note: In the figure, “St” = start and “Sp” = stop

There are two ways to send a block start signal.

- The first method is to deliberately leave an idle time of 10 bits or more by delaying the time between the transmission of the last frame of data in the previous block and the transmission of the address frame of the new block.
- In the second method, the SCI2 port uses the TXWAKE bit (SCICTL.3) to send an idle time of exactly 11 bits. Therefore, the serial communications line is not idle any longer than necessary.

Associated with the TXWAKE bit is the wake-up temporary or WUT flag bit. WUT is an internal flag, double buffered with TXWAKE. When TXSHF is loaded form TXBUF, WUT is loaded from TXWAKE, and the TXWAKE bit is cleared to 0. This arrangement is shown in Figure 10–4.

Figure 10–4. Double-Buffered WUT and TXSHF

To send out a block start signal of exactly one frame time:

1) Write a 1 to the TXWAKE bit.

2) Write a data word (don't care) to TXBUF. (The first data word written is suppressed while the block start signal is sent out, and ignored after that.)

   When TXSHF is free again, TXBUF's contents are shifted to TXSHF, the TXWAKE value is shifted to WUT, and then the TXWAKE bit is cleared. If TXWAKE bit was set to a 1, the start, data, and parity bits are replaced by an idle period of 11 bits transmitted following the last stop bit of the previous frame.

3) Write an address value to the TXBUF.

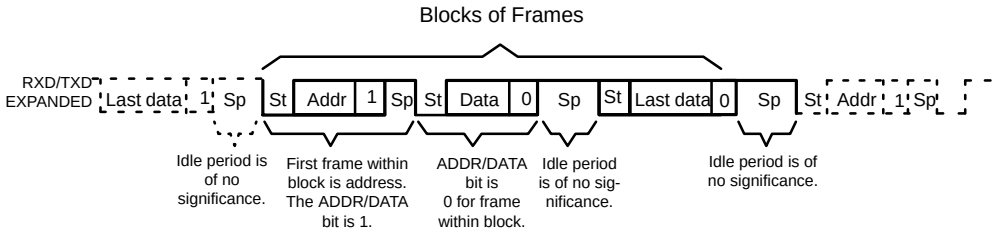
The receiver operates regardless of the SLEEP bit. The receiver does not set RXRDY, RXINT, or the error status bits until an address frame is detected.

10.3.2 Address Bit Multiprocessor Mode

In the address bit mode (ADDRESS/IDLE WUP bit = 1), frames have an extra bit, called an address bit, that immediately follows the last data bit. The address bit is set to 1 in the first frame of the block and to 0 in all other frames. The idle period timing is irrelevant. (The ADDRESS/IDLE WUP bit is SCICCR.3.)

The TXWAKE bit sets the address bit. In SCITX, when the TXBUF and TXWAKE are loaded into TXSHF and WUT, TXWAKE is reset to 0, and WUT is the value of the address bit of the current frame. Thus, to send an address, set the TXWAKE bit to a 1 and write the appropriate address value to the TXBUF. When this address value is transferred to TXSHF and shifted out, its address bit is sent as a 1, which flags the other processors on the serial link to read the address. Since TXSHF and WUT are both double-buffered, TXBUF and TXWAKE can be written to immediately after TXSHF and WUT are loaded. To transmit nonaddress frames in the block, leave the TXWAKE bit at 0.

Figure 10–5. Address Bit Multiprocessor Communication Format



Note: In the figure, "St" = start and "Sp" = stop

10.4 Asynchronous Communications Mode

The SCI2 asynchronous communication mode uses either single-line (one-way) or two-line (two-way) communications. In this mode, the frame consists of a start bit, one to eight data bits, an optional even/odd parity bit, and one or two stop bits.

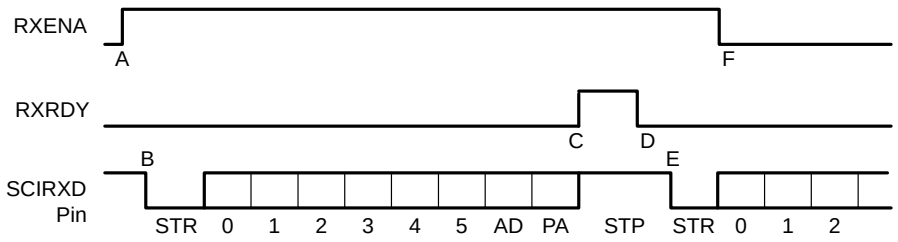
The receiver begins operation on receipt of a valid start bit. A valid start bit consists of eight consecutive zero bits. If any bit is not zero, then the processor starts over and begins looking for another start bit.

10.4.1 Receiver Signals in the Communication Mode

Figure 10–6 illustrates receiver signal timing that assumes these conditions:

- Address bit wake-up mode (address bit would not appear in idle line mode)
- 6 bits per character

Figure 10–6. SCI2 RX Signals in Communications Modes



A) RX ENA goes high to enable the receiver.

B) Data arrives on the SCIRXD pin; start bit detected.

C) RXRDY goes high to signal that a new character has been received; data is shifted to RXBUF; an interrupt is requested.

D) The program reads the RXBUF register; RXRDY is automatically cleared.

E) The next byte of data arrives on the SCIRXD pin; start bit detected, then cleared.

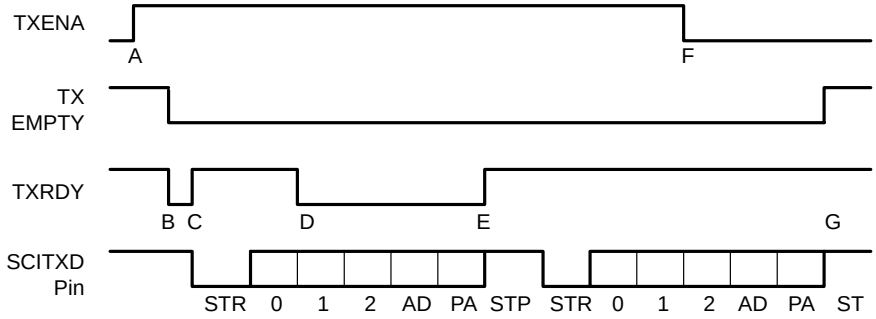
F) RX ENA goes low to disable the receiver; data continues to be assembled in the RXSHF register but is not transferred to the RXBUF register.

10.4.2 Transmitter Signals in the Communication Mode

Figure 10–7 illustrates transmitter signal timing that assumes these conditions:

- Address bit wake-up mode (address bit would not appear in idle line mode)
- 3 bits per character

Figure 10–7. SCI2 TX Signals in the Communication Mode



A) TX ENA goes high to enable the transmitter to send data.

B) Write to TXBUF; TXBUF is no longer empty.

C) SCI transfers data to shift register; TXBUF is ready for new character and requests an interrupt.

D) Program writes new character to TXBUF after TXRDY goes high (item C).

E) Finished transmitting first character; transfer new character to shift register.

F) TX ENA goes low to disable transmitter; SCI finishes transmitting current character.

G) Finished transmitting character; TXBUF is empty and ready for new character.

10.5 Port Interrupts

The SCI2 provides independent interrupt requests and vectors for the receiver and transmitter.

- If the SCI RX INT ENA bit (RXCTL.0) is set, the receiver interrupt is asserted when one of the following events occurs:
  - The SCI2 receives a complete frame and transfers the data in the RXSHF register to the RXBUF register. This action sets the RXRDY flag (RXCTL.6) and initiates an interrupt.
  - A break detect condition occurs (the SCIRXD is low for 10 bit periods following a stop bit). This action sets the BRKDT flag bit (RXCTL.5) and initiates an interrupt.
- If the SCI TX INT ENA bit (TXCTL.0) is set, the transmitter interrupt is asserted whenever the data in the TXBUF register is transferred to the TXSHF register, indicating that the CPU can write to the TXBUF; this action sets the TXRDY flag bit (TXCTL.7) and initiates an interrupt.

SCI2 interrupts can be programmed onto different priority levels by the SCI RX PRIORITY (SCIPRI.5) and SCI TX PRIORITY (SCIPRI.6) control bits. When both RX and TX interrupt requests are made on the same level, the receiver always has higher priority than the transmitter; this reduces the possibility of receiver overrun.

10.6 Clock Source

The SCI2 port can be driven by an internal or external baud generator. The internally generated serial clock is determined by the TMS370 SYSCLK frequency and the baud select registers. The SCI2 uses the 16-bit value of the baud select registers to select one of 64K different serial clock rates for the asynchronous communication mode in the following manner:

- Asynchronous Baud = SYSCLK / [(BAUD REG + 1) × 32]

  BAUD REG = [SYSCLK / (Asynchronous Baud × 32)] – 1

where

BAUD REG = The 16-bit value in the baud select registers.

Refer to Table 10–3.

Table 10–3. Asynchronous Baud Register Values for Common SCI2 Bit Rates

	SYSCLK Frequency (MHz)							
	0.6144‡		1.8432‡		4.9152		5.0	
Baud	Baud Reg†	% Error	Baud Reg†	% Error	Baud Reg†	% Error	Baud Reg†	% Error
75	255	0.00	767	0.00	2047	0.00	2082	0.02
300	63	0.00	191	0.00	511	0.00	520	-0.03
600	31	0.00	95	0.00	255	0.00	259	0.16
1200	15	0.00	47	0.00	127	0.00	129	0.16
2400	7	0.00	23	0.00	63	0.00	64	0.16
4800	3	0.00	11	0.00	31	0.00	32	-1.38
9600	1	0.00	5	0.00	15	0.00	15	1.73
19200	0	0.00	2	0.00	7	0.00	7	1.73
38400	-	-	-	-	3	-	3	1.73
156000	-	-	-	-	-	-	0	0.16

† Baud Reg = 16-bit baud register value
‡ Divide-by-1 clock can only operate from a minimum of 2 MHz SYSCLK to a maximum of 5 MHz SYSCLK.

10.7 Initialization Examples

This section contains two examples that initialize the serial port. In each example, the data is moved to and from the buffers in the interrupt routines.

1) The first example shows a typical RS-232 application that connects to a terminal.

2) The second example illustrates the address bit mode in a multiprocessor application.

In both examples, assume that the register mnemonics have been equated (EQU) with the corresponding peripheral-file location. For more examples using the TMS370 SCI, consult *Using the TMS370 SPI and SCI Modules Application Report*.

10.7.1 RS-232-C Example

This example initializes the transmitter and receiver to accept data at 9600 baud with a format of 8 data bits, 1 stop bit, and even parity.

```
B9600     .EQU    15                ;Value for counter. For 9600 baud
HI        .EQU    00                ;value = (SYSCLK/32/baud) – 1 =
                                    ;(5 MHz/32/9600) – 1 = 15.27 ~ 5
                                    ;1.8 percent error
          AND     #01Fh, SCICTL     ;Make sure that SCI SW RESET bit is
                                    ;clear before writing to the SCI
                                    ;configuration registers
          MOV     #000h,SCIPRI      ;Set TX and RX to high priority
          MOV     #022h,SCIPC2      ;Set pins for RXD and TXD functions
          MOV     #HI,BAUDMSB       ;Set bit rate for 9600 (MSbyte)
          MOV     #B9600,BAUDLSB    ;Set bit rate for 9600 (LSbyte)
          MOV     #077h,SCICCR      ;1 stop bit, even parity, enable
                                    ;asynchronous, and enable 8 data bits/char
          MOV     #033h,SCICTL      ;Enable Rx and Tx, enable internal clock
          MOV     #001h,TXCTL       ;Enable TX interrupt
          MOV     #001h,RXCTL       ;Enable RX interrupt
          EINT                      ;Let the interrupts begin
          MOV     #00,TXBUF         ;Start transmitter by sending null
                                    ;character
```

10.7.2 RS-232-C Multiprocessor Mode Example

This example initializes the transmitter and receiver to accept data at 9600 baud with a format of 8 data bits, 1 stop bit, and even parity. It uses the address bit wake-up mode to implement the multiprocessor protocol.

```
B9600     .EQU    15                ;Value for counter for 9600 baud
HI        .EQU    00                ;value = (SYSCLK/32/baud) – 1 =
                                    ;(5 MHz/32/9600) – 1 = 15.27 ~ 15
                                    ;1.8 percent error
          MOV     #000h,SCIPRI      ;Set TX and RX to high priority
          MOV     #022h,SCIPC2      ;Set pins for RXD and TXD functions
          MOV     #HI,BAUDMSB       ;Set bit rate. For 9600 (MSbyte)
          MOV     #B9600,BAUDLSB    ;Set bit rate for 9600 (LSbyte)
          MOV     #07Fh,SCICCR      ;1 stop bit, even parity, enable
                                    ;asynchronous, and enable 8 data bits/char
          MOV     #037h,SCICTL      ;Enable Rx, Tx; RX to sleep, enable
                                    ;internal clock
          MOV     #001h,TXCTL       ;Enable TX interrupt
          MOV     #001h,RXCTL       ;Enable RX interrupt
          EINT                      ;Let the interrupts begin
                                    ;
                                    ;MAIN ROUTINES
                                    ;

SENDADD   OR      #8,SCICTL         ;Main line routine; set TXWAKE
                                    ;wake bit
          MOV     ADDR,TXBUF        ;Transmit address stored in ADDR
          RTS
                                    ;INTERRUPT ROUTINES
                                    ;
                                    ;The locations of the SCI transmitter and
                                    ;receiver routines, SENDATA and GETDATA,
                                    ;need to be stored in the interrupt vector
                                    ;table at locations 70F0h and 7FF2h,
                                    ;respectively.
```

```
                                    ;SCI2 TRANSMITTER INTERRUPT ROUTINE

SENDDATA  PUSH    A                 ;Address has already been sent by
                                    ;the SENDADD
          MOV     OUTDATA,TXBUF     ;Output character that is
                                    ;stored in DATA
          .                         ;
          .                         ;Other transmitter code
          .                         ;
          POP     A                 ;Restore and exit
          RTI
          .                         ;SCI2 RECEIVER INTERRUPT ROUTINE

GETDATA   PUSH    A                 ;Receive a new character
          BTJZ    #2,RXCTL,ISDATA   ;Is this address or data byte?
          MOV     RXBUF,A           ;Get new character and clear
                                    ;interrupt flag
          CMP     #MYADDR,A         ;Is this my address or
                                    ;another processor's address
          JNE     RXEXIT            ;Exit if another's; still
                                    ;in sleep mode
          AND     #0FBh,SCICTL      ;If my address get out of sleep mode
          JMP     RXEXIT            ;Exit and wait for data
                                    ;
ISDATA    MOV     RXBUF,INDATA      ;Put incoming data in register
          .                         ;
          .                         ;Other receiver code
          .                         ;

RXEXIT    POP     A                 ;Restore and exit
          RTI
```

10.8 SCI2 Control Registers

The SCI2 is controlled and accessed through registers in peripheral file frame 5. These registers are listed in Table 10–4 and described in the following subsections. The bits shown in *shaded boxes* in Table 10–4 are privilege mode bits; that is, they can only be written to in the privilege mode.

Table 10–4. Peripheral File Frame 5: SCI2 Control Registers

Designation	ADDR	PF	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
SCICCR	1050h	P050	STOP BITS (RW–0)	EVEN/ODD PARITY (RW–0)	PARITY ENABLE (RW–0)	ASYNC ENABLE (RW–0)	ADDRESS/ IDLE WUP (RW–0)	SCI CHAR2 (RW–0)	SCI CHAR1 (RW–0)	SCI CHAR0 (RW–0)
SCICTL	1051h	P051	—	—	SCI SW RESET (RW–0)	CLOCK ENABLE (RW–0)	TXWAKE (RS–0)	SLEEP (RW–0)	TXENA (RW–0)	RXENA (RW–0)
BAUD MSB	1052h	P052	BAUDF (MSB) (RW–0)	BAUDE (RW–0)	BAUDD (RW–0)	BAUDC (RW–0)	BAUDB (RW–0)	BAUDA (RW–0)	BAUD9 (RW–0)	BAUD8 (RW–0)
BAUD LSB	1053h	P053	BAUD7 (RW–0)	BAUD6 (RW–0)	BAUD5 (RW–0)	BAUD4 (RW–0)	BAUD3 (RW–0)	BAUD2 (RW–0)	BAUD1 (RW–0)	BAUD0 (LSB) (RW–0)
TXCTL	1054h	P054	TXRDY (R–1)	TX EMPTY (R–1)	—	—	—	—	—	SCI TX INT ENA (RW–0)
RXCTL	1055h	P055	RX ERROR (R–0)	RXRDY (R–0)	BRKDT (R–0)	FE (R–0)	OE (R–0)	PE (R–0)	RXWAKE (R–0)	SCI RX INT ENA (RW–0)
	1056h	P056	Reserved							
RXBUF	1057h	P057	RXDT7 (R–0)	RXDT6 (R–0)	RXDT5 (R–0)	RXDT4 (R–0)	RXDT3 (R–0)	RXDT2 (R–0)	RXDT1 (R–0)	RXDT0 (R–0)
	1058h	P058	Reserved							
TXBUF	1059h	P059	TXDT7 (RW–0)	TXDT6 (RW–0)	TXDT5 (RW–0)	TXDT4 (RW–0)	TXDT3 (RW–0)	TXDT2 (RW–0)	TXDT1 (RW–0)	TXDT0 (RW–0)
	105Ah	P05A	Reserved							
	105Bh	P05B								
	105Ch	P05C								
	105Dh	P05D								
SCIPC2	105Eh	P05E	SCITXD DATA IN (R–0)	SCITXD DATA OUT (RW–0)	SCITXD FUNCTION (RW–0)	SCITXD DATA DIR (RW–0)	SCIRXD DATA IN (R–0)	SCIRXD DATA OUT (RW–0)	SCIRXD FUNCTION (RW–0)	SCIRXD DATA DIR (RW–0)
SCIPRI	105Fh	P05F	SCI STEST (RP–0)	SCITX PRIORITY (RP–0)	SCIRX PRIORITY (RP–0)	SCI ESPEN (RP–0)	—	—	—	—

10.8.1 SCI Communication Control Register (SCICCR)

The SCICCR register defines the character format, protocol, and communications modes used by the SCI2.

SCI Communication Control Register (SCICCR)
[Memory Address 1050h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P050	STOP BITS	EVEN/ODD PARITY	PARITY ENABLE	ASYNC ENABLE	ADDRESS/ IDLE WUP	SCI CHAR2	SCI CHAR1	SCI CHAR0
	RW-0	RW-0	RW-0		RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **STOP BITS.** SCI2 Number of Stop Bits.

This bit determines the number of stop bits transmitted. The receiver checks for one stop bit only.

0 = One stop bit.
1 = Two stop bits.

Bit 6 **EVEN/ODD PARITY.** SCI2 Parity Enable.

If the PARITY ENABLE bit is set, this bit selects odd or even parity (odd or even number of one bits in both transmitted and received characters).

0 = Sets odd parity.
1 = Sets even parity.

Bit 5 **PARITY ENABLE.** SCI2 Parity Enable.

This bit enables or disables the parity function. When parity is enabled during the address bit multiprocessor mode, the address bit is included in the parity calculation.

0 = Disables parity. No parity bit is generated during transmission or expected during reception.
1 = Enables parity.

Bit 4 **ASYNC ENABLE.** SCI2 Asynchronous Mode Enable.

This bit enables the asynchronous mode function. For SCI2 operation, this bit must be written as a 1 when you write to the SCICCR register.

0 = Disables asynchronous mode (SCI2 will not operate).
1 = Enables asynchronous mode (SCI2 operates).

Bit 3 **ADDRESS/IDLE WUP.** SCI2 Multiprocessor Mode Control Bit.

This bit selects the multiprocessor mode.

0 = Selects idle line mode.
1 = Selects address bit mode.

The idle line mode is usually used for normal communications because the address bit mode adds an extra bit to the frame; the idle line mode does not add this extra bit and is compatible with RS-232-type communications. Multiprocessor communication is different from the other communications modes because it uses TXWAKE and SLEEP functions.

Bits 2–0 **SCI CHAR2–0.** SCI2 Character Length Control Bits 2–0.

These bits select the SCI2 character (data) bit length, from 1 to 8 bits. Characters of less than 8 bits are right-justified in RXBUF and TXBUF, and are padded with leading 0s in RXBUF. TXBUF need not be padded with leading zeros.

Table 10–5. Character Bit Length

SCI CHAR2	SCI CHAR1	SCI CHAR0	Character Length
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

10.8.2 SCI Control Register (SCICTL)

The SCICTL register controls the RX/TX enable, TXWAKE and SLEEP functions, internal clock enable, and the SCI2 software reset.

SCI Control Register (SCICTL)
[Memory Address 1051h]

Bit #	7	6	5	4	3	2	1	0
P051	—	—	SCI SW RESET	CLOCK ENABLE	TXWAKE	SLEEP	TXENA	RXENA
			RW-0		RS-0	RW-0	RW-0	RW-0

R = Read, W = Write, S = Set only, -*n* = Value of the bit after the register is reset

Bits 6–7 **Reserved.** Read data is indeterminate.

Bit 5 **SCI SW RESET.** SCI2 Software Reset (Active Low).

Writing a 0 to this bit initializes the SCI2 state machines and operation flags to the reset condition. All affected logic is held in the reset state until a 1 is written to the SCI SW RESET bit. Thus, after a system reset, you must re-enable the SCI2 by writing a 1 to this bit. This bit must be cleared after a receiver break detect.

SCI SW RESET affects the operating flags of the SCI2. This bit does not affect the configuration bits, nor does it put in the reset values. The flags listed in Table 10–6 are set to the values shown when SCI SW RESET is cleared. The operating flags are frozen until the SCI SW RESET bit is set again.

Table 10–6. Flags Affected by SCI SW RESET

SCI Flag	Designation	Value After SCI SW RESET
TXRDY	TXCTL.7	1
TXEMPTY	TXCTL.6	1
RXWAKE	RXCTL.1	0
PE	RXCTL.2	0
OE	RXCTL.3	0
FE	RXCTL.4	0
BRKDT	RXCTL.5	0
RXRDY	RXCTL.6	0
RX ERROR	RXCTL.7	0

Note:

The SCI SW RESET bit must be cleared before the SCI2 configuration registers can be set up or altered. The application program should set up all configuration registers before it sets the SCI SW RESET bit.

Bit 4 **CLOCK ENABLE.** SCI2 Internal Clock Enable.

This bit enables or disables the SCI2 internal clock. For SCI2 operation, this bit must be written as a 1 when you write to the SCICTL register.

0 = Disables SCI2 internal clock (stops SCI2 operation).
1 = Enables SCI2 internal clock (SCI2 operates).

Bit 3 **TXWAKE.** SCI2 Transmitter Wake-up.

The TXWAKE bit controls the transmit features of the multiprocessor communication modes. This bit is cleared only by system reset. The SCI2 hardware clears this bit, once it has been transferred to wake-up temporary (WUT).

Bit 2 **SLEEP.** SCI2 Sleep.

This bit controls the receive features of the multiprocessor communication modes. You must clear this bit to bring the SCI2 out of sleep mode.

0 = Disables sleep mode.
1 = Enables sleep mode.

Bit 1 **TXENA.** SCI2 Transmit Enable.

Data transmission through the SCITXD pin occurs only when this bit is set. If this bit is reset, the transmission is not halted until all the data previously written to TXBUF has been sent.

0 = Disables SCI2 transmitter.
1 = Enables SCI2 transmitter.

Bit 0 **RXENA.** SCI2 Receive Enable.

When this bit is set, received characters are transferred into RXBUF, and the RXRDY flag is set. When cleared, this bit prevents received characters from being transferred into the receiver buffer (RXBUF), and no receiver interrupts are generated. However, the receiver shift register continues to assemble characters. As a result, if RXENA is set during the reception of a character, the complete character is transferred into RXBUF.

0 = Disables SCI2 receiver.
1 = Enables SCI2 receiver.

10.8.3 Baud Select Registers (BAUD MSB and BAUD LSB)

The BAUD MSB and BAUD LSB registers store the data required to generate the bit rate. The asynchronous bit rates are calculated as follows:

Asynchronous Baud = SYSCLK/ [(BAUD REG + 1) × 32]

Baud Select Register (BAUD MSB)
[Memory Address 1052h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P052	BAUDF (MSB)	BAUDE	BAUDD	BAUDC	BAUDB	BAUDA	BAUD9	BAUD8
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Baud Select Register (BAUD LSB)
[Memory Address 1053h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P053	BAUD7	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0 (LSB)
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

10.8.4 SCI Transmitter Interrupt Control and Status Register (TXCTL)

The TXCTL register contains the transmitter interrupt enable bit, the transmitter ready flag, and the transmitter empty flag. The status flags are updated each time a compete character is transmitted.

SCI Transmitter Interrupt Control and Status Register (TXCTL)
[Memory Address 1054h]

Bit #	7	6	5	4	3	2	1	0
P054	TXRDY	TX EMPTY	—	—	—	—	—	SCI TX INT ENA
	R-1	R-1						RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **TXRDY.** SCI2 Transmitter Ready.

The TXRDY bit is set by the transmitter to indicate that TXBUF is ready to receive another character. The bit is automatically cleared when a character is loaded into TXBUF. This flag asserts a transmitter interrupt if the interrupt enable bit SCI TX INT ENA (TXCTL.0) is set. TXRDY is a read-only flag. It is set to 1 by an SCI SW RESET or by a system reset.

0 = TXBUF is full.
1 = TXBUF is ready to receive a character.

Bit 6 **TX EMPTY.** SCI2 Transmitter Empty.

This bit indicates the status of the transmitter-shift register and the TXBUF register. TX EMPTY is set to 1 by an SCI SW RESET or by a system reset.

0 = The CPU has written data to the TXBUF register; the data has not been completely transmitted.
1 = TXBUF and TXSHF registers are empty.

Bits 5–1 **Reserved.** Read data is indeterminate.

Bit 0 **SCI TX INT ENA.** SCI2 Transmitter Ready Interrupt Enable.

This bit controls the ability of the TXRDY bit to request an interrupt but does not prevent the TXRDY bit from being set. The SCI TX INT ENA bit is set to 0 by a system reset.

0 = Disables SCI TXRDY interrupt.
1 = Enables SCI TXRDY interrupt.

10.8.5 SCI Receiver Interrupt Control and Status Register (RXCTL)

The RXCTL register contains one interrupt enable bit and seven receiver status flags (two of which can generate interrupt requests). The status flags are updated each time a complete character is transferred to the RXBUF. They are cleared each time RXBUF is read.

SCI Receiver Interrupt Control and Status Register (RXCTL)
[Memory Address 1055h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P055	RX ERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	SCI RX INT ENA
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **RX ERROR.** SCI2 Receiver Error Flag.

The RX ERROR flag indicates that one of the error flags in the receiver status register is set. It is a logical OR of the parity, overrun, framing error, and break detect flags. The bit can be used for fast error condition checking during the interrupt service routine because a negative value of the status register indicates that an error condition has occurred. This error flag cannot be cleared directly but is cleared if no individual error flags are set. This bit is cleared by an SCI SW RESET, by a system reset, or by reading RXBUF.

Bit 6 **RXRDY.** SCI2 Receiver Ready.

The receiver sets this bit to indicate that RXBUF is ready with a new character and clears the bit when the character is read. A receiver interrupt is generated if the SCI RX INT ENA bit is a 1. RXRDY is reset by an SCI SW RESET or by a system reset.

Bit 5 **BRKDT.** SCI2 Break Detect Flag.

The SCI2 sets this bit when a break condition occurs. A break condition occurs when the SCIRXD line remains continuously low for at least 10 bits, beginning after a missing first stop bit. The occurrence of a break causes a receiver interrupt to be generated if the SCI RX INT ENA bit is a 1, but it does not cause the receiver buffer to be loaded. A BRKDT interrupt can occur, even if the receiver SLEEP bit is set to 1.

BRKDT is cleared by an SCI SW RESET or by a system reset. It is not cleared by receipt of a character after the break is detected. In order to receive more characters, the SCI2 must by reset through toggling the SCI SW RESET bit or by a system reset.

0 = No parity error or parity is disabled.
1 = Parity error detected.

Bit 4 **FE.** SCI2 Framing Error Flag.

The SCI2 sets this bit when it doesn't find a stop bit that it expects. Only the first stop bit is checked. The missing stop bit indicates that synchronization with the start bit has been lost and that the character is incorrectly framed. It is reset by an SCI SW RESET, by a system reset, or by reading RXBUF.

0 = No framing error detected.
1 = Framing error detected.

Bit 3 **OE.** SCI2 Overrun Error Flag.

The SCI2 sets this bit when a character is transferred into RXBUF before the previous character has been read out. The previous character is overwritten and lost. The OE flag is reset by an SCI SW RESET, by a system reset, or by reading RXBUF.

0 = No Overrun error detected.
1 = Overrun error detected.

Bit 2 **PE.** SCI2 Parity Error Flag.

This flag bit is set when a character is received with a mismatch between the number of 1s and its parity bit. The parity checker includes the address bit in the calculation. If parity generation and detection are not enabled, the PE flag is disabled and read as 0. The PE bit is reset by an SCI SW RESET, by a system reset, or by reading RXBUF.

Bit 1 **RXWAKE.** Receiver Wake-Up Detect.

The SCI2 sets this bit when a receiver wake-up condition is detected. In the address bit multiprocessor mode, RXWAKE reflects the value of the address bit for the character contained in RXBUF. In the idle line multiprocessor mode, RXWAKE is set if an idle SCIRXD line is detected. RXWAKE is a read-only flag. It is cleared by transfer of the first byte after the address byte to RXBUF, by reading the address character in RXBUF, by an SCI SW RESET, or by a system reset.

Bit 0 **SCI RX INT ENA.** SCI2 Receiver Interrupt Enable.

The SCI RX INT ENA bit controls the ability of the RXRDY and the BRKDT bits to request an interrupt but does not prevent these flags from being set.

0 = Disables RXRDY/BRKDT interrupt.
1 = Enables RXRDY/BRKDT interrupt.

10.8.6 SCI Receiver Data Buffer Register (RXBUF)

The RXBUF register contains current data from the receiver shift register. RXBUF is cleared by a system reset.

SCI Receiver Data Buffer Register (RXBUF)
[Memory Address 1057h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P057	RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

10.8.7 SCI Transmitter Data Buffer Register (TXBUF)

The TXBUF register is a read/write register that stores data bits to be transmitted by SCITX. Data written to TXBUF must be right-justified because the left-most bits are ignored for characters less than eight bits long.

SCI Transmit Data Buffer Register (TXBUF)
[Memory Address 1059h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P059	TXDT7	TXDT6	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

10.8.8 SCI Port Control Register 2 (SCIPC2)

The SCIPC2 register controls the SCIRXD and SCITXD pin functions.

SCI Port Control Register 2 (SCIPC2)
[Memory Address 105Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P05E	SCITXD DATA IN	SCITXD DATA OUT	SCITXD FUNCTION	SCITXD DATA DIR	SCIRXD DATA IN	SCIRXD DATA OUT	SCIRXD FUNCTION	SCIRXD DATA DIR
	R-0	RW-0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **SCITXD DATA IN.**

This bit contains the current value on the SCITXD pin.

Bit 6 **SCITXD DATA OUT.**

This bit contains the data to be output on the SCITXD pin if the following conditions are met:

- ❏ SCITXD pin has been defined as a general-purpose I/O pin.
- ❏ SCITXD pin data direction has been defined as output.

Bit 5 **SCITXD FUNCTION.**

This bit defines the function of the SCITXD pin.

0 = SCITXD pin is a general-purpose digital I/O pin.
1 = SCITXD pin is the SCI transmit pin.

Bit 4 **SCITXD DATA DIR.** SCITXD Data Direction.

This bit determines the data direction on the SCITXD pin if SCITXD has been defined as a general-purpose I/O pin.

0 = SCITXD pin is a general-purpose input pin.
1 = SCITXD pin is a general-purpose output pin.

Bit 3 **SCIRXD DATA IN.**

This bit contains the current value on the SCIRXD pin.

Bit 2 **SCIRXD DATA OUT.**

This bit contains the data to be output on the SCIRXD pin if the following conditions are met:

- ❏ SCIRXD pin has been defined as a general-purpose I/O pin.
- ❏ SCIRXD pin data direction has been defined as output.

Bit 1 **SCIRXD FUNCTION.**

This bit defines the function of the SCIRXD pin.

0 = SCIRXD pin is a general-purpose digital I/O pin.
1 = SCIRXD pin is the SCI receiver pin.

Bit 0 **SCIRXD DATA DIR.** SCIRXD Data Direction.

This bit determines the data direction on the SCIRXD pin if SCIRXD has been defined as a general-purpose I/O pin.

0 = Pin SCIRXD is a general-purpose input pin.
1 = Pin SCIRXD is a general-purpose output pin.

10.8.9 SCI Priority Control Register (SCIPRI)

The SCIPRI register contains the receiver and transmitter interrupt priority select bits. This register is read-only during normal operation but can be written to in the privilege mode.

SCI Priority Control Register (SCIPRI)
[Memory Address 105Fh]

Bit #	7	6	5	4	3	2	1	0
P05F	SCI STEST	SCITX PRIORITY	SCIRX PRIORITY	SCI ESPEN	—	—	—	—
	RP-0	RP-0	RP-0	RP-0				

R = Read, W = Privilege write only, -*n* = Value of the bit after the register is reset

Bit 7 **SCI STEST.** SCI2 STEST.

This bit must be cleared to ensure proper operation.

Bit 6 **SCI TX PRIORITY.** SCI2 Transmitter Interrupt Priority Select.

This bit assigns the interrupt priority level of the SCI2 transmitter interrupts.

0 = Transmitter interrupts are level 1 (high-priority) requests.
1 = Transmitter interrupts are level 2 (low-priority) requests.

Bit 5 **SCI RX PRIORITY.** SCI2 Receiver Interrupt Priority Select.

This bit assigns the interrupt priority level of the SCI2 receiver interrupts.

0 = Receiver interrupts are level 1 (high-priority) requests.
1 = Receiver interrupts are level 2 (low-priority) requests.

Bit 4 **SCI ESPEN.** SCI2 Emulator Suspend Enable.

This bit has no effect except when you are using the XDS emulator to debug a program. Then, this bit determines how the SCI2 operates when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the SCI2 continues to work until the current transmit or receive sequence is complete.
1 = When the emulator is suspended, the SCI2 state machine is frozen so that the state of the SCI2 can be examined at the point that the emulator was suspended.

Bits 3–0 **Reserved.** Read data is indeterminate.

Chapter 11

Serial Peripheral Interface (SPI) Module

This chapter discusses the architecture and programming of the serial peripheral interface module and covers the following topics:

Topic **Page**

11.1 SPI Overview

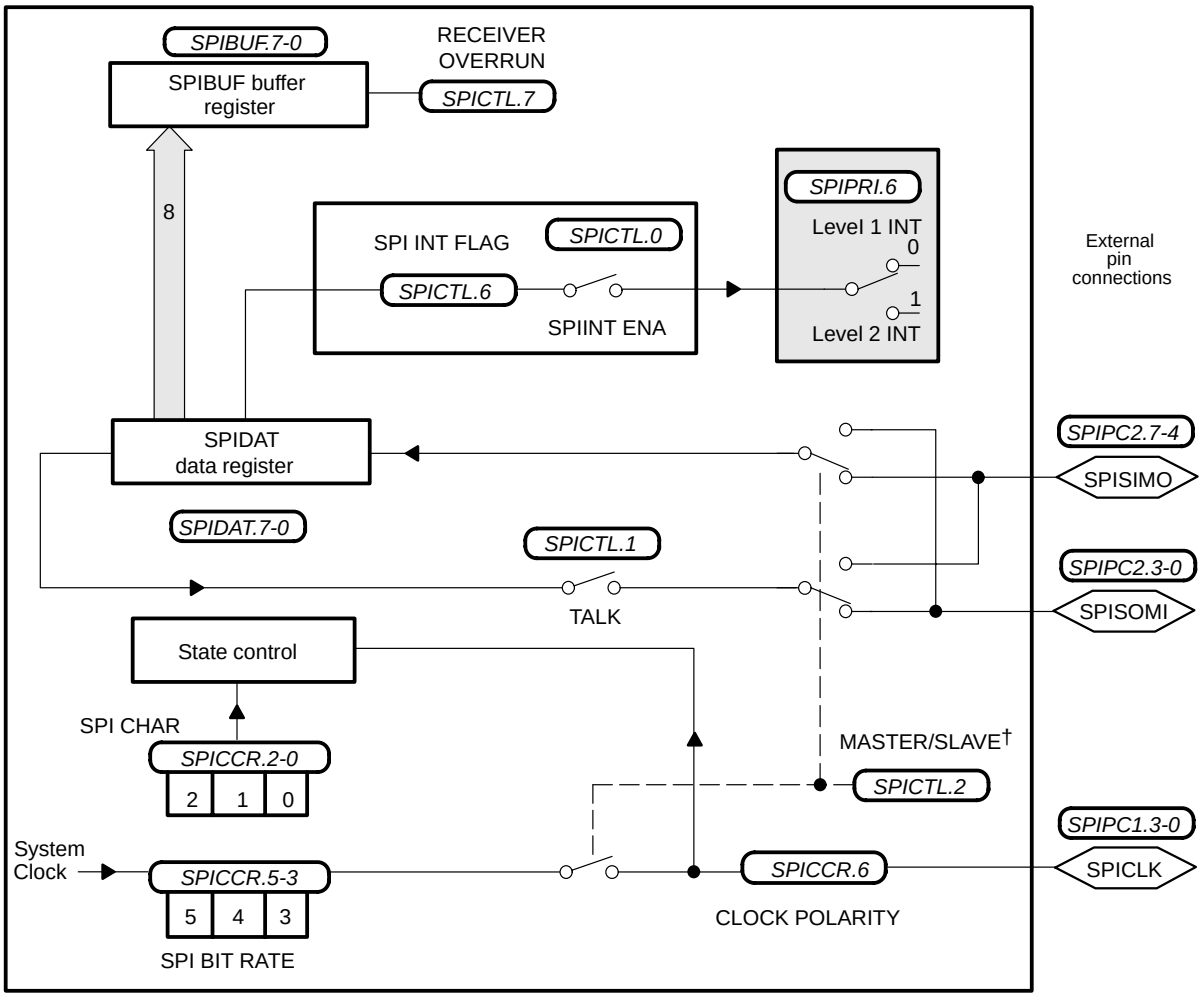
The SPI module is a high-speed synchronous serial I/O port that allows a serial bit stream of programmed length (one to eight bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the microcontroller and external peripherals or another microcontroller. Typical applications include external I/O or peripheral expansion using devices such as shift registers, display drivers, and A/D converters. Multiprocessor communications are also supported by the master/slave operation of the SPI.

11.1.1 Physical Description

The SPI module, as shown in Figure 11–1, consists of:

- Three I/O pins:
  - SPISIMO—SPI slave in, master out
  - SPISOMI—SPI slave out, master in
  - SPICLK—SPI clock
- SPIBUF—the buffer register that contains the data received from the network that is ready for the CPU to read
- SPIDAT—the data shift register that serves as the transmit/receive shift register
- State control logic
- Memory-mapped control and status registers

Figure 11–1. SPI Block Diagram†



† The diagram is shown in slave mode.

11.1.2 Control Registers

The SPI control registers are located at addresses 1030h to 103Fh and occupy peripheral file frame 3. The function of each location is shown in Table 11–1.

Table 11–1. SPI Memory Map

Peripheral File Location	Symbol	Name	Description
P030	SPICCR	SPI Configuration Control Register	Controls the setup of the SPI for operation.
P031	SPICTL	SPI Operation Control Register	Controls data transmission, the SPI's ability to generate interrupts, and the operating mode (slave or master).
P032–P036		Reserved	
P037	SPIBUF	Serial Input Buffer	Contains the data received from the network that is ready for the CPU to read.
P038		Reserved	
P039	SPIDAT	Serial Data Register	Serves as the transmit/receive shift register.
P03A–P03C		Reserved	
P03D	SPIPC1	SPI Port Control Register 1	Controls the SPICLK pin functions.
P03E	SPIPC2	SPI Port Control Register 2	Controls the SPISOMI and SPISIMO pin functions.
P03F	SPIPRI	SPI Interrupt Priority Control Register	Selects the interrupt priority level of the SPI interrupt.

11.2 Communications Between the Master and the Slave

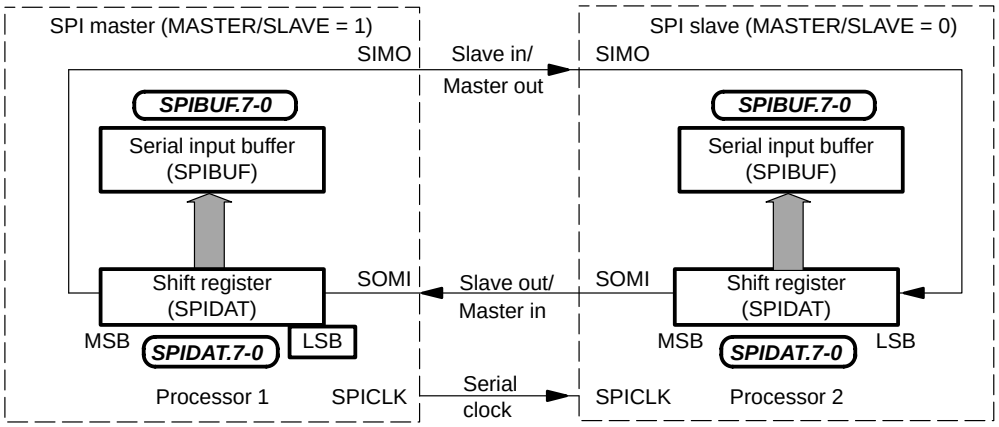
Figure 11–2 shows a typical connection of the SPI for communications between two microcontrollers: the master and the slave. The master initiates data transfer by sending the SPICLK signal. For both the slave and the master, data is shifted out of the shift registers on one edge of the clock and latched into the shift register on the opposite clock edge. As a result, both controllers send and receive data at the same time. The application software determines whether the data is meaningful or dummy data.

There are three possible cases for data transmission:

- Master sends data, and slave sends dummy data
- Master sends data, and slave sends data
- Master sends dummy data, and slave sends data

The master can initiate data transfer at any time because it controls the SPICLK. However, the software protocol determines how the master detects when the slave is ready to broadcast data.

Figure 11–2. SPI Master/Slave Connection



Note: SIMO = Slave In, Master Out; SOMI = Slave Out, Master In

11.3 Operating Modes

The MASTER/SLAVE bit (SPICTL.2) selects the operating mode and the source of SPICLK. The SPI module can operate as a master or a slave.

11.3.1 Master Mode

In the master mode (MASTER/SLAVE = 1), the SPI provides the serial clock on the SPICLK pin for the entire serial communications network. Data is output on the SPISIMO pin on the first SPICLK edge and latched from the SPISOMI pin on the opposite edge of SPICLK.

The SPI BIT RATE0–2 bits of the SPICCR register determine the bit transfer rate for the network, both transmit and receive. Eight data transfer rates can be selected by these control bits as shown in Table 11–2 on page 11-10.

Data written to the SPIDAT register initiates data transmission on the SPISIMO pin, MSB first. Simultaneously, received data is shifted in the SPISOMI pin into the SPIDAT register. When the selected number of bits have been transmitted, the data is transferred to the SPIBUF (double-buffered receiver) for reading by the CPU to permit new transactions to take place. Data is shifted into the SPI MSB first. It is stored right-justified in SPIBUF.

To initiate a character transaction when the SPI is operating as a master, data must be written to the SPIDAT. When the specified number of data bits have been shifted through the SPIDAT register, the following events occur:

- The SPI INT FLAG bit (SPICTL.6) is set,
- The SPIDAT register contents transfer to the SPIBUF register, and
- If the SPI INT ENA bit (SPICTL.1) is set to 1, an interrupt is asserted.

Writing to the SPIDAT register before transmission is complete corrupts the current transmission.

11.3.2 Slave Mode

In the slave mode (MASTER/SLAVE = 0), data shifts out on the SPISOMI pin and in on the SPISIMO pin. The SPICLK pin is used as the input for the serial shift clock, which is supplied from the external network master. The transfer rate is defined by this clock. The SPICLK input frequency should be no greater than the SYSCLK frequency divided by 8.

Data written to the SPIDAT register is transmitted to the network when the SPICLK is received from the network master. To receive data, the SPI waits for the network master to send SPICLK and then shifts the data on the SPISIMO pin into the SPIDAT register. If data is to be transmitted by the slave simultaneously, it must be written to the SPIDAT register before the beginning of SPICLK.

When the TALK bit (SPICTL.1) is cleared, data transmission is disabled, and the output line is put into a high-impedance state. This allows many slave devices to be tied together on the network, but only one slave at a time is allowed to talk.

11.4 Data Format

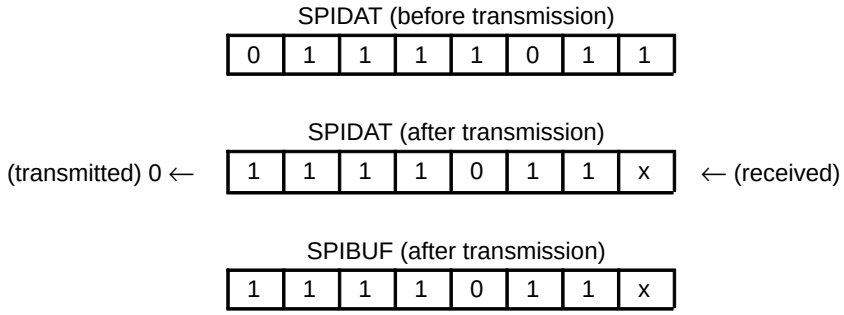
Three character-length bits (SPICCR.2–0) specify the number of bits in the data character (1–8 bits). This information directs the state control logic to count the number of bits received or transmitted to determine when a complete character has been processed.

For characters with fewer than 8 bits:

- Data must be written left-justified to the SPIDAT register.
- Data must be read back right-justified from the SPIBUF register.
- The SPIBUF register contains the most recently received character, right-justified, plus any bits that remain from previous transmission(s) and have been shifted to the MSB position.

Example 11–1. Register Values in an SPI Character-Write Operation

For example:
If the character length = 1 bit, and
the value written into SPIDAT = 07Bh,
then:



Note: x = 1 if SOMI is held high; x = 0 if SOMI is held low

11.5 Interrupts

The interrupt for the SPI is controlled by bits in two registers:

- The SPI INT ENA bit (SPICTL.0), when set, allows assertion of an interrupt request when an interrupt condition occurs.
- The SPI PRIORITY bit (SPIPRI.6) determines whether SPI interrupts are level 1 or level 2 priority requests.

When a complete character has been shifted into or out of the SPIBUF register, the SPI interrupt flag (SPI INT FLAG) of the SCICTL register is set, and an interrupt is generated if enabled by SPI INT ENA bit (SPICTL.0). The interrupt flag remains set until it is cleared by one of the following four events:

- The CPU reads the SPI receiver buffer (SPIBUF),
- The CPU enters the halt or standby mode with an IDLE instruction,
- Software sets the SPI SW RESET bit (SPICCR.7), or
- A system reset occurs.

An interrupt request must be explicitly cleared by one of the four methods listed above to avoid generating another interrupt. An interrupt request can be temporarily disabled by clearing the SPI INT ENA bit. However, unless the SPI INT FLAG itself is cleared, the interrupt request will be reasserted when the enable bit (SPI INT ENA) is again set to 1.

The priority level of the SPI interrupt is specified by the SPI PRIORITY bit (SPIPRI.6).

- If SPI PRIORITY = 0, a level 1 priority interrupt is generated.
- If SPI PRIORITY = 1, a level 2 priority interrupt is generated.

When the SPI INT FLAG bit is set, a character has been placed into the SPIBUF register and is ready to be read. If the CPU does not read the character by the time the next complete character has been received, the new character is written into the SPIBUF, and the RECEIVER OVERRUN bit (SPICTL.7) is set. This indicates that the last character of data has been overwritten with new data before the previous character could be read.

11.6 Clock Sources

The CLOCK POLARITY bit (SPICCR.6) selects the active edge of the clock, either rising or falling.

- In the slave mode, the SPI clock is received from an external source and can be no greater than the SYSCLK frequency divided by 8.
- In the master mode, the SPI clock is generated by the SPI and is output on the SPICLK pin.

The SPI BIT RATE0–2 bits (SPICCR.5–3) determine the bit transfer rate for sending and receiving the data. This transfer rate is defined by:

$$\text{SPI BAUD RATE} = \text{SYSCLK} / (2 \times 2^{b})$$

where b = bit rate in SPICCR.5–3 (range 0–7).

Table 11–2 shows the bit rates for common crystal frequencies versus the SPI bit rate values.

Table 11–2. Common SPI Bit Rates

SPI Value	Divide by	SYSCLK Frequency (MHz)					Bit Rate‡
		0.5†	1.25†	2.5	3	5	
0	2	250	625	1250	1500	2500	kbps
1	4	125	312.5	625	750	1250	kbps
2	8	62.5	156.25	312.5	375	625	kbps
3	16	31.25	78.125	156.25	187.5	312.5	kbps
4	32	15.625	39.0625	78.125	93.75	156.25	kbps
5	64	7.8125	19.53125	39.0625	46.875	78.125	kbps
6	128	3.90625	9.765625	19.53125	23.4375	39.0625	kbps
7	256	1.953125	4.882813	9.765625	11.71875	19.53125	kbps

† Divide-by-1 clock can only operate from a minimum of 2 MHz SYSCLK to a maximum of 5 MHz SYSCLK.
‡ kbps = kilobits/second.

11.7 Initialization Upon Reset

A system reset forces the SPI peripheral module into the following default configuration:

- The unit is configured as a slave module (MASTER/SLAVE = 0).
- The transmit capability is disabled (TALK = 0).
- Data is latched at the input on the falling edge of SPICLK.
- Character length is assumed to be 1 bit.
- The SPI interrupts are disabled.
- Data in the SPI data register is set to 00h.

To change this SPI configuration:

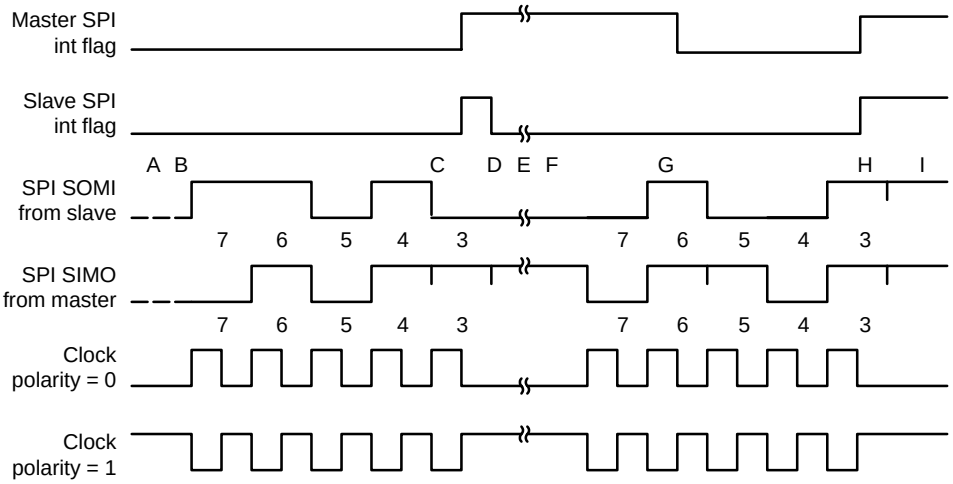
- Set the SPI SW RESET bit (SPICCR.7) to 1,
- Make any changes that you want in the configuration described above, *and*
- Clear the SPI SW RESET bit.

Using the above procedure prevents unwanted and unforeseen events from occurring during or as a result of a mode change.

11.8 SPI Example

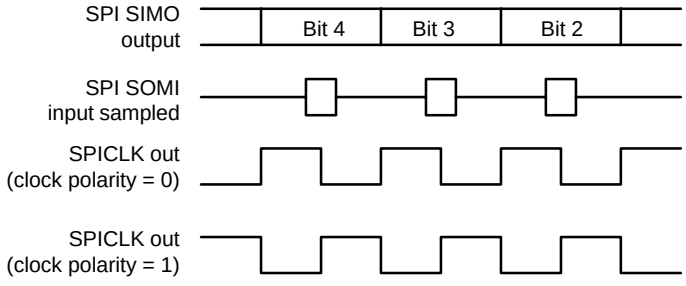
The following timing diagrams illustrate an SPI data transfer between two TMS370 devices using a character length of five bits. The lettered notes following the first diagram are keyed to the letter labels in the diagram.

5 Bits per Character



A. Slave writes 0D0h to SPIDAT and waits for the master to shift out the data.
B. Master writes 058h to SPIDAT, which starts the transmission procedure.
C. First byte is finished and sets the interrupt flags.
D. Slave reads 0Bh from its SPIBUF register (right justified).
E. Slave writes 04Ch to SPIDAT and waits for the master to shift out the data.
F. Master writes 06Ch to SPIDAT, which starts the transmission procedure.
G. Master reads 01Ah from the SPIBUF register (right justified).
H. Second byte is finished and sets the interrupt flags.
I. Master receives 89h, and the slave receives a 8Dh (right justified). Caution should be taken because the SPIBUF register not only contains the five new data (right justified) received, but also the three LSBs of data left over from the previous SPIBUF register (value written to SPInDAT). The user's software routine should mask-off the unused bits.

Signals Connecting to Master Processor



11.9 SPI Control Registers

The SPI is controlled and accessed through registers in peripheral file frame 3. These registers are listed in Figure 11–3 and described in the following subsections. The bits shown in shaded boxes in Figure 11–3 are privilege mode bits; that is, they can be written to only in the privilege mode.

Figure 11–3. Peripheral File Frame 3: SPI Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPICCR	1030h	P030	SPI SW RESET (RW–0)	CLOCK POLARITY (RW–0)	SPI BIT RATE2 (RW–0)	SPI BIT RATE1 (RW–0)	SPI BIT RATE0 (RW–0)	SPI CHAR2 (RW–0)	SPI CHAR1 (RW–0)	SPI CHAR0 (RW–0)
SPICTL	1031h	P031	RECEIVER OVERRUN (R–0)	SPI INT-FLAG (R–0)	—	—	—	MASTER/ SLAVE (RW–0)	TALK (RW–0)	SPI INT ENA (RW–0)
	1032h to 1036h	P032 to P036	Reserved							
SPIBUF	1037h	P037	RCVD7 (R–0)	RCVD6 (R–0)	RCVD5 (R–0)	RCVD4 (R–0)	RCVD3 (R–0)	RCVD2 (R–0)	RCVD1 (R–0)	RCVD0 (R–0)
	1038h	P038	Reserved							
SPIDAT	1039h	P039	SDAT7 (RW–0)	SDAT6 (RW–0)	SDAT5 (RW–0)	SDAT4 (RW–0)	SDAT3 (RW–0)	SDAT2 (RW–0)	SDAT1 (RW–0)	SDAT0 (RW–0)
	103Ah to 103Ch	P03A to P03C	Reserved							
SPIPC1	103Dh	P03D	—	—	—	—	SPICLK DATA IN (R–0)	SPICLK DATA OUT (RW–0)	SPICLK FUNCTION (RW–0)	SPICLK DATA DIR (RW–0)
SPIPC2	103Eh	P03E	SPISIMO DATA IN (R–0)	SPISIMO DATA OUT (RW–0)	SPISIMO FUNCTION (RW–0)	SPISIMO DATA DIR (RW–0)	SPISOMI DATA IN (R–0)	SPISOMI DATA OUT (RW–0)	SPISOMI FUNCTION (RW–0)	SPISOMI DATA DIR (RW–0)
SPIPRI	103Fh	P03F	SPI STEST (RP–0)	SPI PRIORITY (RP–0)	SPI ESPEN (RP–0)	—	—	—	—	—

11.9.1 SPI Configuration Control Register (SPICCR)

The SPICCR register controls the setup of the SPI for operation.

SPI Configuration Control Register (SPICCR)
[Memory Address 1030h]

Bit #	7	6	5	4	3	2	1	0
P030	SPI SW RESET	CLOCK POLARITY	SPI BIT RATE2	SPI BIT RATE1	SPI BIT RATE0	SPI CHAR2	SPI CHAR1	SPI CHAR0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **SPI SW RESET**. SPI Software Reset.

- Writing a 1 to this bit initializes the SPI circuitry and operating flags to the reset condition. Specifically, the RECEIVER OVERRUN and SPI INT FLAG flags are cleared. The SPI configuration remains unchanged. If the module is operating as a master, the SPICLK output level returns to its inactive level.
- When a 0 is written to SPI SW RESET, the SPI is ready to transmit or receive the next character. A character written to the transmitter when SPI SW RESET is a 1 will not be shifted out when the SPI SW RESET bit is cleared. A new character must be written to the serial data register. Use this bit to change any configuration bits (see Section 11.7).

Bit 6 **CLOCK POLARITY**. Shift Clock Polarity.

The CLOCK POLARITY bit controls the polarity of the SPICLK signal.

0 = The inactive level is low; data is output by the rising edge of SPICLK; input data is latched by the falling edge of SPICLK.
1 = The inactive level is high; data is output by the falling edge of SPICLK; input data is latched by the rising edge of SPICLK.

Bits 5–3 **SPI BIT RATE2–0.** SPI Bit Rate Control Bits 2–0.

These bits determine the bit transfer rate if the SPI is the network master. Eight data transfer rates (each a function of the system clock) can be selected. The system clock is divided by an 8-bit, free-running prescaler from which eight taps are available for use as the shift clock. One data bit is shifted per SPICLK cycle.

SPI BIT RATE2†	SPI BIT RATE1†	SPI BIT RATE0†	SPI Clock Frequency
0	0	0	SYSCLK/2
0	0	1	SYSCLK/4
0	1	0	SYSCLK/8
0	1	1	SYSCLK/16
1	0	0	SYSCLK/32
1	0	1	SYSCLK/64
1	1	0	SYSCLK/128
1	1	1	SYSCLK/256

† If the SPI is a network slave, then the module receives a clock on the SPICLK pin from the network master, and these bits have no effect on SPICLK. The frequency of the input clock should be no greater than the SYSCLK frequency divided by 8.

Bits 2–0 **SPI CHAR2–0**. Character Length Control Bits 2–0.

These three bits determine the number of bits to be shifted in or out as a single character during one shift sequence. The value of these bits is represented in the following table.

SPI CHAR2	SPI CHAR1	SPI CHAR0	Character Length
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

11.9.2 SPI Operation Control Register (SPICTL)

The SPI operation control register controls data transmission, the SPI's ability to generate interrupts, and the operating mode (slave or master).

SPI Operation Control Register (SPICTL)
[Memory Address 1031h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P031	RECEIVER OVERRUN	SPI INT FLAG	—	—	—	MASTER/ SLAVE	TALK	SPI INT ENA
	R-0	R-0				RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **RECEIVER OVERRUN.**

This bit is a read-only flag that the SPI hardware sets when a receive or transmit operation completes before the previous character is read from the receive buffer. It indicates that the last received character has been overwritten and, therefore, has been lost. RECEIVER OVERRUN is cleared when the receiver buffer is read. It is also cleared by SPI SW RESET or by a system reset.

Bit 6 **SPI INT FLAG**. Serial Peripheral Interrupt Flag.

The SPI hardware sets this bit to indicate that it has completed sending or receiving the last bit and is ready to be serviced. A character received is placed in the receiver buffer at the time the SPI INT FLAG bit is set. The SPI INT FLAG is cleared when the receiver buffer is read. It is also cleared by an SPI software reset (SPI SW RESET) or by a system reset.

Bits 5–3 **Reserved.** Read data is indeterminate.

Bit 2 **MASTER/SLAVE**. SPI Network Mode Control.

This bit determines whether the SPI is a network master or slave. During reset initialization, the SPI is automatically configured as a slave.

0 = SPI configured as a slave
1 = SPI configured as a master

Bit 1 **TALK**. Master/Slave Transmit Enable.

This bit disables data transmission (master or slave) by placing the serial data output in a high-impedance state. TALK is cleared (disabled) by a system reset.

0 = Disables transmission; if not programmed as a general-purpose I/O pin, the SPI serial output is in a high-impedance state.
1 = Enables transmission

Bit 0 **SPI INT ENA**. SPI Interrupt Enable.

This bit controls the SPI's ability to generate an interrupt. The SPI INT FLAG is unaffected by this bit.

0 = Disables interrupt
1 = Enables interrupt

11.9.3 Serial Input Buffer (SPIBUF)

The SPIBUF register contains the data received from the network ready for the CPU to read.

Serial Input Buffer (SPIBUF)
[Memory Address 1037h]

Bit #	7	6	5	4	3	2	1	0
P037	RCVD7 (MSB)	RCVD6	RCVD5	RCVD4	RCVD3	RCVD2	RCVD1	RCVD0 (LSB)
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

Once the serial data register has received the complete character, the character is then transferred to the SPIBUF register, where it can be read. The SPI INT FLAG bit (SPICTL.6) is set to indicate that the data is available when the received character is transferred. Since data is shifted into the SPI most significant bit first, it is stored, right-justified, in the SPIBUF while the previous LSB(s) data occupies the remaining MSB(s). The user's software routine should mask off the unused bits.

11.9.4 Serial Data Register (SPIDAT)

The SPIDAT register is the transmit/receive shift register. Data written to the SPIDAT is shifted out on subsequent SPICLK cycles. For every bit shifted out of the SPI, a bit is shifted into the other end of the shift register.

Serial Data Register (SPIDAT)
[Memory Address 1039h]

Bit #	7	6	5	4	3	2	1	0
P039	SDAT7	SDAT6	SDAT5	SDAT4	SDAT3	SDAT2	SDAT1	SDAT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Writing to the SPIDAT performs two functions:

- It provides data to be output on the serial output pin if the TALK bit is set.
- It initiates a transaction when the SPI is operating as a master.

A receiver sequence is initiated when dummy data is written to the register. Since the data is not hardware justified for characters that are shorter than eight bits, transmit data must be written in left-justified form and received data read in right-justified form.

11.9.5 SPI Port Control Registers (SPIPC1 and SPIPC2)

Two port control registers (SPIPC1 and SPIPC2) allow you to control all of the functions for a SPI port pin in one write cycle. Each module pin is controlled by a nibble in one of the SPIPCs.

11.9.5.1 SPI Port Control Register 1 (SPIPC1)

This register controls the SPICLK pin.

Port Control Register 1 (SPIPC1)
[Memory Address 103Dh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P03D	—	—	—	—	SPICLK DATA IN	SPICLK DATA OUT	SPICLK FUNCTION	SPICLK DATA DIR
					R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bits 7–4 **Reserved.** Read data is indeterminate.

> **Note: SPICLK Pin in Slave Mode**
>
> The SPICLK pin always functions as the SPICLK input pin in the slave mode (i.e., SPICTL.2 = 0), even if SPICLK FUNCTION = 0.

Bit 3 **SPICLK DATA IN**. SPICLK Pin Port Data In.

This bit contains the current value on the SPICLK pin, regardless of the mode. A write to this bit has no effect.

Bit 2 **SPICLK DATA OUT**. SPICLK Port Data Out.

This bit contains the data to be output on the SPICLK pin if both the following conditions are met:

- SPICLK pin has been defined as a general-purpose I/O pin.
- SPICLK pin data direction has been defined as output.

Bit 1 **SPICLK FUNCTION**. SPICLK Pin Function Select.

This bit defines the function of the SPICLK pin.

0 = SPICLK pin is a general-purpose digital I/O pin.
1 = SPICLK pin contains the SPI clock.

Bit 0 **SPICLK DATA DIR.** SPICLK Data Direction.

This bit determines the data direction on the SPICLK pin if SPICLK has been defined as a general-purpose I/O pin.

0 = SPICLK pin is a general-purpose input pin.
1 = SPICLK pin is a general-purpose output pin.

11.9.5.2 SPI Port Control Register 2 (SPIPC2)

The SPIPC2 register controls the SPISOMI and SPISIMO pin functions.

Port Control Register 2 (SPIPC2)
[Memory Address 103Eh]

Bit #	7	6	5	4	3	2	1	0
P03E	SPISIMO DATA IN	SPISIMO DATA OUT	SPISIMO FUNCTION	SPISIMO DATA DIR	SPISOMI DATA IN	SPISOMI DATA OUT	SPISOMI FUNCTION	SPISOMI DATA DIR
	R-0	RW–0	RW-0	RW-0	R-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **SPISIMO DATA IN**. SPISIMO Pin Data In.

This bit contains the current value on the SPISIMO pin, regardless of the mode. A write to this bit has no effect.

Bit 6 **SPISIMO DATA OUT**. SPISIMO Pin Data Out.

This bit contains the data to be output on the SPISIMO pin if both the following conditions are met:

- ❑ SPISIMO pin has been defined as a general-purpose I/O pin.
- ❑ SPISIMO pin data direction has been defined as output.

Bit 5 **SPISIMO FUNCTION**. SPISIMO Pin Function Select.

This bit defines the function of the SPISIMO pin.

0 = SPISIMO pin is a general-purpose digital I/O pin.
1 = SPISIMO pin contains the SPI data.

Bit 4 **SPISIMO DATA DIR**. SPISIMO Data Direction.

This bit determines the data direction on the SPISIMO pin if SPISIMO has been defined as a general-purpose I/O pin.

0 = SPISIMO pin is a general-purpose input pin.
1 = SPISIMO pin is a general-purpose output pin.

Bit 3 **SPISOMI DATA IN**. SPISOMI Pin Data In.

This bit contains the current value on the SPISOMI pin, regardless of the mode. A write to this bit has no effect.

Bit 2 **SPISOMI DATA OUT**. SPISOMI Pin Data Out.

This bit contains the data to be output on the SPISOMI pin if both the following conditions are met:

- ❑ The SPISOMI pin has been defined as a general-purpose I/O pin.
- ❑ The SPISOMI pin data direction has been defined as output.

Bit 1 **SPISOMI FUNCTION**. SPISOMI Pin Function Select.

This bit defines the function of the SPISOMI pin. When SPISOMI is an input and SPISOMI FUNCTION and SPISOMI DATA DIR are disabled, SPICLK still clocks the internal circuitry.

0 = SPISOMI pin is a general-purpose digital I/O pin.
1 = SPISOMI pin contains the SPI data.

Bit 0 **SPISOMI DATA DIR**. SPISOMI Data Direction.

This bit determines the data direction on the SPISOMI pin if SPISOMI has been defined as a general-purpose I/O pin.

0 = SPISOMI pin is a general-purpose input pin.
1 = SPISOMI pin is a general-purpose output pin.

11.9.6 SPI Interrupt Priority Control Register (SPIPRI)

The SPIPRI register selects the interrupt priority level of the SPI interrupt. The register is read only during normal operation but can be written to in the privilege mode.

SPI Interrupt Priority Control Register (SPIPRI)
[Memory Address 103Fh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P03F	SPI STEST	SPI PRIORITY	SPI ESPEN	—	—	—	—	—
	RP-0	RP-0	RP-0					

R = Read, P = Privilege write only, -*n* = Value of the bit after the register is reset

Bit 7 **SPI STEST.** SPI STEST.

This bit must be cleared to ensure proper operation.

Bit 6 **SPI PRIORITY**. Interrupt Priority Select.

0 = Interrupts are level 1 (high-priority) requests.
1 = Interrupts are level 2 (low-priority) requests.

Bit 5 **SPI ESPEN**. Emulator Suspend Enable.

This bit has no effect, except when you are using the XDS emulator to debug a program; then, this bit determines SPI operation when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the SPI continues to work until the current transmit/receive sequence is complete.
1 = When the emulator is suspended, the the state of the SPI is frozen so that it can be examined at the point that the emulator was suspended.

Bits 4–0 **Reserved.** Read data is indeterminate.

Chapter 12

Analog-To-Digital Converter 1 (ADC1) Module

The TMS370 family contains three different ADC modules (ADC1, ADC2, and ADC3). This chapter discusses the architecture and programming of the analog-to-digital converter 1 (ADC1) module and covers the following topics:

Topic **Page**

12.1 Analog-to-Digital Converter 1 (ADC1) Overview

The analog-to-digital converter 1 module is an 8-bit, successive approximation converter with internal sample-and-hold circuitry. The module has eight multiplexed analog input channels that allow the processor to convert the voltage levels from up to eight different sources. The ADC1 module is available in the following families: TMS370Cx32, TMS370Cx36, TMS370Cx4x†, TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, and TMS370CxBx.

12.1.1 Physical Description

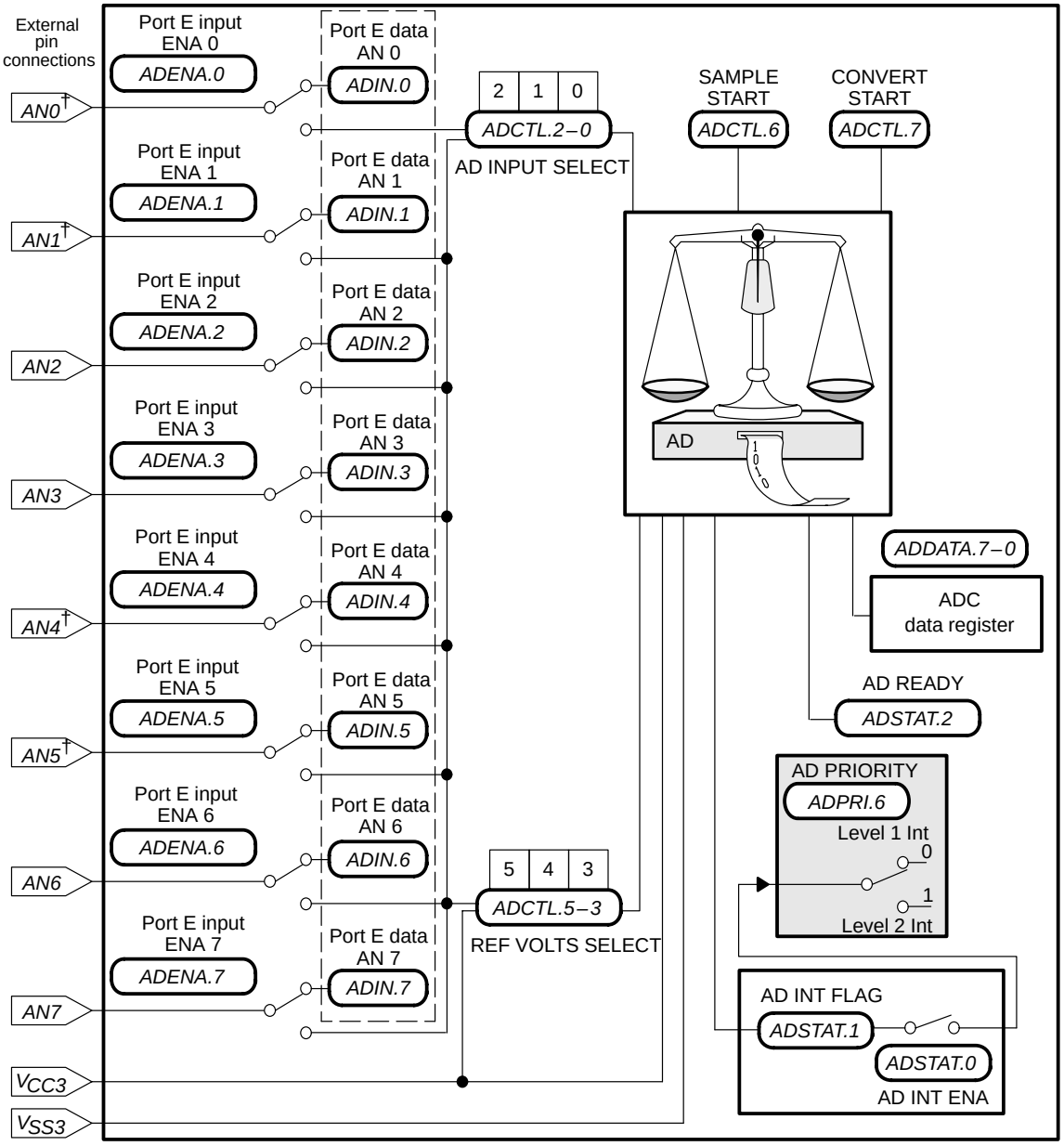
The ADC1 module, shown in Figure 12–1, consists of:

- The ADC1 conversion block
- An ADC1 input selector (INPUT)
  - For all family devices (except the TMS370Cx4x 40-pin device), eight analog input channels (AN0–AN7), any of which can be software-configured as digital inputs (E0–E7) if not needed as analog channels
  - For the TMS370Cx4x 40-pin device, four analog input channels (AN2, AN3, AN6, and AN7), any of which can be software-configured as digital inputs (E2, E3, E6, and E7) if not needed as analog channels
- A $+V_{REF}$ input selector ($+V_{REF}$)
  - For all family devices except the TMS370Cx4x 40-pin device, eight positive input voltage references (AN1-AN7 and V_{CC3}).
  - For the TMS370Cx4x 40-pin device, five positive input voltage references (AN2, AN3, AN6, AN7, and V_{CC3})
- The ADDATA register, which contains the digital value of a completed conversion
- ADC1 module control registers

The input channels can be routed through either the channel selector or the positive voltage selector. The ADC1 converter then processes these signals and puts the result in the ADDATA register. The ADC1 interrupt circuit informs the rest of the system when a conversion is completed.

† For the TMS370Cx4x 40 pin device, the ADC1 module has four multiplexed analog input channels and five positive input voltage references.

Figure 12–1. Analog-to-Digital Converter1 (ADC1) Block Diagram



† These pins are not implemented in the TMS370Cx4x 40-pin device.

12.1.2 Control Registers

The ADC1 control registers are located at addresses 1070h to 107Fh and occupy peripheral file frame 7, as shown in Table 12–1.

Table 12–1. ADC1 Memory Map

Peripheral File Location	Symbol	Name	Description
P070	ADCTL	Analog Control Register	Controls the input selection, reference voltage selection, sample start, and conversion start.
P071	ADSTAT	Analog Status and Interrupt Register	Indicates the converter and interrupt status.
P072	ADDATA	Analog Conversion Data Register	Contains the digital result of the last ADC1 conversion.
P073–P07C		Reserved	
P07D	ADIN	Analog Port E Data Input Register	Contains digital input data when one or more of the AN0–AN7 pins are used as digital ports.
P07E	ADENA	Analog Port E Input Enable Register	Controls the function of the AN0–AN7 pins.
P07F	ADPRI	Analog Interrupt Priority Register	Selects the interrupt priority level of the ADC1 interrupt.

12.2 ADC1 Operation

The following subsections describe the functions and options of the ADC1 module.

12.2.1 Input/Output Pins

The ADC1 module contains up to ten pins.

- The TMS370Cx32, TMS370Cx36, 44-pin TMS370Cx4x, TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, and TMS370CxBx use 10 pins: eight analog channels, $V_{CC3,}$ and $V_{SS3.}$ These pins are described below:
  - Eight (AN0–AN7) of the ten pins are analog channels and can be individually configured as general-purpose input pins (E0–E7) when not used as analog inputs.
  - Seven (AN1–AN7) of the eight analog channels are also available as the positive input voltage reference. This feature allows a weighted measurement or ratio of one channel to another.
  - The analog voltage supply pins, V_{CC3} and V_{SS3}, isolate the ADC1 module from digital switching noise that can be present on the other power supply pins. This isolation provides a more accurate conversion.
- The TMS370Cx4x 40-pin device uses 6 pins: four analog channels, V_{CC3}, and $V_{SS3.}$ These pins are described below:
  - Four (AN2, AN3, AN6, and AN7) of the six pins are analog channels and can be individually configured as general-purpose input pins when not used as analog inputs.
  - Four (AN2, AN3, AN6, and AN7) analog channels are also available as the positive input voltage reference. This feature allows a weighted measurement or ratio of one channel to another.
  - The analog voltage supply pins, V_{CC3} and V_{SS3}, isolate the ADC1 module from digital switching noise that can be present on the other power supply pins. This isolation provides a more accurate conversion.

    To further reduce noise and produce a more accurate conversion, you should run the power to the V_{CC3} and V_{SS3} pins on separate conductors from the other power lines. Additionally, the power conductors to the V_{CC3} and V_{SS3} should be as short as possible, and the two lines should be properly decoupled. Use other standard noise-reduction techniques to help provide a more accurate conversion.

Note that you can select the V_{REF} pin to be either V_{CC3} or one of the analog input channels AN1 to AN7. V_{CC3} must provide power to the ADC1 module even if it does not provide the voltage reference. A channel configured as the $+V_{REF}$ for one conversion can be changed to an analog input channel for the next conversion.

12.2.2 Sampling Time

The application program controls the length of the sample time, which provides the flexibility to optimize the conversion process for both high- and low-impedance sources. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low-impedance sources.

12.2.3 ADC1 Conversion

The digital result of the conversion process is given in the following formula.

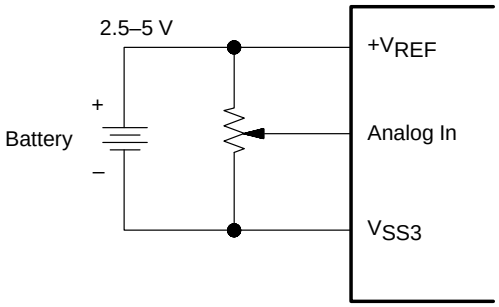
$$\text{digital result} = 255 \times \frac{\text{input voltage}}{\text{reference voltage}}$$

The conversion process requires 164 SYSCLK cycles and results in a conversion time of 32.8 microseconds at 5 MHz SYSCLK. A maximum of 27,600 conversions per second is possible at 5 MHz SYSCLK, including setting up the conversion, sampling, converting, and saving the results.

In ratiometric conversions, the conversion value is a ratio of the V_{REF} source to the analog input. As V_{REF} is increased, the input voltage that is required in order to produce a certain conversion value changes; however, all conversion values keep the same relationship to V_{REF}. That is, one half of V_{REF} always results in the value 080h, regardless of the value of V_{REF} (assuming that V_{REF} is in the range of 2.5 to 5.5 volts above V_{SS3}).

Figure 12–2 shows an example of ratiometric conversion. In this example, the digital result of the conversion indicates the position of the potentiometer wiper, even if the battery loses voltage over time. The ADC1 conversion always gives the ratio of the resistor values on either side of the wiper, even if V_{REF} drops from 5.0 to 2.5 volts.

Figure 12–2. Ratiometric Conversion Example



12.2.4 Interrupts

The ADC1 module sets the AD INT FLAG bit (ADSTAT.1) at the end of the conversion process. If both the AD INT FLAG and the AD INT ENA bit (ADSTAT.0) are set, then the module generates an interrupt request. This interrupt request can be asserted on either high-priority level 1 or the lower-priority level 2, depending on the AD PRIORITY bit (ADPRI.6).

The program must clear the AD INT FLAG bit before exiting the interrupt service routine (ISR), or else the same interrupt will cause the CPU to enter the interrupt routine again. If the AD INT ENA bit is cleared without clearing the flag, the interrupt is reasserted when the AD INT ENA bit is again set.

12.2.5 Programming Considerations

Follow these steps to obtain data from the ADC1:

1) Write to the ADCTL register (described on page 12-12) to:
   a) Select the analog channel (ADCTL.2–0).
   b) Select the V_{REF} source (ADCTL.5–3).
   c) Set the SAMPLE START bit to 1 (ADCTL.6) to begin sampling.
2) Wait for the sample time to elapse. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low impedance sources.
3) When the sample time completes, set the CONVERT START bit (ADCTL.7); leave the SAMPLE START bit (ADCTL.6) set.
4) Wait for either the interrupt flag to be set or the ADC1 interrupt to occur.
5) Read the conversion data register (ADDATA).
6) Clear the interrupt flag bit (ADSTAT.1).

Eighteen SYSCLK cycles after the program sets the CONVERT START bit, the ADC1 module clears both the SAMPLE START and CONVERT START bits to signify the end of the internal sampling phase. After these bits are cleared, the program can change the input channel without affecting the conversion process. The voltage reference source V_{REF} should remain constant throughout the conversion.

To stop a conversion in progress, set the SAMPLE START (ADCTL.6) bit to 1 anytime after the ADC1 clears this bit. The entire conversion process requires 164 SYSCLK cycles after the program sets the CONVERT START bit (ADCTL.7).

12.3 ADC1 Example Program

This example program (next page) samples and converts data from all eight channels and stores the digital results into a table beginning at ATABLE. The routine stops interrupting the main program after it finishes all eight channels. If the main program wants more recent data, it needs to execute only the code at RESTART, and the ADC1 routine will again sample and convert all eight channels of data. The AD INT ENA bit (ADSTAT.0) is cleared by the ADC1 interrupt routine as a signal to the main program that all eight channels have been processed. The address of the label ATOD must be placed into the interrupt vector table located at 7FECh and 7FEDh.

```
ADCTL    .EQU  P070          ;ADC1 control register
ADSTAT   .EQU  P071          ;ADC1 status register
ADDATA   .EQU  P072          ;ADC1 conversion results
ADENA    .EQU  P07E          ;ADC1 input enable
         .REG  ADCHANL       ;keeps current channel number
         .REG  ATABLE,8      ;8 byte table that stores
                             ; channel data, LSB first
;
INIT     MOV   #0,ADENA      ;all channels to ADC1 inputs
                             ; (reset condition)
         CALL  RESTART       ;start taking data
;
;        MAIN PROGRAM GOES HERE
;        .
;        .
         CALL  RESTART       ;start taking more data
;        .
;        .
;        MORE MAIN PROGRAM
;
;        SUBROUTINE SECTION
RESTART  CLR   ADCHANL       ;initialize channel
         MOV   #001h,ADSTAT  ;enable interrupts, clear
                             ; any flag
         MOV   #040h,ADCTL   ;start sampling (approx. 2 µs
                             ; delay)
         MOV   #0C0h,ADCTL   ;start converting now; enter
                             ; main program
         RTS
;
;        INTERRUPT ROUTINE FOR ANALOG TO DIGITAL CONVERTER1
ATOD     PUSH  A             ;save registers
         PUSH  B
         MOV   ADCHANL,B     ;get channel number
         MOV   ADDATA,A      ;get ADC1 conversion value
         MOV   A,*ATABLE[B]  ;store in a table according to
                             ; channel number
         INC   B             ;point to next channel
         BTJZ  #8,B,GOCNVRT
;stop when all channels sampled
                             ;(bit3 =1)
         CLR   ADCHANL       ;reset the ADC1 channel
         MOV   #0,ADSTAT     ;turn off interrupt and
                             ; clear flag
         JMP   EXITA2D       ;all 8 channels taken, enable
                             ;set to 0 now
;
GOCNVRT  MOV   B,ADCHANL     ;store current ADC1 channel
         MOV   #01h,ADSTAT   ;clear interrupt flag
         OR    #040h,B
;set up sample bit in value
         MOV   B,ADCTL       ;start sampling channel data
         OR    #080h,ADCTL   ;start converting data
;
EXITA2D  POP   B             ;Restore data
         POP   A
         RTI
```

12.4 ADC1 Control Registers

The ADC1 module control registers occupy peripheral file frame 7, as shown in Figure 12–3. The bits shown in *shaded boxes* in Figure 12–3 are privilege mode bits; that is, they can be written only to in the privilege mode.

Figure 12–3. Peripheral File Frame 7: ADC1 Converter Control Registers

Designation	ADDR	PF	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	REF VOL SELECT2 (RW–0)	REF VOL SELECT1 (RW–0)	REF VOL SELECT0 (RW–0)	AD INPUT-SELECT2 (RW–0)	AD INPUT-SELECT1 (RW–0)	AD INPUT-SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	Port E Data Input Register (R–0)							
ADENA	107Eh	P07E	Port E Input Enable Register (RW–0)							
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	—	—

12.4.1 Analog Control Register (ADCTL)

The ADCTL register controls the input selection, reference voltage selection, sample start, and conversion start.

Analog Control Register (ADCTL)
[Memory Address 1070h]

Bit #	7	6	5	4	3	2	1	0
P070	CONVERT START	SAMPLE START	REF VOLT SELECT2	REF VOLT SELECT1	REF VOLT SELECT0	AD INPUT SELECT2	AD INPUT SELECT1	AD INPUT SELECT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **CONVERT START.** Conversion Start.

Setting this bit starts the conversion. This bit is cleared by the ADC1 18 system clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any conversion in progress.

Bit 6 **SAMPLE START**. Sample Start.

Setting this bit stops any ongoing conversion and starts sampling the selected input channel to begin a new conversion. This bit is cleared by the ADC1 module 18 system-clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any sampling in progress.

Bits 5–3 **REF VOLT SELECT2-0**. Reference Voltage ($+V_{REF}$) Select Bits 2-0.

These bits select the channel the ADC1 uses for the positive voltage reference. The REF VOLT SELECT bits must not change during the entire conversion.

REF VOLT SELECT2	REF VOLT SELECT1	REF VOLT SELECT0	$+V_{REF}$ Source†
0	0	0	V_{CC3}
0	0	1	AN1‡
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4‡
1	0	1	AN5‡
1	1	0	AN6
1	1	1	AN7

† Pin AN0 cannot be selected as positive voltage reference.
‡ AN0, AN1, AN4, and AN5 are not implemented on the 40-pin TMS370Cx4x device; thus, they cannot be used as positive voltage references.

Bits 2–0 **AD INPUT SELECT2-0**. Analog Input Channel Select Bits 2-0.

These bits select the channel used for conversion. Channels should be changed only after the ADC1 has cleared the SAMPLE START and CONVERT START bits. Changing the channel while either the SAMPLE START bit or the CONVERT START bit is 1 invalidates the conversion in progress.

AD INPUT SELECT2	AD INPUT SELECT1	AD INPUT SELECT0	Channel
0	0	0	AN0†
0	0	1	AN1†
0	1	0	AN2
0	1	1	AN3
1	0	0	AN4†
1	0	1	AN5†
1	1	0	AN6
1	1	1	AN7

† AN0, AN1, AN4, and AN5 are not implemented on the 40-pin TMS370Cx4x device.

12.4.2 Analog Status and Interrupt Register (ADSTAT)

The ADSTAT register indicates the converter and interrupt status.

Analog Status and Interrupt Register (ADSTAT)
[Memory Address 1071h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P071	—	—	—	—	—	AD READY	AD INT FLAG	AD INT ENA
						R-0	RC-0	RW-0

R = Read, W = Write, C = Clear only, -*n* = Value of the bit after the register is reset

Bits 7–3 **Reserved.** Read data is indeterminate.

Bit 2 **AD READY**. ADC1 Converter Ready.

The ADC1 module sets this bit whenever a conversion is not in progress and the ADC1 is ready for a new conversion to start. Writing to this bit has no effect on its state.

0 = Conversion in process.
1 = Converter ready.

Bit 1 **AD INT FLAG**. ADC1 Interrupt Flag.

The ADC1 module sets this bit at the end of an ADC1 conversion. If this bit is set while the AD INT ENA bit is set, an interrupt request is generated. Clearing this flag clears pending ADC1 interrupt requests. This bit is cleared by the system reset. Software cannot set this bit.

Bit 0 **AD INT ENA**. ADC1 Interrupt Enable.

This bit controls the ADC1 module's ability to generate an interrupt.

0 = Disables ADC1 interrupt.
1 = Enables ADC1 interrupt.

12.4.3 Analog Conversion Data Register (ADDATA)

The ADDATA register contains the digital result of the last ADC1 conversion.

Analog Conversion Data Register (ADDATA)
[Memory Address 1072h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P072	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, –*n* = Value of the bit after the register is reset

The analog-to-digital conversion data is loaded into this register at the end of a conversion and remains there until replaced through another conversion.

12.4.4 Analog Port E Data Input Register (ADIN)

The ADIN register contains digital input data when one or more of the AN0 through AN7 pins are used as digital ports.

Analog Port E Data Input Register (ADIN)
[Memory Address 107Dh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07D	PORT E DATA AN 7	PORT E DATA AN 6	PORT E DATA AN 5	PORT E DATA AN 4	PORT E DATA AN 3	PORT E DATA AN 2	PORT E DATA AN 1	PORT E DATA AN 0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

The ADIN register shows the data present at the AN0–AN7 pins when they are configured for general-purpose input instead of for ADC1 channels. A bit is configured as a general-purpose input if the corresponding bit of the port enable register is a 1. Pins configured as ADC1 channels are read as 0s. Writing to this address has no effect.

12.4.5 Analog Port E Input Enable Register (ADENA)

The ADENA register controls the function of the AN0 through AN7 pins.

Analog Port E Data Input Enable Register (ADENA)
[Memory Address 107Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07E	PORT E INPUT ENA 7	PORT E INPUT ENA 6	PORT E INPUT ENA 5	PORT E INPUT ENA 4	PORT E INPUT ENA 3	PORT E INPUT ENA 2	PORT E INPUT ENA 1	PORT E INPUT ENA 0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

The ADENA register individually configures the AN0–AN7 pins as either analog input channels or as general-purpose input pins.

0 = For pins AN1–AN7, the pin becomes an analog input or reference channel for the ADC1; for pin AN0, the pin becomes an analog input channel for the ADC1.
When the bit is 0, the corresponding bit in the ADIN register reads as a 0.

1 = Enables the pin as a general-purpose input pin; its digital value can be read from the corresponding bit in the port E data input register.

12.4.6 Analog Interrupt Priority Register (ADPRI)

The ADPRI register selects the interrupt priority level of the ADC1 interrupt.

Analog Interrupt Priority Register (ADPRI)
[Memory Address 107Fh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07F	AD STEST	AD PRIORITY	AD ESPEN	—	—	—	—	—
	RP-0	RP-0	RP-0					

R = Read, P = Privilege write only, -*n* = Value of the bit after the register is reset

Bit 7 **AD STEST**. This bit must be cleared (0) to ensure proper operation.

Bit 6 **AD PRIORITY**. ADC1 Interrupt Priority Select.

This bit selects the priority level of the ADC1 interrupt.

0 = ADC1 interrupt is a higher priority (level 1) request.
1 = ADC1 interrupt is a lower priority (level 2) request.

Bit 5 **AD ESPEN**. Emulator Suspend Enable.

Normally, this bit has no effect. However, when you are using the XDS emulator to debug a program, this bit determines what happens to the ADC1 when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the ADC1 continues to work until the current conversion is complete.
1 = When the emulator is suspended, the ADC1 is frozen so that its state can be examined at the point that the emulator was suspended. The conversion data is indeterminate upon restart.

Bits 4–0 **Reserved.** Read data is indeterminate.

Chapter 13

Analog-To-Digital Converter 2 (ADC2) Module

The TMS370 family contains three different ADC modules (ADC2, ADC2, and ADC3). This chapter discusses the architecture and programming of the analog-to-digital converter 2 (ADC2) module and covers the following topics:

Topic **Page**

13.1 Analog-to-Digital Converter 2 (ADC2) Overview

The analog-to-digital converter 2 module is an 8-bit, successive approximation converter with internal sample-and-hold circuitry. The module has four multiplexed analog input channels that allow the processor to convert the voltage levels from up to four different sources.

Note: ADC2 Module Availability

The ADC2 module is available for the TMS370CxCx family.

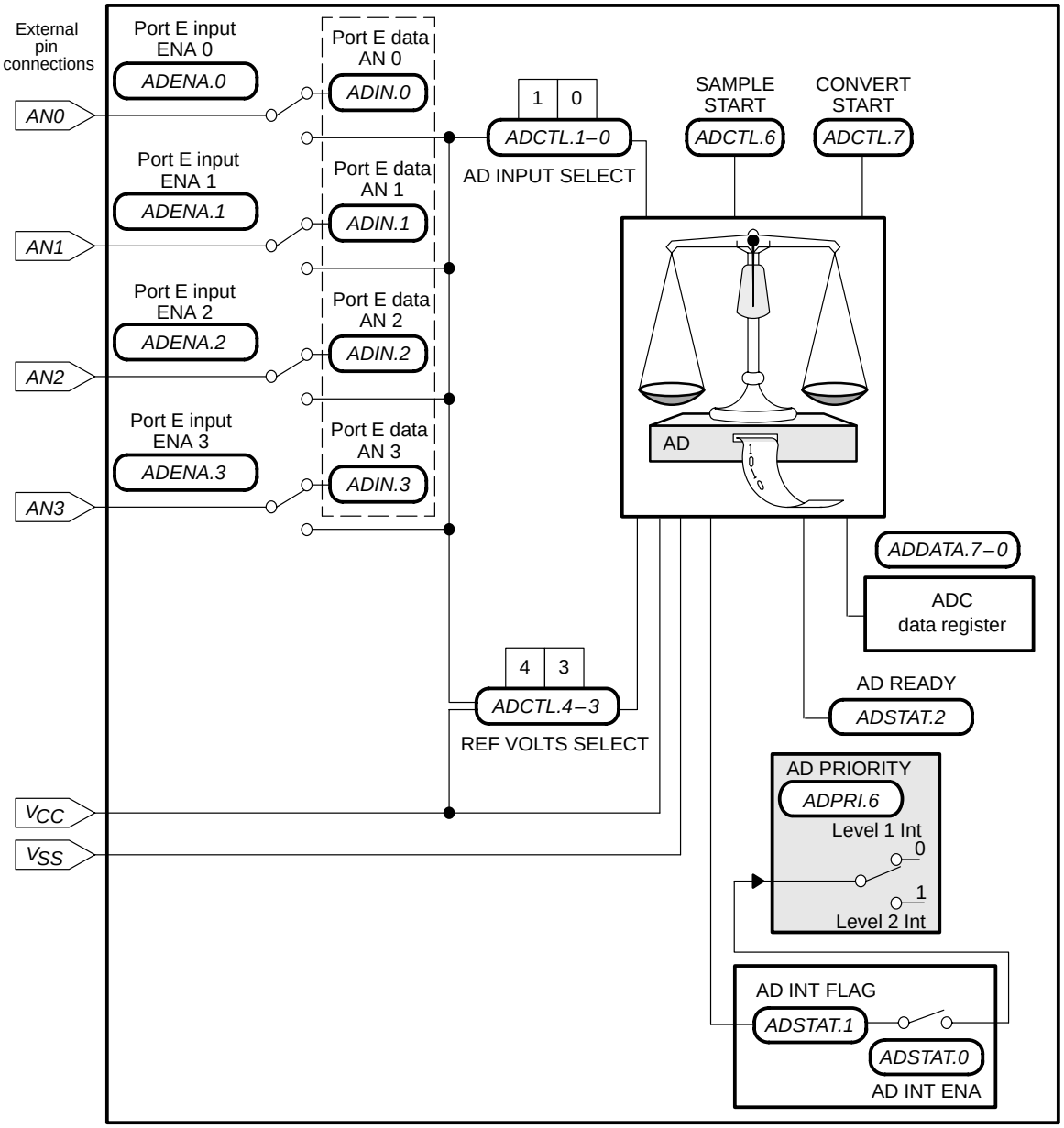
13.1.1 Physical Description

The ADC2 module, shown in Figure 13–1, consists of:

- The ADC2 conversion block
- Four analog input channels (AN0-AN3), any of which can be software-configured as digital inputs (E0-E3) if not needed as analog channels
- An ADC2 input selector (INPUT)
- A $+V_{REF}$ input selector ($+V_{REF}$)
- The ADDATA register, which contains the digital value of a completed conversion
- ADC2 module control registers

The input channels can be routed through either the channel selector or the positive voltage selector. The ADC2 converter then processes these signals and puts the result in the ADDATA register. The ADC2 interrupt circuit informs the rest of the system when a conversion is completed.

Figure 13–1. Analog-to-Digital Converter 2 (ADC2) Block Diagram



13.1.2 Control Registers

The ADC2 control registers are located at addresses 1070h to 107Fh and occupy peripheral file frame 7, as shown in Table 13–1.

Table 13–1. ADC2 Memory Map

Peripheral File Location	Symbol	Name	Description
P070	ADCTL	Analog Control Register	Controls the input selection, reference voltage selection, sample start, and conversion start.
P071	ADSTAT	Analog Status and Interrupt Register	Indicates the converter and interrupt status.
P072	ADDATA	Analog Conversion Data Register	Contains the digital result of the last ADC2 conversion.
P073–P07C		Reserved	
P07D	ADIN	Analog Port E Data Input Register	Contains digital input data when one or more of the AN0–AN3 pins are used as digital ports.
P07E	ADENA	Analog Port E Input Enable Register	Controls the function of the AN0–AN3 pins.
P07F	ADPRI	Analog Interrupt Priority Register	Selects the interrupt priority level of the ADC2 interrupt.

13.2 ADC2 Operation

The following subsections describe the functions and options of the ADC2 module.

13.2.1 Input/Output Pins

The ADC2 module uses six pins to connect itself to the external world: AN0-AN3, V_{CC} and V_{SS}. These pins are described below:

- Four pins (AN0-AN3) are analog channels and can be individually configured as general-purpose input pins (E0–E3) when not used as analog inputs.
- Three (AN1-AN2) of the four analog channels are also available as the positive input voltage reference. This feature allows a weighted measurement or ratio of one channel to another.

Note that you can select the V_{REF} pin to be either V_{CC} or one of the analog input channels AN1 to AN3. A channel configured as the $+V_{REF}$ for one conversion can be changed to an analog input channel for the next conversion.

13.2.2 Sampling Time

The application program controls the length of the sample time, which provides the flexibility to optimize the conversion process for both high- and low-impedance sources. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low-impedance sources.

13.2.3 ADC2 Conversion

The digital result of the conversion process is given in the following formula:

$$\text{digital result} = 255 \times \frac{\text{input voltage}}{\text{reference voltage}}$$

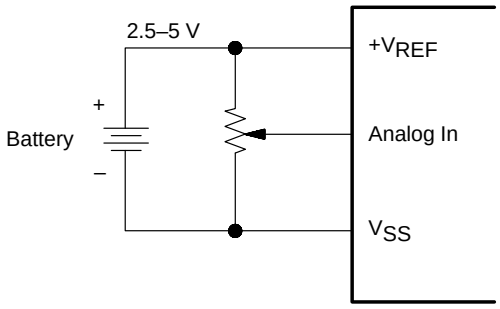
The conversion process requires 164 SYSCLK cycles and results in a conversion time of 32.8 microseconds at 5 MHz SYSCLK. A maximum of 27,600 conversions per second is possible at 5 MHz SYSCLK, including setting up the conversion, sampling, converting, and saving the results.

In ratiometric conversions, the conversion value is a ratio of the V_{REF} source to the analog input. As V_{REF} is increased, the input voltage that is required in order to produce a certain conversion value changes; however, all conversion

values keep the same relationship to V_{REF}. That is, one half of V_{REF} always results in the value 080h, regardless of the value of V_{REF} (assuming that V_{REF} is in the range of 2.5 to 5.5 volts above V_{SS}).

Figure 13–2 shows an example of ratiometric conversion. In this example, the digital result of the conversion indicates the position of the potentiometer wiper, even if the battery loses voltage over time. The ADC2 conversion always gives the ratio of the resistor values on either side of the wiper, even if V_{REF} drops from 5.0 to 2.5 volts.

Figure 13–2. Ratiometric Conversion Example



13.2.4 Interrupts

The ADC2 module sets the AD INT FLAG bit (ADSTAT.1) at the end of the conversion process. If both the AD INT FLAG and the AD INT ENA bit (ADSTAT.0) are set, then the module generates an interrupt request. This interrupt request can be asserted on either high-priority level 1 or the lower-priority level 2, depending on the AD PRIORITY bit (ADPRI.6).

The program must clear the AD INT FLAG bit before exiting the interrupt service routine (ISR), or else the same interrupt will cause the CPU to enter the interrupt routine again. If the AD INT ENA bit is cleared without clearing the flag, the interrupt is reasserted when the AD INT ENA bit is again set.

13.2.5 Programming Considerations

Follow these steps to obtain data from the ADC2:

1) Write to the ADCTL register (described on page 13-11) to:
   a) Select the analog channel (ADCTL.1–0).
   b) Select the V_{REF} source (ADCTL.4–3).
   c) Set the SAMPLE START bit to 1 (ADCTL.6) to begin sampling.
2) Wait for the sample time to elapse. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low impedance sources.
3) When the sample time completes, set the CONVERT START bit (ADCTL.7); leave the SAMPLE START bit (ADCTL.6) set.
4) Wait for either the interrupt flag to be set or the ADC2 interrupt to occur.
5) Read the conversion data register (ADDATA).
6) Clear the interrupt flag bit (ADSTAT.1).

Eighteen SYSCLK cycles after the program sets the CONVERT START bit, the ADC2 module clears both the SAMPLE START and CONVERT START bits to signify the end of the internal sampling phase. After these bits are cleared, the program can change the input channel without affecting the conversion process. The voltage reference source V_{REF} should remain constant throughout the conversion.

To stop a conversion in progress, set the SAMPLE START (ADCTL.6) bit to 1 anytime after the ADC2 clears this bit. The entire conversion process requires 164 SYSCLK cycles after the program sets the CONVERT START bit (ADCTL.7).

13.3 ADC2 Example Program

This example program (next page) samples and converts data from all four channels and stores the digital results into a table beginning at ATABLE. The routine stops interrupting the main program after it finishes all four channels. If the main program wants more recent data, it needs to execute only the code at RESTART, and the ADC2 routine will again sample and convert all four channels of data. The AD INT ENA bit (ADSTAT.0) is cleared by the ADC2 interrupt routine as a signal to the main program that all four channels have been processed. The address of the label ATOD must be placed into the interrupt vector table located at 7FECh and 7FEDh.

```
ADCTL    .EQU  P070          ;ADC2 control register
ADSTAT   .EQU  P071          ;ADC2 status register
ADDATA   .EQU  P072          ;ADC2 conversion results
ADENA    .EQU  P07E          ;ADC2 input enable
         .REG  ADCHANL       ;keeps current channel number
         .REG  ATABLE,8      ;8 byte table that stores
                             ; channel data, LSB first
;
INIT     MOV   #0,ADENA      ;all channels to ADC2 inputs
                             ; (reset condition)
         CALL  RESTART       ;start taking data
;
;        MAIN PROGRAM GOES HERE
;        .
;        .
         CALL  RESTART       ;start taking more data
;        .
;        .
;        MORE MAIN PROGRAM
;
;        SUBROUTINE SECTION
RESTART  CLR   ADCHANL       ;initialize channel
         MOV   #001h,ADSTAT  ;enable interrupts, clear
                             ; any flag
         MOV   #040h,ADCTL   ;start sampling (approx. 2 µs
                             ; delay)
         MOV   #0C0h,ADCTL   ;start converting now; enter
                             ; main program
         RTS
;
;        INTERRUPT ROUTINE FOR ANALOG TO DIGITAL CONVERTER2
ATOD     PUSH  A             ;save registers
         PUSH  B
         MOV   ADCHANL,B     ;get channel number
         MOV   ADDATA,A      ;get ADC2 conversion value
         MOV   A,*ATABLE[B]  ;store in a table according to
                             ; channel number
         INC   B             ;point to next channel
         BTJZ  #4,B,GOCNVRT
;stop when all channels sampled
                             ;(bit3 =1)
         CLR   ADCHANL       ;reset the ADC2 channel
         MOV   #0,ADSTAT     ;turn off interrupt and
                             ; clear flag
         JMP   EXITA2D       ;all four channels taken, enable
                             ;set to 0 now
;
GOCNVRT  MOV   B,ADCHANL     ;store current ADC2 channel
         MOV   #01h,ADSTAT   ;clear interrupt flag
         OR    #040h,B
;set up sample bit in value
         MOV   B,ADCTL       ;start sampling channel data
         OR    #080h,ADCTL   ;start converting data
;
EXITA2D  POP   B             ;Restore data
         POP   A
         RTI
```

13.4 ADC2 Control Registers

The ADC2 module control registers occupy peripheral file frame 7, as shown in Figure 13–3. The bits shown in *shaded* boxes in Figure 13–3 are privilege mode bits; that is, they can be written only to in the privilege mode.

Figure 13–3. Peripheral File Frame 7: ADC2 Converter Control Registers

Desig-nation	ADDR	PF	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	—	REF VOLT SELECT1 (RW–0)	REF VOLT SELECT0 (RW–0)	—	AD INPUT SELECT1 (RW–0)	AD INPUT SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	—	—	—	—	Port E Data Input Register (R–0)			
ADENA	107Eh	P07E	—	—	—	—	Port E Input Enable Register (RW–0)			
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	—	—

13.4.1 Analog Control Register (ADCTL)

The ADCTL register controls the input selection, reference voltage selection, sample start, and conversion start.

Analog Control Register (ADCTL)
[Memory Address 1070h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P070	CONVERT START	SAMPLE START	–	REF VOLT SELECT1	REF VOLT SELECT0	–	AD INPUT SELECT1	AD INPUT SELECT0
	RW-0	RW-0		RW-0	RW-0		RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bit 7 **CONVERT START.** Conversion Start.

Setting this bit starts the conversion. This bit is cleared by the ADC2 18 system clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any conversion in progress.

Bit 6 **SAMPLE START**. Sample Start.

Setting this bit stops any ongoing conversion and starts sampling the selected input channel to begin a new conversion. This bit is cleared by the ADC2 module 18 system-clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any sampling in progress.

Bit 5 **Reserved**. Read data is indeterminate.

Bits 4–3 **REF VOLT SELECT1–0**. Reference Voltage ($+V_{REF}$) Select Bits 1–0.

These bits select the channel the ADC2 uses for the positive voltage reference. The REF VOLT SELECT bits must not change during the entire conversion.

REF VOLT SELECT1	REF VOLT SELECT0	$+V_{REF}$ Source†
0	0	V_{CC}
0	1	AN1
1	0	AN2
1	1	AN3

† Pin AN0 cannot be selected as positive voltage reference.

Bit 2 **Reserved**. Read data is indeterminate

Bits 1–0 **AD INPUT SELECT1–0**. Analog Input Channel Select Bits 1–0.

These bits select the channel used for conversion. Channels should be changed only after the ADC2 has cleared the SAMPLE START and CONVERT START bits. Changing the channel while either the SAMPLE START bit or the CONVERT START bit is 1 invalidates the conversion in progress.

AD INPUT SELECT1	AD INPUT SELECT0	Channel
0	0	AN0
0	1	AN1
1	0	AN2
1	1	AN3

13.4.2 Analog Status and Interrupt Register (ADSTAT)

The ADSTAT register indicates the converter and interrupt status.

Analog Status and Interrupt Register (ADSTAT)
[Memory Address 1071h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P071	—	—	—	—	—	AD READY	AD INT FLAG	AD INT ENA
						R-0	RC-0	RW-0

R = Read, W = Write, C = Clear only, -*n* = Value of the bit after the register is reset

Bits 7–3 **Reserved.** Read data is indeterminate.

Bit 2 **AD READY**. ADC2 Converter Ready.

The ADC2 module sets this bit whenever a conversion is not in progress and the ADC2 is ready for a new conversion to start. Writing to this bit has no effect on its state.

0 = Conversion in process.
1 = Converter ready.

Bit 1 **AD INT FLAG**. ADC2 Interrupt Flag.

The ADC2 module sets this bit at the end of an ADC2 conversion. If this bit is set while the AD INT ENA bit is set, an interrupt request is generated. Clearing this flag clears pending ADC2 interrupt requests. This bit is cleared by the system reset. Software cannot set this bit.

Bit 0 **AD INT ENA**. ADC2 Interrupt Enable.

This bit controls the ADC2 module's ability to generate an interrupt.

0 = Disables ADC2 interrupt.
1 = Enables ADC2 interrupt.

13.4.3 Analog Conversion Data Register (ADDATA)

The ADDATA register contains the digital result of the last ADC2 conversion.

Analog Conversion Data Register (ADDATA)
[Memory Address 1072h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P072	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

The ADC2 data is loaded into this register at the end of a conversion and remains until replaced by another conversion.

13.4.4 Analog Port E Data Input Register (ADIN)

The ADIN register contains digital input data when one or more of the AN0 through AN3 pins are used as digital ports.

Analog Port E Data Input Register (ADIN)
[Memory Address 107Dh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07D	–	–	–	–	PORT E DATA AN 3	PORT E DATA AN 2	PORT E DATA AN1	PORT E DATA AN 0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, *-n* = Value of the bit after the register is reset

The ADIN register shows the data present at the AN0–AN3 pins when they are configured for general-purpose input instead of for ADC2 channels. A bit is configured as a general-purpose input if the corresponding bit of the port enable register is a 1. Pins configured as ADC2 channels are read as 0s. Writing to this address has no effect.

13.4.5 Analog Port E Input Enable Register (ADENA)

The ADENA register controls the function of the AN0 through AN3 pins.

Analog Port E Data Input Enable Register (ADENA)
[Memory Address 107Eh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07E	–	–	–	–	PORT E INPUT ENA 3	PORT E INPUT ENA 2	PORT E INPUT ENA 1	PORT E INPUT ENA 0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

The ADENA register individually configures the AN0–AN3 pins as either analog input channels or as general-purpose input pins.

0 = For pins AN1–AN3, the pin becomes an analog input or reference channel for the ADC2; for pin AN0, the pin becomes an analog input channel for the ADC2.
When the bit is 0, the corresponding bit in the ADIN register reads as a 0.

1 = Enables the pin as a general-purpose input pin; its digital value can be read from the corresponding bit in the port E data input register.

13.4.6 Analog Interrupt Priority Register (ADPRI)

The ADPRI register selects the interrupt priority level of the ADC2 interrupt.

Analog Interrupt Priority Register (ADPRI)
[Memory Address 107Fh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07F	AD STEST	AD PRIORITY	AD ESPEN	—	—	—	—	—
	RP-0	RP-0	RP-0					

R = Read, P = Privilege write only, -*n* = Value of the bit after the register is reset

Bit 7 **AD STEST**. This bit must be cleared (0) to ensure proper operation.

Bit 6 **AD PRIORITY**. ADC2 Interrupt Priority Select.

This bit selects the priority level of the ADC2 interrupt.

0 = ADC2 interrupt is a higher priority (level 1) request.
1 = ADC2 interrupt is a lower priority (level 2) request.

Bit 5 **AD ESPEN**. Emulator Suspend Enable.

Normally, this bit has no effect. However, when you are using the XDS emulator to debug a program, this bit determines what happens to the ADC2 when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the ADC2 continues to work until the current conversion is complete.
1 = When the emulator is suspended, the ADC2 is frozen so that its state can be examined at the point that the emulator was suspended. The conversion data is indeterminate upon restart.

Bits 4–0 **Reserved.** Read data is indeterminate.

Chapter 14

Analog-To-Digital Converter 3 (ADC3) Module

The TMS370 family contains three different ADC modules (ADC1, ADC2, and ADC3). This chapter discusses the architecture and programming of the analog-to-digital converter 3 (ADC3) module and covers the following topics:

Topic **Page**

14.1 Analog-to-Digital Converter 3 (ADC3) Overview

The analog-to-digital converter 3 module is an 8-bit, successive approximation converter with internal sample-and-hold circuitry. The module has 15 analog input channels that allow the processor to convert the voltage levels from up to 15 different sources.

Note: ADC3 Module Availability

The ADC3 module is available for the TMS370Cx9x family.

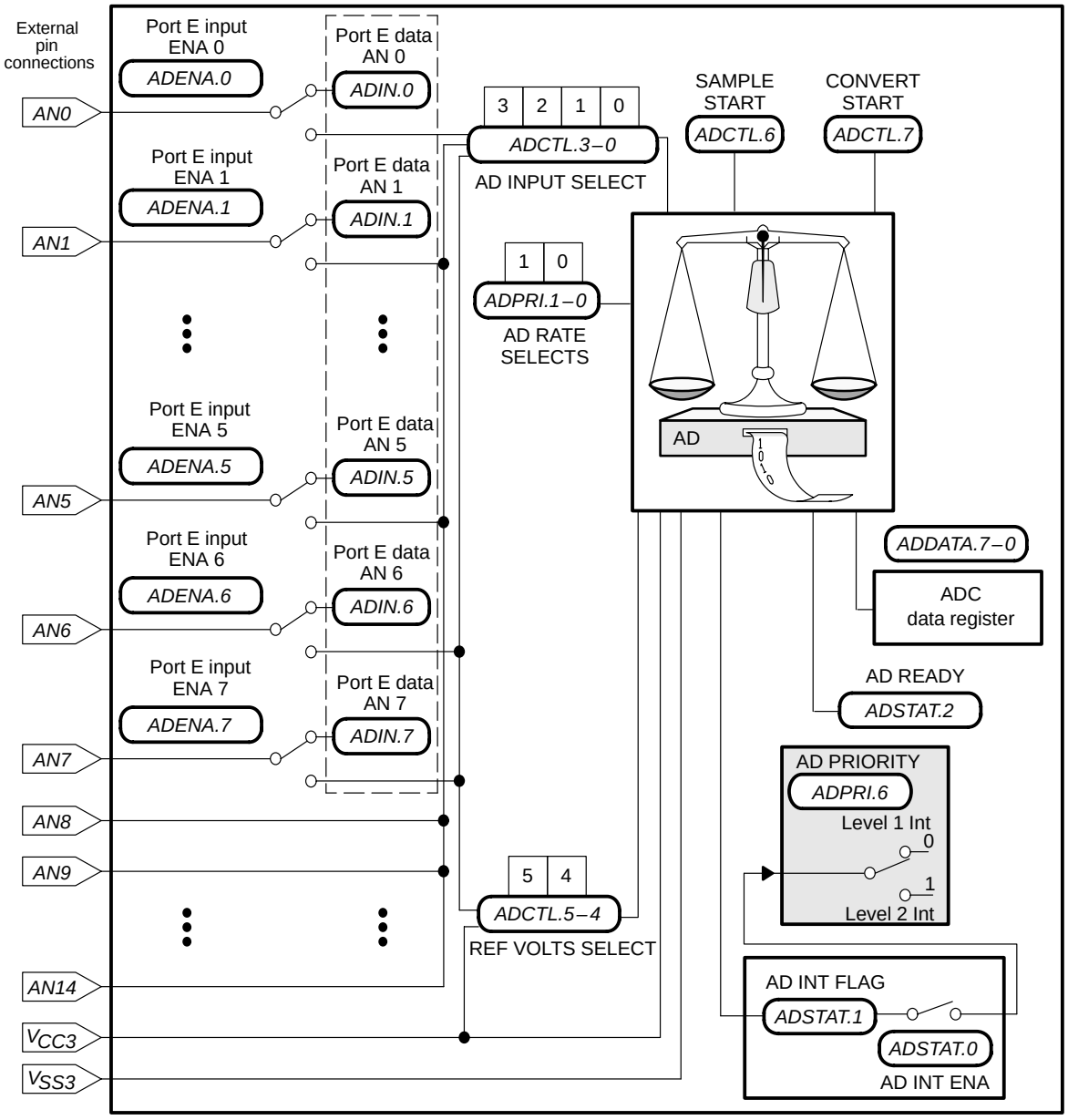
14.1.1 Physical Description

The ADC3 module, shown in Figure 14–1, consists of:

- Fifteen analog input channels (AN0-AN14), eight (AN0–AN7) of which can be software-configured as digital inputs (E0-E7) if not needed as analog channels
- The ADC3 conversion block
- An ADC3 conversion rate selector
- An ADC3 input selector (INPUT)
- A $+V_{REF}$ input selector ($+V_{REF}$)
- The ADDATA register, which contains the digital value of a completed conversion
- ADC3 module control registers

The input channels can be routed through either the channel selector or the positive voltage selector. The ADC3 converter then processes these signals and puts the result in the ADDATA register. The ADC3 interrupt circuit informs the rest of the system when a conversion is completed.

Figure 14–1. Analog-to-Digital Converter 3 (ADC3) Block Diagram



14.1.2 Control Registers

The ADC3 control registers are located at addresses 1070h to 107Fh and occupy peripheral file frame 7, as shown in Table 14–1.

Table 14–1. ADC3 Memory Map

Peripheral File Location	Symbol	Name	Description
P070	ADCTL	Analog Control Register	Controls the input selection, reference voltage selection, sample start, and conversion start.
P071	ADSTAT	Analog Status and Interrupt Register	Indicates the converter and interrupt status.
P072	ADDATA	Analog Conversion Data Register	Contains the digital result of the last ADC3 conversion.
P073–P07C		Reserved	
P07D	ADIN	Analog Port E Data Input Register	Contains digital input data when one or more of the AN0–AN7 pins are used as digital ports.
P07E	ADENA	Analog Port E Input Enable Register	Controls the function of the AN0–AN7 pins.
P07F	ADPRI	Analog Interrupt Priority Register	Selects the interrupt priority level of the ADC3 interrupt and the conversion rate selects.

14.2 ADC3 Operation

The following subsections describe the functions and options of the ADC3 module.

14.2.1 Input/Output Pins

The ADC3 module contains 17 pins: AN0-AN14, V_{CC3}, and V_{SS3}. These pins are described below:

- Fifteen (AN0-AN14) of the 17 pins are analog channels. Eight (AN0-AN7) of the 15 analog channels can be individually configured as general-purpose input pins (E0-E7) when not used as analog inputs.
- Two (AN6-AN7) of the fifteen analog channels are also available as the positive input voltage reference. This feature allows a weighted measurement or ratio of one channel to another.
- The analog voltage supply pins, V_{CC3} and V_{SS3}, isolate the ADC3 module from digital switching noise that can be present on the other power supply pins. This isolation provides a more accurate conversion.
- To further reduce noise and produce a more accurate conversion, you should run the power to the V_{CC3} and V_{SS3} pins on separate conductors from the other power lines. Additionally, the power conductors to the VCC3 and VSS3 should be as short as possible, and the two lines should be properly decoupled. Use other standard noise-reduction techniques to help provide a more accurate conversion.

Note that you can select the V_{REF} pin to be either V_{CC3} or one of the analog input channels AN6 to AN7. A V_{CC3} must provide power to the ADC3 module even if it does not provide the voltage reference. A channel configured as the $+V_{REF}$ for one conversion can be changed to an analog input channel for the next conversion.

14.2.2 Sampling Time

The application program controls the length of the sample time, which provides the flexibility to optimize the conversion process for both high- and low-impedance sources. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low-impedance sources.

14.2.3 ADC3 Conversion

The digital result of the conversion process is given in the following formula:

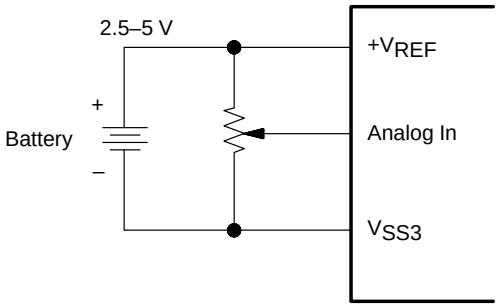
$$\text{digital result} = 255 \times \frac{\text{input voltage}}{\text{reference voltage}}$$

The number of SYSCLK cyles required to complete a conversion is programmable through the AD RATE SELECT0-1 bits (P07F.0–1). For example, the conversion process requires 164 SYSCLK cycles and results in a conversion time of 32.8 microseconds at a SYSCLK frequency of less than or equal 5 MHz and greater than 2.5 MHz (5 MHz ≥ SYSCLK > 2.5 MHz). A maximum of 27,600 conversions per second is possible at a SYSCLK frequency within the range above, including setting up the conversion, sampling, converting, and saving the results.

In ratiometric conversions, the conversion value is a ratio of the V_{REF} source to the analog input. As V_{REF} is increased, the input voltage that is required in order to produce a certain conversion value changes; however, all conversion values keep the same relationship to V_{REF}. That is, one half of V_{REF} always results in the value 080h, regardless of the value of V_{REF} (assuming that V_{REF} is in the range of 2.5 to 5.5 volts above V_{SS3}).

Figure 14–2 shows an example of ratiometric conversion. In this example, the digital result of the conversion indicates the position of the potentiometer wiper, even if the battery loses voltage over time. The ADC3 conversion always gives the ratio of the resistor values on either side of the wiper, even if V_{REF} drops from 5.0 to 2.5 volts.

Figure 14–2. Ratiometric Conversion Example



14.2.4 Interrupts

The ADC3 module sets the AD INT FLAG bit (ADSTAT.1) at the end of the conversion process. If both the AD INT FLAG and the AD INT ENA bit (ADSTAT.0) are set, then the module generates an interrupt request. This interrupt request can be asserted on either high-priority level 1 or the lower-priority level 2, depending on the AD PRIORITY bit (ADPRI.6).

The program must clear the AD INT FLAG bit before exiting the interrupt service routine (ISR), or else the same interrupt will cause the CPU to enter the interrupt routine again. If the AD INT ENA bit is cleared without clearing the flag, the interrupt is reasserted when the AD INT ENA bit is again set.

14.2.5 Programming Considerations

Follow these steps to obtain data from the ADC3:

1) Write to the two bits (AD RATE SELECT 1–0) on the ADPRI register (subsection 14.4.6 on page 14-16).

2) Write to the ADCTL register (subsection 14.4.1 on page 14-12) to:

   a) Select the analog channel (ADCTL.3–0).

   b) Select the V_{REF} source (ADCTL.5–4).

   c) Set the SAMPLE START bit to 1 (ADCTL.6) to begin sampling.

3) Wait for the sample time to elapse. The program should wait 1 µs for each kilohm of source output impedance or a minimum of 1 µs for low impedance sources.

4) When the sample time completes, set the CONVERT START bit (ADCTL.7) and leave the SAMPLE START bit (ADCTL.6) set.

5) Wait for either the interrupt flag to be set or the ADC3 interrupt to occur.

6) Read the conversion data register (ADDATA, subsection 14.4.3 on page 14-15).

7) Clear the interrupt flag bit (ADSTAT.1).

Eighteen SYSCLK cycles after the program sets the CONVERT START bit, the ADC3 module clears both the SAMPLE START and CONVERT START bits to signify the end of the internal sampling phase. After these bits are cleared, the program can change the input channel without affecting the conversion process. The voltage reference source V_{REF} should remain constant throughout the conversion.

To stop a conversion in progress, set the SAMPLE START (ADCTL.6) bit to 1 anytime after the ADC3 clears this bit. For SYSCLK frequencies less than or equal to 5MHz and greater than 2.5MHz, the entire conversion process requires 164 SYSCLK cycles after the program sets the CONVERT START bit (ADCTL.7).

14.3 ADC3 Example Program

This example program samples and converts data from all 15 channels at 1.0 MHz SYSCLK frequency and stores the digital results into a table beginning at ATABLE. The routine stops interrupting the main program after it finishes all 15 channels. If the main program wants more recent data, it needs to execute only the code at RESTART, and the ADC3 routine will again sample and convert all 15 channels of data. The AD INT ENA bit (ADSTAT.0) is cleared by the ADC3 interrupt routine as a signal to the main program that all 15 channels have been processed. The address of the label ATOD must be placed into the interrupt vector table located at 7FECh and 7FEDh.

```
ADCTL    .EQU  P070          ;ADC3 control register
ADSTAT   .EQU  P071          ;ADC3 status register
ADDATA   .EQU  P072          ;ADC3 conversion results
ADENA    .EQU  P07E          ;ADC3 input enable
         .REG  ADCHANL       ;keeps current channel number
         .REG  ATABLE,8      ;8 byte table that stores
                             ; channel data, LSB first
INIT     MOV   #0,ADENA      ;all channels to ADC3 inputs
                             ; (reset condition)
         CALL  RESTART       ;start taking data
;
;        MAIN PROGRAM GOES HERE
;        .
         CALL  RESTART       ;start taking more data
;        .
;        MORE MAIN PROGRAM
;
;        SUBROUTINE SECTION
RESTART  CLR   ADCHANL       ;initialize channel
         MOV   #002h,ADPRI   ;select the conversion rate of
                             ; 44 SYSCLK cycles and ADC3
                             ; interrupt-request priority
                             ; level 1
         MOV   #001h,ADSTAT  ;enable interrupts, clear
                             ; any flag
         MOV   #040h,ADCTL   ;start sampling (approx. 2 µs
                             ; delay)
         MOV   #0C0h,ADCTL   ;start converting now; enter
                             ; main program
         RTS
;
;        INTERRUPT ROUTINE FOR ANALOG TO DIGITAL CONVERTER3
ATOD     PUSH  A             ;save registers
         PUSH  B
         MOV   ADCHANL,B     ;get channel number
         MOV   ADDATA,A      ;get ADC3 conversion value
         MOV   A,*ATABLE[B]  ;store in a table according to
                             ; channel number
         INC   B             ;point to next channel
         BTJZ  #15,B,GOCNVRT
;stop when all channels sampled
                             ;(bit3 =1)
         CLR   ADCHANL       ;reset the ADC3 channel
         MOV   #0,ADSTAT     ;turn off interrupt and
                             ; clear flag
         JMP   EXITA2D       ;all 8 channels taken, enable
                             ;set to 0 now
;
GOCNVRT  MOV   B,ADCHANL     ;store current ADC3 channel
         MOV   #01h,ADSTAT   ;clear interrupt flag
         OR    #040h,B
;set up sample bit in value
         MOV   B,ADCTL       ;start sampling channel data
         OR    #080h,ADCTL   ;start converting data
;
EXITA2D  POP   B             ;Restore data
         POP   A
         RTI
```

14.4 ADC3 Control Registers

The ADC3 module control registers occupy peripheral file frame 7, as shown in Figure 14–3. The bits shown in shaded boxes in Figure 14–3 are privilege mode bits; that is, they can be written only to in the privilege mode.

Figure 14–3. Peripheral File Frame 7: ADC3 Converter Control Registers

Designation	ADDR	PF	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	REF VOL SELECT1 (RW–0)	REF VOL SELECT0 (RW–0)	AD INPUT SELECT3 (RW–0)	AD INPUT SELECT2 (RW–0)	AD INPUT SELECT1 (RW–0)	AD INPUT SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	Port E Data Input Register (R–0)							
ADENA	107Eh	P07E	Port E Input Enable Register (RW–0)							
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	AD RATE-SELECT1 (RW–0)	AD RATE-SELECT0 (RW–0)

14.4.1 Analog Control Register (ADCTL)

The ADCTL register controls the input selection, reference voltage selection, sample start, and conversion start.

Analog Control Register (ADCTL)
[Memory Address 1070h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P070	CONVERT START	SAMPLE START	REF VOLT SELECT1	REF VOLT SELECT0	AD INPUT SELECT3	AD INPUT SELECT2	AD INPUT SELECT1	AD INPUT SELECT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -n = Value of the bit after the register is reset

Bit 7 **CONVERT START.** Conversion Start.

Setting this bit starts the conversion. This bit is cleared by the ADC3 18 system clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any conversion in progress.

Bit 6 **SAMPLE START**. Sample Start.

Setting this bit stops any ongoing conversion and starts sampling the selected input channel to begin a new conversion. This bit is cleared by the ADC3 module 18 system-clock cycles after the program sets the CONVERT START bit. Entering halt or standby mode clears this bit and aborts any sampling in progress.

Bits 5–4 **REF VOLT SELECT1–0**. Reference Voltage ($+V_{REF}$) Select Bits 1–0.

These bits select the channel the ADC3 uses for the positive voltage reference. The REF VOLT SELECT bits must not change during the entire conversion.

REF VOLT SELECT1	REF VOLT SELECT0	$+V_{REF}$ Source†
0	0	V_{CC3}
0	1	AN6
1	0	AN7

† Pins AN0–AN5 and AN8–AN14 cannot be selected as positive voltage reference.

Bits 3–0 **AD INPUT SELECT3–0**. Analog Input Channel Select Bits 3–0.

These bits select the channel used for conversion. Channels should be changed only after the ADC3 has cleared the SAMPLE START and CONVERT START bits. Changing the channel while either the SAMPLE START bit or the CONVERT START bit is 1 invalidates the conversion in progress.

AD INPUT SELECT3	AD INPUT SELECT2	AD INPUT SELECT1	AD INPUT SELECT0	AD INPUT CHANNEL
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13
1	1	1	0	AN14

14.4.2 Analog Status and Interrupt Register (ADSTAT)

The ADSTAT register indicates the converter and interrupt status.

Analog Status and Interrupt Register (ADSTAT)
[Memory Address 1071h]

Bit #	7	6	5	4	3	2	1	0
P071	—	—	—	—	—	AD READY	AD INT FLAG	AD INT ENA
						R-0	RC-0	RW-0

R = Read, W = Write, C = Clear only, -*n* = Value of the bit after the register is reset

Bits 7–3 **Reserved.** Read data is indeterminate.

Bit 2 **AD READY**. ADC3 Converter Ready.

The ADC3 module sets this bit whenever a conversion is not in progress and the ADC3 is ready for a new conversion to start. Writing to this bit has no effect on its state.

0 = Conversion in process.
1 = Converter ready.

Bit 1 **AD INT FLAG**. ADC3 Interrupt Flag.

The ADC3 module sets this bit at the end of an ADC3 conversion. If this bit is set while the AD INT ENA bit is set, an interrupt request is generated. Clearing this flag clears pending ADC3 interrupt requests. This bit is cleared by the system reset. Software cannot set this bit.

Bit 0 **AD INT ENA**. ADC3 Interrupt Enable.

This bit controls the ADC3 module's ability to generate an interrupt.

0 = Disables ADC3 interrupt.
1 = Enables ADC3 interrupt.

14.4.3 Analog Conversion Data Register (ADDATA)

The ADDATA register contains the digital result of the last ADC3 conversion.

Analog Conversion Data Register (ADDATA)
[Memory Address 1072h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P072	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

The ADC3 data is loaded into this register at the end of a conversion and remains until replaced by another conversion.

14.4.4 Analog Port E Data Input Register (ADIN)

The ADIN register contains digital input data when one or more of the AN0 through AN7 pins are used as digital ports.

Analog Port E Data Input Register (ADIN)
[Memory Address 107Dh]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P07D	PORT E DATA AN 7	PORT E DATA AN 6	PORT E DATA AN 5	PORT E DATA AN 4	PORT E DATA AN 3	PORT E DATA AN 2	PORT E DATA AN 1	PORT E DATA AN 0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

The ADIN register shows the data present at the AN0–AN7 pins when they are configured for general-purpose input instead of for ADC3 channels. A bit is configured as a general-purpose input if the corresponding bit of the port enable register is a 1. Pins configured as ADC3 channels are read as 0s. Writing to this address has no effect.

14.4.5 Analog Port E Input Enable Register (ADENA)

The ADENA register controls the function of the AN0 through AN7 pins.

Analog Port E Data Input Enable Register (ADENA)
[Memory Address 107Eh]

Bit #	7	6	5	4	3	2	1	0
P07E	PORT E INPUT ENA 7	PORT E INPUT ENA 6	PORT E INPUT ENA 5	PORT E INPUT ENA 4	PORT E INPUT ENA 3	PORT E INPUT ENA 2	PORT E INPUT ENA 1	PORT E INPUT ENA 0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

The ADENA register individually configures the AN0–AN7 pins as either analog input channels or as general-purpose input pins.

0 = For pins AN1–AN7, the pin becomes an analog input or reference channel for the ADC3; for pin AN0, the pin becomes an analog input channel for the ADC3.
When the bit is 0, the corresponding bit in the ADIN register reads as a 0.

1 = Enables the pin as a general-purpose input pin; its digital value can be read from the corresponding bit in the port E data input register.

14.4.6 Analog Interrupt Priority Register (ADPRI)

The ADPRI register selects the interrupt priority level of the ADC3 interrupt.

Analog Interrupt Priority Register (ADPRI)
[Memory Address 107Fh]

Bit #	7	6	5	4	3	2	1	0
P07F	AD STEST	AD PRIORITY	AD ESPEN	—	—	—	AD RATE SELECT1	ADD RATE SELECT0
	RP-0	RP-0	RP-0					

R = Read, P = Privilege write only, *-n* = Value of the bit after the register is reset

Bit 7 **AD STEST**. This bit must be cleared (0) to ensure proper operation.

Bit 6 **AD PRIORITY**. ADC3 Interrupt Priority Select.

This bit selects the priority level of the ADC3 interrupt.

0 = ADC3 interrupt is a higher priority (level 1) request.
1 = ADC3 interrupt is a lower priority (level 2) request.

Bit 5 **AD ESPEN**. Emulator Suspend Enable.

Normally, this bit has no effect. However, when you are using the XDS emulator to debug a program, this bit determines what happens to the ADC3 when the program is suspended by an action such as a hardware or software breakpoint.

0 = When the emulator is suspended, the ADC3 continues to work until the current conversion is complete.
1 = When the emulator is suspended, the ADC3 is frozen so that its state can be examined at the point that the emulator was suspended. The conversion data is indeterminate upon restart.

Bits 4–2 **Reserved.** Read data is indeterminate.

Bits 1–0 **AD RATE SELECT1-0** ADC3 Conversion Rate Select Bits 1–0.

These bits determine the conversion rate of the ADC3 as a function of the system clock frequency. Note in Table 14–2 that only the default selection (0,0) provides full SYSCLK frequency range together with 8–bit precision. Other selections allow maintaining minimum conversion time at lower system clock rates.

Table 14–2. Conversion Rate Selection

AD RATE SELECT1	AD RATE SELECT0	Conversion Time (number of system clock cycles)	Max SYSCLK Frequency
0	0	164	5 MHz ≥ SYSCLK > 2.5 MHz
0	1	84	2.5 MHz ≥ SYSCLK > 1.25 MHz
1	0	44	1.25 MHz ≥ SYSCLK > 0.625 MHz
1	1	24	0.625 MHz ≥ SYSCLK ≥ 0.5 MHz

If selections different from (0,0) are used at SYSCLK frequencies higher than specified in the above table the 8-bit precision of the ADC3 is not guaranteed.

Chapter 15

Programmable Acquisition and Control Timer (PACT)

This chapter discusses the architecture and programming of the programmable acquisition and control timer (PACT) module. Even if you have extensive experience with microcontroller timers, you should read this chapter to fully understand how to use the TMS370 PACT module. The chapter covers the following topics:

Topic **Page**

15.1 PACT Overview

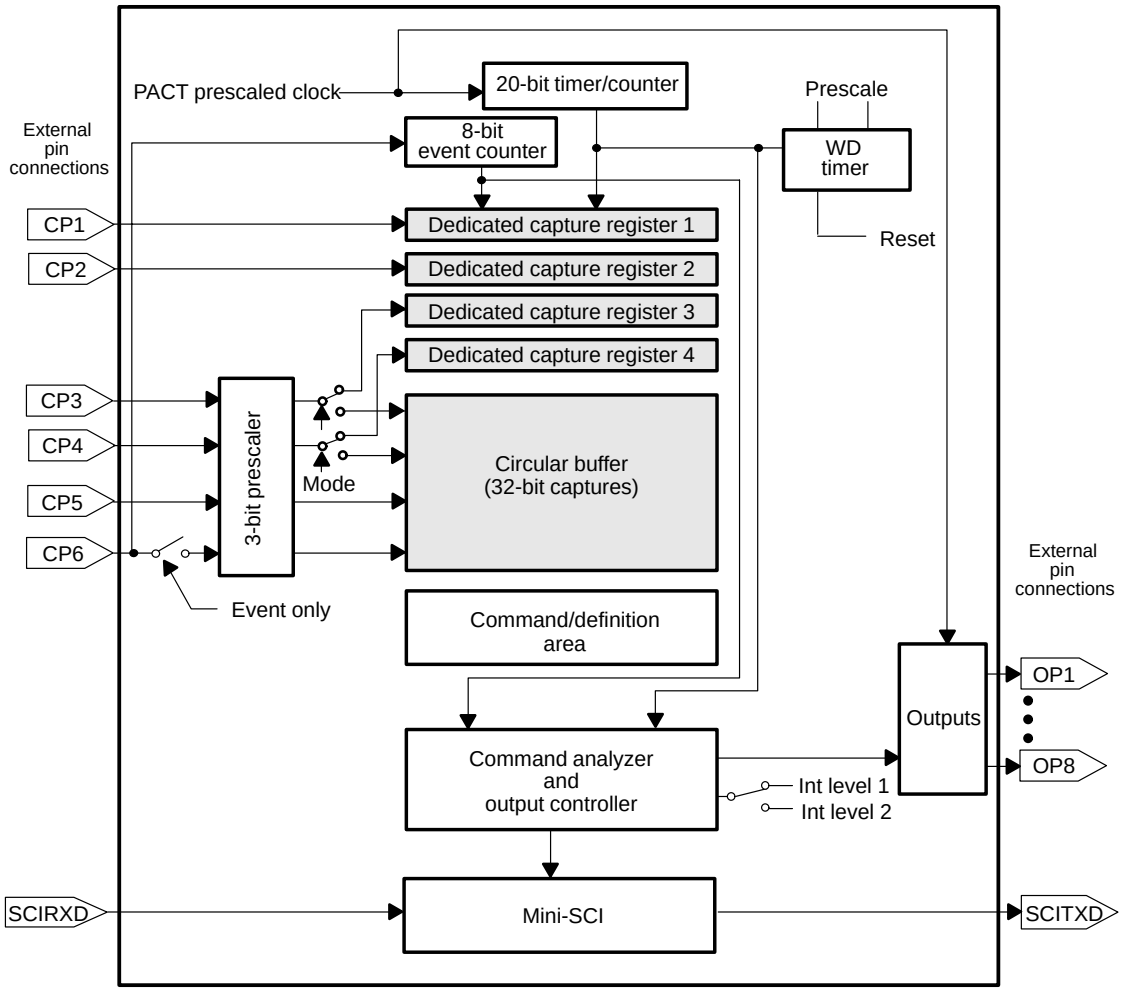
The PACT module acts as a timer coprocessor by gathering timing information on input signals and controlling output signals with little or no intervention by the CPU. The coprocessor nature of this module allows for levels of flexibility and power not found in traditional microcontroller timers. The PACT module is available in the TMS370Cx32 and TMS370Cx36 families.

15.1.1 Physical Description

The PACT module, shown in Figure 15–1, consists of:

- Input capture functions on up to six input pins, four of which (CP3–CP6) may have a programmable prescaler
- Timer-driven outputs on eight pins
- Configurable timer overflow rates for different functions
- One 8-bit event counter driven by CP6
- Timer capability of up to 20 bits
- Interaction between event counter and timer activity
- Register-based organization, allowing single-cycle accesses to parameters
- Eighteen independent interrupt vectors with two priority levels
- Integrated, configurable watchdog (WD) timer with selectable time-out period
- Mini-serial communications interface with independent setup of bit rate (baud) for receive and transmit lines

Figure 15–1. PACT Block Diagram



15.1.2 Control Registers

The PACT control registers are located at addresses 1040h to 104Fh and occupy peripheral file frame 4. The function of each register is shown in Table 15–1.

Table 15–1. PACT Peripheral Frame

Peripheral File Location	Symbol	Name	Description
P040	PACT SCR	Setup Control Register	Determines the time base for the PACT module, enables the command/definition area, and controls the default timer overflow
P041	CDSTART	Command/Definition Area Start Register	Defines the starting address of the command/definition area and enables the interrupts for that area
P042	CDEND	Command/Definition Area End Register	Defines the end address of the command/definition area
P043	BUFPTR	Buffer Pointer Register	Defines the address of the buffer pointer
P044		Reserved	
P045	SCICTLP	PACT-SCI Control Register	Controls the functions of the mini-SCI
P046	RXBUFP	PACT-SCI RX Data Register	Contains the data received by the SCI
P047	TXBUFP	PACT-SCI TX Data Register	Contains the data to be transmitted by the SCI
P048	OPSTATE	Output Pins 1–8 State Register	Contains information about the current state of the output pins
P049	CDFLAGS	Command/Definition Entry Flags Register	Contains information about the command/definition interrupts
P04A	CPCTL1	Setup CP Control Register 1	Controls the functions of the CP1 and CP2 pins
P04B	CPCTL2	Setup CP Control Register 2	Controls the functions of the CP3 and CP4 pins
P04C	CPCTL3	Setup CP Control Register 3	Controls the functions of the CP5 and CP6 pins
P04D	CPPRE	CP Input Control Register	Controls input and output functions
P04E	WDRST	WD Reset Key	Location that is written to when serving the WD
P04F	PACTPRI	Global Function Control Register	Controls the WD time-out rate, the PACT interrupt priority levels, and the PACT operating mode

The PACT module is controlled not only by the peripheral file but also by the defined areas of the dual-port RAM. Refer to Section 15.3 on page 15-9 for more information on the dual-port RAM.

15.2 PACT Operation

The following subsections describe the functions and options of the PACT module.

15.2.1 Hardware Pins

The PACT module has 16 external hardware pins allocated to its functions. There are three groups of pins:

- **Input Pins.** The input or capture pins are CP1 to CP6. The function of these pins depends on the mode selected:

Mode A		Mode B	
CP1–2	Dedicated capture	CP1–4	Dedicated capture
CP3–6	Circular buffer capture	CP5–6	Circular buffer capture
CP6	Event pin	CP6	Event pin

- **Output pins.** There are eight output pins: OP1–OP8.
- **SCI receive/transmit pins.** The PACT module has two pins for the mini-SCI: SCIRXD (receive data) and SCITXD (transmit data).

In the TMS370Cx32 devices, the input pins CP3, CP4, and CP5 are internally bonded with I/O pins D4, D6, and D7 respectively. In the TMS370Cx36 devices, the input pins CP1, CP4, CP5, and CP6 are internally bonded with I/O pins D5, D4, D7, and D6 respectively, giving you a software-governed choice of function. Output pins (OP1–8) are initialized to a logic low on reset.

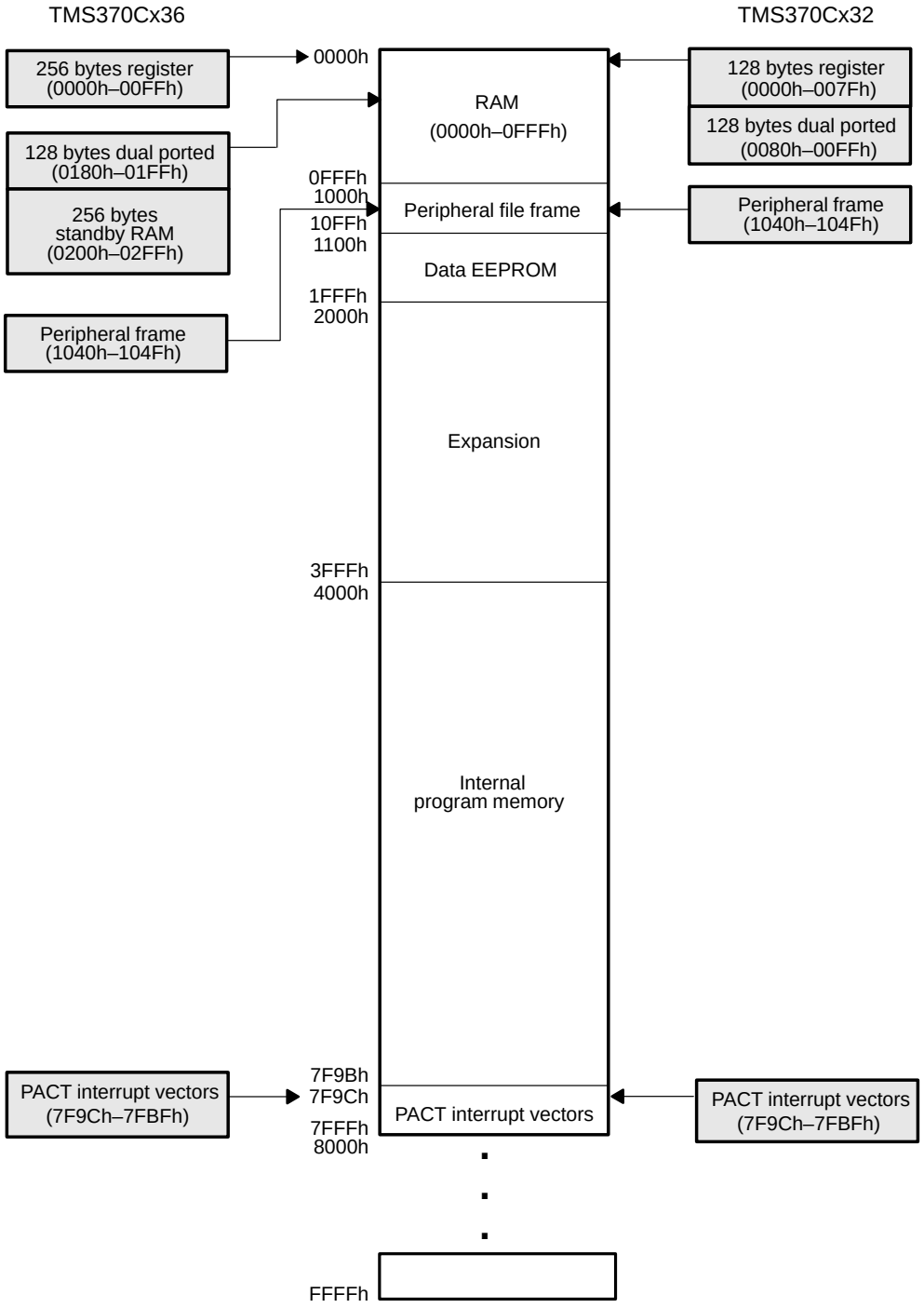
15.2.2 Memory Organization

To use the PACT module, you must set up three distinct areas of memory:

- **128 bytes of dual-port RAM** contain the capture registers, the circular buffer, and a command/definition area. Dual-port RAM is described in Section 15.3 on page 15-9.
- **Peripheral file frame 4** contains the hardware registers used for initial setup. These registers are described in Section 15.11 on page 15-36.
- Three groups of **interrupt vectors** are available. To use interrupts, you must set up the interrupt vectors. Interrupts are described in Section 15.7 on page 15-29.

The memory map in Figure 15–2 is a typical implementation of the PACT module.

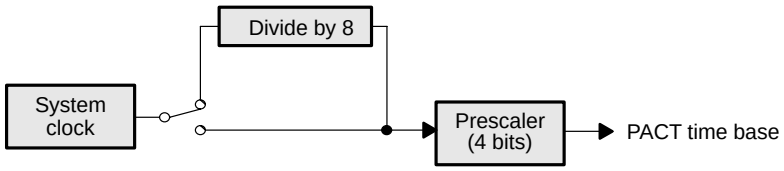
Figure 15–2. TMS370 Memory Map Highlighting PACT Areas of the ’Cx36 and ’Cx32



15.2.3 Time Base

The time-base section of PACT is similar to that used in traditional timers. The microcontroller system clock is routed to a prescaler that feeds a hardware counter. The prescale section consists of a 4-bit prescaler and an optional divide-by-8 circuit, as shown in Figure 15–3. The hardware counter is 20 bits wide.

Figure 15–3. Prescaler Circuit



The divide rate is the binary value of the 4-bit prescaler plus one except for the value zero which, by hardware, provides a divide rate of two. The five bits that control the prescaler are located in the PACTSCR register at address 1040h. Refer to subsection 15.11.1, on page 15-38, for more information.

PACT PRESCALE SELECT Bits	Divide Rate		PACT PRESCALE SELECT Bits	Divide Rate	
	FAST MODE SELECT Bit Value			FAST MODE SELECT Bit Value	
3 2 1 0	= 1	= 0	3 2 1 0	= 1	= 0
0 0 0 0	2	16	1 0 0 0	9	72
0 0 0 1	2	16	1 0 0 1	10	80
0 0 1 0	3	24	1 0 1 0	11	88
0 0 1 1	4	32	1 0 1 1	12	96
0 1 0 0	5	40	1 1 0 0	13	104
0 1 0 1	6	48	1 1 0 1	14	112
0 1 1 0	7	56	1 1 1 0	15	120
0 1 1 1	8	64	1 1 1 1	16	128

15.2.4 Command/Definition File Format

All entries in the command/definition area are 32 bits in length. Normally, commands are executed sequentially starting with entry zero (0), the entry next to the circular buffer. Each entry requires a specified number of time slots (the command/definition descriptions in Section 15.6, on page 15-21, explain the number of time slots required for each command).

15.2.5 Available Time Slots

The number of time slots available to you, and thus the number of definitions/commands allowed, is controlled by the crystal frequency and the resolution

required by PACT functions. Therefore, you must balance these factors to suit your application.

The PACT prescaler value determines the number of slots available. The prescaler gives the ratioed crystal frequency against resolution achievable by the PACT. Table 15–2 defines the maximum number of time slots available for all prescaler options.

Table 15–2. Number of Time Slots Available for Each Prescale Setting

Divide Rate	Time Slots	Divide Rate	Time Slots	Divide Rate	Time Slots
—	—	11	31	56	179
2	2	12	35	64	205
3	5	13	38	72	231
4	9	14	41	80	257
5	12	15	45	88	283
6	15	16	48	96	309
7	19	24	74	104	336
8	21	32	101	112	362
9	25	40	127	120	389
10	29	48	153	128	415

If your application requires a prescaler value that does not provide sufficient time slots, you can use the STEP command to cut in half the resolution of all commands later in the command/definition area. The STEP command affects the second entry after the entry that contains the command. (An example of PACT.H commands is in Section I.3 in Appendix I on page I-5).

For example, if 10 entries exist in the command file and entry 2 contains a STEP instruction, then commands 0–3 run at full resolution, but 4–9 run at half resolution.

Note that this use of the STEP command affects *all* operations, including the clocking of virtual and offset timers. Repeated use of the STEP instruction within a command file is not necessary—the commands run at either full or half speed (no slower).

Note: Consideraton Using STEP Command

When you use STEP, the end address of the command/definition area must be programmed as the next to last address that will be executed.

15.3 Dual-Port RAM

The PACT is a RAM-based module that occupies an area of the internal RAM. The size of the RAM is determined by the functions that you select, according to the end device.

For the TMS370Cx32 devices, the dual-port RAM is located in the register file from 0080h to 00FFh. This gives the CPU maximum speed in accessing the registers in the register file when you use the register address mode. For the TMS370Cx36 devices, the dual-port RAM is located in the memory map from 0180h to 01FFh.

The dual-port RAM contains the following major areas:

- **Command/definition area.** The length of the command/definition area (described in Section 15.6 on page 15-21) is defined in the software. Four bits define the start, and five bits define the end of the command area. The two least significant bits (LSBs) are not defined, because this area must start and finish on a 32-bit boundary. The start address of the command area also defines the length of the circular buffer because the buffer resides between the last dedicated storage location and the start of the command area.

  For the TMS370Cx32 devices, all RAM at lower addresses than the end of the command area can be used for general purposes such as registers, stack, etc. However, since all of the RAM locations above the end (0080h–00FFh) are also within the register file, the data in these locations can be directly manipulated with normal register-based instructions.

  For the TMS370Cx36 devices, all RAM between 0180h and the end of the command area can be used for data memory storage, program instructions, or a general purpose register. However, all of the RAM locations above the end (0180h–01FFh) can be used only as data memory storage.

- **Circular buffer.** The circular buffer is an area in memory that stores the value of a PACT timer when a capture request is made. As new values are captured, they are put into successive locations in the buffer. When the buffer is full, the oldest captures are replaced with newer captures.

  The length of the circular buffer is defined in the software. The circular buffer resides between the command/definition area and the capture registers.

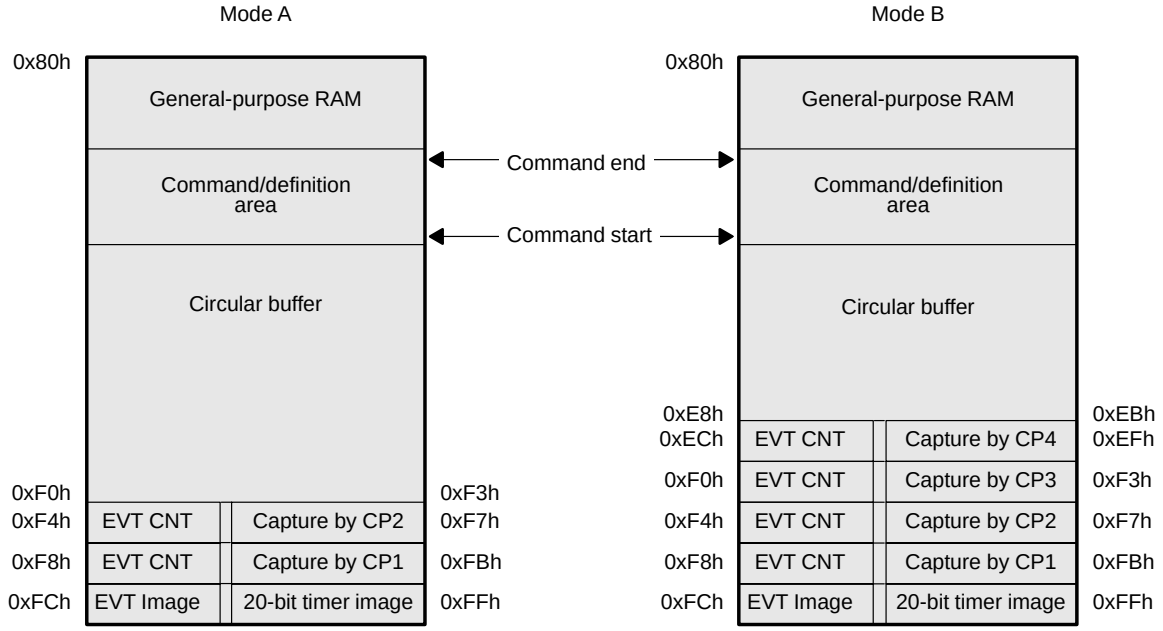
- **Capture registers.** One of the major differences between the PACT module and standard timers is the location of the 32-bit capture registers in dual-port RAM. These locations can be read from or written to by the CPU.

They are automatically written to by PACT when the appropriate feature is enabled. The capture registers reside between the circular buffer and the timer and event counter images. For more information about the input capture pins, refer to Section 15.4 on page 15-12.

The addresses 00FCh–00FFh (TMS370Cx32 devices) and 01FCh–01FFh (TMS370Cx36 devices) of the dual-port memory contain an image of the 20-bit default timer and an image of the 8-bit event counter. Since they are images or copies of the actual hardware registers, they can be overwritten by the application software. However, they are rewritten every time the PACT module receives another prescaled clock.

The dual-port RAM is 128 bytes long as shown in Figure 15–4.

Figure 15–4. Dual-Port RAM Organization†



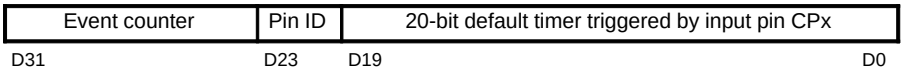
† For TMS370Cx32 devices, the dual-port RAM is located at addresses 0080h–00FFh. For TMS370Cx36 devices, the dual-port RAM is located at addresses 0180h–01FFh.

Mode A provides two, and Mode B provides four dedicated 32-bit storage locations, which follow a circular storage buffer. Modes A and B are described in Section 15.4 on page 15-12.

The PACT module uses memory starting from the highest address going to lower addresses. For the TMS370Cx32 devices, the highest address of the PACT module's dual-port RAM is 00FFh. For TMS370Cx36 devices, the highest address of the PACT module's dual-port RAM is 01FFh. The first 32-bit block always contains an image of the 20-bit default timer, a copy of the flag bits for capture pins 3 to 6, and an image of the 8-bit event counter. Thereafter, allocation depends on the mode selected.

15.4 Inputs

The PACT module has six input capture pins (CP1 through CP6) that cause data to be stored into fixed locations. Each location is defined by the pin that triggers the capture. When triggered directly from the pin, capture values are 32 bits long and consist of the 20-bit hardware timer, the 8-bit event counter, and four extra bits that identify the pin that caused the capture in the circular buffer:



The pin ID is set according to which input caused the capture. Only one of these bits will be set:

Bit	Transition on Pin	Result
D20 = 1	CP3	Causes the capture
D21 = 1	CP4	Causes the capture
D22 = 1	CP5	Causes the capture
D23 = 1	CP6	Causes the capture

Each input capture pin has a rising edge select bit, a falling edge select bit, and an interrupt enable bit, along with a fourth bit that acts as a flag to cause an interrupt. Table 15–3 lists these bits for each of the pins; the bits are described in Section 15.11 on page 15-36.

Table 15–3. Bits That Control Functions on the Input Capture Pins

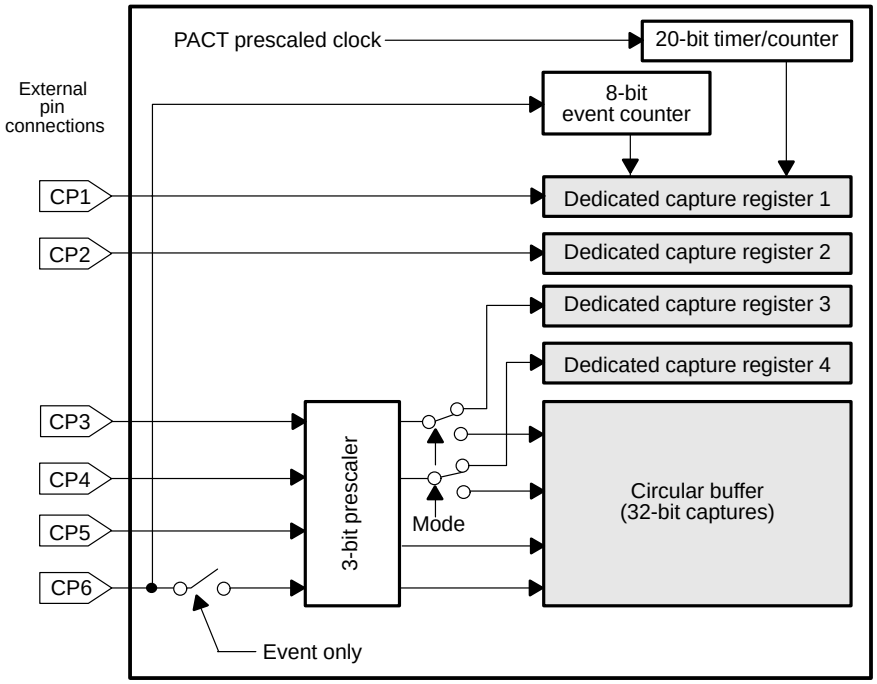
Pin	Rising Edge Select Bit	Falling Edge Select Bit	Interrupt Enable Bit	Interrupt Flag Bit
CP1	CP1 CAPT RISING EDGE (CPCTL1.1)	CP1 CAPT FALLING EDGE (CPCTL1.0)	CP1 INT ENA (CPCTL1.3)	CP1 INT FLAG (CPCTL1.2)
CP2	CP2 CAPT RISING EDGE (CPCTL1.5)	CP2 CAPT FALLING EDGE (CPCTL1.4)	CP2 INT ENA (CPCTL1.7)	CP2 INT FLAG (CPCTL1.6)
CP3	CP3 CAPT RISING EDGE (CPCTL2.1)	CP3 CAPT FALLING EDGE (CPCTL2.0)	CP3 INT ENA (CPCTL2.3)	CP3 INT FLAG (CPCTL2.2)
CP4	CP4 CAPT RISING EDGE (CPCTL2.5)	CP4 CAPT FALLING EDGE (CPCTL2.4)	CP4 INT ENA (CPCTL2.7)	CP4 INT FLAG (CPCTL2.6)
CP5	CP5 CAPT RISING EDGE (CPCTL3.1)	CP5 CAPT FALLING EDGE (CPCTL3.0)	CP5 INT ENA (CPCTL3.3)	CP5 INT FLAG (CPCTL3.2)
CP6	CP6 CAPT RISING EDGE (CPCTL3.5)	CP6 CAPT FALLING EDGE (CPCTL3.4)	CP6 INT ENA (CPCTL3.7)	CP6 INT FLAG (CPCTL3.6)

When you select the rising or falling edge, or both, the capture function for that pin is enabled, and the timer value captured is stored in the location that is determined by the mode of operation.

Two operating modes are available for the PACT module: mode A and mode B. You can select between these modes according to the capture functions that you need (refer to Figure 15–5):

- **Mode A** offers two dedicated capture locations (associated with pins CP1 and CP2) plus four other pins (CP3–CP6), each with a programmable prescaler to store 32-bit data in the circular buffer. The prescaler rate is the same for all of the four pins (CP3–CP6). Pin CP6 also clocks the 8-bit event counter.
- **Mode B** offers four dedicated capture locations (associated with pins CP1–CP4). Pins CP3–CP6 have a programmable prescaler. Pin CP5 can capture 32-bit data in the circular buffer when the software defined edge occurs. The remaining capture pin, CP6, clocks the 8-bit event counter and can capture 32-bit data in the circular buffer.

Figure 15–5. Input Capture Block Diagram



Captures can be set to occur on the falling, rising, or both edges of the input signal.

Capture pins CP3 through CP6 can be prescaled with a divide value from 1 to 8. Each of these four pins has its own edge counter, but the maximum count value (1– 8) before an actual capture occurs must be the same for all four pins.

Since it takes several system clock periods for the CPU to read a 20-bit timer capture value, an additional capture could occur while the original capture is being read. Your program can detect this situation by clearing the capture flag in the peripheral file before the read and then verifying that the flag has not been set again after the read is complete. If the flag was set again, the value read may be invalid and should be reread.

The buffer pointer (BUFPTR register) is available to tell the program when the last capture value was stored. You can modify the buffer pointer under processor control.

Note: Circular-Buffer Area Captures

The 16-bit captures in the circular buffer area are available when triggered by commands in the command/definition area. A 32-bit capture can overwrite the last 16-bit capture if the 16-bit capture is located at the two higher addresses (address bit1 = 1) of a 32-bit block.

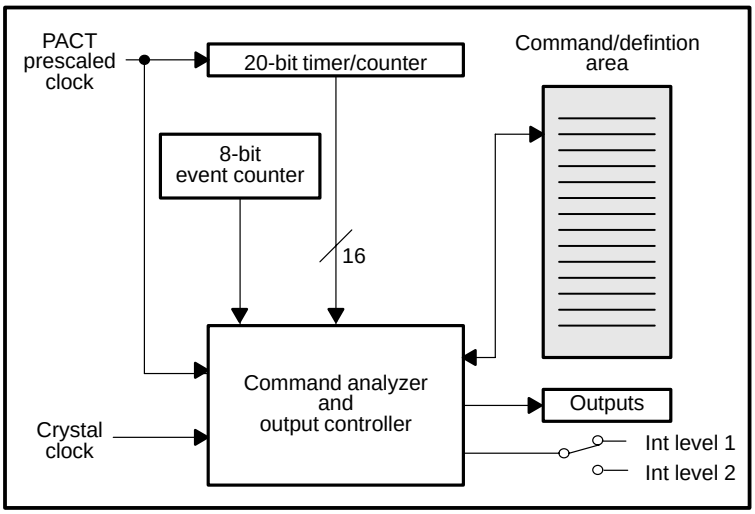
15.5 Control and Outputs

The control and outputs section of PACT is, perhaps, the most unique and most powerful part of this timer. Figure 15–6 shows the output control section of the PACT block diagram.

- The controller acts like a state machine and starts when it receives a rising edge from the PACT prescaled clock. The controller reads its commands (or state microcode) from the command definition part of the dual-port RAM.
- The 8-bit event counter and the 16 LSBs of the 20-bit default counter are also input into the controller for use in comparisons.
- The outputs from the controller set or clear the eight output pins (OP1–8).
- The prescaled clock from the PACT time base is used only to start the controller.

The controller steps through its commands, using the system clock phases for synchronization. The controller must step through all of the commands in the command/definition area before the next rising edge of the prescaled clock. The next prescaled clock increments the 20-bit default counter and restarts the whole process.

Figure 15–6. Output Control Section



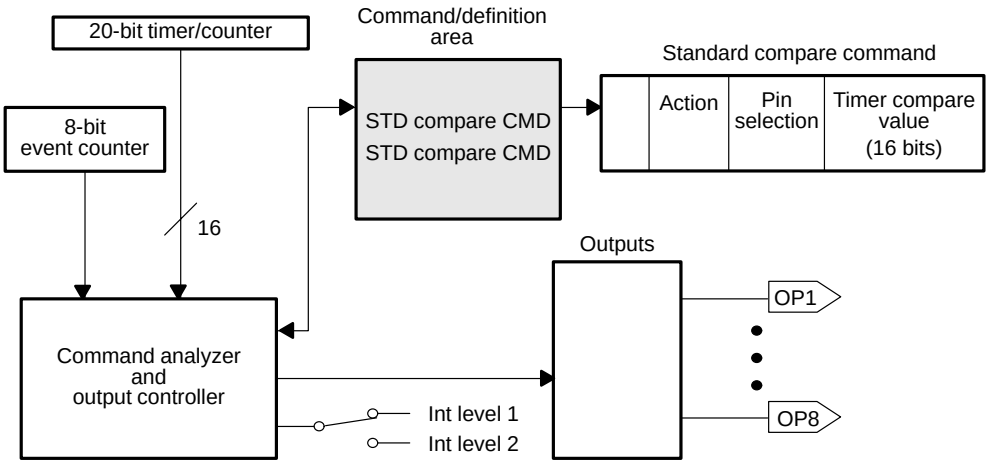
15.5.1 Standard Compare Command

To use the controller, you should understand the commands that it can execute. All of the commands or definitions in the command/definition area are 32 bits long. The simplest command is the standard compare command. The standard compare command sets or clears an output pin whenever the timer/counter is equal to a certain value. As shown in Figure 15–7, the standard compare command consists of the following:

- A 16-bit compare value
- Three bits to select one of the eight output pins
- Bits to select what action to take
- Bits to distinguish this command from the others

For more information or actual bit definitions, refer to subsection 15.6.4 on page 15-25.

Figure 15–7. Standard Compare Command



The standard compare command can:

- Set or clear the chosen output pin when the counter matches the compare value,
- Execute the opposite action (clear or set) when the 16 LSBs of the counter are equal to zero, or
- Generate an interrupt when the compare value is reached.

Therefore, you can make a pulse-width modulated (PWM) output of limited usefulness using a standard compare command. Assume that you want a PWM output with an initial duty cycle of 75 percent. Using the standard compare command, conduct the following:

- ❑ Set the timer compare value to 4000h (1/4 the overflow rate)
- ❑ Set the actions to cause an output pin to go high when the count is equal to the compare value, and then low again when the 16 LSBs of the counter are zero.
- ❑ Vary the duty cycle by changing the 16-bit compare value.
- ❑ Invert the signal by selecting clear on compare equal, as opposed to set on compare equal.

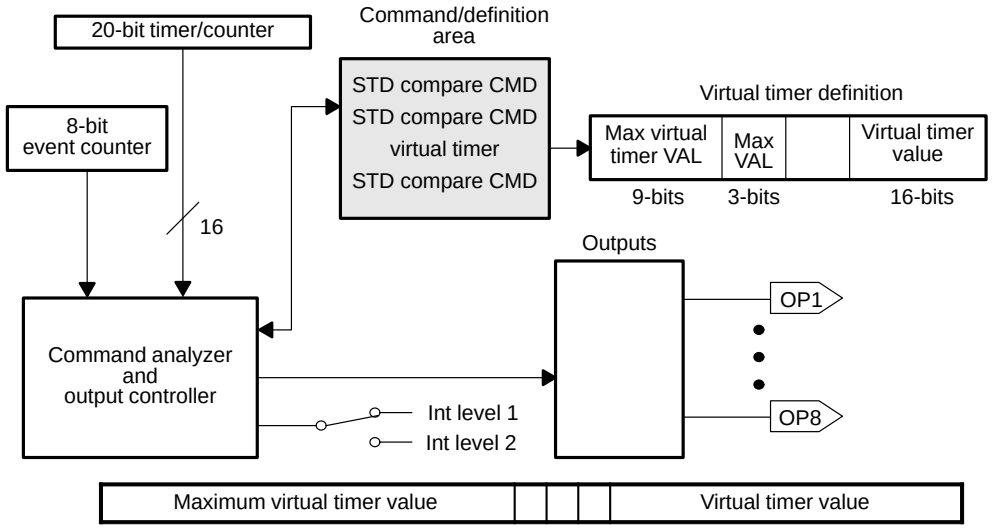
You cannot use this command to vary the period of the PWM.

15.5.2 Virtual Timers

You can vary the period of the PWM by using a virtual timer. Remember that the command/definition area is implemented in RAM. Figure 15–8 shows the virtual timer definition and its implementation. The virtual timer definition consists of the following:

- ❑ 16 bits that are read, incremented, and rewritten on each tic of the PACT clock.
- ❑ 13 bits that define a maximum value. When the virtual timer reaches this maximum value, it is reset to zero.

Figure 15–8. Virtual Timer Implementation



For more information about the actual bit definitions, refer to subsection 15.6.1 on page 15-22.

The command/definition area in Figure 15–8 shows two standard compare commands, a virtual timer definition, and a third standard compare command. Assume that you are using a microcontroller with a 200-ns (5-MHz) internal system clock and that you are prescaling the PACT clock with divide by five so that each PACT clock tic is one microsecond.

- The first two standard compare commands generate PWM signals of variable duty cycle with a period of 65 536 prescaled clock tics (65.536 ms).
- If you want the third PWM to have a period of one millisecond, set up the virtual timer with a maximum value of 1000.
- When the controller sees the timer definition, it increments the virtual timer and then uses the virtual timer value for future comparisons.
- The third standard compare command generates a PWM of variable duty cycle with a period of one millisecond.

You can use combinations of standard compare commands and virtual timers to create complex repeating waveforms. Multiple standard compare commands can be used on a single output pin to create multiple pulses of different duration.

You can use virtual timers to provide periodic interrupts to the processor.

15.5.3 Double Event Compare Command

Actions can also be taken as determined by comparisons to the 8-bit event counter. Since all commands are 32 bits wide, the double event compare command actually defines two event compare values and the actions that can be performed based on each value. The actions that are allowed according to the event 1 compare value matching the event counter are as follows:

- Setting or resetting the selected output pin (OP1–OP8),
- Generating an interrupt, or
- Generating a 32-bit capture into the circular buffer.

The actions that are allowed according to the event 2 compare value matching the event counter are as follows:

- Setting or resetting the selected output pin (OP1–OP8),
- Generating an interrupt,
- Generating a 32-bit capture into the circular buffer, or
- Resetting the 20-bit default timer.

Because of synchronization, these actions occur two or three prescaled clock cycles after the input edge that incremented the event counter. A block diagram of the double-event compare command is shown below. This diagram shows the information contained in the command. For more information or actual bit definitions, refer to subsection 15.6.5 on page 15-26.

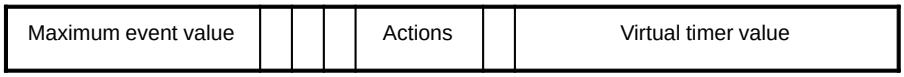
So far, you can manipulate output lines depending on time values or the number of external events. An additional virtual timer definition allows you to manipulate output lines according to a combination of the event counter and time.

15.5.4 Offset Timer Definition—Time From the Last Event

The offset timer definition—time from the last event—creates a 16-bit virtual timer that is cleared on each occurrence of an event on pin CP6. This definition also sets an event counter maximum, so that the event counter is reset after reaching this maximum value. The offset timer definition can perform the following actions:

- ❑ Generate an interrupt when the maximum event count is reached,
- ❑ Store the 16-bit virtual timer in the circular buffer on each event,
- ❑ Store the 20-bit default timer and 8-bit event counter in the circular buffer when the maximum event count is reached, or
- ❑ Reset the 20-bit hardware default timer when the maximum event count is reached.

A block diagram of the offset timer definition is shown below. This diagram shows the information contained in the command. For more information or actual bit definitions, refer to subsection 15.6.2 on page 15-23.



15.5.5 Conditional Compare Command

A special compare command, conditional compare, has a timer compare value and an event compare value. Both of these values must match for the defined action to take place. Usually, a series of these commands follows an offset-timer definition time from the last event, and provides output pulses on different pins, based on the event count and an elapsed time from the event. The conditional compare command generates the following actions:

- Generates an interrupt when both the following two conditions are met:
  - The event compare value equals the event counter.
  - The timer compare value equals the last defined timer.
- Sets or clears one of seven output pins (OP1–OP7) when the following two conditions are met:
  - The event compare value equals the event counter.
  - The timer compare value equals the last defined timer.

The actions described above can be enabled on the event counter reaching the event compare value plus one, regardless of the timer compare value. This allows for when the next event occurs before the delay period (specified by the timer compare value) is reached.

A block diagram of the conditional compare command is shown below. This diagram shows the information contained in the command. For more information or actual bit definitions, refer to subsection 15.6.6 on page 15-28.

15.5.6 Baud Rate Timer Definition

The last item that can be put into the command/definition area of the PACT module is a baud rate virtual timer. This virtual timer runs the serial communications port built into the PACT module. Set up the maximum timer value for one-quarter bit period of the desired bit rate. Separate timers can be defined for transmit and receive. For more information on the baud rate timer definition, see subsection 15.6.3 on page 15-24. For more information about the SCI, see subsection 15.9 on page 15-32.

15.6 Command/Definition Area

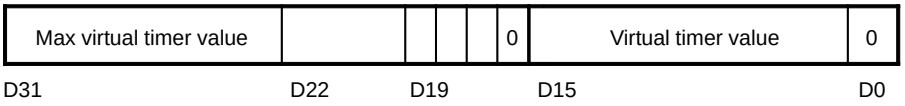
All commands/definitions are 32 bits long. They are stored in memory with the most significant byte (MSbyte) first. If byte 3 is stored at location N, then byte 0 would be at location N+3. The bits are referenced as D0–D31.

This section summarizes the available commands and the number of time slots required for each command.

Definitions	**Time Slots**
Virtual timer definition	2
Offset timer definition	2/3
Baud rate timer definition	2

Commands	**Time Slots**
Standard Compare command	1
Conditional Compare command	1
Double Event Compare command	1

15.6.1 Virtual Timer Definition



Requires two time slots.

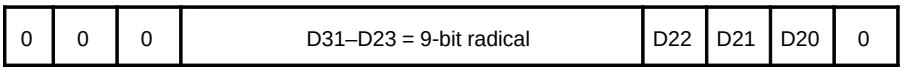
D31–23 Sets the radical of the maximum value of the virtual timer. Used with D20–D22 to specify the maximum value of the virtual timer. The maximum virtual timer value is equal to the desired period minus 2. For example, if you want a timer with a period of 100 PACT prescaled clocks, set the maximum virtual timer value to 98. The virtual timer increments from 0 to 99 and is then reset to 0.

D22–20 Define a further three bits of the maximum value of the virtual timer. Either the MSBs (bits 15–13) of the maximum value if the range bit = 1, or the LSBs (bits 3–1) if the range bit = 0. The undefined bits of the maximum value for the virtual timer are set to 0.

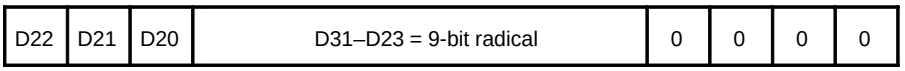
D19 Range Bit

Used in conjunction with D22–D20 to define the maximum value (see illustration below).

Maximum Value Format (D19=0)



Maximum Value Format (D19=1)



D18 Enable—Active = 1

Enables the timer update, and stops or starts the timer.

D17 Interrupt on 0—Active = 1

Sets the interrupt flag when the virtual timer (D15–D1) overflows to zero.

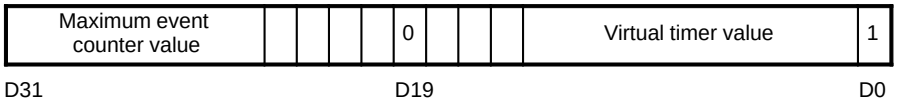
D16 This bit must be written as a 0 for this command to be valid.

D15–1 Virtual Timer Value

These are the 15 MSBs of a 16-bit virtual timer. This timer is resident at this location, so any write to this address by the CPU modifies the timer value. Because of hardware limitations, the LSB of the virtual timer cannot be read from or written to by the CPU, but it is used by PACT commands such as the Standard Compare command.

D0 This bit must be written as a 0 for this definition to be valid.

15.6.2 Offset Timer Definition—Time From Last Event



Requires two time slots if bit D21 = 0; requires three time slots if bit D21 = 1.

D31–24 Event Maximum Value

Specifies a maximum for the event counter. Upon reaching this value, the event counter will be reset to zero by the next event on CP6.

D23 Interrupt on Event—Active = 1

Sets the interrupt flag when an event on pin CP6 occurs.

D22 Default Capture—Active = 1

Captures 32-bit data into the circular buffer when the event counter reaches the maximum value (D31–D24).

D21 Virtual Capture—Active = 1

Stores the 16-bit virtual offset timer (defined by this definition) in the circular buffer on every event on CP6 before it is cleared.

D20 Reset Default Timer—Active = 1

Clears the default timer when the event counter reaches the maximum value (D31–D24).

D19 This bit must be written as a 0 for this definition to be valid.

D18 Enable—Active = 1

Enables the timer update. Used to stop and start the timer.

D17 Interrupt on Maximum Event—Active = 1

Sets the interrupt flag when the event counter reaches the maximum value (D31–D24).

D16 Step—Active = 1

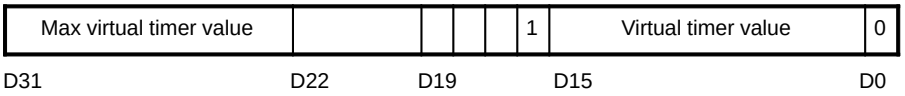
Allows lower resolution on the following commands in this definition area (see subsection 15.2.5, on page 15-7, for details on the use of this function).

D15–1 Virtual Timer Value

These are the 15 MSBs of a 16-bit virtual timer. This timer is resident at this location, so any write to this address by the CPU modifies the timer value. Because of hardware limitations, the LSB of the virtual timer cannot be read from or written to by the CPU, but it is used by the PACT commands such as the standard compare command.

D0 This bit must be written as a 1 for this definition to be valid.

15.6.3 Baud Rate Timer Definition



Requires two time slots.

D31–23 Set the radical of the maximum value of the virtual timer. Used with D22–D20 to specify the maximum value of the virtual timer. When the virtual timer reaches the defined value, the next prescaler clock cycle causes the timer to be cleared. The maximum virtual timer value should be set to one quarter of the desired bit time minus 2. For more details, see Section 15.9 on page 15-32.

D22–20 Define a further three bits of the maximum value of the virtual timer. Either the MSBs (bits 13–15) of the maximum value if range bit = 1, or the LSBs (bits 3–1) if range bit = 0. The undefined bits of the maximum value for the virtual timer are set to 0.

D19 Range Bit

Used in conjunction with D20–D22 (see the illustration below).

Maximum Value Format (D19 = 0)

0	0	0	D31–D23 = 9-bit radical	D22	D21	D20	0

Maximum Value Format (D19 = 1)

D22	D21	D20	D31–D23 = 9-bit radical	0	0	0	0

D18 RXselect—Active = 1

Selects this timer definition to use for the receive baud-rate generator.

D17 TXselect—Active = 1

Selects this timer definition to use for the transmit baud-rate generator.

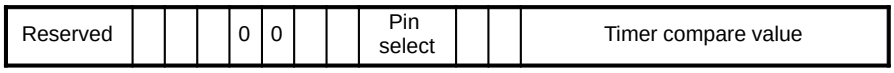
D16 This bit must be written as a 1 for this definition to be valid.

D15–1 Baud Rate Timer

These are the 15 MSBs of a 16-bit virtual timer used as the baud-rate generator. The timer is resident at this location, so any write to this address modifies its value.

D0 This bit must be written as a 0 for this definition to be valid.

15.6.4 Standard Compare Command



Requires one time slot.

D31–D28 Reserved.

D27 Enable Pin—Active = 1

Enables output pin actions on this command.

D26 Interrupt on Reset—Active = 1

Sets the interrupt flag when the referred timer is reset to zero.

D25 Reset Action

Sets or resets the pin defined by D20–D18 when the referred timer is reset to zero.

0 = No action when the referred timer is zero.
1 = When the referred timer is zero, execute the opposite action.

D24–23 These bits must be both written as 0s for this definition to be valid.

D22 Step—Active = 1

Allows lower resolution on the following commands in this definition area (see subsection 15.2.5, on page 15-7, for details on the use of this function).

D21 Compare Action—Set = 1, Clear = 0

Sets or resets the pin defined by D20–D18 when the compare value is matched by the referred timer.

D20–18 Pin Selection

Select an output pin whose state is modified when the compare value is matched. The pin number is the binary value of D20–D18 (20 = MSB,18 = LSB) plus one.

D17 Interrupt on Compare—Active = 1

Sets the interrupt flag when the compare value is matched by the referred timer.

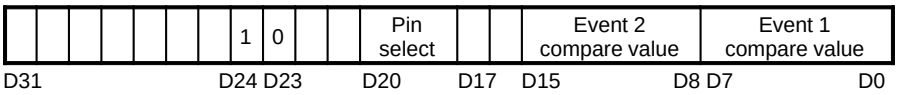
D16 Next Command Is a Definition—Active = 1

Indicates that the following entry in the command/definition area is a definition.

D15–0 Timer Compare Value

These 16 bits provide a timer-compare value. The timer, to which this value is compared, is either the last virtual timer defined above this command or, if no virtual timer has been defined, the default timer.

15.6.5 Double Event Compare Command



Requires one time slot.

D31 Reserved.

D30 Event 2 Default Timer Capture—Active = 1

Stores a 32-bit data capture in the circular buffer when event 2 occurs.

D29 Event 1 Default Timer Capture—Active = 1

Stores a 32-bit data capture in the circular buffer when event 1 occurs.

D28 Event 2 Default Timer Reset—Active = 1

Resets the default timer when event 2 occurs.

D27 Enable pin—Active = 1

Enables the output pin actions for this command.

D26 Interrupt on Compare 2—Active = 1

Sets the interrupt flag when event 2 occurs.

D25 Compare Action 2—No Action = 0 Inverted Action = 1

Sets or resets the pin defined by pin selection when the event 2 compare value (D8–D15) is matched by the event counter.

D24 This bit must be written as a 1 for this command to be valid.

D23 This bit must be written as a 0 for this command to be valid.

D22 Step—Active = 1

Allows lower resolution on the following commands (see subsection 15.2.5, on page 15-7, for details on the use of this function).

D21 Compare Action 1—Set = 1, Clear = 0

Sets or resets the pin defined by pin selection when the event 1 compare value (D7–D0) is matched by the event counter.

D20–18 Pin Selection

Selects an output pin whose state is modified when the compare value is matched. The pin number is the binary value of D20–D18 (20 = MSB,18 = LSB) + 1 (OP1–OP8).

D17 Interrupt on Compare 1—Active = 1

Sets the interrupt flag when the event 1 compare value is matched by the event counter.

D16 Next Command Is a Definition—Active = 1

Indicates that the following entry in the command/definition area is a definition.

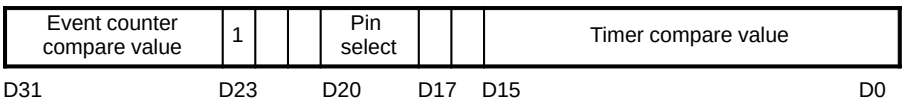
D15–8 Event 2

Sets an 8-bit value that, when matched by the 8-bit event counter, causes the action defined by D25, D26, D28, and D30.

D7–0 Event 1

Sets an 8-bit value that, when matched by the 8-bit event counter, causes the action defined by D17, D21, and D29.

15.6.6 Conditional Compare Command



Requires one time slot.

D31–24 Event Compare Value

Sets an 8-bit value that is compared with the 8-bit event counter. The actions selected by this command occur under either of the following conditions:

- ❑ The event compare value matches the value of the event counter, *and* the timer compare value matches the referred timer value, **or**
- ❑ The same action active bit is set, *and* the event counter matches the event compare value + 1.

D23 This bit must be written as a 1 for this command to be valid.

D22 Same Action—Active = 1

Indicates the same action as compare action when the event counter reaches the event compare value plus one. This allows an action on the next event if the next event occurs before the time value is reached. If Same Action = 0, then there will be no action on event compare + 1.

D21 Compare Action—Set = 1, Clear = 0

Sets or resets the pin defined by pin selection when both compare values are matched by the referred timer and the event counter.

D20–18 Pin Selection

Select an output pin whose state will be modified when the compare value is matched. The pin number is the binary value of D20–D18 (20 = MSB,18 = LSB) plus one, except the binary value 111, which disables any pin action. Therefore, OP8 is not available for this command.

D17 Interrupt on Compare—Active = 1

Sets the interrupt flag when the timer compare value (D0–D15) is matched by the referred timer, *and* the event compare value (D24–D31) is matched by the event counter.

D16 Next Command is a Definition Active = 1

Indicates that the following entry in the command/definition area is a definition.

D15–0 Timer Compare Value

These 16 bits provide a timer compare value. The timer, to which this value is compared, is either the last virtual timer defined above this command or, if no virtual timer has been defined, the default timer. This is called the referred timer. The value that you write *must* be greater than one.

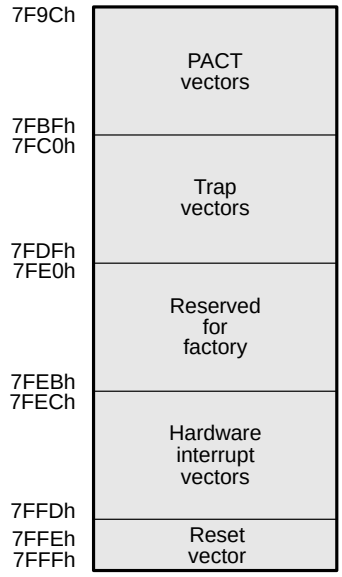
15.7 Interrupts

This section discusses interrupts that are specific to the PACT module. There are three groups of interrupt vectors.

- The first group is associated with the events on a particular capture pin.
- The second group is associated with the SCI interrupts, such as the receive buffer full.
- The third group of interrupts is associated with the absolute position of the command or definition within the RAM area.

The 18 vectors available for PACT functions are located immediately after the trap vectors in the TMS370 address space. Refer to the memory map in Figure 15–9.

Figure 15–9. Interrupt Vector Memory Map



As is standard in the TMS370, two levels of priority (1 and 2) exist for each of the three groups of interrupts described above. These interrupt groups can be allocated to one of two interrupt levels:

- A priority that determines the order in which multiple interrupts within a level are serviced (see Table 15–4).
- An order for servicing groups on the same level (see Table 5–2, page 5-5).

Interrupts are enabled either in peripheral frame 4 or within the command/definition line. The service routine must clear the flag associated with the interrupt to prevent multiple servicing of the same interrupt.

Table 15–4. Interrupt Vector Sources

Module	Vector Address	Interrupt Source	Interrupt Flag	System Interrupt	Priority in Group†
PACT	7F9Ch, 7F9Dh	PACT SCI TXINT	PACT TXRDY	PTXINT	2
(Group 2)	7F9Eh, 7F9Fh	PACT SCI RXINT	PACT RXRDY	PRXINT	1
PACT	7FA0h, 7FA1h	PACT Cmd/Def Entry 0	CMD/DEF INT 0 FLAG	CDINT0	1
(Group 3)	7FA2h, 7FA3h	PACT Cmd/Def Entry 1	CMD/DEF INT 1 FLAG	CDINT1	2
	7FA4h, 7FA5h	PACT Cmd/Def Entry 2	CMD/DEF INT 2 FLAG	CDINT2	3
	7FA6h, 7FA7h	PACT Cmd/Def Entry 3	CMD/DEF INT 3 FLAG	CDINT3	4
	7FA8h, 7FA9h	PACT Cmd/Def Entry 4	CMD/DEF INT 4 FLAG	CDINT4	5
	7FAAh, 7FABh	PACT Cmd/Def Entry 5	CMD/DEF INT 5 FLAG	CDINT5	6
	7FACh, 7FADh	PACT Cmd/Def Entry 6	CMD/DEF INT 6 FLAG	CDINT6	7
	7FAEh, 7FAFh	PACT Cmd/Def Entry 7	CMD/DEF INT 7 FLAG	CDINT7	8
PACT (Group 1)	7FB0h, 7FB1h	PACT Circular Buffer (Half/ Full)	BUFFER HALF/FULL INT FLAG	BUFINT	1
	7FB2h, 7FB3h	PACT CP6 Edge	CP6 INT FLAG	CP6INT	2
	7FB4h, 7FB5h	PACT CP5 Edge	CP5 INT FLAG	CP5INT	3
	7FB6h, 7FB7h	PACT CP4 Edge	CP4 INT FLAG	CP4INT	4
	7FB8h, 7FB9h	PACT CP3 Edge	CP3 INT FLAG	CP3INT	5
	7FBAh, 7FBBh	PACT CP2 Edge	CP2 INT FLAG	CP2INT	6
	7FBCh, 7FBDh	PACT CP1 Edge	CP1 INT FLAG	CP1INT	7
	7FBEh, 7FBFh	PACT Default Timer Overflow	DEFTIM OVRFL INT FLAG	POVRFL INT	8

† 1 is the highest priority.

Notes:

1) CP1–6 interrupts are caused by the software edge(s) selected for that particular pin in peripheral frame 4. Also, the associated interrupt enable bit in frame 4 must be set for that particular pin. Interrupts set in a command/definition line relate to their position in that area.

2) The entry address is derived from bits 2, 3, and 4 of the address. If the command or definition is at address 006Ch, 006Dh, 006Eh, or 006Fh (32 bits), then 0110 11xx is the binary value of the address. The entry address comes from bits 4–3–2 = 011 = Entry 3. Thus, the vector associated with entry address 3 is used when this command or definition causes an interrupt.

For command/definition areas that contain more than eight entries, the entry vectors become overlaid, and the program must determine the correct source. Entry 0 has the same vector as entry 8, so the command/definition at addresses 04Ch, 04Dh, 04Eh, or 04Fh has the same interrupt vector as the command/definition located at addresses 06Ch, 06Dh, 06Eh, or 06Fh; both are entry 3.

15.8 WD Timer

At powerup, the WD timer is enabled with the shortest time-out period (bit 9 of default timer).

A WD-originated reset is generated when a software-selected bit of the default timer toggles. Three options determine the WD time-out period and a disable WD code. These options are specified according to how bits 0 and 1 of the global function control register (PACTPRI) are configured:

PACT WD PRES-CALE SELECT 1 (PACTPRI.1)	PACT WD PRES-CALE SELECT 0 (PACTPRI.0)	Option Selected
0	0	Reset when bit 9 of default timer toggles
0	1	Reset when bit 15 of default timer toggles
1	0	Reset when bit 19 of default timer toggles
1	1	Disable WD

These bits are described in subsection 15.11.14, on page 15-59. They are available only in privilege mode immediately after powerup.

Once a time-out period has been selected as shown above, the alternate key bytes, 55h (first) and AAh must be written to the WDRST register (peripheral frame 4, 104Eh) to avoid issuing a WD-originated reset. The only exception to this occurs when the default counter is cleared by the PACT module. In this case, a WD-originated reset occurs, unless the correct keyword (55h/AAh) has been written since the previous clear.

The WD timer is stopped in standby mode and halt mode.

15.9 Mini-Serial Communications Interface (SCI)

The mini-SCI works as a simplified full duplex UART by transmitting 8-bit words with a fixed format of one start bit and one stop bit.

- If parity transmission is required, the parity bit must be calculated by the CPU and placed in the transmit buffer as part of the 8-bit word. Parity reception is facilitated by the parity result bit. This bit allows the processor to check for parity errors by comparing the PACT PARITY bit (SCICTLP.5) against 0 or 1 for even or odd parity. Hence, there is no parity error bit to be checked by the processor.
- There is no overrun detection. The PACT SCI has a shift register and a buffer register. This gives the program a full data byte reception time to read the previous byte before it is overwritten.

During reception, the start bit is detected on the falling edge and then sampled again in the center of the bit to avoid false detection. All other bits are sampled once at their centers. If at least one stop bit is not detected when it is expected, the framing error flag (PACT FE bit, P045.3) is set. This bit remains set until cleared by reset, by SCI software reset, or by writing a zero to it.

The bit rate (baud) is determined by setting the maximum virtual timer value to one quarter of the desired bit time minus 2. For example, if the system clock period is 200 ns and the prescale value is 5, then the PACT resolution is 1 µs. If a baud of 9600 is desired, the maximum virtual timer value should be:

$$\text{Max Virtual Timer Value} = \frac{1}{(\text{Baud})\ (4)\ (\text{PACT Resolution})^{*}} - 2$$

$$= \frac{1}{(9600)\ (4)\ (10^{-6})} - 2$$

$$= 24$$

*Where PACT Resolution = SYSCLK x Prescale Value

The software selectable bauds (RX and TX can be different) are set up as shown in subsection 15.6.3 on page 15-24. Receive and transmit operations can be stopped or started by using the control bits within the baud rate definition command.

The data being received and transmitted is accessed in the same peripheral frame (4) as the control bits are. Received data is held at 1046h; transmitted data at 1047h.

15.10 PWM Example

The following three-part routine provides an example of how to set up a pulse-width-modulated signal with the PACT module:

- The first part defines the bytes that make up the command/definition area.
- The second part copies the command/definition area bytes from ROM to the dual-port RAM.
- The third part sets up the PACT peripheral file.

Refer to Example 15–1, on page 15-34, as you read the following subsections.

15.10.1 Defining the Command/Definition Area

The macro file PACT.H (Example 15–1, page 15-34) simplifies the task of setting up the command/definition area. (Appendix I of this manual further describes PACT.H macros.) Since this file is subject to change as improvements are found, TI recommends that you download the latest version of this file from the microcontroller bulletin board.

To set up the PWM signal,

- A virtual timer definition must establish the timer period.
- A standard compare command must follow to determine the period and the polarity of the signal.
- An additional standard compare command must be inserted at the beginning with only the D16 bit set. This is because the PACT command/definition area cannot start with a definition.
- Line 8 of the routine causes the bytes that will become the commands and definitions to be located in a separate section. In this example, this section starts at location 7800h.
- Line 10 is the dummy standard compare command.
- Line 11 defines the virtual timer. The period is set to 1000 µs, and the virtual timer enabled. Note that the macro takes care of subtracting two from the maximum count value as it creates the proper byte sequence.
- Line 12 is the standard compare command that sets the period to 800 µs or 80%, and selects output pin 1. The default value sets the pin high on a compare equal and opp_act is selected to cause the pin to go low when the timer is reset. Notice how multiple actions are concatenated with the | operator in this command.

Example 15–1. Routine to Perform a PWM FOR ’X32

```
0001   ;This is an example program to do PWM using the PACT module
0002           .include “PACT.H”
0003
0004   ;MACRO DESCRIPTION
0005   ;stdcmp <compare value>,<pin>,<actions>
0006   ;virtmr <period>,<actions>,<initial timer value>
0007
0008           .sect “pact”,7800h
0009   ;PACT instructions to do PWM
0010   table   stdcmp  0,0,nxt_def           ;dummy cmd, next line = def
0001   #       .byte   0,0,1,0
0011       virtmr  1000,enable               ;period = 1000 uSec
0001   #       .byte   0,0,52,31
0012       stdcmp  800,op1,opp_act|enable    ;80% duty, pin 1
0001   #       .byte   32,3,0,10
0013   len     .equ    $–table
0014
0015           .text   6000h
0016   cmd_st  .equ    0EBh                  ;for ’x36 device, replace
                                               0EBh with 1EBH
0017   start   mov #7,P04F                   ;disable the WD
0018   ;copy PACT commands/def. into ram
0019           mov     #len,b                ;length of cmd/def area
0020           movw    #(cmd_st–len+1),r3    ;R2:R3 points to area
0021   loop    mov     table–1[b],a
0022           mov     a,*r3
0023           inc     r3
0024           djnz    b,loop
0025
0026   ;set up the peripheral file
0027           mov     #07,p04f              ;set to mode B
0028           mov     #cmd_st,p041          ;cmd/def start at 0ebh
0029           mov     #(cmd_st–len+1),p042  ;cmd/def end = 0E0h
0030   ;set prescale to 5, 1 usec res, enable cmd/def area
0031           mov     #034h,p040
0032
0033   ;PWM running without processor intervention
0034       idle
0035           .end
```

15.10.2 Copying the Command/Definition Area to Dual-Port RAM

Since the dual-port RAM must be initialized after powerup, and the initial values for the command/definition area were defined in nonvolatile memory; they must be copied from the nonvolatile memory to the dual-port RAM. Since the PACT module works through memory from high addresses to lower addresses, and the assembler works through memory from low addresses to higher addresses, this routine flips the memory table as it is copied into the dual-port RAM. This makes the table easier to read.

Two variables must be defined before this routine is used.

- The first variable, cmd_st, is the start of the command/definition area (the largest address in that area). Its value is dependent on the mode used and on the size of the desired circular buffer. cmd_st is defined in line 16 of this routine (Example 15–1).
- The second variable, len, is defined in line 13 as the number of bytes in the command/ definition area.

15.10.3 Initializing the PACT Peripheral Frame

The last part of the PWM routine sets up the PACT peripheral frame.

1) Line 27 chooses the mode (Example 15–1).
2) Line 28 chooses the command/definition start, and line 29 chooses the command/definition end.
3) The timer resolution is set to one microsecond, and the command/definition area is enabled in line 31. This line causes the PWM signal to start.

You should always verify that the PACT clock prescale value allows enough time slots for the entire command/definition area. In this example, four time slots are required—one for each standard compare command and two for the virtual timer definition. The prescale value of 5 provides 12 time slots, which is more than enough for this application.

15.11 PACT Control Registers

The PACT module is controlled and accessed through registers in peripheral frame 4. These registers are listed in Figure 15–10 and described in the following subsections. The bits shown in shaded boxes in Figure 15–10 are privilege mode bits; they can be written to only in the privilege mode.

Note: Inadvertent Modification of an Interrupt Flag

Be careful using the AND, OR, XOR, CMPBIT, SBIT0, or SBIT1 instructions to modify any registers that contain status flags. The read/modify/write nature of these instructions can inadvertently clear an interrupt flag that was set between the read and the write cycles. If the state of the nonflag bits is known, use the MOV #iop8,Pd instruction. If the state of the non-flag bits is not known, use a sequence similar to the example shown below.

```
; Clearing an interrupt flag
    MOV   P04n,A
    OR    #flag_mask,A     ; set all flag bits to one
    AND   #desired_flag,A  ; clear the desired flag bit
    MOV   A,P04n
```

Figure 15–10. Peripheral File Frame 4: PACT Control Registers

Desig-nation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PACTSCR	1040h	P040	DEFTIM OVRFL INT ENA (RW–0)	DEFTIM OVRFL INT FLAG (RC–0)	CMD/DEF AREA ENA (RW–0)	FAST MODE SELECT (RP–0)	PACT PRESCALE SELECT 3 (RP–0)	PACT PRESCALE SELECT 2 (RP–0)	PACT PRESCALE SELECT 1 (RP–0)	PACT PRESCALE SELECT 0 (RP–0)
CDSTART	1041h	P041	CMD/DEF AREA INT ENA (RW–0)	—	CMD/DEF AREA START BIT 5 (RW–0)	CMD/DEF AREA START BIT 4 (RW–0)	CMD/DEF AREA START BIT 3 (RW–0)	CMD/DEF AREA START BIT 2 (RW–0)	—	—
CDEND	1042h	P042	—	CMD/DEF AREA END BIT 6 (RW–0)	CMD/DEF AREA END BIT 5 (RW–0)	CMD/DEF AREA END BIT 4 (RW–0)	CMD/DEF AREA END BIT 3 (RW–0)	CMD/DEF AREA END BIT 2 (RW–0)	—	—
BUFPTR	1043h	P043	1 (R–1)	1 (R–1)	BUFFER POINTER BIT 5 (R–1)	BUFFER POINTER BIT 4 (R–1)	BUFFER POINTER BIT 3 (R–0)	BUFFER POINTER BIT 2 (R–0)	BUFFER POINTER BIT 1 (R–0)	—
	1044h	P044	Reserved							
SCICTLP	1045h	P045	PACT RXRDY (RC–0)	PACT TXRDY (R–1)	PACT PARITY (R–0)	PACT FE (RC–0)	PACT SCI RX INT ENA (RW–0)	PACT SCI TX INT ENA (RW–0)	—	PACT SCI SW RESET (RW–0)
RXBUFP	1046h	P046	PACT RXDT7 (R–0)	PACT RXDT6 (R–0)	PACT RXDT5 (R–0)	PACT RXDT4 (R–0)	PACT RXDT3 (R–0)	PACT RXDT2 (R–0)	PACT RXDT1 (R–0)	PACT RXDT0 (R–0)
TXBUFP	1047h	P047	PACT TXDT7 (RW–0)	PACT TXDT6 (RW–0)	PACT TXDT5 (RW–0)	PACT TXDT4 (RW–0)	PACT TXDT3 (RW–0)	PACT TXDT2 (RW–0)	PACT TXDT1 (RW–0)	PACT TXDT0 (RW–0)
OPSTATE	1048h	P048	PACT OP8 STATE (RW–0)	PACT OP7 STATE (RW–0)	PACT OP6 STATE (RW–0)	PACT OP5 STATE (RW–0)	PACT OP4 STATE (RW–0)	PACT OP3 STATE (RW–0)	PACT OP2 STATE (RW–0)	PACT OP1 STATE (RW–0)
CDFLAGS	1049h	P049	CMD/DEF INT 7 FLAG (RC–0)	CMD/DEF INT 6 FLAG (RC–0)	CMD/DEF INT 5 FLAG (RC–0)	CMD/DEF INT 4 FLAG (RC–0)	CMD/DEF INT 3 FLAG (RC–0)	CMD/DEF INT 2 FLAG (RC–0)	CMD/DEF INT 1 FLAG (RC–0)	CMD/DEF INT 0 FLAG (RC–0)
CPCTL1	104Ah	P04A	CP2 INT ENA (RW–0)	CP2 INT FLAG (RC–0)	CP2 CAPT RISING EDGE (RW–0)	CP2 CAPT FALLING EDGE (RW–0)	CP1 INT ENA (RW–0)	CP1 INT FLAG (RC–0)	CP1 CAPT RISING EDGE (RW–0)	CP1 CAPT FALLING EDGE (RW–0)
CPCTL2	104Bh	P04B	CP4 INT ENA (RW–0)	CP4 INT FLAG (RC–0)	CP4 CAPT RISING EDGE (RW–0)	CP4 CAPT FALLING EDGE (RW–0)	CP3 INT ENA (RW–0)	CP3 INT FLAG (RC–0)	CP3 CAPT RISING EDGE (RW–0)	CP3 CAPT FALLING EDGE (RW–0)
CPCTL3	104Ch	P04C	CP6 INT ENA (RW–0)	CP6 INT FLAG (RC–0)	CP6 CAPT RISING EDGE (RW–0)	CP6 CAPT FALLING EDGE (RW–0)	CP5 INT ENA (RW–0)	CP5 INT FLAG (RC–0)	CP5 CAPT RISING EDGE (RW–0)	CP5 CAPT FALLING EDGE (RW–0)
CPPRE	104Dh	P04D	BUFFER HALF/FULL INT ENA (RW–0)	BUFFER HALF/FULL INT FLAG (RC–0)	INPUT CAPT PRESCALE SELECT 3 (RW–0)	INPUT CAPT PRESCALE SELECT 2 (RW–0)	INPUT CAPT PRESCALE SELECT 1 (RW–0)	CP6 EVENT ONLY (RW–0)	EVENT COUNTER SW RESET (RW–0)	OP SET/CLR SELECT (RW–0)
WDRST	104Eh	P04E	WD Reset Key							
PACTPRI	104Fh	P04F	PACT STEST (RP–0)	—	PACT GROUP 1 PRIORITY (RP–0)	PACT GROUP 2 PRIORITY (RP–0)	PACT GROUP 3 PRIORITY (RP–0)	PACT MODE SELECT (RP–0)	PACT WD PRESCALE SELECT 1 (RP–0)	PACT WD PRESCALE SELECT 0 (RP–0)

15.11.1 Setup Control Register (PACTSCR)

The PACTSCR register determines the time base for the PACT module, enables the command/definition area, and controls the default timer overflow.

Setup Control Register (PACTSCR)
[Memory Address 1040h]

Bit #	7	6	5	4	3	2	1	0
P040	DEFTIM OVRFL INT ENA	DEFTIM OVRFL INT FLAG	CMD/DEF AREA ENA	FAST MODE SELECT	PACT PRE-SCALE SELECT3	PACT PRE-SCALE SELECT2	PACT PRE-SCALE SELECT1	PACT PRE-SCALE SELECT0
	RW-0	RC-0	RW-0	RP-0	RP-0	RP-0	RP-0	RP-0

R = Read, W = Write, P = Privilege write only, C = Clear, -*n* = Value of the bit after the register is reset

Bit 7 **DEFTIM OVRFL INT ENA**. Default Timer Overflow Interrupt Enable.

This bit controls the default timer overflow interrupting capability.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **DEFTIM OVRFL INT FLAG**. Default Timer Overflow Interrupt Flag.

This bit indicates the status of the PACT default timer overflow interrupt. This bit is cleared by reset or when a zero is written to it; it is set by overflow.

0 = Default timer overflow interrupt inactive
1 = Default timer overflow interrupt pending

Bit 5 **CMD/DEF AREA ENA**. Command and Definition Area Enable.

This bit determines if the command/definition area of the dual-port RAM is enabled. This allows the PACT module to use this area for commands and definitions.

0 = Command/definition area is ignored
1 = Enables the commands and definitions

Bit 4 **FAST MODE SELECT**.

This bit determines if the system clock is divided by 8 before entering into the 4-bit prescale. This bit, as well as the PACT PRESCALE SELECT 0–3 bits, determines the time base for the PACT module. The possible combinations are shown below.

Table 15–5. Bits Determining PACT-Module Time Base

PACT PRESCALE SELECTn Bits	Divide Rate FAST MODE SELECT Bit Value	
3 2 1 0	= 1	= 0
0 0 0 0	2	16
0 0 0 1	2	16
0 0 1 0	3	24
0 0 1 1	4	32
0 1 0 0	5	40
0 1 0 1	6	40
0 1 1 0	7	56
0 1 1 1	8	64
1 0 0 0	9	72
1 0 0 1	10	82
1 0 1 0	11	88
1 0 1 1	12	96
1 1 0 0	13	104
1 1 0 1	14	112
1 1 1 0	15	120
1 1 1 1	16	128

Bits 3–0 **PACT PRESCALE SELECT3–0**.

These four bits select a prescaler divide rate for the PACT module. The bits specify the divide of the system clock from /2 to /16, giving 15 possible choices. The actual divide rate is also determined by the value of the FAST MODE SELECT bit. The possible combinations for a FAST MODE SELECT bit in either 1 or 0 are shown in the right half of Table 15–5.

15.11.2 Command/Definition Area Start Register (CDSTART)

The CDSTART register defines the starting address of the command/definition area and enables the interrupts for that area.

Command/Definition Area Start Register (CDSTART)
[Memory Address 1041h]

Bit #	7	6	5	4	3	2	1	0
P041	CMD/DEF AREA INT ENA	—	CMD/DEF AREA START BIT 5	CMD/DEF AREA START BIT 4	CMD/DEF AREA START BIT 3	CMD/DEF AREA START BIT 2	—	—
	RW-0		RW-0	RW-0	RW-0	RW-0		

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **CMD/DEF AREA INT ENA**. Command and Definition Area Interrupt Enable. This bit enables interrupts from the command/definition area.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **Reserved.** Read data is indeterminate.

Bits 5–2 **CMD/DEF AREA START BIT 5–2**. Command and Definition Area Start Bits 5–2.

These bits define the start address of the Command/Definition Area. There are 16 possible locations for this area to start. The address is the same if bits 7, 6, 1, and 0 of this register are set to 1. A table of the bits and their corresponding addresses is shown in Table 15–6.

Bits 1–0 **Reserved.** Read data is indeterminate.

Table 15–6. Bits Defining the Command/Definition Area Start Addresses

CMD/DEF AREA START Bits				CMD/DEF Area Start Addresses			Notes
				TMS370Cx36	TMS370Cx32		
5	4	3	2	Address	Address	Register	
0	0	0	0	01C3h	00C3h	R0C3	
0	0	0	1	01C7h	00C7h	R0C7	
0	0	1	0	01CBh	00CBh	R0CB	
0	0	1	1	01CFh	00CFh	R0CF	
0	1	0	0	01D3h	00D3h	R0D3	
0	1	0	1	01D7h	00D7h	R0D7	
0	1	1	0	01DBh	00DBh	R0DB	
0	1	1	1	01DFh	00DFh	R0DF	
1	0	0	0	01E3h	00E3h	R0E3	
1	0	0	1	01E7h	00E7h	R0E7	
1	0	1	0	01EBh	00EBh	R0EB	
1	0	1	1	01EFh	00EFh	R0EF	†
1	1	0	0	01F3h	00F3h	R0F3	†
1	1	0	1	01F7h	00F7h	R0F7	invalid
1	1	1	0	01FBh	00FBh	R0FB	invalid
1	1	1	1	01FFh	00FFh	R0FF	invalid

† Invalid in Mode B.

15.11.3 Command/Definition Area End Register (CDEND)

The CDEND register defines the end address of the command/definition area.

Command/Definition Area End Register (CDEND)
[Memory Address 1042h]

Bit #	7	6	5	4	3	2	1	0
P042	—	CMD/DEF AREA END BIT 6	CMD/DEF AREA END BIT 5	CMD/DEF AREA END BIT 4	CMD/DEF AREA END BIT 3	CMD/DEF AREA END BIT 2	—	—
		RW-0	RW-0	RW-0	RW-0	RW-0		

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bit 7 **Reserved.** Read data is indeterminate.

Bits 6–2 **CMD/DEF AREA END BIT 6–2**. Command/Definition Area End Bits 6–2.

These bits define the end address of the command/definition area. There are 32 possible locations for the command/definition area end. The address is the same if bit 7 of this register is set to 1, and bit 1 and bit 0 are set to 0. Table 15–7 lists the bits and their corresponding addresses.

Bits 1–0 **Reserved.** Read data is indeterminate.

Table 15–7. Bits Defining the Command/Definition Area End Addresses

CMD/DEF AREA END BIT					CMD/DEF Area End Addresses		
					TMS370Cx36	TMS370Cx32	
6	5	4	3	2	Address	Address	Register
0	0	0	0	0	0180h	0080h	R080h
0	0	0	0	1	0184h	0084h	R084h
0	0	0	1	0	0188h	0088h	R088h
0	0	0	1	1	018Ch	008Ch	R08Ch
0	0	1	0	0	0190h	0090h	R090h
0	0	1	0	1	0194h	0094h	R094h
0	0	1	1	0	0198h	0098h	R098h
0	0	1	1	1	019Ch	009Ch	R09Ch
0	1	0	0	0	01A0h	00A0h	R0A0h
0	1	0	0	1	01A4h	00A4h	R0A4h
0	1	0	1	0	01A8h	00A8h	R0A8h
0	1	0	1	1	01ACh	00ACh	R0ACh
0	1	1	0	0	01B0h	00B0h	R0B0h
0	1	1	0	1	01B4h	00B4h	R0B4h
0	1	1	1	0	01B8h	00B8h	R0B8h
0	1	1	1	1	01BCh	00BCh	R0BCh
1	0	0	0	0	01C0h	00C0h	R0C0h
1	0	0	0	1	01C4h	00C4h	R0C4h
1	0	0	1	0	01C8h	00C8h	R0C8h
1	0	0	1	1	01CCh	00CCh	R0CCh
1	0	1	0	0	01D0h	00D0h	R0D0h
1	0	1	0	1	01D4h	00D4h	R0D4h
1	0	1	1	0	01D8h	00D8h	R0D8h
1	0	1	1	1	01DCh	00DCh	R0DCh
1	1	0	0	0	01E0h	00E0h	R0E0h
1	1	0	0	1	01E4h	00E4h	R0E4h
1	1	0	1	0	01E8h	00E8h	R0E8h
1	1	0	1	1	01ECh	00ECh	R0ECh
1	1	1	0	0	01F0h	00F0h	R0F0h
1	1	1	0	1	01F4h	00F4h	R0F4h
1	1	1	1	0	01F8h	00F8h	R0F8h
1	1	1	1	1	01FCh	00FCh	R0FCh

15.11.4 Buffer Pointer Register (BUFPTR)

The BUFPTR register defines the address of the buffer pointer.

Buffer Pointer Register (BUFPTR)
[Memory Address 1043h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P043	1	1	BUFFER POINTER BIT 5	BUFFER POINTER BIT 4	BUFFER POINTER BIT 3	BUFFER POINTER BIT 2	BUFFER POINTER BIT 1	0
	R-1	R-1	R-1	R-1	R-0	R-0	R-0	R-0

R = Read, *-n* = Value of the bit after the register is reset

Bits 7–6 **Reserved.**

Bits 5–1 **BUFFER POINTER BIT 5–1**.

These bits define the address of the buffer pointer that points to the next available address out of 32 possible locations in the circular buffer. These addresses are the same if bit 7 and bit 6 of this register are set to 1, and bit 0 is set to 0. A table of the bits and their corresponding addresses are listed in Table 15–8.

Bit 0 **Reserved.**

Table 15–8. Buffer Pointer Address Generation

Buffer Pointer Bits					Buffer Pointer		
					TMS370Cx36	TMS370Cx32	
5	4	3	2	1	Address	Address	Register
0	0	0	0	0	01C0h	00C0h	R0C0h
0	0	0	0	1	01C2h	00C2h	R0C2h
0	0	0	1	0	01C4h	00C4h	R0C4h
0	0	0	1	1	01C6h	00C6h	R0C6h
0	0	1	0	0	01C8h	00C8h	R0C8h
0	0	1	0	1	01CAh	00CAh	R0CAh
0	0	1	1	0	01CCh	00CCh	R0CCh
0	0	1	1	1	01CEh	00CEh	R0CEh
0	1	0	0	0	01D0h	00D0h	R0D0h
0	1	0	0	1	01D2h	00D2h	R0D2h
0	1	0	1	0	01D4h	00D4h	R0D4h
0	1	0	1	1	01D6h	00D6h	R0D6h
0	1	1	0	0	01D8h	00D8h	R0D8h
0	1	1	0	1	01DAh	00DAh	R0DAh
0	1	1	1	0	01DCh	00DCh	R0DCh
0	1	1	1	1	01DEh	00DEh	R0DEh
1	0	0	0	0	01E0h	00E0h	R0E0h
1	0	0	0	1	01E2h	00E2h	R0E2h
1	0	0	1	0	01E4h	00E4h	R0E4h
1	0	0	1	1	01E6h	00E6h	R0E6h
1	0	1	0	0	01E8h	00E8h	R0E8h
1	0	1	0	1	01EAh	00EAh	R0EAh
1	0	1	1	0	01ECh	00ECh	R0ECh
1	0	1	1	1	01EEh	00FEh	R0EEh
1	1	0	0	0	01E0h	00F0h	R0F0h
1	1	0	0	1	01F2h	00F2h	R0F2h
1	1	0	1	0	01F4h	00F4h	R0F4h
1	1	0	1	1	01F6h	00F6h	R0F6h
1	1	1	0	0	01F8h	00F8h	R0F8h
1	1	1	0	1	01FAh	00FAh	R0FAh
1	1	1	1	0	01FCh	00FCh	R0FCh
1	1	1	1	1	01FEh	00FEh	R0FEh

15.11.5 PACT-SCI Control Register (SCICTLP)

The SCICTLP register controls the functions of the mini-SCI.

PACT-SCI Control Register (SCICTLP)
[Memory Address 1045h]

Bit #	7	6	5	4	3	2	1	0
P045	PACT RXRDY	PACT TXRDY	PACT PARITY	PACT FE	PACT SCI RX INT ENA	PACT SCI TX INT ENA	—	PACT SCI SW RESET
	RC-0	R-1	R-0	RC-0	RW-0	RW-0		RW-0

R = Read, W = Write, C = Clear, *-n* = Value of the bit after the register is reset

Bit 7 **PACT RXRDY**. PACT Receive Ready.

This bit shows when the receive buffer is full. This bit is cleared by a system reset, by the PACT SCI software reset, when a zero is written to it, or when the SCI RX DATA register is read.

0 = Receive buffer is empty
1 = Receive buffer is full

Bit 6 **PACT TXRDY**. PACT Transmit Ready.

This bit shows when the transmit buffer is empty. This bit is set by a system reset, by the PACT SCI software reset, or when the SCI TX DATA register has been shifted out.

0 = Transmit buffer is full
1 = Transmit buffer is empty

Bit 5 **PACT PARITY**. PACT Receive Data Parity Bit.

This bit is set as the result of the incoming parity calculation. To perform a parity check on incoming data, this bit is compared to a 0 or 1 for even or odd parity respectively.

0 = Received data was even parity
1 = Received data was odd parity

Bit 4 **PACT FE**. PACT Framing Error.

This flag bit shows the detection of a framing error. This bit remains set until cleared by a PACT SCI software reset, by a system reset, or by writing a zero to it.

0 = No framing error
1 = Framing error was detected

Bit 3 **PACT SCI RX INT ENA**. PACT SCI Receive Interrupt Enable.

This bit enables the interrupt to occur when the receive buffer is full.

0 = Does not generate an interrupt when the receive buffer is full
1 = Generates an interrupt when the receive buffer is full

Bit 2 **PACT SCI TX INT ENA**. PACT SCI Transmit Interrupt Enable.

This bit enables the interrupt to occur when the transmit buffer is empty.

0 = Does not generate an interrupt when the transmit buffer is empty
1 = Generates an interrupt when the transmit buffer is empty

Bit 1 **Reserved.** Read data is indeterminate.

Bit 0 **PACT SCI SW RESET**. PACT SCI Software Reset.

When set, this bit puts the SCI into a software reset state so that the parameters of the SCI can be set up. This bit must be cleared to allow the SCI to function.

0 = SCI in operating mode
1 = SCI in software reset mode

15.11.6 PACT-SCI RX Data Register (RXBUFP)

This register contains the data received by the SCI.

PACT-SCI RX Data Register (RXBUFP)
[Memory Address 1046h]

Bit #	7	6	5	4	3	2	1	0
P046	PACT RXDT7	PACT RXDT6	PACT RXDT5	PACT RXDT4	PACT RXDT3	PACT RXDT2	PACT RXDT1	PACT RXDT0
	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

R = Read, -*n* = Value of the bit after the register is reset

Bits 7–0 **PACT RXDT 7–0.** PACT Receive Data 7–0.

15.11.7 PACT-SCI TX Data Register (TXBUFP)

This register contains the data to be transmitted by the SCI.

PACT-SCI TX Data Register (TXBUFP)
[Memory Address 1047h]

Bit #	7	6	5	4	3	2	1	0
P047	PACT TXDT7	PACT TXDT6	PACT TXDT5	PACT TXDT4	PACT TXDT3	PACT TXDT2	PACT TXDT1	PACT TXDT0
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, -*n* = Value of the bit after the register is reset

Bits 7–0 **PACT TXDT 7–0.** PACT Transmit Data 7–0.

15.11.8 Output Pins 1–8 State Register (OPSTATE)

The OPSTATE register contains information about the current state of the output pins.

Output Pin 1–8 State Register (OPSTATE)
[Memory Address 1048h]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P048	PACT OP8 STATE	PACT OP7 STATE	PACT OP6 STATE	PACT OP5 STATE	PACT OP4 STATE	PACT OP3 STATE	PACT OP2 STATE	PACT OP1 STATE
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, *-n* = Value of the bit after the register is reset

Bits 7–0 **PACT OP8–1 STATE**. PACT Output Pins 8–1 State Bits.

These bits reflect the current state of the output pins OP8 to OP1. Each bit is the actual state of the corresponding pin. Writing a 1 to any bit in this register modifies the corresponding output pin as determined by the OP SET/CLR SELECT bit (P04D.0).

Bit OPx Write	OP SET/CLR SELECT	Result
1	1	PACT OPx STATE = 1
1	0	PACT OPx STATE = 0
0	x	PACT OPx STATE remains unchanged

Upon reset, all pins are initialized to the low state.

Example 15–2. Example 1, if OP SET/CLR SELECT = 1

```
11001011    OP STATE Register
11110000    Write to OP STATE Register
————————
11111011    New value in OP STATE Register.
```

Example 15–3. Example 2, if OP SET/CLR SELECT = 0

```
11001011    OP STATE Register
11110000    Write to OP STATE Register
————————
00001011    New value in OP STATE Register.
```

15.11.9 Command/Definition Entry Flags Register (CDFLAGS)

The CDFLAGS register contains information about the command/definition interrupts.

Command/Definition Entry Flags Register (CDFLAGS)
[Memory Address 1049h]

Bit #	7	6	5	4	3	2	1	0
P049	CMD/DEF INT 7 FLAG	CMD/DEF INT 6 FLAG	CMD/DEF INT 5 FLAG	CMD/DEF INT 4 FLAG	CMD/DEF INT 3 FLAG	CMD/DEF INT 2 FLAG	CMD/DEF INT 1 FLAG	CMD/DEF INT 0 FLAG
	RC-0	RC-0	RC-0	RC-0	RC-0	RC-0	RC-0	RC-0

R = Read, C = Clear, -*n* = Value of the bit after the register is reset

Bits 7–0 **CMD/DEF INT 7–0 FLAG**. Command/Definition Interrupt 7–0 Flag.

These bits are the interrupt flags for the command/definition area. If an interrupt has been enabled in a 4-byte command or definition, then the appropriate bit in this register will be set when the interrupt conditions are met. The actual bit set is determined by the command or definition's entry address. The entry address is derived from bits 2, 3, and 4 of the address. If the command or definition is at address 006Ch, 006Dh, 006Eh, or 006Fh (32 bits), then 0110 11xx is the binary value of the address. The entry address comes from bits 4–3–2 = 011 = Entry 3. Thus, the flag associated with entry address 3 is used when this command or definition causes an interrupt.

These flags are not affected by the CMD/DEF AREA INT ENA bit (CDSTART.7).

CMD/DEF INT FLAG Set	Command or Definition Entry That Generated Interrupt Request
CMD/DEF INT 0 FLAG	(Entry Address >> 2) ENTRY mod 8 = 0
CMD/DEF INT 1 FLAG	(Entry Address >> 2) ENTRY mod 8 = 1
CMD/DEF INT 2 FLAG	(Entry Address >> 2) ENTRY mod 8 = 2
CMD/DEF INT 3 FLAG	(Entry Address >> 2) ENTRY mod 8 = 3
CMD/DEF INT 4 FLAG	(Entry Address >> 2) ENTRY mod 8 = 4
CMD/DEF INT 5 FLAG	(Entry Address >> 2) ENTRY mod 8 = 5
CMD/DEF INT 6 FLAG	(Entry Address >> 2) ENTRY mod 8 = 6
CMD/DEF INT 7 FLAG	(Entry Address >> 2) ENTRY mod 8 = 7

15.11.10 Setup CP Control Register 1 (CPCTL1)

The CPCTL1 register controls the functions of the CP1 and CP2 pins.

Setup CP Control Register 1 (CPCTL1)
[Memory Address 104Ah]

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
P04A	CP2 INT ENA	CP2 INT FLAG	CP2 CAPT RISING EDGE	CP2 CAPT FALLING EDGE	CP1 INT ENA	CP1 INT FLAG	CP1 CAPT RISING EDGE	CP1 CAPT FALLING EDGE
	RW-0	RC-0	RW-0	RW-0	RW-0	RC-0	RW-0	RW-0

R = Read, W = Write, C = Clear, *-n* = Value of the bit after the register is reset

Bit 7 **CP2 INT ENA**. CP2 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP2.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **CP2 INT FLAG**. CP2 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP2. This bit must be cleared by the program during an interrupt routine when CP2 INT ENA is set.

0 = Capture interrupt from selected edge of CP2 inactive
1 = Capture interrupt from selected edge of CP2 pending

Bit 5 **CP2 CAPT RISING EDGE**. CP2 Capture Rising Edge.

This bit selects the rising edge on pin CP2 to cause a timer capture. Table 15–9 describes all possible combinations.

Table 15–9. Rising/Falling Edge Capture Bits

CPx CAPT RISING EDGE	CPx CAPT FALLING EDGE	Capture on Selected Edges
0	0	Disables captures
0	1	Captures on falling edges only
1	0	Captures on rising edges only
1	1	Captures on both rising and falling edges

Bit 4 **CP2 CAPT FALLING EDGE**. CP2 Capture Falling Edge.

This bit selects the falling edge on pin CP2 to cause a timer capture. Table 15–9 describes all possible combinations.

Bit 3 **CP1 INT ENA**. CP1 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP1.

0 = Disables interrupt
1 = Enables interrupt

Bit 2 **CP1 INT FLAG**. CP1 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP1. This bit must be cleared by the program during an interrupt routine when CP1 INT ENA is set.

0 = Capture interrupt from selected edge of CP1 inactive
1 = Capture interrupt from selected edge of CP1 pending

Bit 1 **CP1 CAPT RISING EDGE**. CP1 Capture Rising Edge.

This bit selects the rising edge on pin CP1 to cause a timer capture. Table 15–9 describes all possible combinations.

Bit 0 **CP1 CAPT FALLING EDGE**. CP1 Capture Falling Edge.

This bit selects the falling edge on pin CP1 to cause a timer capture. Table 15–9 describes all possible combinations.

15.11.11 Setup CP Control Register 2 (CPCTL2)

The CPCTL2 register controls the functions of the CP3 and CP4 pins.

Setup CP Control Register 2 (CPCTL2)
[Memory Address 104Bh]

Bit #	7	6	5	4	3	2	1	0
P04B	CP4 INT ENA	CP4 INT FLAG	CP4 CAPT RISING EDGE	CP4 CAPT FALLING EDGE	CP3 INT ENA	CP3 INT FLAG	CP3 CAPT RISING EDGE	CP3 CAPT FALLING EDGE
	RW-0	RC-0	RW-0	RW-0	RW-0	RC-0	RW-0	RW-0

R = Read, W = Write, C = Clear, *-n* = Value of the bit after the register is reset

Bit 7 **CP4 INT ENA**. CP4 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP4.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **CP4 INT FLAG**. CP4 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP4. This bit must be cleared by the program during an interrupt routine when CP4 INT ENA is set.

0 = Capture interrupt from selected edge of CP4 inactive
1 = Capture interrupt from selected edge of CP4 pending

Bit 5 **CP4 CAPT RISING EDGE**. CP4 Capture Rising Edge.

This bit selects the rising edge on pin CP4 to cause a timer capture. Table 15–10 describes all possible combinations.

Table 15–10. Rising/Falling Edge Capture Bits

CPx CAPT RISING EDGE	CPx CAPT FALLING EDGE	Capture on Selected Edges
0	0	Disables captures
0	1	Captures on falling edges only
1	0	Captures on rising edges only
1	1	Captures on both rising and falling edges

Bit 4 **CP4 CAPT FALLING EDGE**. CP4 Capture Falling Edge.

This bit selects the falling edge on pin CP4 to cause a timer capture. Table 15–10 describes all possible combinations.See the table following the bit 1 description for all possible combinations.

Bit 3 **CP3 INT ENA**. CP3 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP3.

0 = Disables interrupt
1 = Enables interrupt

Bit 2 **CP3 INT FLAG**. CP3 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP3. This bit must be cleared by the program during an interrupt routine when CP3 INT ENA is set.

0 = Capture interrupt from selected edge of CP3 inactive
1 = Capture interrupt from selected edge of CP3 pending

Bit 1 **CP3 CAPT RISING EDGE**. CP3 Capture Rising Edge.

This bit selects the rising edge on pin CP3 to cause a timer capture. Table 15–10 describes all possible combinations.

Bit 0 **CP3 CAPT FALLING EDGE**. CP3 Capture Falling Edge.

This bit selects the falling edge on pin CP3 to cause a timer capture. Table 15–10 describes all possible combinations.

15.11.12 Setup CP Control Register 3 (CPCTL3)

The CPCTL3 register controls the functions of the CP5 and CP6 pins.

Setup CP Control Register 3 (CPCTL3)
[Memory Address 104Ch]

Bit #	7	6	5	4	3	2	1	0
P04C	CP6 INT ENA	CP6 INT FLAG	CP6 CAPT RISING EDGE	CP6 CAPT FALLING EDGE	CP5 INT ENA	CP5 INT FLAG	CP5 CAPT RISING EDGE	CP5 CAPT FALLING EDGE
	RW-0	RC-0	RW-0	RW-0	RW-0	RC-0	RW-0	RW-0

R = Read, W = Write, C = Clear, *-n* = Value of the bit after the register is reset

Bit 7 **CP6 INT ENA**. CP6 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP6.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **CP6 INT FLAG**. CP6 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP6. This bit must be cleared by the program during an interrupt routine when CP6 INT ENA is set.

0 = Capture interrupt from selected edge of CP6 inactive
1 = Capture interrupt from selected edge of CP6 pending

Bit 5 **CP6 CAPT RISING EDGE**. CP6 Capture Rising Edge.

This bit selects the rising edge on pin CP6 to cause a timer capture. Table 15–11 describes all possible combinations.

Table 15–11. Rising/Falling Edge Capture Bits

CPx CAPT RISING EDGE	CPx CAPT FALLING EDGE	Capture on Selected Edges
0	0	Disables captures
0	1	Captures on falling edges only
1	0	Captures on rising edges only
1	1	Captures on both rising and falling edges

Bit 4 **CP6 CAPT FALLING EDGE**. CP6 Capture Falling Edge.

This bit selects the falling edge on pin CP6 to cause a timer capture. Table 15–11 describes all possible combinations.

Bit 3 **CP5 INT ENA**. CP5 Interrupt Enable.

If set, this bit enables the interrupt when the selected edge occurs on pin CP5.

0 = Disables interrupt
1 = Enables interrupt

Bit 2 **CP5 INT FLAG**. CP5 Interrupt Flag.

This bit indicates that the selected edge has occurred on pin CP5. This bit must be cleared by the program during an interrupt routine when CP5 INT ENA is set.

0 = Capture interrupt from selected edge of CP5 inactive
1 = Capture interrupt from selected edge of CP5 pending

Bit 1 **CP5 CAPT RISING EDGE**. CP5 Capture Rising Edge.

This bit selects the rising edge on pin CP5 to cause a timer capture. Table 15–11 describes all possible combinations.

Bit 0 **CP5 CAPT FALLING EDGE**. CP5 Capture Falling Edge.

This bit selects the falling edge on pin CP5 to cause a timer capture. Table 15–11 describes all possible combinations.

15.11.13 CP Input Control Register (CPPRE)

The CPPRE register controls input and output functions.

CP Input Control Register (CPPRE)
[Memory Address 104Dh]

Bit #	7	6	5	4	3	2	1	0
P04D	BUFFER HALF/FULL INT ENA	BUFFER HALF/FULL INT FLAG	INPUT CAPT PRE-SCALE SELECT3	INPUT CAPT PRE-SCALE SELECT2	INPUT CAPT PRE-SCALE SELECT1	CP6 EVENT ONLY	EVENT COUNTER SW RESET	OP SET/CLR SELECT
	RW-0	RC-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, C = Clear, *-n* = Value of the bit after the register is reset

Bit 7 **BUFFER HALF/FULL INT ENA**. Buffer Half/Full Interrupt Enable.

This bit determines whether or not the circular buffer can generate an interrupt on the half full and full buffer boundaries.

0 = Disables interrupt
1 = Enables interrupt

Bit 6 **BUFFER HALF/FULL INT FLAG**. Buffer Half/Full Interrupt Flag.

This bit is set when the circular buffer becomes half or completely full. It is cleared when a zero is written to this bit or during RESET.

0 = Interrupt inactive
1 = Interrupt pending

Bits 5–3 **INPUT CAPT PRESCALE SELECT3–1**. Input Capture Prescale Select 3–1.

These bits set the prescaler rate for pins CP3 to CP6. The bits allow a divide rate of /1 to /8 as shown in the table below. The prescale rate does not affect the event counter.

INPUT CAPT PRESCALE SELECTx Bits			Divide Rate
3	2	1	
0	0	0	/1
0	0	1	/2
0	1	0	/3
0	1	1	/4
1	0	0	/5
1	0	1	/6
1	1	0	/7
1	1	1	/8

Bit 2 **CP6 EVENT ONLY**. CP6 8-Bit Event Counter Input Only.

This bit must be cleared to allow 32-bit captures triggered by CP6. This bit does not disable the 16-bit captures on event (CP6) when triggered by a command/definition area command.

0 = CP6 increments event counter and causes 32-bit captures
1 = CP6 increments event counter only

Bit 1 **EVENT COUNTER SW RESET**. 8-Bit Event Counter Software Reset.

This bit resets the 8-bit event counter. When set, the 8-bit counter is continuously cleared. This bit *must* be cleared to enable the event counter to operate.

0 = Event counter operating
1 = Event counter cleared

Bit 0 **OP SET/CLR SELECT**. Output Pin Set/Clear Write Function Select.

This bit controls how the outputs OP1 to OP8 are set or cleared by software.

- When OP SET/CLR = 1, a write to P048 causes the output pins corresponding to the locations that were written as 1 to be set in the high state. The output pins corresponding to the locations that were written as 0 remain unchanged.
- When OP SET/CLR = 0, a write to P048 causes the output pins corresponding to the locations that were written as 1 to be set in the low state. The output pins corresponding to the locations that were written as 0 remain unchanged.

Refer to the following table and to the examples in subsection 15.11.8 on page 15-49.

OPx WRITE Bit	OP SET/CLR SELECT Bit	Result
1	1	PACT OPx STATE = 1
1	0	PACT OPx STATE = 0
0	x	PACT OPx STATE remains unchanged

15.11.14 Global Function Control Register (PACTPRI)

The PACTPRI register controls the WD time-out rate, the PACT interrupt priority levels, and the PACT operating mode.

Global Function Control Register (PACTPRI)
[Memory Address 104Fh]

Bit #	7	6	5	4	3	2	1	0
P04F	PACT STEST	—	PACT GROUP 1 PRIORITY	PACT GROUP 2 PRIORITY	PACT GROUP 3 PRIORITY	PACT MODE SELECT	PACT WD PRE-SCALE SELECT1	PACT WD PRE-SCALE SELECT0
	RP-0		RP-0	RP-0	RP-0	RP-0	RP-0	RP-0

R = Read, P = Privileged write only, C = Clear, -*n* = Value of the bit after the register is reset

Bit 7 **PACT STEST**.

This bit must be cleared to ensure proper operation.

Bit 6 **Reserved.** Read data is indeterminate

Bit 5 **PACT GROUP 1 PRIORITY.** PACT Group 1 Priority Select.

This bit assigns the interrupt priority level of the PACT group 1 interrupt vectors.

0 = PACT group 1 interrupts are level 1 (high-priority) requests.
1 = PACT group 1 interrupts are level 2 (low-priority) requests.

Bit 4 **PACT GROUP 2 PRIORITY**. PACT Group 2 Priority Select.

This bit assigns the interrupt priority level of the PACT group 2 interrupt vectors.

0 = PACT group 2 interrupts are level 1 (high-priority) requests.
1 = PACT group 2 interrupts are level 2 (low-priority) requests.

Bit 3 **PACT GROUP 3 PRIORITY**. PACT Group 3 Priority Select.

This bit assigns the interrupt priority level of the PACT group 3 interrupt vectors.

0 = PACT group 3 interrupts are level 1 (high-priority) requests.
1 = PACT group 3 interrupts are level 2 (low-priority) requests.

Bit 2 **PACT MODE SELECT**. PACT Mode Select.

This bit selects the mode for the PACT module to operate in.

0 = PACT operates in mode A
1 = PACT operates in mode B

Bit 1–0 **PACT WD PRESCALE SELECT1–0**. PACT WD Prescale Select 1–0.

These bits select the WD time-out rate. You can write to these bits only during privilege mode (after reset).

PACT WD PRESCALE SELECT1 Bit	PACT WD PRESCALE SELECT0 Bit	Options
0	0	WD reset on bit 9 of default timer
0	1	WD reset on bit 15 of default timer
1	0	WD reset on bit 19 of default timer
1	1	Disable WD

Chapter 16

Assembly Language Instruction Set

An assembly language instruction set is a symbolic language that presents binary machine code in a more readable form. The TMS370 family is supported by a 73-function instruction set that uses a wide variety of addressing modes.

This chapter includes the following topics:

Topic **Page**

16.1 Instruction Operation

The assembly language instruction set provides a convenient method of programming the CPU. Each TMS370 assembly language instruction converts directly to one machine operation and consists of the following elements:

- **A function mnemonic.** The mnemonic specifies the type of CPU operation.
- **Zero to three operands.** The operands indicate where the CPU can find or store data during an instruction execution. The type and combination of operands determine the actual opcode(s) for an instruction. The MOV instruction, for example, has 27 different options, each with its own opcode.

A typical two-operand instruction is shown below:

Mnemonic	Source	Destination
ADD	#9,	R3

The example above can be read as follows: add the value 9 to the contents of register number 3 and place the result back into register number 3. The destination serves as a second source as well as the final address of the result; moreover, registers can be directly manipulated without having to use intermediate registers. Note that this instruction form differs from the mnemonic-destination-source arrangement that some microprocessors use.

The following example shows how the instruction above might appear in a complete program line.

Label	Instruction	Operands	Comment
XXXXX	ADD	#9,R3	;comment

There should be at least one space between each entry type. The label and comment entries are optional.

The 73 instructions are supported by 246 opcodes that provide flexible control of CPU program flow. Some instructions, such as CLRC and TEST A, share the same opcode to help you understand all of the functions of an opcode. Some instructions use 16-bit opcodes, depending on the type of instruction and/or the addressing mode used. The assembler constructs several bit manipulation instructions from other instructions to simplify writing the instructions and to enhance the readability of the program.

16.2 Symbol Definitions

To understand the instructions described in this chapter, you must know what the symbols in the syntax descriptions represent. Table 16–1 lists the instruction set symbols.

Table 16–1. TMS370 Symbols Defined

Symbol	Definition	Symbol	Definition
A	Register A or R0 in register file	Rd	Destination register in register file ($0 \leq n \leq 255$)
B	Register B or R1 in register file	Rn	Register n of register file ($0 \leq n \leq 255$)
C	Carry flag/no borrow flag	Rname	Symbol-defined register bit
cnd	Condition	Rp	Register pair
d/D	Destination operand (8-bit/16-bit)	Rpd	Destination register pair
iop8	8-bit immediate operand	Rps	Source register pair
iop16	16-bit immediate operand	Rs	Source register in register file ($0 \leq s \leq 255$)
label	16-bit label	s	Source operand
LSB	Least significant bit	SP	Stack pointer
LSbyte	Least significant byte	ST	Status register
MSB	Most significant bit	V	Overflow
MSbyte	Most significant byte	XADDR	16-bit address
N	Sign flag	Z	Zero flag
name	Symbol-defined for a bit	(x)	Contents of memory at address x
PC	Program counter	((x))	Contents of memory location designated by contents at address x
PCN	16-bit address of next instruction (program counter next)	< >	Indicates an entry that must be typed in. For example, <label> indicates that a label must be entered. The brackets themselves are not entered.
Pd	Destination register in peripheral file ($0 \leq d \leq 255$)	$\rightarrow$	Is assigned to
Pn	Register n of peripheral file ($0 \leq n \leq 255$)	$\leftarrow$	Becomes equal to
Pname	Symbol-defined peripheral bit	#	Immediate operand prefix
Ps	Source register in peripheral file ($0 \leq s \leq 255$)	*	Indirect addressing operand prefix
off8	8-bit signed offset	&	Direct addressing operand prefix
off16	16-bit signed offset		

16.3 Addressing Modes

Each TMS370 assembly language instruction includes from zero to three operands. Each operand has an addressing mode. The addressing mode specifies how the CPU calculates the address of the data needed by the instruction. The power of the TMS370 is enhanced by the large number of addressing modes available.

The 18 addressing modes are divided into two classes:

- **Data Manipulation**
- **Program Flow**

Table 16–2 shows the 18 addressing modes, each with a sample instruction and its execution. The subsections that follow describe these modes.

Table 16–2. Overview of Addressing Modes

Addressing Mode	Example	Execution
Data Manipulation:		
Implied	LDSP	(B) → (SP)
Register	MOV R05,R04	(0005h) → (0004h)
Peripheral	MOV P025,A	(1025h) → (A)
Immediate	ADD #23h,R03	23h + (0003h) → (0003h)
Stack Pointer Relative	MOV *2h[SP],A	(2h + (SP)) → (A)
Direct	MOV A,&1234h	(A) → (1234h)
Indexed	MOV *1234h[B],A	(1234h + (B)) → (A)
Indirect	MOV *R04,A	((R03:R04)) → (A)
Offset Indirect	MOV *12h[R04],A	(12h + (R03:R04)) → (A)
Program Flow:		
PC Relative	JMP #offset8	PCN + offset8 → (PC)
Absolute Direct	BR 4567h	4567h → (PC)
Absolute Indexed	BR *1234h[B]	1234h + (B) → (PC)
Absolute Indirect	BR *R04	(R03:R04) → (PC)
Absolute Offset Indirect	BR *2h[R04]	2h + (R03:R04) → (PC)
Relative Direct	JMPL #1234h	PCN + 1234h → (PC)
Relative Indexed	JMPL *1234h[B]	PCN + 1234h + (B) → (PC)
Relative Indirect	JMPL *R04	PCN + (R03:R04) → (PC)
Relative Offset Indirect	JMPL *12h[R04]	PCN + 12h + (R03:R04) → (PC)

A number of instructions use more than one addressing mode, and several instructions, such as MOV, are very versatile.

16.3.1 Data Manipulation Addressing Modes

Instructions using the **data manipulation addressing** modes deal with the register file, peripheral file, or a nearby destination. The data manipulation addressing modes are as follows:

- ❑ Implied
- ❑ Register
- ❑ Peripheral
- ❑ Immediate
- ❑ Stack pointer relative
- ❑ Direct
- ❑ Indexed
- ❑ Indirect
- ❑ Offset Indirect

Most of these modes can use *any* register as a source and/or destination, preventing the bottleneck found on other microprocessors that use only one or two registers. The direct, indexed, indirect, and offset indirect addressing modes always use Register A as the operand to generate a 16-bit address. These addressing modes are used only by the compare (CMP) and move (MOV) instructions.

16.3.1.1 Implied Addressing Mode

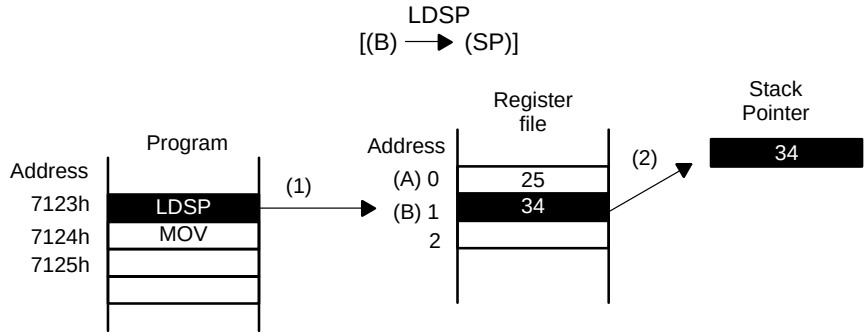
In the **implied addressing** mode, the instruction type alone determines where the data is to be found. You do not have to specify the operands, because they are inherently specified in the instruction.

For example, the LDSP (load stack pointer) instruction always copies the contents of register B to the stack pointer (SP). Neither the source nor the destination is explicitly stated; they are implied in the instruction itself. These are the instructions that use the implied addressing mode:

CLRC	Clear the carry bit	LDSP	Load stack pointer
RTS	Return from subroutine	RTI	Return from interrupt
SETC	Set carry	STSP	Store stack pointer
EINT	Enable interrupts	EINTH	Enable high-level interrupts
EINTL	Enable low-level interrupts		

Figure 16–1 shows an example of the implied addressing mode.

Figure 16–1. Implied Addressing Mode



Note: Numbers in parentheses represent the order of execution.

16.3.1.2 Register Addressing Mode

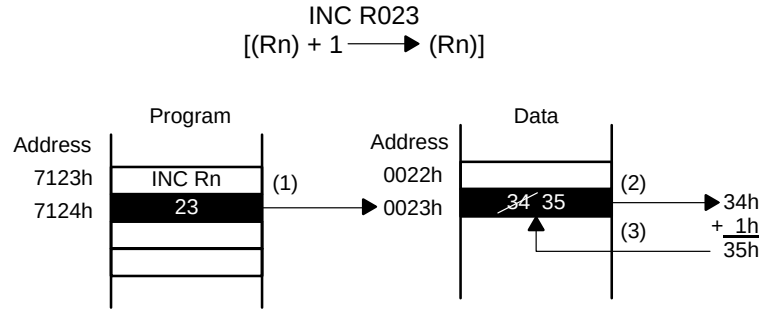
The register file (RF) of the TMS370 consists of the first 128 or 256 bytes of memory (the number of bytes differs according to the device that you are using). In the **register addressing** mode, instructions use a one-byte value to specify an address (location) in the RF. Any location in the RF can be accessed in one memory cycle by instructions using this mode. Other addressing modes require two cycles to access the register file.

In register file addressing, the operand is stated by Rn, where n is the 8-bit address number. The address number can be a decimal (0–255) or hexadecimal (0–0FF) number. Hexadecimal numbers require a leading zero, but no suffix. Registers R0 and R1 of the register file are also known as registers A and B and are referenced as such by most instructions to reduce the size of the program. For example, the instruction `MOV A,B` uses one byte of code, while the instruction `MOV R3,R4` uses three bytes of code. Any register can be specified by a symbol that has been equated to that register. This is illustrated in the following example:

```
       MOV   R16,R011  ;Move contents of 0010h to 0011h
CAT    .EQU  R16       ;Equate register 16 to symbol CAT
DOG    .EQU  R17       ;Equate register 17 to symbol DOG
       MOV   CAT,DOG   ;Move contents of 0010h to 0011h
```

Note that the entry .EQU is an assembler directive, not an assembly language instruction. For more information on assembler directives, refer to the *TMS370 Family Assembly Language Tools User's Guide*. Figure 16–2 shows an example of the register addressing mode.

Figure 16–2. Register Addressing Mode



Note: Numbers in parentheses represent the order of execution.

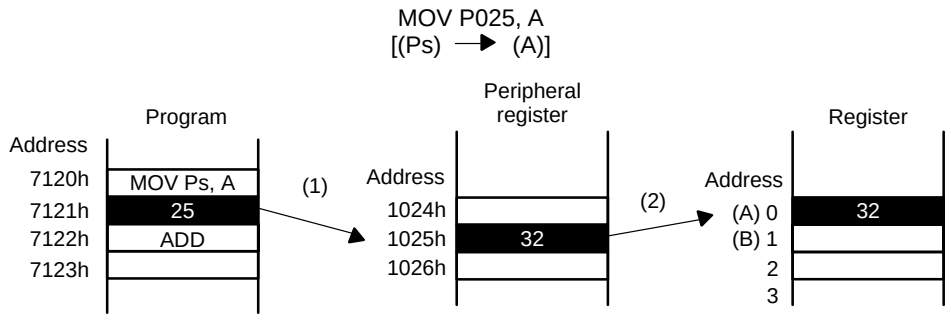
16.3.1.3 Peripheral Addressing Mode

The **peripheral addressing** mode is used for program control of the peripheral on-chip modules such as timers, interrupts, and I/O ports. Devices with bus expansion can address a small amount of external memory as peripheral file (PF) space. The PF of the TMS370 is allocated 256 bytes of memory. Each PF register is accessed by an 8-bit operand designated as Pn, with n being either a decimal (0–255) or hexadecimal (0–0FF) number. Hexadecimal numbers require a leading zero but no suffix. The CPU assumes the most significant byte of a peripheral address to be 010h. As described for register file addressing in subsection 16.3.1.2, the Pn designation, like the Rn designation, can be replaced with a symbol by using the equate (.EQU) assembler directive as shown in the example below.

```
       MOV   R16,P020 ;Move contents of 0010h to 1020h
CAT    .EQU  R16      ;Equate register 16 to symbol CAT
DOG    .EQU  P32      ;Equate peripheral file 32 to symbol DOG
       MOV   CAT,DOG  ;Move contents of 0010h to 1020h
```

The use of designated symbols is optional but is particularly suited for the register and peripheral addressing modes. Figure 16–3 shows an example of peripheral file addressing.

Figure 16–3. Peripheral Addressing Mode



Note: Numbers in parentheses represent the order of execution.

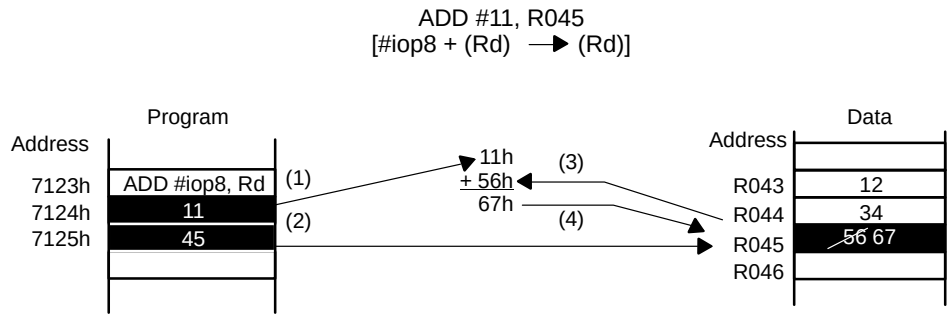
16.3.1.4 Immediate Addressing Mode

The **immediate addressing** mode uses a constant value as the operand that immediately follows the function mnemonic. This mode allows nonchanging data to be incorporated into the instruction. The constant can be in the form of a decimal number, a hexadecimal number, or a symbolic label, but the constant is always preceded by the number sign (#). Note that hexadecimal numbers require *both* a leading numeric digit *and* the h suffix. Some examples of immediate addressing are as follows:

```
     MOV   #0Fh,A      ;Store the value 15 in register A
     MOV   #(3*54),R022;Store the value 162 at location 022h
CNT  .EQU  12          ;Equate 12 to symbol CNT
     ADD   #CNT,R34    ;Add the value 12 to register 34, place
                       ; result in register 34.
```

Figure 16–4 illustrates an instruction using the immediate addressing mode.

Figure 16–4. Immediate Addressing Mode



Note: Numbers in parentheses represent the order of execution.

16.3.1.5 Stack Pointer Relative Addressing Mode

The **stack pointer relative addressing** mode adds an 8-bit signed constant to the existing 8-bit contents of the stack pointer register. The result is truncated to an 8-bit address of the data. The second operand in the stack pointer relative mode is always register A.

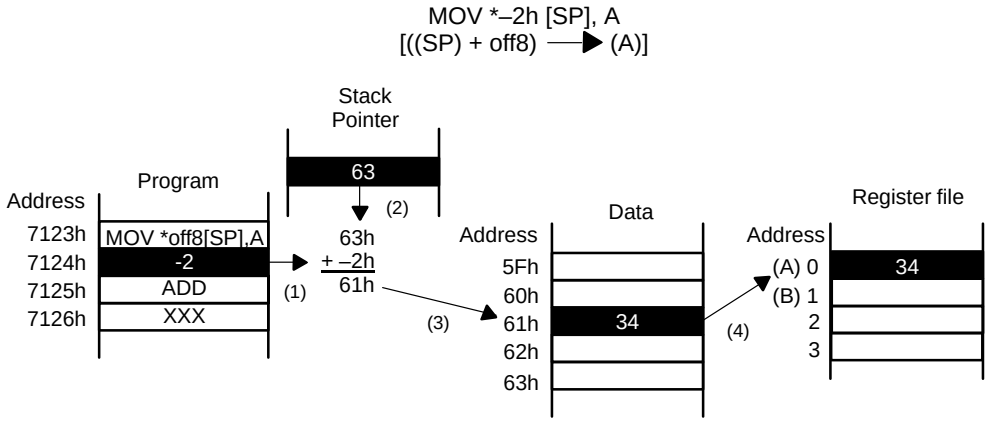
This addressing mode is useful in accessing arguments that are passed to a subroutine on the stack. You must ensure that the resulting address location is within the implemented register file, because overflows or underflows will execute without warning.

Only the CMP and MOV instructions use this mode. An example of stack relative addressing is as follows:

```
MOV *–2h[SP], A
```

In this example, the value of –2 plus the stack pointer equals the address of the data to be moved to register A. Figure 16–5 illustrates this instruction operation.

Figure 16–5. Stack Pointer Relative Addressing Mode

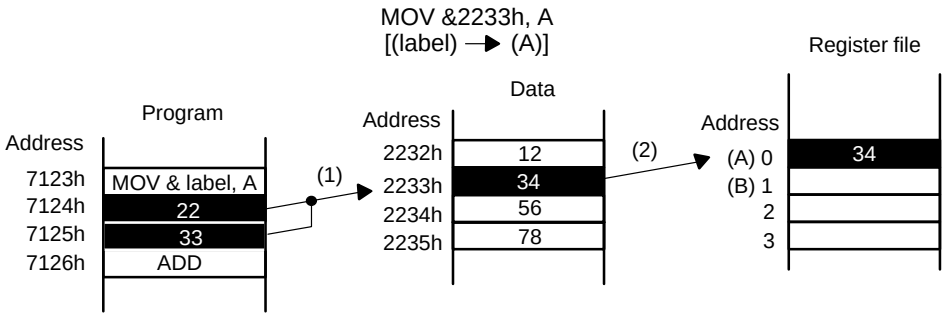


Note: Numbers in parentheses represent the order of execution.

16.3.1.6 Direct Addressing Modes

Direct addressing mode instructions use an address as the operand. The 16-bit address is written as either a constant value or a label and immediately follows the opcode in the source code. The direct addressing mode acts on the address itself as shown in Figure 16–6.

Figure 16–6. Direct Addressing Mode



Note: Numbers in parentheses represent the order of execution.

16.3.1.7 Indexed Addressing Mode

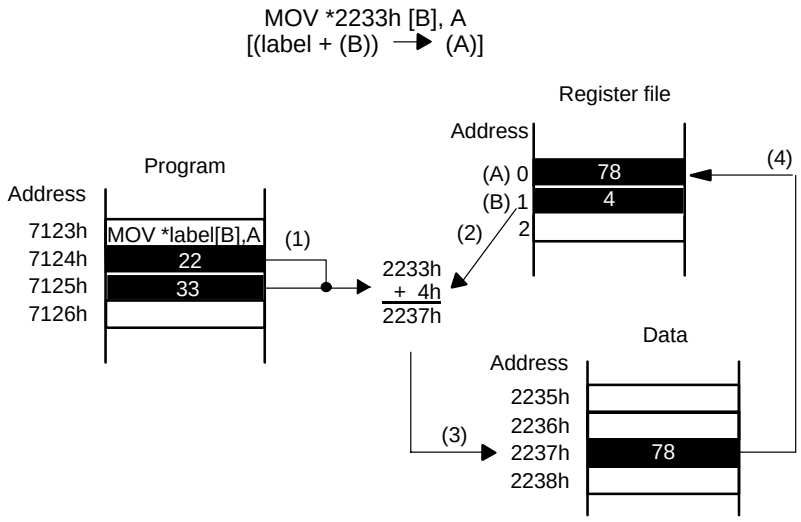
The **indexed addressing** mode generates a 16-bit address by adding the unsigned contents of register B to a 16-bit unsigned constant. The assembly language statement for the indexed addressing modes contains the direct memory address written as a 16-bit value or a label, preceded by an asterisk (*) and followed by a B in brackets; for example, MOV *1234[B], or MOV *LABEL[B].

Note the following:

- The MOV and CMP instructions can use indexed addressing to step easily through a small table or to pick out a particular array value.

Figure 16–7 illustrates how the object code produced by an instruction using this mode generates a 16-bit effective address.

Figure 16–7. Indexed Addressing Mode



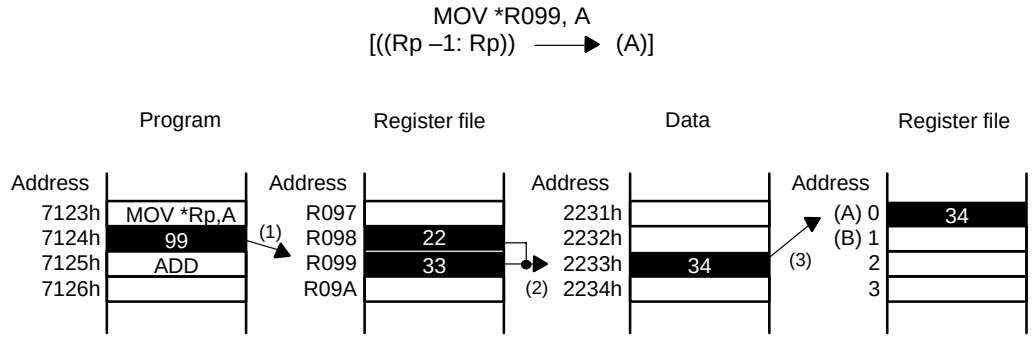
Note: Numbers in parentheses represent the order of execution.

16.3.1.8 Indirect Addressing Mode

In **indirect addressing** modes, instructions use the contents of a register pair as the 16-bit address of the data. The indirect register file address is written as a register number (Rn) preceded by an asterisk (*) symbol. The LSbyte of the address is contained in Rn, and the MSbyte of the address is contained in the previous register (Rn–1). The TMS370 can use any register pair as an indirect register.

Figure 16–8 shows how the indirect addressing mode uses the register pair in the calculation.

Figure 16–8. Indirect Addressing Mode

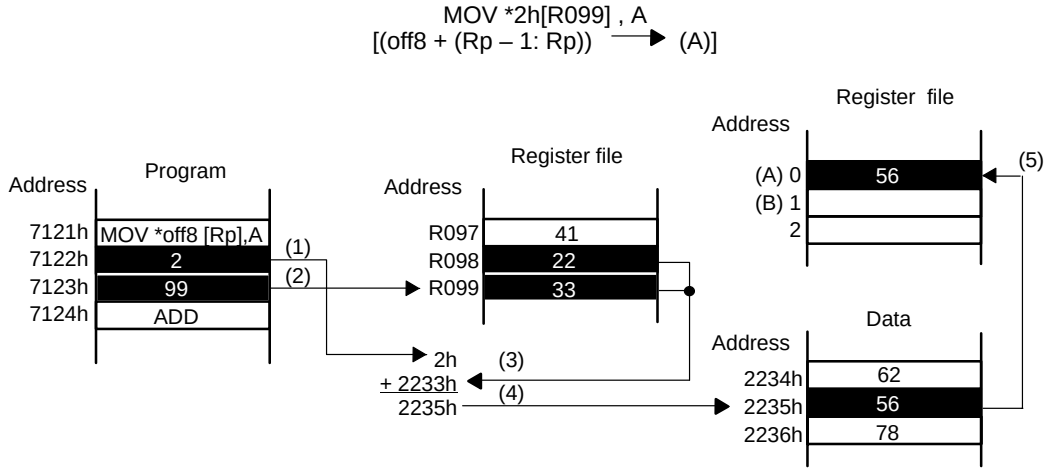


Note: Numbers in parentheses represent the order of execution.

16.3.1.9 Offset Indirect Addressing Mode

The **offset indirect addressing** mode is similar to the indirect addressing mode previously described. The **offset indirect addressing** mode generates a 16-bit address by adding an 8-bit signed offset to an address taken from a register pair. Offset indirect addressing is useful for stepping through tables or for finding a particular value in a table by using two values to generate the address. Figure 16–9 illustrates how the object code produced by an instruction using the offset indirect addressing mode generates a 16-bit effective address.

Figure 16–9. Offset Indirect Addressing Mode



Note: Numbers in parentheses represent the order of execution.

16.3.2 Program Flow Addressing Modes

The **program flow addressing** modes provide a sophisticated method for stepping through tables, picking out array values, and generating addresses. These modes allow the program to access data from anywhere in the memory. Program flow addressing modes consist of the following nine types:

- Absolute Direct
- Absolute Indexed
- Absolute Indirect
- Absolute Offset Indirect
- PC Relative
- Relative Direct
- Relative Indexed
- Relative Indirect
- Relative Offset Indirect

There are two modes of program flow addressing; absolute addressing and relative addressing. Each mode is described below.

- **Program flow absolute addressing** modes always use the PC (program counter) as the destination operand to store the generated 16-bit address. The program flow absolute addressing modes are used only by the branch (BR) and CALL instructions.

- The **program flow relative addressing** modes are similar to the program flow absolute addressing modes but include the additional step of combining the operand with the PCN value before placing the 16-bit address into the program counter. These modes are similar to the program counter relative mode. A 16-bit signed offset is used to calculate the succeeding instruction address. The succeeding instruction address is calculated at execution time by using the signed 16-bit offset according to the instruction's addressing mode.

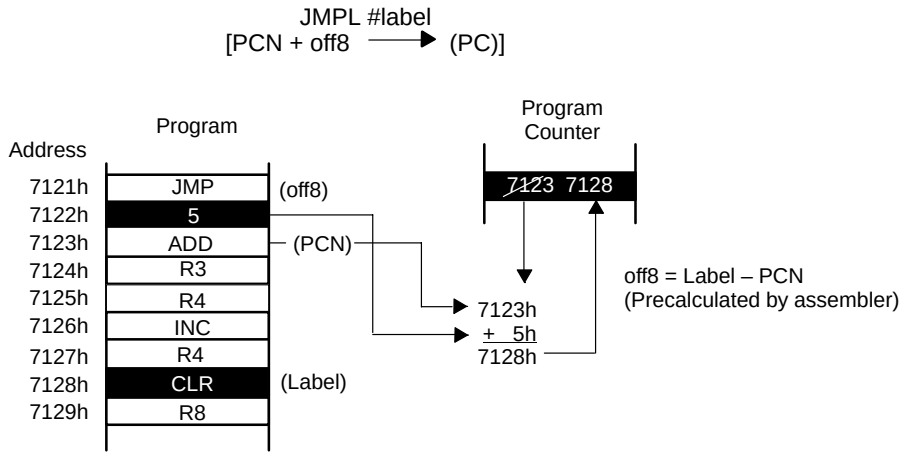
  The program flow relative addressing modes are useful in relocatable code because their operation is based on the differences in the address position instead of on the addresses themselves. This makes the program flow relative addressing modes well suited for high-level languages that often use position-independent code. Program flow relative addressing is used by the CALLR and JMPL instructions.

16.3.2.1 Program Counter Relative Addressing Mode

The **program counter relative addressing** mode adds an 8-bit signed offset to the address of the next instruction to produce the address of the succeeding instruction. The new address is placed in the program counter register. The range of the 8-bit offset is within 128 bytes before or 127 bytes after the instruction following the jump. When labels are used, the signed offset is automatically calculated by the assembler. Note that the PCN is the location (address) of the next instruction.

Figure 16–10 illustrates object code generated by a jump instruction using the program counter relative mode.

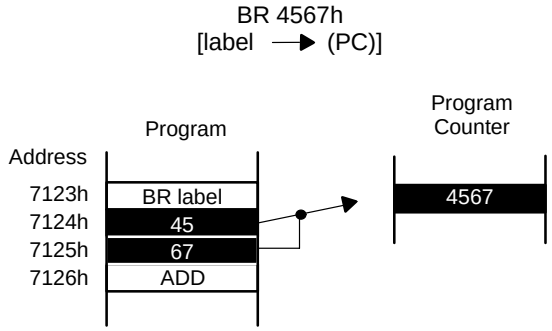
Figure 16–10. Program Counter Relative Addressing Mode



16.3.2.2 Absolute and Relative Direct Addressing Modes

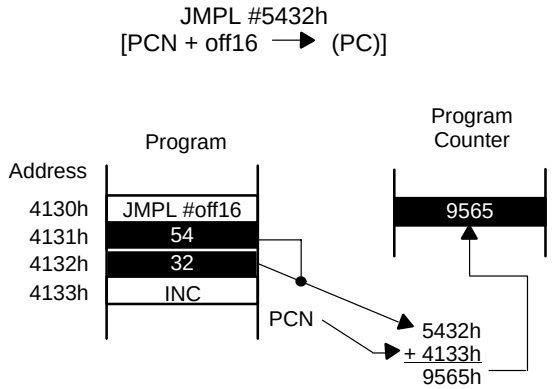
Absolute and relative direct addresssing mode instructions use an address as the operand. The 16-bit address is written as either a constant value or a label and immediately follows the opcode in the source code. The absolute direct addressing mode takes a label and places the value into the PC as shown in Figure 16–11.

Figure 16–11. Absolute Direct Addressing Mode



The **relative direct addressing** mode (see Figure 16–12) adds the address of the next instruction to the 16-bit operand to produce the address of the succeeding instruction. If a label is used in the instruction, the assembler automatically calculates the offset to use as the operand.

Figure 16–12. Relative Direct Addressing Mode



16.3.2.3 Absolute and Relative Indexed Addressing Modes

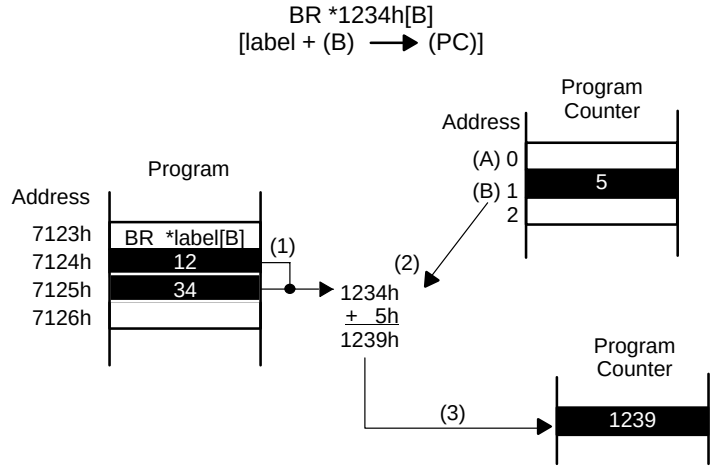
The **absolute indexed addressing** mode generates a 16-bit address by adding the unsigned contents of register B to a 16-bit unsigned constant. The 16-bit address is then placed into the PC. The assembly language statement for the absolute indexed addressing modes contains the direct memory address written as a 16-bit value or a label preceded by an asterisk (*) and followed by a B in brackets; for example, BR *1234[B] or BR *LABEL[B].

Note: Use of Indexed Addressing by BR and CALL Instructions

The CALL and BR instructions can use this mode to execute code according to a decision table and the value in register B.

Figure 16–13 illustrates how the object code produced by an instruction using this mode generates a 16-bit effective address.

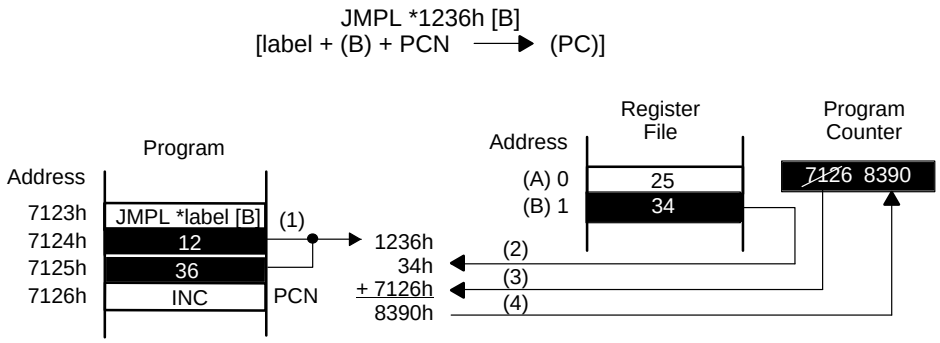
Figure 16–13. Absolute Indexed Addressing Mode



Note: Numbers in parentheses represent the order of execution.

The **relative indexed addressing** mode includes the operation described for the absolute indexed addressing mode with the following additional step: the address of the next instruction is added to the sum of register B and the signed 16-bit constant offset to produce the address of the next instruction. The relative indexed addressing mode is shown in Figure 16–14.

Figure 16–14. Relative Indexed Addressing Mode



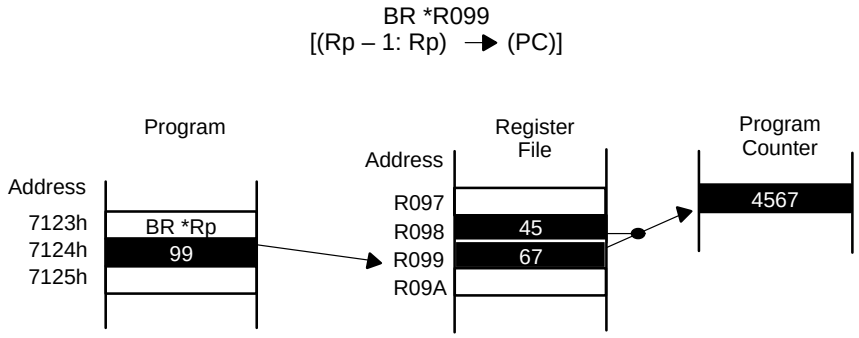
Note: Numbers in parentheses represent order of execution.

16.3.2.4 Absolute and Relative Indirect Addressing Modes

In **absolute indirect addressing** mode, instructions use the contents of a register pair as the 16-bit address. The absolute and relative indirect register file address is written as a register number (Rn) preceded by an asterisk (*) symbol. The LSbyte of the address is contained in Rn, and the MSbyte of the address is contained in the previous register (Rn–1). The TMS370 can use any register pair as an absolute or relative indirect register.

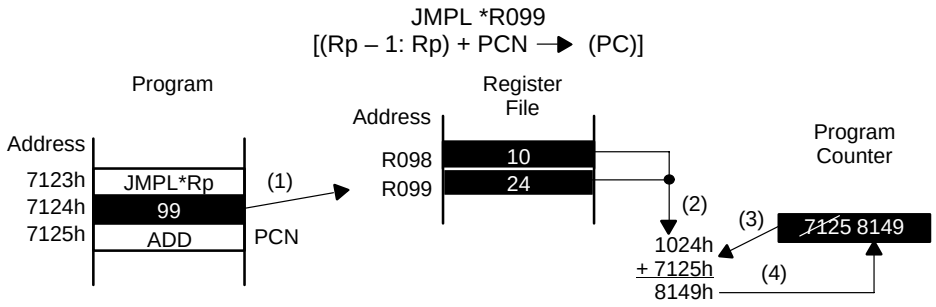
Figure 16–15 shows how the absolute indirect addressing mode uses the register pair in the calculation.

Figure 16–15. Absolute Indirect Addressing Mode



The **relative indirect addressing** mode (Figure 16–16) adds the address of the next instruction to the register pair contents before obtaining the destination address.

Figure 16–16. Relative Indirect Addressing Mode

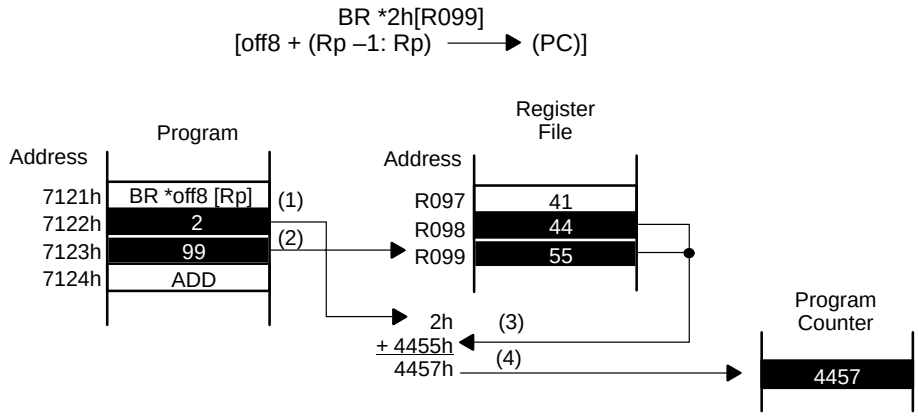


Note: Numbers in parentheses represent the order of execution.

16.3.2.5 Absolute and Relative Offset Indirect Addressing Modes

The **absolute and relative offset indirect addressing** modes are similar to the **absolute and relative** indirect addressing modes previously described. The absolute offset indirect addressing mode generates a 16-bit address by adding an 8-bit signed offset to an address taken from a register pair. The 16-bit address is then placed into the PC. Offset indirect addressing is useful for stepping through tables or for finding a particular value in a table by using two values to generate the address. Figure 16–17 illustrates how the object code produced by an instruction using the absolute offset indirect addressing mode generates a 16-bit effective address.

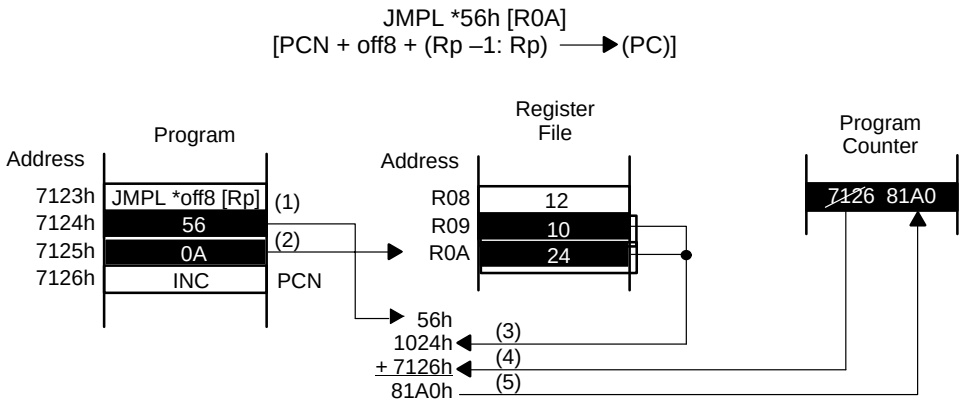
Figure 16–17. Absolute Offset Indirect Addressing Mode



Note: Numbers in parentheses represent the order of execution.

The **relative offset indirect addressing** mode adds the address of the next instruction with the sum of the 8-bit signed offset and the register pair before obtaining the destination address.

Figure 16–18. Relative Offset Indirect Addressing Mode



Note: Numbers in parentheses represent the order of execution.

16.3.3 Additional Addressing Modes

In some cases, the operation of an instruction does not fit into any of the addressing modes previously described. Some modes illustrated by instructions such as MOVW #iop[B],Rpd provide unique capabilities for table addressing. Other modes illustrated by instructions like the LDST #iop8 give access to the status register bits (shown in Figure 16–19). The individual instruction description can be referenced for a list of that instruction's operations.

16.3.4 Status Register

Most of the instructions affect the bits in the status register. The status register is presented in Figure 16–19 as a quick reference to aid in programming.

Figure 16–19. Status Register (ST)

Status Register (ST)

Bit #	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
	C	N	Z	V	IE2	IE1	—	—
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0		

R = Read, W = Write, -*n* = Value of the bit after the register is reset

16.4 Instruction Set Overview

The tables in this section list the instruction set, including pertinent characteristics and an opcode/instruction map.

Table 16–3 lists all instruction formats, opcodes, byte lengths, cycles/instructions, operands, status bits affected, and an operational description.

Table 16–3. TMS370 Family Instruction Overview

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
ADC	B,A	69	1	8	x x x x	(s) + (d) + (C) → (d) Add the source, destination, and carry bit together. Store at the destination address.
	Rs,A	19	2	7		
	Rs,B	39	2	7		
	Rs,Rd	49	3	9		
	#iop8,A	29	2	6		
	#iop8,B	59	2	6		
	#iop8,Rd	79	3	8		
ADD	B,A	68	1	8	x x x x	(s) + (d) → (d) Add the source and destination operands and store at the destination address.
	Rs,A	18	2	7		
	Rs,B	38	2	7		
	Rs,Rd	48	3	9		
	#iop8,A	28	2	6		
	#iop8,B	58	2	6		
	#iop8,Rd	78	3	8		
AND	A,Pd	83	2	9	0 x x 0	(s) AND (d) → (d) AND the source and destination operands together and store at the destination address.
	B,A	63	1	8		
	B,Pd	93	2	9		
	Rs,A	13	2	7		
	Rs,B	33	2	7		
	Rs,Rd	43	3	9		
	#iop8,A	23	2	6		
	#iop8,B	53	2	6		
	#iop8,Rd	73	3	8		
	#iop8,Pd	A3	3	10		
BR	label	8C	3	9	– – – –	XADDR → (PC) Branch to the destination address.
	*Rp	9C	2	8		
	*label[B]	AC	3	11		
	*off8[Rp]	F4 EC	4	16		
BTJO†	A,Pd,#off8	86	3	10	0 x x 0	If (s) AND (d) ≠ 0, then PCN + off8 → (PC). If the AND of the source and destination operands ≠ 0 (corresponding 1 bit), the PC will add the offset, and the jump will be taken.
	B,A,#off8	66	2	10		
	B,Pd,#off8	96	3	10		
	Rs,A,#off8	16	3	9		
	Rs,B,#off8	36	3	9		
	Rs,Rd,#off8	46	4	11		
	#iop8,A,#off8	26	3	8		
	#iop8,B,#off8	56	3	8		
	#iop8,Rd,#off8	76	4	10		
	#iop8,Pd,#off8	A6	4	11		

† Add two to the cycle count if a jump is taken.

‡ Status Values:

- 0 Status bit always cleared.
- 1 Status bit always set.
- x Status bit cleared or set on results.
- – Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Continued)

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
BTJZ†	A,Pd,#off8	87	3	10	0 x x 0	If (s) AND (not d) ≠ 0, then (PCN) + off8→ (PC). If any 1 in the source corresponds to a 0 in the destination, the PC adds the offset, and the jump is taken.
	B,A,#off8	67	2	10		
	B,Pd,#off8	97	3	10		
	Rs,A,#off8	17	3	9		
	Rs,B,#off8	37	3	9		
	Rs,Rd,#off8	47	4	11		
	#iop8,A,#off8	27	3	8		
	#iop8,B,#off8	57	3	8		
	#iop8,Rd,#off8	77	4	10		
	#iop8,Pd,#off8	A7	4	11		
CALL	label	8E	3	13	– – – –	Push PC MSbyte, PC LSbyte, XADDR → (PC)
	*Rp	9E	2	12		
	*label[B]	AE	3	15		
	*off8[Rp]	F4 EE	4	20		
CALLR	#off16	8F	3	15	– – – –	Call relative Push PC MSbyte, PC LSbyte, PCN + (XADDR) → (PC)
	label	8F	3	15		
	*Rp	9F	2	14		
	*off16[B]	AF	3	17		
	*label[B]	AF	3	17		
	*off8[Rp]	F4 EF	4	22		
CLR	A	B5	1	8	0 0 1 0	0 → (Rd) Clear the destination operand.
	B	C5	1	8		
	Rd	D5	2	6		
CLRC		B0	1	9	0 x x 0	0 → (C) Clear the carry bit. N and Z bits are set on the value in A.
CMP	*label,A	8D	3	11	x x x x	Compare; (d) – (s) computed. Set flags on the result of the source operand subtracted from the destination operand. Operands are not affected by operation.
	*Rp,A	9D	2	10		
	*label[B],A	AD	3	13		
	*off8[Rp],A	F4 ED	4	18		
	*off8[SP],A	F3	2	8		
	B,A	6D	1	8		
	Rs,A	1D	2	7		
	Rs,B	3D	2	7		
	Rs,Rd	4D	3	9		
	#iop8,A	2D	2	6		
	#iop8,B	5D	2	6		
	#iop8,Rd	7D	3	8		
CMPBIT	Rname	75	3	8	0 x x 0	Complement bit; invert the bit
	Pname	A5	3	10		
COMPL	A	BB	1	8	x x x 0	2s complement; 00h – (s) → (d)
	B	CB	1	8		
	Rd	DB	2	6		

† Add two to the cycle count if a jump is taken.

‡ Status Values:

- 0 Status bit always cleared.
- 1 Status bit always set.
- x Status bit cleared or set on results.
- – Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Continued)

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
DAC	B,A	6E	1	10	x x x x	(s) + (d) + (C) → (d) (BCD) The source, destination, and carry bit are added, and the BCD sum is stored at the destination address.
	Rs,A	1E	2	9		
	Rs,B	3E	2	9		
	Rs,Rd	4E	3	11		
	#iop8,A	2E	2	8		
	#iop8,B	5E	2	8		
	#iop8,Rd	7E	3	10		
DEC	A	B2	1	8	x x x x	(d) – 1 → (d) Decrement destination operand by 1.
	B	C2	1	8		
	Rd	D2	2	6		
DINT		F0 00	2	6	0 0 0 0	0 → (ST)(global interrupt enable bits) 0 → IE1, 0 ⟶ IE2.
DIV	Rs,A	F4 F8	3	55–63 14	0 x x 0 1 1 1 1	A:B/Rs → A(= QUO),B(= REM) Integer divide, 16 by 8 bits. Overflow detected.
DJNZ†	A,#off8	BA	2	10	– – – –	(d) – 1 → (d); If (d) ≠ 0, then PCN + off8 → (PC). Decrement and jump if not 0.
	B,#off8	CA	2	10		
	Rd,#off8	DA	3	8		
DSB	B,A	6F	1	10	x x x x	(d) – (s) – 1 + (C) → (d) (BCD) The source operand is subtracted from the destination; this sum is then reduced by 1, and the carry bit is then added to it. The result is stored as a BCD number.
	Rs,A	1F	2	9		
	Rs,B	3F	2	9		
	Rs,Rd	4F	3	11		
	#iop8,A	2F	2	8		
	#iop8,B	5F	2	8		
	#iop8,Rd	7F	3	10		
EINT		F0 0C	2	6	0 0 0 0	0Ch → (ST)(global interrupt enable bit) 1 → IE1, 1 → IE2.
EINTH		F0 04	2	6	0 0 0 0	04h → (ST)(high-priority global interrupt enable bit) 1 → IE1, 0 → IE2
EINTL		F0 08	2	6	0 0 0 0	08h→ (ST)(low-priority global interrupt enable bit) 0 → IE1, 1 → IE2
IDLE		F6	1	6	– – – –	(PC) → (PC) until interrupt (PC) + 1 → (PC) after return from interrupt. Stops µC execution until an interrupt. Entry to low-power modes.
INC	A	B3	1	8	x x x x	(d) + 1 → (d) Increase the destination operand by 1.
	B	C3	1	8		
	Rd	D3	2	6		
INCW	#iop8,Rp	70	3	11	x x x x	(Rp) + operand→ (Rp) Add 8-bit immediate operand to register pair.

† Add two to the cycle count if a jump is taken.

‡ Status Values:

- 0 Status bit always cleared.
- 1 Status bit always set.
- x Status bit cleared or set on results.
- – Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Continued)

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
INV	A	B4	1	8	0 x x 0	NOT(d) → (d) 1s complement the destination operand.
	B	C4	1	8		
	Rd	D4	2	6		
JBIT0†	Rname,#off8	77	4	10	0 x x 0	Jump if bit = 0 This instruction may have as many as 8 operands. All operands must reference the same byte.
	Pname,#off8	A7	4	11		
JBIT1†	Rname,#off8	76	4	10	0 x x 0	Jump if bit = 1
	Pname,#off8	A6	4	11		
JMP	off8	00	2	7	– – – –	PCN + off8 → (PC) Jump unconditionally using an 8-bit offset.
JMPL	#off16	89	3	9	– – – –	PCN + D → (PC) Jump unconditionally using a 16-bit offset.
	label	89	3	9		
	*Rp	99	2	8		
	*off16[B]	A9	3	11		
	*label[B]	A9	3	11		
	*off8[Rp]	F4 E9	4	16		
J*cnd*†					– – – –	Conditional jump
	JC	03	2	5		Carry
	JEQ	02	2	5		Jump equal
	JG	0E	2	5		Greater than, signed
	JGE	0D	2	5		Greater than or equal, signed
	JHS	0B	2	5		Higher or same, unsigned
	JL	09	2	5		Less than, signed
	JLE	0A	2	5		Less than or equal, signed
	JLO	0F	2	5		Lower value, unsigned
	JN	01	2	5		Negative, signed
	JNC	07	2	5		No carry
	JNE	06	2	5		Jump not equal
	JNV	0C	2	5		No overflow, signed
	JNZ	06	2	5		Not zero
	JP	04	2	5		Positive, signed
	JPZ	05	2	5		Positive or zero, signed
	JV	08	2	5		Overflow, signed
	JZ	02	2	5		Zero
LDSP		FD	1	7	– – – –	(B) → (SP) Load stack pointer with contents of register B.
LDST	#iop8	F0	2	6	x x x x	(s) → (ST) Load ST register.

† Add two to the cycle count if a jump is taken.

‡ Status Values:

0 Status bit always cleared.
1 Status bit always set.
x Status bit cleared or set on results.
– Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Continued)

Mnemonics and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
MOV	A,B	C0	1	9	0 x x 0	(s) → (d) Replace the destination operand with the source operand.
	A,Rd	D0	2	7		
	A,Pd	21	2	8		
	A,&label	8B	3	10		
	A,*Rp	9B	2	9		
	A,*label[B]	AB	3	12		
	A,*off8[Rp]	F4 EB	4	17		
	A,*off8[SP]	F2	2	7		
	Rs,A	12	2	7		
	Rs,B	32	2	7		
	&label,A	8A	3	10		
	*Rp,A	9A	2	9		
	*label[B],A	AA	3	12		
	*off8[Rp],A	F4 EA	4	17		
	*off8[SP],A	F1	2	7		
	B,A	62	1	8		
	B,Rd	D1	2	7		
	B,Pd	51	2	8		
	Rs,Rd	42	3	9		
	Rs,Pd	71	3	10		
	Ps,A	80	2	8		
	Ps,B	91	2	8		
	Ps,Rd	A2	3	10		
	#iop8,A	22	2	6		
	#iop8,B	52	2	6		
	#iop8,Rd	72	3	8		
	#iop8,Pd	F7	3	10		
MOVW	Rps,Rpd	98	3	12	0 x x 0	(s) → (Rpd–1:Rpd) Copy the source register word to the destination register pair.
	#iop16,Rpd	88	4	13		
	#iop16[B],Rpd	A8	4	15		
	#off8[Rps],Rpd	F4 E8	5	20		
MPY	B,A	6C	1	47	0 x x 0	(s) x (d) → (A:B) Multiply the source and destination operands; store the result in registers A (MSbyte) and B (LSbyte).
	Rs,A	1C	2	46		
	Rs,B	3C	2	46		
	Rs,Rd	4C	3	48		
	#iop8,A	2C	2	45		
	#iop8,B	5C	2	45		
	#iop8,Rd	7C	3	47		
NOP		FF	1	7	– – – –	No operation

‡ Status Values:

0 Status bit always cleared.
1 Status bit always set.
x Status bit cleared or set on results.
– Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Continued)

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
OR	A,Pd	84	2	9	0 x x 0	(s) OR (d) → (d)
	B,A	64	1	8		
	B,Pd	94	2	9		Logically OR the source and destination operands, and store the results at the destination address.
	Rs,A	14	2	7		
	Rs,B	34	2	7		
	Rs,Rd	44	3	9		
	#iop8,A	24	2	6		
	#iop8,B	54	2	6		
	#iop8,Rd	74	3	8		
	#iop8,Pd	A4	3	10		
POP	A	B9	1	9	0 x x 0	((SP)) → (d) (SP) –1 → (SP)
	B	C9	1	9		
	Rd	D9	2	7		
	ST	FC	1	8	x x x x	
PUSH	A	B8	1	9	0 x x 0	(SP) + 1 → (SP) (s) → ((SP)) Copy the operand onto the stack.
	B	C8	1	9		
	Rs	D8	2	7		
	ST	FB	1	8	– – – –	Copy the status register onto the stack.
RL	A	BE	1	8	x x x 0	Bit(n) → Bit(n + 1) Bit(7) → Bit(0) and Carry
	B	CE	1	8		
	Rd	DE	2	6		
RLC	A	BF	1	8	x x x 0	Bit(n) → Bit(n + 1) Carry → Bit(0) Bit(7) → Carry
	B	CF	1	8		
	Rd	DF	2	6		
RR	A	BC	1	8	x x x 0	Bit(n + 1) → Bit(n) Bit(0) → Bit(7) and Carry
	B	CC	1	8		
	Rd	DC	2	6		
RRC	A	BD	1	8	x x x 0	Bit(n + 1) → Bit(n) Carry → Bit(7) Bit(0) → Carry
	B	CD	1	8		
	Rd	DD	2	6		
RTI		FA	1	12	x x x x	Pop PCL, PCH, POP ST Return from interrupt.
RTS		F9	1	9	– – – –	Pop PCL, PCH
SBB	B,A	6B	1	8	x x x x	(d) – (s) – 1 + (C) → (d) Subtract with borrow. Destination minus source minus 1 plus carry; stored at the destination address.
	Rs,A	1B	2	7		
	Rs,B	3B	2	7		
	Rs,Rd	4B	3	9		
	#iop8,A	2B	2	6		
	#iop8,B	5B	2	6		
	#iop8,Rd	7B	3	8		
SBIT0	Rname	73	3	8	0 x x 0	Set bit to 0
	Pname	A3	3	10		
SBIT1	Rname	74	3	8	0 x x 0	Set bit to 1
	Pname	A4	3	10		

‡ Status Values:

- 0 Status bit always cleared.
- 1 Status bit always set.
- x Status bit cleared or set on results.
- – Status bit not affected.

Table 16–3. TMS370 Family Instruction Overview (Concluded)

Mnemonic and Operands		Opcode	Bytes	Cycles (t_C)	Status C N Z V‡	Operation Description
SETC		F8	1	7	1 0 1 0	Axh → (ST) Set the carry bit. IE1 and IE2 unchanged.
STSP		FE	1	8	– – – –	(SP) → (B) Copy the SP into register B.
SUB	B,A	6A	1	8	x x x x	(d) – (s) → (d) Store the destination operand minus the source operand into the destination.
	Rs,A	1A	2	7		
	Rs,B	3A	2	7		
	Rs,Rd	4A	3	9		
	#iop8,A	2A	2	6		
	#iop8,B	5A	2	6		
	#iop8,Rd	7A	3	8		
SWAP	A	B7	1	11	0 x x 0	s(7–4,3–0) → d(3–0,7–4) Swap the operand's high and low nibbles.
	B	C7	1	11		
	Rd	D7	2	9		
TRAP	#n	EF–E0	1	14	– – – –	Vector n → (PC), n = 0 → 15 = trap number Trap to subroutine; Push PCN Trap 0 = EF, Trap 15 = E0
TST	A	B0	1	9	0 x x 0	Test; Set status bit according to the value in register A or B.
	B	C6	1	10		
XCHB	A	B6	1	10	0 x x 0	(B) ←→ (Rn) (with Rn = A or B or Rd) Swap the contents of register B with (Rn).
	B	C6	1	10		
	Rd	D6	2	8		
XOR	A,Pd	85	2	9	0 x x 0	(s) XOR (d) → (d) Logically exclusive OR the source and destination operands; store at the destination address.
	B,A	65	1	8		
	B,Pd	95	2	9		
	Rs,A	15	2	7		
	Rs,B	35	2	7		
	Rs,Rd	45	3	9		
	#iop8,A	25	2	6		
	#iop8,B	55	2	6		
	#iop8,Rd	75	3	8		
	#iop8,Pd	A5	3	10		

‡ Status Values:

0 Status bit always cleared.
1 Status bit always set.
x Status bit cleared or set on results.
– Status bit not affected.

Table 16–4 provides an opcode-to-instruction cross-reference of all 73 instructions and 274 opcodes of the TMS370 instruction set. To check the instruction of a known opcode, locate the left (high) digit across the top or bottom of the table, then find the right (low) digit along the side of the table. The intersection contains the instruction mnemonic, operands, and byte/cycle peculiar to that opcode. Some opcodes, such as B0, are shared by two instructions, in which case, both mnemonics are shown with the byte/cycle count.

Table 16–4. TMS370 Family Opcode/Instruction Map†

LSN \ MSN	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	JMP ra 2/7							INCW #n,Rd 3/11	MOV Ps,A 2/8			CLRC / TST A 1/9	MOV A,B 1/9	MOV A,Rd 2/7	TRAP 15 1/14	LDST n 2/6
1	JN ra 2/5		MOV A,Pd 2/8			MOV B,Pd 2/8		MOV Rs,Pd 3/10		MOV Ps,B 2/7				MOV B,Rd 2/7	TRAP 14 1/14	MOV *n[SP],A 2/7
2	JZ ra 2/5	MOV Rs,A 2/7	MOV #n,A 2/6	MOV Rs,B 2/7	MOV Rs,Rd 3/9	MOV #n,B 2/6	MOV B,A 1/8	MOV #n,Rd 3/8			MOV Ps,Rd 3/10	DEC A 1/8	DEC B 1/8	DEC Rn 2/6	TRAP 13 1/14	MOV *A,n[SP] 2/7
3	JC ra 2/5	AND Rs,A 2/7	AND #n,A 2/6	AND Rs,B 2/7	AND Rs,Rd 3/9	AND #n,B 2/6	AND B,A 1/8	AND #n,Rd 3/8	AND A,Pd 2/9	AND B,Pd 2/9	AND #n,Pd 3/10	INC A 1/8	INC B 1/8	INC Rn 2/6	TRAP 12 1/14	CMP *n[SP],A 2/8
4	JP ra 2/5	OR Rs,A 2/7	OR #n,A 2/6	OR Rs,B 2/7	OR Rs,Rd 3/9	OR #n,B 2/6	OR B,A 1/8	OR #n,Rd 3/8	OR A,Pd 2/9	OR B,Pd 2/9	OR #n,Pd 3/10	INV A 1/8	INV B 1/8	INV Rn 2/6	TRAP 11 1/14	extend inst,2 opcodes
5	JPZ ra 2/5	XOR Rs,A 2/7	XOR #n,A 2/6	XOR Rs,B 2/7	XOR Rs,Rd 3/9	XOR #n,B 2/6	XOR B,A 1/8	XOR #n,Rd 3/8	XOR A,Pd 2/9	XOR B,Pd 2/9	XOR #n,Pd 3/10	CLR A 1/8	CLR B 1/8	CLR Rd 2/6	TRAP 10 1/14	
6	JNZ ra 2/5	BTJO Rs,A,ra 3/9	BTJO #n,A,ra 3/8	BTJO Rs,B,ra 3/9	BTJO Rs,Rd,ra 4/11	BTJO #n,B,ra 3/8	BTJO B,A,ra 2/10	BTJO #n,Rd,ra 4/10	BTJO A,Pd,ra 3/11	BTJO B,Pd,ra 3/10	BTJO #n,Pd,ra 4/11	XCHB A 1/10	XCHB A / TST B 1/10	XCHB Rn 2/8	TRAP 9 1/14	IDLE 1/6
7	JNC ra 2/5	BTJZ Rs,A,ra 3/9	BTJZ #n,A,ra 3/8	BTJZ Rs,B,ra 3/9	BTJZ Rs,Rd,ra 4/11	BTJZ #n,B,ra 3/8	BTJZ B,A,ra 2/10	BTJZ #n,Rd,ra 4/10	BTJZ A,Pd,ra 3/10	BTJZ B,Pd,ra 3/10	BTJZ #n,Pd,ra 4/11	SWAP A 1/11	SWAP B 1/11	SWAP Rn 2/9	TRAP 8 1/14	MOV #n,Pd 3/10
8	JV ra 2/5	ADD Rs,A 2/7	ADD #n,A 2/6	ADD Rs,B 2/7	ADD Rs,Rd 3/9	ADD #n,B 2/6	ADD B,A 1/8	ADD #n,Rd 3/8	MOVW #16,Rd 4/13	MOVW Rs,Rd 3/12	MOVW #16[B],Rd 4/15	PUSH A 1/9	PUSH B 1/9	PUSH Rs 2/7	TRAP 7 1/14	SETC 1/7
9	JL ra 2/5	ADC Rs,A 2/7	ADC #n,A 2/6	ADC Rs,B 2/7	ADC Rs,Rd 3/9	ADC #n,B 2/6	ADC B,A 1/8	ADC #n,Rd 3/8	JMPL lab 3/9	JMPL *Rd 2/8	JMPL *lab[B] 3/11	POP A 1/9	POP B 1/9	POP Rd 2/7	TRAP 6 1/14	RTS 1/9
A	JLE ra 2/5	SUB Rs,A 2/7	SUB #n,A 2/6	SUB Rs,B 2/7	SUB Rs,Rd 3/9	SUB #n,B 2/6	SUB B,A 1/8	SUB #n,Rd 3/8	MOV &lab,A 3/10	MOV *Rs,A 2/9	MOV *lab[B],A 3/12	DJNZ A,ra 2/10	DJNZ B,ra 2/10	DJNZ Rn,ra 3/8	TRAP 5 1/14	RTI 1/12
B	JHS ra 2/5	SBB Rs,A 2/7	SBB #n,A 2/6	SBB Rs,B 2/7	SBB Rs,Rd 3/9	SBB #n,B 2/6	SBB B,A 1/8	SBB #n,Rd 3/8	MOV A,&lab 3/10	MOV A,*Rp 2/9	MOV A,*lab[B] 3/12	COMPL A 1/8	COMPL B 1/8	COMPL Rn 2/6	TRAP 4 1/14	PUSH ST 1/8

† All conditional jumps (opcodes 01–0F), BTJO, BTJZ, and DJNZ instructions use two additional cycles if the branch is taken. The BTJO, BTJZ, and DJNZ instructions have a relative address as the last operand. Abbreviations are explained on the next page of this table.

Table 16–4. TMS370 Family Opcode/Instruction Map† (Concluded)

LSN \ MSN	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	JNV ra 2/5	MPY Rs,A 2/46	MPY #n,A 2/45	MPY Rs,B 2/46	MPY Rs,Rd 3/48	MPY #n,B 2/45	MPY B,A 1/47	MPY #n,Rs 3/47	BR lab 3/9	BR *Rd 2/8	BR *lab[B] 3/11	RR A 1/8	RR B 1/8	RR Rn 2/6	TRAP 3 1/14	POP ST 1/8
D	JGE ra 2/5	CMP Rs,A 2/7	CMP #n,A 2/6	CMP Rs,B 2/7	CMP Rs,Rd 3/9	CMP #n,B 2/6	CMP B,A 1/8	CMP #n,Rd 3/8	CMP &lab,A 3/11	CMP *Rs,A 2/10	CMP *lab[B],A 3/13	RRC A 1/8	RRC B 1/8	RRC Rn 2/6	TRAP 2 1/14	LDSP 1/7
E	JG ra 2/5	DAC Rs,A 2/9	DAC #n,A 2/8	DAC Rs,B 2/9	DAC Rs,Rd 3/11	DAC #n,B 2/8	DAC B,A 1/10	DAC #n,Rd 3/10	CALL lab 3/13	CALL *Rd 2/12	CALL *lab[B)] 3/15	RL A 1/8	RL B 1/8	RL Rn 2/6	TRAP 1 1/14	STSP 1/8
F	JLO ra 2/5	DSB Rs,A 2/9	DSB #n,A 2/8	DSB Rs,B 2/9	DSB Rs,Rd 3/11	DSB #n,B 2/8	DSB B,A 1/10	DSB #n,Rd 3/10	CALLR lab 3/15	CALLR *Rd 2/14	CALLR *lab[B] 3/17	RLC A 1/8	RLC B 1/8	RLC Rn 2/6	TRAP 0 1/14	NOP 1/7

Second byte of two-byte instructions (F4xx):			
F4	8	MOVW *n[Rp] 4/15	DIV Rn,A 3/14-63
F4	9	JMPL *n[Rp] 4/16	
F4	A	MOV *n[Rp],A 4/17	
F4	B	MOV A,*n[Rp] 4/16	
F4	C	BR *n[Rp] 4/17	
F4	D	CMP *n[Rp],A 4/18	
F4	E	CALL *n[Rp] 4/20	
F4	F	CALLR *n[Rp] 4/22	

* = Indirect addressing operand prefix
& = Direct addressing operand prefix
# = Immediate operand
#16 = Immediate 16-bit number
lab = 16-bit label
n = Immediate 8-bit number
Pd = Peripheral register containing destination type
Pn = Peripheral register
Ps = Peripheral register containing source byte
ra = Relative address
Rd = Register containing destination type
Rn = Register file
Rp = Register pair
Rpd = Destination register pair
Rps = Source register pair
Rs = Register containing source byte

† All conditional jumps (opcodes 01–0F), BTJO, BTJZ, and DJNZ instructions use two additional cycles if the branch is taken. The BTJO, BTJZ, and DJNZ instructions have a relative address as the last operand.

16.5 Instruction Set Descriptions

The TMS370 instruction set contains 73 instructions that are supported by 246 opcodes. Each operation has an associated opcode. Some instructions, including those using the offset indirect addressing mode, have 16-bit (or dual) opcodes. Several bit manipulation instructions are constructed by the assembler out of other instructions to simplify writing the instructions and to enhance the readability of the program.

The following pages contain the individual instruction descriptions. The instructions are in alphabetical order by mnemonic. Refer to Table 16–1 on page 16-3 for the symbol definitions that are used in the instruction descriptions. Refer to subsection 3.2.2, on page 3-5, on the status register for a definition of the status bits.

Syntax **ADC** *s, Rd*

Execution (s) + (Rd) + (C) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
ADC	B,A	1	8	69	(B)+(A)+(C) → (A)
ADC	Rs,A	2	7	19	(Rs)+(A)+(C) → (A)
ADC	Rs,B	2	7	39	(Rs)+(B)+(C) → (B)
ADC	Rs,Rd	3	9	49	(Rs)+(Rd)+(C) → (Rd)
ADC	#iop8,A	2	6	29	iop8+(A)+(C) → (A)
ADC	#iop8,B	2	6	59	iop8+(B)+(C) → (B)
ADC	#iop8,Rd	3	8	79	iop8+(Rd)+(C) → (Rd)

Status Bits Affected

C Set to 1 on carryout of (s) + (Rd) + (C)
Z Set on result
N Set on result
V (C XOR N) AND (source [bit 7] XNOR destination [bit 7])

Description

ADC adds the contents of the source, the destination register, and the carry bit and stores the result in the destination register.

Adding a 0 to the destination register is equivalent to a conditional increment (increment on carry).

You can use ADC for the multiprecision addition of signed or unsigned integers. For example, the 16-bit integer in register pair (R2,R3) can be added to the 16-bit integer in (A,B) as follows:

```
        ADD  R3,B          ; Low-order bytes added
        ADC  R2,A          ; High-order bytes added
```

Examples

```
LABEL1  ADC R66,R117       ; Adds the contents of
                           ;   register 66, register
                           ;   117, and the carry bit,
                           ;   and stores the sum in
                           ;   register 117

        ADC   B,A          ; Adds the contents of
                           ;   register B, register A,
                           ;   and the carry bit, and
                           ;   stores the sum in
                           ;   register A

        ADC  #03Ch,R29     ; Adds #3Ch, contents of
                           ;   register 29, and the
                           ;   carry bit, and stores
                           ;   the sum in register 29
```

ADD *Add*

Syntax **ADD** *s, Rd*

Execution (s) + (Rd) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
ADD	B,A	1	8	68	(B)+(A) → (A)
ADD	Rs,A	2	7	18	(Rs)+(A) → (A)
ADD	Rs,B	2	7	38	(Rs)+(B) → (B)
ADD	Rs,Rd	3	9	48	(Rs)+(Rd) → (Rd)
ADD	#iop8,A	2	6	28	iop8+(A) → (A)
ADD	#iop8,B	2	6	58	iop8+(B) → (B)
ADD	#iop8,Rd	3	8	78	iop8+(Rd) → (Rd)

Status Bits Affected

C Set to 1 on carry-out of (s) + (Rd)
Z Set on result
N Set on result
V (C XOR N) AND (Source [bit 7] XNOR Destination [bit 7])

Description ADD adds the source value and destination register and stores the result in the destination register. You can use ADD for signed 2s complement or unsigned addition.

Examples

```
LABEL     ADD B,A           ; Adds the contents of
                            ;   registers B and A and
                            ;   stores the results in A

          ADD R7,A          ; Adds the contents of R7
                            ;   and A and stores the
                            ;   results in A

          ADD #TOTAL,R13    ; Adds the value of
                            ;   TOTAL to R13 and stores
                            ;   the result in R13
```

Syntax **AND** *s, Rd*

Execution (s) AND (Rd) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
AND	A,Pd	2	9	83	(A) AND (Pd) → (Pd)
AND	B,A	1	8	63	(B) AND (A) → (A)
AND	B,Pd	2	9	93	(B) AND (Pd) → (Pd)
AND	Rs,A	2	7	13	(Rs) AND (A) → (A)
AND	Rs,B	2	7	33	(Rs) AND (B) → (B)
AND	Rs,Rd	3	9	43	(Rs) AND (Rd) → (Rd)
AND	#iop8,A	2	6	23	iop8 AND (A) → (A)
AND	#iop8,B	2	6	53	iop8 AND (B) → (B)
AND	#iop8,Rd	3	8	73	iop8 AND (Rd) → (Rd)
AND	#iop8,Pd	3	10	A3	iop8 AND (Pd) → (Pd)

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description AND logically ANDs the two 8-bit source and destination operands. Each bit in the first operand is ANDed with the corresponding bit in the second operand. This is useful for clearing bits. If you need to clear a bit in the destination operand, put a 0 in the corresponding source bit. A 1 in a source bit will not change the corresponding destination bit.

Examples

```
LABEL     AND #01h,R12      ; Clear all bits in R12 except
                            ;   bit 0, which remains
                            ;   unchanged

          AND  R7,A         ; AND the contents of R7 to A
                            ;   and store the contents in A

          AND  B,P025       ; AND contents of B to P025,
                            ;   store the contents in P025
```

Syntax **BR** *XADDR*

Execution XADDR → (PC)

Options

inst	operands	bytes	cycles	opcode	operation
BR	label	3	9	8C	label → (PC)
BR	*Rp	2	8	9C	(Rp–1:Rp) → (PC)
BR	*label[B]	3	11	AC	label+(B) → (PC)
BR	*off8[Rp]	4	16	F4 EC	(Rp–1:Rp)+off8 → (PC)

Status Bits Affected None

Description BR branches to *any* location in memory, including the on-chip RAM. BR supports the following four program flow absolute addressing modes:

- Direct
- Indirect
- Indexed
- Offset Indirect

The powerful concept of computed GOTOs is supported by the BR *Rp instruction. Additionally, an indexed branch instruction of the form BR *TABLE[B] is an efficient way to execute one of several actions on the basis of a control input; this is similar to the Pascal CASE statement. The program can branch to up to 128 different jump statements. You can use BR to transfer control on character inputs, error codes, etc.

Examples

```
LABEL   BR   LABEL4       ; (PC) ← LABEL4

        BR   5432h         ; (PC) ← 5432h

        BR   *LABEL5[B]    ; (PC) ← LABEL5 + (B)

        BR   *1234h[B]     ; (PC) ← 1234h + (B)

        BR   *R12          ; (PC) ← (R11:R12)
                           ;    R12 = LSbyte

        BR   *56[R10]      ; (PC) ← 56 + (R9:R10)
                           ;   R10 = LSbyte
```

Syntax **BTJO** *s1,s2,off8*

Execution If (s1) AND (s2) ≠ 0, then PCN + off8 → (PC), else PCN → (PC)

Options

inst	operands	bytes	cycles†	opcode	jump if
BTJO	A,Pd,#off8	3	10/12	86	(A) AND (Pd) ≠ 0
BTJO	B,A,#off8	2	10/12	66	(B) AND (A) ≠ 0
BTJO	B,Pd,#off8	3	10/12	96	(B) AND (Pd) ≠ 0
BTJO	Rs,A,#off8	3	9/11	16	(Rs) AND (A) ≠ 0
BTJO	Rs,B,#off8	3	9/11	36	(Rs) AND (B) ≠ 0
BTJO	Rs,Rd,#off8	4	11/13	46	(Rs) AND (Rd)≠ 0
BTJO	#iop8,A,#off8	3	8/10	26	iop8 AND (A) ≠ 0
BTJO	#iop8,B,#off8	3	8/10	56	iop8 AND (B) ≠ 0
BTJO	#iop8,Rd,#off8	4	10/12	76	iop8 AND (Rd) ≠ 0
BTJO	#iop8,Pd,#off8	4	11/13	A6	iop8 AND (Pd) ≠ 0

† The number of cycles to the left of the slash are valid when the jump is not taken; the number of cycles to the right of the slash are valid when the jump is taken.

Status Bits Affected

C ← 0
N Set on (s) AND (d)
Z Set on (s) AND (d)
V ← 0

Description BTJZ jumps if a bit in source operand s1 has a 1 and its corresponding bit position in s2 also has a 1 (refer to the table below — jumps if any two corresponding bit positions in s1 and s2 are both 1). The s1 source operand can be used as a bit mask to test for one or more 1 bits in the specified register. The operands are not changed by this instruction. The table below contains examples wher jumps occur and don't occur.

s1	s2	Jump?
00000001	xxxxxxx0	No
00000001	xxxxxxx1	Yes
00000011	xxxxxx00	No
11110000	1000xxxx	Yes
11110000	1001xxxx	Yes

Examples

```
LABEL   BTJO #014,R4,#ISSET    ; Jump to ISSET if R4 (bit 2)
                               ;   or R4 (bit 4) is a 1

        BTJO #01,A,#LOOP       ; Jump to LOOP if bit 0
                               ;   of register A is a 1

        BTJO R37,R113,#START   ; Jump to START if any 1 bit
                               ;   of R113 corresponds to a
                               ;   1 bit in R37
```

Syntax **BTJZ** *s1,s2,off8*

Execution If (s1) AND NOT (s2) ≠ 0, then PCN + off8 → (PC), else PCN → (PC)

Options

inst	operands	bytes	cycles†	opcode	jump if ‡
BTJZ	A,Pd,#off8	3	10/12	87	(A) AND NOT(Pd) ≠ 0
BTJZ	B,A,#off8	2	10/12	67	(B) AND NOT(A) ≠ 0
BTJZ	B,Pd,#off8	3	10/12	97	(B) AND NOT(Pd) ≠ 0
BTJZ	Rd,A,#off8	3	9/11	17	(Rd) AND NOT(A) ≠ 0
BTJZ	Rd,B,#off8	3	9/11	37	(Rd) AND NOT(B) ≠ 0
BTJZ	Rs,Rd,#off8	4	11/13	47	(Rs) AND NOT(Rd) ≠ 0
BTJZ	#iop8,A,#off8	3	8/10	27	iop8 AND NOT (A) ≠ 0
BTJZ	#iop8,B,#off8	3	8/10	57	iop8 AND NOT (B) ≠ 0
BTJZ	#iop8,Rd,#off8	4	10/12	77	iop8 AND NOT (Rd) ≠ 0
BTJZ	#iop8,Pd,#off8	4	11/13	A7	iop8 AND NOT (Pd) ≠ 0

† The number of cycles to the left of the slash are valid when the jump is not taken; the number of cycles to the right of the slash are valid when the jump is taken.
‡ The instruction must have at least one 1 value bit in s1.

Status Bits Affected
C ← 0
N Set on (s) AND NOT (Rd)
Z Set on (s) AND NOT (Rd)
V ← 0

Description BTJZ jumps if a bit in source operand s1 has a 1 and its corresponding bit position in s2 has a 0 (refer to the table below). The s1 source operand can be used as a bit mask to test for 0 bits in the specified register. The operands are not changed by this instruction. The jump is calculated starting from the opcode of the instruction immediately after the BTJZ.

s1	s2	Jump?
00000001	xxxxxxx0	Yes
00000001	xxxxxxx1	No
11000000	11xxxxxx	No
11110000	0111xxxx	Yes
11110000	0110xxxx	Yes

Examples

```
LABEL     BTJZ  A,P23,#ZERO   ; If any 1 bit in reg. A
                               ;   corresponds to a 0 bit
                               ;   in P23, then jump to ZERO

          BTJZ #0FFh,A,#NEXT  ; If reg. A contains any 0
                               ;   bits, jump to NEXT

          BTJZ  R7,R15,#OUT   ; If any 0 bits in R15
                               ;   correspond to 1 bits
                               ;   in R7, jump to OUT
```

Syntax **CALL** *XADDR*

Execution

(SP) + 1 → (SP)
PCN MSbyte → ((SP))
(SP) + 1 → (SP)
PCN LSbyte → ((SP))
XADDR → (PC)

(The stack contains the address of the instruction immediately following the CALL.)

Options

inst	operands	bytes	cycles	opcode	operation
CALL	label	3	13	8E	label → (PC)
CALL	*label[B]	3	15	AE	label+(B) → (PC)
CALL	*off8[Rp]	4	20	F4 EE	(Rp–1:Rp)+off8 → (PC)
CALL	*Rp	2	12	9E	(Rp–1:Rp) → (PC)

Status Bits Affected None

Description CALL invokes a subroutine and pushes the PC contents on the stack. The operand indicates the starting address of the subroutine. The program flow addressing modes of the CALL instruction support the powerful transfer of control functions.

Examples

```
LABEL   CALL  LABEL4      ; Push PC; (PC) ← LABEL4

        CALL  5432h       ; Push PC; (PC) ← 5432h

        CALL  *LABEL5[B]  ; Push PC; (PC) ← LABEL5 + (B)

        CALL  *1234h[B]   ; Push PC; (PC) ← 1234h + (B)

        CALL  *R12        ; Push PC; (PC) ← (R11:R12)
                          ;   R12 = LSbyte

        CALL  *56[R10]    ; Push PC; (PC) ← 56 + (R9:R10)
                          ;   R10 = LSbyte
```

Syntax **CALLR** *XADDR*

Execution

(SP) + 1 → (SP)
PCN MSbyte → ((SP))
(SP) + 1 → (SP)
PCN LSbyte → ((SP))
XADDR + PCN → (PC)

Options

inst	operands	bytes	cycles	opcode	operation
CALLR	#off16	3	15	8F	off16 + PCN → (PC)
CALLR	label	3	15	8F	off16 + PCN → (PC)
CALLR	*Rp	2	14	9F	(Rp–1:Rp) + PCN → (PC)
CALLR	*off16[B]	3	17	AF	off16 + (B)+ PCN → (PC)
CALLR	*label[B]	3	17	AF	off16 + (B) + PCN → (PC)
CALLR	*off8[Rp]	4	22	F4 EF	(Rp–1:Rp) + off8 + PCN → (PC)

Status Bits Affected None

Description CALLR is similar to CALL, but it uses a value relative to the current program counter (PCN). The program flow relative addressing modes of the CALLR instruction support the powerful transfer of control functions. This is useful for relocatable code produced by linkers, compilers, or other high-level language structures. The assembler automatically calculates the correct offset value for the two modes by using labels in the operands.

Examples

Relative Direct Addressing:

```
LABEL     CALLR LABEL4        ; Push PC ; (PC) ← PCN + off16,
                              ;   off16 = LABEL4–PCN

          CALLR #5432h        ; Push PC ; (PC) ← PCN + 5432h
```

Relative Indexed Addressing:

```
          CALLR *LABEL5[B]    ; Push PC ; (PC) ← PCN +
                              ;   off16 +(B)
                              ;   off16=LABEL5 – PCN

          CALLR *1234h[B]     ; Push PC ; (PC) ← PCN +
                              ;   1234h + (B)
```

Relative Indirect Addressing:

```
          CALLR *R12          ; Push PC ; (PC) ← PCN +
                              ;   (R11:R12)
                              ;   R12=LSbyte
```

Relative Offset Indirect Addressing:

```
          CALLR *56[R10]      ; Push PC ; (PC) ← PCN
                              ;   + 56 + (R9:R10)
                              ;   R10=LSbyte
```

Syntax **CLR** *Rd*

Execution 0 → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
CLR	A	1	8	B5	0 → (A)
CLR	B	1	8	C5	0 → (B)
CLR	Rd	2	6	D5	0 → (Rd)

Status Bits Affected

C ← 0
N ← 0
Z ← 1
V ← 0

Description CLR clears or initializes to 0 any register, including registers A and B.

Examples

```
LABEL     CLR  B          ; Clear register B

          CLR  A          ; Clear register A

          CLR  R105       ; Clear register 105
```

Syntax **CLRC**

Execution Set status bits

Options

inst	operands	bytes	cycles	opcode
CLRC	none	1	9	B0

Status Bits Affected

C ← 0
N Set on value of register A
Z Set on value of register A
V ← 0

Description CLRC clears the carry flag. This instruction may be required before an arithmetic or rotate instruction. The logical and move instructions typically clear the carry bit. The CLRC opcode is equivalent to the TST A opcode.

Example

```
LABEL  CLRC    ; Clear the carry bit
```

Syntax **CMP** *s,d*

Execution (d) – (s) computed but not stored

Options

inst	operands	bytes	cycles	opcode	operation
Data Manipulation:					
CMP	B,A	1	8	6D	(A)–(B)
CMP	Rs,A	2	7	1D	(A)–(Rs)
CMP	Rs,B	2	7	3D	(B)–(Rs)
CMP	Rs,Rd	3	9	4D	(Rd)–(Rs)
CMP	#iop8,A	2	6	2D	(A)–iop8
CMP	#iop8,B	2	6	5D	(B)–iop8
CMP	#iop8,Rd	3	8	7D	(Rd)–iop8
CMP	&label,A	3	11	8D	(A)–(label)
CMP	*label[B],A	3	13	AD	(A)–(label+(B))
CMP	*off8[Rp],A	4	18	F4 ED	(A)–((Rp–1:Rp)+off8)
CMP	*Rp,A	2	10	9D	(A)–((Rp–1:Rp))
CMP	*off8[SP],A	2	8	F3	(A)–((SP)+off8)

Note: Operations are computed but not stored.
Status bits are set on results.

Status Bits Affected

C 1 if (d) ≥ (s)
N Sign of result
Z 1 if (d) = (s)
V (C XOR N) AND (Source [bit 7] XOR Destination [bit 7])

Description CMP compares the destination operand to the source operand and sets the status bits. The CMP instruction is usually used in conjunction with a jump instruction. Table 16–5 shows which jump instructions can be used on status conditions set by CMP execution. There are only seven possible outcomes of the status register after a compare instruction. The jump instructions JC and JHS are equivalent after a compare.

Table 16–5. Compare Instruction Examples—Status Bit Values

Operand Opcodes (s) (d)	Status Bits CNZV	JGE	JG	JL	JLE	JLO	JHS	JC	JNC	JN	JP	JEQ/ JZ	JPZ	JNE/ JNZ	JV	JNV
FF 00 81 00	0000	1	1	0	0	1	0	0	1	0	1	0	1	1	0	1
80 00 80 7F	0101	1	1	0	0	1	0	0	1	1	0	0	0	1	1	0
00 7F 20 30 90 A0	1000	1	1	0	0	0	1	1	0	0	1	0	1	1	0	1
7F 00 30 20 A0 90	0100	0	0	1	1	1	0	0	1	1	0	0	0	1	0	1
7F 80	1001	0	0	1	1	0	1	1	0	0	1	0	1	1	1	0
00 FF 00 81 00 80	1100	0	0	1	1	0	1	1	0	1	0	0	0	1	0	1
7F 7F	1010	1	0	0	1	0	1	1	0	0	0	1	1	0	0	1

Notes: 1) Signed Jumps: JGE, JG, JL, JLE.
Unsigned Jumps: JLO, JHS.
Test Bits: JC, JNC, JN, JP, JEQ/JZ, JPZ, JNE/JNZ, JV, JNZ
2) 1 = jump was taken; 0 = jump was not taken

Examples

```
LABEL   CMP  R13,R89       ; Set status bits on
                           ;   result of R89 minus R13

        CMP  R39,B         ; Set status bits on result
                           ;   of (B) minus R39

        CMP  #003h,A       ; Set status bits on result
                           ;   of (A) minus #03h

        CMP  *TABLE[B],A   ; Set status bits  on result
                           ;   of (A) minus (TABLE + (B))
```

Syntax **CMPBIT** *name*

Execution NOT <name> → <name>

Options

inst	operands	bytes	cycles	opcode	operation
CMPBIT	Rname	3	8	75	NOT (bit) → (bit) Register bits
CMPBIT	Pname	3	10	A5	NOT (bit) → (bit) PF bits

Status Bits Affected

C ← 0
N Set on result of (Mask XOR (s))
Z Set on result of (Mask XOR (s))
V ← 0

Description CMPBIT is an assembler-constructed instruction that conveniently complements the value of the named bit without having to specify a register or mask. This enhances the readability of the program. The CMPBIT instruction assembles to the instructions XOR #iop8,Rd or XOR #iop8,Pd. The name for the bit is defined by the .DBIT assembler directive.

Examples

```
INT1ENA   .DBIT 7,P017      ; Interrupt 1 bit is now
                            ;   named INT1ENA

TEST      .DBIT  4,R33      ; Bit 4 of register 33 is now
                            ;   named TEST

LABEL     CMPBIT TEST       ; Invert the value of the TEST
                            ;   bit.

          CMPBIT INT1ENA    ; Change the interrupt 1
                            ;   enabled condition.
```

Syntax **COMPL** *Rd*

Execution 0 – Rd → Rd

Options

inst	operands	bytes	cycles	opcode	operation
COMPL	A	1	8	BB	NOT(A)+1 → (A)
COMPL	B	1	8	CB	NOT(B)+1 → (B)
COMPL	Rd	2	6	DB	NOT(Rd)+1 → (Rd)

Status Bits Affected

C Set on result
N Set on result
Z Set on result
V 1 if 80h is the result; 0 otherwise

Description COMPL provides a logical or 2s complement of the operand. This is the equivalent to an inversion of all the bits followed by an increment. The instruction is useful in doing arithmetic with signed numbers.

Examples

```
LABEL     COMPL  A          ; Complement register A

          COMPL  B          ; Complement register B

          COMPL  R82        ; Complement register 82
```

Syntax **DAC** *s,Rd*

Execution (s) + (Rd) + (C) → (Rd), produces a decimal result

Options

inst	operands	bytes	cycles	opcode	operation
DAC	B,A	1	10	6E	(B)+(A)+(C) → (A)
DAC	Rs,A	2	9	1E	(Rs)+(A)+(C) → (A)
DAC	Rs,B	2	9	3E	(Rs)+(B)+(C) → (B)
DAC	Rs,Rd	3	11	4E	(Rs)+(Rd)+(C) → (Rd)
DAC	#iop8,A	2	8	2E	iop8+(A)+(C) → (A)
DAC	#iop8,B	2	8	5E	iop8+(B)+(C) → (B)
DAC	#iop8,Rd	3	10	7E	iop8+(Rd)+(C) → (Rd)

Status Bits Affected

C 1 if value of (s) + (Rd) + C > 99
N Set on result
Z Set on result
V Undefined

Description

DAC adds bytes in binary-coded decimal (BCD) form. Each byte is assumed to contain two BCD digits. DAC is not defined for non-BCD operands. DAC with an immediate operand of zero value is equivalent to a conditional increment of the destination operand (increment destination on carry).

The DAC instruction automatically performs a decimal adjust on the binary sum of (s) + (d) + C. The carry bit is added to facilitate adding multibyte BCD strings; therefore, the carry bit must be cleared before the first DAC instruction is executed.

Examples

```
LABEL     DAC #024h,A    ; If register A contains 097h
                         ;   and C = 0,then the final result
                         ;   put into A is 021h and the carry
                         ;   bit is set

          DAC  R55,R7    ; Add the BCD value of R55,
                         ;   and the carry bit to the
                         ;   BCD value of R7

          DAC  B,A       ; Add the carry bit to the
                         ;   BCD value in register B
                         ;   to register A
```

Syntax **DEC** *Rd*

Execution (Rd) – 1 → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
DEC	A	1	8	B2	(A)–1 → (A)
DEC	B	1	8	C2	(B)–1 → (B)
DEC	Rd	2	6	D2	(Rd)–1 → (Rd)

Status Bits Affected

C 0 if (Rd) decrements from 00h to FFh; 1 otherwise
N Set on result
Z Set on result
V 1 if (Rd) decrements from 80h to 7Fh; 0 otherwise

Description DEC subtracts 1 from any register. It is useful for counting and addressing byte arrays.

Examples

```
LABEL     DEC R102          ; Decrement R102 by 1

          DEC  A            ; Subtract 1 from the contents of
                            ;   register A

          DEC  B            ; Subtract 1 from the contents of
                            ;   register B
```

Syntax **DINT**

Execution 0 → (ST)

Options

inst	operands	bytes	cycles	opcode
DINT	none	2	6	F0 00

Status Bits Affected

C ← 0
N ← 0
Z ← 0
V ← 0
IE1 ← 0
IE2 ← 0

Description DINT simultaneously disables all interrupts. Since the interrupt enable flags are stored in the status register, the POP ST or RETI instructions may re-enable interrupts, even though a DINT instruction has been executed. During the interrupt service, the interrupt enable bit is automatically cleared after the old status register value has been pushed onto the stack. The DINT instruction is equal to the LDST #00 instruction.

Example

```
LABEL     DINT      ; Disable high- and low-level interrupts.
```

Syntax **DIV** *Rs, A*

Execution A:B/(Rs) → A(=quo), B(=rem)

Options

inst	operands	bytes	cycles	opcode	operation
DIV	Rs,A	3	55-63	F4 F8	(A:B)/(Rs) Quotient → A Remainder → B

Note: If overflow occurs, 14 cycles are used, and C,N,Z,V = 1

Status Bits Affected

C ← 0
N Set on results (register A)
Z Set on results (register A)
V ← 0

Description DIV divides the 16-bit value in the A:B register pair by the 8-bit value in the specified register. The resulting 8-bit quotient is stored in A with the remainder stored in B (note these are 8-bit values maximum). Overflow conditions are checked before the DIV instruction is executed; if an overflow is detected, the operands are left unchanged, the status bits C,N,Z, and V are set to 1, and the instruction is aborted. Execution time varies from 55–63 cycles, depending on the operands, with an overflow condition taking only 14 cycles. The average execution time is 57 cycles.

Example

```
LABEL   DIV  R10,A    ;R10 is divided into the
                      ;A:B register pair (A = MSbyte)

        JC OVERFLOW   ;Carry is 1 on overflow conditions
```

The *TMS370 Microcontroller Family Application Book* (SPNA017) contains example routines to enhance capabilities of the DIV instruction such as providing a 16-bit quotient from the division of a 16-bit concatenated number by an 8-bit number.

Syntax **DJNZ** *Rn,off8*

Execution (Rn) – 1 → (Rn)
If (Rn) ≠ 0, then PCN + (off8) → (PC), else PCN → (PC)

Type Single Operand

Options

inst	operands	bytes	cycles†	opcode	operation
DJNZ	A,#off8	2	10/12	BA	(A)–1 → (A), jump if (A) ≠ 0
DJNZ	B,#off8	2	10/12	CA	(B)–1 → (B), jump if (B) ≠ 0
DJNZ	Rd,#off8	3	8/10	DA	(Rd)–1 → (Rd), jump if (Rd) ≠ 0

† The number of cycles to the left of the slash are valid when the jump is not taken; the number of cycles to the right of the slash are valid when the jump is taken.

Status Bits Affected None

Description DJNZ is used for loop control. It combines the DEC and the JNZ instructions, providing a faster and more compact instruction. DJNZ does not change the status bits.

Examples

```
LABEL     DJNZ  R15,THERE   ; Decrement R15. If R15 ≠ 0,
                            ;   jump to THERE

          DJNZ  A,AGAIN     ; Decrement A; if A ≠ 0,
                            ;   jump to AGAIN

          DJNZ  B,BACK      ; Decrement B; if B ≠ 0,
                            ;   jump to BACK
```

Syntax **DSB** *s,Rd*

Execution (Rd) – (s) – 1 + (C) → (Rd) (decimal result)

Options

inst	operands	bytes	cycles	opcode	operation
DSB	B,A	1	10	6F	(A)–(B)–1+(C) → (A)
DSB	Rs,A	2	9	1F	(A)–(Rs)–1+(C) → (A)
DSB	Rs,B	2	9	3F	(B)–(Rs)–1+(C) → (B)
DSB	Rs,Rd	3	11	4F	(Rd)–(Rs)–1+(C) → (Rd)
DSB	#iop8,A	2	8	2F	(A)–iop8–1+(C) → (A)
DSB	#iop8,B	2	8	5F	(B)–iop8–1+(C) → (B)
DSB	#iop8,Rd	3	10	7F	(Rd)–iop8–1+(C) → (Rd)

Status Bits Affected

C 1 if no borrow required, 0 if borrow required
N Set on result
Z Set on result
V Undefined

Description DSB performs multiprecision BCD subtraction. The DSB instruction with an immediate operand of zero value is equivalent to a conditional decrement of the destination operand, depending on the carry bit. The carry bit functions as a no-borrow bit; if a no-borrow is required, the carry bit should be set to 1. You can accomplish this by executing the SETC instruction. The DSB instruction is undefined for non-BCD operands.

Examples

```
LABEL     DSB  R15,R76      ; R76 minus R15 minus 1 plus
                            ;   the carry bit is stored in R76

          DSB  A,B         ; Register B minus register
                            ;   A minus 1 plus the carry
                            ;   bit is stored in
                            ;   register B

          DSB  #0,R5       ; R5 - 1 →  R5, if C = 0
                            ; R5 →  R5, if C = 1
```

Syntax **EINT**

Execution 0Ch → (ST)

Options

inst	operands	bytes	cycles	opcode
EINT	none	2	6	F0 0C

Status Bits Affected

C ← 0
N ← 0
Z ← 0
V ← 0
IE1 ← 1
IE2 ← 1

Description EINT simultaneously enables all global interrupts. Since the interrupt enable flags are stored in the status register, the POP ST or RTI instructions may disable interrupts, even though an EINT instruction has been executed. During the interrupt service, the interrupt enable bit is automatically cleared after the old status register value has been pushed onto the stack. Thus, the EINT instruction must be included inside the interrupt service routine to permit nested or multilevel interrupts. This instruction is equivalent to the LDST #00Ch instruction.

Example

```
LABEL     EINT              ; All interrupts are enabled
```

EINTH *Enable High-Level Interrupts*

Syntax **EINTH**

Execution 04h → (ST)

Options

inst	operands	bytes	cycles	opcode
EINTH	none	2	6	F0 04

Status Bits Affected

C ← 0
N ← 0
Z ← 0
V ← 0
IE1 ← 1
IE2 ← 0

Description EINTH is similar to the EINT instruction but enables only high-level (1) interrupts and disables low-level interrupts. This assembles to the LDST #04h instruction.

Example

```
LABEL     EINTH       ; All level 1 interrupts are enabled
```

Syntax **EINTL**

Execution 08h → (ST)

Options

inst	operands	bytes	cycles	opcode
EINTL	none	2	6	F0 08

Status Bits Affected

C ← 0
N ← 0
Z ← 0
V ← 0
IE1 ← 0
IE2 ← 1

Description EINTL is similar to the EINT instruction but enables only low-level (2) interrupts while disabling high-level interrupts. This assembles to the LDST #08h instruction.

Example

```
LABEL     EINTL       ; All level 2 interrupts are enabled
```

Syntax **IDLE**

Execution (PC) + 1 → (PC) after return from interrupt

Options

inst	operands	bytes	cycles	opcode
IDLE	none	1	6 (minimum)	F6

Status Bits Affected None

Description The IDLE instruction causes the device to enter one of three modes: halt, standby, or idle. Two of these modes, halt and standby, use only a fraction of the normal operating power.

- In standby mode, the on-chip oscillator and timer 1 module remain active.
- In halt mode, the oscillator is off, and the chip consumes the least amount of power.

If you execute an IDLE instruction when low-power modes are disabled through a programmable contact (mask-ROM devices only), the device will *always* enter the idle mode.

Appropriate interrupts must be enabled before the device enters idle mode. For more information on the low-power or idle modes, refer to Section 4.2.

Example

```
LABEL     IDLE              ; Enter idle mode and
                            ;   wait for interrupt
```

Syntax **INC** *Rd*

Execution (Rd) + 1 → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
INC	A	1	8	B3	(A)+1 → (A)
INC	B	1	8	C3	(B)+1 → (B)
INC	Rd	2	6	D3	(Rd)+1 → (Rd)

Status Bits Affected

C 1 if (Rd) incremented from FFh to 00h; 0 otherwise
N Set on result
Z Set on result
V 1 if (Rd) incremented from 7Fh to 80h; 0 otherwise

Description INC increments the value of any register. It is useful for incrementing counters.

Examples

```
LABEL     INC  A          ; Increment register A by 1

          INC  B          ; Increment register B by 1

          INC  R43        ; Increment register 43 by 1
```

Syntax **INCW** *#iop8,Rp*

Execution (Rp) + #iop8 → (Rp)

Options

inst	operands	bytes	cycles	opcode	operation
INCW	#iop8,Rp	3	11	70	iop8+(Rp–1:Rp) → (Rp–1:Rp) iop8 = 8-bit immediate operand

Status Bits Affected

C Set to 1 on carry out of iop8 + (Rp)
N Set on result
Z Set on MSbyte
V (C XOR N) AND (MSB iop8 XNOR MSB (Rd))

Description INCW increments the value of any register pair by the amount specified. The register pair can be incremented by as much as 127 or decremented by as much as 128. This instruction is useful for incrementing counters into large tables. The iop8 is sign-extended to perform 16-bit 2s-complement addition. The JC and JNC are commonly used after the INCW instruction for loop control.

Examples

```
LABEL     INCW #1,R10       ; Increment R9:R10 by 1

          INCW #–1,R10      ; Decrement register R9:R10 by 1

          INCW #064h,R255   ; Increment register pair
                            ;   R254:R255 by #064h
```

Syntax **INV** *Rd*

Execution NOT(Rd) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
INV	A	1	8	B4	NOT(A) → (A)
INV	B	1	8	C4	NOT(B) → (B)
INV	Rd	2	6	D4	NOT(Rd) → (Rd)

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description INV performs a 1s complement of the operand. A 1s complement inverts the value of every bit in the register. You can perform a 2s complement of the operand by following the INV instruction with an increment (INC) or by simply using the COMPL instruction.

Examples

```
LABEL     INV A            ; Invert register A (0s become 1s,
                           ;   1s become 0s)

          INV B            ; Invert register B

          INV R82          : Invert register 82
```

JBIT0 *Jump If Bit = 0*

Syntax **JBIT0** *name, #off8*

Execution If bit (name) = 0, then PCN + off8 → (PC), else PCN → PC

Options

inst	operands	bytes	cycles	opcode
JBIT0	Rname,#off8	4	10	77
JBIT0	Pname,#off8	4	11	A7

Note: Add two cycles if jump is taken

Status Bits Affected

C ← 0
N Set on (s) AND NOT (Rd)
Z Set on (s) AND NOT (Rd)
V ← 0

Description The JBIT0 is an assembler-constructed instruction that conveniently jumps to the label if the value of the named bit is zero. This enhances the readability of the program because the source does not have to specify both the register containing the bit and a mask. JBIT0 is assembled to BTJZ #iop8,Rd,label or BTJZ #iop8,Pd,label. The name for the bit is defined by the .DBIT assembler directive.

Examples

```
MCDATA    .DBIT 2,P010        ; MC data in bit 2 of
                              ;   SCCR0 (P010) is now
                              ;   named MCDATA
                              ;
 BIT4     .DBIT  4,R3         ; Bit 4 of register 3 is
                              ;   now named BIT4
                              ;
          JBIT0 BIT4,#THERE   ; Jump to THERE if bit 4 in
                              ;   register 3 is zero.
                              ;
          JBIT0 MCDATA,#HERE  ; Jump to HERE if the MC pin
                              ;   is zero
```

Syntax **JBIT1** *name, #off8*

Execution bit (name) = 1, then PCN + off8 → (PC), else PCN → (PC)

Options

inst	operands	bytes	cycles	opcode	operation
JBIT1	Rname,#off8	4	10	76	register bits
JBIT1	Pname,#off8	4	11	A6	peripheral bits

Note: Add two cycles if jump is taken.

Status Bits Affected

C ← 0
N Set on (s) AND (Rd)
Z Set on (s) AND (Rd)
V ← 0

Description The JBIT1 is an assembler-constructed instruction that conveniently jumps to the label if the value of the named bit is one. This instruction enhances the readability of the program because the source does not have to specify both the register containing the bit and a mask. This instruction assembles to BTJO #iop8,Rd,label or BTJO #iop8,Pd,label. The name for the bit is defined by the .DBIT assembler directive.

Examples

```
BUSYP     .DBIT 7,P01C            ; Busy bit in PEECTL
                                  ;   (program EEPROM) is now
                                  ;   named BUSYP
                                  ;
BIT0      .DBIT 0,R100            ; Bit 0 of register 100 is
                                  ;   now named BIT0
                                  ;
LABEL     JBIT1 BIT0, #THERE      ; Jump to THERE if bit 0 in
                                  ;   register 100 is a one.
                                  ;
          JBIT1 BUSYP,#HERE       ; Jump to HERE if the
                                  ;   program EEPROM is busy.
```

Syntax **J*cnd*** *#off8*

Execution If tested condition is true, (PCN) + off8 → (PC), else PCN → (PC)

Status Bits Affected None

Description The J*cnd* instructions (listed in Table 16–6) are commonly used after a CMP instruction to branch according to the relative values of the operands tested. After MOV operations, a JZ or JNZ can be used to test whether the value moved was equal to zero; in this case, JN and JPZ are used to test the sign bit of the value moved. In addition, the program can check the overflow bit V after executing an arithmetic instruction with the JV or JNV instructions.

All J*cnd* instructions are two bytes in length and require five cycles to execute; however, if the jump is taken, the instruction requires seven cycles.

Table 16–6. Conditional-Jump Instructions

Instruction	Mnemonic	Opcode	C‡	N‡	Z‡	V‡	Operation
Jump if Carry	JC	03	1	x	x	x	
Jump if No Carry	JNC	07	0	x	x	x	
Jump if Equal	JEQ	02	x	x	1	x	
Jump if Not Equal	JNE	06	x	x	0	x	
Jump if Nonzero	JNZ	06	x	x	0	x	
Jump if Zero	JZ	02	x	x	1	x	
Jump if Lower	JLO	0F	0	x	0	x	
Jump if Higher or Same	JHS	0B	–	x	–	x	(C = 1) OR (Z = 1)
							—Signed Operation—
Jump if Greater	JG	0E	x	–	–	–	Z OR (N XOR V) = 0
Jump if Greater or Equal	JGE	0D	x	–	x	–	N XOR V = 0
Jump if Less	JL	09	x	–	x	–	N XOR V = 1
Jump if Less or Equal	JLE	0A	x	–	–	–	Z OR (N XOR V) = 1
Jump if Negative	JN	01	x	1	x	x	
Jump if Positive	JP	04	x	0	0	x	
Jump if Positive or Zero	JPZ	05	x	0	x	x	
Jump if No Overflow	JNV	0C	x	x	x	0	
Jump if Overflow	JV	08	x	x	x	1	

‡ Status Values:
- 0 Status bit always cleared.
- 1 Status bit always set.
- x Status bit cleared or set on results.
- – Status bit not affected.

These two-instruction jump sequences are used in place of the single conditional jumps for the numbers shown in the tables. ($ is the current PC value.)

Signed Number Jumps			Unsigned Number Jumps		
Condition	**True**	**False**	**Condition**	**True**	**False**
d < s	JL	JGE	d < s	JLO	JHS
d ≤ s	JLE	JG	d ≤ s	Note 2	Note 3
d = s	JEQ	JNE	d = s	JEQ	JNE
d ≥ s	JGE	JL	d ≥ s	JHS	JLO
d > s	JG	JLE	d > s	Note 3	Note 2
Negative	JN	JPZ			
Positive	JP	Note 1			
Positive or 0	JPZ	JN			

Notes: 1) JZ LABEL
JN LABEL

2) JEQ LABEL
JLO LABEL

3) JEQ $+4
JHS LABEL

Status Bit Jumps		
Bits	**True**	**False**
C	JC	JNC
N	JN	JPZ
Z	JZ	JNZ
V	JV	JNV

Examples

```
LABEL     JNC  TABLE      ; If the carry bit is clear,
                          ;   jump to TABLE

          JP   HERE       ; If the negative and zero flags
                          ;   are clear, jump to HERE

          JZ   NEXT       ; If the zero flag is set, jump
                          ;   to NEXT
```

Syntax **JMP** *#off8*

Execution PCN + off8 → (PC)

Options

inst	operands	bytes	cycles	opcode	operation
JMP	#off8	2	7	00	PCN+off8 → (PC)

Status Bits Affected None

Description JMP jumps unconditionally to the address specified in the operand. The second byte of the JMP instruction contains the 8-bit relative address of the operand. The operand address must therefore be within –128 to +127 bytes of the location of the instruction following the JMP instruction. The assembler will indicate an error if the target address is beyond –128 to +127 bytes from the next instruction. For a longer jump, you can use the BR (branch) or the JMPL instruction.

Example

```
LABEL     JMP #THERE        ; Load the PC with the address
                            ;   of THERE
```

Syntax **JMPL** *XADDR*

Execution PCN + D → (PC)

Options

inst	operands	bytes	cycles	opcode	operation
JMPL	#off16	3	9	89	off16+PCN → (PC)
JMPL	label	3	9	89	off16+PCN → (PC)
JMPL	*Rp	2	8	99	(Rp–1:Rp)+PCN → (PC)
JMPL	*off16[B]	3	11	A9	off16+(B) +PCN → (PC)
JMPL	*label[B]	3	11	A9	off16+(B)+PCN → (PC)
JMPL	*off8[Rp]	4	16	F4 E9	(Rp–1:Rp)+off8+PCN → (PC)

Status Bits Affected None

Description JMPL is similar to the the JMP instruction but generates a 16-bit (instead of 8-bit) signed offset to the program counter.

Example

```
LABEL     JMPL   LABEL4     ; (PC)← PCN + offset
                            ;   offset=LABEL4–PCN

          JMPL   #5432h     ; (PC)← PCN + 5432h

          JMPL   *LABEL5[B] ; (PC)← PCN + off8 + (B)
                            ;   offset=LABEL5 – PCN

          JMPL   *1234h[B]  ; (PC)← PCN + 1234h + (B)

          JMPL   *R12       ; (PC)← PCN + (R11:R12)
                            ;   R12=LSbyte

          JMPL   *56[R10]   ; (PC)← PCN + 56 + (R9:R10)
                            ;   R10 = LSbyte

          JMPL   *–2[R10]   ; (PC)← PCN – 2 + (R9:R10)
                            ;   R10 = LSbyte
```

LDSP *Load Stack Pointer*

Syntax **LDSP**

Execution (B) → (SP)

Options

inst	operands	bytes	cycles	opcode
LDSP	none	1	7	FD

Status Bits Affected None

Description LDSP copies the contents of register B to the stack pointer (SP). Use LDSP to initialize the stack pointer.

Example

```
MOV       #080h,B          ; Register B = SP value.
                           ;
LABEL     LDSP             ; Copy register B to the stack
                           ;   pointer.
```

Syntax **LDST** *#iop8*

Execution (iop8) → (ST)

Options

inst	operands	bytes	cycles	opcode	operation
LDST	#iop8	2	6	F0	iop8 → (ST)

Status Bits Affected

C Set on value loaded
N Set on value loaded
Z Set on value loaded
V Set on value loaded
IE1 Set on value loaded
IE2 Set on value loaded

Description The LDST copies the immediate value operand to the status register (ST). Any combination of bits can be loaded into the status register using this command. Some instructions such as EINT, EINTL, EINTH, DINT are assembled into this instruction. Status bits are defined in detail in subsection 3.2.2 on page 3-5.

Example

```
LABEL     LDST #08Ch        ; Copy immediate value to
                            ;   the status register and
                            ;   set IE2 bit
```

Syntax **MOV** *s,d*

Execution (s) → (d)

Options

inst	operands	bytes	cycles	opcode	operation
Register:					
MOV	A,B	1	9	C0	(A) → (B)
MOV	A,Rd	2	7	D0	(A) → (Rd)
MOV	B,A	1	8	62	(B) → (A)
MOV	B,Rd	2	7	D1	(B) → (Rd)
MOV	Rs,A	2	7	12	(Rs) → (A)
MOV	Rs,B	2	7	32	(Rs) → (B)
MOV	Rs,Rd	3	9	42	(Rs) → (Rd)
MOV	#iop8,A	2	6	22	iop8 → (A)
MOV	#iop8,B	2	6	52	iop8 → (B)
MOV	#iop8,Rd	3	8	72	iop8 → (Rd)
Peripheral:					
MOV	A,Pd	2	8	21	(A) → (Pd)
MOV	B,Pd	2	8	51	(B) → (Pd)
MOV	Rs,Pd	3	10	71	(Rs) → (Pd)
MOV	Ps,A	2	8	80	(Ps) → (A)
MOV	Ps,B	2	8	91	(Ps) → (B)
MOV	Ps,Rd	3	10	A2	(Ps) → (Rd)
MOV	#iop8,Pd	3	10	F7	iop8 → (Pd)
Direct:					
MOV	A,*label	3	10	8B	(A) → (label)
MOV	*label,A	3	12	8A	(label) → (A)
Indirect:					
MOV	A,*Rp	2	9	9B	(A) → ((Rp-1:Rp))
MOV	*Rp,A	2	9	9A	((Rp-1:Rp)) → (A)
Indexed:					
MOV	A,*label[B]	3	12	AB	(A) → (label+(B))
MOV	*label[B],A	3	12	AA	(label+(B)) → (A)
Offset Indirect:					
MOV	A,*off8[Rp]	4	17	F4 EB	(A) → (off8+(Rp-1:Rp))
MOV	*off8[Rp],A	4	17	F4 EA	(off8+(Rp-1:Rp)) → (A)
Stack Pointer Relative:					
MOV	A,*off8[SP]	2	7	F2	(A) → (off8+(SP))
MOV	*off8[SP],A	2	7	F1	(off8+(SP)) → (A)

Status Bits Affected

C ← 0
N Set on value loaded
Z Set on value loaded
V ← 0

Description

MOV transfers values within the memory space. Immediate values can be loaded directly into the registers. In direct, indirect, indexed, offset indirect, and stack pointer relative addressing modes, the processor must use register A. A MOV instruction that uses register A or B as an operand requires fewer bytes. When the MOV Pn,Rn and MOV Rn,Pn instructions are assembled into machine code, their operands are reversed.

Examples

```
MOV   A,B          ; Move the contents of register
                   ;   A to register B

MOV   R32,R105     ; Move the contents of register
                   ;   32 to register 105

MOV   #010h,R3     ; Move #010h to register 3

MOV   A,*LABEL[B]  ; Move the contents of register A to the
                   ;   location LABEL+B

MOV   A,*2Fh[R32]  ; Move the contents of register A to the
                   ;   location 002Fh+(R31:R32)

MOV   *ROF,A       ; Use the contents of the register
                   ;   pair ROE:ROF as address.
                   ;   Move the contents of that
                   ;   address to register A

MOV   LABEL,A      ; Move the contents of the location at
                   ;   LABEL to register A
```

Syntax **MOVW** *s,Rpd*

Execution (s) → (Rpd)

Options

inst	operands	bytes	cycles	opcode	operation
MOVW	Rps,Rpd	3	12	98	(Rps–1:Rps) → (Rpd–1:Rpd)
MOVW	#iop16,Rpd	4	13	88	iop16 → (Rpd–1:Rpd)
MOVW	#iop16[B],Rpd	4	15	A8	(B) + iop16 → (Rpd–1:Rpd)
MOVW	#off8[Rps],Rpd	5	20	F4 E8	(Rp–1:Rp)+iop8 → (Rpd–1:Rpd)

Status Bits Affected

C ← 0
N Set on MSbyte moved
Z Set on MSbyte moved
V ← 0

Description

MOVW moves a 2-byte value to the register pair indicated by the destination register number. (Note that Rpd should be greater than 0.) The destination points to the LSbyte of the destination register pair. The source can be a 16-bit constant, another register pair, or an indexed address.

For the indexed address, the source must be of the form “#ADDR[B]” where ADDR is a 16-bit constant or address. This 16-bit value is added (via 16-bit addition) to the contents of register B, and the result is placed in the destination register pair. This stores an indexed address into a register pair for use later in indirect addressing mode. This is not to be confused with the program flow addressing instruction *LABEL[B].

Examples

```
LABEL   MOVW  #1234h,R3       ; 1234h → (R2:R3)

        MOVW  R5,R3           ; (R4:R5) → (R2:R3)
                              ;   R5,R3 = LSbyte

        MOVW #TAB[B],R3       ; TAB + (B) → (R2:R3)
                              ;   R3 = LSbyte

        MOVW #127[R200],R34   ; 127 + (R199:R200) →
                              ;   (R33:R34)

        MOVW #-128[R200],R34  ; (R199:R200) - 128 →
                              ;   (R33:R34)
```

Syntax **MPY** *s,Rn*

Execution (s) × (Rn) → (A:B)
Result always stored in A,B; A = MSbyte

Options

inst	operands	bytes	cycles	opcode	operation
MPY	B,A	1	47	6C	(A) X (B) → (A:B)
MPY	Rs,A	2	46	1C	(A) X (Rs) → (A:B)
MPY	Rs,B	2	46	3C	(B) X (Rs) → (A:B)
MPY	Rs,Rd	3	48	4C	(Rd) X (Rs) → (A:B)
MPY	#iop8,A	2	45	2C	(A) X iop8 → (A:B)
MPY	#iop8,B	2	45	5C	(B) X iop8 → (A:B)
MPY	#iop8,Rd	3	47	7C	(Rd) X iop8 → (A:B)

Status Bits Affected
C ← 0
N Set on MSbyte of results (register A)
Z Set on MSbyte of results (register A)
V ← 0

Description MPY performs an 8-bit multiply for a general source and destination operand. The 16-bit result is placed in the A, B register pair with the most significant byte in A. Multiplying by a power of two is a convenient means of performing double-byte shifts.

- If a double-byte shift is three places or less, then it can be faster to use RLC or RRC instead of multiply.
- To shift a single byte, it is almost always faster to use RLC or RRC.

Examples

```
LABEL     MPY  R3,A          ; Multiply (R3) with (A), store
                             ;   result in A, B register pair

          MPY #032h,B       ; Multiply 32h with (B), store
                             ;   in register pair A, B

          MPY  R12,R7       ; Multiply (R12) with (R7) and
                             ;   store in A, B register pair
```

NOP *No Operation*

Syntax **NOP**

Execution (PC) + 1 → (PC)

Options

inst	operands	bytes	cycles	opcode
NOP	none	1	7	FF

Status Bits Affected None

Description NOP is useful as a pad instruction during program development to “patch out” unwanted or erroneous instructions or to leave room for code changes during development. It is also useful in software timing loops. NOP uses seven clock cycles.

Example

```
LABEL     NOP              ; Use 7 cycles of time
```

Syntax **OR** *s,Rd*

Execution (s) OR (Rd) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
OR	A,Pd	2	9	84	(A) OR (Pd) → (Pd)
OR	B,A	1	8	64	(B) OR (A) → (A)
OR	B,Pd	2	9	94	(B) OR (Pd) → (Pd)
OR	Rs,A	2	7	14	(Rs) OR (A) → (A)
OR	Rs,B	2	7	34	(Rs) OR (B) → (B)
OR	Rs,Rd	3	9	44	(Rs) OR (Rd) → (Rd)
OR	#iop8,A	2	6	24	iop8 OR (A) → (A)
OR	#iop8,B	2	6	54	iop8 OR (B) → (B)
OR	#iop8,Rd	3	8	74	iop8 OR (Rd) → (Rd)
OR	#iop8,Pd	3	10	A4	iop8 OR (Pd) → (Pd)

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description OR logically ORs the two operands. The OR operation is used to set bits in a register. If a register needs a 1 in the destination, then a 1 is placed in the corresponding bit location in the source operand.

Examples

```
LABEL     OR   A,R12        ; OR register A with R12,
                            ;   store in R12

          OR  #00Fh,A       ; Set lower nibble of A to 1s,
                            ;   leave upper nibble unchanged

          OR  R8,B          ; OR (R8) with (B), store in B
```

Syntax **POP** *d*

Execution
((SP)) → (d)
(SP) – 1 → (SP)
(Move value then decrement SP)

Options

inst	operands	bytes	cycles	opcode	operation
POP	A	1	9	B9	((SP)) → (A); (SP) – 1 → (SP)
POP	B	1	9	C9	((SP)) → (B); (SP) – 1 → (SP)
POP	Rd	2	7	D9	((SP)) → (Rd); (SP) – 1 → (SP)
POP	ST	1	8	FC	((SP)) → (ST); (SP) – 1 → (SP)

Status Bits Affected
C ← 0
N Set on value POPed
Z Set on value POPed
V ← 0

> **Note:**
>
> POP ST affects all status bits.

Description POP pulls a value from the top of the stack. The stack can be used to save or to pass values between routines. POP ST can replace the status register with the contents on the stack. This 1-byte instruction is usually executed in conjunction with a previously performed PUSH ST instruction.

Examples

```
LABEL     POP  R32     ; Load R32 with value on top of stack

          POP  ST      ; Load status register with
                       ;     value on top of stack
```

Syntax **PUSH** *s*

Execution
(SP) + 1 → (SP)
(s) → ((SP))
(Increment SP then move value)

Options

inst	operand	bytes	cycles	opcode	operation
PUSH	A	1	9	B8	(SP) + 1 → (SP); (A) → ((SP))
PUSH	B	1	9	C8	(SP) + 1 → (SP); (B) → ((SP))
PUSH	Rs	2	7	D8	(SP) + 1 → (SP); (Rs) → ((SP))
PUSH	ST	1	8	FB	(SP) + 1 → (SP); (ST) → ((SP))

Status Bits Affected
C ← 0
N Set on value PUSHed
Z Set on value PUSHed
V ← 0

> **Note:**
> Status bits are unchanged for PUSH ST.

Description PUSH places a value on the top of the stack. The stack is used to save or pass values between routines.

The status register can be pushed on the stack with the statement PUSH ST. This 1-byte instruction is usually executed in conjunction with a subsequently performed POP ST instruction.

Examples

```
LABEL     PUSH  A          ; Move (A) to top of stack

          PUSH  ST         ; Move status to top of stack
```

Syntax **RL** *Rd*

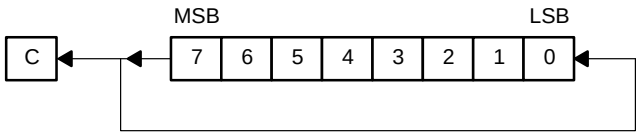
Execution Bit(n) → Bit(n+1)
Bit(7) → Bit(0) and carry

Options

inst	operands	bytes	cycles	opcode
RL	A	1	8	BE
RL	B	1	8	CE
RL	Rd	2	6	DE

Status Bits Affected
C Set to bit 7 of the original operand
N Set on result
Z Set on result
V ← 0

Description RL circularly shifts the destination contents one bit to the left. The MSB is shifted into the LSB; the carry bit is also set to the original MSB value.



For example, if register B contains the value 93h, then RL changes the contents of B to 27h and sets the carry bit.

Examples

```
LABEL   RL R102   ; Circularly rotate reg. R102 left one bit

        RL  A     ; Circularly rotate reg. A left one bit

        RL  B     ; Circularly rotate reg. B left one bit
```

Syntax **RLC** *Rd*

Execution
Bit(n) → Bit(n+1)
Carry → Bit(0)
Bit(7) → Carry

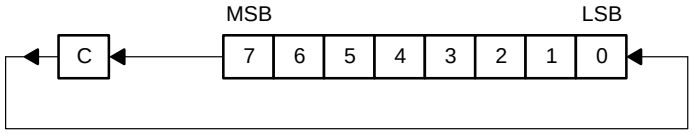
Options

inst	operands	bytes	cycles	opcode
RLC	A	1	8	BF
RLC	B	1	8	CF
RLC	Rd	2	6	DF

Status Bits Affected
C Set to bit 7 of the original operand
N Set on result
Z Set on result
V ← 0

Description

RLC circularly shifts the destination contents one bit to the left and through the carry. The original carry bit contents shift into the LSB, and the original MSB shifts into the carry bit.



For example, if register B contains the value 93h and the carry bit is a zero, then the RLC instruction changes the operand value to 26h and the carry bit to one.

Rotating left effectively multiplies the value by 2. Using multiple rotations, any power of 2 (2, 4, 8, 16,...) can be achieved. This type of multiplication can be faster than the MPY (multiply) instruction. This instruction is also useful in rotations where a value is contained in more than one byte (such as an address) or in multiplying a large multibyte number by 2. Take care to ensure that the carry bit is at the proper value. The SETC or CLRC instructions can be used to set up the correct value.

Examples

```
LABEL     RLC  R72    ; Circular rotate reg. R72 left one bit
                      ;    with MSbit –-> Carry bit

          RLC  A      ; Circular rotate reg. A left one bit
                      ;    with MSbit –-> Carry bit

          RLC  B      ; Circular rotate reg. B left one bit
                      ;    with MSbit –-> Carry bit
```

Syntax **RR** *Rd*

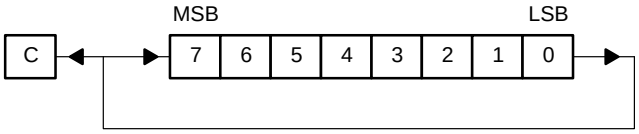
Execution
Bit(n+1) → Bit(n)
Bit(0) → Bit (7) and carry

Options

inst	operands	bytes	cycles	opcode
RR	A	1	8	BC
RR	B	1	8	CC
RR	Rd	2	6	DC

Status Bits Affected
C Set to bit 0 of the original value
N Set on result
Z Set on result
V ← 0

Description
RR circularly shifts the destination contents one bit to the right. The LSB is shifted into the MSB, and the carry bit is also set to the original LSB value.



For example, if register B contains the value 93h, then the RR B instruction changes the contents of B to C9h and sets the carry status bit.

Examples

```
LABEL     RR  A     ; Circular rotate right one bit the A reg.
                    ;   with LSbit --> Carry bit
          RR  B     ; Circular rotate right one bit the B reg.
                    ;   with LSbit --> Carry bit
          RR  R56   ; Circular rotate right one bit register
                    ;   R56 with LSbit --> Carry bit
```

Syntax **RRC** *Rd*

Execution
Bit(n+1) → Bit(n)
Carry → Bit(7)
Bit(0) → Carry

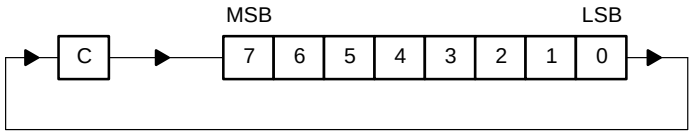
Options

inst	operands	bytes	cycles	opcode
RRC	A	1	8	BD
RRC	B	1	8	CD
RRC	Rd	2	6	DD

Status Bits Affected
C Set to bit 0 of the original value
N Set on result
Z Set on result
V ← 0

Description

RRC circularly shifts the destination contents one bit to the right through the carry. The carry bit contents shift into the MSB, and the LSB shifts into the carry bit.



For example, if register B contains the value 93h and the carry bit is zero, then RRC changes the operand value to 49h and sets the carry bit.

When the carry bit is 0, this instruction effectively divides the value by 2. A value of 80h becomes 40h. Repetitive use of this instruction can divide the value by any power of 2. Care must be taken to ensure the correct value in the carry bit.

Examples

```
LABEL     RRC R32      ; Rotate right R32; LSbit –-> Carry and
                       ;   Carry bit to R32 MSbit
          RRC A        ; Rotate right reg, A; LSbit –-> Carry
                       ;   Carry bit to reg. A MSbit
          RRC B        ; Rotate right reg. B; LSbit –-> Carry
                       ;   Carry bit to reg. B MSbit
```

Syntax **RTI**

Execution

((SP)) → (PC LSbyte)
(SP) – 1 → (SP)
((SP)) → (PC MSbyte)
(SP) – 1 → (SP)
((SP)) → (ST)
(SP) – 1 → (SP)

Options

inst	operands	bytes	cycles	opcode
RTI	none	1	12	FA

Status Bits Affected Status register is loaded from the stack

Description RTI is typically the last instruction executed in an interrupt service routine. RTI restores the status register to the state it was in immediately before the interrupt occurred and branches back to the program at the instruction boundary where the interrupt occurred. In an interrupt routine, there must be an equal number of POPs and PUSHs so that the stack pointer is pointing to the correct return address and not some other data.

Example

```
LABEL RTI    ; Return to main program from interrupt routine
```

Syntax **RTS**

Execution
((SP)) → (PC LSbyte)
(SP) – 1 → (SP)
((SP)) → (PC MSbyte)
(SP) – 1 → (SP)

Options

inst	operands	bytes	cycles	opcode
RTS	none	1	9	F9

Status Bits Affected None

Description RTS is typically the last instruction executed in a subroutine. RTS branches to the location immediately following the subroutine call instruction. In the called subroutine, there must be an equal number of POPs and PUSHes so that the stack pointer is pointing to the return address and not some other data.

Example
```
LABEL RTS      ; Return to main program from subroutine
```

Syntax **SBB** *s,Rd*

Execution (Rd) – (s) – 1 + (C) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
SBB	B,A	1	8	6B	(A) – (B) – 1 + (C) → (A)
SBB	Rs,A	2	7	1B	(A) – (Rs) – 1 + (C) → (A)
SBB	Rs,B	2	7	3B	(B) – (Rs) – 1 + (C) → (B)
SBB	Rs,Rd	3	9	4B	(Rd) – (Rs) – 1 + (C) → (Rd)
SBB	#iop8,A	2	6	2B	(A) – iop8 – 1 + (C) → (A)
SBB	#iop8,B	2	6	5B	(B) – iop8 – 1 + (C) → (B)
SBB	#iop8,Rd	3	8	7B	(Rd) – iop8 – 1 + (C) → (Rd)

Status Bits Affected

C Set to 1 if no borrow; 0 otherwise
N Set on result
Z Set on result
V ((C XOR N) AND (Source[Bit 7] XOR Destination[Bit 7]))

Description SBB performs multibyte 2s-complement subtraction. An SBB instruction with an immediate operand of zero value is equivalent to a conditional decrement of the destination operand, depending on the carry value. If (s) = 0 and (C) = 0, then (Rd) is decremented. A borrow operation occurs if the result is negative. In this case, the carry bit is set to 0. The carry bit acts as the no-borrow bit.

Examples

```
LABEL    SBB   #023h,B   ; Subtract 23h from (B), sub-
                         ;   tract 1, add the carry bit
                         ;   and store in register B

         SUB   R3,R21    ; R20:R21 and R2:R3 contain 1
         SBB   R2,R20    ;   bit numbers. SUB subtracts
                         ;   the LSbyte, and the SBB will
                         ;   use the carry as a borrow
                         ;   during the subtract of
                         ;   the MSbyte.
```

Syntax **SBIT0** *name*

Execution 0 → <*name*>

Options

inst	operands	bytes	cycles	opcode	operation	
SBIT0	Rname	3	8	73	0 → <bit>	Register bits
SBIT0	Pname	3	10	A3	0 → <bit>	Peripheral bits

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description SBIT0 is an assembler-constructed instruction that conveniently clears the value of the named bit without having to specify a register or mask. This enhances the readability of the program. This instruction assembles to the instructions AND #iop8,Rd or AND #iop8,Pd. The name for the bit is defined by the .DBIT assembler directive.

Example

```
INT1ENA   .DBIT 7,P01C      ; The interrupt 1 enable
                            ;   bit is now named INT1ENA

TEST      .DBIT 4,R33       ; Bit 4 of register 33
                            ;   is now named TEST

LABEL     SBIT0 TEST        ; Clears the value of the
                            ;   TEST bit

          SBIT0 INT1ENA     ; Disables interrupt 1
```

SBIT1 *Set Bit to 1*

Syntax **SBIT1** *name*

Execution 1 → <name>

Option

inst	operands	bytes	cycles	opcode	operation
SBIT1	Rname	3	8	74	1 → <bit> Register bits
SBIT1	Pname	3	10	A4	1 → <bit> Peripheral bits

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description SBIT1 is an assembler-constructed instruction that conveniently sets the value of the named bit without having to specify a register or mask. This enhances the readability of the program. This instruction assembles to the instructions OR #iop8,Rd or OR #iop8,Pd. The name for the bit is defined by the .DBIT assembler directive.

Examples

```
INT1ENA   .DBIT 7,P01C      ; The interrupt 1 enable bit
                            ;   is now named INT1ENA

TEST      .DBIT 4,R33       ; Bit 4 of register 33 is now
                            ;   named TEST

LABEL     SBIT1 TEST        ; Sets the value of the TEST
                            ;   bit to 1

          SBIT1 INT1ENA     ; Enables interrupt 1
```

Syntax **SETC**

Execution 1 → (C)

Options

inst	operands	bytes	cycles	opcode
SETC	none	1	7	F8

Status Bits Affected

C ← 1
N ← 0
Z ← 1
V ← 0

Description SETC sets the carry flag. This instruction can be used before an arithmetic or rotate instruction. The IE1 and IE2 enable bits are not affected.

Example

```
LABEL     SETC    ; Set the carry bit in the status register
                  ; Status register = 0Axh
                  ; Where X (lower nibble) is don't care
```

Syntax **STSP**

Execution (SP) → (B)

Options

inst	operands	bytes	cycles	opcode
STSP	none	1	8	FE

Status Bits Affected None

Description STSP copies the contents of the stack pointer to register B. This instruction can test the stack size. The indexed addressing mode can reference operands on the stack after executing this instruction.

Example

```
LABEL     STSP       ; Copy the contents of stack pointer
                     ;   to register B
```

Syntax **SUB** *s,Rd*

Execution (Rd) – (s) → (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
SUB	B,A	1	8	6A	(A) – (B) → (A)
SUB	Rs,A	2	7	1A	(A) – (Rs) → (A)
SUB	Rs,B	2	7	3A	(B) – (Rs) → (B)
SUB	Rs,Rd	3	9	4A	(Rd) – (Rs) → (Rd)
SUB	#iop8,A	2	6	2A	(A) – iop8 → (A)
SUB	#iop8,B	2	6	5A	(B) – iop8 → (B)
SUB	#iop8,Rd	3	8	7A	(Rd) – iop8 → (Rd)

Status Bits Affected

C Set to 1 if no borrow, otherwise set to 0
N Set on result
Z Set on result
V ((C XOR N) AND (Source[Bit 7] XOR Destination[Bit 7]))

Description SUB performs 2s-complement subtraction. The carry bit is set to 0 if a borrow is required. The carry bit acts as a no-borrow bit in this case.

Examples

```
LABEL     SUB  R19,B        ; (B) minus (R19) is
                            ;   stored in B

          SUB  #076h,A      ; (A) minus 076h is stored
                            ;   in A

          SUB  R4,R9        ; (R9) minus (R4) is stored
                            ;   in R9
```

Syntax **SWAP** *Rd*

Execution Bits (7,6,5,4 / 3,2,1,0) → Bits (3,2,1,0 / 7,6,5,4)

Options

inst	operands	bytes	cycles	opcode
SWAP	A	1	11	B7
SWAP	B	1	11	C7
SWAP	Rd	2	9	D7

Status Bits Affected

C Set to bit 4 of original register or bit 0 of result register
N Set on results
Z Set on results
V ← 0

Description SWAP exchanges the first four bits with the second four bits. This instruction is equivalent to four consecutive RL (rotate left) instructions. It is especially useful for packed BCD operations.

Examples

```
LABEL     SWAP R45    ; Switch low and high nibbles of R45

          SWAP  A     ; Switch low and high nibbles of A

          SWAP  B     ; Switch low and high nibbles of B
```

Syntax **TRAP** *#n* where n = the trap number = 0 thru 15

Execution

(SP) + 1 → (SP)
(PC MSbyte) → ((SP))
(SP) + 1 → (SP)
(PC LSbyte) → ((SP))
(Entry vector) → (PC)

Options

					entry-vector	
inst	**operands**	**bytes**	**cycles**	**opcode**	**MSbyte**	**LSbyte**
TRAP	#0	1	14	EF	7FDEh	7FDFh
TRAP	#1	1	14	EE	7FDCh	7FDDh
TRAP	#2	1	14	ED	7FDAh	7FDBh
TRAP	#3	1	14	EC	7FD8h	7FD9h
TRAP	#4	1	14	EB	7FD6h	7FD7h
TRAP	#5	1	14	EA	7FD4h	7FD5h
TRAP	#6	1	14	E9	7FD2h	7FD3h
TRAP	#7	1	14	E8	7FD0h	7FD1h
TRAP	#8	1	14	E7	7FCEh	7FCFh
TRAP	#9	1	14	E6	7FCCh	7FCDh
TRAP	#10	1	14	E5	7FCAh	7FCBh
TRAP	#11	1	14	E4	7FC8h	7FC9h
TRAP	#12	1	14	E3	7FC6h	7FC7h
TRAP	#13	1	14	E2	7FC4h	7FC5h
TRAP	#14	1	14	E1	7FC2h	7FC3h
TRAP	#15	1	14	E0	7FC0h	7FC1h

Status Bits Affected None

Description Trap is a 1-byte subroutine call. The operand <#n> is a trap number that identifies a location in the trap vector table (addresses 07FC0h to 07FDFh in memory). The contents of the 2-byte vector location form a 16-bit trap vector to which a subroutine call is performed. When you invoke the same routine more than once, TRAP is a more efficient instruction than CALL because fewer bytes are needed. The subroutine addresses are stored like all other addresses in memory, with the least significant byte in the higher-addressed location, as indicated above.

Examples

```
LABEL     TRAP #0          ; Execute subroutine at TRAPONE

.sect trap,07FC0h          ; Define section starting
                           ;   at 7FC0h

.word TRAP15,TRAP14        ; Define TRAPS 15 AND 14
                           ;   subroutine entry points
```

Syntax **TST** {**A** | **B**}

Execution C,N,Z,V bits affected

Options

inst	operands	bytes	cycles	opcode
TST	A	1	9	B0
TST	B	1	10	C6

Status Bits Affected

C ← 0
N Set or cleared based on operand
Z Set or cleared based on operand
V ← 0

Description TST sets the status bits according to the value in register A or B. This allows conditional jumps on the value in the register.

Examples

```
LABEL     TST A       ; Check for zero and negative
                      ;   conditions in register A

          TST  B      ; Check for zero and negative
                      ;   conditions in register B
```

Syntax **XCHB** *Rd*

Execution (B) ←→ (Rd)

Options

inst	operands	bytes	cycles	opcode	operation
XCHB	A	1	10	B6	(A) ← → (B)
XCHB	B	1	10	C6	(B) ← → (B) (TST B)
XCHB	Rd	2	8	D6	(Rd) ← → (B)

Status Bits Affected

C ← 0
N Set on original contents of B
Z Set on original contents of B
V ← 0

Description XCHB exchanges a register with register B without going through an intermediate location. The XCHB instruction with register B as the operand is equivalent to the TST B instruction.

Examples

```
LABEL     XCHB  A     ; Exchange register B with
                      ;   register A

          XCHB  R3    ; Exchange register B with R3
```

Syntax **XOR** *s,d*

Execution (s) XOR (d) → (d)

Options

inst	operands	bytes	cycles	opcode	operation
XOR	A,Pd	2	9	85	(A) XOR (Pd) → (Pd)
XOR	B,A	1	8	65	(B) XOR (A) → (A)
XOR	B,Pd	2	9	95	(B) XOR (Pd) → (Pd)
XOR	Rs,A	2	7	15	(Rs) XOR (A) → (A)
XOR	Rs,B	2	7	35	(Rs) XOR (B) → (B)
XOR	Rs,Rd	3	9	45	(Rs) XOR (Rd) → (Rd)
XOR	#iop8,A	2	6	25	iop8 XOR (A) → (A)
XOR	#iop8,B	2	6	55	iop8 XOR (B) → (B)
XOR	#iop8,Rd	3	8	75	iop8 XOR (Rd) → (Rd)
XOR	#iop8,Pd	3	10	A5	iop8 XOR (Pd) → (Pd)

Status Bits Affected

C ← 0
N Set on result
Z Set on result
V ← 0

Description XOR performs a bit-wise exclusive OR operation on the operands. The XOR instruction can complement bits in the destination operand. This operation can also toggle a bit in a register. If the bit value in the destination must be the opposite from what it currently is, then the source should contain a 1 in that bit location.

Examples

```
LABEL     XOR  R98,R125     ; XOR (R98) with (R125),
                           ;   store in R125

          XOR  #01,R20      ; Toggle bit 0 in R20

          XOR  B,A          ; XOR (B) with (A), store
                           ;   in register A
```

Chapter 17

Development Support

This chapter discusses the key features of the TMS370 development tools. These tools are currently available for PC-DOS or MS-DOS (version 3.0 and up) systems. Some tools are also available for UNIX systems. For a detailed description of system components, refer to the documents listed in the preface.

The topics in this chapter include:

Topic **Page**

17.1 TMS370 Development Tools

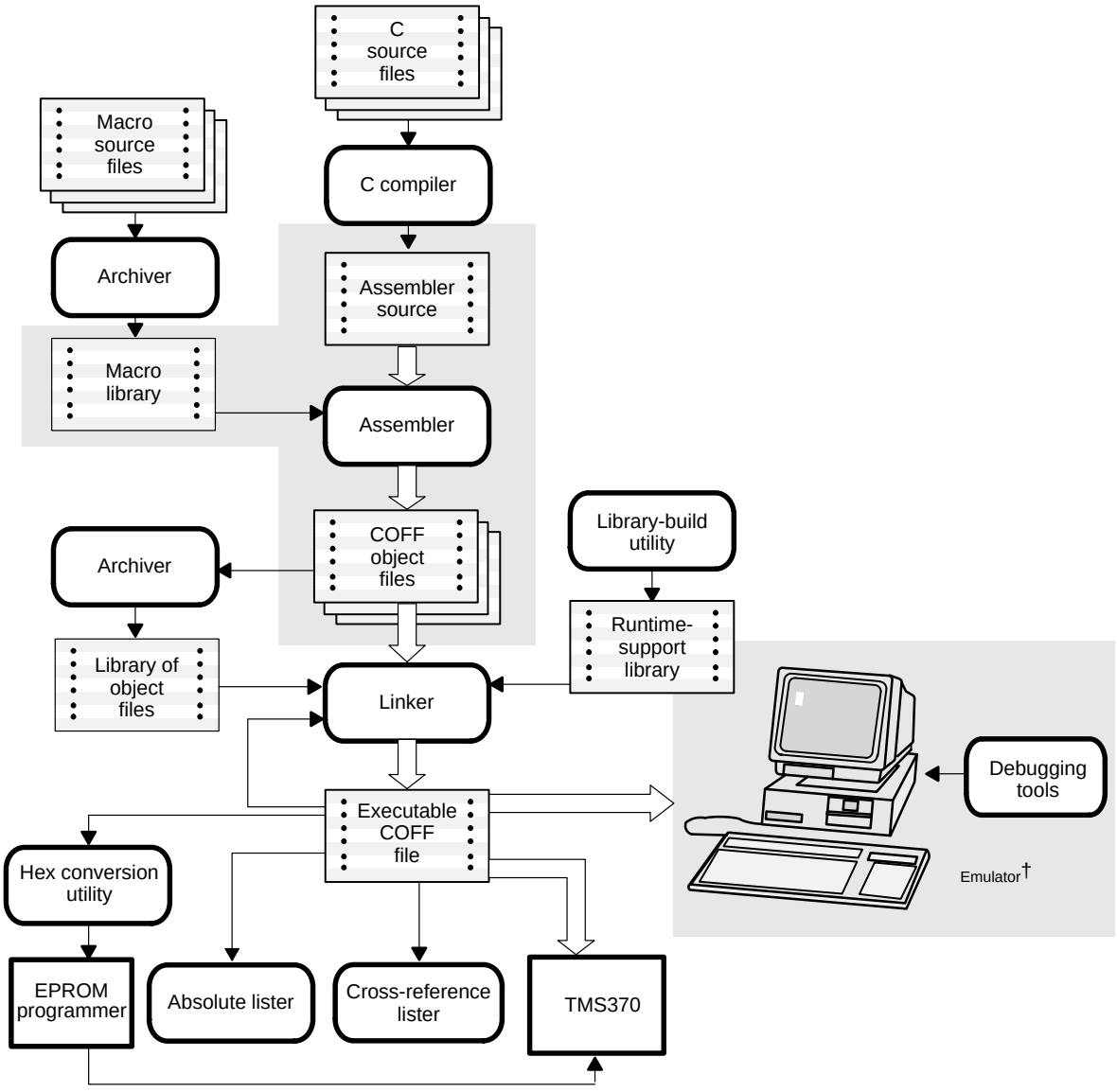
Texas Instruments provides extensive development support for the TMS370 family. This support includes:

- Assembly language tools, such as the assembler and linker, that convert assembly language into executable object code
- Additional software-support tools, such as an archiver and hex conversion utility
- An optimizing C compiler that supports high-level language programming and is a full implementation of the standard ANSI C language
- A C source debugger—an advanced software interface to the TMS370 XDS/22 emulator, CDT370, and design kit
- Development support tools that offer full-speed emulation for testing your object files:
  - XDS/22 (extended development support system) that provides real-time breakpoint, trace, and timing functions to facilitate hardware and software integration during system development
  - CDT370 (compact development tool) that supports realtime in-circuit emulation of the TMS370 family
  - Design kit for evaluation of the TMS370 family
  - Starter kit for programming and software simulation of the TMS370 family
- A software simulator
- TMS370 microcontroller programmer to program the programmable memory of any TMS370 device
- EPROM devices for prototype and small production runs

These development tools are designed to work with IBM-compatible PCs. Additionally, the TMS370C6xx and TMS370C7xx devices prototype and emulate masked-ROM parts and also act as a medium for submitting the program to TI for mask-ROM production.

Figure 17–1 shows the software development flow. The shaded portion highlights the most common development path; the other portions are optional.

Figure 17–1. Software Development Flow



† The emulator can be the XDS/22 or the CDT370. The design kit contains a simulator and EPROM/EEPROM programmer.

17.2 The Assembler

The TMS370 assembler translates assembly language source files into machine language object files. Source files can contain instructions, assembler directives, and macro directives. The assembler directives control various aspects of the assembly process, such as the source listing format, symbol definition, conditional assembly blocks, macro library definition, and the way the machine code is placed into the TMS370 memory space.

The format of the object files created by the assembler and linker is called *common object file format* (COFF). COFF encourages and facilitates modular programming. It allows the assembler to maintain a section program counter (SPC) for each section of object code generated. The SPC defines the virtual program-memory addresses assigned to the associated object code. The assembler uses the SPC while it builds the symbol table.

The symbol tables contained in the COFF object files allow the C source debugger to provide you with *symbolic debugging*. The debugger also provides for direct referencing of any assembler label and arithmetic expressions involving assembler labels when the labels are part of the downloaded COFF object file. The TMS370 microcontroller programmer also uses COFF object files to form a PC memory image of the data loaded for programming.

17.3 The Linker

The TMS370 linker creates executable modules by combining COFF object files. The concept of user-definable COFF *sections* is basic to the linker operation. The linker accepts several types of files as input:

- Relocatable COFF object files produced by the TMS370 assembler
- Command files
- Archive object libraries
- Output modules created by a previous linker run (referred to as partially linked files)

As the linker combines object files, it performs the following tasks:

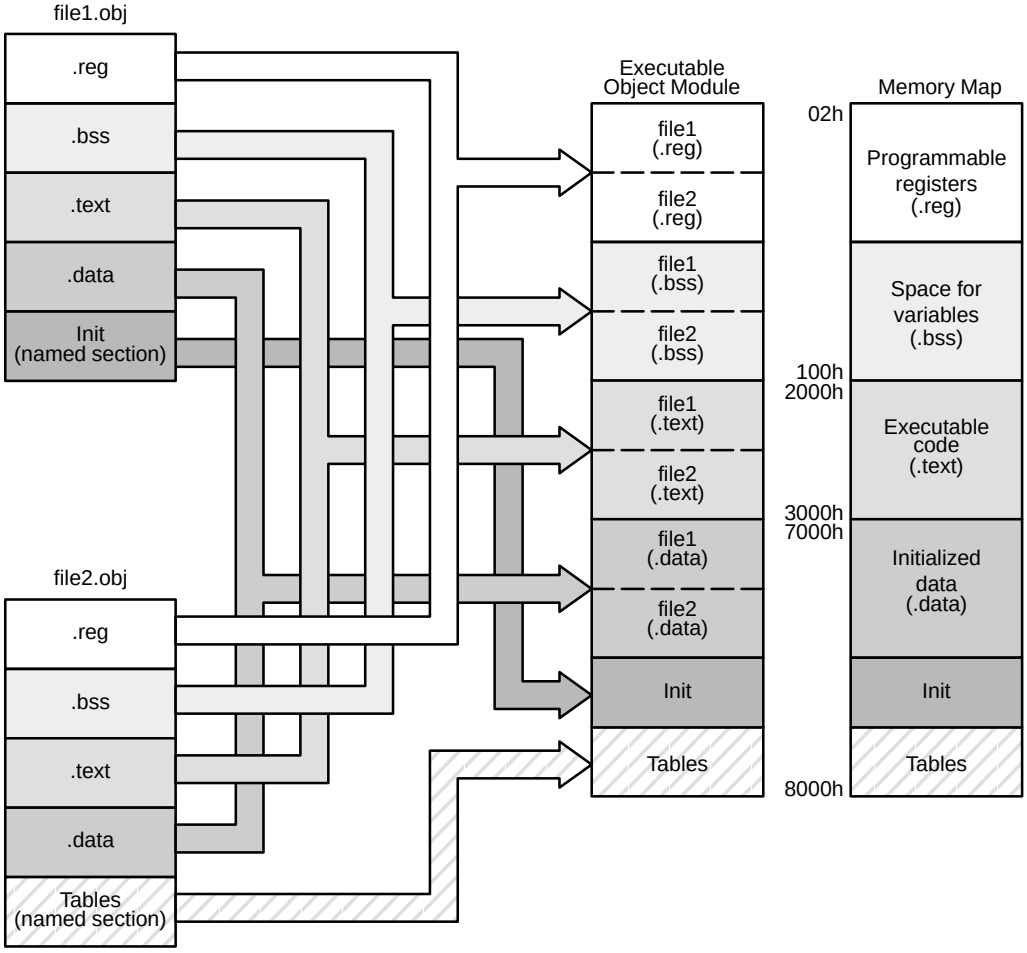
- Allocates sections into the target system's configured memory
- Relocates symbols and sections to assign them to final addresses
- Resolves undefined external references between input files

The linker supports a command language similar to C that controls memory configuration, section definition, and address binding. The language supports expression assignment and evaluation. It also provides two powerful directives, MEMORY and SECTIONS, that allow you to:

- Define a memory model that conforms to target system memory
- Combine object file sections
- Allocate sections into specific areas of memory
- Define overlayed memory structures
- Define or redefine global symbols at link time

Figure 17–2 shows the operation of the linker on two source code files. Each file has been assembled and contains four default sections and one named section. The various sections are arranged in the order dictated either by the linker's default method or by a user-supplied control file. The executable object module shows the combined sections, and the memory map indicates the location of the sections in memory.

Figure 17–2. Linker Output Generation



17.4 Additional Software Support

Two additional software utilities are included with the TMS370 assembly language tools:

- An archiver
- A hex conversion utility

17.4.1 The Archiver

The archiver provides file management by allowing a group of files to be collected into a single library. For example, macros can be collected by the archiver, then fetched by the assembler as directed by the source file. In addition, object modules can be collected into a library for convenient access by the linker. While not necessary for program development, the archiver can provide valuable organization in the building of the executable COFF object file.

17.4.2 The Hex Conversion Utility

The hex conversion utility converts a COFF object file into TI-tagged, ASCII-hex, Intel, Motorola-S, or Tektronix object format. The converted file can be downloaded to an EPROM programmer.

17.5 The Optimizing C Compiler

The TMS370 optimizing C compiler translates the widely used ANSI C language directly into highly optimized assembly language, enhancing productivity by enabling you to program in C. C code is easier to prototype, debug, and benchmark than assembly language. Also, it produces information that is used by the debugger, which allows source-level debugging in both C and assembly. This shortens the development cycle for TMS370 applications.

Key features of the C compiler include:

- Conformance with the ANSI C specification
- Highly efficient code—the C compiler incorporates state-of-the-art generic and target-specific optimizations
- ANSI standard runtime-support library
- A C shell program that facilitates one-step translation from C source to executable code
- ROM-able, relocatable, and re-entrant code (where declared)
- A source interlist utility that can interlist your original C source statements into assembly language output of the compiler
- A utility for building object libraries from source libraries
- Fast compilation to increase productivity

The following is a partial list of key optimizations used by the compiler:

- Performs control-flow graph simplification
- Allocates variables to registers
- Performs loop rotation
- Performs loop unrolling
- Performs loop optimizations expressions
- Eliminates dead code
- Simplifies expressions and statements
- Performs local copy/constant propagations
- Removes dead assignments
- Eliminates local common expressions
- Eliminates global common expressions
- Eliminates global dead assignments

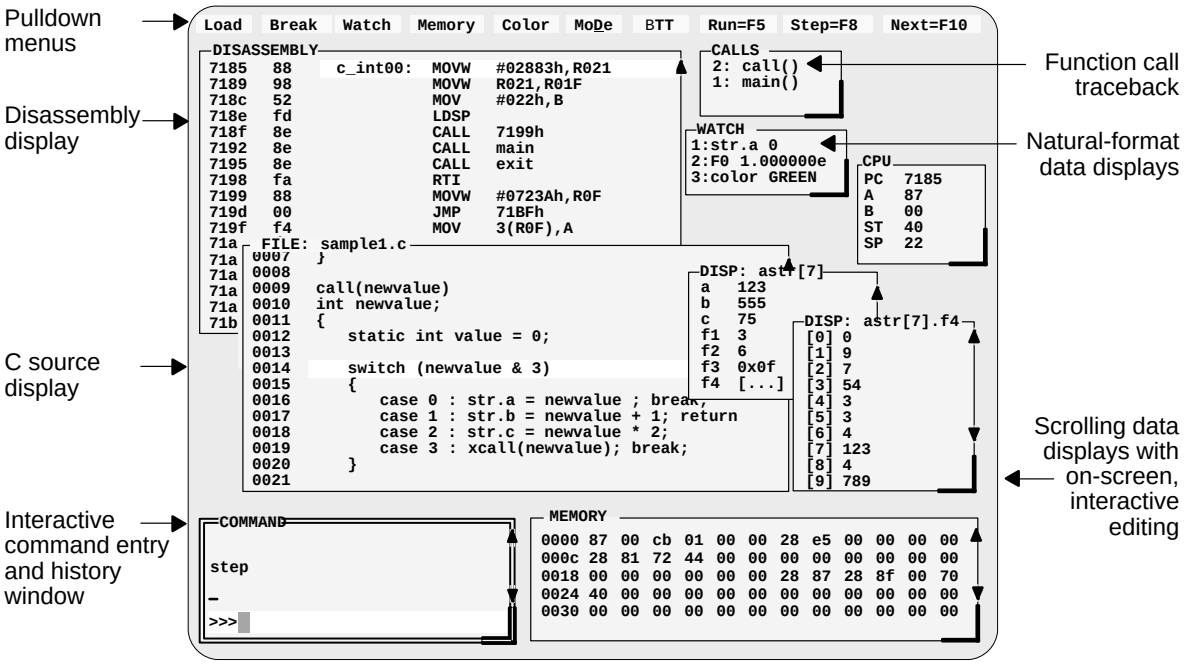
17.6 The C Source Debugger

The C source debugger is an advanced software interface that runs on IBM-compatible PCs and helps you to develop, test, and refine TMS370 C programs (compiled with the TMS370 optimizing ANSI C compiler) and assembly language programs. The debugger is the interface to the TMS370 in-circuit XDS/22 emulator, design kit, CDT370, and software simulator. The debugger has a standard screen appearance for the supported products, allowing you to move from one tool to the next without having to learn a new interface—only the new features.

The C source debugger improves productivity by enabling you to debug a program in the language it was written in. You can choose to debug your programs in C, assembly language, or both. Also, the debugger is easy to learn and use. Its friendly window-, mouse-, and menu-oriented interface reduces learning time and eliminates the need to memorize complex commands.

Figure 17–3 identifies several features of the debugger display.

Figure 17–3. The Basic Debugger Display



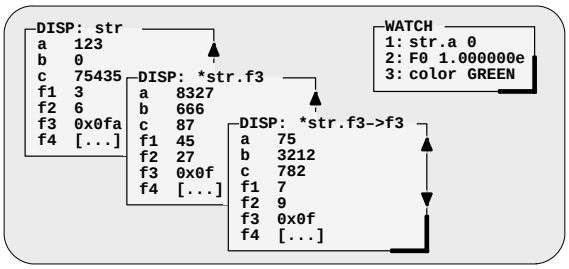
Key features of the C source debugger include:

- **Multilevel debugging**. The debugger allows you to debug both C and assembly language code. If you're debugging a C program, you can choose to view just the C source, the disassembly of the object code created from the C source, or both. You can also use the debugger as an assembly language debugger.

- **Fully configurable, state-of-the-art, window-oriented interface.** The C source debugger separates code, data, and commands into manageable portions. Use any of the default displays, or select the windows you want to display, size them, and move them where you want them. The debugger display is completely configurable, allowing you to create the interface that is suited for your use.

- **Comprehensive data displays.** You can easily create windows for displaying *and editing* the values of variables, arrays, structures, pointers—any kind of data—in their natural format (*float*, *int*, *char*, *enum*, or *pointer*). You can even display entire linked lists (see Figure 17–4).

Figure 17–4. Debugger Display



- **On-screen editing.** Change any data value displayed in any window—just point the mouse, click, and type.

- **Continuous update.** The debugger continuously updates information on the screen, highlighting changed values.

- **Powerful command set.** The C source debugger supports a small, but powerful, command set that makes full use of C expressions. One debugger command performs actions that would require several commands in another system.

- **Flexible command entry.** There are a variety of ways to enter commands. You can type commands or use a mouse, function keys, or the pulldown menus; choose the method that you like best.

- **Variety of screen sizes.** The debugger's default configuration is set up for a typical PC display, with 25 lines by 80 characters. If you use a sophisticated graphics card, you can take advantage of the debugger's additional screen sizes. A larger screen size allows you to display more information and provides you with more screen space for organizing the display—bringing the benefits of workstation displays to your PC.

- **All the standard features you expect in a world-class debugger.** The debugger provides you with complete control over program execution with features like conditional execution and single-stepping (including single-stepping into or over function calls). You can set or clear a breakpoint with a click of the mouse or by typing commands. You can define a memory map that identifies the portions of target memory that the debugger can access. You can choose to load only the symbol table portion of an object file to work with systems that have code in ROM. The debugger can execute commands from a batch file, providing you with an easy method for entering often-used command sequences.

17.7 The XDS/22 System

The XDS/22 system is a self-contained package that provides full-speed, in-circuit emulation and debugging functions required for program development of the TMS370 family devices. These are key features of the XDS/22 emulation function:

- A 0.5- to 5-MHz SYSCLK full-speed in-circuit emulation of all TMS370 family members
- Realtime hardware breakpoint/trace/timing analysis capabilities
- Execution of programs from internal XDS/22 memory (64K byte) or target memory
- Support of both microcomputer and microprocessor modes
- Large trace buffer—2047 samples
- Full logic tracing with logic analyzer interface cable

The XDS/22 system set of boards consists of an emulator, communications board, and a breakpoint/trace/timing board. At the heart of the XDS/22 system is a special system emulator chip containing all of the peripheral modules and I/O line circuits that precisely duplicate the TMS370's logic and performance. You can use the internal XDS/22 memory to emulate on-chip ROM and/or external memory.

The target cables that are necessary for using the XDS/22 are sold separately. See subsection 19.4.5, page 19-26, for ordering information.

The XDS/22 is supported by a debugger, as described in Section 17.6, on page 17-9.

17.7.1 Breakpoint, Trace, and Timing (BTT) Functions

Included with the XDS/22 emulation system is a separate board called the BTT board, which provides breakpoint, trace, and timing features. The BTT monitors the TMS370 CPU; when a preselected pattern of bus activity is detected, the BTT performs an action such as executing a hardware breakpoint or storing information in the trace buffer.

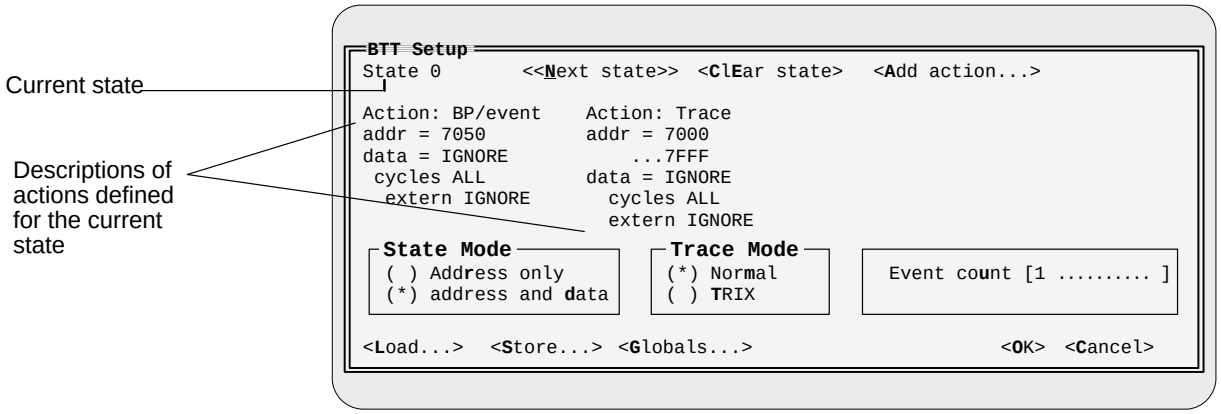
The BTT supports these features:

- **Full range of actions.** The BTT functions allow you to set hardware breakpoints, count event occurrences, collect trace samples, jump to a BTT state, or start or stop timers. These actions occur when they are *qualified*—that is, when bus activity matches conditions that you have defined.

- **Four separate states.** The BTT supports four separate states, called state 0 through state 3. Each state can be associated with up to four actions. You can define actions for as many states as you need. By default, the BTT cycles through the states, beginning with state 0 and ending with the last state that you defined actions for. You can control this sequencing by jumping to another state or by using counters to loop through a sequence of states.

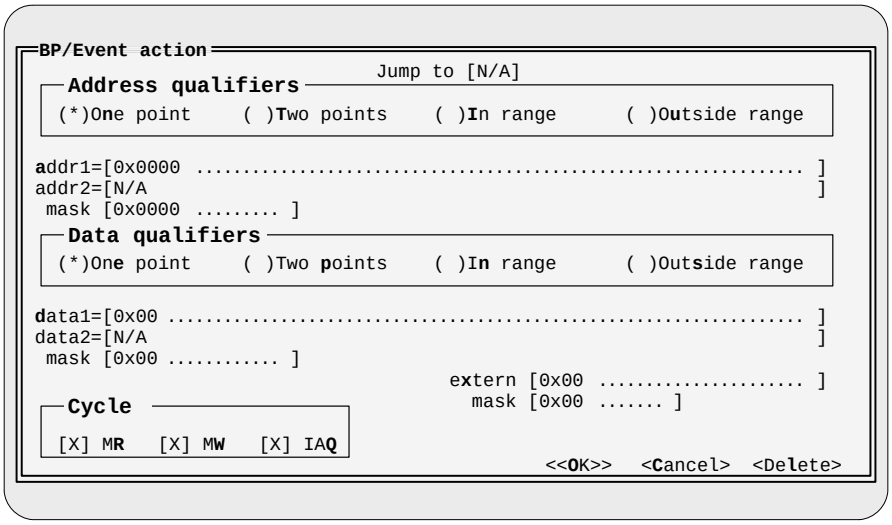
  To view information about specific states, add an action, delete an action, or access global settings, use the BTT Setup dialog box (see Figure 17–5).

Figure 17–5. BTT Setup Dialog Box



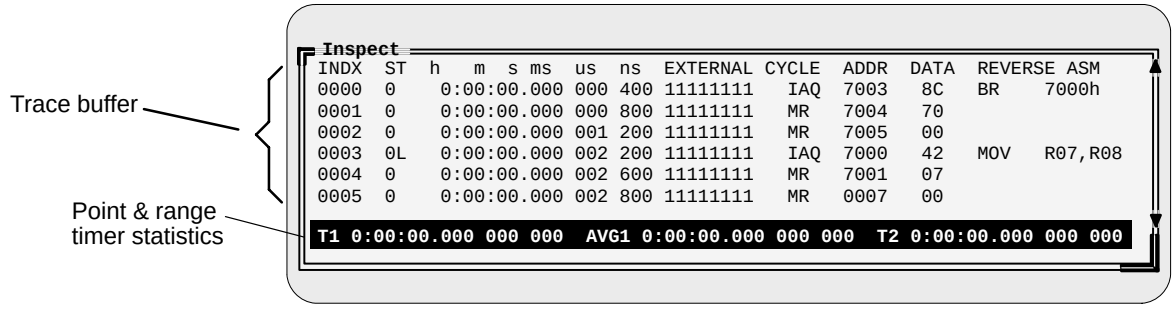
- **Flexible qualification for actions.** You can qualify actions according to address or data values (either singly or in relation to ranges) and to the memory-cycle type. You can combine these conditions; for example, you could qualify an action whenever a certain data value is accessed during an instruction acquisition cycle. You can define conditions or qualify actions by using a dialog box (refer to Figure 17–6).

Figure 17–6. Dialog Box for Defining Conditions



- **Informative trace reporting.** The BTT can store up to 2047 trace samples in the trace buffer. You can display the trace samples and associated information by opening the Inspect window (see Figure 17–7).

Figure 17–7. Example of the Inspect Window



- **External signal access.** The BTT has eight external probes that can be connected to eight signals. You can qualify actions by looking for a particular pattern of activity on these probes. Additionally, the trace buffer reports the values that were on the signals when each trace sample was collected.

- **Complete timing analysis.** The BTT supports two timers that you can start and stop on a variety of conditions. The BTT reports the total time for each of these timers; it also reports the average for one of the timers. Additionally, the BTT collects timing information that relates specifically to the samples in the trace buffer.

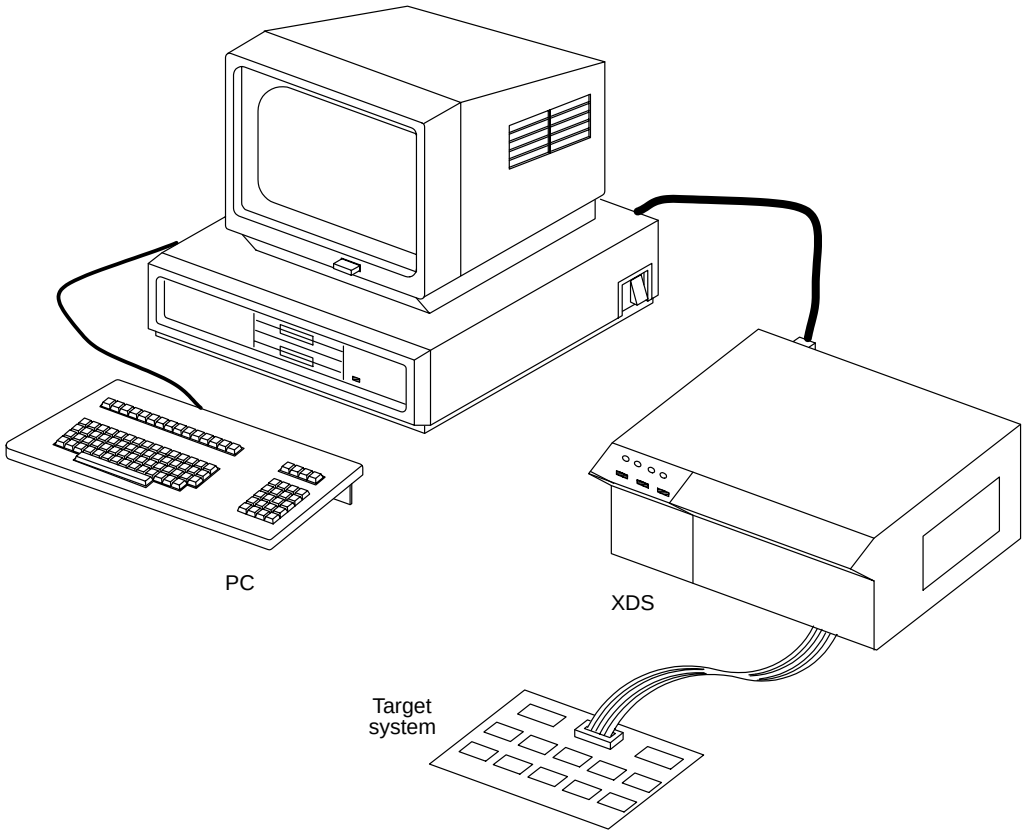
- **External filing.** Once you have defined a complex BTT setup, you may want to reuse the setup. The BTT allows you to save the setup to a file and then load it again for a later session. You can also save the contents of the trace buffer to a file for later use or to compare to another trace collection.

17.7.2 XDS System Configuration Requirements

A functional XDS system configuration consists of the XDS system and the following user-supplied components:

- ❑ IBM-compatible PC, a serial communication port, and a 3-1/2-inch high-density disk drive
- ❑ Windows version 3.1 or later
- ❑ Monitor (preferably color, to better highlight field and value changes)

Figure 17–8. Typical XDS System Configuration



17.7.3 XDS System Operating Considerations

The emulation hardware of the XDS systems (XDS/22) generally exhibits the same characteristics as the actual TMS370 devices. There are, however, a few subtle differences that you should be aware of when building a prototype circuit for use with the XDS system.

- **Mode control pin.** To allow an XDS system to function without being attached to a target system, a 20-kΩ pulldown resistor is connected to the mode control line in the XDS unit. This increases the minimum input current needed to drive this line high (I_I) from 100 μA to 300 μA. If you are using a pull-up resistor to put the device in the microprocessor mode, then its value should be no greater than 1 kΩ.

- **Reset.** The XDS systems add an analog switch and a 51-kΩ pull-up resistor to the reset line. This increases the current necessary to pull this line to a logic low from 10 μA to 100 μA.

- **Clock in.** The XDS systems cannot drive a crystal located on the target system. Therefore, either the crystal must be moved to socket Y1 of the emulator board, or an external clock signal must be connected to the XTAL2/CLKIN pin. The external clock signal must meet the V_{IH} specifications for the XTAL2/CLKIN described in Chapter 18, *Electrical Specifications and Timings*.

17.7.4 XDS Target Connectors

For additional or replacement XDS target connectors, contact the TI factory repair department.

17.8 The CDT370 (Compact Development Tool)

The CDT370 supports real-time in-circuit emulation of most of the TMS370 family of devices. It offers a low-cost, highly efficient route to TMS370 family development of software and hardware with the target system.

The CDT370 contains a single board, called an emulator, which can be plugged into the expansion chassis of any IBM-compatible PC, or connected through the RS-232 serial communication link. Under Windows ’95, it must be connected to an RS-232 serial communication link. Attached to the emulator is a target cable with the same pinout as the system’s circuit board; the cable uses the same socket that would normally hold the TMS370 microcontroller.

The CDT370 uses a debugger interface that is similar to that of the XDS/22 systems.

With the CDT370, you can:

- ❑ Program the EEPROM and EPROM contained within the TMS370 devices
- ❑ Inspect and modify memory locations
- ❑ Upload/download program and data memory
- ❑ Execute programs and software routines
- ❑ Use a large trace buffer with 2,048 samples
- ❑ Single-step executable instructions
- ❑ Use software breakpoints to halt program execution at selected addresses

The CDT package is shown in Figure 17–9, on page 17-19, and comes complete with:

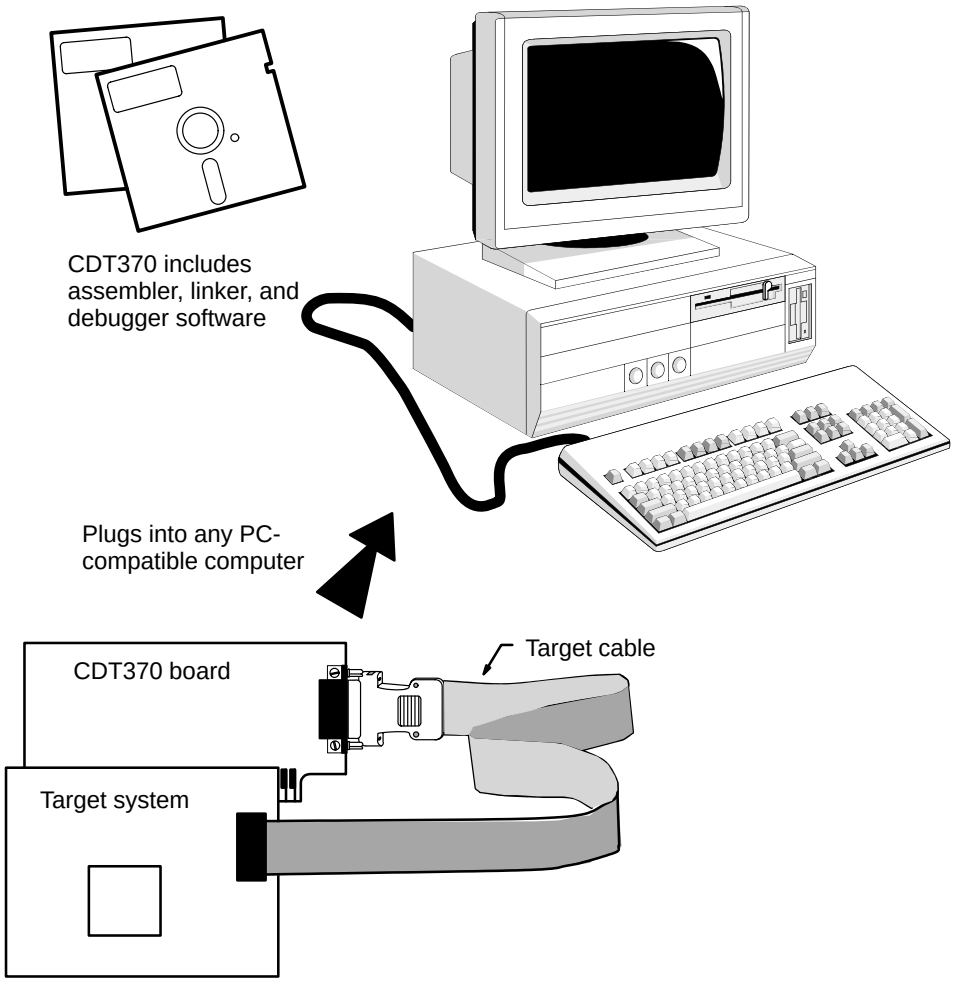
- ❑ A CDT370 emulator board
- ❑ An assembler and linker
- ❑ The C source debugger

If the CDT is connected externally with an RS-232 cable, the following requirements must be supplied by the user:

- ❑ A 5 V, 3 A power supply
- ❑ A 9-pin RS-232 cable

The cables that you need to use the CDT370 are sold separately. See subsection 19.4.6, page 19-26, for ordering information.

Figure 17–9. CDT370 Configuration



17.8.1 The CDT370 PACT (Compact Development Tool PACT)

The CDT370 PACT supports real-time in-circuit emulation for the ’x32, ’x36, and ’x9x family devices. The CDT370 PACT offers the same features and functionality as the CDT370.

17.8.2 The CDT370 Timer (Compact Development Tool Timer)

The CDT370 timer supports real-time in-circuit emulation for the ’x6x, ’x7x, and ’xBx family devices. The CDT370 timer offers the same features and functionality as the CDT370, but with the addition of a logic analyzer interface.

17.9 The Design Kit

The TMS370 design kit is a low-cost evaluation tool that lets you analyze the hardware and software capabilities of the TMS370 family by actually using the TMS370 devices. The kit consists of a TMS370 application board, assembler, linker, and debugger software. The design kit helps you quickly assess the feasibility of using a member of the TMS370 family for your applications. However, it can only be used to evaluate the TMS370Cx0x, TMS370Cx1x, and TMS370Cx5x (can evaluate neither TMS370Cx58 nor TMS370Cx59) devices. See subsection 19.4.2, on page 19-25, for ordering information.

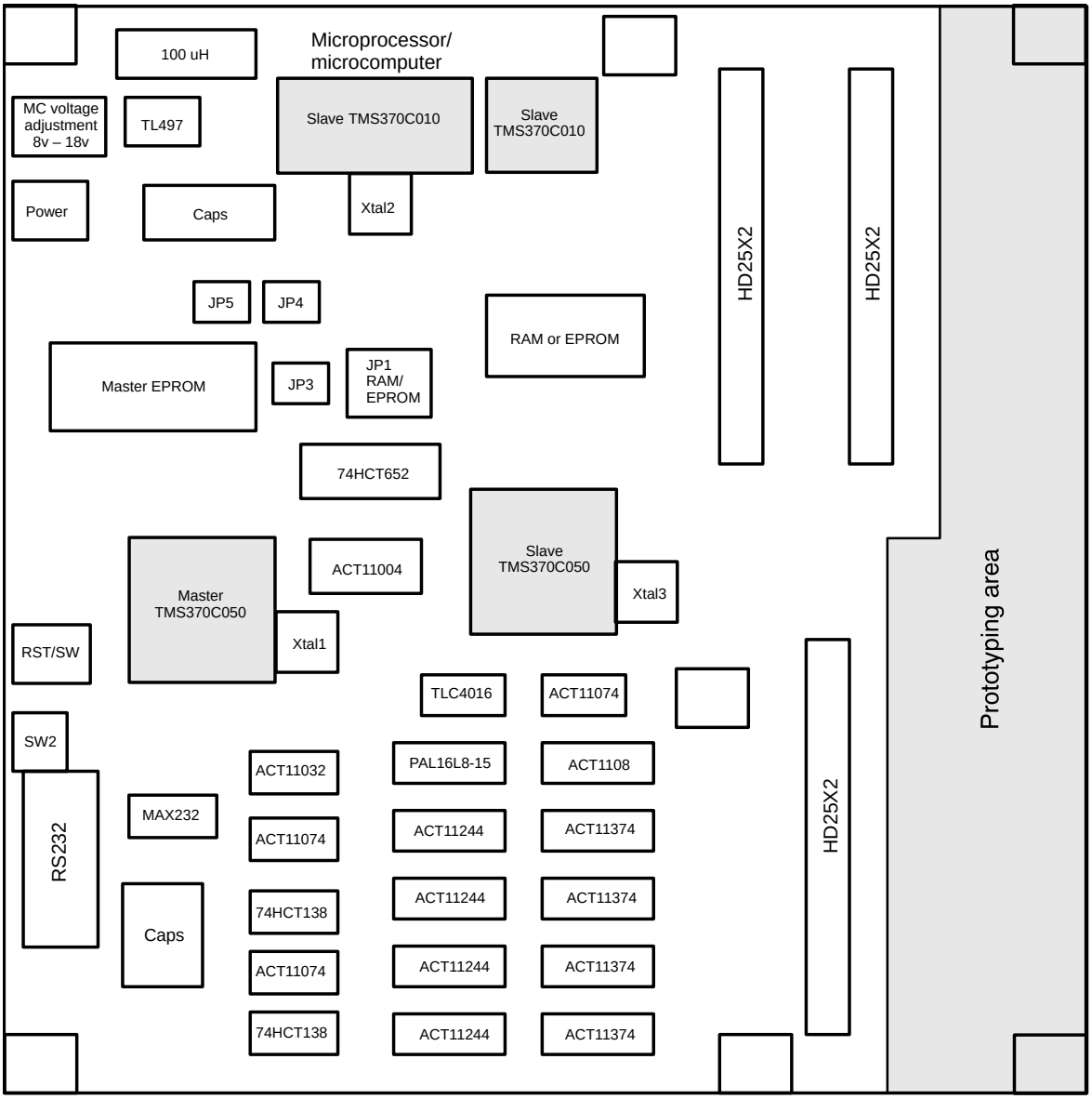
The design kit package allows you to:

- ❑ Upload and download code
- ❑ Access any register or memory location
- ❑ Read and modify memory locations
- ❑ Execute programs and software routines
- ❑ Single-step executable instructions
- ❑ Use software breakpoints to halt program execution at selected addresses
- ❑ Program the EEPROM and EPROM contained within the following device families: TMS370Cx0x, TMS370Cx1x, and TMS370Cx5x (except TMS370Cx58 and TMS370Cx59). For devices other than the TMS370Cx0x, TMS370Cx1x, and TMS370Cx5x devices, you must wire your own socket to the board.
- ❑ Use the assembler on a PC
- ❑ Use a wire-wrap prototyping area
- ❑ Use a patch assembler in the debugger mode
- ❑ Use a reverse assembler

The design kit includes:

- ❑ An assembler
- ❑ A C-source debugger
- ❑ An application board, also referred to as a design kit board (shown in Figure 17–10 on page 17-21)

Figure 17–10. Application Board (Design Kit Board)



Requirements that you supply:

- A 5 V, 500 mA power supply
- A 25-pin RS-232 cable (DB-25)

The design kit application board operates in one of three modes:

- Debugger mode
- TTY mode
- Isolated mode

Debugger Mode

The debugger mode is the standard TI programmer's interface. There are many advantages to using the debugger mode of operation:

- Multilevel debugging
- Powerful command set
- Window-oriented interface
- Comprehensive data displays
- On-screen editing
- Patch assembly
- Direct COFF download

To use the debugger mode of the design kit, the PC must meet the XDS system configuration requirements as discussed in subsection 17.7.2, on page 17-16. For more information on the debugger function, refer to Section 17.6.

TTY Mode

The TTY mode allows you to communicate using an ASCII serial protocol with equipment such as dumb terminals, PCs running terminal interface programs (for example, Crosstalk or Procomm), or computers that are not PC-compatible. This mode is most useful when you don't have access to a PC or if you need to do only simple operations or programming. The TTY mode has several two-letter commands that allow you to check the operation of an application board.

Additionally, you can perform the following memory operations in the TTY mode; these operations aren't available in the debugger mode:

- ❏ Run system tests
- ❏ Upload memory
- ❏ Verify memory
- ❏ Find bytes
- ❏ Move memory
- ❏ Compare memory
- ❏ Block program
- ❏ Directly read the analog pin
- ❏ Load on-chip EEPROM from on-board UVEPROM
- ❏ Use a computer in dumb terminal mode
- ❏ Reverse assemble with cycle times

Isolated Mode

After you have written and debugged a program for a slave, you can check the accuracy of that program by using the isolated mode. This mode turns off all debugger software and allows the slave device to run on its own.

17.10 The Starter Kit

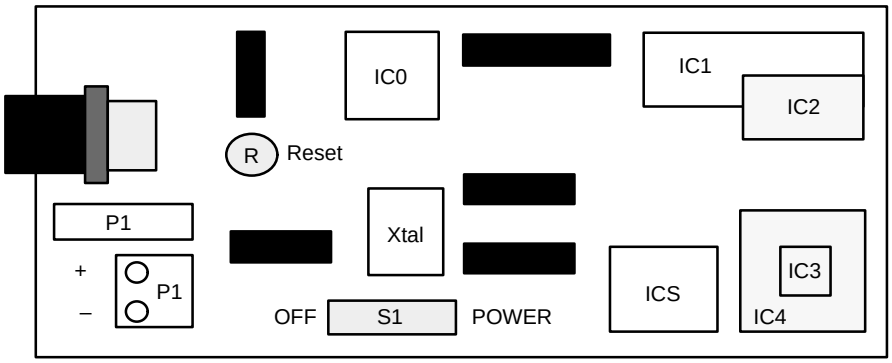
The TMS370 starter kit helps you to quickly assess the feasibility of using a member of the TMS370 family for your application. This low-cost programmer and simulation tool allows you to conduct software programming and simulations for the EEPROM and EPROM of the following families:

TMS370Cx0x	TMS370Cx1x	TMS370Cx2x	TMS370Cx32
TMS370Cx4x	TMS370Cx5x (except ’759)	TMS370Cx6x (except ’769)	TMS370Cx7x
TMS370CxAx	TMS370CxBx		

The starter kit package includes the following:

- ❑ TMS370 Assembler diskette and documentation
- ❑ TMS370 Simulator. Its capabilities include the following:
  - Upload and download code
  - Access any register or memory location
  - Read and modify memory locations
  - Execute programs and software routines
  - Single-step executable instructions
  - Use software breakpoints to halt program execution at selected addresses
- ❑ Programming adapter board and programming software (shown in Figure 17–11)

Figure 17–11. Starter Kit



Requirements supplied by user:

- ❑ A 5 V, 500 mA power supply
- ❑ A 9-pin RS-232 cable
- ❑ ZIF sockets

17.11 The Microcontroller Programmer

The TMS370 microcontroller programmer is an interactive, menu-driven system that provides a method of programming TMS370 family devices and EPROMs.

The TMS370 microcontroller programmer system consists of a PDS (programmable device support) base unit, a microcontroller programmer top, and an IBM-compatible PC running microcontroller programmer software under MS/PC-DOS.

The programmer top and PDS base unit are sold separately. Five types of tops are available for the packages listed below:

Type of Top	TMS370 Packages	Devices Supported
1	28-Pin PDIP 40-Pin PDIP 40-Pin PSDIP 64-Pin PSDIP	’x1x ’x2x, ’x4x, ’x8x, and ’xAx ’x2x and ’x4x ’x5x, ’x7x, and ’xBx
2	28-Pin PLCC 44-Pin PLCC 68-Pin PLCC	’x0x and ’x1x ’x2x, ’x32, ’x4x, and ’x8x ’x5x, ’x6x, ’x7x, and ’xBx
3	44-Pin PLCC	’x36
4	40-Pin PSDIP 44-Pin PLCC	’x9x ’x9x
5	28–Pin PLCC/PDIP	’xCx

NOTE: Refer to subsection 19.4.4, on page 19-26, for ordering information.

The programmer software provides both interactive and limited batch control with the following features:

- ❑ Window-oriented screens with a menu-driven command structure
- ❑ Intermediate PC memory, which provides a storage area for downloading a file or for uploading from a device, which allows you to inspect and patch load data
- ❑ Relocatable programming capability, which allows source data bytes within a certain address range to be programmed at a specified address

17.12 Reprogrammable EPROM and OTP Devices

In the TMS370 one-time programmable (OTP) and the SE370 reprogrammable EPROM devices, program ROM has been replaced by a programmable program memory, such as EPROM.

- The TMS370 OTP devices are in plastic packages and can be programmed one time. The OTP device is an effective microcontroller to use for immediate production updates for other members of the TMS370 family or for low-volume production runs that cannot satisfy minimum volume or cycle times for low-cost mask-ROM devices.
- The SE370 reprogrammable EPROM devices are in windowed ceramic packages to allow reprogramming of the EPROM memory during the prototype development phase of design. This capability provides form factor preproduction parts with zero lead time for field testing and production qualifications, thereby reducing the overall time to market. This also supports applications with small production runs. You can program all TMS370 devices directly from the assembler or linker output file with the TMS370 microcontroller programmer. The application board can program the corresponding devices for which it demonstrates the capabilities.

Table 17–1, on page 17-27, shows the TMS370 ROM devices and their corresponding OTP devices and reprogrammable EPROM devices.

Table 17–1. OTP and Reprogrammable EPROM Support of ROM Devices

TMS370 ROM Device	OTP EPROMs	Reprogrammable EPROMs
TMS370C002A TMS370C302A	TMS370C702	SE370C702
TMS370C010A TMS370C012A TMS370C310A TMS370C311A TMS370C312A	TMS370C712A TMS370C712B	SE370C712A SE370C712B
TMS370C020A TMS370C320A TMS370C022A TMS370C322A	TMS370C722	SE370C722
TMS370C032A TMS370C036A TMS370C332A	TMS370C732A TMS370C736A	SE370C732A SE370C736A
TMS370C040A TMS370C340A TMS370C042A TMS370C342A	TMS370C742A	SE370C742A
(See Note 1) TMS370C050A TMS370C350A TMS370C052A TMS370C352A TMS370C452A TMS370C353A TMS370C056A TMS370C356A TMS370C456A TMS370C058A TMS370C358A TMS370C059A	TMS370C756A TMS370C758A TMS370C758B TMS370C759A	SE370C756A SE370C758A SE370C758B SE370C759A
TMS370C067A TMS370C068A TMS370C069A	TMS370C768A TMS370C769A	SE370C768A SE370C769A
TMS370C077A	TMS370C777A	SE370C777A
TMS370C080 (See Note 2) TMS370C380A	TMS370C686A	SE370C686A
TMS370C090A	TMS370C792	SE370C792
TMS370C3A7A	(See Note 2)	(See Note 2)
TMS370C0B6A	(See Note 2)	(See Note 2)
TMS370C3C0A	TMS370C6C2A	SE370C6C2A

Notes: 1) ROM-less devices (TMS370C15x, TMS370C25x) do not have EPROM devices.
2) For OTP EPROMs, the TMS370C758A and TMS370C758B are recommended. For reprogrammable EPROMS, the SE370C758A and SE370C758B are recommended.
   – For the TMS370C080 and TMS370C3A7 ROM devices, a converter socket is required in conjunction with the ’C758 EPROMS.
   – For the TMS370C0B6 device, refer to the TMS370CxBx Data Sheet (SPNS038) for pin-to-pin compatibility with the ’C758 EPROMs.

Chapter 18

Electrical Specifications and Timings

This chapter contains electrical and timing information for the TMS370 family devices. This information is presented according to device category.

Topic **Page**

18.1 Timing Parameter Symbols

Timing parameter symbols occur throughout this chapter. They were created in accordance with JEDEC standard 100A. To keep the symbols short, some of the signal names and other operational terms were abbreviated as follows:

A	Address	RXD	SCIRXD
AR	Array mode	S	Slave mode
B	Byte mode	SC	SYSCLK
CI	XTAL2/CLKIN	SCC	SCICLK
D	Data	SIMO	SPISIMO
E	$\overline{\text{EDS}}$	SOMI	SPISOMI
FE	Final	SPC	SPICLK
IE	Initial	TXD	SCITXD
M	Master mode	W	Write
PGM	Programming	WT	$\overline{\text{WAIT}}$
R	Read		

Lowercase subscripts and their meanings are:

c	Cycle time (period)	r	Rise time
d	Delay time	su	Setup time
f	Fall time	v	Valid time
h	Hold time	w	Pulse duration (width)

The following additional letters are used with these meanings:

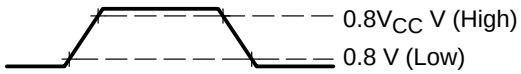
H	High	V	Valid
L	Low	Z	High impedance

18.2 Parameter Measurements

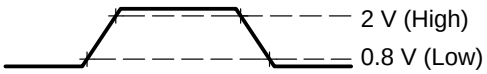
All timings are calculated between high and low measurement points as indicated in Figure 18–1.

Figure 18–1. Measurement Points for Timings

(a) XTAL2/CLKIN measurement points



(b) General measurement points



18.3 Absolute Maximum Ratings for All TMS370 Devices

For all TMS370 devices, Table 18–1 provides the absolute maximum ratings over the operating free-air temperature range. This operating free-air temperature range is specified for your device in its respective section of this chapter.

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions for the specific device is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–1. Absolute Maximum Ratings Over Operating Free-Air Temperature Range (See Note 1)

Parameter		Maximum Rating
Supply voltage range, V_{CC} (see Note 1)		–0.6 V to 7 V
Input voltage range	All pins except MC	–0.6 V to 7 V
	MC	–0.6 V to 14 V
Input clamp current, I_{IK} (V_I < 0 V or V_I > V_{CC})		±20 mA
Output clamp current, I_{OK} (V_O < 0 V or V_O > V_{CC})		±20 mA
Continuous output current per buffer, I_O (V_O = 0 V to V_{CC}) (see Note 2)		±10 mA
Maximum supply current, I_{CC}		170 mA
Maximum supply current, I_{SS}		–170 mA
Continuous power dissipation	TMS370Cx0xA, TMS370Cx0x, TMS370Cx1xA, TMS370Cx1xB, TMS370CxCxA	500 mW
	TMS370Cx32A	800 mW
	TMS370Cx2xA, TMS370Cx2x, TMS370Cx36A, TMS370Cx4xA, TMS370Cx5xA, TMS370Cx5xB, TMS370Cx6xA, TMS370Cx7xA, TMS370Cx8xA, TMS370Cx9xA, TMS370CxAxA, TMS370CxBxA	1 W
Storage temperature range, T_{stg}		–65°C to 150°C

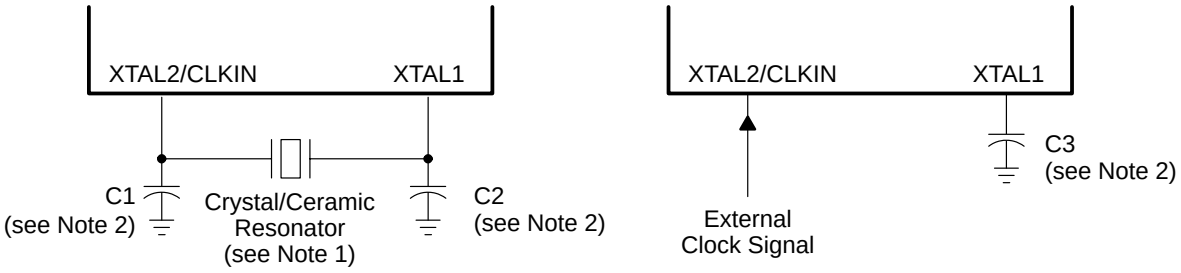
Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).

2) Electrical characteristics are specified with all output buffers loaded with the specified I_O. Exceeding the specified I_O in any buffer may affect the levels on other buffers.

18.4 External Crystal/Clock Connections and Typical Circuits for Loads and Buffers

Figure 18–2 illustrates how to connect the crystal/ceramic resonator and the external clock signal. This figure is valid for all TMS370 family devices.

Figure 18–2. Recommended Crystal/Clock Connections

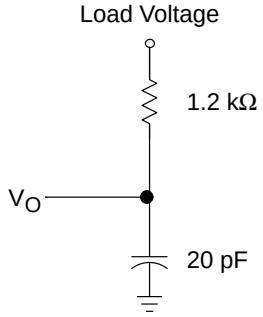


Notes: 1) The crystal/ceramic resonator frequency is four times the reciprocal of the system clock period.

2) The values of C1 and C2 are typically 15 pF and the value of C3 is typically 50 pF. See the manufacturer's recommendations for ceramic resonators.

Figure 18–3 is an output load circuit that you can use for any TMS370 device.

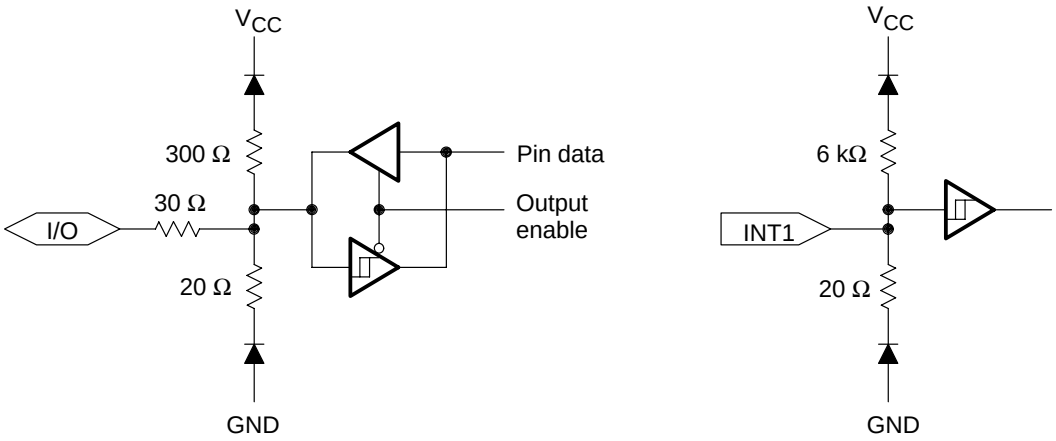
Figure 18–3. Typical Output Load Circuit



Case 1: $V_O = V_{OH} = 2.4$ V; Load Voltage = 0 V
Case 2: $V_O = V_{OL} = 0.4$ V; Load Voltage = 2.1 V

Note: All measurements are made with the pin loading as shown unless otherwise noted. All measurements are made with XTAL2/CLKIN driven by an external square wave signal with a 50% duty cycle and rise and fall times less than 10 ns unless otherwise stated.

Figure 18–4. Typical Buffer Circuitry



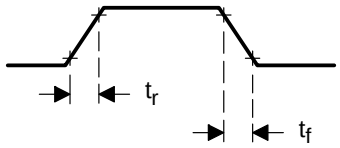
18.5 General-Purpose Output Signal Timings

Refer to Sections 18.1 and 18.2 for timing symbol definitions and parameter measurement points. The timings shown in this section are valid for all TMS370 family devices.

Table 18–2. General-Purpose Output Signal Timing Requirements

Parameter		Min	Nom	Max	Unit
t_r	Rise time		30		ns
t_f	Fall time		30		ns

Figure 18–5. Switching Time Measurement Points



18.6 EPROM/EEPROM Specifications

Refer to Sections 18.1 and 18.2 for timing symbol definitions and parameter measurement points. The timings shown in this section are valid for all ’C702, ’C722, ’CxxxA, and ’CxxxB family devices except the ’C3xxA device.

Table 18–3. EEPROM Timing Requirements for Programming

Parameter		Min	Max	Unit
$t_{w(PGM)B}$	Pulse duration, programming signal to ensure valid data is stored (byte mode)	10		ms
$t_{w(PGM)AR}$	Pulse duration, programming signal to ensure valid data is stored (array mode)	20		ms

Table 18–4. Recommended EPROM Operating Conditions for Programming

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.75	5.5	6	V
V_{PP}	Supply voltage at MC pin		13	13.2	13.5	V
I_{PP}	Supply current at MC pin during programming (V_{PP} = 13 V)			30	50	mA
SYSCLK	System clock operating frequency	Divide-by-4 clock	0.5		5	MHz
		Divide-by-1 clock	2		5	MHz

Table 18–5. EPROM Timing Requirements for Programming

Parameter		Min	Nom	Max	Unit
$t_{w(EPGM)}$	Pulse duration, programming signal (see Note)	0.40	0.50	3	ms

Note: Programming pulse is active when both EXE (EPCTL.0) and VPPS (EPCTL.6) are set.

Note:

The parameters V_{PP} and $t_{w(EPGM)}$ for EPROM are different for the TMS370Cxxx devices (vs. the TMS370CxxxA or TMS370CxxxB devices). Refer to subsection A.9.3, *Differences in EPROM Specifications,* on page A-10.

18.7 TMS370Cx0xA and TMS370Cx0x Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx0xA and TMS370Cx0x categories. These devices include the TMS370C002A, TMS370C302A, TMS370C702, and SE370C702.

18.7.1 TMS370Cx0xA and TMS370Cx0x Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–6 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–6. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage (see Note 1)		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).
2) $\overline{\text{RESET}}$ must be activated externally when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–7. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V ≤ V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		20	36	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		13	25	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		5	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		6.5	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		4.5	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		1.5	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		1	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA x (total load capacitance + crystal capacitance in pF).
4) Maximum operating current = 5.6 (SYSCLK) + 8 mA.
5) Maximum standby current = 3 (SYSCLK) + 2 mA (OSC POWER bit = 0).
6) Maximum standby current = 2.24 (SYSCLK) + 1.9 mA (OSC POWER bit = 1, only valid up to 3 MHz of SYSCLK).

18.7.2 TMS370Cx0xA and TMS370Cx0x Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–8. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–6.

2) This pulse can be either a high pulse, as illustrated in Figure 18–6, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–6. External Clock Timing for Divide-by-4 Clock

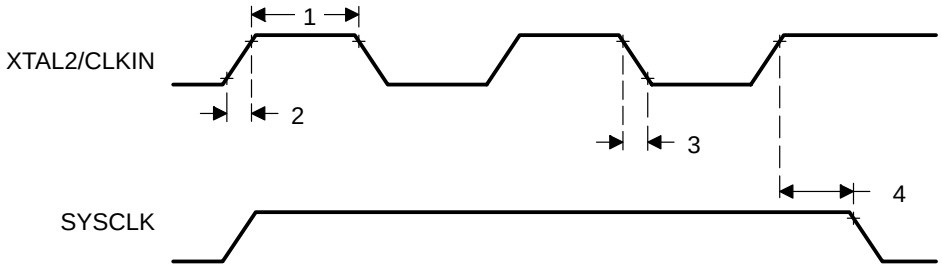


Table 18–9. External Clocking Requirements for Divide-by-1 Clock (PLL) (see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–6.

2) This pulse can be either a high pulse, as illustrated in Figure 18–7, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–7. External Clock Timing for Divide-by-1 Clock

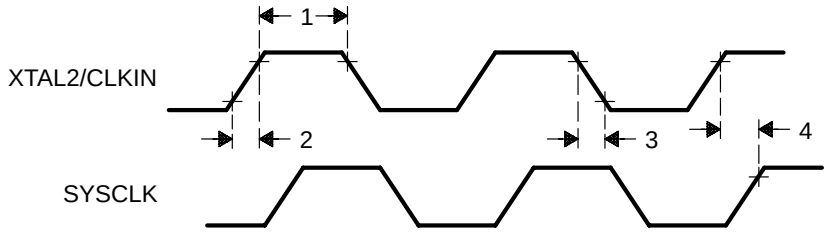
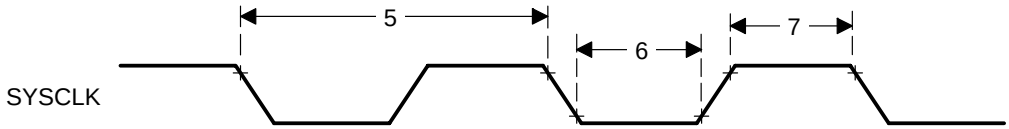


Table 18–10. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–8. SYSCLK TIming



18.8 TMS370Cx1xA and TMS370Cx1xB Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx1xA and the TMS370Cx1xB categories. These devices include the TMS370C010A, TMS370C012A, TMS370C310A, TMS370C311A, TMS370C312A, TMS370C712A, TMS370C712B, SE370C712A, and SE370C712B.

18.8.1 TMS370Cx1xA and TMS370Cx1xB Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–11 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–11. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage (see Note 1)		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).
2) $\overline{\text{RESET}}$ must be externally activated when V_{CC} or SYSCLK is not within the recommended operating range.

Table 18–12. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V ≤ V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			± 10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		20	36	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		13	25	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		5	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		6.5	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		4.5	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		1.5	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		1	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
4) Maximum operating current = 5.6(SYSCLK) + 8 mA.
5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.8.2 TMS370Cx1xA and TMS370Cx1xB Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–13. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–11.

2) This pulse may be either a high pulse, as illustrated in Figure 18–9, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–9. External Clock Timing for Divide-by-4 Clock

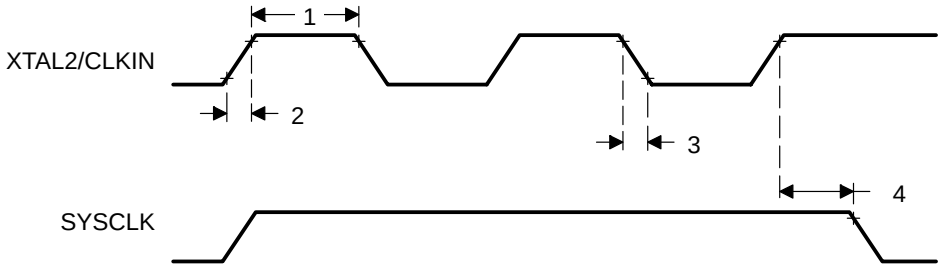


Table 18–14. External Clocking Requirements for Divide-by-1 Clock (PLL) (see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–11.

2) This pulse may be either a high pulse, as illustrated in Figure 18–10, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–10. External Clock Timing for Divide-by-1 Clock

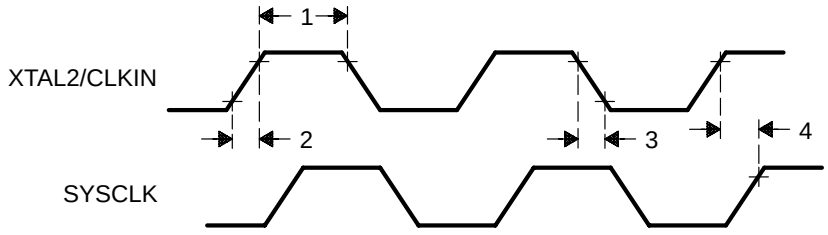
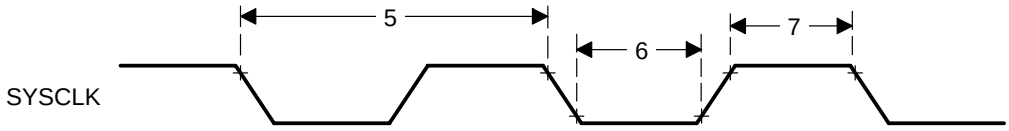


Table 18–15. Switching Characteristics and Timing Requirements (see Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system clock cycle time = 1 / SYSCLK

Figure 18–11. SYSCLK Timing



18.9 TMS370Cx2xA and TMS370Cx2x Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx2xA TMS370Cx2x categories. These devices include the TMS370C020A, TMS370C022A, TMS370C320A, TMS370C322A, TMS370C722, and SE370C722.

18.9.1 TMS370Cx2xA and TMS370Cx2x Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–16 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–16. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltages are with respect to V_{SS} (ground).
2) $\overline{\text{RESET}}$ must be externally activated when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–17. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		30	45	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		20	30	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		7	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		2	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.

2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.

3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

4) Maximum operating current = 7.6(SYSCLK) + 7 mA.

5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).

6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.9.2 TMS370Cx2xA and TMS370Cx2x Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–18. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–16.

2) This pulse may be either a high pulse, as illustrated in Figure 18–12, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–12. External Clock Timing for Divide-by-4 Clock

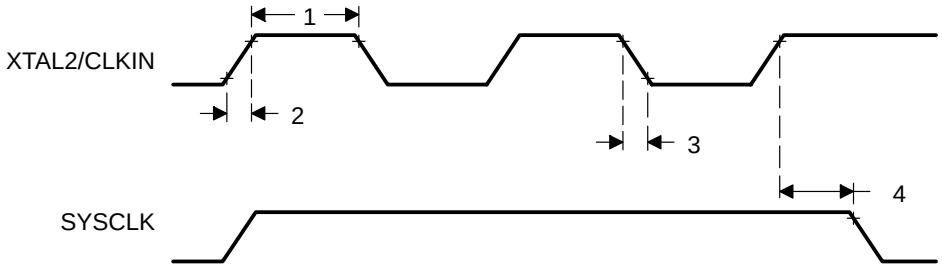


Table 18–19. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–16.

2) This pulse can be either a high pulse, as illustrated in Figure 18–13, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–13. External Clock Timing for Divide-by-1 Clock

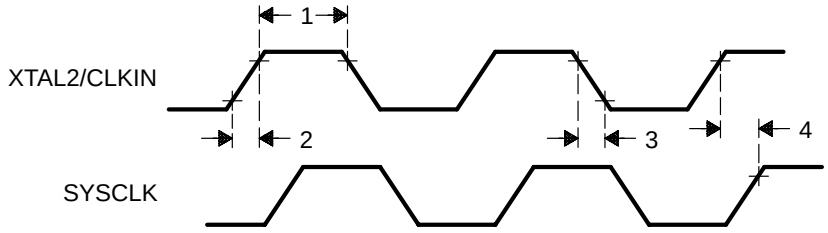
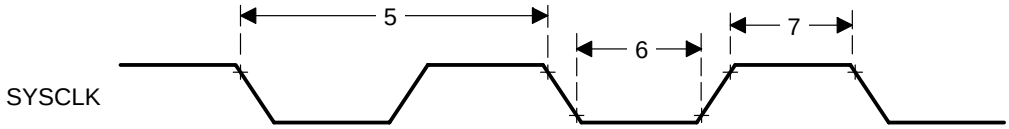


Table 18–20. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system-clock cycle time = 1/SYSCLK

Figure 18–14. SYSCLK Timing



18.10 TMS370Cx32A Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx32A category. These devices include the TMS370C032A, TMS370C332A, TMS370C732A, and SE370C732A.

18.10.1 TMS370Cx32A Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–21 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–21. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.
2) $\overline{\text{RESET}}$ must be externally activated when V_{CC1} or SYSCLK is not within the recommended operating range.

Table 18–22. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 μA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V ≤ V_I ≤ 0.3 V			10	μA
			0.3 V < V_I ≤ 13 V			650	μA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC1}			± 10	μA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC1}$	–50			μA
			V_{OH} = 2.4 V	–2			mA
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0		See Notes 2 and 3 SYSCLK = 5 MHz		35	45	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		25	35	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		10	14	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 0		See Notes 2 and 3 SYSCLK = 5 MHz		12	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	13	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		3	4	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 1		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.0	mA
I_{CC1}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		15	40	μA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

18.10.2 TMS370Cx32A Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–23. External Clocking Requirements For Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–21.

2) This pulse may be either a high pulse, as illustrated in Figure 18–15, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–15. External Clock Timing for Divide-by-4 Clock

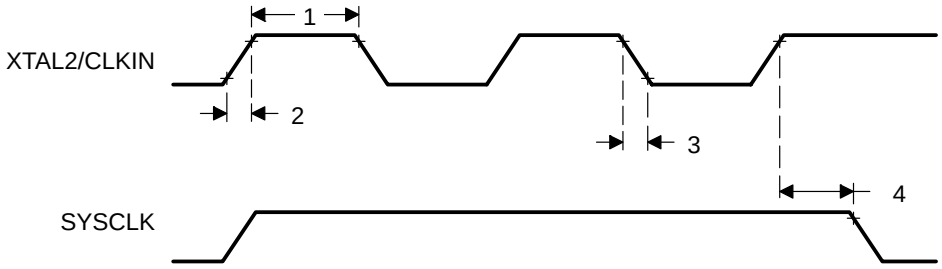


Table 18–24. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–21.

2) This pulse can be either a high pulse, as illustrated in Figure 18–16, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–16. External Clock Timing for Divide-by-1 Clock

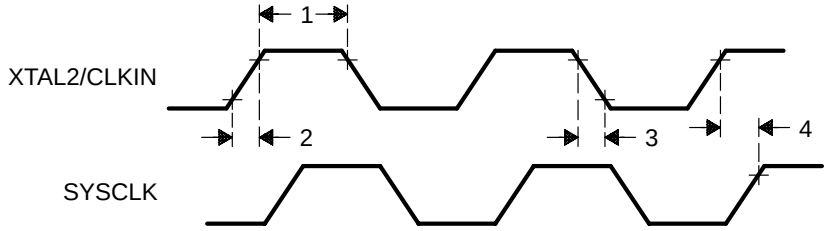
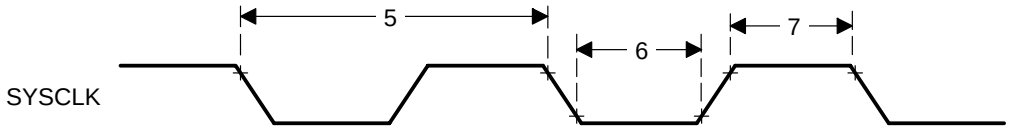


Table 18–25. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c$–20	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c$ + 20	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–17. SYSCLK Timing



18.11 TMS370Cx36A Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx36A category. These devices include the TMS370C036A, TMS370C736A, and SE370C736A.

18.11.1 TMS370Cx36A Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–26 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–26. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CCSTBY}	Standby RAM supply voltage		4.5	5	5.5	V
V_{CCSTBY}	Standby RAM data retention supply voltage (see Note 2)		3		5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.

2) $\overline{\text{RESET}}$ must be externally activated when V_{CC1} or SYSCLK is not within the recommended operating range.

Table 18–27. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage, all outputs		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage	All outputs except PACT outputs	I_{OH} = –50 µA	$0.9V_{CC1}$			V
		PACT outputs	I_{OH} = –50 µA	$0.7V_{CC1}$			V
		All outputs	I_{OH} = –2 mA	2.4			V
I_I	Input current	MC pin	0 V < V_I < 0.3 V			10	µA
			0.3 V < V_I < V_{CC1} – 0.3 V			50	µA
			V_{CC1} – 0.3 < V_I < V_{CC1} + 0.3 V			10	µA
			V_{CC1} + 0.3 V < V_I < 13 V			650	µA
		I/O pins	0 V < V_I < V_{CC1}			±10	µA
I_{OL}	Low-level output current, all outputs		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current, all outputs		V_{OH} = 0.9 V_{CC1}	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0		See Notes 1 and 2 SYSCLK = 5 MHz		36	45	mA
	Supply current (STANDBY mode) OSC POWER bit = 0		See Notes 1 and 2 SYSCLK = 5 MHz		7	12	mA
	Supply current (HALT mode)		See Notes 1 and 2 XTAL2/CLKIN < 0.2 V		5	30	µA
I_{CCSTBY}	Standby RAM supply current (operating mode) OSC POWER bit = 0		SYSCLK = 5 MHz V_{CCSTBY} = 4.5 V		1	1.5	mA

Notes: 1) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.

2) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

18.11.2 TMS370Cx36A Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–28. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–26.

2) This pulse may be either a high pulse, as illustrated in Figure 18–18, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–18. External Clock Timing for Divide-by-4 Clock

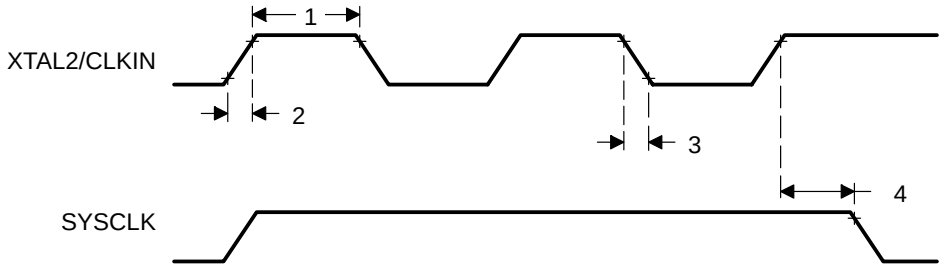


Table 18–29. External Clocking Requirements For Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–26.

2) This pulse can be either a high pulse, as illustrated in Figure 18–19, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–19. External Clock Timing for Divide-by-1 Clock

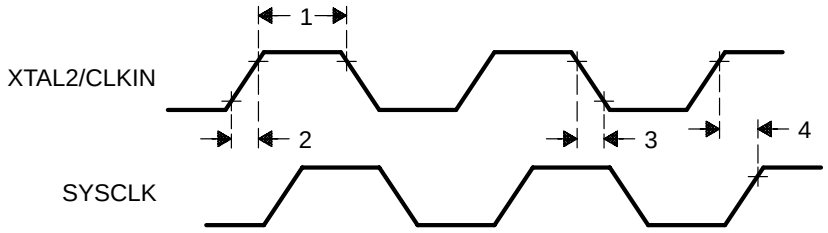
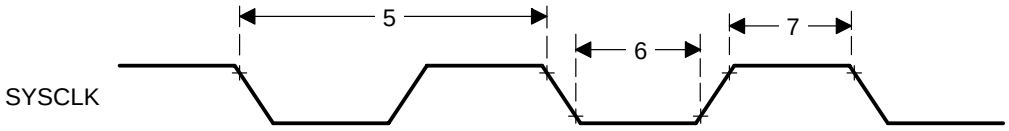


Table 18–30. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c$–20	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c$ + 20	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–20. SYSCLK Timing



18.12 TMS370Cx4xA Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx4xA category. These devices include the TMS370C040A, TMS370C042A, TMS370C340A, TMS370C342A, TMS370C742A, and SE370C742A.

18.12.1 TMS370Cx4xA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–31 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–31. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		Microcomputer mode	V_{SS}		0.3	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).

2) $\overline{\text{RESET}}$ must be externally activated when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–32. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level digital output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 6)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 3)		See Notes 1 and 2 SYSCLK = 5 MHz		30	45	mA
			See Notes 1 and 2 SYSCLK = 3 MHz		20	30	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		7	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 4)		See Notes 1 and 2 SYSCLK = 5 MHz		10	17	mA
			See Notes 1 and 2 SYSCLK = 3 MHz		8	11	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 5)		See Notes 1 and 2 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		2	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 1 XTALK2/CLKIN < 0.2 V		2	30	µA

Notes:
1) Microcontroller single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.
2) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
3) Maximum operating current = 7.6(SYSCLK) + 7 mA.
4) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0.)
5) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK.)
6) Input current I_{PP} is a maximum of 50 mA only when an EPROM is being programmed.

18.12.2 TMS370Cx4xA Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–33. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–31.

2) This pulse can be either a high pulse, as illustrated in Figure 18–21, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–21. External Clock Timing for Divide-by-4 Clock

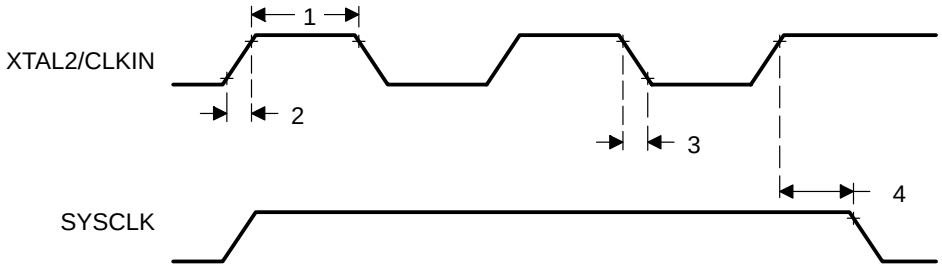


Table 18–34. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–31.

2) This pulse can be either a high pulse, as illustrated in Figure 18–22, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–22. External Clock Timing for Divide-by-1 Clock

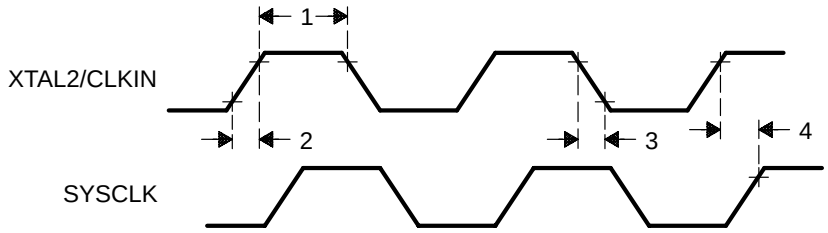
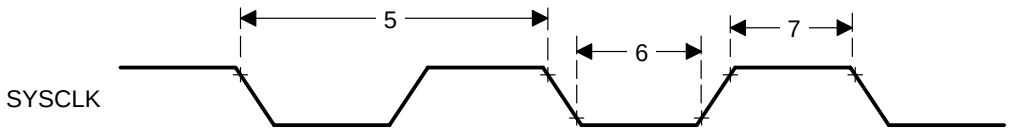


Table 18–35. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system-clock cycle time = 1/SYSCLK.

Figure 18–23. SYSCLK Timing



18.13 TMS370Cx5xA and TMS370Cx5xB Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx5xA and TMS370Cx5xB categories. These devices include the following:

TMS370C050A	TMS370C150A	TMS370C250A	TMS370C350A
TMS370C052A	TMS370C352A	TMS370C452A	TMS370C353A
TMS370C056A	TMS370C156A	TMS370C256A	TMS370C356A
TMS370C456A	TMS370C756A	SE370C756A	TMS370C058A
TMS370C358A	TMS370C758A	TMS370C758B,	SE370C758A
SE370758B	TMS370C059A	TMS370C759A	SE370C759A

Note:

Some electrical specifications and timings differ for TMS370Cx5x devices. Refer to Appendix A.

18.13.1 TMS370Cx5xA and TMS370Cx5xB Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–36 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–36. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC2}	Digital I/O supply voltage		4.5	5	5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS2}	Digital I/O supply ground		–0.3	0	0.3	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		MC (non-WPO mode)	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage (see Note 3)	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microprocessor mode	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.

2) $\overline{\text{RESET}}$ must be externally activated when V_{CC1} or SYSCLK is not within the recommended operating range.

3) The basic microcomputer and microprocessor operating modes are selected by the voltage level applied to the dedicated MC pin two system-clock cycles ($2t_C$) before $\overline{\text{RESET}}$ goes inactive (high). The WPO mode can be selected any time a sufficient voltage is present on MC.

CAUTION

You cannot use the internal connectors between pins (e.g., the connector between V_{SS1} and V_{SS2}) for a jumper from one side of the chip to another.

Table 18–37. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage (see Note 1)		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I < V_{CC1} – 0.3 V			50	µA
			V_{CC1} – 0.3 V ≤ V_I ≤ V_{CC1} + 0.3 V			10	µA
			V_{CC1} + 0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 2)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC1}			±10	µA
I_{OL}	Low-level output current (see Note 1)		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC1}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0 (see Note 5)	TMS370Cx50A TMS370Cx52A	See Notes 3 and 4 SYSCLK = 5 MHz		30	45	mA
		TMS370Cx53A TMS370Cx56A TMS370Cx58A TMS370Cx58B			35	56	mA

† TMS370Cx59 operates only up to 3-MHz SYSCLK.

Notes: 1) In prior versions of the TMS370 family, I_{OL} was equal to 2 mA for ports A, B, C, and D and $\overline{\text{RESET}}$.

2) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.

3) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.

4) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

5) Maximum operating current for TMS370Cx50A and TMS370Cx52A = 7.6(SYSCLK) + 7 mA. Maximum operating current for the ’Cx53A, ’Cx56A, ’Cx58A, and ’Cx58B = 10(SYSCLK) + 5.8 mA.

6) Maximum standby current for the ’Cx5xA and ’Cx5xB = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).

7) Maximum standby current for the ’Cx5xA and ’Cx5xB = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK.)

Table 18–37. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range (Continued)

Parameter			Test Conditions	Min	Typ	Max	Unit
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0 (see Note 5)	TMS370Cx50A TMS370Cx52A	See Notes 3 and 4 SYSCLK = 3 MHz		20	30	mA
		TMS370Cx53A TMS370Cx56A TMS370Cx58A TMS370Cx58B			25	36	mA
		TMS370Cx59A†			46	55	mA
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0 (see Note 5)	TMS370Cx50A TMS370Cx52A	See Notes 3 and 4 SYSCLK = 0.5 MHz		5	11	mA
		TMS370Cx53A TMS370Cx56A TMS370Cx58A TMS370Cx58B			13	18	mA
		TMS370Cx59A†			22	28	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 6)		See Notes 3 and 4 SYSCLK = 5 MHz		12	17	mA
			See Notes 3 and 4 SYSCLK = 3 MHz		8	11	mA
			See Notes 3 and 4 SYSCLK = 0.5 MHz		2.5	3.5	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 7)		See Notes 3 and 4 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 3 and 4 SYSCLK = 0.5 MHz		2	3	mA
I_{CC1}	Supply current (HALT mode)		See Note 3 XTAL2/CLKIN < 0.2 V		2	30	µA

† TMS370Cx59A operates only up to 3-MHz SYSCLK.

Notes: 1) I then prior versions of the TMS370 family, I_{OL} was equal to 2 mA for ports A, B, C, and D and $\overline{\text{RESET}}$.
2) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
3) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.
4) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
5) Maximum operating current for the ’Cx50A and ’Cx52A = 7.6(SYSCLK) + 7 mA. Maximum operating current for the ’Cx53A, ’Cx56A, ’Cx58A and ’Cx58B = 10(SYSCLK) + 5.8 mA.
6) Maximum standby current for the ’Cx5xA and ’Cx5xB = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
7) Maximum standby current for the ’Cx5xA and ’Cx5xB = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK.)

18.13.2 TMS370Cx5xA and TMS370Cx5xB Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–38. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency (see Note 3)	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Notes 4 and 5)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–36.

2) This pulse can be either a high pulse, as illustrated in Figure 18–24, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) TMS370Cx59A operates up to 12-MHz CLKIN.

4) TMS370Cx59A operates up to 3-MHz SYSCLK.

5) SYSCLK = CLKIN/4

Figure 18–24. External Clock Timing for Divide-by-4 Clock

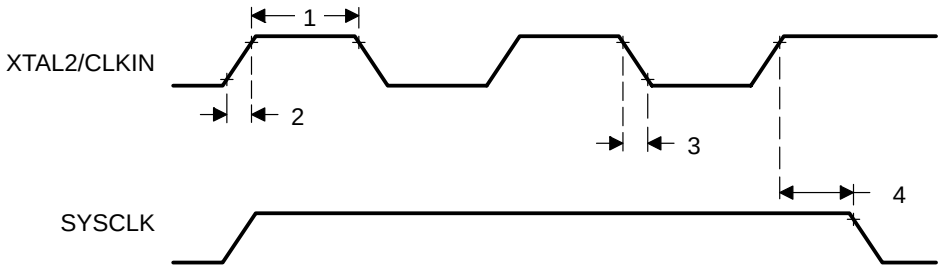


Table 18–39. External Clocking Requirements for Divide-by-1 Clock (PLL) (see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency (see Note 3)	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Notes 4 and 5)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–36.

2) This pulse can be either a high pulse, as illustrated in Figure 18–25, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) TMS370Cx59A operates up to 3-MHz CLKIN (for the divide-by-1 clock option).

4) TMS370Cx59A operates up to 3-MHz SYSCLK.

5) SYSCLK = CLKIN/1

Figure 18–25. External Clock Timing for Divide-by-1 Clock

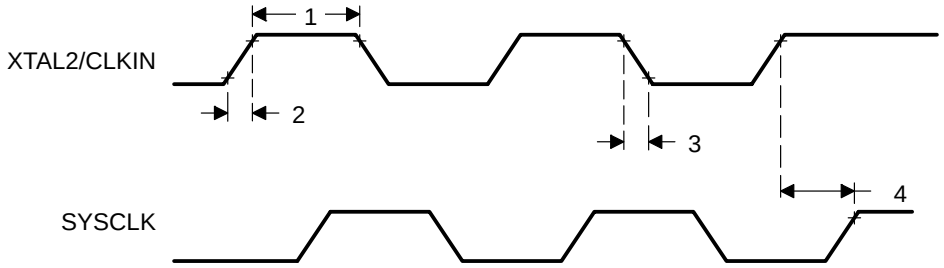


Table 18–40. Switching Characteristics and Timing Requirements for External Read and Write (See Note 1)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 25$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns
8	$t_{d(SCL-A)}$	Delay time, SYSCLK low to address, R/$\overline{W}$, and $\overline{OCF}$ valid			$0.25t_c + 75$	ns
9	$t_{v(A)}$	Valid time, address valid to $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSE2}$, $\overline{CSH1}$, $\overline{CSH2}$, $\overline{CSH3}$, and $\overline{CSPF}$ low		$0.5t_c - 90$		ns
10	$t_{su(D)}$	Setup time, write data to $\overline{EDS}$ high		$0.75t_c - 80$ See Note 2		ns
11	$t_{h(EH-A)}$	Hold time, address, R/$\overline{W}$, and $\overline{OCF}$ from $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSE2}$, $\overline{CSH1}$, $\overline{CSH2}$, $\overline{CSH3}$, and $\overline{CSPF}$ high		$0.5t_c - 60$		ns
12	$t_{h(EH-D)W}$	Hold time, write data from $\overline{EDS}$ high		$0.75t_c + 15$		ns
13	$t_{d(DZ-EL)}$	Delay time, data bus high impedance to $\overline{EDS}$ low (read cycle)		$0.25t_c - 35$		ns
14	$t_{d(EH-D)}$	Delay time, $\overline{EDS}$ high to data bus enable (read cycle)		$1.25t_c - 40$		ns
15	$t_{d(EL-DV)R}$	Delay time, $\overline{EDS}$ low to read data valid			$t_c - 95$ See Note 2	ns
16	$t_{h(EH-D)R}$	Hold time, read from $\overline{EDS}$ high		0		ns
17	$t_{su(WT-SCH)}$	Setup time, $\overline{WAIT}$ to SYSCLK high		$0.25t_c + 70$ See Note 3		ns
18	$t_{h(SCH-WT)}$	Hold time, $\overline{WAIT}$ time from SYSCLK high		0		ns
19	$t_{d(EL-WTV)}$	Delay time, $\overline{EDS}$ low to $\overline{WAIT}$ valid			$0.5t_c - 60$	ns
20	t_w	Pulse duration, $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSE2}$, $\overline{CSH1}$, $\overline{CSH2}$, $\overline{CSH3}$, and $\overline{CSPF}$ low		$t_c - 80$ See Note 2	$t_c + 40$ See Note 2	ns
21	$t_{d(AV-DV)R}$	Delay time, address valid to read data valid			$1.5t_c - 115$ See Note 2	ns
22	$t_{d(AV-WTV)}$	Delay time, address valid to $\overline{WAIT}$ valid			$t_c - 115$	ns
23	$t_{d(AV-EH)}$	Delay time, address valid to $\overline{EDS}$ high (end of write)		$1.5t_c - 85$ See Note 2		ns

Notes: 1) t_c = system-clock cycle time = 1/SYSCLK

2) If wait states, PFWait, or the autowait feature is used, add t_c to this value for each wait state invoked.

3) If the autowait feature is enabled, the $\overline{WAIT}$ input can assume a “don’t care” condition until the third cycle of the access. The $\overline{WAIT}$ signal must be synchronized with the high pulse of the SYSCLK signal while still conforming to the minimum setup time.

Figure 18–26. External Read Timing

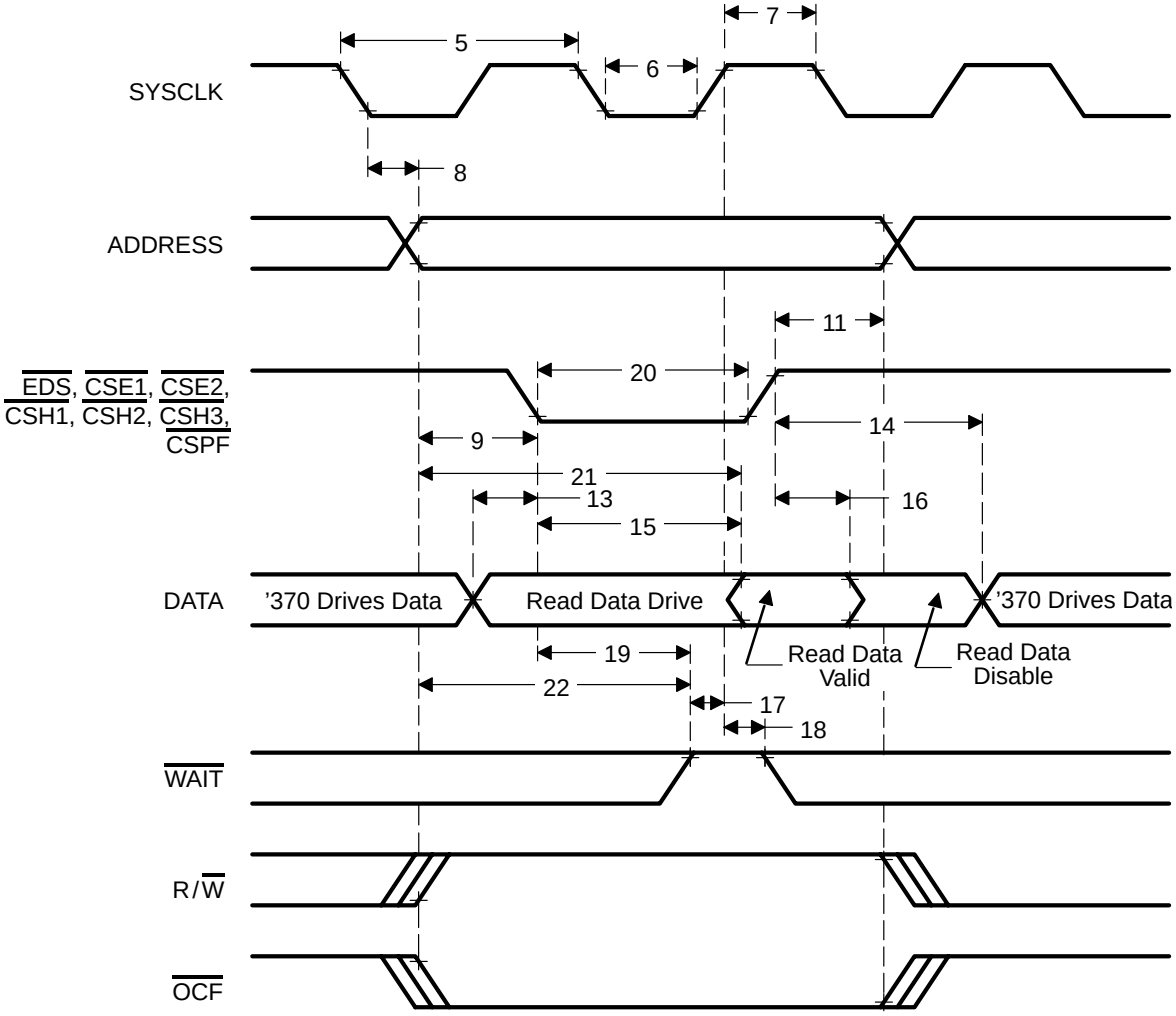
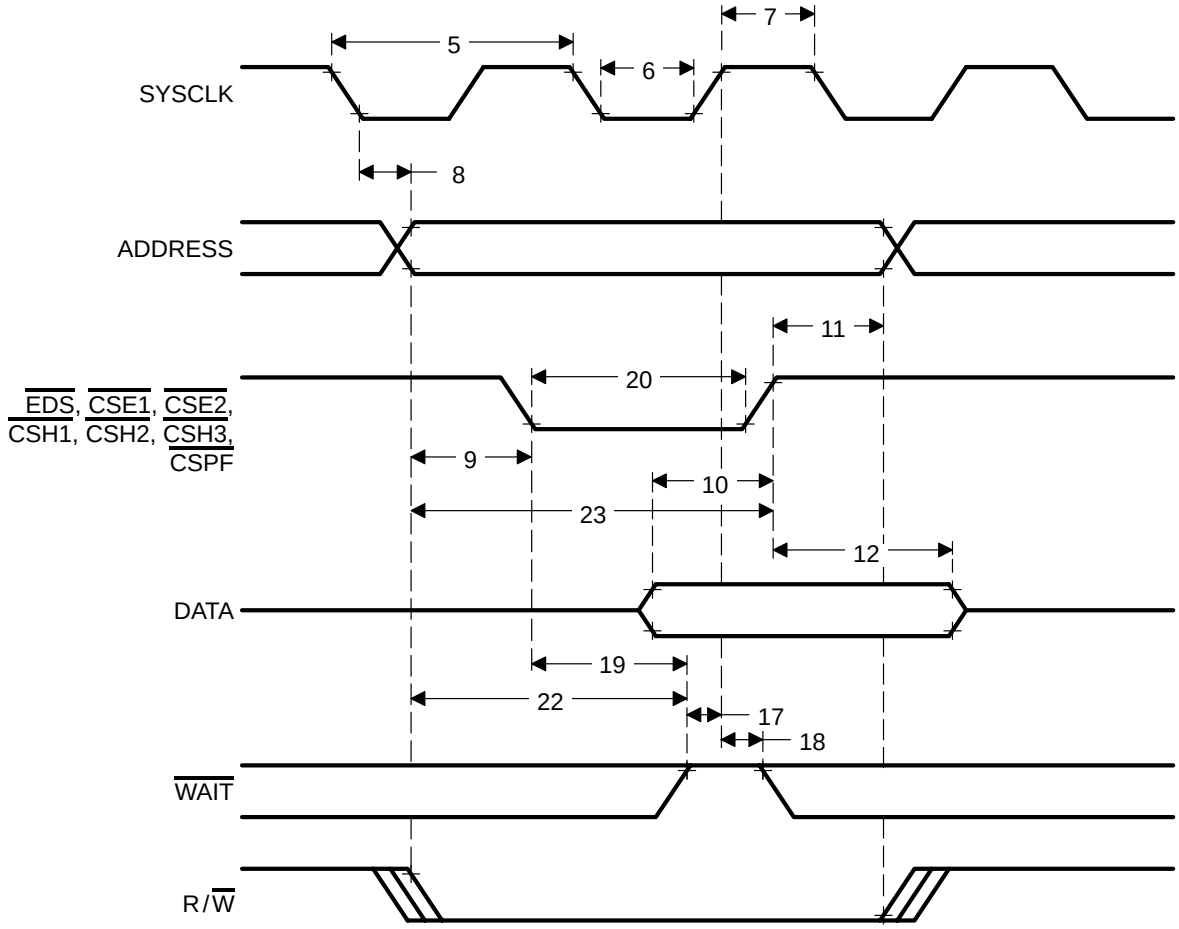


Figure 18–27. External Write Timing



18.14 TMS370Cx6xA Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx6xA category. These devices include the TMS370C067A, TMS370C068A, TMS370C069A, TMS370C768A, TMS370C769A, SE370C768A, and SE370C769A.

18.14.1 TMS370Cx6xA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–41 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–41. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC2}	Digital I/O supply voltage		4.5	5	5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS2}	Digital I/O supply ground		–0.3	0	0.3	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		MC (non-WPO mode)	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage (see Note 3)	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microprocessor mode	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.
2) $\overline{\text{RESET}}$ must be externally activated when V_{CC1} or SYSCLK is out of the recommended operating range.
3) The basic microcomputer and microprocessor operating modes are selected by the voltage level applied to the dedicated MC pin two system-clock cycles ($2t_C$) before $\overline{\text{RESET}}$ goes inactive (high). The WPO mode can be selected any time a sufficient voltage is present on MC.

You cannot use the internal connections between pins (for example, the connection between V_{SS1} and V_{SS2}) for a jumper from one side of the chip to the other.

Table 18–42. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC1}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I < V_{CC1}–0.3 V			50	µA
			V_{CC1}– 0.3 V ≤ V_I ≤ V_{CC1} + 0.3 V			10	µA
			V_{CC1} + 0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC1}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC1}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC1}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)	’Cx67A, ’Cx68A	See Notes 2 and 3 SYSCLK = 5 MHz		35	56	mA
		’Cx67A, ’Cx68A	See Notes 2 and 3 SYSCLK = 3 MHz		25	36	mA
		’Cx69A†			46	55	mA
		’Cx67A, ’Cx68A	See Notes 2 and 3 SYSCLK = 0.5 MHz		13	18	mA
		’Cx69A†			22	28	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		12	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2.5	3.5	mA
I_{CC1}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3	mA
I_{CC1}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		2	30	µA

† TMS370Cx69A operates only up to 3 MHz SYSLCK.

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
4) Maximum operating current = 10(SYSCLK) + 5.8 mA.
5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.14.2 TMS370Cx6xA Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–43. External Clocking Requirements for Divide-by-4 Clock (see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency (see note 3)	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Notes 4 and 5)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions table in Table 18–41.

2) This pulse can be either a high pulse, as illustrated in Figure 18–28, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) TMS370Cx69A operates up to 12 MHz CLKIN.

4) TMS370Cx69A operates up to 3 MHz SYSCLK.

5) SYSCLK = CLKIN/4

Figure 18–28. External Clock Timing for Divide-by-4 Clock

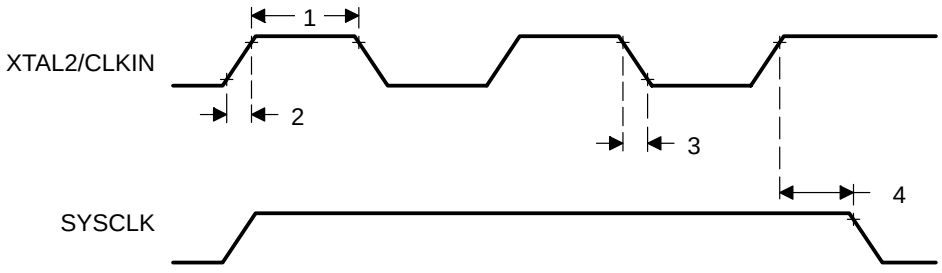


Table 18–44. External Clocking Requirements for Divide-by-1 Clock (PLL)(See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency (see Note 3)	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Notes 4 and 5)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–41.

2) This pulse can be either a high pulse, as illustrated in Figure 18–29, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) TMS370Cx69A operates up to 3-MHz CLKIN (for the divide-by-1 clock option).

4) TMS370Cx69A operates up to 3 MHz SYSCLK.

5) SYSCLK = CLKIN/1

Figure 18–29. External Clock Timing for Divide-by-1 Clock

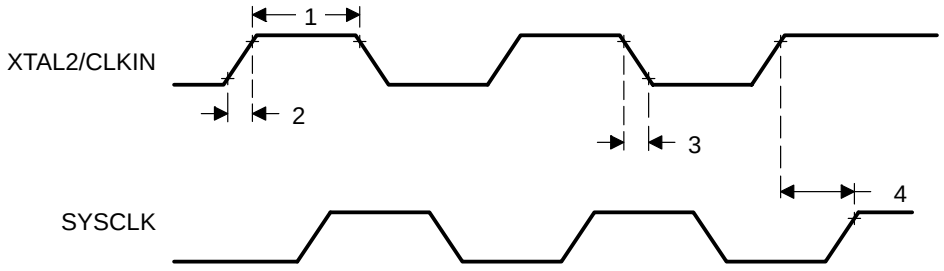


Table 18–45. Switching Characteristics and Timing Requirements for External Read and Write (See Note 1)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 25$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns
8	$t_{d(SCL-A)}$	Delay time, SYSCLK low to address, R/$\overline{W}$, and $\overline{OCF}$ valid			$0.25t_c + 75$	ns
9	$t_{v(A)}$	Valid time, address valid to $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSH1}$, and $\overline{CSPF}$ low		$0.5t_c - 90$		ns
10	$t_{su(D)}$	Setup time, write data to $\overline{EDS}$ high		$0.75t_c - 80$ See Note 2		ns
11	$t_{h(EH-A)}$	Hold time, address, R/$\overline{W}$, and $\overline{OCF}$ from $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSH1}$, and $\overline{CSPF}$ high		$0.5t_c - 60$		ns
12	$t_{h(EH-D)W}$	Hold time, write data time from $\overline{EDS}$ high		$0.75t_c + 15$		ns
13	$t_{d(DZ-EL)}$	Delay time, data bus high impedance to $\overline{EDS}$ low (read cycle)		$0.25t_c - 35$		ns
14	$t_{d(EH-D)}$	Delay time, $\overline{EDS}$ high to data bus enable (read cycle)		$1.25t_c - 40$		ns
15	$t_{d(EL-DV)R}$	Delay time, $\overline{EDS}$ low to read data valid			$t_c - 95$ See Note 2	ns
16	$t_{h(EH-D)R}$	Hold time, read from $\overline{EDS}$ high		0		ns
17	$t_{su(WT-SCH)}$	Setup time, $\overline{WAIT}$ to SYSCLK high		$0.25t_c + 70$ See Note 3		ns
18	$t_{h(SCH-WT)}$	Hold time, $\overline{WAIT}$ from SYSCLK high		0		ns
19	$t_{d(EL-WTV)}$	Delay time, $\overline{EDS}$ low to $\overline{WAIT}$ valid			$0.5t_c - 60$	ns
20	t_w	Pulse duration, $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSH1}$, and $\overline{CSPF}$ low		$t_c - 80$ See Note 2	$t_c + 40$ See Note 2	ns
21	$t_{d(AV-DV)R}$	Delay time, address valid to read data valid			$1.5t_c - 115$ See Note 2	ns
22	$t_{d(AV-WTV)}$	Delay time, address valid to $\overline{WAIT}$ valid			$t_c - 115$	ns
23	$t_{d(AV-EH)}$	Delay time, address valid to $\overline{EDS}$ high (end of write)		$1.5t_c - 85$ See Note 2		ns

Notes: 1) t_c = system clock cycle time = 1/SYSCLK

2) If wait states, PFWait, or the autowait feature is used, add t_c to this value for each wait state invoked.

3) If the autowait feature is enabled, the $\overline{WAIT}$ input can assume a “don’t care” condition until the third cycle of the access. The $\overline{WAIT}$ signal must be synchronized with the high pulse of the SYSCLK signal while still conforming to the minimum setup time.

Figure 18–30. External Read Timing

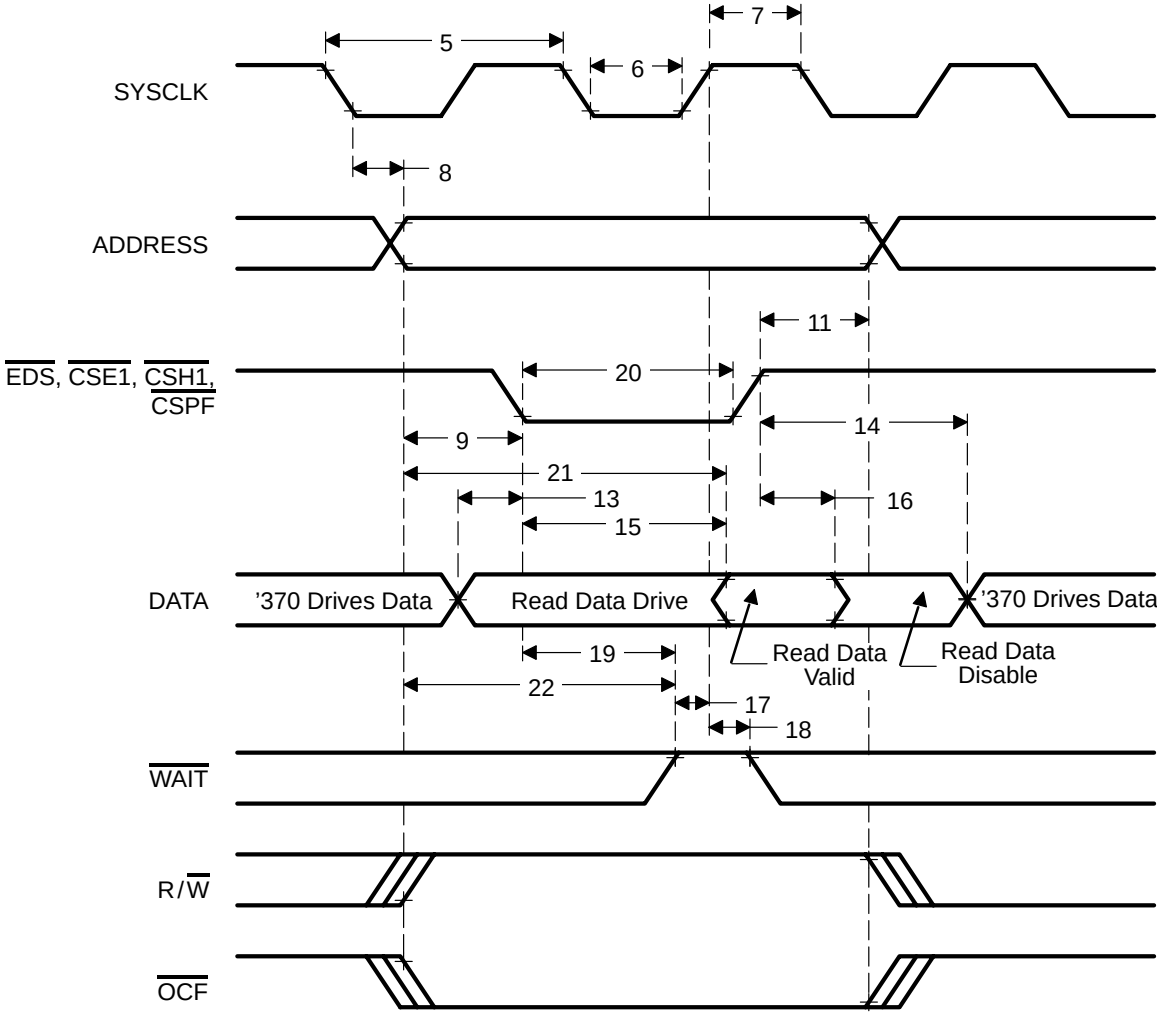
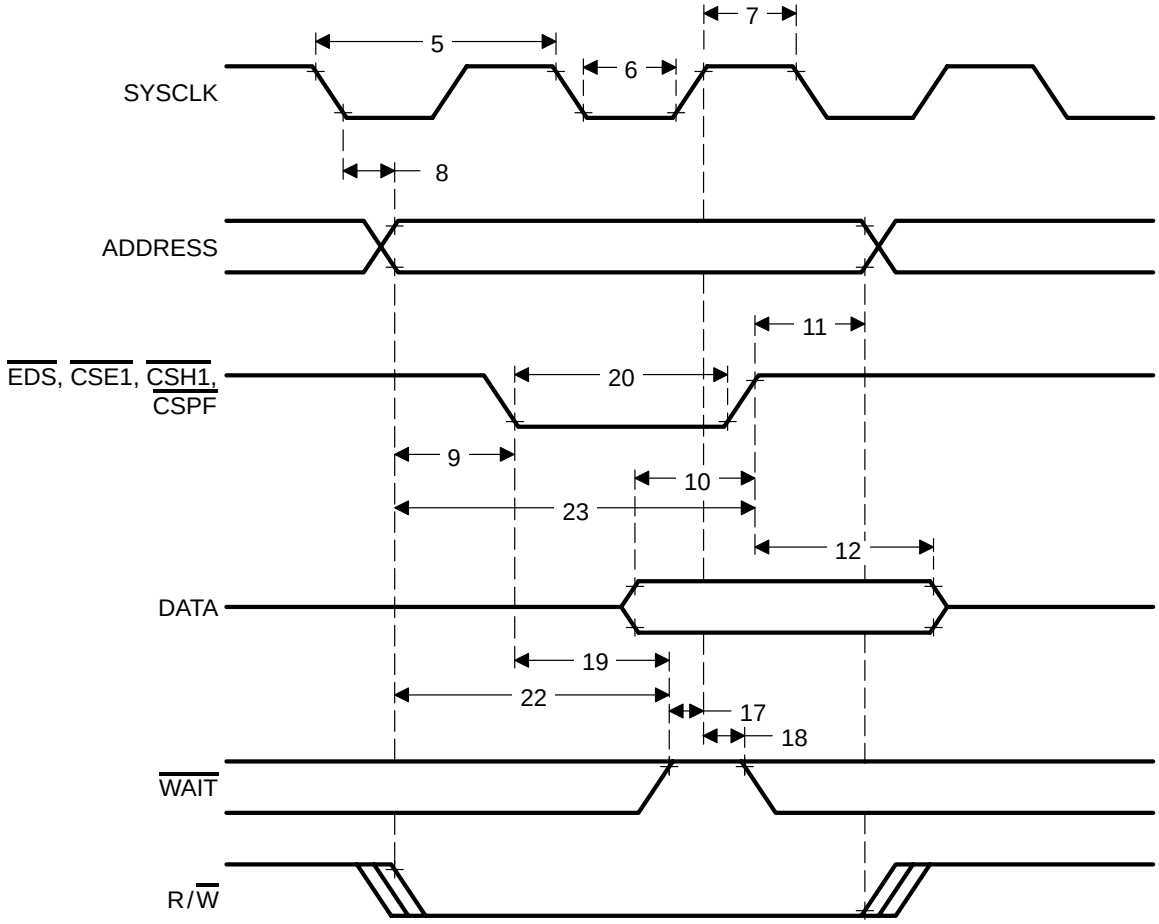


Figure 18–31. External Write Timing



18.15 TMS370Cx7xA Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx7xA category. These devices include the TMS370C077A, TMS370C777A, and SE370C777A.

18.15.1 TMS370Cx7xA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–46 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–46. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC2}	Digital I/O supply voltage		4.5	5	5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS2}	Digital I/O supply ground		–0.3	0	0.3	V
V_{SS3}	Analog supply ground		–0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		MC (non-WPO mode)	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage (see Note 3)	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microprocessor mode	$V_{CC1}-0.3$		$V_{CC1}+0.3$	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.
2) $\overline{\text{RESET}}$ must be activated externally when V_{CC1} or SYSCLK is out of the recommended operating range.
3) The basic microcomputer and microprocessor operating modes are selected by the voltage level applied to the dedicated MC pin two system clock cycles ($2t_c$) before $\overline{\text{RESET}}$ goes inactive (high). The WPO mode can be selected any time a sufficient voltage is present on MC.

You cannot use the internal connections between pins (for example, the connection between V_{SS1} and V_{SS2}) for a jumper from one side of the chip to the other.

Table 18–47. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC1}$			V
			I_{OH} = –2 mA	2.4			
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I < V_{CC1} – 0.3 V			50	
			V_{CC1} – 0.3 V ≤ V_I ≤ V_{CC1} + 0.3 V			10	
			V_{CC1} + 0.3 V < V_I ≤ 13 V			650	
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC1}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC1}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		35	56	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		25	36	
			See Notes 2 and 3 SYSCLK = 0.5 MHz		13	18	
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		12	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2.5	3.5	
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3	
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		2	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5 MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
4) Maximum operating current = 10(SYSCLK) + 5.8 mA.
5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit =1; valid only up to 3-MHz SYSCLK).

18.15.2 TMS370Cx7xA Timings

Refer to Sections18.1 and 18.2 (page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–48. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–46.

2) This pulse can be either a high pulse, as illustrated in Figure 18–32, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–32. External Clock Timing for Divide-by-4 Clock

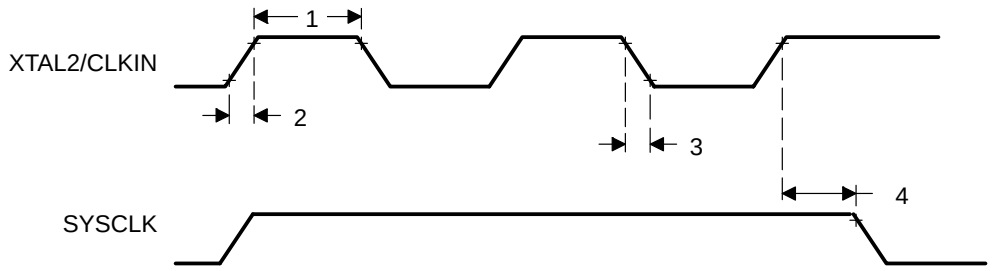


Table 18–49. External Clocking Requirements for Divide-by-1 Clock (PLL)(see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–46.

2) This pulse can be either a high pulse, as illustrated in Figure 18–33, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–33. External Clock Timing for Divide-by-1 Clock

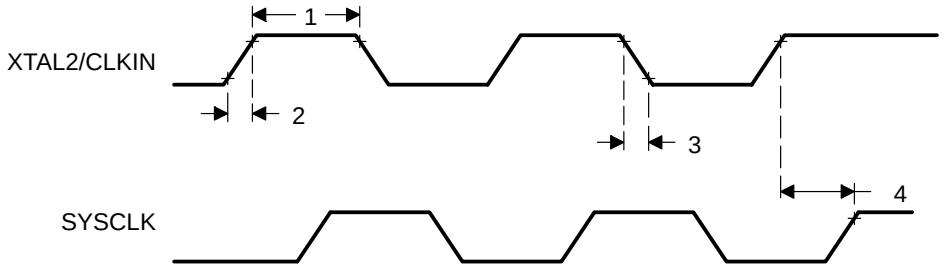
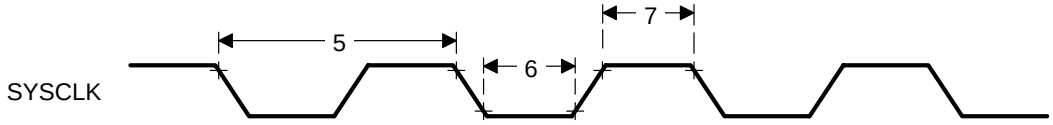


Table 18–50. Switching Characteristics and TIming Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 25$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–34. SYSCLK Timing



18.16 TMS370Cx8xA Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx8xA category. These devices include the TMS370C080A, TMS370C380A, TMS370C686A, and SE370C686A.

18.16.1 TMS370Cx8xA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–51 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–51. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).
2) $\overline{\text{RESET}}$ must be externally activated when V_{CC} or SYSCLK is not within the recommended operating range.

Table 18–52. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			± 10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		30	45	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		20	30	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		7	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		2	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.

2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.

3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

4) Maximum operating current = 7.6(SYSCLK) + 7 mA.

5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).

6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.16.2 TMS370Cx8xA Timings

Refer to Section 18.1 and Section 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–53. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–51.

2) This pulse may be either a high pulse, as illustrated in Figure 18–35, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–35. External Clock Timing for Divide-by-4 Clock

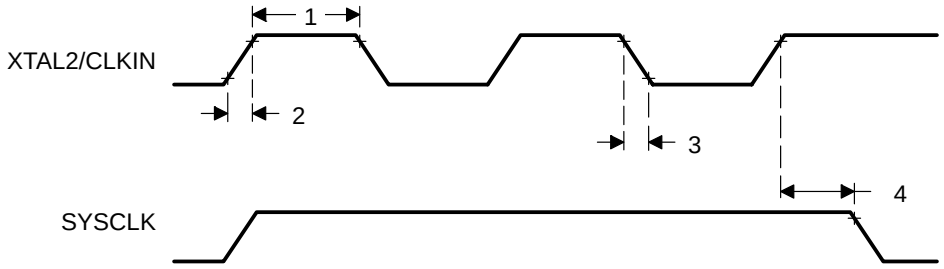


Table 18–54. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–51.

2) This pulse can be either a high pulse, as illustrated in Figure 18–36, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–36. External Clock Timing for Divide-by-1 Clock

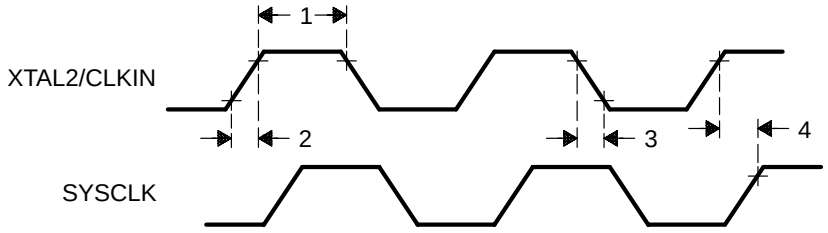
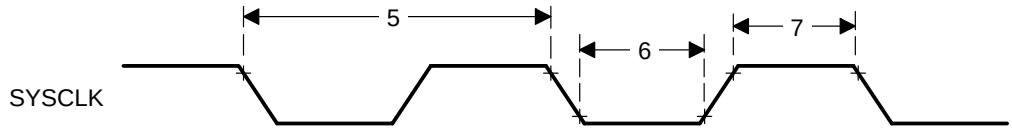


Table 18–55. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system-clock cycle time = 1/SYSCLK

Figure 18–37. SYSCLK Timing



18.17 TMS370Cx9xA Specifications

The tables in this section give specifications that apply to the devices in the TMS370Cx9xA category. These devices include the TMS370C090A, TMS370C792A, and SE370C792A.

18.17.1 TMS370Cx9xA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–56 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–56. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC3}	Analog supply voltage		3		5.5	V
V_{SS3}	Analog supply ground		– 0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).

2) $\overline{\text{RESET}}$ must be externally activated when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–57. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 μA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < $V_I \le$ 0.3 V			10	μA
			0.3 V < $V_I \le$ 13 V			650	μA
			12 V ≤ $V_I \le$ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ $V_I \le V_{CC}$			±10	μA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		$V_{OH} = 0.9V_{CC}$	–50			μA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		30	45	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		20	30	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		7	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		2	30	μA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.
2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.
3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
4) Maximum operating current = 7.6(SYSCLK) + 7 mA.
5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit =1; valid only up to 3-MHz SYSCLK).

18.17.2 TMS370Cx9xA Timings

Refer to Section 18.1 and Section 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–58. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–56.

2) This pulse can be either a high pulse, as illustrated in Figure 18–38, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–38. External Clock Timing for Divide-by-4 Clock

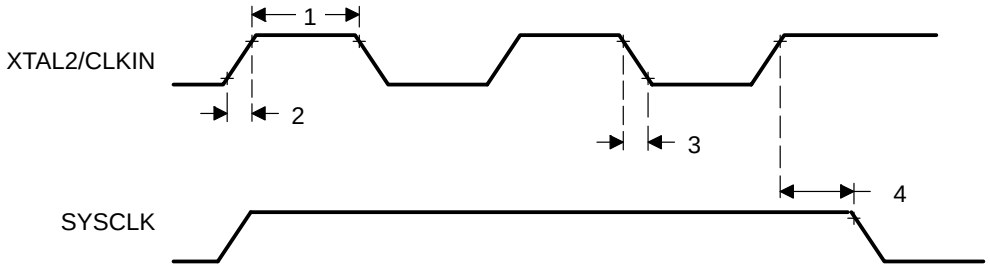


Table 18–59. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–56.

2) This pulse can be either a high pulse, as illustrated in Figure 18–39, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–39. External Clock Timing for Divide-by-1 Clock

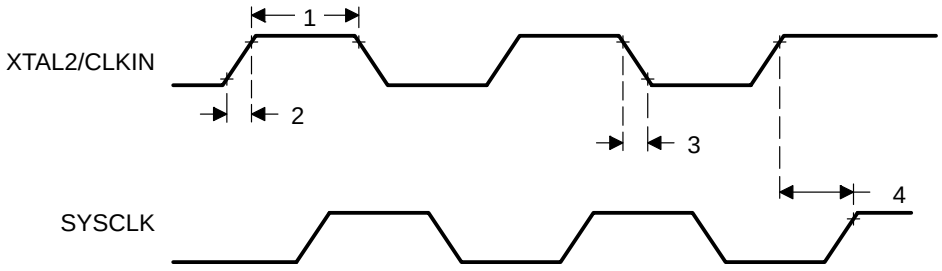
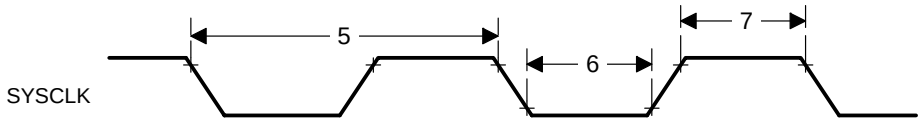


Table 18–60. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system-clock cycle time = 1/SYSCLK

Figure 18–40. SYSCLK Timing



18.18 TMS370CxAxA Specifications

The tables in this section give specifications that apply to the devices in the TMS370CxAxA category. These devices include the TMS370C3A7A.

18.18.1 TMS370CxAxA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–61 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–61. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).

2) $\overline{\text{RESET}}$ must be activated externally when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–62. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			±10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 3)		See Notes 2 and 3 SYSCLK = 5 MHz		30	45	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		20	30	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		7	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		8	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 5)		See Notes 2 and 3 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 1 XTAL2/CLKIN < 0.2 V		2	30	µA

Notes: 1) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.
2) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).
3) Maximum operating current = 7.6(SYSCLK) + 7 mA.
4) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).
5) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.18.2 TMS370CxAxA Timings

Refer to Section 18.1 and Section 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement points.

Table 18–63. External Clocking Requirements for Divide-by-4 Clock (see Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–61.

2) This pulse can be either a high pulse, as illustrated in Figure 18–41, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–41. External Clock Timing for Divide-by-4 Clock

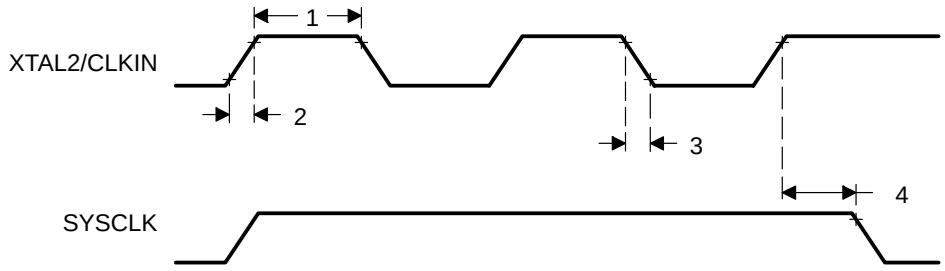


Table 18–64. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–61.

2) This pulse can be either a high pulse, as illustrated in Figure 18–42, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–42. External Clock Timing for Divide-by-1 Clock

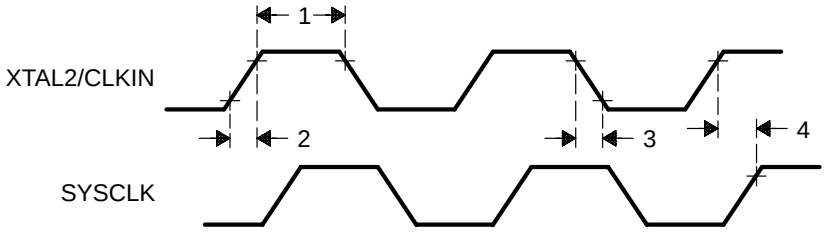
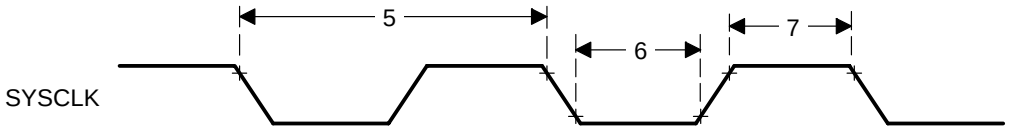


Table 18–65. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system-clock cycle time = 1/SYSCLK

Figure 18–43. SYSCLK Timing



18.19 TMS370CxBxA Specifications

The tables in this section give specifications that apply to the devices in the TMS370CxBxA category. These devices include the TMS370C0B6A.

18.19.1 TMS370CxBxA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–66 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–66. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC1}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{CC2}	Digital I/O supply voltage		4.5	5	5.5	V
V_{CC3}	Analog supply voltage		4.5	5	5.5	V
V_{SS2}	Digital I/O supply ground		– 0.3	0	0.3	V
V_{SS3}	Analog supply ground		– 0.3	0	0.3	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS1}		0.8	V
		MC, normal operation	V_{SS1}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC1}	V
		MC (non-WPO mode)	V_{CC1} – 0.3		V_{CC1} + 0.3	V
		XTAL2/CLKIN	$0.8V_{CC1}$		V_{CC1}	V
		$\overline{\text{RESET}}$	$0.7V_{CC1}$		V_{CC1}	V
V_{MC}	MC (mode control) voltage (see Note 3)	EEPROM write-protect override (WPO) mode	11.7	12	13	V
		Microprocessor mode	V_{CC1} – 0.3		V_{CC1} + 0.3	V
		Microcomputer mode	V_{SS1}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS1}.

2) $\overline{\text{RESET}}$ must be externally activated when V_{CC1} or SYSCLK is out of the recommended operating range.

3) The basic microcomputer and microprocessor operating modes are selected by the voltage level applied to the dedicated MC pin two system clock cycles ($2t_C$) before $\overline{\text{RESET}}$ goes inactive (high). The WPO mode can be selected any time a sufficient voltage is present on MC.

You cannot use the internal connections between pins (for example, the connection between V_{SS1} and V_{SS2}) for a jumper from one side of the chip to the other.

Table 18–67. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC1}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V < $V_I \le$ 0.3 V			10	µA
			0.3 V < $V_I < V_{CC1}$ – 0.3 V			50	µA
			V_{CC1} – 0.3 V ≤ $V_I \le V_{CC1}$ + 0.3 V			10	µA
			V_{CC1} + 0.3 V < $V_I \le$ 13 V			650	µA
		I/O pins	0 V ≤ $V_I \le V_{CC1}$			± 10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		$V_{OH} = 0.9V_{CC1}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode), OSC POWER bit = 0 (see Note 3)		See Notes 1 and 2 SYSCLK = 5 MHz		35	56	mA
			See Notes 1 and 2 SYSCLK = 3 MHz		25	36	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		13	18	mA
I_{CC}	Supply current (STANDBY mode), OSC POWER bit = 0 (see Note 4)		See Notes 1 and 2 SYSCLK = 5 MHz		12	17	mA
			See Notes 1 and 2 SYSCLK = 3 MHz		8	11	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		2.5	3.5	mA
I_{CC}	Supply current (STANDBY mode), OSC POWER bit = 1 (see Note 5)		See Notes 1 and 2 SYSCLK = 3 MHz		6	8.6	mA
			See Notes 1 and 2 SYSCLK = 0.5 MHz		2	3	mA
I_{CC}	Supply current (HALT mode)		See Note 1 XTAL2/CLKIN < 0.2 V		2	30	µA

Notes: 1) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC1} – 0.2 V.

2) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5-MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

3) Maximum operating current = 10(SYSCLK) + 5.8 mA.

4) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).

5) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit =1; valid only up to 3-MHz SYSCLK).

18.19.2 TMS370CxBxA Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2) for timing symbol definitions and parameter measurement information.

Table 18–68. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–66.

2) This pulse can be either a high pulse, as illustrated in Figure 18–44, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–44. External Clock Timing for Divide-by-4 Clock

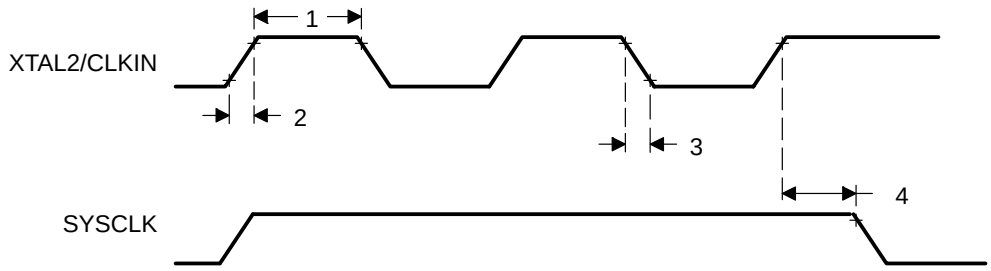


Table 18–69. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–66.

2) This pulse can be either a high pulse, as illustrated in Figure 18–45, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–45. External Clock Timing for Divide-by-1 Clock

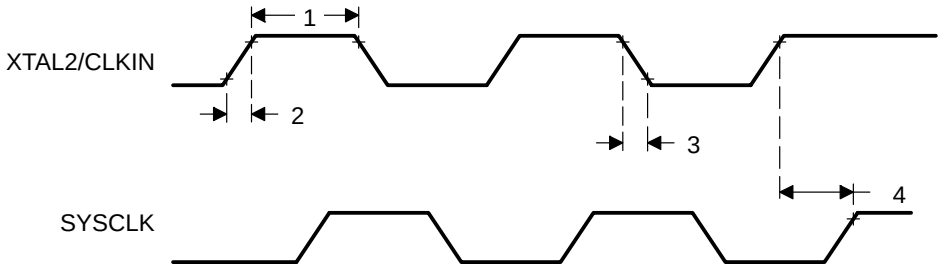
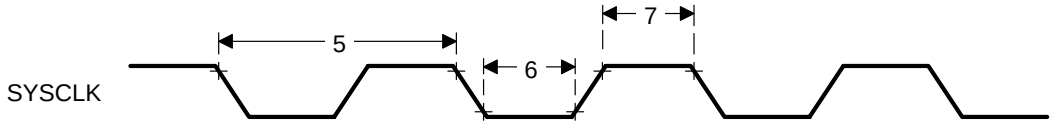


Table 18–70. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 25$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–46. SYSCLK Timing



18.20 TMS370CxCxA Specifications

The tables in this section give specifications that apply to the devices in the TMS370CxCxA category. These devices include the TMS370C3C0A, TMS370C6C2A and SE370C6C2A.

18.20.1 TMS370CxCxA Electrical Specifications

Stresses beyond those listed in Table 18–1 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 in the recommended operating conditions in Table 18–71 is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 18–71. Recommended Operating Conditions (See Note 1)

Parameter			Min	Nom	Max	Unit
V_{CC}	Supply voltage		4.5	5	5.5	V
	RAM data-retention supply voltage (see Note 2)		3		5.5	V
V_{IL}	Low-level input voltage	All pins except MC	V_{SS}		0.8	V
		MC, normal operation	V_{SS}		0.3	V
V_{IH}	High-level input voltage	All pins except MC, XTAL2/CLKIN, and $\overline{\text{RESET}}$	2		V_{CC}	V
		XTAL2/CLKIN	$0.8V_{CC}$		V_{CC}	V
		$\overline{\text{RESET}}$	$0.7V_{CC}$		V_{CC}	V
V_{MC}	MC (mode control) voltage	EPROM programming voltage (V_{PP})	13	13.2	13.5	V
		Microcomputer mode	V_{SS}		0.3	V
T_A	Operating free-air temperature	L version	0		70	°C
		A version	–40		85	°C
		T version	–40		105	°C

Notes: 1) Unless otherwise noted, all voltage values are with respect to V_{SS} (ground).

2) $\overline{\text{RESET}}$ must be activated externally when V_{CC} or SYSCLK is out of the recommended operating range.

Table 18–72. Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter			Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage		I_{OL} = 1.4 mA			0.4	V
V_{OH}	High-level output voltage		I_{OH} = –50 µA	$0.9V_{CC}$			V
			I_{OH} = –2 mA	2.4			V
I_I	Input current	MC	0 V ≤ V_I ≤ 0.3 V			10	µA
			0.3 V < V_I ≤ 13 V			650	µA
			12 V ≤ V_I ≤ 13 V (see Note 1)			50	mA
		I/O pins	0 V ≤ V_I ≤ V_{CC}			± 10	µA
I_{OL}	Low-level output current		V_{OL} = 0.4 V	1.4			mA
I_{OH}	High-level output current		V_{OH} = $0.9V_{CC}$	–50			µA
			V_{OH} = 2.4 V	–2			mA
I_{CC}	Supply current (operating mode) OSC POWER bit = 0 (see Note 4)		See Notes 2 and 3 SYSCLK = 5 MHz		20	36	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		13	25	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		5	11	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 0 (see Note 5)		See Notes 2 and 3 SYSCLK = 5 MHz		10	17	mA
			See Notes 2 and 3 SYSCLK = 3 MHz		6.5	11	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		2	3.5	mA
I_{CC}	Supply current (STANDBY mode) OSC POWER bit = 1 (see Note 6)		See Notes 2 and 3 SYSCLK = 3 MHz		4.5	8.6	mA
			See Notes 2 and 3 SYSCLK = 0.5 MHz		1.5	3.0	mA
I_{CC}	Supply current (HALT mode)		See Note 2 XTAL2/CLKIN < 0.2 V		1	30	µA

Notes: 1) Input current I_{PP} is a maximum of 50 mA only when EPROM is being programmed.

2) Single-chip mode, ports configured as inputs or as outputs with no load. All inputs ≤ 0.2 V or ≥ V_{CC} – 0.2 V.

3) XTAL2/CLKIN is driven with an external square wave signal with 50% duty cycle and rise and fall times less than 10 ns. Current can be higher with a crystal oscillator. At 5 MHz SYSCLK, this extra current = 0.01 mA × (total load capacitance + crystal capacitance in pF).

4) Maximum operating current = 5.6(SYSCLK) + 8 mA.

5) Maximum standby current = 3(SYSCLK) + 2 mA (OSC POWER bit = 0).

6) Maximum standby current = 2.24(SYSCLK) + 1.9 mA (OSC POWER bit = 1; valid only up to 3-MHz SYSCLK).

18.20.2 TMS370CxCxA Timings

Refer to Sections 18.1 and 18.2 (both on page 18-2)) for timing symbol definitions and parameter measurement points.

Table 18–73. External Clocking Requirements for Divide-by-4 Clock (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(Cl)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(Cl)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(Cl)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCL)}$	Delay time, XTAL2/CLKIN rise to SYSCLK fall		100	ns
	CLKIN	Crystal operating frequency	2	20	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	0.5	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–71.

2) This pulse can be either a high pulse, as illustrated in Figure 18–47, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/4

Figure 18–47. External Clock Timing for Divide-by-4 Clock

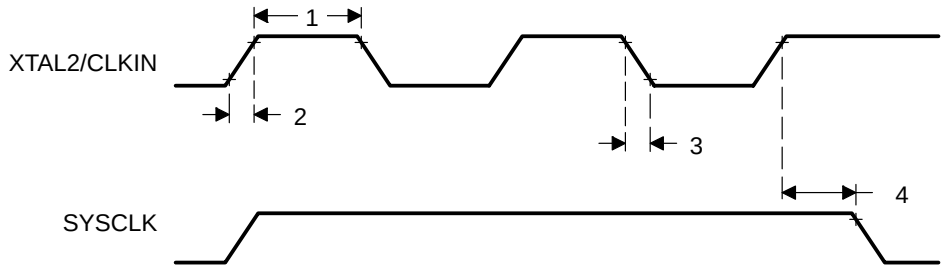


Table 18–74. External Clocking Requirements for Divide-by-1 Clock (PLL) (See Note 1)

No.	Parameter		Min	Max	Unit
1	$t_{w(CI)}$	Pulse duration, XTAL2/CLKIN (see Note 2)	20		ns
2	$t_{r(CI)}$	Rise time, XTAL2/CLKIN		30	ns
3	$t_{f(CI)}$	Fall time, XTAL2/CLKIN		30	ns
4	$t_{d(CIH-SCH)}$	Delay time, XTAL2/CLKIN rise to SYSCLK rise		100	ns
	CLKIN	Crystal operating frequency	2	5	MHz
	SYSCLK	Internal system clock operating frequency (see Note 3)	2	5	MHz

Notes: 1) For V_{IL} and V_{IH}, refer to recommended operating conditions in Table 18–71.

2) This pulse can be either a high pulse, as illustrated in Figure 18–48, which extends from the earliest valid high to the final valid high in an XTAL2/CLKIN cycle, or a low pulse, which extends from the earliest valid low to the final valid low in an XTAL2/CLKIN cycle.

3) SYSCLK = CLKIN/1

Figure 18–48. External Clock Timing for Divide-by-1 Clock

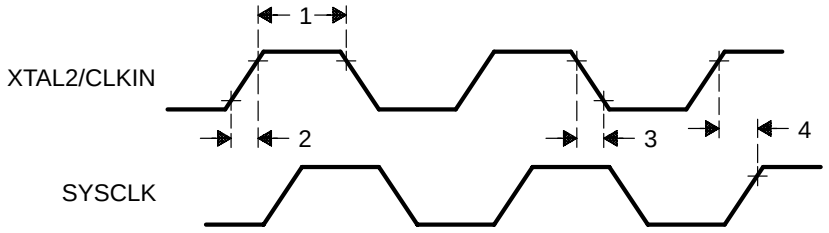
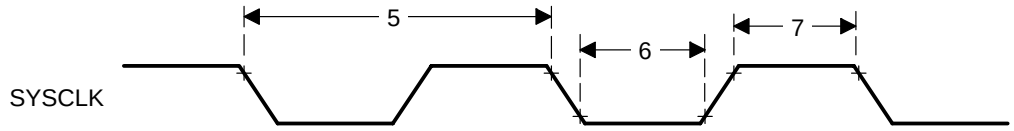


Table 18–75. Switching Characteristics and Timing Requirements (See Note)

No.	Parameter			Min	Max	Unit
5	t_c	Cycle time, SYSCLK	Divide-by-4 clock	200	2000	ns
			Divide-by-1 clock (PLL)	200	500	ns
6	$t_{w(SCL)}$	Pulse duration, SYSCLK low		$0.5t_c - 20$	$0.5t_c$	ns
7	$t_{w(SCH)}$	Pulse duration, SYSCLK high		$0.5t_c$	$0.5t_c + 20$	ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–49. SYSCLK Timing



18.21 SCI Timings

This section contains timing tables and figures for devices that have the serial communications interface (SCI) module.

Note: Parameter Difference

The $t_{d(SCCL–TXDV)}$ parameter differs for TMS370Cxxx devices. Refer to subsection A.9.2 on page A-9.

Table 18–76. SCI Isosynchronous Mode Timing Characteristics and Requirements for Internal Clock (See Note)

No.	Parameter		Min	Max	Unit
24	$t_{c(SCC)}$	Cycle time, SCICLK	$2t_c$	$131\,072t_c$	ns
25	$t_{w(SCCL)}$	Pulse duration, SCICLK low	$t_c - 45$	$0.5t_{c(SCC)} + 45$	ns
26	$t_{w(SCCH)}$	Pulse duration, SCICLK high	$t_c - 45$	$0.5t_{c(SCC)} + 45$	ns
27	$t_{d(SCCL-TXDV)}$	Delay time, SCITXD valid after SCICLK low	–50	60	ns
28	$t_{v(SCCH-TXD)}$	Valid time, SCITXD data valid after SCICLK high	$t_{w(SCCH)} - 50$		ns
29	$t_{su(RXD-SCCH)}$	Setup time, SCIRXD to SCICLK high	$0.25\ t_c + 145$		ns
30	$t_{v(SCCH-RXD)}$	Valid time, SCIRXD data valid after SCICLK high	0		ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–50. SCI Isosynchronous Mode Timing for Internal Clock

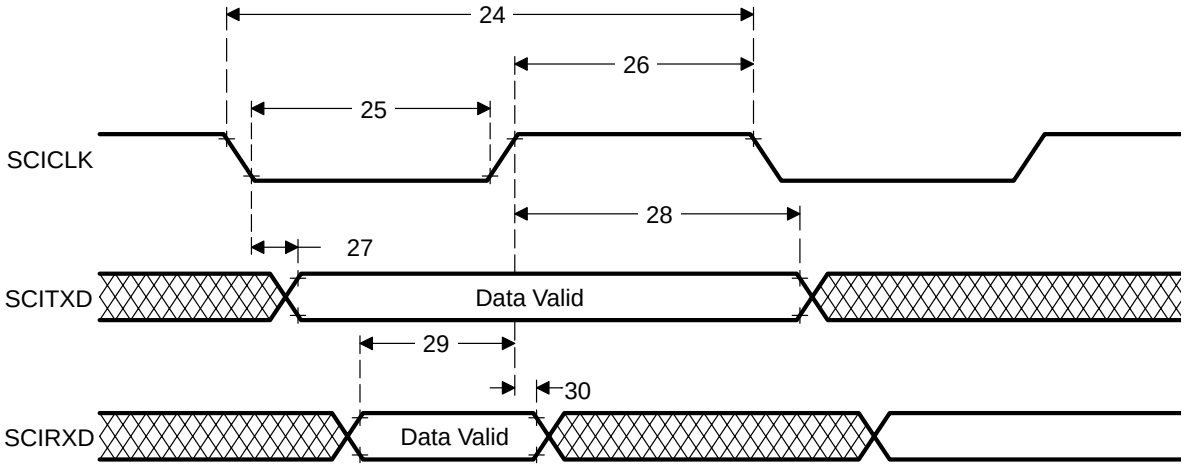
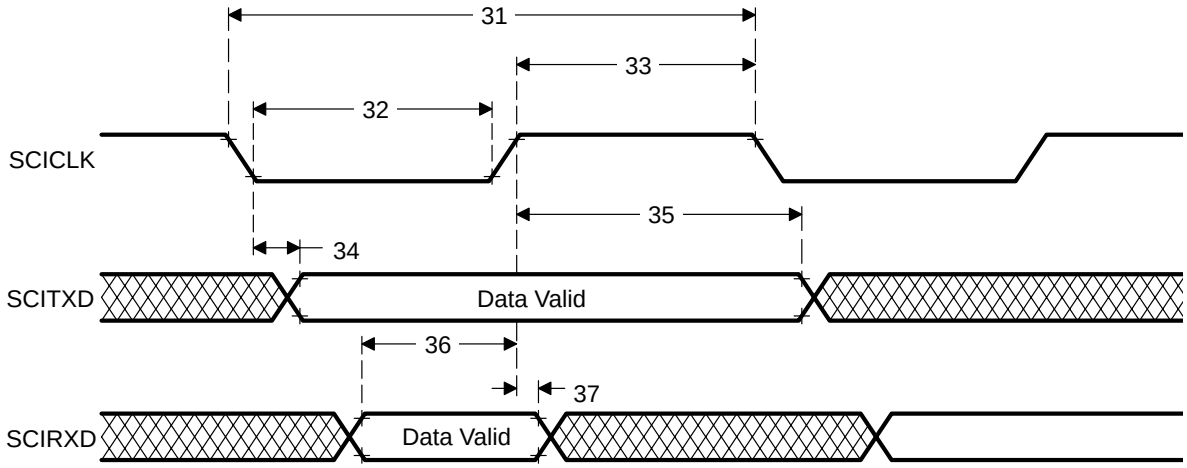


Table 18–77. SCI Isosynchronous Mode Timing Characteristics and Requirements for External Clock (See Note)

No.	Parameter		Min	Max	Unit
31	$t_{c(SCC)}$	Cycle time, SCICLK	$10t_c$		ns
32	$t_{w(SCCL)}$	Pulse duration, SCICLK low	$4.25t_c + 120$		ns
33	$t_{w(SCCH)}$	Pulse duration, SCICLK high	$t_c + 120$		ns
34	$t_{d(SCCL-TXDV)}$	Delay time, SCITXD valid after SCICLK low		$4.25t_c + 145$	ns
35	$t_{v(SCCH-TXD)}$	Valid time, SCITXD data valid after SCICLK high	$t_{w(SCCH)}$		ns
36	$t_{su(RXD-SCCH)}$	Setup time, SCIRXD to SCICLK high	40		ns
37	$t_{v(SCCH-RXD)}$	Valid time, SCIRXD data after SCICLK high	$2t_C$		ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–51. SCI Isosynchronous Mode Timing for External Clock



18.22 SPI Timings

This section contains timing tables and diagrams for the devices that have the serial peripheral interface (SPI) module in the following categories: TMS370Cx1x, TMS370Cx2x, TMS370Cx36, TMS370Cx5x, and TMS370Cx6x.

Note: Electrical/Timing Specification Differences

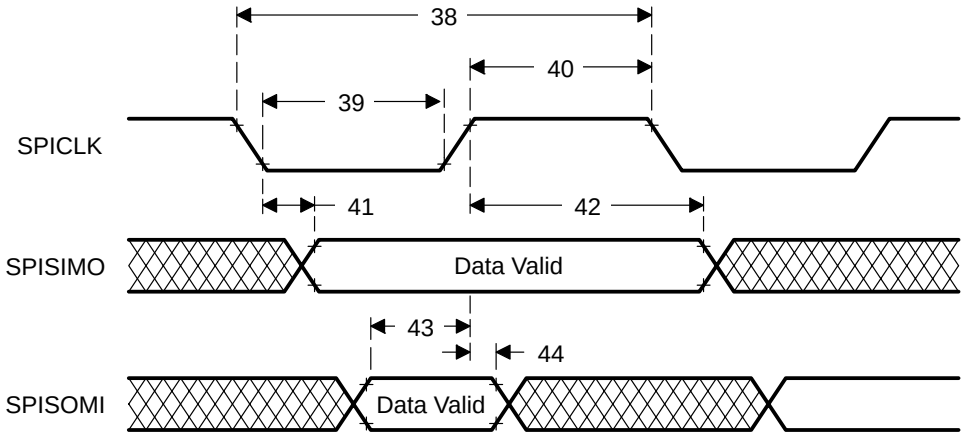
Some SPI electrical specifications and timings differ for TMS370Cxxx devices. Refer to subsection A.9.2 on page A-9.

Table 18–78. SPI Master Mode External Timing Characteristics and Requirements (See Note)

No.	Parameter		Min	Max	Unit
38	$t_{c(SPC)M}$	Cycle time, SPICLK	$2t_c$	$256t_c$	ns
39	$t_{w(SPCL)M}$	Pulse duration, SPICLK low	$t_c - 45$	$0.5t_{c(SPC)} + 45$	ns
40	$t_{w(SPCH)M}$	Pulse duration, SPICLK high	$t_c - 55$	$0.5t_{c(SPC)} + 45$	ns
41	$t_{d(SPCL-SIMOV)M}$	Delay time, SPISIMO valid after SPICLK low (polarity = 1)	– 65	50	ns
42	$t_{v(SPCH-SIMO)M}$	Valid time, SPISIMO after SPICLK high (polarity =1)	$t_{w(SPCH)} - 50$		ns
43	$t_{su(SOMI-SPCH)M}$	Setup time, SPISOMI to SPICLK high (polarity = 1)	$0.25\ t_c + 150$		ns
44	$t_{v(SPCH-SOMI)M}$	Valid time, SPISOMI after SPICLK high (polarity = 1)	0		ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–52. SPI Master Mode External Clock Timing



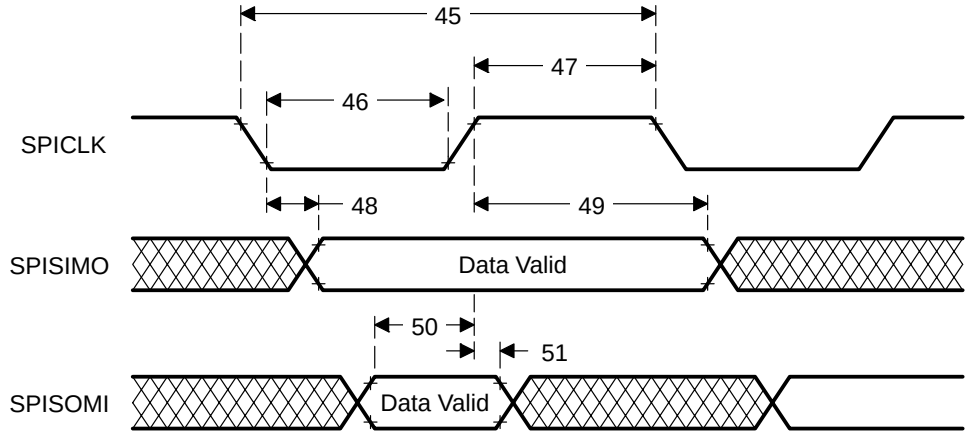
Note: In this figure, polarity = 1. SPICLK is inverted when polarity = 0.

Table 18–79. SPI Slave Mode External Timing Characteristics and Requirements (See Note)

No.			Min	Max	Unit
45	$t_{c(SPC)S}$	Cycle time, SPICLK	$8t_c$		ns
46	$t_{w(SPCL)S}$	Pulse duration, SPICLK low	$4t_c - 45$	$0.5t_{c(SPC)S} + 45$	ns
47	$t_{w(SPCH)S}$	Pulse duration, SPICLK high	$4t_c - 45$	$0.5t_{c(SPC)S} + 45$	ns
48	$t_{d(SPCL-SOMIV)S}$	Delay time, SPISOMI valid after SPICLK low (polarity = 1)		$3.25t_c + 130$	ns
49	$t_{v(SPCH-SOMI)S}$	Valid time, SPISOMI after SPICLK high (polarity =1)	$t_{w(SPCH)S}$		ns
50	$t_{su(SIMO-SPCH)S}$	Setup time, SPISIMO to SPICLK high (polarity = 1)	0		ns
51	$t_{v(SPCH-SIMO)S}$	Valid time, SPISIMO after SPICLK high (polarity = 1)	$3t_c + 100$		ns

Note: t_c = system clock cycle time = 1/SYSCLK

Figure 18–53. SPI Mode Slave External Timing



Notes: 1) In this figure, polarity = 1. SPICLK is inverted when polarity = 0.

2) As a slave, the SPICLK pin is used as the input for the serial clock, which is supplied from the network master.

18.23 Analog-to-Digital Converter 1 (ADC1) Module Specifications

This section contains specifications for the devices that have the analog-to-digital converter 1 (ADC1) module in the following categories: TMS370Cx32, TMS370Cx36, TMS370Cx4x, TMS370Cx5x, TMS370Cx6x, TMS370Cx7x and TMS370CxBx.

The ADC1 module has a separate power bus for its analog circuitry. These pins are referred to as V_{CC3} and V_{SS3}. The purpose is to enhance ADC1 performance by preventing digital switching noise of the logic circuitry that may be present on V_{SS} and V_{CC} from coupling into the ADC1 analog stage. All ADC1 specifications are given with respect to V_{SS3} unless otherwise noted.

Resolution . . . . . 8-bits (256 values)
Monotonic . . . . . Yes
Output conversion code . . . . . 00h to FFh
(00h for $V_I \leq V_{SS3}$; FFh for $V_I \geq V_{ref}$)
Conversion time (excluding sample time) . . . . . 164 t_c
(where t_c = system clock cycle time)

Table 18–80. Recommended Operating Conditions

Parameter		Min	Nom	Max	Unit
V_{CC3}	Analog supply voltage	4.5	5	5.5	V
		V_{CC} – 0.3		V_{CC} + 0.3	V
V_{SS3}	Analog ground	V_{SS} – 0.3		V_{SS} + 0.3	V
V_{ref}	Non-V_{CC3} reference (see Note)	2.5	V_{CC3}	V_{CC3} + 0.1	V
	Analog input for conversion	V_{SS3}		V_{ref}	V

Note: V_{ref} must be stable, within ± 1/2 LSB of the required resolution, during the entire conversion time. $V_{CC} = V_{CC1}$ and $V_{SS} = V_{SS1}$ for ’x32, ’x36, ’x5x, ’x6x, ’x7x, and ’xBx devices.

Table 18–81. ADC1 Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter		Test Conditions	Min	Max	Unit
	Absolute accuracy (see Note 1)	V_{CC3} = 5.5 V, V_{ref} = 5.1 V		±1.5	LSB
	Differential/integral linearity error (see Notes 1 and 2)	V_{CC3} = 5.5 V, V_{ref} = 5.1 V		±0.9	LSB
I_{CC3}	Analog supply current	Converting		2	mA
		Nonconverting		5	µA
I_I	Input current, AN0–AN7	0 V ≤ V_I ≤ 5.5 V		2	µA
I_{ref}	input charge current			1	mA
Z_{ref}	Source impedance of V_{ref}	SYSCLK ≤ 3 MHz		24	kΩ
		3 MHz < SYSCLK ≤ 5 MHz		10	kΩ

Notes: 1) Absolute resolution = 20 mV. At V_{ref} = 5 V, this is one LSB. As V_{ref} decreases, LSB size decreases. Therefore, the absolute accuracy and differential/integral linearity errors in terms of LSBs increase.

2) Excluding quantization error of 1/2 LSB

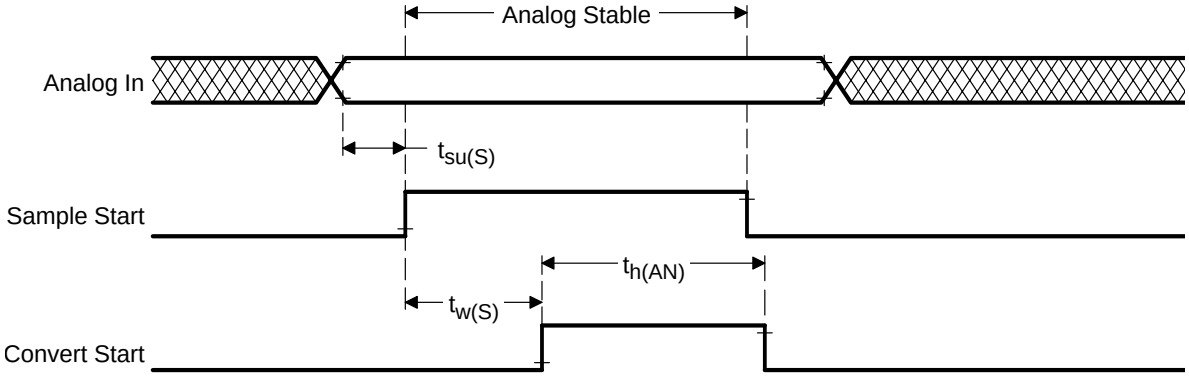
The ADC1 module allows complete freedom in design of the sources for the analog inputs. The period of the sample time is user-defined so that high-impedance sources can be accommodated without penalty to low-impedance sources. The sample period begins when the SAMPLE START bit of the ADC1 control register (ADCTL.6) is set to 1. The end of the signal sample period occurs when the conversion bit (CONVERT START, ADCTL.7) is set to 1. After a hold time, the converter resets the SAMPLE START and CONVERT START bits, signaling that a conversion has started and that the analog signal can be removed.

Table 18–82. Analog Timing Requirements

Parameter		Min	Max	Unit
$t_{su(S)}$	Setup time, analog to sample command	0		ns
$t_{h(AN)}$	Hold time, analog input from start of conversion	$18t_c$		ns
$t_{w(S)}$	Pulse duration, sample time per kilohm of source impedance (see Note)	1		µs/kΩ

Note: The value given is valid for a signal with a source impedance > 1 kΩ. If the source impedance is < 1 kΩ, use a minimum sampling time of 1 µs.

Figure 18–54. Analog Timing



18.24 Analog-to-Digital Converter 2 (ADC2) Module Specifications

This section contains specifications for the TMS370CxCx device category, which has the analog-to-digital converter 2 (ADC2) module. ADC2 shares the V_{CC} power bus for its analog and digital circuitry. All ADC2 specifications are given with respect to V_{SS} unless otherwise noted.

Resolution . . . . . 8-bits (256 values)
Monotonic . . . . . Yes
Output conversion code . . . . . 00h to FFh
(00h for $V_I \le V_{SS} \le$; FFh for $V_I \le V_{ref}$)
Conversion time (excluding sample time) . . . . . 164 t_c
(where t_c = system clock cycle time)

Table 18–83. Recommended Operating Conditions

Parameter		Min	Nom	Max	Unit
V_{CC}	Analog supply voltage	4.5	5	5.5	V
V_{ref}	Non-V_{CC} reference (see Note)	2.5	V_{CC}	V_{CC} + 0.1	V
	Analog input for conversion	V_{SS}		V_{ref}	V

Note: V_{ref} must be stable, within ±1/2 LSB of the required resolution during the entire conversion time.

Table 18–84. ADC2 Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter		Test Conditions	Min	Max	Unit
	Absolute accuracy (see Note 1)	V_{CC} = 5.5 V, V_{ref} = 5.1 V		+1.5	LSB
	Differential/integral linearity error (see Notes 1 and 2)	V_{CC} = 5.5 V, V_{ref} = 5.1 V		±0.9	LSB
I_{CC}	Analog supply current	Converting		2	mA
		Nonconverting		5	µA
I_I	Input current, AN0-AN3	0 V ≤ V_I ≤ 5.5 V		2	µA
I_{ref}	Input charge current			1	mA
Z_{ref}	Source impedance of V_{ref}	SYSCLK ≤ 3 MHz		24	kΩ
		3 MHz < SYSCLK ≤ 5 MHz		10	kΩ

Notes: 1) Absolute resolution = 20 mV. At V_{ref} = 5 V, this is 1 LSB. As V_{ref} decreases, LSB size decreases and thus absolute accuracy and differential / integral linearity errors in terms of LSBs increases.
2) Excluding quantization error of 1/2 LSB

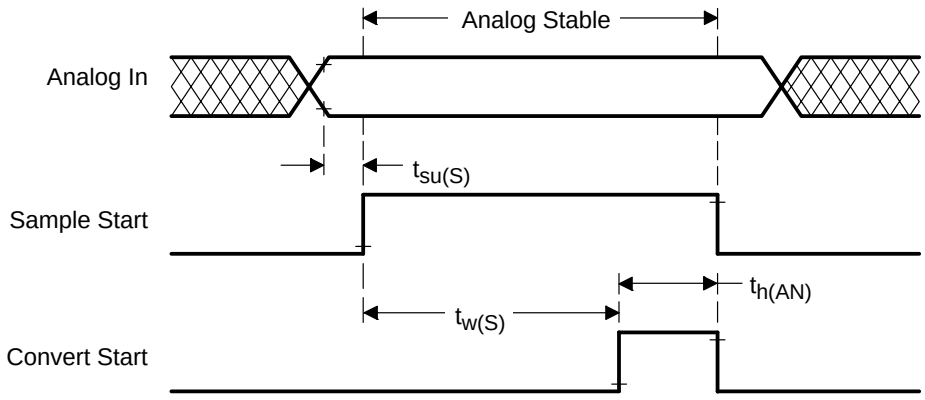
The ADC2 module allows complete freedom in design of the sources for the analog inputs. The period of the sample time is user-defined so that high-impedance sources can be accommodated without penalty to low-impedance sources. The sample period begins when the SAMPLE START bit of the ADC2 control register (ADCTL.6) is set to 1. The end of the signal sample period occurs when the conversion bit (CONVERT START, ADCTL.7) is set to 1. After a hold time, the converter resets the SAMPLE START and CONVERT START bits, signaling that a conversion has started and the analog signal can be removed.

Table 18–85. Analog Timing Requirements

Parameter		Min	Max	Unit
$t_{su(S)}$	Setup time, analog input to sample command	0		ns
$t_{h(AN)}$	Hold time, analog input from start of conversion	$18t_c$		ns
$t_{w(S)}$	Pulse duration, sample time per kilohm of source impedance (see Note)	1		μs/kΩ

Note: The value given is valid for a signal with a source impedance > 1 kΩ. If the source impedance is < 1 kΩ, use a minimum sampling time of 1 μs.

Figure 18–55. Analog Timing



18.25 Analog-to-Digital Converter 3 (ADC3) Module Specifications

This section contains specifications for the TMS370Cx9x device category, which has the analog-to-digital converter 3 (ADC3) module.

The ADC3 module has a separate power bus for its analog circuitry. These pins are referred to as V_{CC3} and V_{SS3}. The purpose is to enhance ADC3 performance by preventing digital switching noise of the logic circuitry that may be present on V_{SS} and V_{CC} from coupling into the ADC3 analog stage. All ADC3 specifications are given with respect to V_{SS3} unless otherwise noted.

Resolution . . . . . 8-bits (256 values)
Monotonic . . . . . Yes
Output conversion code . . . . . 00h to FFh
(00h for $V_I \leq V_{SS3} \leq$; FFh for $V_I \leq V_{ref)}$
Conversion time (excluding sample time) . . . . . 164 t_c
(where t_c = system clock cycle time)

Table 18–86. Recommended Operating Conditions

Parameter		Min	Nom	Max	Unit
V_{CC3}	Analog supply voltage	4.5	5	5.5	V
		V_{CC} – 0.3		V_{CC} + 0.3	V
V_{SS3}	Analog input ground	V_{SS} – 0.3		V_{SS} + 0.3	V
V_{ref}	Non-V_{CC3} reference (see Note)	2.5	V_{CC3}	V_{CC3} + 0.1	V
	Analog input for conversion	V_{SS3}		V_{ref}	V

Note: V_{ref} must be stable, within ±1/2 LSB of the required resolution during the entire conversion time.

Table 18–87. ADC3 Electrical Characteristics Over Recommended Operating Free-Air Temperature Range

Parameter		Test Conditions	Min	Max	Unit
	Absolute accuracy (see Note 1)	V_{CC3} = 5.5 V, V_{ref} = 5.1 V		±1.5	LSB
	Differential/integral linearity error (see Notes 1 and 2)	V_{CC3} = 5.5 V , V_{ref} = 5.1 V		±0.9	LSB
I_{CC3}	Analog supply current	Converting		2	mA
		Nonconverting		5	μA
I_I	Input current, AN0–AN14	0 V ≤ V_I ≤ 5.5 V		2	μA
I_{ref}	Input charge current			1	mA
Z_{ref}	Source impedance of V_{ref}	SYSCLK ≤ 3 MHz		24	kΩ
		3 MHz < SYSCLK ≤ 5 MHz		10	kΩ

Notes: 1) Absolute resolution = 20 mV. At V_{ref} = 5 V, this is 1 LSB. As V_{ref} decreases, LSB size decreases; therefore, the absolute accuracy and differential/integral linearity errors in terms of LSBs increase.

2) Excluding quantization error of 1/2 LSB

The ADC3 module allows complete freedom in design of the sources for the analog inputs. The period of the sample time is user-defined so that high-impedance sources can be accommodated without penalty to low-impedance sources. The sample period begins when the SAMPLE START bit of the ADC3 control register (ADCTL.6) is set to 1. The end of the signal sample period occurs when the conversion bit (CONVERT START, ADCTL.7) is set to 1. After a hold time, the converter resets the SAMPLE START and CONVERT START bits, signaling that a conversion has started and the analog signal can be removed.

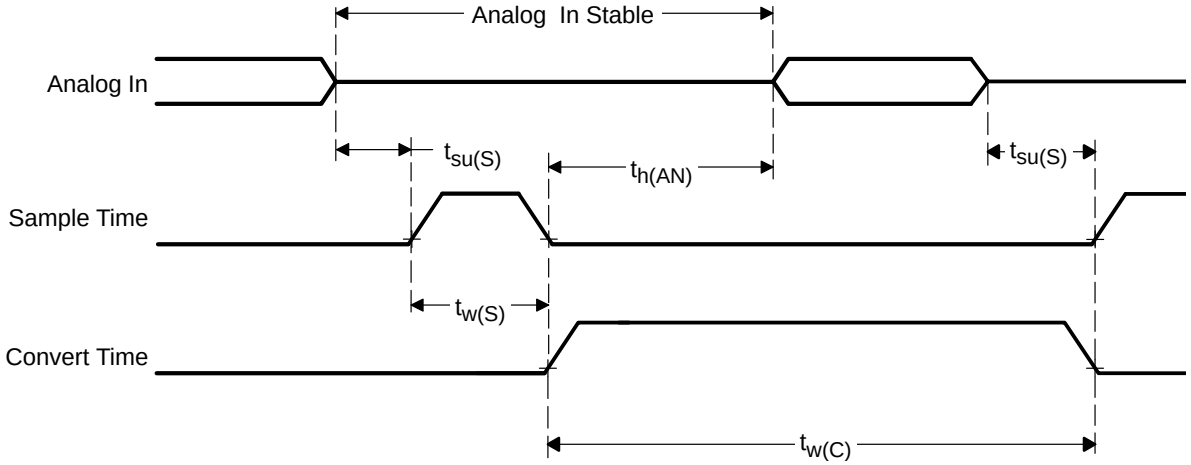
Table 18–88. Analog Timing Requirements

Parameter		Min	Max	Unit
$t_{su(S)}$	Setup time, analog to sample command	0		ns
$t_{h(AN)}$	Hold time, analog input from start of conversion (see Note 2)	$(N+2)t_c$		ns
$t_{w(C)}$	Conversion time (see Note 2)	$(10N + 4)t_c$		ns
$t_{w(S)}$	Pulse duration, sample time per kilohm of source impedance (see Note 1)	1		µs/kΩ

Notes: 1) The value given is valid for a signal with a source impedance > 1 kΩ. If the source impedance is < 1 kΩ, use a minimum sampling time of 1µs.

2) N = 16, 8, 4, or 2 upon selected conversion rate.

Figure 18–56. Analog Timing



Chapter 19

Customer Information

This chapter includes general information on mask-ROM prototyping, TMS370 physical characteristics, and parts ordering.

Topic **Page**

19.1 Mask-ROM Prototype and Production Flow

The TMS370 family includes many mask-ROM microcontrollers. The ROM is manufactured containing your application code. The custom-programmed nature of these devices requires a standard, defined interface between you and the factory during production. The prototype and production flow is described in the following steps.

1) **Customer-required information.** For TI to accept the receipt of your ROM algorithm, you must follow these steps:

   a) Complete and submit a **new code release form** (NCRF—available from a TI Field Sales Office and shown in Example 19–1, page 19-5) describing the custom features of the device (for example, customer information, prototype and production quantities and dates, any exceptions to standard electrical specifications, part numbers and symbols, package type, etc.).

   b) If you requested nonstandard specifications on the NCRF, submit a copy of the description of the microcontroller that includes the functional description and electrical specifications (including absolute maximum ratings, recommended operating conditions, and timing values). TI will then respond to the requested specification changes.

   c) When you have developed and verified your code with the development system, submit the object file in one of the following formats: Intel Hex, TI-Tagged, or COFF (generated by TMS370 development system). Acceptable media include the following:

      - Modem transfer: PC-to-PC via Xmodem, Ymodem, or Zmodem protocol or Microstuf's CROSSTALK XVI protocol
      - DOS-formatted 3-1/2-inch high-density disk
      - EPROM devices (currently supported: TMS27C128, TMS27C256, and TMS27C512)
      - TMS370 EPROM devices (OTP and reprogrammable—refer to Table 17–1 on page 17-27)

   Send the completed NCRF, purchase order, customer specification (if required), and ROM code to the local representative or to the nearest field sales office.

2) **TI performs ROM receipt.** Code review and ROM receipt is performed on your code, and a unique manufacturing ROM code number (such as R1501234FN) is assigned to your algorithm. All future correspondence should indicate this number. During the ROM receipt procedure, TI:

   ❑ Reads the ROM code information.

- ❑ Processes the ROM code information, including writing identification information into the reserved locations.
- ❑ Reproduces your ROM object code on the media that you requested on the NCRF. (Note: You must provide the EPROM device if that is the type of media that you have requested on the NCRF.)
- ❑ Returns the processed code and the original code for your verification.

Notes:

- ❑ All TMS370 family devices contain mask-ROM space reserved for TI use only. This space includes addresses 7FE0h through 7FEBh. This reserved area should not be used in your software algorithm, nor should it be used during mask-ROM/firmware development. **The reserved location contents are changed only by TI.**
- ❑ Data EEPROM locations are not programmed in the standard production flow.

3) **Customer ROM receipt approval.** Verify that the ROM code received and processed by TI is correct and that no information was misinterpreted in the transfer. Next, return an algorithm approval form (available from the field sales office) to verify the correct ROM receipt or resubmit the code for processing. This written verification constitutes the contractual agreement for creating the custom mask and manufacturing the ROM verification prototype units.

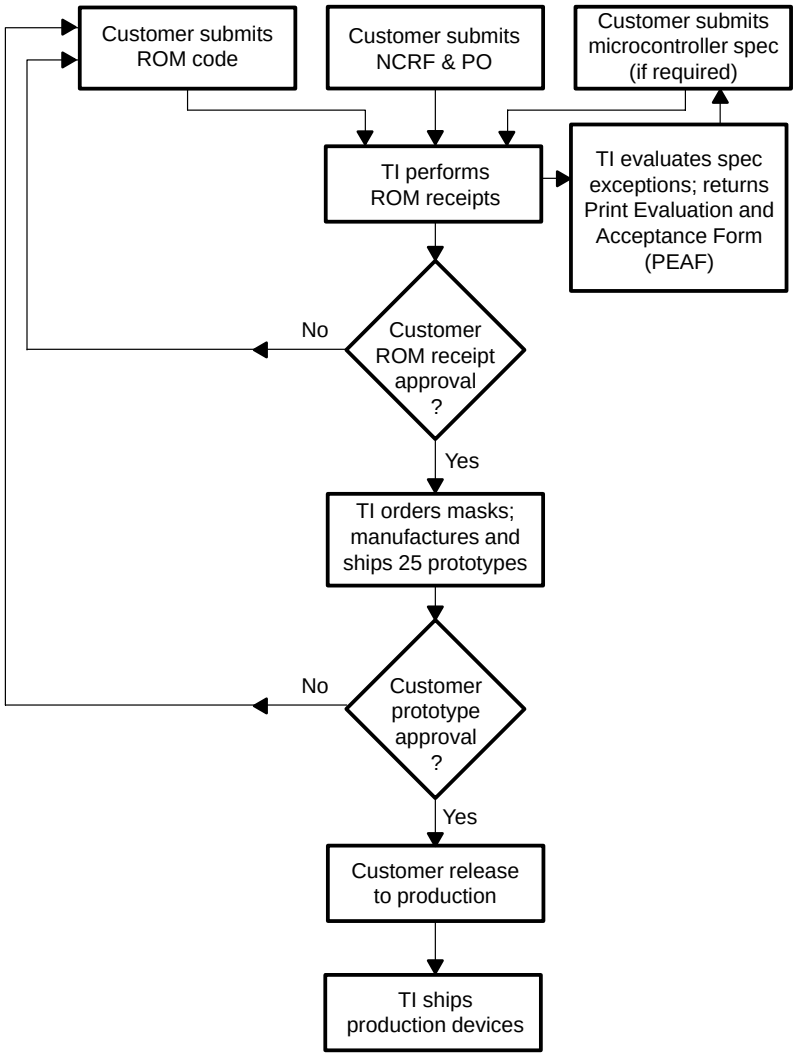
4) **TI orders masks; manufactures and ships 25 prototypes.** TI generates the prototype photomasks and processes, manufactures, and tests 25 microcontroller prototypes containing your ROM pattern. TI then ships the prototypes to you for ROM code verification; these microcontroller devices have been made using the custom mask but are for the purposes of ROM verification only.

5) **Customer prototype approval.** Verify the operation of these prototypes in the system and respond with written approval or disapproval. Written approval constitutes the contractual agreement to initiate volume microcontroller production using the verified prototype ROM code.

6) **Customer release to production.** With your algorithm approval, the ROM code is released to production. TI begins shipping production devices (with production lead time) according to your final specification and order requirements.

Figure 19–1 illustrates the standard of prototype/production.

Figure 19–1. Prototype and Production Flow



Two lead times are quoted in reference to the preceding flow. For the latest lead times, contact the nearest TI field sales office.

- **Prototype lead time** is the elapsed time from the receipt of written ROM receipt verification to the delivery of 25 prototype devices.
- **Production lead time** is the elapsed time from the receipt of written customer prototype approval to delivery of production devices.

Example 19–1 shows a sample new code release form.

Example 19–1. New Code Release Form

NEW CODE RELEASE FORM
TEXAS INSTRUMENTS
TMS370 MICROCONTROLLER PRODUCTS

DATE: _____

To release a new customer algorithm to TI incorporated into a TMS370 family microcontroller, complete this form and submit with the following information:

1. A ROM description in object form on Floppy Disk, Modem XFR, or EPROM (Verification file will be returned via same media)
2. An attached specification if not using TI standard specification as incorporated in TI's applicable device data book.

Company Name: _____
Street Address: _____
Street Address: _____
City: _____ State _____ Zip _____

Contact Mr./Ms.: _____
Phone: (_____)_____ Ext.: _____

Customer Purchase Order Number: _____

Customer Part Number: _____
Customer Application: _____

Customer Print Number *Yes:____ # _____
No: ___ (Std. spec to be followed)
*If Yes: Customer must provide "print" to TI w/NCRF for approval before ROM code processing starts.

TMS370 Device: _____

TI Customer ROM Number: _____
(provided by Texas Instruments)

OSCILLATOR FREQUENCY

	MIN	TYP	MAX
[] External Drive (CLKIN)	_____	_____	_____
[] Crystal	_____	_____	_____
[] Ceramic Resonator	_____	_____	_____

[] Supply Voltage MIN: _____ MAX: _____
(std range: 4.5V to 5.5V)

CONTACT OPTIONS FOR THE 'A' VERSION TMS370 MICROCONTROLLERS

Low Power Modes	Watchdog counter	Clock Type
[] Enabled	[] Standard	[] Standard (/4)
[] Disabled	[] Hard Enabled	[] PLL (/1)
	[] Simple Counter	

NOTE:
Non 'A' version ROM devices of the TMS370 microcontrollers will have the "Low power modes Enabled", "Divide-by-4" Clock, and "Standard" Watchdog options. See the *TMS370 Family User's Guide* (literature number SPNU127) or the *TMS370 Family Data Manual* (literature number SPNS014B).

TEMPERATURE RANGE

[] 'L': 0°C to 70°C (standard)
[] 'A': –40°C to 85°C
[] 'T': –40°C to 105°C

PACKAGE TYPE

[] 'N' 28-pin PDIP
[] "FN" 28-pin PLCC
[] "N" 40-pin PDIP
[] "NJ" 40-pin PSDIP (formerly known as N2)
[] "FN" 44-pin PLCC
[] "FN" 68-pin PLCC
[] "NM" 64-pin PSDIP

SYMBOLIZATION

[] TI standard symbolization
[] TI standard w/customer part number
[] Customer symbolization
(per attached spec, subject to approval)

BUS EXPANSION

[] YES [] NO

NON-STANDARD SPECIFICATIONS:
ALL NON-STANDARDS SPECIFICATIONS MUST BE APPROVED BY THE TI ENGINEERING STAFF: If the customer requires expedited production material (i.e., product which must be started in process prior to prototype approval and full production release) and non-standard spec issues are not resolved to the satisfaction of both the customer and TI in time for a scheduled shipment, the specification parameters in question will be processed/tested to the standard TI spec. Any such devices which are shipped without conformance to a mutually approved spec, will be identified by a 'P' in the symbolization preceding the TI part number.

RELEASE AUTHORIZATION:
This document, including any referenced attachments, is and will be the controlling document for all orders placed for this TI custom device. Any changes must be in writing and mutually agreed to by both the customer and TI. The prototype cycletime commences when this document is signed off and the verification code is approved by the customer.

1. Customer: _____ Date: _____

2. TI: Field Sales: _____
Marketing: _____
Prod. Eng.: _____
Proto. Release: _____

19.2 Mechanical Package Information

The TMS370 microcontroller family devices are assembled in six package types according to the type of material and outline used for the package. These package types are:

- Plastic dual-in-line package (PDIP)
- Plastic shrink dual-in-line package (PSDIP)
- Plastic leaded chip carrier (PLCC)
- Ceramic dual-in-line package (CDIP)
- Ceramic shrink dual-in-line package (CSDIP)
- Ceramic leaded chip carrier (CLCC)

Package types are designated by the suffix on the ROM code number for devices manufactured with your ROM code (for example, R1501234FN) and by the suffix of the standard device number for devices with EPROM. Table 19–1 indicates the package type, suffix indicator, and family members supported on that package type.

The various package types are shown in Figure 19–2 through Figure 19–9 as listed below:

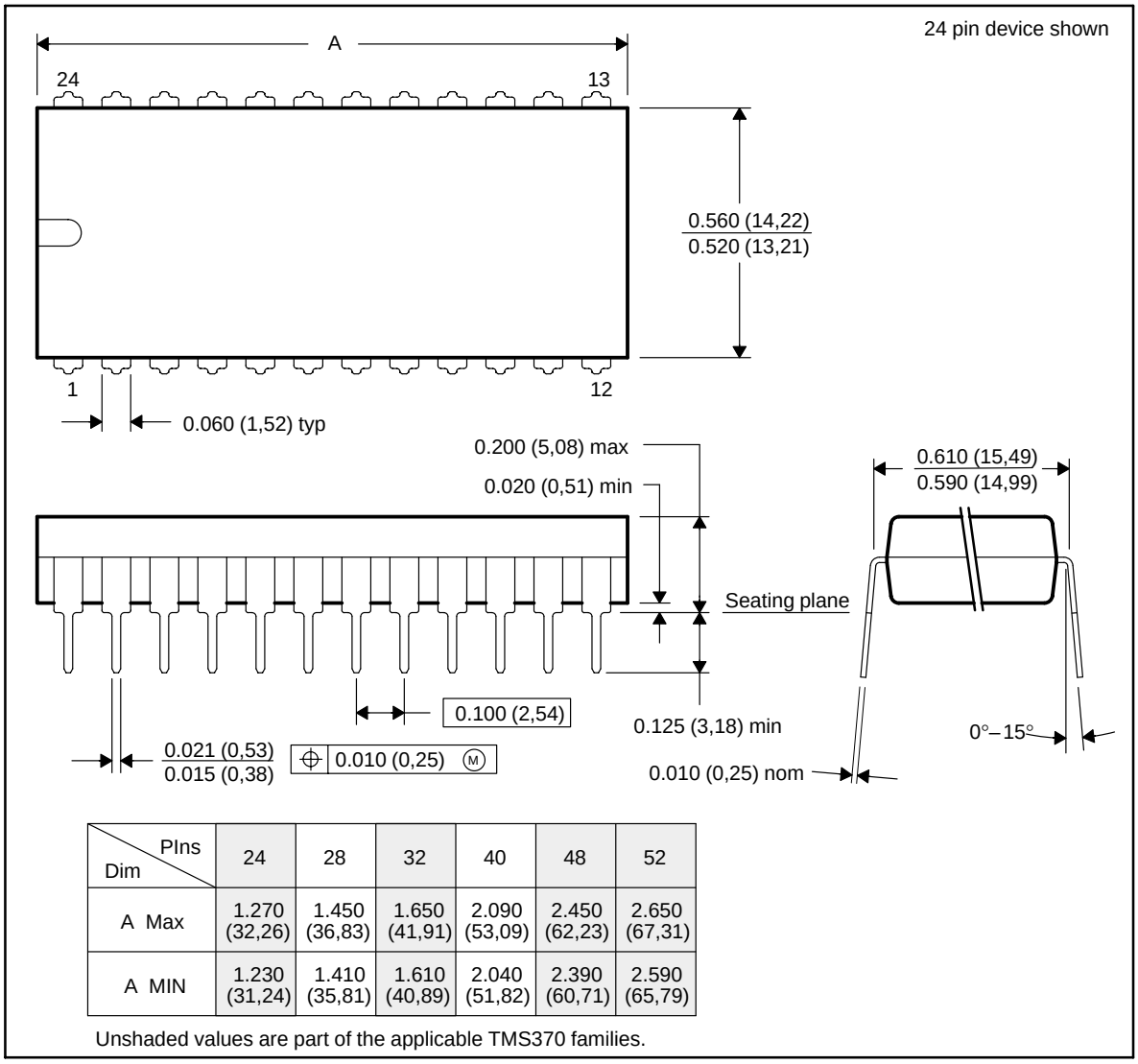
Table 19–1. Package Types and Associated Family Members

Package Type	Suffix Indicator	Family Members
28-pin plastic DIP (100-mil pin spacing)	N	TMS370Cx1x and TMS370CxCx
28-pin ceramic DIP (100-mil pin spacing)	JD	SE370C712 and SE370C6C2
28-pin PLCC (50-mil pin spacing)	FN	TMS370Cx0x, TMS370Cx1x, and TMS370CxCx
28-pin CLCC (50-mil pin spacing)	FZ	SE370C702, SE370C712, and SE370C6C2
40-pin plastic DIP (100-mil pin spacing)	N	TMS370Cx2x, TMS370Cx4x, TMS370C080, and TMS370C3A7
40-pin plastic shrink DIP (70-mil pin spacing)	NJ†	TMS370Cx2x, TMS370Cx4x, and TMS370Cx9x
40-pin ceramic DIP (100-mil pin spacing)	JD	SE370C722 and SE370C742
40-pin ceramic shrink DIP (70-mil pin spacing)	JC	SE370C722, SE370C742, and SE370C792
44-pin PLCC (50-mil pin spacing)	FN	TMS370Cx2x, TMS370Cx3x, TMS370Cx4x, TMS370Cx8x, and TMS370Cx9x
44-pin CLCC (50-mil pin spacing)	FZ	SE370C722, SE370C732, SE370C736, SE370C742, SE370C686, and SE370C792
64-pin plastic shrink DIP (70-mil pin spacing)	NM	TMS370Cx5x, TMS370Cx7x, and TMS370C0B6
64-pin ceramic shrink DIP (70-mil pin spacing)	JN	SE370C756, SE370C758, SE370C759, and SE370C777
68-pin PLCC (50-mil pin spacing)	FN	TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, and TMS370C0B6
68-pin CLCC (50-mil pin spacing)	FZ	SE370C756, SE370C758, SE370C759, SE370C768, SE370C769, and SE370C777

† The NJ designator for the 40-pin plastic shrink DIP package was formerly known as N2. The mechanical drawing of the NJ is identical to the N2 package and did not need to be requalified.

Figure 19–2. Plastic Dual-In-Line Package (N Suffix)

N (R-PDIP-T**) Plastic Dual-In-Line Package



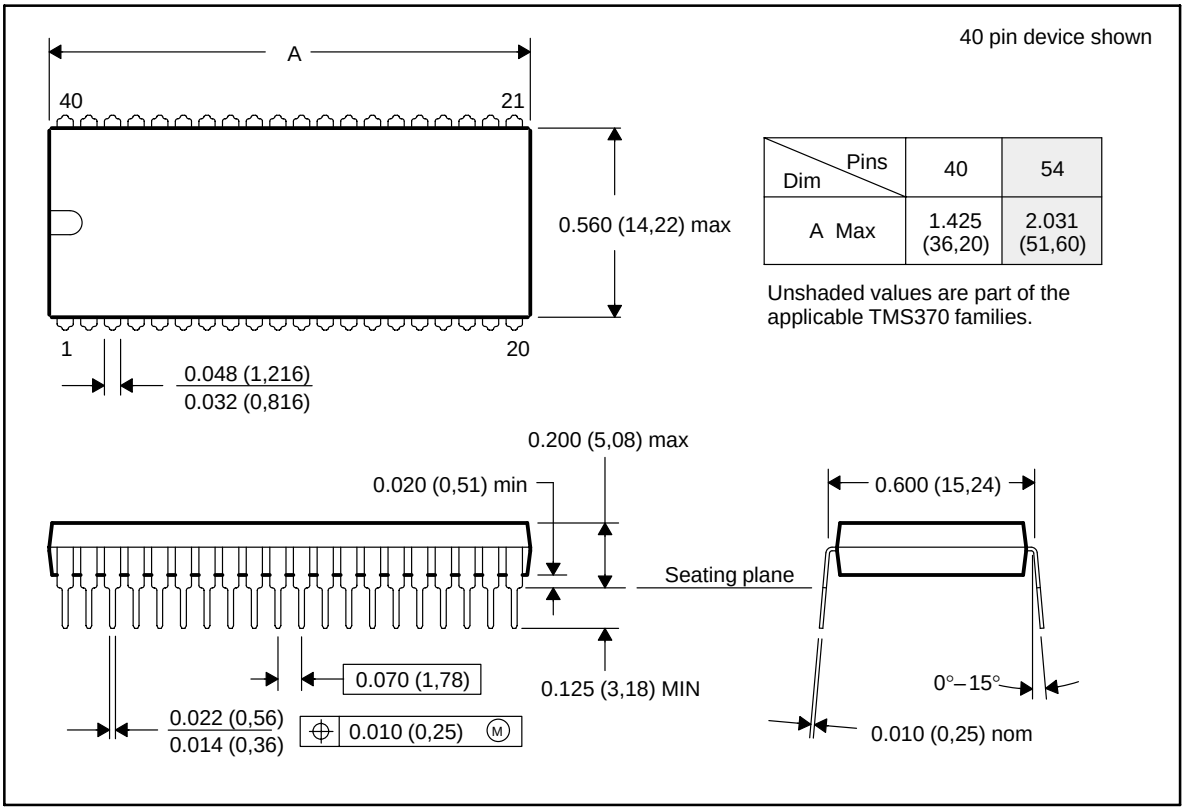
Dim \ PIns	24	28	32	40	48	52
A Max	1.270 (32,26)	1.450 (36,83)	1.650 (41,91)	2.090 (53,09)	2.450 (62,23)	2.650 (67,31)
A MIN	1.230 (31,24)	1.410 (35,81)	1.610 (40,89)	2.040 (51,82)	2.390 (60,71)	2.590 (65,79)

Unshaded values are part of the applicable TMS370 families.

Notes: 1) All linear dimensions are in inches (millimeters).
2) Falls within JEDEC MS-011
3) Falls within JEDEC MS-015 (32 pin only)
4) This drawing is subject to change without notice.

Figure 19–3. 40-Pin Plastic Shrink Dual-In-Line Package (NJ Suffix)

NJ (R-PDIP-T**) Plastic Shrink Dual-In-Line Package



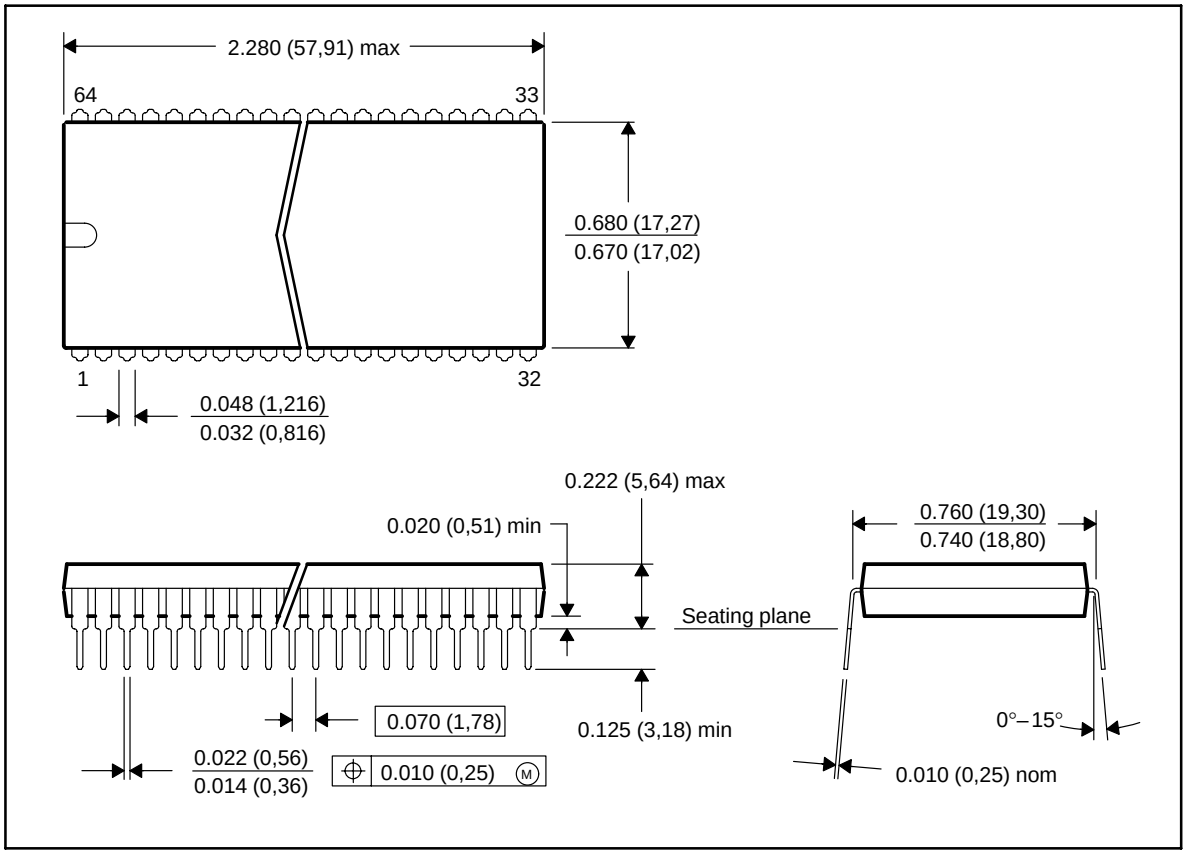
Dim \ Pins	40	54
A Max	1.425 (36,20)	2.031 (51,60)

Unshaded values are part of the applicable TMS370 families.

Notes: 1) Each pin centerline is located within 0,26 (0.010) of its true longitudinal position.
2) All linear dimensions are in inches (millimeters).
3) This drawing is subject to change without notice.

Figure 19–4. 64-Pin Plastic Shrink Dual-In-Line Package, 70-mil Pin Spacing (NM Suffix)

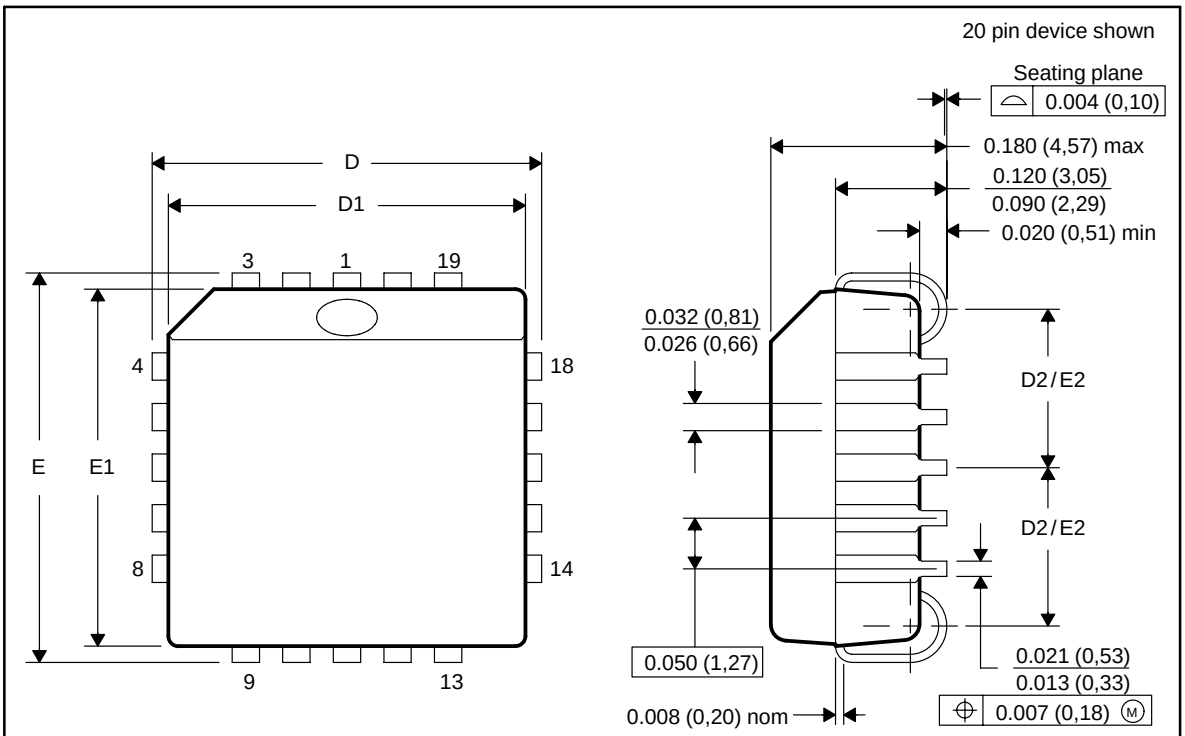
NM (R-PDIP-T64) Plastic Shrink Dual-In-Line Package



Notes: 1) All linear dimensions are in inches (millimeters).
2) This drawing is subject to change without notice.

Figure 19–5. Plastic-Leaded Chip Carrier Package (FN Suffix)

FN (S-PQCC-J**) Plastic J-Leaded Chip Carrier



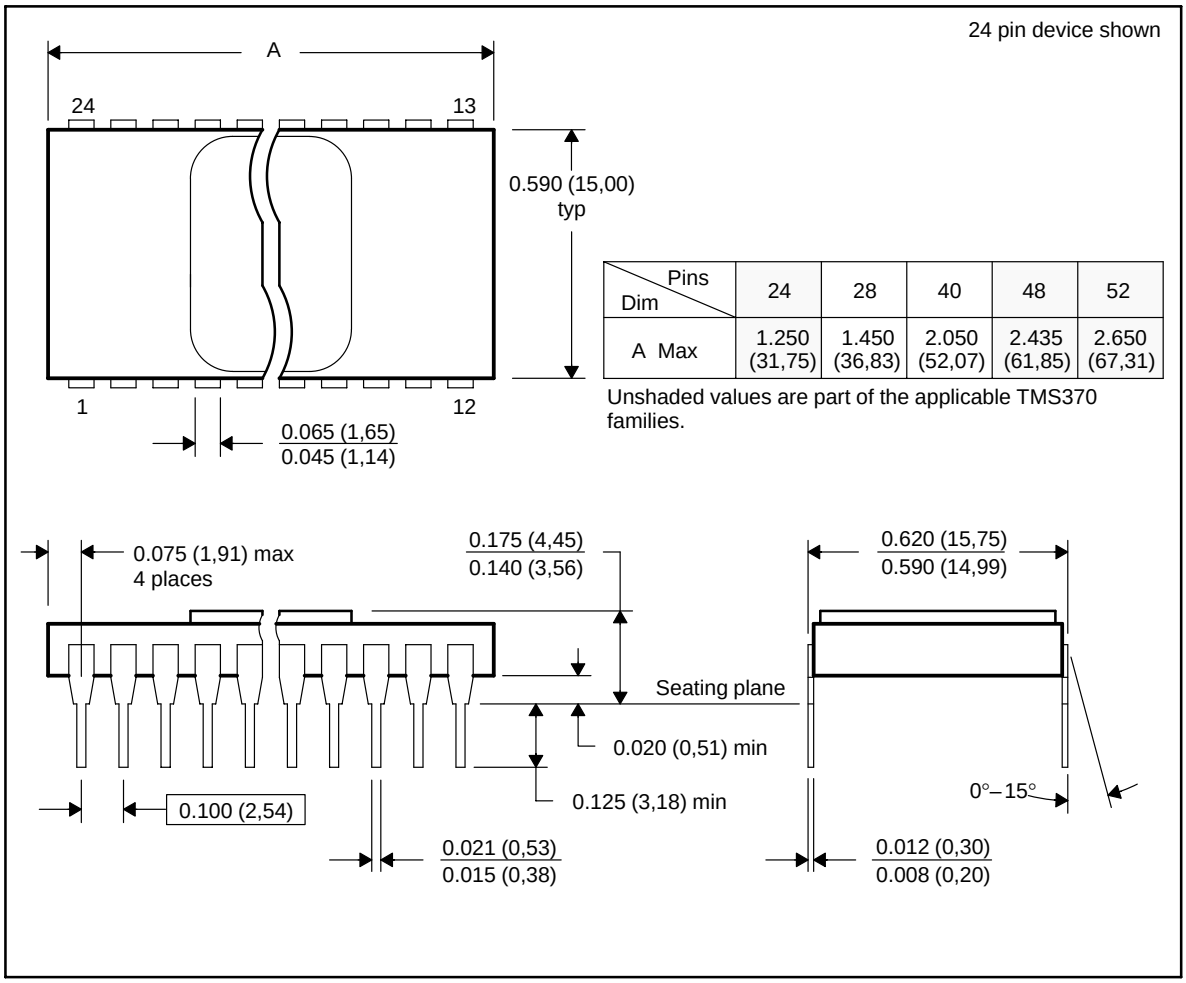
No. of Pins	D/E		D1/E1		D2/E2	
	Min	Max	Min	Max	Min	Max
20	0.385 (9,78)	0.395 (10,03)	0.350 (8,89)	0.356 (9,04)	0.141 (3,58)	0.169 (4,29)
28	0.485 (12,32)	0.495 (12,57)	0.450 (11,43)	0.456 (11,58)	0.191 (4,85)	0.219 (5,56)
44	0.685 (17,40)	0.695 (17,65)	0.650 (16,51)	0.656 (16,66)	0.291 (7,39)	0.319 (8,10)
52	0.785 (19,94)	0.795 (20,19)	0.750 (19,05)	0.756 (19,20)	0.341 (8,66)	0.369 (9,37)
68	0.985 (25,02)	0.995 (25,27)	0.950 (24,13)	0.958 (24,33)	0.441 (11,20)	0.469 (11,91)
84	1.185 (30,10)	1.195 (30,35)	1.150 (29,21)	1.158 (29,41)	0.541 (13,74)	0.569 (14,45)

Unshaded values are part of the applicable TMS370 families.

Notes: 1) All linear dimensions are in inches (millimeters).
2) Falls within JEDEC MS-018
3) This drawing is subject to change without notice.

Figure 19–6. Ceramic (Side-Braze) Dual-In-Line Package (JD Suffix)

JD (R-CDIP-T**) Ceramic Side-Braze Dual-In-Line Package



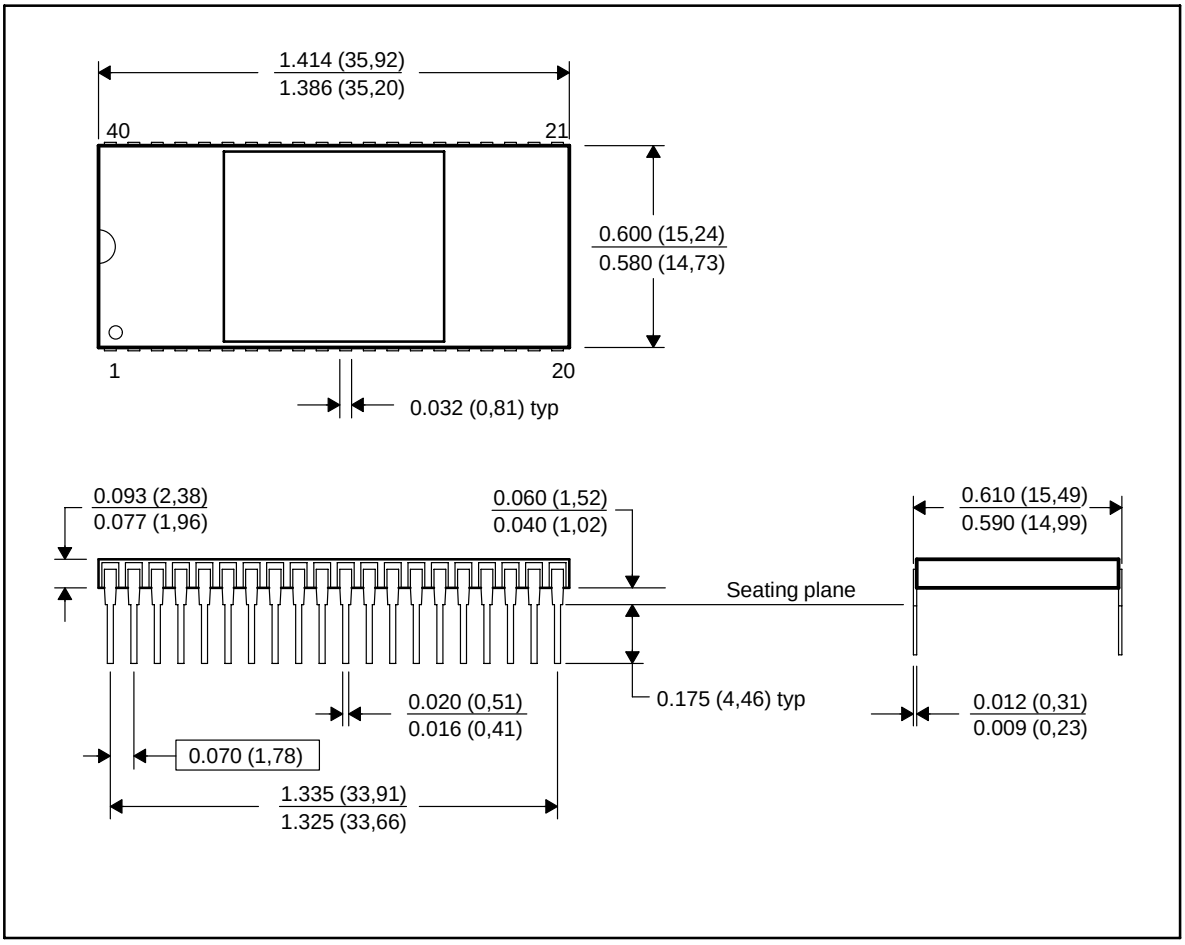
Dim \ Pins	24	28	40	48	52
A Max	1.250 (31,75)	1.450 (36,83)	2.050 (52,07)	2.435 (61,85)	2.650 (67,31)

Unshaded values are part of the applicable TMS370 families.

Notes: 1) All linear dimensions are in inches (millimeters).
2) This package can be hermetically sealed with a metal lid.
3) The terminals are gold plated.
4) This drawing is subject to change without notice.

Figure 19–7. 40-Pin Ceramic Shrink (Side-Braze) Dual-In-Line Package (JC Suffix)

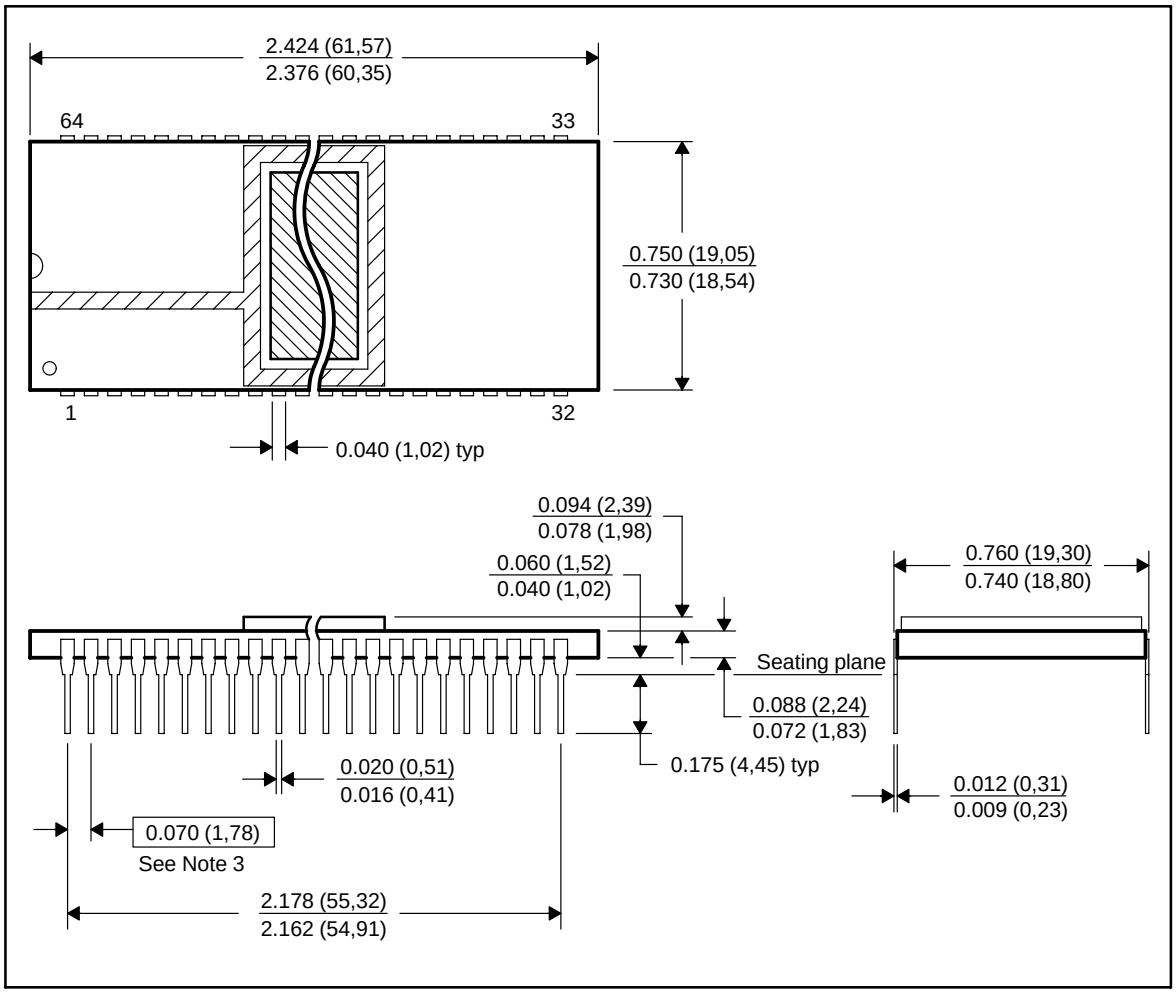
JC (R-CDIP-T40) Ceramic Side-Braze Dual-In-Line Package



Notes: 1) All linear dimensions are in inches (millimeters).
2) This package can be hermetically sealed with a metal lid.
3) The terminals are gold plated.
4) This drawing is subject to change without notice.

Figure 19–8. 64-Pin Ceramic Shrink Dual-In-Line Package (JN Suffix)

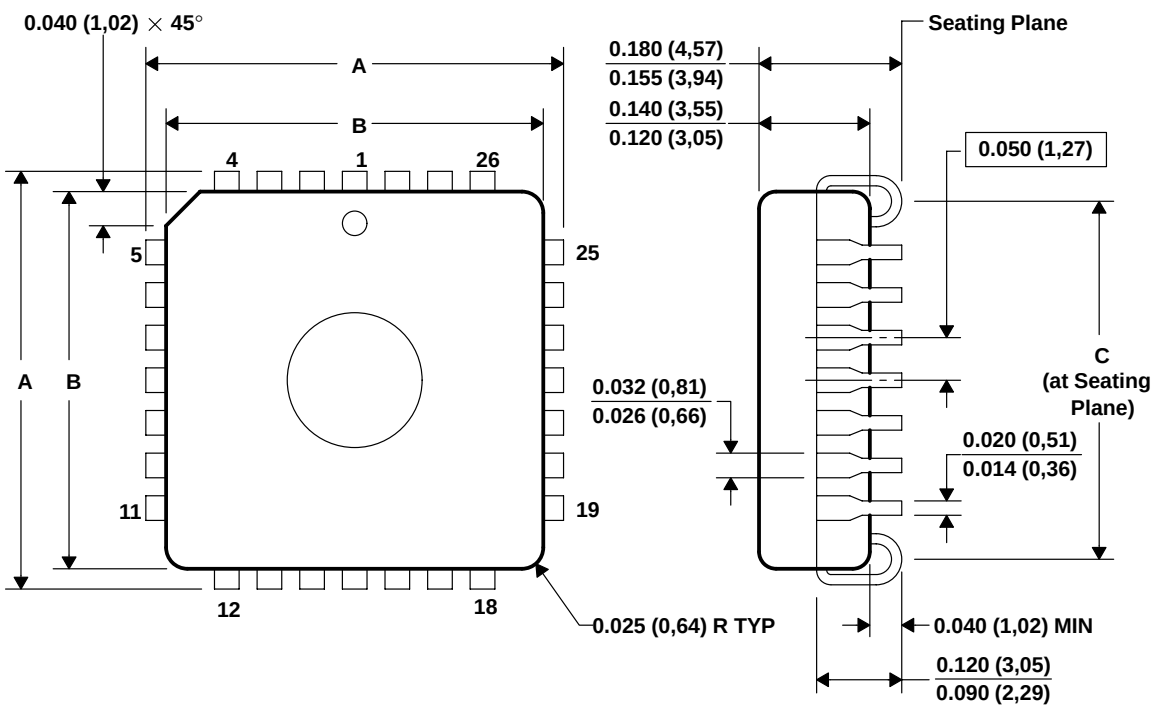
JN (R-CDIP-T64) Ceramic Dual-In-Line Package



Notes: 1) All linear dimensions are in inches (millimeters).
2) Each pin centerline located within 0.010 (0,26) of it true longitudinal position.
3) This drawing is subject to change without notice.

Figure 19–9. Ceramic-J-Leaded Chip Carrier Package (FZ Suffix)

FZ (S-CQCC-J**) J-Leaded Ceramic Chip Carrier



JEDEC OUTLINE	NO. OF PINS**	A		B		C	
		MIN	MAX	MIN	MAX	MIN	MAX
MO-087AA	28	0.485 (12,32)	0.495 (12,57)	0.430 (10,92)	0.455 (11,56)	0.410 (10,41)	0.430 (10,92)
MO-087AB	44	0.685 (17,40)	0.695 (17,65)	0.630 (16,00)	0.655 (16,64)	0.610 (15,49)	0.630 (16,00)
MO-087AC	52	0.785 (19,94)	0.795 (20,19)	0.730 (18,54)	0.765 (19,43)	0.680 (17,28)	0.740 (18,79)
MO-087AD	68	0.985 (25,02)	0.995 (25,27)	0.930 (23,62)	0.955 (24,26)	0.910 (23,11)	0.930 (23,62)

Notes: 1) All linear dimensions are in inches (millimeters).
2) This package can be hermetically sealed with a ceramic lid using glass frit.
3) This drawing is subject to change without notice.

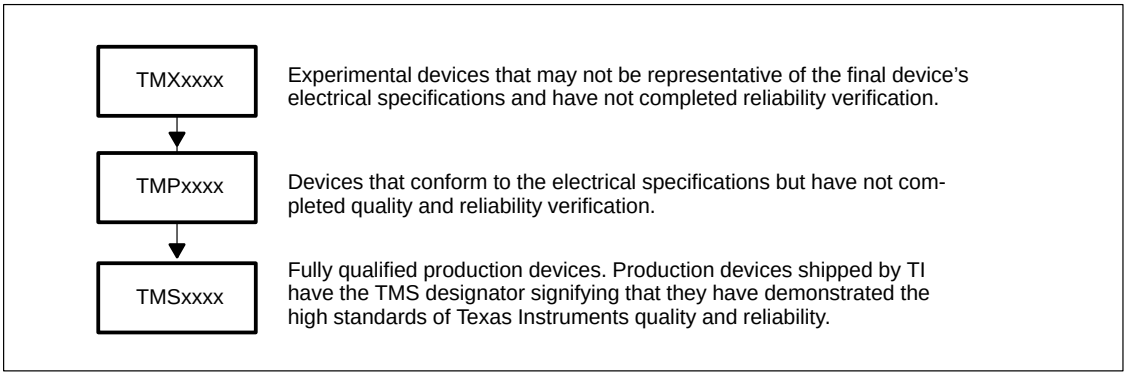
19.3 TMS370 Family Numbering and Symbol Conventions

All TMS370 devices are marked with information as to the type, package, copyright date(s), place of manufacture, and manufacturing data.

19.3.1 Production Device Prefix Designators

To help you identify the development stage of a product, Texas Instruments assigns a prefix designator: TMX, TMP, or TMS. These prefixes represent the evolutionary stages of product development from engineering prototypes through fully qualified production devices. Figure 19–10 depicts this evolutionary development flowchart.

Figure 19–10. Development Flowchart



- TMX devices are shipped against the following disclaimer:
  - Experimental product and its reliability has not been characterized.
  - Product is sold "as is."
  - Product is not warranted to be exemplary of final production version if or when released by Texas Instruments.
- TMP devices are shipped against the following disclaimer:
  - Customer understands that the product purchased here under has not been fully characterized and the expectation of reliability cannot be defined; therefore, Texas Instruments standard warranty refers only to the device's specifications.
  - No warranty of merchantability or fitness is expressed or implied.
- TMS devices have been fully characterized and the quality and reliability of the device has been fully demonstrated. Texas Instruments standard warranty applies.

19.3.2 Support Device Prefix Designators

The SE prefix designation is given to the system evaluator devices that are used for prototyping purposes. This designation applies only to the prototype members of the TMS370 family (SE370C702, SE370C712, SE370C722, SE370C732, SE370C736, SE370C742, SE370C756, SE370C758, SE370C759, SE370C768, SE370C769, SE370C777, SE370C686, SE370C792, and SE370C6C2).

SE devices are shipped against the following disclaimer:

Development products are intended for internal evaluation purposes only.

19.3.3 Device Numbering Conventions

Figure 19–11 illustrates the numbering and symbol nomenclature for the TMS370 family.

19.3.4 Device Symbols

Table 19–2 lists the graphic and letter designators for the symbolization.

Figure 19–11. TMS370 Family Number and Symbol Nomenclature

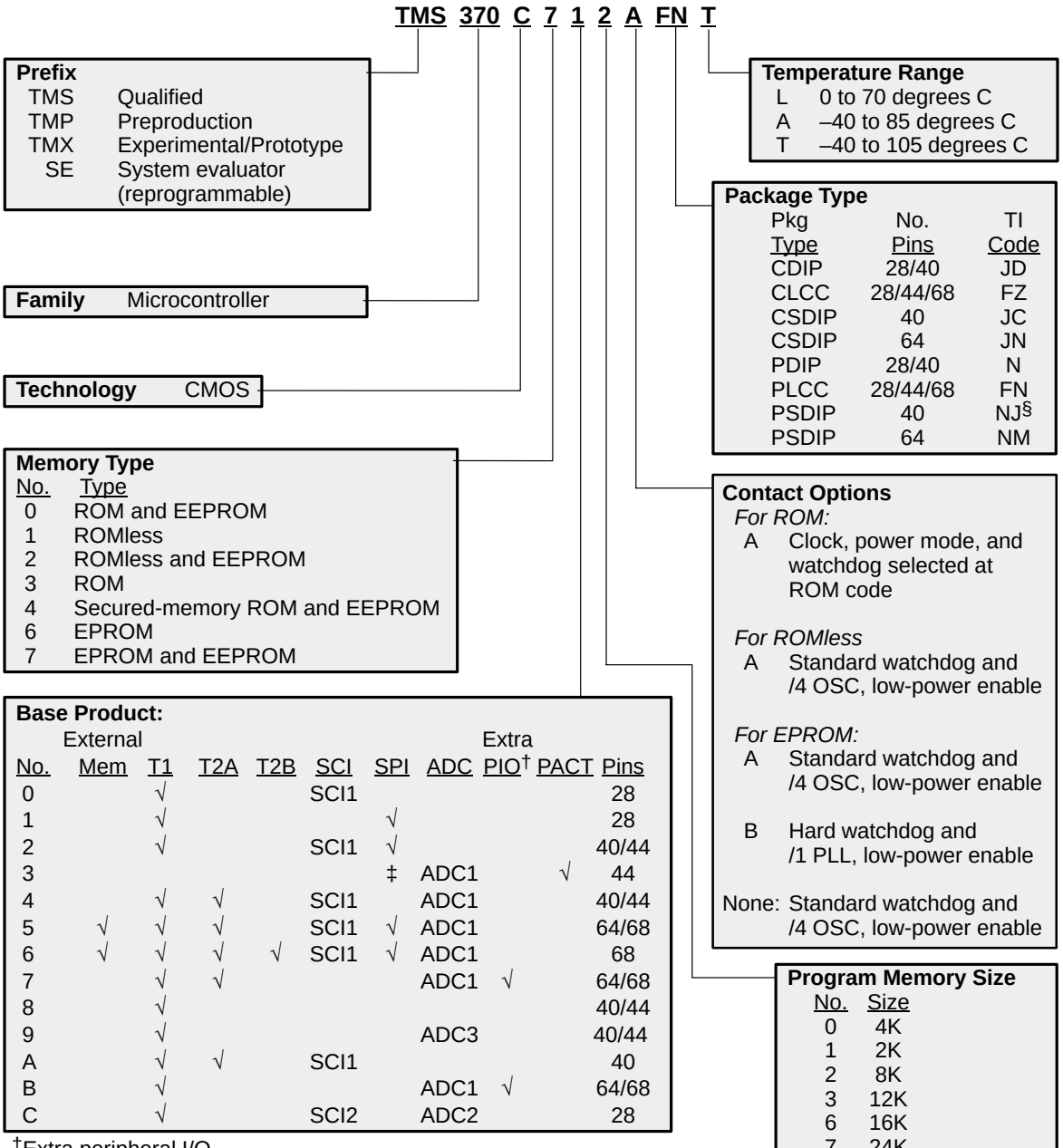


Table 19–2. Symbolization Designators

Letter Designator and Name		Description	Example
a)	Texas Instruments trademark	This is the standard TI signature.	
b)	Optional customer part number	You can specify a one-line number that will appear on your mask-ROM device: ❑ 28-pin PLCC: you can have 11 characters. ❑ 44-pin PLCC: you can have 12 characters. ❑ 68-pin PLCC: you can have 15 characters. ❑ 64-pin PSDIP: you can have 21 characters. ❑ 28- or 40-pin PDIP or 40-pin PSDIP: you can have 23 characters.	12345678901
c)	Customer's ROM code and package type	On mask-ROM devices, you will find up to a ten-digit number that TI assigns. This number is unique to the device that you ordered.	R1x00xxxN
d)	Standard device part number	This is the TMS- part number on OTP devices.	TMS370CxxxxNT
e)	Evaluation device part number	This is the SE- evaluation part number on reprogrammable EPROM devices.	SE370CxxxxFZT
f)	Tracking mark and date code	This six-character code denotes the date and place of the device assembly. **W** stands for the wafer fabrication plant, **A** represents the device revision, and **YY** and **WW** stand for the year and week the device was assembled.	WAYYWW
g)	TI microcode copyright	If your device has only one copyright date, it is that of the TI microcode.	©1986TI
h)	ROM code copyright	If your device has more than one copyright date, the second copyright date is that of the ROM code.	©1995TI
i)	Lot code	This eight-digit number designates the wafer lot.	12345678
j)	EIA code	This code is an alternate to the TI signature. It denotes that the device is a TI device. You **must** use either the EIA code or the TI trademark on your custom-programmed devices.	980
k)	Assembly site	This is the TI site at which the part was assembled. This designation may appear on the backside of the part.	Philippines
l)	Lot trace code	This seven–character code denotes the year, month, assembly lot , and assembly site code place of the device assembly. **YM** stands for year and month, **LLLL** represents the assembly lot, and **W** stands for the assembly site code.	YMLLLLW
m)	Wafer fab lot and die revision	**W** represents wafer fab lot and **A** represents the die revision.	WA

19.3.5 Symbolization by Device

The device symbolization of the TMS370 family members can be divided into two main categories:

TMS370 family members with mask-ROM

Family members with mask-ROM are custom-programmed devices in which the ROM is mask programmed according to your application code. These devices follow the prototyping and production flow outlined in Section 19.1. Since they are semicustom devices, they receive a unique ROM code identification number.

Figure 19–12 through Figure 19–18 illustrate typical labels for mask-ROM devices. Refer to Table 19–2, for the key to the lettered designators.

Figure 19–12. Typical Symbolization for 28-pin PLCC Mask-ROM Devices

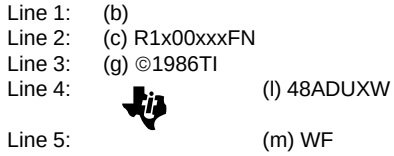


Figure 19–13. Typical Symbolization for 44-pin PLCC Mask-ROM Devices

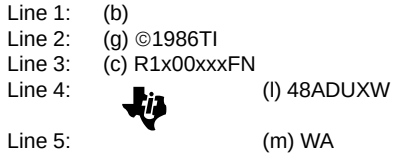


Figure 19–14. Typical Symbolization for 68-pin PLCC Mask-ROM Devices

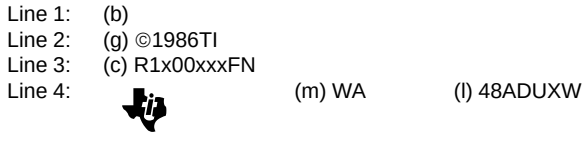


Figure 19–15. Typical Symbolization for 28-pin PDIP Mask-ROM Devices

Line 1:		(b)	
Line 2:		(c) R1x00xxxN	(f) WAYYWW
Line 3:	(g) ©1986TI	(h) ©1995TI	
Line 4:	(i) 12345678	(k) USA	

Figure 19–16. Typical Symbolization for 40-pin PDIP Mask-ROM Devices

Line 1:		(b)	
Line 2:		(c) R1x00xxxN	(f) WAYYWW
Line 3:	(g) ©1986TI	(h) ©1995TI	
Line 4:	(i) 12345678	(k) USA	

Figure 19–17. Typical Symbolization for 40-pin PSDIP Mask-ROM Devices

Line 1:		(b)	
Line 2:		(c) R1x00xxxNJ	
Line 3:	(g) ©1986TI	(h) ©1995TI	
Line 4:		(m) WA	(l) YMLLLLW

Figure 19–18. Typical Symbolization for 64-pin PSDIP Mask-ROM Devices

Line 1:		(b)	
Line 2:		(c) R1x00xxxNM	
Line 3:	(g) ©1986TI	(h) ©1995TI	
Line 4:	(i) 12345678	(f) WAYYWW	

TMS370 family members with program EPROM (OTP)

Family members with program EPROM (OTP) are standard device types and have a standard identification. The TMS370 family members with program EPROM include the TMS370C6xx and TMS370C7xx.

Figure 19–19 through Figure 19–23 illustrate typical labels for OTP devices. Refer to Table 19–2, on page 19-19, for the key to the lettered designators.

Figure 19–19. Typical Symbolization for 28-pin PLCC Program EPROM Devices (OTP)

Line 1: (d) TMS
Line 2: (d) 370CxxxxFNT
Line 3: (g) ©1986TI
Line 4:  (l) 59ADUXW
Line 5: (m) WF

Figure 19–20. Typical Symbolization for 44-pin PLCC Program EPROM Devices (OTP)

Line 1: (g) ©1986TI
Line 2: (d) TMS370CxxxxFNT
Line 3:  (l) 59ADUXW
Line 4: (m) WF

Figure 19–21. Typical Symbolization for 68-pin PLCC Program EPROM Devices (OTP)

Line 1: (g) ©1986TI
Line 2: (d) TMS370CxxxxFNT
Line 3:  (m) WA (l) 59ADUXW

Figure 19–22. Typical Symbolization for 28-pin PDIP Program EPROM Devices (OTP)

Line 1:  (d) TMS370CxxxxNT
Line 2: (f) WAYYWW
Line 3: (g) ©1986TI
Line 4: (i) 12345678 (k) USA

Figure 19–23. Typical Symbolization for 40-pin PDIP Program EPROM Devices (OTP)

Line 1:  (d) TMS370CxxxxNT
Line 2: (f) WAYYWW
Line 3: (g) ©1986TI
Line 4: (i) 12345678 (k) USA

Figure 19–24. Typical Symbolization for 40-pin PSDIP EPROM Devices (OTP)

Line 1:  (d) TMS370CxxxxNJT
Line 2: (g) ©1986TI
Line 3: (m) WA (l) YMLLLLW

Figure 19–25. Typical Symbolization for 64-pin PSDIP Reprogrammable EPROM Devices

Line 1: (d) TMS370CxxxxNMT

Line 2: (g) ©1986TI (f) WAYYWW
Line 3: (m) WA

TMS370 family members with reprogrammable EPROM

Family members with reprogrammable EPROM are standard device types and have a standard identification. The TMS370 family members with program EPROM include the SE370Cxxxx.

Figure 19–26 through Figure 19–30 illustrate typical labels for reprogrammable EPROM devices. Refer to Table 19–2, on page 19-19, for the key to the lettered designators.

Figure 19–26. Typical Symbolization for 28-pin and 40-pin CDIP Reprogrammable EPROM Devices

Line 1: (e) SE (j) 980
Line 2: (e) 370CxxxxJDT
Line 3: (f) WAYYWW
Line 4: (g) ©1986TI
Line 5: (i) 12345678 (k) USA

Figure 19–27. Typical Symbolization for 40-pin CSDIP Reprogrammable EPROM Devices

Line 1: (e) SE (j) 980
Line 2: (e) 370CxxxxJCT
Line 3: (f) WAYYWW
Line 4: (g) ©1986TI
Line 5: (i) 12345678 (k) USA

Figure 19–28. Typical Symbolization for 64-pin CSDIP Reprogrammable EPROM Devices

Line 1: (e) SE 

Line 2: (e) 370CxxxxJNT
Line 3: (f) WAYYWW
Line 4: (g) ©1986TI
Line 5: (i) 12345678 (k) USA

Figure 19–29. Typical Symbolization for 44-pin and 68-pin CLCC Reprogrammable EPROM Devices

Line 1:	(e) SE370CxxxxFZT	
Line 2:	(j) 980	(i) 4
Line 3:	(f) W	(i) 2
Line 4:	(f) A	(i) 6
Line 5:	(f) Y	(i) 9
Line 6:	(f) Y	(i) 0
Line 7:	(f) W	(i) 2
Line 8:		(i) 3
Line 8:	(k) USA	(g) ©1986TI

Figure 19–30. Typical Symbolization for 28-pin CLCC Reprogrammable EPROM Devices

Line 1:	(e) SE	
Line 2:	(e) 370CxxxxFZT	
Line 3:	(j) 980	(f) WAYYWW
Line 4:	(i) 12345678	
Line 5:	(g) ©1986TI	
Line 6:	(k) USA	

19.4 Ordering Information for Development Support Tools

The following development tools are available from TI.

19.4.1 TMS370 Macro Assembler, Linker, C Compiler, and Utilities

These software packages include all of the utilities required for developing object code for the TMS370 devices.

Part Number	Description
TMDS3740850-02	Assembler/Linker (DOS version)
TMDS3740855-02	C Compiler with Assembler/Linker (DOS version)
TMDS3740555-09	C Compiler with Assembler/Linker (Sun and HP700 versions)

19.4.2 TMS370 Design Kit

The TMS370 design kit contains the necessary software and hardware for you to evaluate TMS370Cx0x, TMS370Cx1x, and TMS370Cx5x devices.

Part Number	Description
TMDS3770110	TMS370 Design Kit

19.4.3 TMS370 Starter Kit

The TMS370 starter kit contains the necessary software and hardware for you to evaluate TMS370Cx0x, TMS370Cx1x, TMS370Cx2x, TMS370Cx32, TMS370Cx4x, TMS370Cx5x, TMS370Cx6x, TMS370Cx7x, TMS370CxAx, and TMS370CxBx devices; an Assembler diskette is included in the kit.

Part Number	Description
TMDS37000	TMS370 Starter Kit

19.4.4 TMS370 Microcontroller Programmer

With the TMS370 microcontroller programmer, you can program the TMS370 prototype devices. Each programmer comes with the necessary cables and control software for interfacing with a PC.

Part Number	Description
TMDS3760500A	TMS370 Programmer Base Unit (requires top)
TMDS3780510A	TMS370 Single-Unit 28- , 44- , and 68-pin PLCC (adapter top only)
TMDS3780511A	TMS370 Single-Unit 28- and 40-pin DIP, and 40- and 64-pin PSDIP (adapter top only)
TMDS3780512A	TMS370 Single-Unit 44-pin PLCC for TMS370Cx36 (adapter top only)
TMDS3780513A	TMS370 Single-Unit 44-pin PLCC/40-pin SDIP for TMS370Cx9x (adapter top only)
TMDS3780514A	TMS370 Single-Unit 28-pin PLCC/DIP for TMS370CxCx (adapter top only)

19.4.5 TMS370 XDS Systems

The XDS system provides software debugging and overall evaluation of a TMS370-based system. The XDS comes complete with necessary cables and the debugging program (not available in Europe).

Part Number	Description
TMDS3762210	TMS370 XDS/22 Emulator—110 V_{AC} (target cables sold separately)
TMDS3788828	28-pin DIP/PLCC Target Cable (’x1x)
TMDS3788844	40-pin DIP/SDIP, 44-pin PLCC Target Cable (’x2x and ’x4x)
TMDS3788868	64-pin SDIP, 68-pin PLCC Target Cable (’x5x)

19.4.6 TMS370 Compact Development Tool

The CDT370 includes a debugger, an assembler/linker, and the CDT board.

Part Number	Description
EDSCDT370	TMS370 Compact Development Tool for ’Cx0x, ’Cx1x, ’Cx2x, ’Cx4x, ’Cx5x, ’Cx8x, ’CxAx, and ’CxCx (target cables sold separately)
EDSCDT37T	TMS370 Compact Development Tool timer for ’Cx6x, ’Cx7x, and ’CxBx (target cables sold separately)
EDSCDT37P	TMS370 Compact Development Tool PACT for ’Cx32, ’Cx36, and ’Cx9x (target cables sold separately)

EDSTRG28PLCC02	28-pin PLCC Target Cable (’Cx0x devices)
EDSTRG28DIL	28-pin DIP Target Cable (’Cx1x devices)
EDSTRG28PLCC	28-pin PLCC Target Cable (’Cx1x devices)
EDSTRG40DILX	40-pin DIP Target Cable (’Cx2x devices)
EDSTRG40SDILX	40-pin SDIP Target Cable (’Cx2x devices)
EDSTRG44PLCCX	44-pin PLCC Target Cable (’Cx2x devices)
EDSTRG44PLCC32	44-pin PLCC Target Cable (’Cx32 devices)
EDSTRG44PLCC36	44-pin PLCC Target Cable (’Cx36 devices)
EDSTRG40SDIL	40-pin SDIP Target Cable (’Cx4x devices)
EDSTRG40DIL	40-pin DIP Target Cable (’Cx4x devices)
EDSTRG44PLCC	44-pin PLCC Target Cable (’Cx4x devices)
EDSTRG68PLCC	68-pin PLCC Target Cable (’Cx5x, ’Cx6x, ’Cx7x, and ’CxBx)
EDSTRG64SDIL	64-pin SDIP Target Cable (’Cx5x, ’Cx7x, and ’CxBx)
EDSTRG40DIL8X	40-pin DIP Target Cable (’Cx8x devices)
EDSTRG44PLCC8X	44-pin PLCC Target Cable (’Cx8x devices)
EDSTRG40SDIL05	40-pin SDIP Target Cable (’Cx9x devices)
EDSTRG44PLCC05	44-pin PLCC Target Cable (’Cx9x devices)
EDSTRG40DILAX	40-pin DIP Target Cable (’CxAx devices)
EDSTRG28DILCX	28-pin DIP Target Cable (’CxCx devices)
EDSTRG28PLCCCX	28-pin PLCC Target Cable (’CxCx devices)

19.4.7 TMS370 Converter Socket

The 40-pin TMS370C080 and TMS370C3A7 ROM devices rely on the 68-pin TMS370C758 device (with 32K EPROM) with a converter socket for field testing or software development.

Part Number	Description
TMDS37788OTP	Converter socket

19.4.8 XDS Target Connectors

For additional or replacement XDS target connectors, contact the TI factory repair department.

Appendix A

Differences Among the TMS370CxxxA, TMS370C7xxB, and TMS370Cxxx Devices (Contact Options)

The TMS370 family of microcontroller devices have been revised to include features that enhance performance and enable new application technologies. The specifications and descriptions included in this manual apply to the TMS370CxxxA and TMS370C7xxB devices. This appendix describes how the TMS370CxxxA and TMS370C7xxB devices differ from the TMS370Cxxx devices:

Topic **Page**

A.1 Watchdog (WD) Options

As described in Section 7.7, on page 7-21, you can configure the WD timer for the TMS370CxxxA devices as one of three mask options to accommodate various applications:

- A **standard WD** for ROMless, EPROM, and mask-ROM devices. This option lets you configure the WD timer as a non-WD or as a WD.
- A **hard WD** for mask-ROM devices. In this configuration, you can operate the WD timer only as a WD, providing additional system integrity.
- A **simple counter** for mask-ROM devices. In this configuration, you can use the WD timer as an event counter, a pulse accumulator, or an interval timer (similar to the non-WD mode in the standard WD configuration).

These WD options are summarized in Table A–1.

Table A–1. WD Option Summary for TMS370CxxxA Devices

	Standard WD			
Option	**WD Mode**	**Non-WD Mode**	**Hard WD**	**Simple Counter**
WD OVRFL TAP SEL bit and WD INPUT SELECT0–2 bits	Once the WD OVRFL RST ENA is set, these bit values can be changed only after any system reset.	These bits can be changed at any time, as long as WD OVRFL RST ENA is not set.	These WD bit values can be changed at any time, even if WD OVRFL RST ENA bit is set. However, the WD INPUT SELECT2 bit is not available.	Once the WD OVRFL RST ENA is set, these bit values can be changed only after any system reset.
Generates an interrupt when the WD counter overflows	No	Yes (If enabled)	No	Yes (If enabled)
Generates a system reset	Yes	No	Yes	No
WD OVRFL RST ENA bit	Select to be a WD. If bit = 1, WD counter does initiate a reset upon overflow. This bit is cleared by any system reset.	Select to be a non-WD. If bit = 0, WD counter does not initiate reset upon overflow.	This bit is ignored.	If bit = 0, WD bits and WD OVRFL TAP SELECT are not locked. If bit = 1, WD bits and WD OVRFL TAP SELECT are locked.
INT1 operation during low-power modes	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).	Automatically enabled as an NMI upon entry to a low-power mode.	Controlled by INT1 ENABLE bit (INT1.0) and INT1 NMI bit (SCCR2.1).
Available WD clock resources	8 options: – No clock – Event – Pulse accumulator – SYSCLK – 4 prescalers	8 options: – No clock – Event – Pulse accumulator – SYSCLK – 4 prescalers	4 options: – SYSCLK/4 – SYSCLK/16 – SYSCLK/64 – SYSCLK/256	8 options: – No clock – Event – Pulse accumulator – SYSCLK – 4 prescalers
Available devices	All devices	All devices	Mask-ROM devices	Mask-ROM devices

For the TMS370C7xxA devices (EPROM only), the only WD option is the standard WD (refer to Table A–3 on page A-4).

For the TMS370C7xxB devices (EPROM only), the only WD option is the hard WD mode (refer to Table A–3).

For the TMS370Cxxx devices (ROMless, mask-ROM, and EPROM), the only WD option is the standard WD. In this mode, you can set the WD OVRFL RST ENA bit (T1CTL2.7) to configure the WD timer as a WD, or clear the WD OVRFL RST ENA bit to configure the WD timer as a non-WD (refer to Table A–3).

A.2 Clock Options

For the TMS370CxxxA devices, the clock options are ROMless, ROM, and EPROM dependent. Clock options for ROM devices are configured through a programmable contact during manufacture. These clock options are summarized in Table A–2. (Refer to Table A–3, on page A-4.)

Table A–2. Clock Option Summary for TMS370CxxxA Devices

Option	Divide-by-4 (Standard Oscillator)	Divide-by-1 (Phase-Locked Loop)
SYSCLK range	0.5 MHz to 5 MHz	2 MHz to 5MHz
Available Devices	All devices	Mask-ROM devices

For the TMS370C7xxB devices (EPROM only), the only clock option is the divide-by-1 (PLL). (Refer to Table A–3).

For the TMS370Cxxx devices (ROMless, mask-ROM, and EPROM), the only clock option is the divide-by-4 (standard oscillator). (Refer to Table A–3).

A.3 Low-Power and Idle Modes

For the TMS370CxxxA devices (ROMless, mask-ROM, and EPROM), there are two low-power modes (halt and standby) and an idle mode. Bits 6 and 7 of SCCR2 select the halt, standby, or idle modes. For mask-ROM devices, low-power modes can be disabled permanently through a programmable contact at the time when the mask is manufactured. Low-power modes for EPROM and ROMless devices can be disabled or enabled through software only and cannot be disabled permanently through a mask option. Refer to Section 4.2, page 4-7 for more information about the low-power and idle modes.

For the TMS370C7xxB (EPROM only) and TMS370Cxxx (ROMless, mask-ROM, and EPROM) devices, the low-power modes cannot be disabled permanently. (Refer to Table A–3).

Table A–3. Summary for TMS370CxxxA, TMS370C7xxB, and TMS370Cxxx Devices

OPTIONS	EPROM			ROM		ROMLESS	
WD, Clock, & Low-power Mode	**Original Rev.**	**Revision A**	**Revision B**	**Original Rev.**	**Revision A**	**Original Rev.**	**Revision A**
	TMS370C6xx TMS370C7xx SE370C6xx SE370C7xx	**TMS370C6xxA TMS370C7xxA SE370C6xxA SE370C7xxA**	**TMS370C7xxB SE370C7xxB**	**TMS370C0xx TMS370C3xx**	**TMS370C0xxA TMS370C3xxA TMS370C4xxA**	**TMS370C1xx TMS370C2xx**	**TMS370C1xxA TMS370C2xxA**
Standard WD	Yes	Yes		Yes	Selected through contact option‡	Yes	Yes
Hard WD			Yes				
Simple Counter							
Divide-by-1			Yes		Selected through contact option§		
Divide-by-4	Yes	Yes		Yes		Yes	Yes
Low-power mode	Software configurable†	Software configurable†	Software configurable†	Software configurable†	Contact option enable/disable¶	Software configurable†	Software configurable†

† Low-power mode can be enabled or disabled through software. Refer to Table 4–3, on page 4-7, for the powerdown/idle control bits.
‡ Standard WD, hard WD, and simple counter options are available for ROM with revision A. One of these options must be specified when the devices are ordered.
§ Divide-by-1 and divide-by-4 options are avilable for ROM with revision A. One of these options must be specified when the devices are ordered.
¶ When ordering a device, low power must be specified to be enabled or disabled permanently through a programmable contact.

A.4 Timer 1 Control Register 2 (T1CTL2) Bits

Two of the T1CTL2 bits (register T1CTL2 shown in subsection 7.9.2 on page 7-34) operate differently according to which WD contact option you are using:

- **WD OVRFL INT FLAG bit (T1CTL2.5)**: For the TMS370CxxxA and TMS370C7xxB devices, this bit is set if the last reset is initiated by the WD counter. This bit is cleared by writing a zero to it or by any system reset that is not initiated by the WD counter; it is not cleared following a WD-initiated reset. WD resets can occur, regardless of the status of this flag.

  For the TMS370Cxxx devices, the bit indicates the status of the WD overflow interrupt. If the WD OVRFL RST ENA bit = 1, this bit is cleared by writing a zero to it or by a power cycle. If the WD OVRFL RST ENA bit = 0, this bit is cleared by any system reset. Once a WD reset has occurred and if the flag is not cleared, any subsequent WD-generated reset will not reset the system.

- **WD OVRFL RST ENA bit (T1CTL2.7)**: For the TMS370CxxxA and TMS370C7xxB devices, this bit is cleared by any system reset (for example, a power-up reset, an oscillator fault, or a reinitialized incorrect value to the WD timer). For the TMS370Cxxx devices, only a power-up reset can clear this bit.

A.5 System Control and Configuration Register 2 (SCCR2) Bits

Two of the SCCR2 bits (SCCR2 register shown in subsection 4.3.3 on page 4-16) operate differently according to which device you're using:

- **Bit SCCR2.2**:
  - For the TMS370CxxxA and TMS370C7xxxB devices, this bit is reserved; writing a zero or a one to this bit has no effect.
  - For the TMS370Cxxx device, this is the OSC FLT DISABLE bit (oscillator fault disable). **You must clear** (zero) this bit to ensure proper operation.

    (Note, this bit is shown reserved — without a name — in the register figure in subsection 4.3.3 on page 4-16).

- **Bit SCCR2.5**:
  - For the TMS370CxxxA and TMS370C7xxB devices, this bit is reserved; writing a zero or a one to this bit has no effect. When the system detects an oscillator fault, the device generates a system reset, regardless of the status of this bit.

- For the TMS370Cxxx devices, this bit is the OSC FLT RST ENA bit (oscillator failt reset enable). It determines whether or not a system reset is generated when an oscillator fault is detected.

  0 = A reset will not be generated if an oscillator fault is detected. An external hardware reset is required to restart the program.

  1 = A reset is generated when an oscillator fault is detected if bit SCCR.2 is cleared.

  (Note, this bit is shown reserved — without a name — in the register figure in subsection 4.3.3 on page 4-16).

A.6 AP Bit in the DEECTL Register (DEECTL.2)

The AP (array program) bit in the Data EEPROM Control Register (DEECTL, Section 6.2.2 on page 6-4) operates differently depending on which device you are using:

- *TMS370CxxxA and TMS370C7xxB devices with a single or multiple 256-bye array:* In a single programming cycle, the AP bit programs the entire array space with the value specified by the W1W0 bit. However, the device must be in the WPO mode for the array to be programmed. Moreover, there is no write protection during WPO mode; the write protection register (WPR, described in subsection 6.2.1 on page 6-3) is considered a normal data location within the data EEPROM array during this time. If the device is not in the WPO mode, the AP bit has no effect on the programming operation, and a single byte is programmed.

- *TMS370Cxxx devices with multiple 256-byte arrays:* In this case, the AP bit operates like that described for the TMS370CxxxA and TMS370C7xxB devices above.

- *TMS370Cxxx devices with a single 256-byte array:* In this case, the AP bit programs the entire array in a single programming cycle with the value specified by the W1W0 bit (Section 18.6, *EPROM/EEPROM Specifications,* on page 18-7 contains EPROM/EEPROM timings). Blocks that are protected in the WPR register are *not* programmed. If BLK0 is unprotected and bit W1W0 is zero, this function clears the WPR. Any array locations previously protected then lose their protection, but their contents are not altered during the current programming cycle.

A.7 Program EPROM Control Register (EPCTLx)

The EPCTLx register (subsection 6.4.2 on page 6-12) operates differently depending on the TMS370C758 device:

- *TMS370C758A and TMS370C758B devices:* The first 16K-byte array is controlled by register EPCTLL, located at address 101Eh (P01E), and the second 16K-byte array is controlled by register EPCTLM, located at address 101Ch (P01C). See Table A–4.
- *TMS370C758 devices:* In comparison to the ’C758A and ’C758B devices described above, control registers EPCTLL and EPCTLM are switched as shown in Table A–4. The first 16K-byte array is controlled by EPCTLM (at address 101Ch, P01C), and the second 16K-byte array is controlled by EPCTLL (at address 101Eh, P01E).

Table A–4. EPROM Memory Map for TMS370C758 Devices

	TMSC70C758A and TMS370C758B		**TMS370C758**	
EPROM size	32K bytes		32K bytes	
Mapped memory	First 16K (2000h–5FFFh)	Second 16K (6000h–9FFFh)	First 16K (2000h–5FFFh)	Second 16K (6000h–9FFFh)
Control registers	EPCTLL (P01E)	EPCTLM (P01C)	EPCTLM (P01C)	EPCTLL (P01E)

A.8 V_{CC1} and V_{CC2} Pins

The V_{CC1} and V_{CC2} pins are internally connected on the TMS370CxxxA and TMS370C7xxxB devices. For the TMS370Cxxx devices, these pins are not internally connected.

CAUTION

Remember, when using a TMS370CxxxA device, you cannot use the internal connections between pins (for example, the connection between V_{SS1} and V_{SS2}) for a jumper from one side of the chip to the other.

A.9 Electrical Specifications

This section describes the differences among the TMS370Cx5xA devices and TMS370Cx5x devices, both in terms of electrical specifications and SCI and SPI specifications.

A.9.1 Differences for TMS370Cx5x Devices

Table A–5 summarizes the differences in electrical specifications.

Table A–5. Switching Characteristics and Timing Requirements for External Read and Write †

Parameter			TMS370Cx5xA and TMS370Cx5xB Devices		TMS370Cx5x Devices		Unit
			Min	Max	Min	Max	
t_c	SYSCLK (system clock) cycle time	Divide-by-4 clock	200	2000	200	2000	ns
		Divide-by-1 (PLL)	200	500	Not available	Not available	ns
$t_{w(SCL)}$	SYSCLK low pulse duration		0.5 t_c – 25	0.5 t_c	0.5 t_c – 20	0.5 t_c	ns
$t_{w(SCH)}$	SYSCLK high pulse duration		0.5 t_c	0.5 t_c + 20	0.5 t_c	0.5 t_c + 20	ns
$t_{d(SCL-A)}$	Delay time, SYSCLK low to address R/W, and OCF valid			0.25 t_c + 75		0.25 t_c + 40	ns
$t_{v(A)}$	Address valid to EDS, CSE1, CSE2, CSH1, CSH2, CSH3, and CSPF low		0.5 t_c – 90		0.5 t_c – 50		ns
$t_{su(D)}$	Write data setup time to EDS high		0.75 t_c – 80‡		0.75 t_c – 40‡		ns
$t_{h(EH–A)}$	Address, R/W, and OCF hold time from EDS, CSE1, CSE2, CSH1, CSH2, CSH3, and CSPF high		0.5 t_c – 60		0.5 t_c – 40		ns
$t_{h(EH-D)W}$	Write data hold time from EDS high		0.75 t_c + 15		0.75 t_c + 15		ns
$t_{d(DZ–EL)}$	Delay time, data bus high impedance to EDS low (read cycle)		0.25 t_c – 35		0.25 t_c – 30		ns
$t_{d(EH-D)}$	Delay time, EDS high to data bus enable (read cycle)		1.25 t_c – 40		1.25 t_c – 40		ns
$t_{d(EL–DV)R}$	Delay time, EDS low to read data valid			t_c – 95‡		t_c – 65‡	ns
$t_{h(EH–D)R}$	Read data hold time from EDS high		0		0		ns
$t_{su(WT-SCH)}$	WAIT setup time to SYSCLK high		0.25 t_c + 70§		0.25 t_c + 75§		ns
$t_{h(SCH-WT)}$	WAIT hold time from SYSCLK high		0		0		ns
$t_{d(EL-WTV)}$	Delay time, EDS low to WAIT valid			0.5 t_c – 60		0.5 t_c – 70	ns

† t_c = system clock cycle time = 1/SYSCLK.
‡ If wait states, PFWait, or the autowait feature is used, add one SYSCLK cycle (t_c) to this value for each wait state invoked.
§ If the autowait feature is enabled, the WAIT input may assume a “don’t care” condition until the third cycle of the access. The WAIT signal must be synchronized with the high pulse of the SYSCLK signal while still conforming to the minimum setup time.

Table A–5. Switching Characteristics and Timing Requirements for External Read and Write (Continued)†

Parameter		TMS370Cx5xA and TMS370Cx5xB Devices		TMS370Cx5x Devices		Unit
		Min	Max	Min	Max	
t_w	Pulse duration, $\overline{EDS}$, $\overline{CSE1}$, $\overline{CSE2}$, CSH1, CSH2, CSH3, and $\overline{CSPF}$ low	$t_c - 80$‡	$t_c + 40$‡	$t_c - 40$‡	$t_c + 40$‡	ns
$t_{d(AV–DV)R}$	Delay time, address valid to read data valid		1.5 $t_c - 115$‡		1.5 $t_c - 75$‡	ns
$t_{d(AV–WTV)}$	Delay time, address valid to $\overline{WAIT}$ valid		$t_c - 115$		$t_c - 85$	ns
$t_{d(AV–EH)}$	Delay time, address valid to $\overline{EDS}$ high (end of write)	1.5 $t_c - 85$‡		1.5 $t_c - 40$‡		ns

† t_c = system clock cycle time = 1/SYSCLK.
‡ If wait states, PFWait, or the autowait feature is used, add one SYSCLK cycle (t_c) to this value for each wait state invoked.
§ If the autowait feature is enabled, the $\overline{WAIT}$ input may assume a "don't care" condition until the third cycle of the access. The $\overline{WAIT}$ signal must be synchronized with the high pulse of the SYSCLK signal while still conforming to the minimum setup time.

Table A–6. I_{OL} (Low-Level Output Current)

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{OL}	Low-level output voltage (port A, B, C, D, and $\overline{RESET}$)	I_{OL} = 1.4 mA for TMS370Cx5xA and TMS370Cx5xB I_{OL} = 2.0 mA for TMS370Cx5x			0.4	V

A.9.2 Differences in SCI1 and SPI Specifications

The differences in SCI1 and SPI electrical specifications among the TMS370CxxxA, TMS370C7xxB, and TMS370Cxxx devices are summarized in this subsection.

Table A–7. SCI1 Isosynchronous Mode Timing Characteristics for Internal Clock

Parameter		TMS370CxxxA and TMS370C7xxB		TMS370Cxxx		Unit
		Min	Max	Min	Max	
$t_{d(SCCL–TXDV)}$	Delay time, SCITXD valid after SCICLK low	– 50	60	– 50	50	ns

Table A–8. SPI External Timing Characteristics and Requirements

Parameter		TMS370CxxxA and TMS370C7xxB		TMS370Cxxx		Unit
		Min	Max	Min	Max	
$t_{w(SPCH)M}$	SPICLK high pulse duration	$t_c - 55$	0.5 $t_{c(SPC)}$ + 45	$t_c - 45$	0.5 $t_{c(SPC)}$ + 45	ns
$t_{d(SPCL–SIMOV)M}$	Delay time, SPISIMO valid after SPICLK low (polarity = 1)	–65	50	–50	50	ns
$t_{d(SPCL–SOMIV)S}$	Delay time, SPISOMI valid after SPICLK low (polarity = 1)		3.25 t_c + 130		3.25 t_c + 125	ns

A.9.3 Differences in EPROM Specifications

The differences in EPROM electrical specifications among the TMS370CxxxA, TMS370CxxxB, and TMS370xxx devices are summarized in Table A–9 and Table A–10.

Table A–9. Recommended EPROM Operating Conditions for Programming

Parameter		TMS370CxxxA and TMS370CxxxB			TMS370Cxxx			Unit
		Min	Nom	Max	Min	Nom	Max	
V_{PP}	Supply voltage at MC pin	13	13.2	13.5	12	12.5	13	V

Table A–10. Recommended EPROM Timing Requirements for Programming

Parameter		TMS370CxxxA and TMS370CxxxB			TMS370Cxxx			Unit
		Min	Nom	Max	Min	Nom	Max	
$t_{w(EPGM)}$	Programming signal pulse duration	0.40	0.50	3	0.95	2	30	ms

A.10 Summary of Differences

The differences among the TMS370CxxxA, TMS370C7xxB, and TMS370Cxxx devices are summarized in Table A–11.

Table A–11. Summary of Differences Among the TMS370Cxxx, TMS370CxxxA, and TMS370C7xxB Devices

Options	TMS370Cxxx ROMless, Mask-ROM, and EPROM	TMS370CxxxA ROMless and-Mask-ROM	TMS370CxxxA EPROM	TMS370C7xxB EPROM
Standard WD option (WD and non-WD modes)	✓	✓	✓	
Hard WD option		✓		✓
Simple counter option		✓		
Value of WD OVRFL RST ENA Bit (T1CTL2.7)	Cleared by power-up reset only	Cleared by any system reset	Cleared by any system reset	Cleared by any system reset
Value of WD OVRFL INT FLAG Bit (T1CTL2.5)	Once a WD reset has occurred and the bit remains set (1), a subsequent WD reset will not reset the system.	If a WD reset has occurred, it must be cleared by the software in order to determine the source of resets that follow.	If a WD reset has occurred, it must be cleared by the software in order to determine the source of resets that follow.	If a WD reset has occurred, it must be cleared by the software in order to determine the source of resets that follow.
Value of OSC FLT RST ENA Bit (SCCR2.5)	Determines whether or not a system reset is generated when an oscillator fault is detected	Reserved bit (value has no effect)	Reserved bit (value has no effect)	Reserved bit (value has no effect)
Value of OSC FLT DISABLE Bit (SCCR2.2)	Must be cleared to ensure proper operation	Reserved bit (value has no effect)	Reserved bit (value has no effect)	Reserved bit (value has no effect)
V_{CC1} and V_{CC2} pins	Not internally connected	Internally connected	Internally connected	Internally connected
Method of enabling low-power mode	Software configurable	Programmable contact when mask is manufactured	Software configurable	Software configurable

Table A–11. Summary of Differences Among the TMS370Cxxx, TMS370CxxxA, and TMS370C7xxB Devices(Continued)

Options	TMS370Cxxx ROMless, Mask-ROM, and EPROM	TMS370CxxxA ROMless and-Mask-ROM	TMS370CxxxA EPROM	TMS370C7xxB EPROM
Value of AP Bit (DEECTL.2)	If set, programs the entire array with the value specified by the W1W0 bit. 1) For multiple 256-byte arrays, must be in the WPO mode for the array to be programmed; otherwise this bit has no effect on the programming operation, and a single bit is programmed. 2) For a single 256-byte array, blocks that are protected in the WPR register are not programmed.	If set, programs the entire array with the value specified by the W1W0 bit. For single or multiple 256-byte arrays, must be in the WPO mode for the array to be programmed; otherwise this bit has no effect on the programming operation, and a single bit is programmed.	If set, programs the entire array with the value specified by the W1W0 bit. For single or multiple 256-byte arrays, must be in the WPO mode for the array to be programmed; otherwise this bit has no effect on the programming operation, and a single bit is programmed.	If set, programs the entire array with the value specified by the W1W0 bit. For single or multiple 256-byte arrays, must be in the WPO mode for the array to be programmed; otherwise this bit has no effect on the programming operation, and a single bit is programmed.

Appendix B

Peripheral File Memory Map

This appendix summarizes the peripheral file (PF) and control bit information. Each PF register is presented as a row of bit representations. The register designation (for example, SCCR0), memory map address, and the PF hex address (for example, P010) are to the left of each register. The registers described in this appendix are as follows:

Topic **Page**

B.1 Read/Write Definitions

The read/write accessibility of each bit is indicated in parentheses below each bit symbol, with the following definitions:

R	=	Read
W	=	Write
P	=	Write in privilege mode only
C	=	Clear only
S	=	Set only
–0	=	Cleared when the register is reset
–1	=	Set when the register is reset
–*	=	This bit exhibits a special behavior during or after a reset; see the description for this bit in the appropriate section

The shaded boxes in the tables denote privilege mode bits; that is, these bits can be written to only in the privilege mode.

B.2 Peripheral File Frame 1: System Configuration Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCCR0	1010h	P010	COLD START (RW–*)	OSC POWER (RP–0)	PF AUTOWAIT (RP–0)	OSC FLT FLAG (RW–0)	MC PIN WPO (R–0)	MC PIN DATA (R–*)	—	µP/µC MODE (R–*)
SCCR1	1011h	P011	—	—	—	AUTOWAIT DISABLE (RP–0)	—	MEMORY DISABLE (RP–*)	—	—
SCCR2	1012h	P012	HALT/ STANDBY (RP–0)	PWRDWN/ IDLE (RP–0)	—	BUS STEST (RP–0)	CPU STEST (RP–1)	—	INT1 NMI (RP–0)	PRIVILEGE DISABLE (RS–0)
	1013h	P013	Reserved							
EPCTLH	1014h	P014	BUSY (R-0)	VPPS (RW-0)	—	—	—	—	W0 (RW-0)	EXE (RW-0)
	1015h to 1016h	P015 to P016	Reserved							
INT1	1017h	P017	INT1 FLAG (RC–0)	INT1 PIN DATA (R–0)	—	—	—	INT1 POLARITY (RW–0)	INT1 PRIORITY (RW–0)	INT1 ENABLE (RW–0)
INT2	1018h	P018	INT2 FLAG (RC–0)	INT2 PIN DATA (R–0)	—	INT2 DATA DIR (RW–0)	INT2 DATA OUT (RW–0)	INT2 POLARITY (RW–0)	INT2 PRIORITY (RW–0)	INT2 ENABLE (RW–0)
INT3	1019h	P019	INT3 FLAG (RC–0)	INT3 PIN DATA (R–0)	—	INT3 DATA DIR (RW–0)	INT3 DATA OUT (RW–0)	INT3 POLARITY (RW–0)	INT3 PRIORITY (RW–0)	INT3 ENABLE (RW–0)
DEECTL	101Ah	P01A	BUSY (R–*)	—	—	—	—	AP (RW–0)	W1W0 (RW–0)	EXE (RW–0)
	101Bh	P01B	Reserved							
EPCTLM	101Ch	P01C	BUSY (R–0)	VPPS (RW–0)	—	—	—	—	W0 (RW–0)	EXE (RW–0)
	101Dh	P01D	Reserved							
EPCTLL	101Eh	P01E	BUSY (R-0)	VPPS (RW-0)	—	—	—	—	W0 (RW-0)	EXE (RW-0)
	101Fh	P01F	Reserved							

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to the following sections:

- ❏ SCCR0, SCCR1, and SCCR2: Section 4.3, page 4-13
- ❏ INT1, INT2, and INT3: Section 5.2, page 5-12
- ❏ DEECTL: subsection 6.2.2, page 6-4
- ❏ EPCTLL, EPCTLM, and EPCTLH: subsection 6.4.2, page 6-12

B.3 Peripheral File Frame 2: Digital Port Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
APORT1	1020h	P020	Reserved							
APORT2	1021h	P021	Port A Control Register 2							
ADATA	1022h	P022	Port A Data							
ADIR	1023h	P023	Port A Direction							
BPORT1	1024h	P024	Reserved							
BPORT2	1025h	P025	Port B Control Register 2							
BDATA	1026h	P026	Port B Data							
BDIR	1027h	P027	Port B Direction							
CPORT1	1028h	P028	Reserved							
CPORT2	1029h	P029	Port C Control Register 2							
CDATA	102Ah	P02A	Port C Data							
CDIR	102Bh	P02B	Port C Direction							
DPORT1	102Ch	P02C	Port D Control Register 1							
DPORT2	102Dh	P02D	Port D Control Register 2							
DDATA	102Eh	P02E	Port D Data							
DDIR	102Fh	P02F	Port D Direction							
	1030h to 1035h	P030 to P035	Reserved							
GDATA	1036h	P036	Port G Data							
GDIR	1037h	P037	Port G Direction							
	1038h to 1039h	P038 to P039	Reserved							
HDATA	103Ah	P03A	Reserved							Port H Data
HDIR	103Bh	P03B	Reserved							Port H Direction

For more information about these registers and control bits, refer to Section 4.4 on page 4-18.

B.4 Peripheral File Frame 3: SPI Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPICCR	1030h	P030	SPI SW RESET (RW–0)	CLOCK POLARITY (RW–0)	SPI BIT RATE2 (RW–0)	SPI BIT RATE1 (RW–0)	SPI BIT RATE0 (RW–0)	SPI CHAR2 (RW–0)	SPI CHAR1 (RW–0)	SPI CHAR0 (RW–0)
SPICTL	1031h	P031	RECEIVER OVERRUN (R–0)	SPI INT FLAG (R–0)	—	—	—	MASTER/ SLAVE (RW–0)	TALK (RW–0)	SPI INT ENA (RW–0)
	1032h to 1036h	P032 to P036	Reserved							
SPIBUF	1037h	P037	RCVD7 (R–0)	RCVD6 (R–0)	RCVD5 (R–0)	RCVD4 (R–0)	RCVD3 (R–0)	RCVD2 (R–0)	RCVD1 (R–0)	RCVD0 (R–0)
	1038h	P038	Reserved							
SPIDAT	1039h	P039	SDAT7 (RW–0)	SDAT6 (RW–0)	SDAT5 (RW–0)	SDAT4 (RW–0)	SDAT3 (RW–0)	SDAT2 (RW–0)	SDAT1 (RW–0)	SDAT0 (RW–0)
	103Ah to 103Ch	P03A to P03C	Reserved							
SPIPC1	103Dh	P03D	—	—	—	—	SPICLK DATA IN (R–0)	SPICLK DATA OUT (RW–0)	SPICLK FUNCTION (RW–0)	SPICLK DATA DIR (RW–0)
SPIPC2	103Eh	P03E	SPISIMO DATA IN (R–0)	SPISIMO DATA OUT (RW–0)	SPISIMO FUNCTION (RW–0)	SPISIMO DATA DIR (RW–0)	SPISOMI DATA IN (R–0)	SPISOMI DATA OUT (RW–0)	SPISOMI FUNCTION (RW–0)	SPISOMI DATA DIR (RW–0)
SPIPRI	103Fh	P03F	SPI STEST (RP–0)	SPI PRIORITY (RP–0)	SPI ESPEN (RP–0)	—	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 11.9 on page 11-13.

B.5 Peripheral File Frame 4: Timer 1 (T1) Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTR	1040h	P040	Bit 15	T1 Counter MSbyte						Bit 8
T1CNTR	1041h	P041	Bit 7	T1 Counter LSbyte						Bit 0
T1C	1042h	P042	Bit 15	Compare Register MSbyte						Bit 8
T1C	1043h	P043	Bit 7	Compare Register LSbyte						Bit 0
T1CC	1044h	P044	Bit 15	Capture/Compare Register MSbyte						Bit 8
T1CC	1045h	P045	Bit 7	Capture/Compare Register LSbyte						Bit 0
WDCNTR	1046h	P046	Bit 15	Watchdog Counter MSbyte						Bit 8
WDCNTR	1047h	P047	Bit 7	Watchdog Counter LSbyte						Bit 0
WDRST	1048h	P048	Bit 7	Watchdog Reset Key						Bit 0
T1CTL1	1049h	P049	WD OVRFL TAP SEL † (RP–0)	WD INPUT SELECT2† (RP–0)	WD INPUT SELECT1† (RP–0)	WD INPUT SELECT0† (RP–0)	—	T1 INPUT SELECT2 (RW–0)	T1 INPUT SELECT1 (RW–0)	T1 INPUT SELECT0 (RW–0)
T1CTL2	104Ah	P04A	WD OVRFL RST ENA † (RS–0)	WD OVRFL INT ENA (RW–0)	WD OVRFL INT FLAG (RC–*)	T1 OVRFL INT ENA (RW–0)	T1 OVRFL INT FLAG (RC–0)	—	—	T1 SW RESET (S–0)
T1CTL3	104Bh	P04B	Dual Compare Mode							
			T1EDGE INT FLAG (RC–0)	T1C2 INT FLAG (RC–0)	T1C1 INT FLAG (RC–0)	—	—	T1EDGE INT ENA (RW–0)	T1C2 INT ENA (RW–0)	T1C1 INT ENA (RW–0)
			Capture / Compare Mode							
			T1EDGE INT FLAG (RC–0)	—	T1C1 INT FLAG (RC–0)	—	—	T1EDGE INT ENA (RW–0)	—	T1C1 INT ENA (RW–0)
T1CTL4	104Ch	P04C	Dual Compare Mode							
			T1 MODE = 0 (RW–0)	T1C1 OUT ENA (RW–0)	T1C2 OUT ENA (RW–0)	T1C1 RST ENA (RW–0)	T1CR OUT ENA (RW–0)	T1EDGE POLARITY (RW–0)	T1CR RST ENA (RW–0)	T1EDGE DET ENA (RW–0)
			Capture / Compare Mode							
			T1 MODE = 1 (RW–0)	T1C1 OUT ENA (RW–0)	—	T1C1 RST ENA (RW–0)	—	T1EDGE POLARITY (RW–0)	—	T1EDGE DET ENA (RW–0)
T1PC1	104Dh	P04D	—	—	—	—	T1EVT DATA IN (R–0)	T1EVT DATA OUT (RW–0)	T1EVT FUNCTION (RW–0)	T1EVT DATA DIR (RW–0)
T1PC2	104Eh	P04E	T1PWM DATA IN (R–0)	T1PWM DATA OUT (RW–0)	T1PWM FUNCTION (RW–0)	T1PWM DATA DIR (RW–0)	T1IC/CR DATA IN (R–0)	T1IC/CR DATA OUT (RW–0)	T1IC/CR FUNCTION (RW–0)	T1IC/CR DATA DIR (RW–0)
T1PRI	104Fh	P04F	T1 STEST (RP–0)	T1 PRIORITY (RP–0)	—	—	—	—	—	—

† Once the WD OVRFL RST ENA bit is set, these bits cannot be changed until a reset; this applies only to the standard watchdog (WD) and to the simple counter. In the hard WD, these bits can be modified at any time; the WD INPUT SELECT2 bit is ignored.

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 7.9 on page 7-30.

B.6 Peripheral File Frame 4: PACT Control Registers

Desig-nation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PACTSCR	1040h	P040	DEFTIM OVRFL INT ENA (RW–0)	DEFTIM OVRFL INT FLAG (RC–0)	CMD/DEF AREA ENA (RW–0)	FAST MODE SELECT (RP–0)	PACT PRESCALE SELECT 3 (RP–0)	PACT PRESCALE SELECT 2 (RP–0)	PACT PRESCALE SELECT 1 (RP–0)	PACT PRESCALE SELECT 0 (RP–0)
CDSTART	1041h	P041	CMD/DEF AREA INT ENA (RW–0)	—	CMD/DEF AREA START BIT 5 (RW–0)	CMD/DEF AREA START BIT 4 (RW–0)	CMD/DEF AREA START BIT 3 (RW–0)	CMD/DEF AREA START BIT 2 (RW–0)	—	—
CDEND	1042h	P042	—	CMD/DEF AREA END BIT 6 (RW–0)	CMD/DEF AREA END BIT 5 (RW–0)	CMD/DEF AREA END BIT 4 (RW–0)	CMD/DEF AREA END BIT 3 (RW–0)	CMD/DEF AREA END BIT 2 (RW–0)	—	—
BUFPTR	1043h	P043	1 (R–1)	1 (R–1)	BUFFER POINTER BIT 5 (R–1)	BUFFER POINTER BIT 4 (R–1)	BUFFER POINTER BIT 3 (R–0)	BUFFER POINTER BIT 2 (R–0)	BUFFER POINTER BIT 1 (R–0)	—
	1044h	P044	Reserved							
SCICTLP	1045h	P045	PACT RXRDY (RC–0)	PACT TXRDY (R–1)	PACT PARITY (R–0)	PACT FE (RC–0)	PACT SCI RX INT ENA (RW–0)	PACT SCI TX INT ENA (RW–0)	—	PACT SCI SW RESET (RW–0)
RXBUFP	1046h	P046	PACT RXDT7 (R–0)	PACT RXDT6 (R–0)	PACT RXDT5 (R–0)	PACT RXDT4 (R–0)	PACT RXDT3 (R–0)	PACT RXDT2 (R–0)	PACT RXDT1 (R–0)	PACT RXDT0 (R–0)
TXBUFP	1047h	P047	PACT TXDT7 (RW–0)	PACT TXDT6 (RW–0)	PACT TXDT5 (RW–0)	PACT TXDT4 (RW–0)	PACT TXDT3 (RW–0)	PACT TXDT2 (RW–0)	PACT TXDT1 (RW–0)	PACT TXDT0 (RW–0)
OPSTATE	1048h	P048	PACT OP8 STATE (RW–0)	PACT OP7 STATE (RW–0)	PACT OP6 STATE (RW–0)	PACT OP5 STATE (RW–0)	PACT OP4 STATE (RW–0)	PACT OP3 STATE (RW–0)	PACT OP2 STATE (RW–0)	PACT OP1 STATE (RW–0)
CDFLAGS	1049h	P049	CMD/DEF INT 7 FLAG (RC–0)	CMD/DEF INT 6 FLAG (RC–0)	CMD/DEF INT 5 FLAG (RC–0)	CMD/DEF INT 4 FLAG (RC–0)	CMD/DEF INT 3 FLAG (RC–0)	CMD/DEF INT 2 FLAG (RC–0)	CMD/DEF INT 1 FLAG (RC–0)	CMD/DEF INT 0 FLAG (RC–0)
CPCTL1	104Ah	P04A	CP2 INT ENA (RW–0)	CP2 INT FLAG (RC–0)	CP2 CAPT RISING EDGE (RW–0)	CP2 CAPT FALLING EDGE (RW–0)	CP1 INT ENA (RW–0)	CP1 INT FLAG (RC–0)	CP1 CAPT RISING EDGE (RW–0)	CP1 CAPT FALLING EDGE (RW–0)
CPCTL2	104Bh	P04B	CP4 INT ENA (RW–0)	CP4 INT FLAG (RC–0)	CP4 CAPT RISING EDGE (RW–0)	CP4 CAPT FALLING EDGE (RW–0)	CP3 INT ENA (RW–0)	CP3 INT FLAG (RC–0)	CP3 CAPT RISING EDGE (RW–0)	CP3 CAPT FALLING EDGE (RW–0)
CPCTL3	104Ch	P04C	CP6 INT ENA (RW–0)	CP6 INT FLAG (RC–0)	CP6 CAPT RISING EDGE (RW–0)	CP6 CAPT FALLING EDGE (RW–0)	CP5 INT ENA (RW–0)	CP5 INT FLAG (RC–0)	CP5 CAPT RISING EDGE (RW–0)	CP5 CAPT FALLING EDGE (RW–0)
CPPRE	104Dh	P04D	BUFFER HALF/FULL INT ENA (RW–0)	BUFFER HALF/FULL INT FLAG (RC–0)	INPUT CAPT PRESCALE SELECT 3 (RW–0)	INPUT CAPT PRESCALE SELECT 2 (RW–0)	INPUT CAPT PRESCALE SELECT 1 (RW–0)	CP6 EVENT ONLY (RW–0)	EVENT COUNTER SW RESET (RW–0)	OP SET/CLR SELECT (RW–0)
WDRST	104Eh	P04E	WD Reset Key							
PACTPRI	104Fh	P04F	PACT STEST (RP–0)	—	PACT GROUP 1 PRIORITY (RP–0)	PACT GROUP 2 PRIORITY (RP–0)	PACT GROUP 3 PRIORITY (RP–0)	PACT MODE SELECT (RP–0)	PACT WD PRESCALE SELECT 1 (RP–0)	PACT WD PRESCALE SELECT 0 (RP–0)

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 15.11 on page 15-36.

B.7 Peripheral File Frame 5: SCI1 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCICCR	1050h	P050	STOP BITS (RW–0)	EVEN/ODD PARITY (RW–0)	PARITY ENABLE (RW–0)	ASYNC/ ISOSYNC (RW–0)	ADDRESS/ IDLE WUP (RW–0)	SCI CHAR2 (RW–0)	SCI CHAR1 (RW–0)	SCI CHAR0 (RW–0)
SCICTL	1051h	P051	—	—	SCI SW RESET (RW–0)	CLOCK (RW–0)	TXWAKE (RS–0)	SLEEP (RW–0)	TXENA (RW–0)	RXENA (RW–0)
BAUD MSB	1052h	P052	BAUDF (MSB) (RW–0)	BAUDE (RW–0)	BAUDD (RW–0)	BAUDC (RW–0)	BAUDB (RW–0)	BAUDA (RW–0)	BAUD9 (RW–0)	BAUD8 (RW–0)
BAUD LSB	1053h	P053	BAUD7 (RW–0)	BAUD6 (RW–0)	BAUD5 (RW–0)	BAUD4 (RW–0)	BAUD3 (RW–0)	BAUD2 (RW–0)	BAUD1 (RW–0)	BAUD0 (LSB) (RW–0)
TXCTL	1054h	P054	TXRDY (R–1)	TX EMPTY (R–1)	—	—	—	—	—	SCI TX INT ENA (RW–0)
RXCTL	1055h	P055	RX ERROR (R–0)	RXRDY (R–0)	BRKDT (R–0)	FE (R–0)	OE (R–0)	PE (R–0)	RXWAKE (R–0)	SCI RX INT ENA (RW–0)
	1056h	P056	Reserved							
RXBUF	1057h	P057	RXDT7 (R–0)	RXDT6 (R–0)	RXDT5 (R–0)	RXDT4 (R–0)	RXDT3 (R–0)	RXDT2 (R–0)	RXDT1 (R–0)	RXDT0 (R–0)
	1058h	P058	Reserved							
TXBUF	1059h	P059	TXDT7 (RW–0)	TXDT6 (RW–0)	TXDT5 (RW–0)	TXDT4 (RW–0)	TXDT3 (RW–0)	TXDT2 (RW–0)	TXDT1 (RW–0)	TXDT0 (RW–0)
	105Ah to 105Ch	P05A to P05C	Reserved							
SCIPC1	105Dh	P05D	—	—	—	—	SCICLK DATA IN (R–0)	SCICLK DATA OUT (RW–0)	SCICLK FUNCTION (RW–0)	SCICLK DATA DIR (RW–0)
SCIPC2	105Eh	P05E	SCITXD DATA IN (R–0)	SCITXD DATA OUT (RW–0)	SCITXD FUNCTION (RW–0)	SCITXD DATA DIR (RW–0)	SCIRXD DATA IN (R–0)	SCIRXD DATA OUT (RW–0)	SCIRXD FUNCTION (RW–0)	SCIRXD DATA DIR (RW–0)
SCIPRI	105Fh	P05F	SCI STEST (RP–0)	SCITX PRIORITY (RP–0)	SCIRX PRIORITY (RP–0)	SCI ESPEN (RP–0)	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 9.8 on page 9-21.

B.8 Peripheral File Frame 5: SCI2 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCICCR	1050h	P050	STOP BITS (RW–0)	EVEN/ODD PARITY (RW–0)	PARITY ENABLE (RW–0)	ASYNC ENABLE (RW–0)	ADDRESS/ IDLE WUP (RW–0)	SCI CHAR2 (RW–0)	SCI CHAR1 (RW–0)	SCI CHAR0 (RW–0)
SCICTL	1051h	P051	—	—	SCI SW RESET (RW–0)	CLOCK ENABLE (RW–0)	TXWAKE (RS–0)	SLEEP (RW–0)	TXENA (RW–0)	RXENA (RW–0)
BAUD MSB	1052h	P052	BAUDF (MSB) (RW–0)	BAUDE (RW–0)	BAUDD (RW–0)	BAUDC (RW–0)	BAUDB (RW–0)	BAUDA (RW–0)	BAUD9 (RW–0)	BAUD8 (RW–0)
BAUD LSB	1053h	P053	BAUD7 (RW–0)	BAUD6 (RW–0)	BAUD5 (RW–0)	BAUD4 (RW–0)	BAUD3 (RW–0)	BAUD2 (RW–0)	BAUD1 (RW–0)	BAUD0 (LSB) (RW–0)
TXCTL	1054h	P054	TXRDY (R–1)	TX EMPTY (R–1)	—	—	—	—	—	SCI TX INT ENA (RW–0)
RXCTL	1055h	P055	RX ERROR (R–0)	RXRDY (R–0)	BRKDT (R–0)	FE (R–0)	OE (R–0)	PE (R–0)	RXWAKE (R–0)	SCI RX INT ENA (RW–0)
	1056h	P056	Reserved							
RXBUF	1057h	P057	RXDT7 (R–0)	RXDT6 (R–0)	RXDT5 (R–0)	RXDT4 (R–0)	RXDT3 (R–0)	RXDT2 (R–0)	RXDT1 (R–0)	RXDT0 (R–0)
	1058h	P058	Reserved							
TXBUF	1059h	P059	TXDT7 (RW–0)	TXDT6 (RW–0)	TXDT5 (RW–0)	TXDT4 (RW–0)	TXDT3 (RW–0)	TXDT2 (RW–0)	TXDT1 (RW–0)	TXDT0 (RW–0)
	105Ah to 105Dh	P05A to P05D	Reserved							
SCIPC2	105Eh	P05E	SCITXD DATA IN (R–0)	SCITXD DATA OUT (RW–0)	SCITXD FUNCTION (RW–0)	SCITXD DATA DIR (RW–0)	SCIRXD DATA IN (R–0)	SCIRXD DATA OUT (RW–0)	SCIRXD FUNCTION (RW–0)	SCIRXD DATA DIR (RW–0)
SCIPRI	105Fh	P05F	SCI STEST (RP–0)	SCITX PRIORITY (RP–0)	SCIRX PRIORITY (RP–0)	SCI ESPEN (RP–0)	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 10.8 on page 10-17.

B.9 Peripheral File Frame 6: Timer 2A (T2A) Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2ACNTR	1060h	P060	Bit 15	T2A Counter MSbyte						Bit 8
T2ACNTR	1061h	P061	Bit 7	T2A Counter LSbyte						Bit 0
T2AC	1062h	P062	Bit 15	Compare Register MSbyte						Bit 8
T2AC	1063h	P063	Bit 7	Compare Register LSbyte						Bit 0
T2ACC	1064h	P064	Bit 15	Capture/Compare Register MSbyte						Bit 8
T2ACC	1065h	P065	Bit 7	Capture/Compare Register LSbyte						Bit 0
T2AIC	1066h	P066	Bit 15	Capture Register 2 MSbyte						Bit 8
T2AIC	1067h	P067	Bit 7	Capture Register 2 LSbyte						Bit 0
T2ACTL1	106Ah	P06A	—	—	—	T2A OVRFL INT ENA (RW–0)	T2A OVRFL INT FLAG (RC–0)	T2A INPUT SELECT1 (RW–0)	T2A INPUT SELECT0 (RW–0)	T2A SW RESET (S–0)
			Dual Compare Mode							
T2ACTL2	106Bh	P06B	T2AEDGE1 INT FLAG (RC–0)	T2AC2 INT FLAG (RC–0)	T2AC1 INT FLAG (RC–0)	—	—	T2AEDGE1 INT ENA (RW–0)	T2AC2 INT ENA (RW–0)	T2AC1 INT ENA (RW–0)
			Dual Capture Mode							
			T2AEDGE1 INT FLAG (RC–0)	T2AEDGE2 INT FLAG (RC–0)	T2AC1 INT FLAG (RC–0)	—	—	T2AEDGE1 INT ENA (RW–0)	T2AEDGE2 INT ENA (RW–0)	T2AC1 INT ENA (RW–0)
			Dual Compare Mode							
T2ACTL3	106Ch	P06C	T2A MODE= 0 (RW–0)	T2AC1 OUT ENA (RW–0)	T2AC2 OUT ENA (RW–0)	T2AC1 RST ENA (RW–0)	T2AEDGE1 OUT ENA (RW–0)	T2AEDGE1 POLARITY (RW–0)	T2AEDGE1 RST ENA (RW–0)	T2AEDGE1 DET ENA (RW–0)
			Dual Capture Mode							
			T2A MODE= 1 (RW–0)	—	—	T2AC1 RST ENA (RW–0)	T2AEDGE2 POLARITY (RW–0)	T2AEDGE1 POLARITY (RW–0)	T2AEDGE2 DET ENA (RW–0)	T2EDGE1 DET ENA (RW–0)
T2APC1	106Dh	P06D	—	—	—	—	T2AEVT DATA IN (R–0)	T2AEVT DATA OUT (RW–0)	T2AEVT FUNCTION (RW–0)	T2AEVT DATA DIR (RW–0)
T2APC2	106Eh	P06E	T2AIC2/PWM DATA IN (R–0)	T2AIC2/PWM DATA OUT (RW–0)	T2AIC2/PWM FUNCTION (RW–0)	T2AIC2/PWM DATA DIR (RW–0)	T2AIC1/CR DATA IN (R–0)	T2AIC1/CR DATA OUT (RW–0)	T2AIC1/CR FUNCTION (RW–0)	T2AIC1/CR DATA DIR (RW–0)
T2APRI	106Fh	P06F	T2A STEST (RP–0)	T2A PRIORITY (RP–0)	—	—	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 8.8 on page 8-17.

B.10 Peripheral File Frame 7: ADC1 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	REF VOLT SELECT2 (RW–0)	REF VOLT SELECT1 (RW–0)	REF VOLT SELECT0 (RW–0)	AD INPUT SELECT2 (RW–0)	AD INPUT SELECT1 (RW–0)	AD INPUT SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	Port E Data Input Register (R–0)							
ADENA	107Eh	P07E	Port E Data Input Enable Register (RW–0)							
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 12.3 on page 12-9.

B.11 Peripheral File Frame 7: ADC2 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	—	REF VOLT SELECT1 (RW–0)	REF VOLT SELECT0 (RW–0)	—	AD INPUT SELECT1 (RW–0)	AD INPUT SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	—	—	—	—	Port E Data Input Register (R-0)			
ADENA	107Eh	P07E	—	—	—	—	Port E Data Input Enable Register (RW-0)			
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 13.4 on page 13-10.

B.12 Peripheral File Frame 7: ADC3 Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	1070h	P070	CONVERT START (RW–0)	SAMPLE START (RW–0)	REF VOLT SELECT1 (RW-0)	REF VOLT SELECT0 (RW–0)	AD INPUT SELECT3 (RW–0)	AD INPUT SELECT2 (RW-0)	AD INPUT SELECT1 (RW–0)	AD INPUT SELECT0 (RW–0)
ADSTAT	1071h	P071	—	—	—	—	—	AD READY (R–0)	AD INT FLAG (RC–0)	AD INT ENA (RW–0)
ADDATA	1072h	P072	A-to-D Conversion Data Register (R–0)							
	1073h to 107Ch	P073 to P07C	Reserved							
ADIN	107Dh	P07D	Port E Data Input Register (R-0)							
ADENA	107Eh	P07E	Port E Data Input Enable Register (RW-0)							
ADPRI	107Fh	P07F	AD STEST (RP–0)	AD PRIORITY (RP–0)	AD ESPEN (RP–0)	—	—	—	AD RATE SELECT1 (RW-0)	AD RATE SELECT0 (RW-0)

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 14.4 on page 14-11.

B.13 Peripheral File Frame 8: Timer 2B (T2B) Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2BCNTR	1080h	P080	Bit 15	T2B Counter MSbyte						Bit 8
T2BCNTR	1081h	P081	Bit 7	T2B Counter LSbyte						Bit 0
T2BC	1082h	P082	Bit 15	Compare Register MSbyte						Bit 8
T2BC	1083h	P083	Bit 7	Compare Register LSbyte						Bit 0
T2BCC	1084h	P084	Bit 15	Capture/Compare Register MSbyte						Bit 8
T2BCC	1085h	P085	Bit 7	Capture/Compare Register LSbyte						Bit 0
T2BIC	1086h	P086	Bit 15	Capture Register 2 MSbyte						Bit 8
T2BIC	1087h	P087	Bit 7	Capture Register 2 LSbyte						Bit 0
T2BCTL1	108Ah	P08A	—	—	—	T2B OVRFL INT ENA (RW–0)	T2B OVRFL INT FLAG (RC–0)	T2B INPUT SELECT1 (RW–0)	T2B INPUT SELECT0 (RW–0)	T2B SW RESET (S–0)
T2BCTL2	108Bh	P08B	Dual Compare Mode							
			T2BEDGE1 INT FLAG (RC–0)	T2BC2 INT FLAG (RC–0)	T2BC1 INT FLAG (RC–0)	—	—	T2BEDGE1 INT ENA (RW–0)	T2BC2 INT ENA (RW–0)	T2BC1 INT ENA (RW–0)
			Dual Capture Mode							
			T2BEDGE1 INT FLAG (RC–0)	T2BEDGE2 INT FLAG (RC–0)	T2BC1 INT FLAG (RC–0)	—	—	T2BEDGE1 INT ENA (RW–0)	T2BEDGE2 INT ENA (RW–0)	T2BC1 INT ENA (RW–0)
T2BCTL3	108Ch	P08C	Dual Compare Mode							
			T2B MODE= 0 (RW–0)	T2BC1 OUT ENA (RW–0)	T2BC2 OUT ENA (RW–0)	T2BC1 RST ENA (RW–0)	T2BEDGE1 OUT ENA (RW–0)	T2BEDGE1 POLARITY (RW–0)	T2BEDGE1 RST ENA (RW–0)	T2BEDGE1 DET ENA (RW–0)
			Dual Capture Mode							
			T2B MODE= 1 (RW–0)	—	—	T2BC1 RST ENA (RW–0)	T2BEDGE2 POLARITY (RW–0)	T2BEDGE1 POLARITY (RW–0)	T2BEDGE2 DET ENA (RW–0)	T2BEDGE1 DET ENA (RW–0)
T2BPC1	108Dh	P08D	—	—	—	—	T2BEVT DATA IN (R–0)	T2BEVT DATA OUT (RW–0)	T2BEVT FUNCTION (RW–0)	T2BEVT DATA DIR (RW–0)
T2BPC2	108Eh	P08E	T2BIC2/PWM DATA IN (R–0)	T2BIC2/PWM DATA OUT (RW–0)	T2BIC2/PWM FUNCTION (RW–0)	T2BIC2/PWM DATA DIR (RW–0)	T2BIC1/CR DATA IN (R–0)	T2BIC1/CR DATA OUT (RW–0)	T2BIC1/CR FUNCTION (RW–0)	T2BIC1/CR DATA DIR (RW–0)
T2BPRI	108Fh	P08F	T2B STEST (RP–0)	T2B PRIORITY (RP–0)	—	—	—	—	—	—

Note: Shaded boxes can be written to in the privilege mode.

For more information about these registers and control bits, refer to Section 8.8 on page 8-17.

Appendix C

Block Diagrams

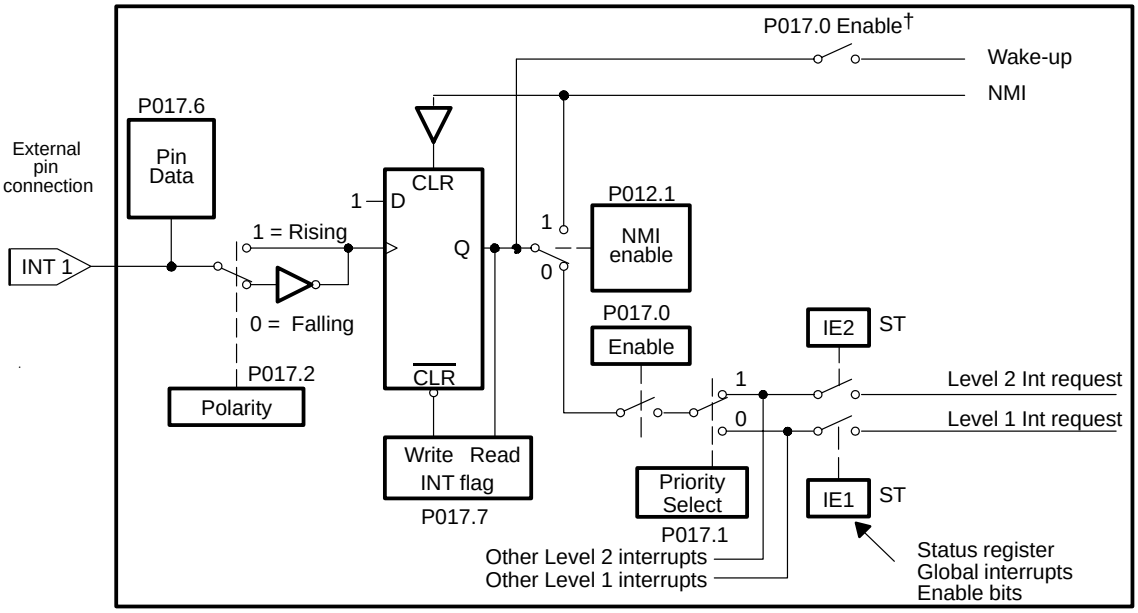
This appendix summarizes the block diagrams of the major circuits.

Topic **Page**

C.1 Interrupts

The block diagram for interrupt 1 is shown in Figure C–1. This interrupt is controlled by the SCCR2 (P012) and INT1 (P017) registers. The control bits for these registers are shown in Section B.2, page B-3.

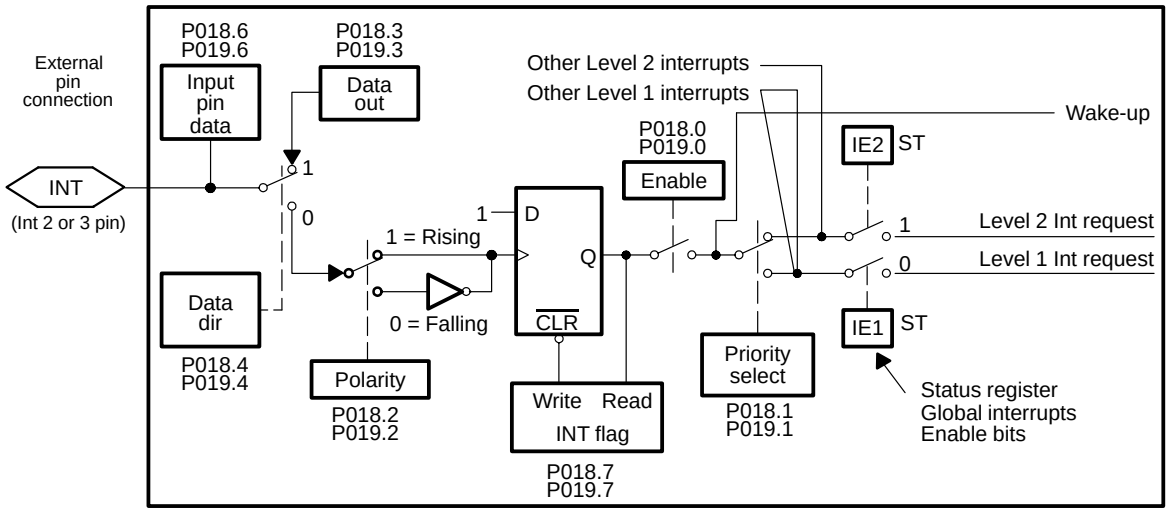
Figure C–1. Interrupt 1 Block Diagram



† This bit is ignored if you are using the hard watchdog option.

The block diagram for interrupts 2 and 3 is shown in Figure C–2. These interrupts are controlled by the INT2 (P018) and INT3 (P019) registers. The control bits for these registers are shown in Section B.2, page B-3.

Figure C–2. Interrupts 2 and 3 Block Diagram



C.2 Timer 1 (T1) Module

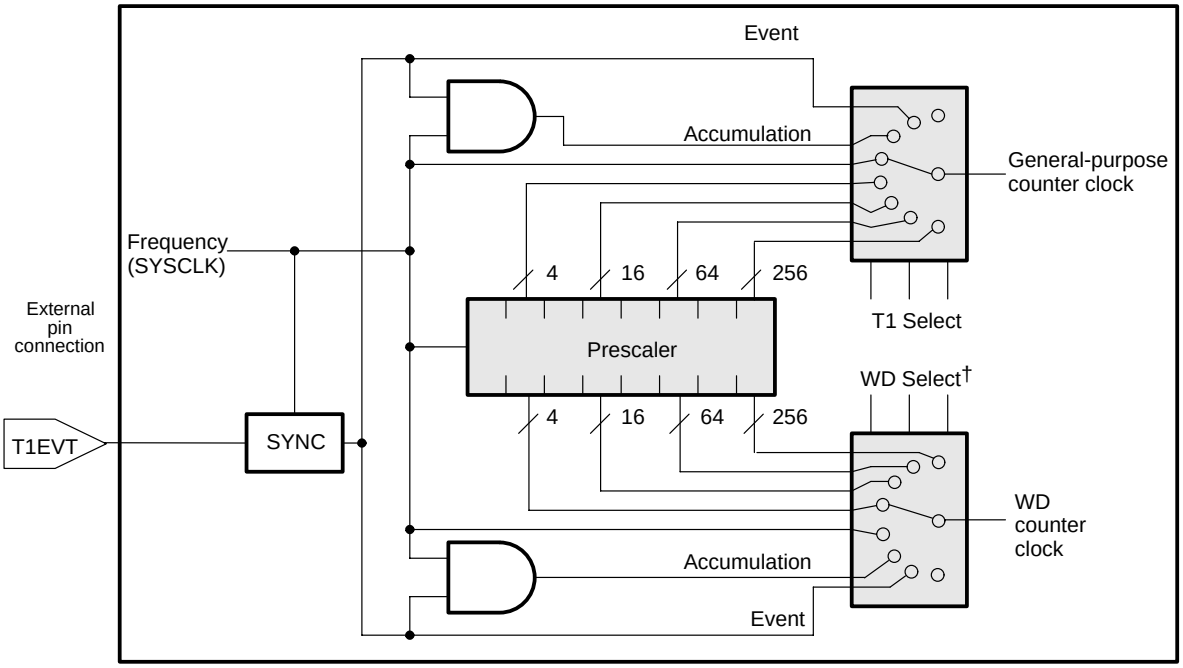
The prescaler for the T1 module is shown in Figure C–3. The prescaler is controlled by the T1CTL1 register, located at P049 in peripheral file frame 4. The control bits for this register are shown in Section B.5, page B-6.

The T1 WD timer is shown in Figure C–4 *(a)* through *(c)*. The WD timer is controlled by the following registers:

- WDCNTR (P046 and P047)
- WDRST (P048)
- T1CTL1 (P049)
- T1CTL2 (P04A)

The control bits for these registers are shown in Section B.5, page B-6.

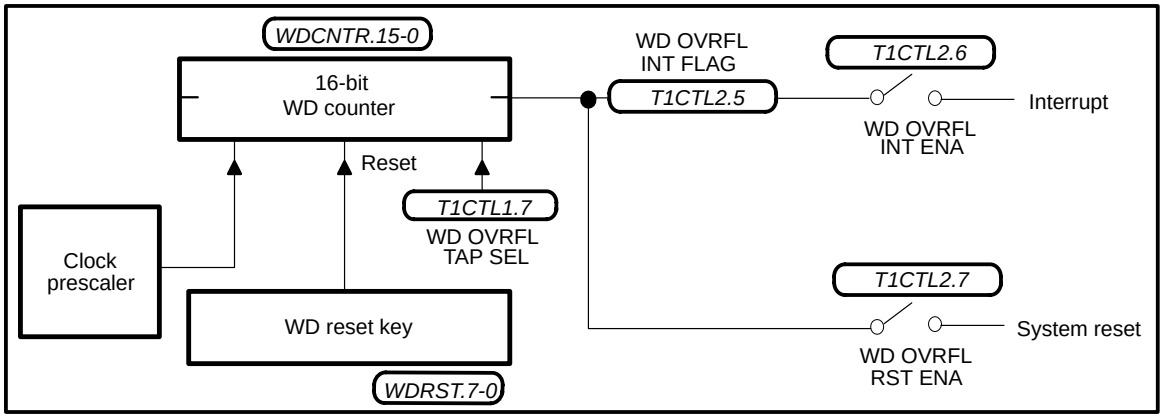
Figure C–3. T1 System Clock Prescaler Block Diagram



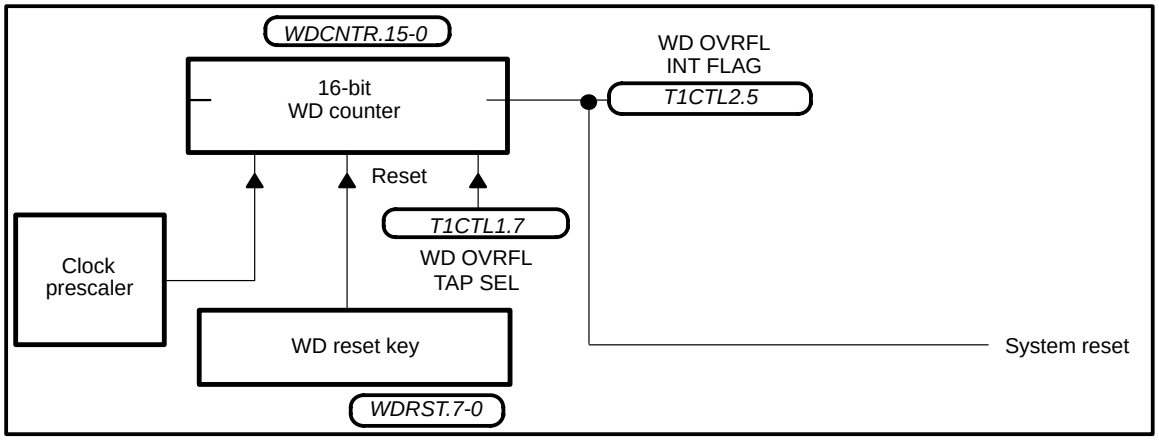
Note: For the hard WD option, the 8-bit prescaler provides four possible clock sources by dividing the system clock by 4, 16, 64, or 256.

Figure C–4. WD Timer Block Diagrams

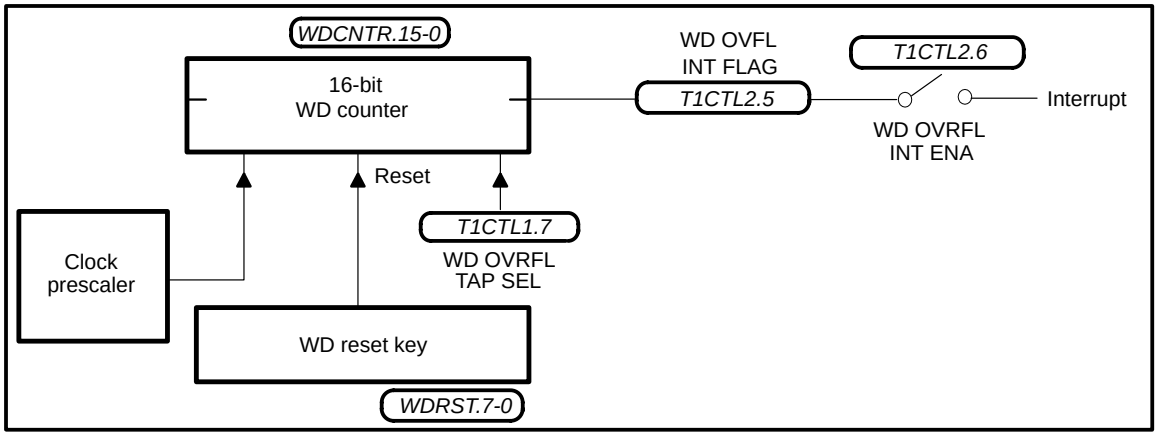
(a) Standard WD



(b) Hard WD



(c) Simple Counter

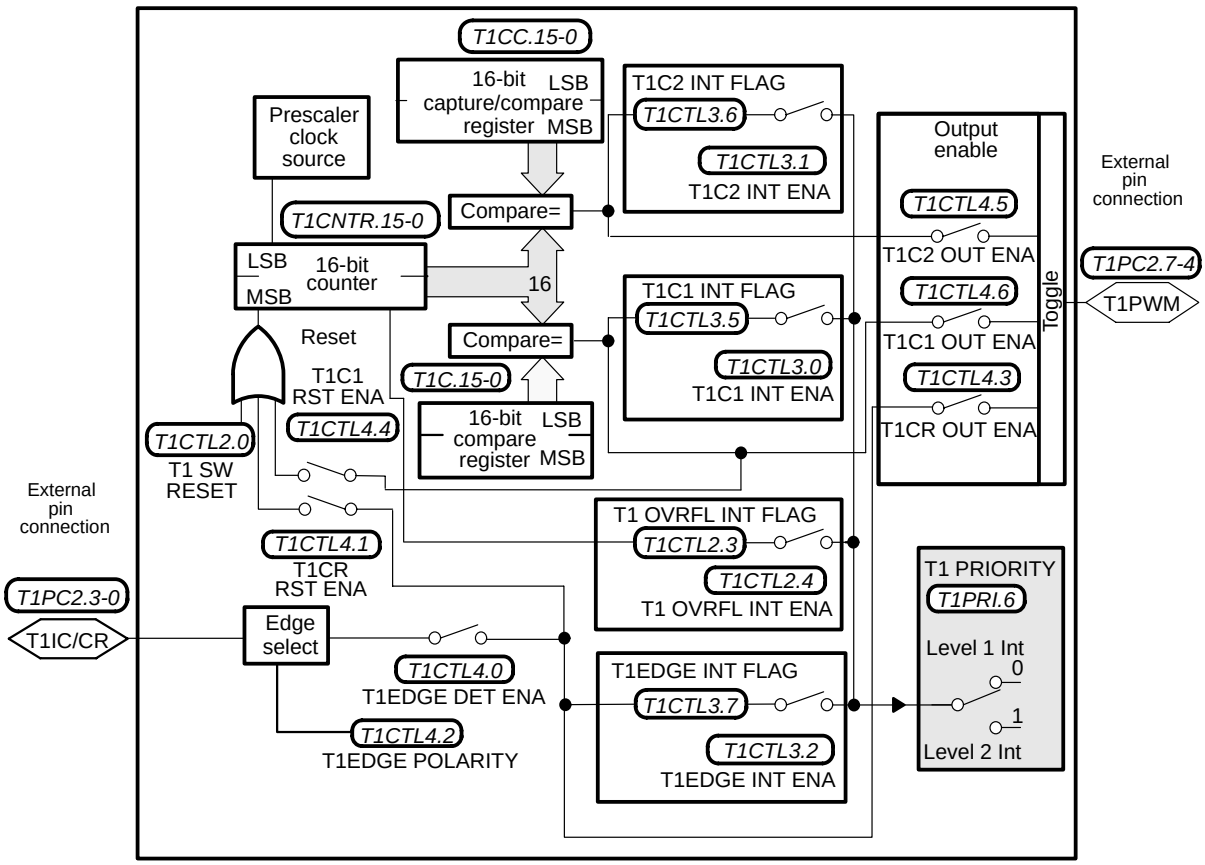


The T1 module has two operating modes:

- **Dual compare mode** is shown in Figure C–5
- **Capture/compare mode** is shown in Figure C–6

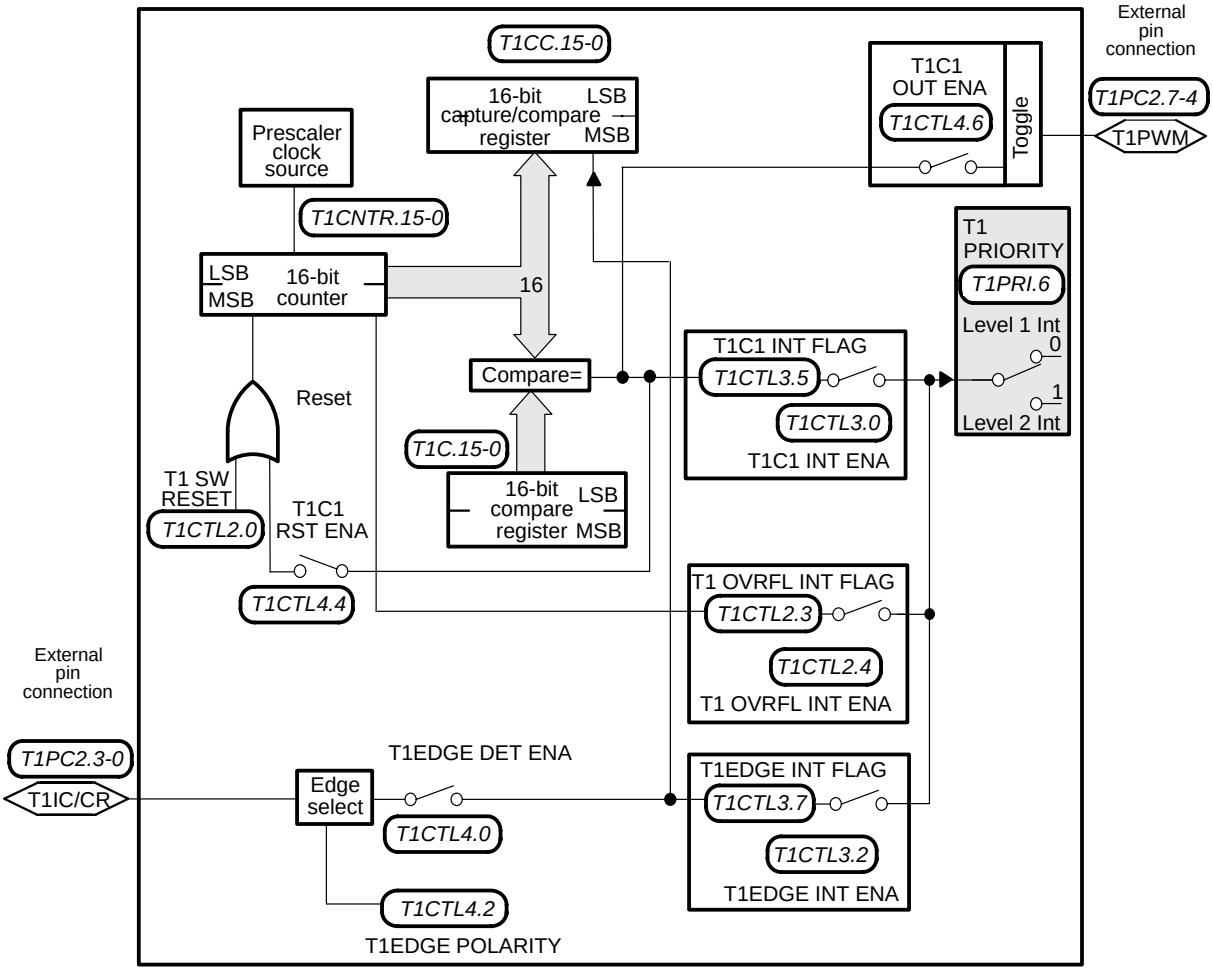
For a summary of the T1 module control registers and bits, refer to Section B.5, page B-6.

Figure C–5. T1: Dual Compare Mode Block Diagram



Note: The annotations on this diagram identify the register and the bit(s) in the peripheral frame. For example, the actual address of T1CTL2.0 is 104Ah, bit 0, in the T1CTL2 register.

Figure C–6. T1: Capture/Compare Mode Block Diagram



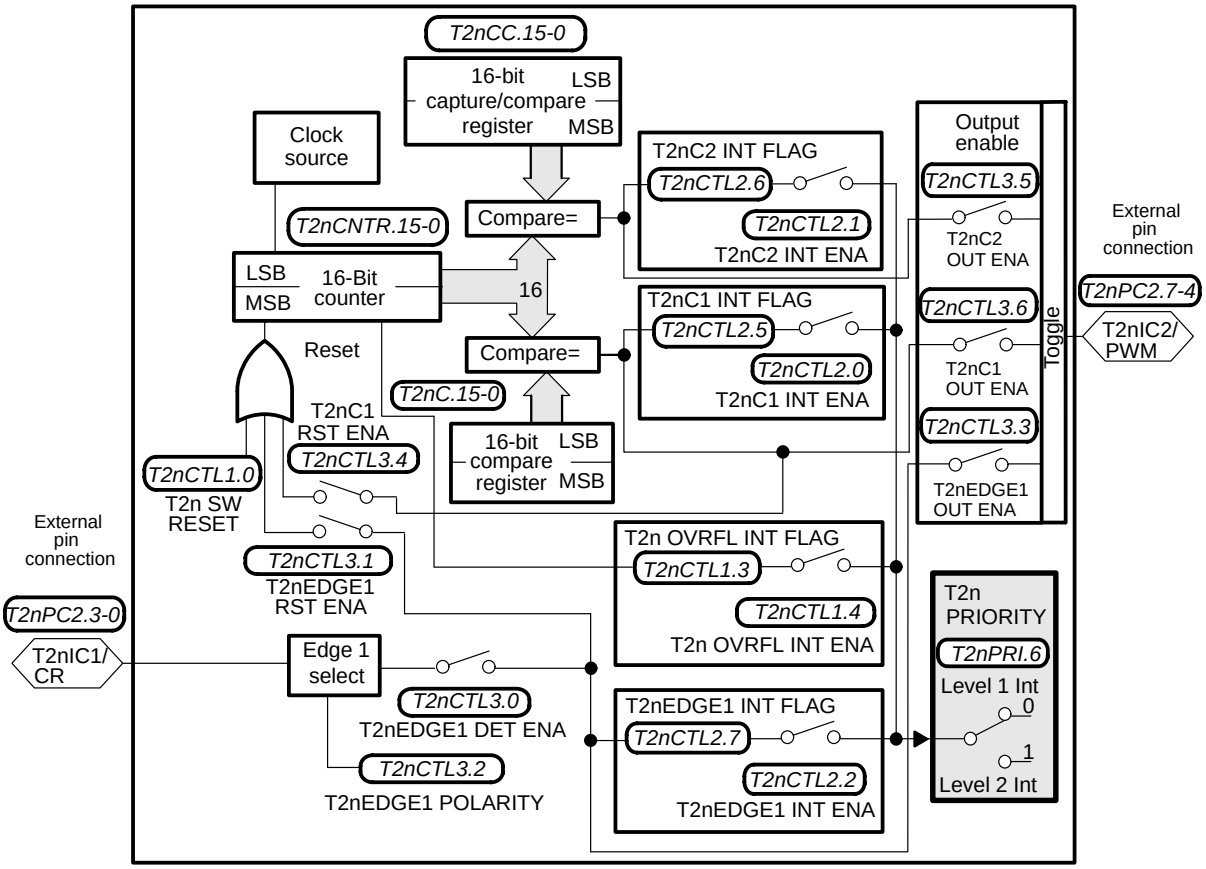
C.3 Timer 2n (T2n) Module

The T2n module has two operating modes:

- **Dual compare mode** is shown in Figure C–7
- **Dual capture mode** is shown in Figure C–8

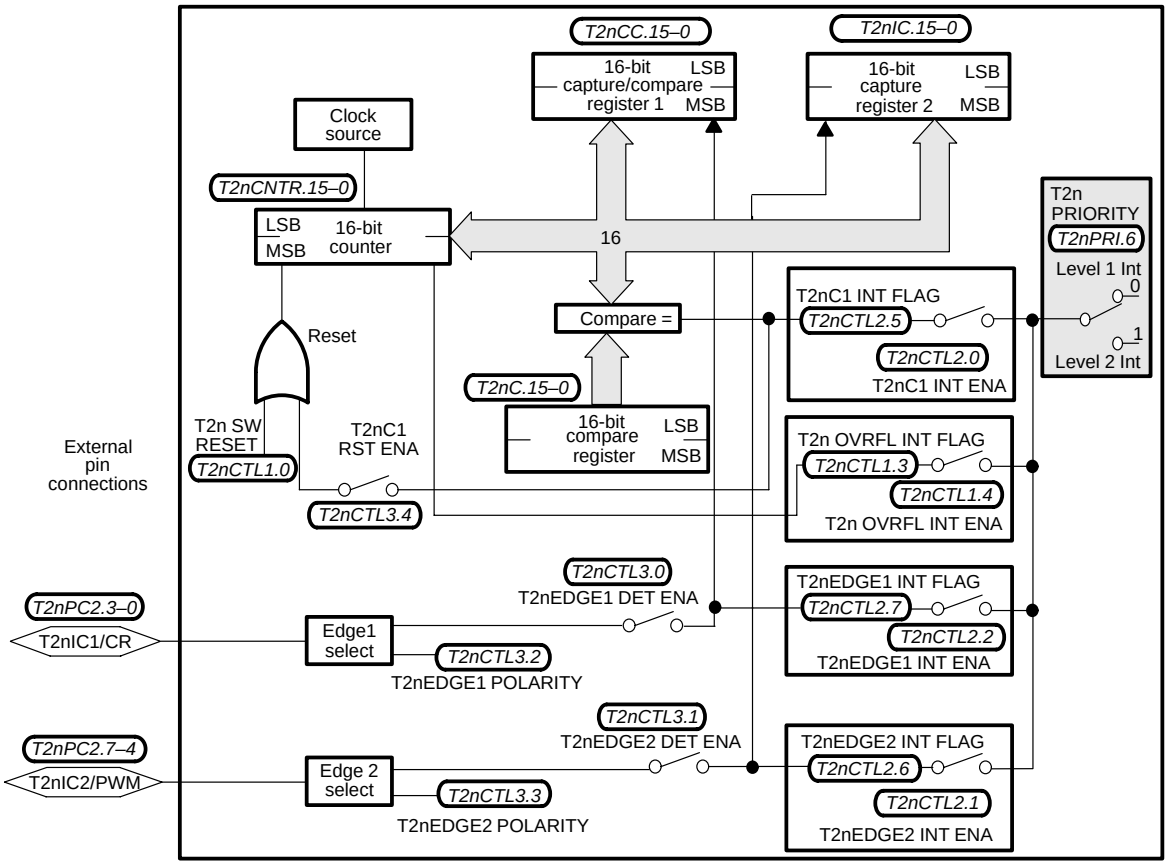
For a summary of the T2n module control registers and bits, refer to Section B.9, page B-10.

Figure C–7. T2n: Dual Compare Mode Block Diagram



Note: The annotations on the diagram identify the register and the bit (s) in the peripheral frame. For example, the actual address of T2nCTL2.0 is 106Bh (n=A) or 108Bh (n=B), bit 0, in the T2nCTL2 register.

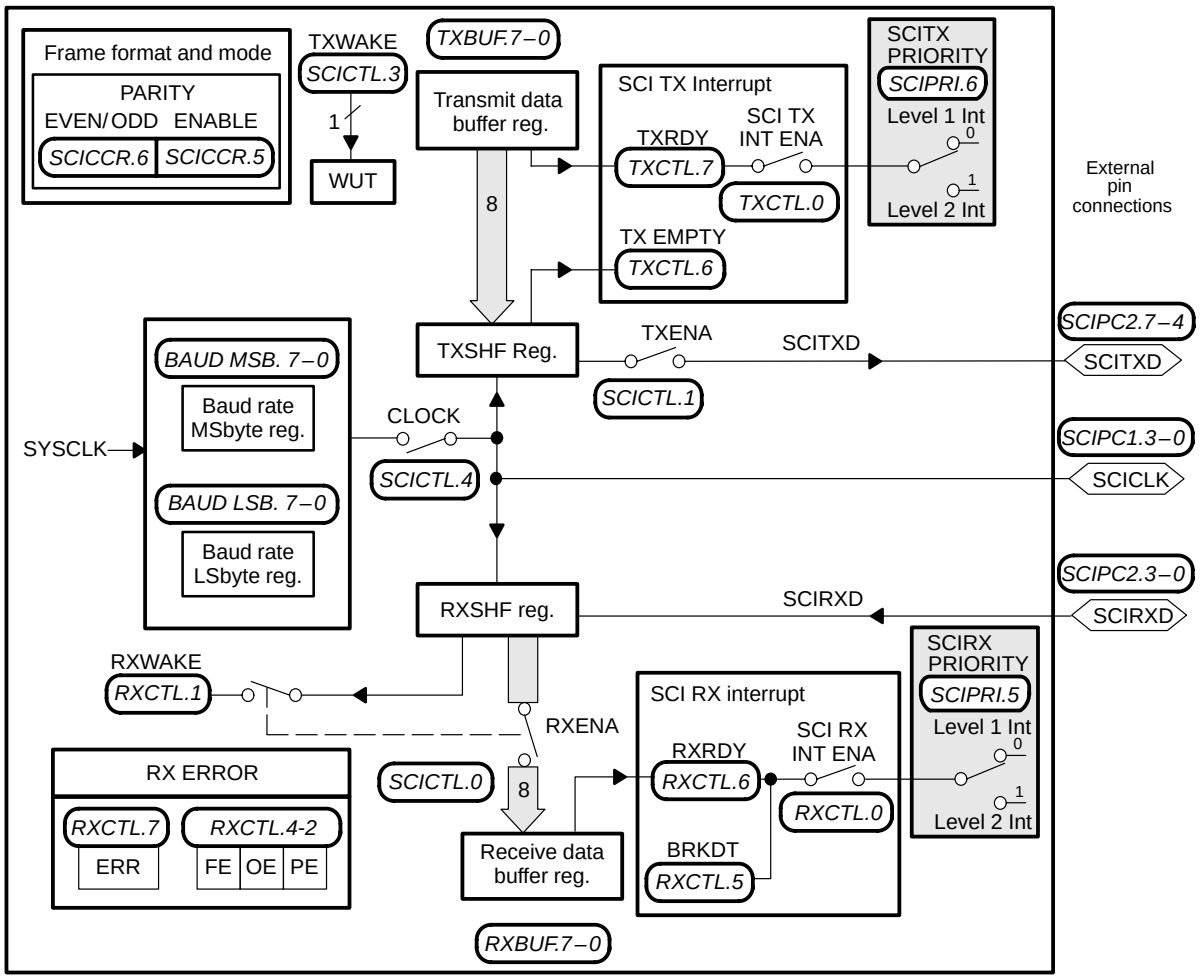
Figure C–8. T2n: Dual Capture Mode Block Diagram



C.4 Serial Communications Interface 1

The block diagram for the SCI1 module is shown in Figure C–9. For a summary of the SCI1 control registers and bits, refer to Section B.7, page B-8.

Figure C–9. SCI1 Block Diagram - Three Pin Configuration

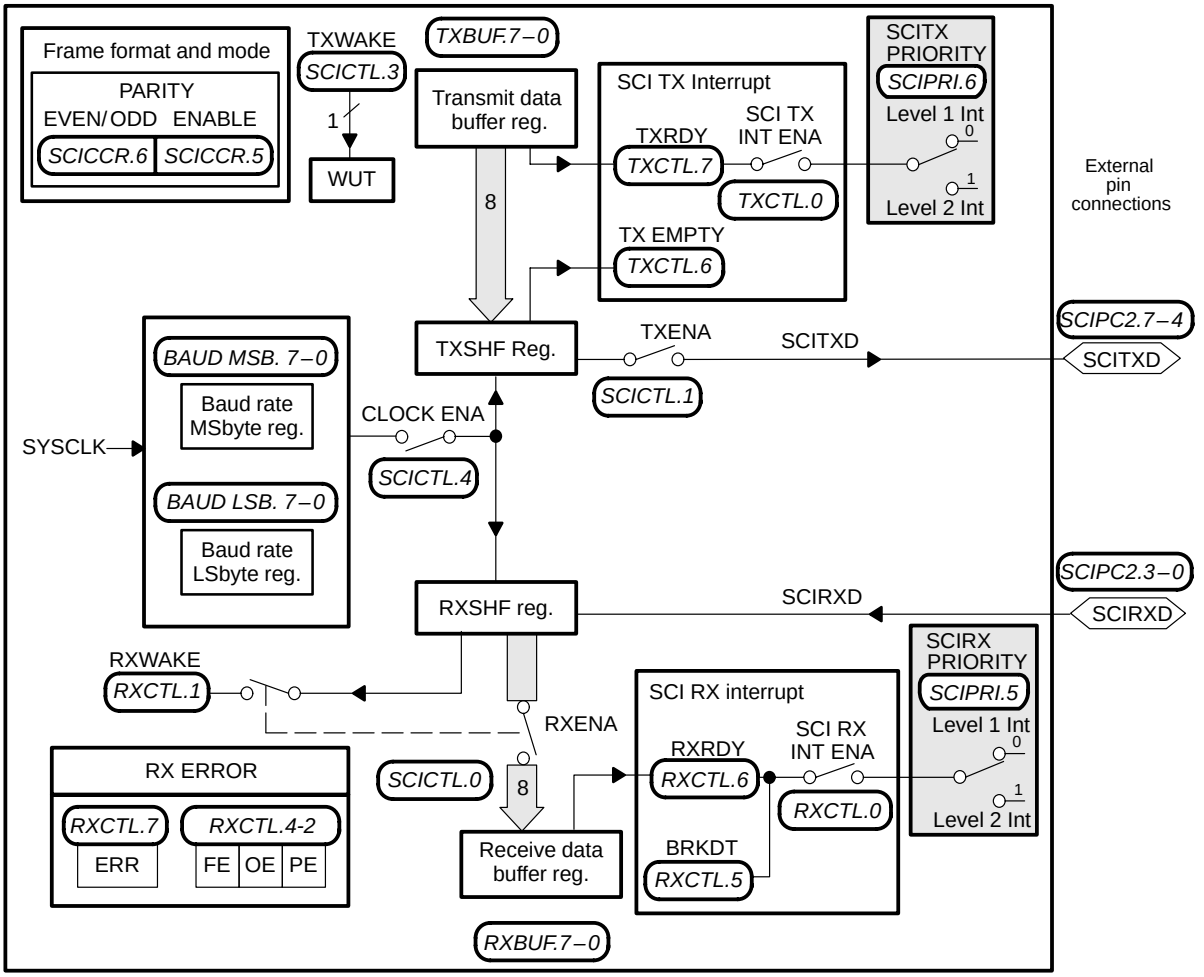


Note: SCI1 registers are described in detail in Section 9.8.1 beginning on page 9-22.

C.5 Serial Communications Interface 2

The block diagram for the SCI2 module is shown in Figure C–10. For a summary of the SCI2 control register and bits, refer to section B.8, on page B-9.

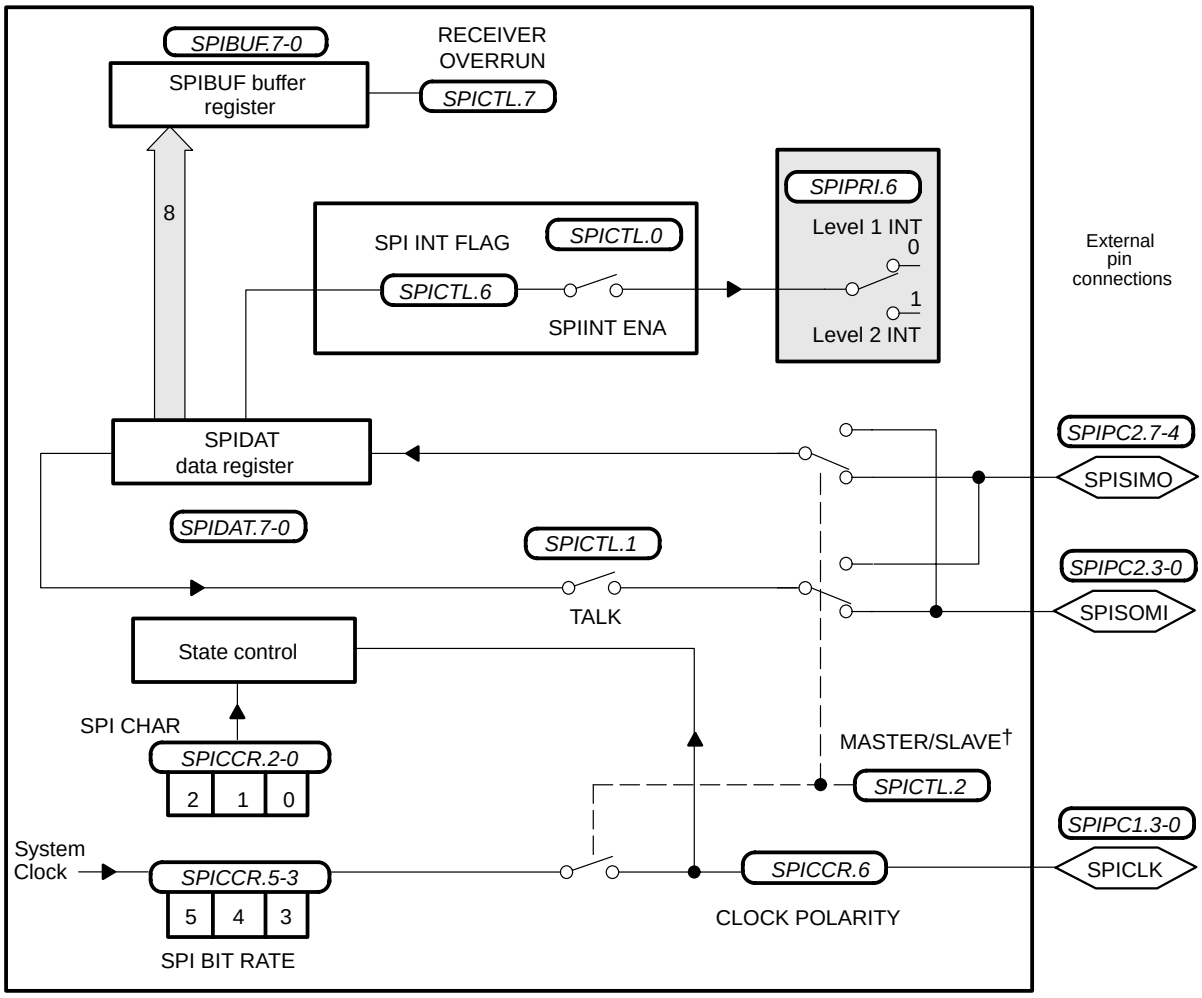
Figure C–10. SCI2 Block Diagram



C.6 Serial Peripheral Interface

The block diagram for the SPI module is shown in Figure C–11. For a summary of the SPI control registers and bits, refer to Section B.4, page B-5.

Figure C–11. SPI Block Diagram

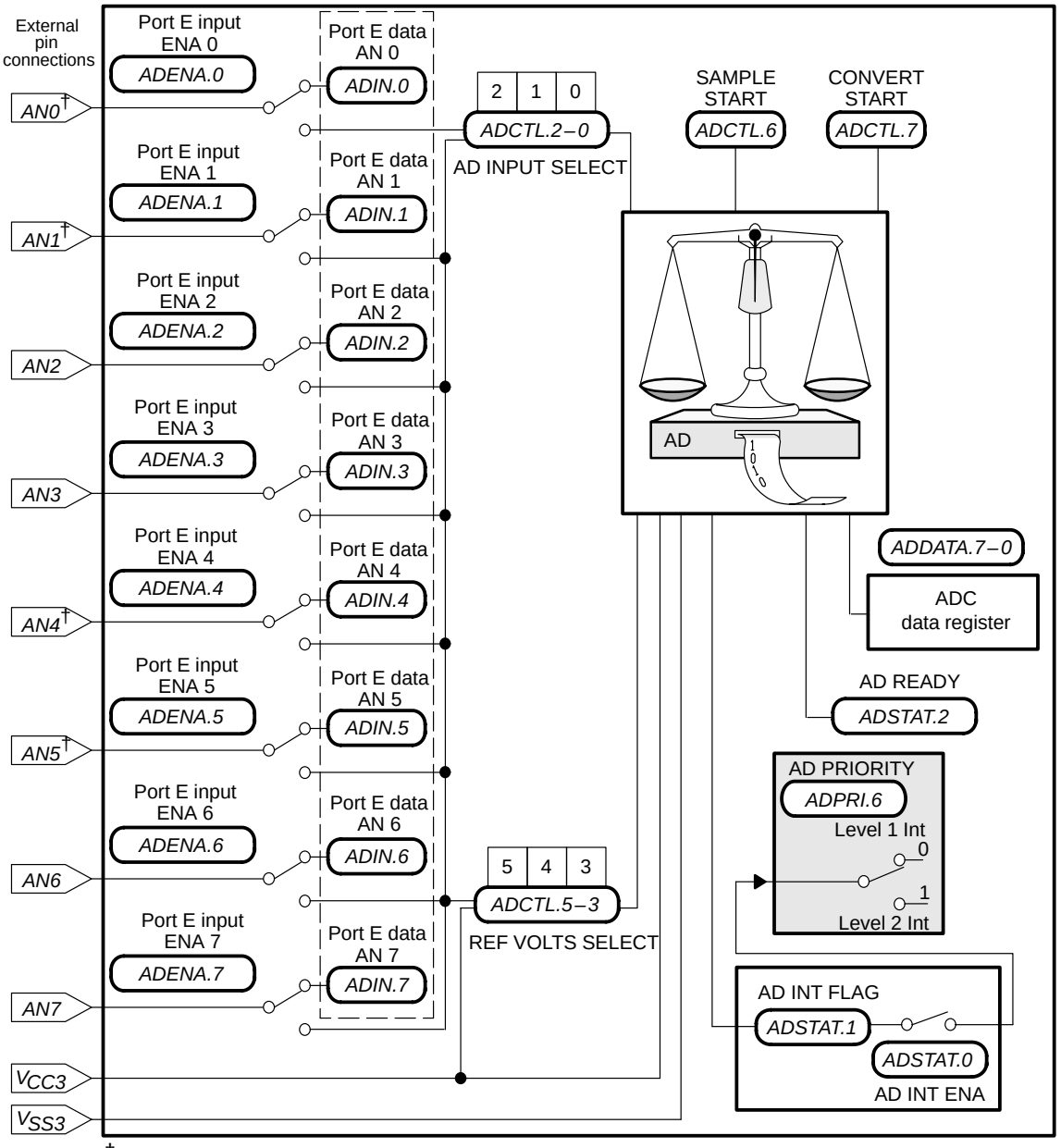


† The diagram is shown in slave mode.

C.7 Analog-to-Digital Converter 1

The block diagram for the ADC1 module is shown in Figure C–12. For a summary of the ADC1 registers and bits, refer to Section B.10, on page B-11.

Figure C–12. ADC1 Block Diagram

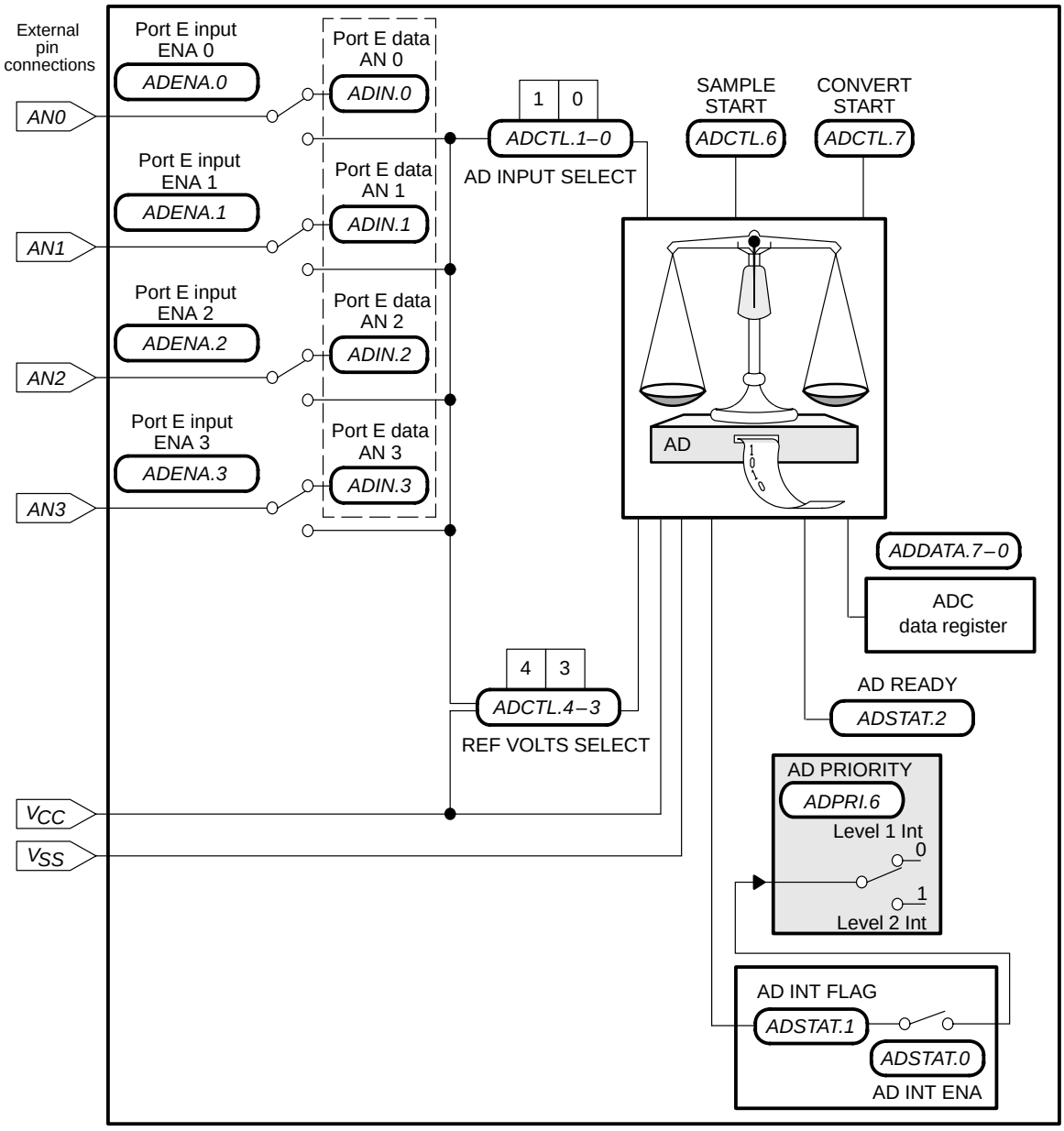


† These pins are not implemented in the TMS370Cx4x 40-pin device.

C.8 Analog-to-Digital Converter 2

The block diagram for the ADC2 module is shown in Figure C–13. For a summary of the ADC2 control registers and bits, refer to Section B.11, page B-12.

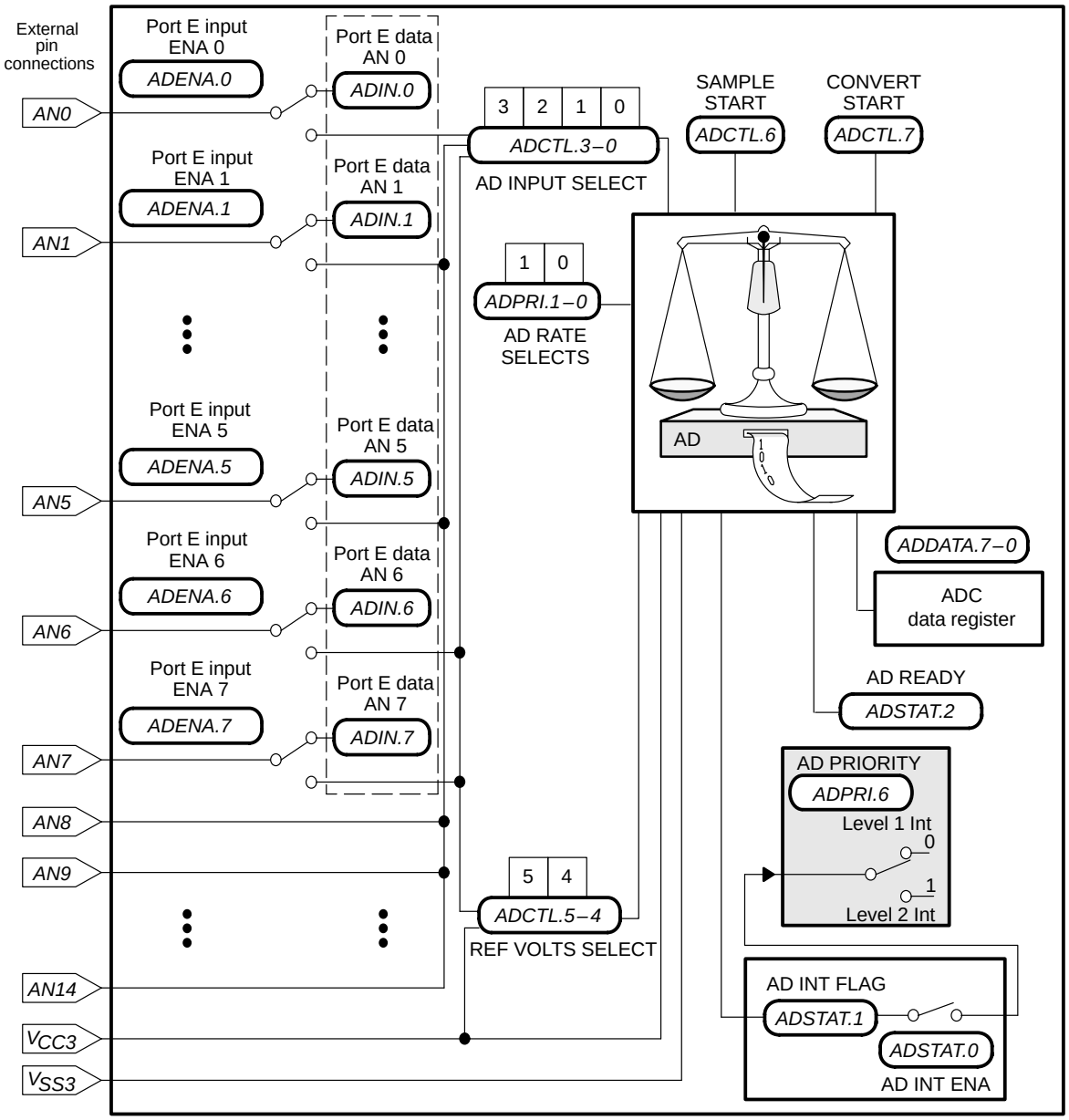
Figure C–13. ADC2 Block Diagram



C.9 Analog-to-Digital Converter 3

The block diagram for the ADC3 module is shown in Figure C–14. For a summary of the ADC3 control registers and bits, refer to B.12, page B-13.

Figure C–14. ADC3 Block Diagram



Appendix D

ASCII Character Set

The TMS370 assembler recognizes the following ASCII characters.

ASCII Character Set (7-Bit Code)

Least significant nibble (LSN)	Most significant nibble (MSN)							
	0	1	2	3	4	5	6	7
	(0)	(16)	(32)	(48)	(64)	(80)	(96)	(112)
0	NUL	DLE	SP	0	@	P	’	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	”	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	’	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Note: To obtain the decimal value, concatenate the decimal MSN to the decimal LSN. For example, a capital A = 41_{16}.

Appendix E

Opcode/Instruction Cross-Reference

Table E–1 provides an opcode-to-instruction cross-reference of all 73 mnemonics and 274 opcodes of the TMS370 instruction set. To check the instruction of a known opcode, locate the left (high) digit across the top of the table, then find the right (low) digit along the side of the table. The intersection contains the instruction mnemonic, operands, and byte/cycle peculiar to that opcode. Some opcodes, such as B0, are shared by two instructions, in which case both mnemonics are shown along with the byte/cycles count.

Table E–1. TMS370 Family Opcode/Instruction Map†

LSN \ MSN	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	JMP ra 2/7							INCW #n,Rd 3/11	MOV Ps,A 2/8			CLRC / TST A 1/9	MOV A,B 1/9	MOV A,Rd 2/7	TRAP 15 1/14	LDST n 2/6
1	JN ra 2/5		MOV A,Pd 2/8			MOV B,Pd 2/8		MOV Rs,Pd 3/10		MOV Ps,B 2/7				MOV B,Rd 2/7	TRAP 14 1/14	MOV *n[SP],A 2/7
2	JZ ra 2/5	MOV Rs,A 2/7	MOV #n,A 2/6	MOV Rs,B 2/7	MOV Rs,Rd 3/9	MOV #n,B 2/6	MOV B,A 1/8	MOV #n,Rd 3/8			MOV Ps,Rd 3/10	DEC A 1/8	DEC B 1/8	DEC Rn 2/6	TRAP 13 1/14	MOV *A,n[SP] 2/7
3	JC ra 2/5	AND Rs,A 2/7	AND #n,A 2/6	AND Rs,B 2/7	AND Rs,Rd 3/9	AND #n,B 2/6	AND B,A 1/8	AND #n,Rd 3/8	AND A,Pd 2/9	AND B,Pd 2/9	AND #n,Pd 3/10	INC A 1/8	INC B 1/8	INC Rn 2/6	TRAP 12 1/14	CMP *n[SP],A 2/8
4	JP ra 2/5	OR Rs,A 2/7	OR #n,A 2/6	OR Rs,B 2/7	OR Rs,Rd 3/9	OR #n,B 2/6	OR B,A 1/8	OR #n,Rd 3/8	OR A,Pd 2/9	OR B,Pd 2/9	OR #n,Pd 3/10	INV A 1/8	INV B 1/8	INV Rn 2/6	TRAP 11 1/14	extend inst,2 opcodes
5	JPZ ra 2/5	XOR Rs,A 2/7	XOR #n,A 2/6	XOR Rs,B 2/7	XOR Rs,Rd 3/9	XOR #n,B 2/6	XOR B,A 1/8	XOR #n,Rd 3/8	XOR A,Pd 2/9	XOR B,Pd 2/9	XOR #n,Pd 3/10	CLR A 1/8	CLR B 1/8	CLR Rd 2/6	TRAP 10 1/14	
6	JNZ ra 2/5	BTJO Rs,A,ra 3/9	BTJO #n,A,ra 3/8	BTJO Rs,B,ra 3/9	BTJO Rs,Rd,ra 4/11	BTJO #n,B,ra 3/8	BTJO B,A,ra 2/10	BTJO #n,Rd,ra 4/10	BTJO A,Pd,ra 3/11	BTJO B,Pd,ra 3/10	BTJO #n,Pd,ra 4/11	XCHB A 1/10	XCHB A / TST B 1/10	XCHB Rn 2/8	TRAP 9 1/14	IDLE 1/6
7	JNC ra 2/5	BTJZ Rs,A,ra 3/9	BTJZ #n,A,ra 3/8	BTJZ Rs,B,ra 3/9	BTJZ Rs,Rd,ra 4/11	BTJZ #n,B,ra 3/8	BTJZ B,A,ra 2/10	BTJZ #n,Rd,ra 4/10	BTJZ A,Pd,ra 3/10	BTJZ B,Pd,ra 3/10	BTJZ #n,Pd,ra 4/11	SWAP A 1/11	SWAP B 1/11	SWAP Rn 2/9	TRAP 8 1/14	MOV #n,Pd 3/10
8	JV ra 2/5	ADD Rs,A 2/7	ADD #n,A 2/6	ADD Rs,B 2/7	ADD Rs,Rd 3/9	ADD #n,B 2/6	ADD B,A 1/8	ADD #n,Rd 3/8	MOVW #16,Rd 4/13	MOVW Rs,Rd 3/12	MOVW #16[B],Rd 4/15	PUSH A 1/9	PUSH B 1/9	PUSH Rs 2/7	TRAP 7 1/14	SETC 1/7
9	JL ra 2/5	ADC Rs,A 2/7	ADC #n,A 2/6	ADC Rs,B 2/7	ADC Rs,Rd 3/9	ADC #n,B 2/6	ADC B,A 1/8	ADC #n,Rd 3/8	JMPL lab 3/9	JMPL *Rd 2/8	JMPL *lab[B] 3/11	POP A 1/9	POP B 1/9	POP Rd 2/7	TRAP 6 1/14	RTS 1/9
A	JLE ra 2/5	SUB Rs,A 2/7	SUB #n,A 2/6	SUB Rs,B 2/7	SUB Rs,Rd 3/9	SUB #n,B 2/6	SUB B,A 1/8	SUB #n,Rd 3/8	MOV &lab,A 3/10	MOV *Rs,A 2/9	MOV *lab[B],A 3/12	DJNZ A,ra 2/10	DJNZ B,ra 2/10	DJNZ Rn,ra 3/8	TRAP 5 1/14	RTI 1/12
B	JHS ra 2/5	SBB Rs,A 2/7	SBB #n,A 2/6	SBB Rs,B 2/7	SBB Rs,Rd 3/9	SBB #n,B 2/6	SBB B,A 1/8	SBB #n,Rd 3/8	MOV A,&lab 3/10	MOV A,*Rp 2/9	MOV A,*lab[B] 3/12	COMPL A 1/8	COMPL B 1/8	COMPL Rn 2/6	TRAP 4 1/14	PUSH ST 1/8

† All conditional jumps (opcodes 01–0F), BTJO, BTJZ, and DJNZ instructions use two additional cycles if the branch is taken. The BTJO, BTJZ, and DJNZ instructions have a relative address as the last operand. Abbreviations are listed on the next page of this table.

Table E–1. TMS370 Family Opcode/Instruction Map† (Concluded)

LSN \ MSN	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C	JNV ra 2/5	MPY Rs,A 2/46	MPY #n,A 2/45	MPY Rs,B 2/46	MPY Rs,Rd 3/48	MPY #n,B 2/45	MPY B,A 1/47	MPY #n,Rs 3/47	BR lab 3/9	BR *Rd 2/8	BR *lab[B] 3/11	RR A 1/8	RR B 1/8	RR Rn 2/6	TRAP 3 1/14	POP ST 1/8
D	JGE ra 2/5	CMP Rs,A 2/7	CMP #n,A 2/6	CMP Rs,B 2/7	CMP Rs,Rd 3/9	CMP #n,B 2/6	CMP B,A 1/8	CMP #n,Rd 3/8	CMP &lab,A 3/11	CMP *Rs,A 2/10	CMP *lab[B],A 3/13	RRC A 1/8	RRC B 1/8	RRC Rn 2/6	TRAP 2 1/14	LDSP 1/7
E	JG ra 2/5	DAC Rs,A 2/9	DAC #n,A 2/8	DAC Rs,B 2/9	DAC Rs,Rd 3/11	DAC #n,B 2/8	DAC B,A 1/10	DAC #n,Rd 3/10	CALL lab 3/13	CALL *Rd 2/12	CALL *lab[B)] 3/15	RL A 1/8	RL B 1/8	RL Rn 2/6	TRAP 1 1/14	STSP 1/8
F	JLO ra 2/5	DSB Rs,A 2/9	DSB #n,A 2/8	DSB Rs,B 2/9	DSB Rs,Rd 3/11	DSB #n,B 2/8	DSB B,A 1/10	DSB #n,Rd 3/10	CALLR lab 3/15	CALLR *Rd 2/14	CALLR *lab[B] 3/17	RLC A 1/8	RLC B 1/8	RLC Rn 2/6	TRAP 0 1/14	NOP 1/7

Second byte of two-byte instructions (F4xx):

F4	8	MOVW *n[Rp] 4/15	DIV Rn,A 3/14-63
F4	9	JMPL *n[Rp] 4/16	
F4	A	MOV *n[Rp],A 4/17	
F4	B	MOV A,*n[Rp] 4/16	
F4	C	BR *n[Rp] 4/17	
F4	D	CMP *n[Rp],A 4/18	
F4	E	CALL *n[Rp] 4/20	
F4	F	CALLR *n[Rp] 4/22	

* = Indirect addressing operand prefix
& = Direct addressing operand prefix
# = Immediate operand
#16 = Immediate 16-bit number
lab = 16-bit label
n = Immediate 8-bit number
Pd = Peripheral register containing destination type
Pn = Peripheral register
Ps = Peripheral register containing source byte
ra = Relative address
Rd = Register containing destination type
Rn = Register file
Rp = Register pair
Rpd = Destination register pair
Rps = Source register pair
Rs = Register containing source byte

† All conditional jumps (opcodes 01–0F), BTJO, BTJZ, and DJNZ instructions use two additional cycles if the branch is taken. The BTJO, BTJZ, and DJNZ instructions have a relative address as the last operand.

Appendix F

Instruction/Opcode Cross-Reference and Bus Activity Table

This appendix contains both an instruction-to-opcode cross-reference and an instruction bus activity table. The bus activity table specifies the cycle-by-cycle actions of a given instruction.

Topic **Page**

F.1 Instruction/Opcode Cross-Reference

Table F–1 provides an instruction-to-opcode cross-reference of all 73 mnemonics and 274 opcodes of the TMS370 instruction set. The columns are grouped according to addressing modes.

Table F–1. TMS370 Family Instruction/Opcode Set

	Data Manipulation																				Program Flow								Other
	A	B	Rd	A,B	B,A	Rs, A	#n, A	Rs, B	#n, B	Rs, Rd	#n, Rd	A, Rd	B, Rd	A, Pd	Ps, A	B, Pd	Ps, B	#n, Pd	†	‡	§	¶	#	‖	☆	□	◊	⊞	¥	♣	Φ
ADC					69	19	29	39	59	49	79																				
ADD					68	18	28	38	58	48	78																				
AND					63	13	23	33	53	43	73			83		93		A3													
BR																							8C	AC	9C	EC					
BTJO					66	16	26	36	56	46	76			86		A6		96													
BTJZ					67	17	27	37	57	47	77			87		A7		97													
CALL																							8E	9E	AE	EE					
CALLR																											8F	9F	AF	EF	
CLR	B5	C5	D5																												
CLRC																															B0
CMP					6D	1D	2D	3D	5D	4D	7D								8D	AD	9D	ED									
CMPBIT																															75,A5
COMPL	BB	CB	DB																												
DAC					6E	1E	2E	3E	5E	4E	7E																				
DEC	B2	C2	D2																												
DINT																															F0 00
DIV																															F4 F8
DJNZ	BA	CA	DA																												
DSB					6F	1F	2F	3F	5F	4F	7F																				
EINT																															F0 0C
EINTH																															F0 04
EINTL																															F0 08
IDLE																															F6
INC	B3	C3	D3																												
INV	B4	C4	D4																												
JBIT0																															77,A7

†Direct {(label) → (A)}
‡Indexed {(label + (B)) → (A)}
§Indirect {(Rn - 1:Rn) → (A)}
¶Offset indirect {(b + (Rn - 1:Rn)) → (A)}
#Absolute direct {label → (PC)}
‖Absolute indexed {label + (B) → (PC)}
☆Absolute indirect {(Rn - 1:Rn) → (PC)}
□Absolute offset indirect {b + (Rn - 1:Rn) → (PC)}
◊Rel. direct {PCN + label → (PC)}
⊞Rel. indexed {PCN + label + (B) → (A)}
¥Rel. indirect {PCN + (Rn - 1:Rn) → (PC)}
♣Rel. off. indirect {PCN + b + (Rn - 1:Rn) → (PC)}

ΦUnless otherwise indicated, includes instructions that do not qualify as a data manipulation or program flow addressing mode.
°The MOV instruction also includes the following options and their opcodes: Rn,Pn {71}; Pn,Rn {A2}; A,label(B) {AB}; A,n(SP) {F2}; A,n(Rn) {F4 EB}; label,A {8A}; n(SP),A {F1}
δThe SBIT0 instruction consists of the following options and their opcodes: Rname {73}; Pname {A3}
∞The SBIT1 instruction consists of the following options and their opcodes: Rname {74}; Pname {A4}
∧The TRAP instruction consists of 16 options using operands 0 through 15 with opcodes EFh through E0h, respectively

Table F–1. TMS370 Family Instruction/Opcode Set (Continued)

	Data Manipulation																					Program Flow								Other
	A	B	Rd	A,B	B,A	Rs, A	#n, A	Rs, B	#n, B	Rs, Rd	#n, Rd	A, Rd	B, Rd	A, Pd	Ps, A	B, Pd	Ps, B	#n, Pd	†	‡	§	¶	#	‖	☆	□	◊	⊞	¥	♣	Φ
JBIT1																															76,A6
JMP																															00
JMPL																											89	A9	99	E9	
JC																															03
JEQ/JZ																															02
JG																															0E
JGE																															0D
JHS																															0B
JL																															09
JLE																															0A
JLO																															0F
JN																															01
JNC																															07
JNE/ JNZ																															06
JNV																															0C
JP																															04
JPZ																															05
JV																															08
LDSP																															FD
LDST																															F0
MOV				C0	62	12	22	32	52	42	72	D0	D1	21	80	51	91	F7	8B	AA	9A	EA									°
MOVW																			88	A8	98	E8									
MPY					6C	1C	2C	3C	5C	4C	7C																				
NOP																															FF
OR					64	14	24	34	54	44	74			84		94		A4													
POP	B9	C9	D9																												FC
PUSH	B8	C8	D8																												FB

†Direct {(label) → (A)}
‡Indexed {(label + (B)) → (A)}
§Indirect {(Rn - 1:Rn) → (A)}
¶Offset indirect {(b + (Rn - 1:Rn)) → (A)}
#Absolute direct {label → (PC)}
‖Absolute indexed {label + (B) → (PC)}
☆Absolute indirect {(Rn - 1:Rn) → (PC)}
□Absolute offset indirect {b + (Rn - 1:Rn) → (PC)}
◊Rel. direct {PCN + label → (PC)}
⊞Rel. indexed {PCN + label + (B) → (A)}
¥Rel. indirect {PCN + (Rn - 1:Rn) → (PC)}
♣Rel. off. indirect {PCN + b + (Rn - 1:Rn) → (PC)}

ΦUnless otherwise indicated, includes instructions that do not qualify as a data manipulation or program flow addressing mode.
°The MOV instruction also includes the following options and their opcodes: Rn,Pn {71}; Pn,Rn {A2}; A,label(B) {AB}; A,n(SP) {F2}; A,n(Rn) {F4 EB}; label,A {8A}; n(SP),A {F1}
δThe SBIT0 instruction consists of the following options and their opcodes: Rname {73}; Pname {A3}
∞The SBIT1 instruction consists of the following options and their opcodes: Rname {74}; Pname {A4}
∧The TRAP instruction consists of 16 options using operands 0 through 15 with opcodes EFh through E0h, respectively

Table F–1. TMS370 Family Instruction/Opcode Set (Continued)

	Data Manipulation																					Program Flow								Other
	A	B	Rd	A,B	B,A	Rs, A	#n, A	Rs, B	#n, B	Rs, Rd	#n, Rd	A, Rd	B, Rd	A, Pd	Ps, A	B, Pd	Ps, B	#n, Pd	†	‡	§	¶	#	‖	☆	□	◊	⊞	¥	♣	Φ
RL	BE	CE	DE																												
RLC	BF	CF	DF																												
RR	BC	CC	DC																												
RRC	BD	CD	DD																												
RTI																															FA
RTS																															F9
SBB					6B	1B	2B	3B	5B	4B	7B																				
SBIT0																															δ
SBIT1																															∞
SETC																															F8
STSP																															FE
SUB					6A	1A	2A	3A	5A	4A	7A																				
SWAP	B7	C7	D7																												
TRAP																															^
TST	B0	C6																													
XCHB	B6	C6	D6																												
XOR					65	15	25	35	55	45	75			85		95		A5													

†Direct {(label) → (A)}
‡Indexed {(label + (B)) → (A)}
§Indirect {(Rn - 1:Rn) → (A)}
¶Offset indirect {(b + (Rn - 1:Rn)) → (A)}
#Absolute direct {label → (PC)}
‖Absolute indexed {label + (B) → (PC)}
☆Absolute indirect {(Rn - 1:Rn) → (PC)}
□Absolute offset indirect {b + (Rn - 1:Rn) → (PC)}
◊Rel. direct {PCN + label → (PC)}
⊞Rel. indexed {PCN + label + (B) → (A)}
¥Rel. indirect {PCN + (Rn - 1:Rn) → (PC)}
♣Rel. off. indirect {PCN + b + (Rn - 1:Rn) → (PC)}

ΦUnless otherwise indicated, includes instructions that do not qualify as a data manipulation or program flow addressing mode.
°The MOV instruction also includes the following options and their opcodes: Rn,Pn {71}; Pn,Rn {A2}; A,label(B) {AB}; A,n(SP) {F2}; A,n(Rn) {F4 EB}; label,A {8A}; n(SP),A {F1}
δThe SBIT0 instruction consists of the following options and their opcodes: Rname {73}; Pname {A3}
∞The SBIT1 instruction consists of the following options and their opcodes: Rname {74}; Pname {A4}
^The TRAP instruction consists of 16 options using operands 0 through 15 with opcodes EFh through E0h, respectively

F.2 Bus Activity Table

The TMS370 family employs a microcoded instruction set. Each instruction is broken down into microcode states, and a miniprogram is executed for the instruction using these states. Each microcode state lasts for one internal clock cycle and includes provisions for conditional jumps, branching, and control of internal CPU operations. To increase efficiency and variety, each instruction can use sections of common microcode states (similar to subroutines).

Table F–2 and Table F–3 explain the abbreviations used, while Table F–4 provides a cycle-by-cycle accounting for each of the 274 instruction and operand combinations for the TMS370 family microcontrollers. Each line in Table F–4 consists of the opcode value, the mnemonic and operands, and a summary of all the cycles.

Table F–4 gives the minimum time for each instruction by showing the number of internal states. Each state lasts for four CLKIN or crystal periods, so each state represents 200 ns for a crystal running at 5 MHz SYSCLK. This time may increase if the application uses the autowait mode, peripheral autowait mode, or wait pin. **The wait modes affect only the “<<” cycles that access external memory.**

The instructions are typically expressed in this format:

```
OPCODE  INSTR  O1[,O2][,O3]
```

Operands are always read in increasing address order. This distinction is especially important for the MOV Rs,Pd and the MOV Ps,Rd instructions, in which the operands are in reversed address order. A and B are implied operands and do not require additional bytes.

The *operand* symbols used in Table F–4 are as follows:

*	=	Indirect addressing operand prefix
&	=	Direct addressing operand prefix
#	=	Immediate operand
#16n	=	Immediate 16-bit number
lab	=	16-bit label
n	=	Immediate 8-bit number
Pd	=	Peripheral register containing destination byte
Pn	=	Peripheral register
Ps	=	Peripheral register containing source byte
ra	=	Relative address (8-bit immediate offset value)
Rd	=	Register containing destination byte
Rn	=	Register file
Rp	=	Register pair
Rpd	=	Destination register pair
Rps	=	Source register pair
Rs	=	Register containing source byte

The two-letter abbreviations used in each line are divided into two groups.

- The first group (shown in Table F–2) generates valid external bus cycles when accessing external memory.
- The second group (shown in Table F–3) operates only internally and does not generate valid bus cycles, because internal memory accesses do not give valid external bus cycles.

Table F–2. Possible Bus Cycles

	Name	Type	Description
OC	Opcode read	Long	Opcode fetch, $\overline{\text{OCF}}$, active during this cycle.
PO	Pseudo-operand read	Long	Operand is read then discarded, and PC does not advance. Operand usually becomes an opcode on next instruction cycle.
O*n*	Operand *n* read	Long	Read instruction's operands (*n*=operand order in instruction).
PF	Prefix byte		First byte of a two-byte opcode. $\overline{\text{OCF}}$ is active during this cycle and not during next OC cycle. Always 0F4h.
Pr	Peripheral read	Long	Read of 1000h–10FFh. The external reads are affected by PF auto-wait.
Pw	Peripheral write	Long	Write to 1000h–10FFh. The external writes are affected by PF auto-wait.
Lr	Long read	Long	General long read, range of 0–0FFFFh.
Lw	Long write	Long	General long write, range of 0–0FFFFh.
Vr	Vector read	Long	Reads vectors in trap table (7FC0h–7FDFh).
<<	Continuation	Long	Finish memory access, nominal one cycle. If external memory is accessed using autowait, PF autowait, or external wait signals, it could require two or more cycles. Only cycle type affected by wait modes.

Table F–3. Internal Cycles

	Name	Type	Description
IC	Internal cycle	Other	Indeterminate internal operation
RJ	Relative jump	Other	Add these if jump is taken; takes two cycles, so they always come in pairs.
SH	Stack push	Other	(SP)+1 –> (SP); n –> ((SP))
SP	Stack pop	Other	((SP)) –> n ; (SP)–1 –> (SP)
Sr	Stack read	Other	((SP)) usually dummy cycle
Ar	Register A read	Register	1-cycle register accesses
Br	Register B read	Register	1-cycle register accesses
Rr	Register read	Register	1-cycle register accesses
Aw	Register A write	Register	1-cycle register accesses
Bw	Register B write	Register	1-cycle register accesses
Rw	Register write	Register	1-cycle register accesses

Table F–4. Bus Activity Table

Code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
29	ADC	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
59	ADC	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
79	ADC	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
69	ADC	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
19	ADC	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
39	ADC	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
49	ADC	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
28	ADD	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
58	ADD	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
78	ADD	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw							.					.		
68	ADD	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
18	ADD	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
38	ADD	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
48	ADD	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
23	AND	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
53	AND	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
A3	AND	#n,Pd	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					.					.		
73	AND	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
83	AND	A,Pd	OC	<<	O1	<<	Ar	Pr	<<	Pw	<<	.					.					.		
63	AND	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
93	AND	B,Pd	OC	<<	O1	<<	Br	Pr	<<	Pw	<<	.					.					.		
13	AND	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
33	AND	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
43	AND	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
9C	BR	*Rp	OC	<<	O1	<<	Rr	Rr	IC	IC		.					.					.		
8C	BR	lab	OC	<<	O1	<<	O2	<<	IC	IC	IC	.					.					.		
AC	BR	*lab[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	IC				.					.		
F4 EC	BR	*n[Rp]	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	IC	IC				.		
26	BTJO	#n,A,#ra	OC	<<	O1	<<	Ar	IC	O2	<<	RJ	RJ					.					.		
56	BTJO	#n,B,#ra	OC	<<	O1	<<	Br	IC	O2	<<	RJ	RJ					.					.		
A6	BTJO	#n,Pd,#ra	OC	<<	O1	<<	O2	<<	Pr	<<	IC	O3	<<	RJ	RJ		.					.		
76	BTJO	#n,Rd,#ra	OC	<<	O1	<<	O2	<<	Rr	IC	O3	<<	RJ	RJ			.					.		
86	BTJO	A,Pd,#ra	OC	<<	O1	<<	Ar	Pr	<<	IC	O2	<<	RJ	RJ			.					.		
66	BTJO	B,A,#ra	OC	<<	PO	<<	Br	IC	Ar	IC	O1	<<	RJ	RJ			.					.		
96	BTJO	B,Pd,#ra	OC	<<	O1	<<	Br	Pr	<<	IC	O2	<<	RJ	RJ			.					.		
16	BTJO	Rs,A,#ra	OC	<<	O1	<<	Rr	Ar	IC	O2	<<	RJ	RJ				.					.		
36	BTJO	Rs,B,#ra	OC	<<	O1	<<	Rr	Br	IC	O2	<<	RJ	RJ				.					.		
46	BTJO	Rs,Rd,#ra	OC	<<	O1	<<	Rr	O2	<<	Rr	IC	O3	<<	RJ	RJ		.					.		
27	BTJZ	#n,A,#ra	OC	<<	O1	<<	Ar	IC	O2	<<	RJ	RJ					.					.		
57	BTJZ	#n,B,#ra	OC	<<	O1	<<	Br	IC	O2	<<	RJ	RJ					.					.		
A7	BTJZ	#n,Pd,#ra	OC	<<	O1	<<	O2	<<	Pr	<<	IC	O3	<<	RJ	RJ		.					.		
77	BTJZ	#n,Rd,#ra	OC	<<	O1	<<	O2	<<	Rr	IC	O3	<<	RJ	RJ			.					.		
87	BTJZ	A,Pd,#ra	OC	<<	O1	<<	Ar	Pr	<<	IC	O2	<<	RJ	RJ			.					.		
67	BTJZ	B,A,#ra	OC	<<	PO	<<	Br	IC	Ar	IC	O1	<<	RJ	RJ			.					.		
97	BTJZ	B,Pd,#ra	OC	<<	O1	<<	Br	Pr	<<	IC	O2	<<	RJ	RJ			.					.		
17	BTJZ	Rs,A,#ra	OC	<<	O1	<<	Rr	Ar	IC	O2	<<	RJ	RJ				.					.		
37	BTJZ	Rs,B,#ra	OC	<<	O1	<<	Rr	Br	IC	O2	<<	RJ	RJ				.					.		
47	BTJZ	Rs,Rd,#ra	OC	<<	O1	<<	Rr	O2	<<	Rr	IC	O3	<<	RJ	RJ		.					.		
9E	CALL	*Rp	OC	<<	O1	<<	Rr	Rr	IC	SH	IC	SH	IC	IC			.					.		
8E	CALL	lab	OC	<<	O1	<<	O2	<<	IC	IC	SH	IC	SH	IC	IC		.					.		
AE	CALL	*lab[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	SH	IC	SH	IC	IC					.		
F4 EE	CALL	*n[Rp]	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	IC	SH	IC	SH	IC	IC		
code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Table F–4. Bus Activity Table (Continued)

Code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
9F	CALLR	*Rp	OC	<<	O1	<<	Rr	Rr	IC	IC	IC	SH	IC	SH	IC	IC	.					.		
8F	CALLR	#16n	OC	<<	O1	<<	O2	<<	IC	IC	IC	IC	SH	IC	SH	IC	IC					.		
8F	CALLR	lab	OC	<<	O1	<<	O2	<<	IC	IC	IC	IC	SH	IC	SH	IC	IC					.		
AF	CALLR	*16n[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	IC	IC	SH	IC	SH	IC	IC			.		
AF	CALLR	*lab[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	IC	IC	SH	IC	SH	IC	IC			.		
F4 EF	CALLR	*n[Rp]	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	IC	IC	IC	SH	IC	SH	IC	IC
B5	CLR	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
C5	CLR	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
D5	CLR	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.		
B0	CLRC		OC	<<	PO	<<	Ar	IC	IC	Ar	Aw	.					(same as TST A instruction)							
F3	CMP	*n[SP],A	OC	<<	O1	<<	IC	RJ	Rr	IC		.					.					.		
2D	CMP	#n,A	OC	<<	O1	<<	Ar	IC				.					.					.		
5D	CMP	#n,B	OC	<<	O1	<<	Br	IC				.					.					.		
7D	CMP	#n,Rp	OC	<<	O1	<<	O2	<<	Rr	IC		.					.					.		
9D	CMP	*Rp,A	OC	<<	O1	<<	Rr	Rr	Lr	<<	Ar	IC					.					.		
6D	CMP	B,A	OC	<<	PO	<<	Br	IC	Ar	IC		.					.					.		
AD	CMP	*lab[B],A	OC	<<	O1	<<	O2	<<	Br	IC	IC	Lr	<<	Ar	IC		.					.		
8D	CMP	&lab,A	OC	<<	O1	<<	O2	<<	IC	Lr	<<	Ar	IC				.					.		
F4 ED	CMP	*n[Rp],A	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	Lr	<<	Ar	IC		.		
1D	CMP	Rs,A	OC	<<	O1	<<	Rr	Ar	IC			.					.					.		
3D	CMP	Rs,B	OC	<<	O1	<<	Rr	Br	IC			.					.					.		
4D	CMP	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	IC	.					.					.		
A5	CMPBIT	Pname	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					(same as XOR #n,Pd instruction)							
75	CMPBIT	Rname	OC	<<	O1	<<	O2	<<	Rr	Rw		.					(same as XOR #n,Rn instruction)							
BB	COMPL	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
CB	COMPL	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
DB	COMPL	Rd	OC	<<	O1	<<	Rr	Rw				.					.							
2E	DAC	#n,A	OC	<<	O1	<<	Ar	IC	IC	Aw		.					.					.		
5E	DAC	#n,B	OC	<<	O1	<<	Br	IC	IC	Bw		.					.					.		
7E	DAC	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	IC	IC	Rw					.					.		
6E	DAC	B,A	OC	<<	PO	<<	Br	IC	Ar	IC	IC	Aw					.					.		
1E	DAC	Rs,A	OC	<<	O1	<<	Rr	Ar	IC	IC	Aw	.					.					.		
3E	DAC	Rs,B	OC	<<	O1	<<	Rr	Br	IC	IC	Bw	.					.					.		
4E	DAC	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	IC	IC	Rw				.					.		
B2	DEC	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
C2	DEC	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
D2	DEC	Rd	OC	<<	O1	<<	Rr	Rw				.					.							
F4 F0	DINT		OC	<<	PO	<<	Sr	Br	IC			.					(same as LDST 00h instruction)							
F8	DIV	Rs,A	PF	<<	OC	<<	IC	IC	O1	<<	Ar	Rr	Br	IC	IC	IC	overflow condition					.		
F4 F8	DIV	Rs,A	PF	<<	OC	<<	IC	IC	O1	<<	Ar	Rr	Br	IC	IC	Aw	[Bw	IC	IC	IC	Aw]	IC		
												.					. Repeat Block 55–63 cycles							
BA	DJNZ	A,#ra	OC	<<	PO	<<	Ar	IC	IC	Aw	O1	<<	RJ	RJ			.					.		
CA	DJNZ	B,#ra	OC	<<	PO	<<	Br	IC	IC	Bw	O1	<<	RJ	RJ			.					.		
DA	DJNZ	Rd,#ra	OC	<<	O1	<<	Rr	Rw	O2	<<	RJ	RJ					.					.		
2F	DSB	#n,A	OC	<<	O1	<<	Ar	IC	IC	Aw		.					.					.		
5F	DSB	#n,B	OC	<<	O1	<<	Br	IC	IC	Bw		.					.					.		
7F	DSB	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	IC	IC	Rw					.					.		
6F	DSB	B,A	OC	<<	PO	<<	Br	IC	Ar	IC	IC	Aw					.					.		
1F	DSB	Rs,A	OC	<<	O1	<<	Rr	Ar	IC	IC	Aw	.					.					.		
3F	DSB	Rs,B	OC	<<	O1	<<	Rr	Br	IC	IC	Bw	.					.					.		
4F	DSB	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	IC	IC	Rw				.					.		
F00C	EINT		OC	<<	PO	<<	Sr	Br	IC			.					(same as LDST 0Ch instruction)							
F004	EINTH		OC	<<	PO	<<	Sr	Br	IC			.					(same as LDST 04h instruction)							
code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Table F–4. Bus Activity Table (Continued)

Code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
F008	EINTL		OC	<<	PO	<<	Sr	Br	IC			.					(same as LDST 08h instruction)						
F6	IDLE		OC	<<	PO	<<	IC	IC				.					(usually exited by interrupt)						
B3	INC	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.	
C3	INC	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.	
D3	INC	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.	
70	INCW	#n,Rp	OC	<<	O1	<<	O2	<<	Rr	Rw	IC	Rr	Rw				.					.	
B4	INV	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.	
C4	INV	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.	
D4	INV	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.	
A7	JBIT0	Pname,#ra	OC	<<	O1	<<	O2	<<	Pr	<<	IC	O3	<<	RJ	RJ		(same as BTJZ #n,Pd,ra instruction)						
77	JBIT0	Rname,#ra	OC	<<	O1	<<	O2	<<	Rr	IC	O3	<<	RJ	RJ			(same as BTJZ #n,Rn,ra instruction)						
A6	JBIT1	Pname,#ra	OC	<<	O1	<<	O2	<<	Pr	<<	IC	O3	<<	RJ	RJ		(same as BTJO #n,Pd,ra instruction)						
76	JBIT1	Rname,#ra	OC	<<	O1	<<	O2	<<	Rr	IC	O3	<<	RJ	RJ			(same as BTJO #n,Rn,ra instruction)						
01–0F	Jcnd	#ra	OC	<<	O1	<<	IC	RJ	RJ			.					.				.		
00	JMP	#ra	OC	<<	O1	<<	IC	RJ	RJ			.					.					.	
99	JMPL	*Rp	OC	<<	O1	<<	Rr	Rr	IC	IC		.					.					.	
89	JMPL	#16n	OC	<<	O1	<<	O2	<<	IC	IC	IC	.					.					.	
89	JMPL	lab	OC	<<	O1	<<	O2	<<	IC	IC	IC	.					.					.	
A9	JMPL	*16n[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	IC				.					.	
A9	JMPL	*lab[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	IC	IC				.					.	
F4 E9	JMPL	*n[Rp]	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	IC	IC					
FD	LDSP		OC	<<	PO	<<	Sr	Br	IC			.					.					.	
F0	LDST	#n	OC	<<	O1	<<	IC	IC				.					.					.	
F2	MOV	A,*n[SP]	OC	<<	O1	<<	IC	Ar	Rw			.					.					.	
F1	MOV	*n[SP],A	OC	<<	O1	<<	IC	Rr	Aw			.					.					.	
22	MOV	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.	
52	MOV	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.	
F7	MOV	#n,Pd	OC	<<	O1	<<	IC	O2	<<	IC	Pw	<<					.					.	
72	MOV	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.	
9A	MOV	*Rp,A	OC	<<	O1	<<	Rr	Rr	Lr	<<	Aw	.					.					.	
9B	MOV	A,*Rp	OC	<<	O1	<<	Rr	Rr	Ar	Lw	<<	.					.					.	
C0	MOV	A,B	OC	<<	PO	<<	Br	IC	IC	Ar	Bw	.					.					.	
8B	MOV	A,&lab	OC	<<	O1	<<	O2	<<	IC	Ar	Lw	<<					.					.	
AB	MOV	A,*lab[B]	OC	<<	O1	<<	O2	<<	Br	IC	IC	Ar	Lw	<<			.					.	
F4 EB	MOV	A,*n[Rpd]	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	Ar	LW	<<			.	
21	MOV	A,Pd	OC	<<	O1	<<	Ar	IC	Pw	<<		.					.					.	
D0	MOV	A,Rd	OC	<<	O1	<<	Rr	Ar	Rw			.					.					.	
62	MOV	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.	
51	MOV	B,Pd	OC	<<	O1	<<	Br	IC	Pw	<<		.					.					.	
D1	MOV	B,Rd	OC	<<	O1	<<	Rr	Br	Rw			.					.					.	
AA	MOV	*lab[B],A	OC	<<	O1	<<	O2	<<	Br	IC	IC	Lr	<<	Aw			.					.	
8A	MOV	&lab,A	OC	<<	O1	<<	O2	<<	IC	Lr	<<	Aw					.					.	
F4 EA	MOV	*n[Rp],A	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	LR	<<	Aw		.		
80	MOV	Ps,A	OC	<<	O1	<<	Ar	Pr	<<	Aw		.					.					.	
91	MOV	Ps,B	OC	<<	O1	<<	Br	Pr	<<	Bw		.					.					.	
A2	MOV	Ps,Rd	OC	<<	O2r	<<	O1p	<<	Pr	<<	IC	Rw					operand order is reversed during assembly						
12	MOV	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.	
32	MOV	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.	
71	MOV	Rs,Pd	OC	<<	O2p	<<	O1r	<<	Rr	IC	Pw	<<					operand order is reversed during assembly						
42	MOV	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.	
A8	MOVW	#16n[B],Rpd	OC	<<	O1	<<	O2	<<	Br	IC	IC	O3	<<	IC	Rw	IC	Rw					.	
88	MOVW	#16n,Rpd	OC	<<	O1	<<	O2	<<	IC	O3	<<	IC	Rw	IC	Rw		.					.	
E8	MOVW	#n[Rp],Rpd	PF	<<	OC	<<	IC	IC	O1	<<	O2	<<	Rr	Rr	IC	IC	O3	<<	IC	RW	IC	RW	
code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Table F–4. Bus Activity Table (Continued)

Code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
98	MOVW	Rps,Rpd	OC	<<	O1	<<	Rr	Rr	O2	<<	IC	Rw	IC	Rw			.					.		
2C	MPY	#n,A	OC	<<	O1	<<	Ar	. . . . .									45	cycles total						
5C	MPY	#n,B	OC	<<	O1	<<	Br	. . . . .									45	cycles total						
7C	MPY	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	. . . . .							47	cycles total						
6C	MPY	B,A	OC	<<	PO	<<	Br	IC	Ar	. . . . .							47	cycles total						
1C	MPY	Rs,A	OC	<<	O1	<<	Rr	Ar	Br	IC	IC	Br	Bw	IC	Aw	. . . . .	46	cycles total						
3C	MPY	Rs,B	OC	<<	O1	<<	Rr	Br	. . . . .								46	cycles total						
4C	MPY	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	. . . . .						48	cycles total						
FF	NOP		OC	<<	PO	<<	Sr	Br	IC			.					.					.		
24	OR	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
54	OR	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
A4	OR	#n,Pd	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					.					.		
74	OR	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
84	OR	A,Pd	OC	<<	O1	<<	Ar	Pr	<<	Pw	<<	.					.					.		
64	OR	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
94	OR	B,Pd	OC	<<	O1	<<	Br	Pr	<<	Pw	<<	.					.					.		
14	OR	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
34	OR	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
44	OR	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
B9	POP	A	OC	<<	PO	<<	Ar	IC	IC	SP	Aw	.					.					.		
C9	POP	B	OC	<<	PO	<<	Br	IC	IC	SP	Bw	.					.					.		
D9	POP	Rd	OC	<<	O1	<<	Rr	SP	Rw			.					.					.		
FC	POP	ST	OC	<<	PO	<<	SP	Br	IC	IC		.					.					.		
B8	PUSH	A	OC	<<	PO	<<	Ar	IC	IC	IC	SH	.					.					.		
C8	PUSH	B	OC	<<	PO	<<	Br	IC	IC	IC	SH	.					.					.		
D8	PUSH	Rs	OC	<<	O1	<<	Rr	IC	SH			.					.					.		
FB	PUSH	ST	OC	<<	PO	<<	Sr	Br	IC	SH		.					.					.		
BE	RL	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
CE	RL	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
DE	RL	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.		
BF	RLC	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
CF	RLC	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
DF	RLC	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.		
BC	RR	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
CC	RR	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
DC	RR	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.		
BD	RRC	A	OC	<<	PO	<<	Ar	IC	IC	Aw		.					.					.		
CD	RRC	B	OC	<<	PO	<<	Br	IC	IC	Bw		.					.					.		
DD	RRC	Rd	OC	<<	O1	<<	Rr	Rw				.					.					.		
FA	RTI		OC	<<	PO	<<	SP	Br	IC	SP	SP	IC	IC	IC			(PCl,PCm, ST)							
F9	RTS		OC	<<	PO	<<	SP	Br	IC	SP	IC	.					(PCl, PCm)							
2B	SBB	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
5B	SBB	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
7B	SBB	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
6B	SBB	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
1B	SBB	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
3B	SBB	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
4B	SBB	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
A3	SBIT0	Pname	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					(same as AND #n,Pd)							
73	SBIT0	Rname	OC	<<	O1	<<	O2	<<	Rr	Rw		.					(same as AND #n,Rn)							

code	Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Table F–4. Bus Activity Table (Continued)

Code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A4	SBIT1	Pname	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					(same as OR #n,Pd)							
74	SBIT1	Rname	OC	<<	O1	<<	O2	<<	Rr	Rw		.					(same as OR #n,Rn)							
F8	SETC		OC	<<	PO	<<	Sr	Br	IC			.					.					.		
FE	STSP		OC	<<	PO	<<	Sr	Br	IC	Bw		.					.					.		
2A	SUB	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
5A	SUB	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
7A	SUB	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
6A	SUB	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
1A	SUB	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
3A	SUB	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
4A	SUB	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
B7	SWAP	A	OC	<<	PO	<<	Ar	IC	IC	IC	IC	IC	Aw				.					.		
C7	SWAP	B	OC	<<	PO	<<	Br	IC	IC	IC	IC	IC	Bw				.					.		
D7	SWAP	Rd	OC	<<	O1	<<	Rr	IC	IC	IC	Rw	.					.					.		
EF–E0	TRAP	#n	OC	<<	PO	<<	IC	IC	IC	IC	SH	Vr	<<	SH	Vr	<<	.					.		
B0	TST	A	OC	<<	PO	<<	Ar	IC	IC	Ar	Aw	.					.					.		
C6	TST	B	OC	<<	PO	<<	Br	IC	IC	Br	Bw	Bw					.					.		
B6	XCHB	A	OC	<<	PO	<<	Ar	IC	IC	Br	Bw	Aw					.					.		
C6	XCHB	B	OC	<<	PO	<<	Br	IC	IC	Br	Bw	Bw					.					.		
D6	XCHB	Rd	OC	<<	O1	<<	Rr	Br	Bw	Rw		.					.					.		
25	XOR	#n,A	OC	<<	O1	<<	Ar	Aw				.					.					.		
55	XOR	#n,B	OC	<<	O1	<<	Br	Bw				.					.					.		
A5	XOR	#n,Pd	OC	<<	O1	<<	O2	<<	Pr	<<	Pw	<<					.					.		
75	XOR	#n,Rd	OC	<<	O1	<<	O2	<<	Rr	Rw		.					.					.		
85	XOR	A,Pd	OC	<<	O1	<<	Ar	Pr	<<	Pw	<<	.					.					.		
65	XOR	B,A	OC	<<	PO	<<	Br	IC	Ar	Aw		.					.					.		
95	XOR	B,Pd	OC	<<	O1	<<	Br	Pr	<<	Pw	<<	.					.					.		
15	XOR	Rs,A	OC	<<	O1	<<	Rr	Ar	Aw			.					.					.		
35	XOR	Rs,B	OC	<<	O1	<<	Rr	Br	Bw			.					.					.		
45	XOR	Rs,Rd	OC	<<	O1	<<	Rr	O2	<<	Rr	Rw	.					.					.		
code	Instruction		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

The instructions in Table F–4 have the following general characteristics:

- Opcodes and operands are executed in the same order as in the written instruction. (except MOV Rs,Pd and MOV Ps,Rd).
- All register pairs are accessed least significant byte first then most significant byte. (n then, n–1).
- Calls push PCH then PCL.
- All external writes occur on the last cycle of the instruction.
- All instructions make at least one operand fetch, even if not needed. The PC does not advance; it treats the pseudo-operand as an opcode on the next opcode fetch. Identified by PO.
- MPY performs register A,B accesses and internal cycles for the number of cycles shown.
- DIV execution time depends on the values divided but ranges from 55–63 cycles—fourteen cycles if overflow is detected. Overflow is detected if divisor <= dividend MSB.

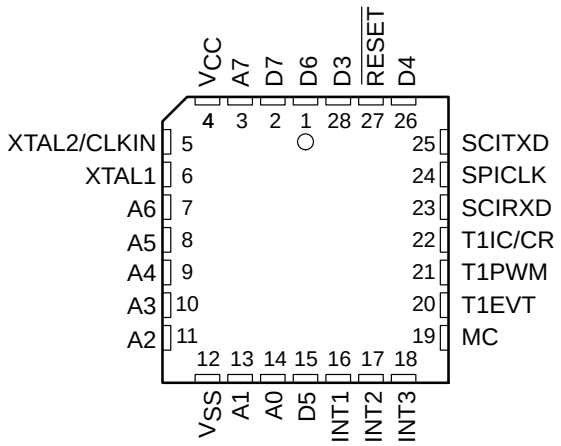
Appendix G

Device Pinouts

This appendix provides pinouts for the following individual device categories:

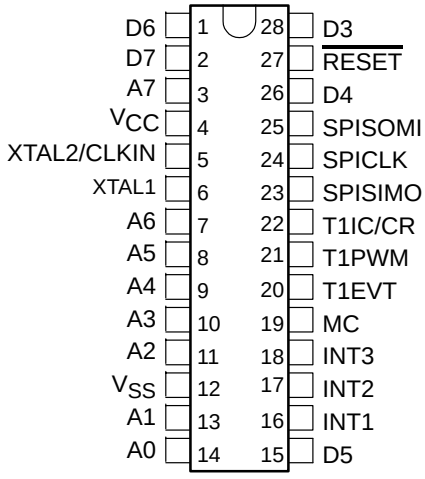
For individual pin descriptions, refer to Chapter 2.

Figure G–1. Pinouts for TMS370Cx0x Devices (Top View)

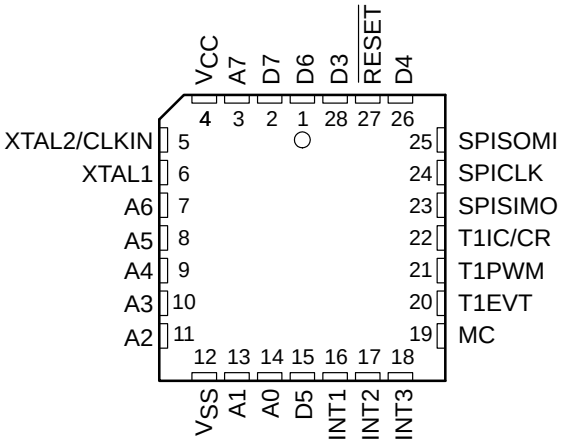


A. 28-Pin PLCC (FN)
B. 28-Pin CLCC (FZ)

Figure G–2. Pinouts for TMS370Cx1x Devices (Top View)

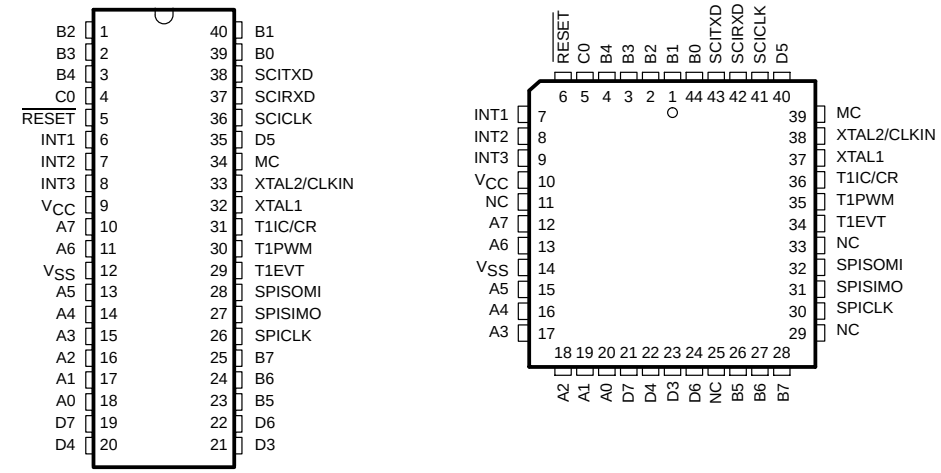


A. 28-Pin PDIP (N)
B. 28-Pin CDIP (JD)



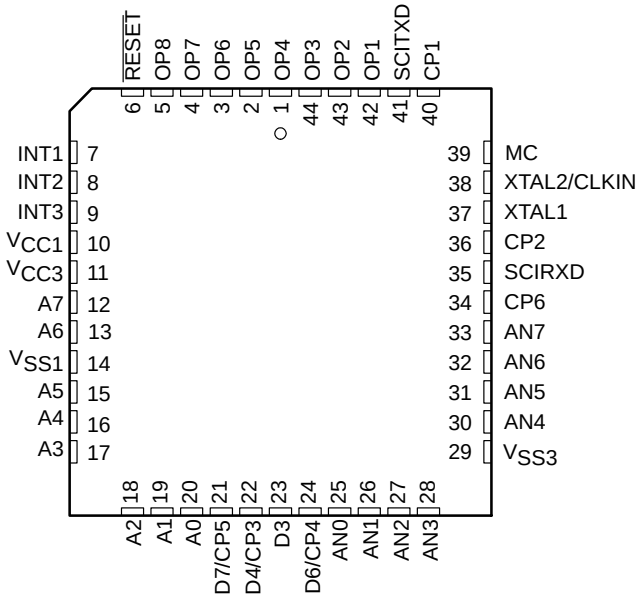
C. 28-Pin PLCC (FN)
D. 28-Pin CLCC (FZ)

Figure G–3. Pinouts for TMS370Cx2x Devices (Top View)



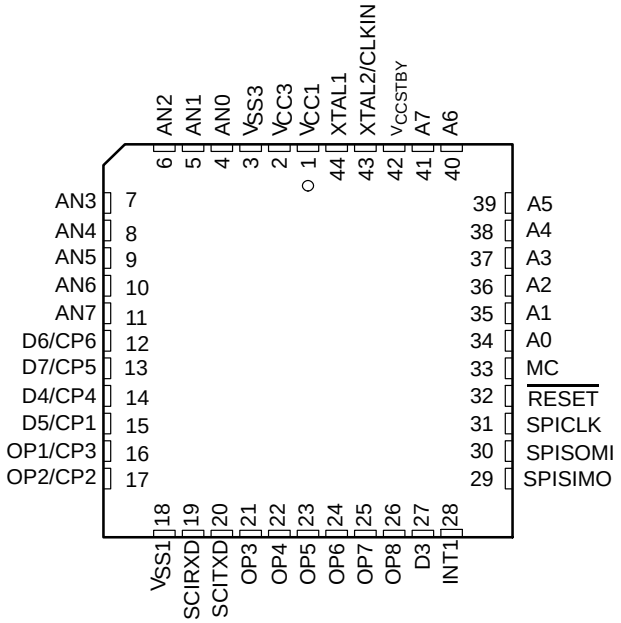
A. 40-Pin PDIP (N)
B. 40-Pin CDIP (JD)
C. 40-Pin PSDIP (NJ) formerly known as N2
D. 40-Pin CSDIP (JC)
E. 44-Pin PLCC (FN)
F. 44-Pin CLCC (FZ)

Figure G–4. Pinouts for TMS370Cx32 Devices (Top View)



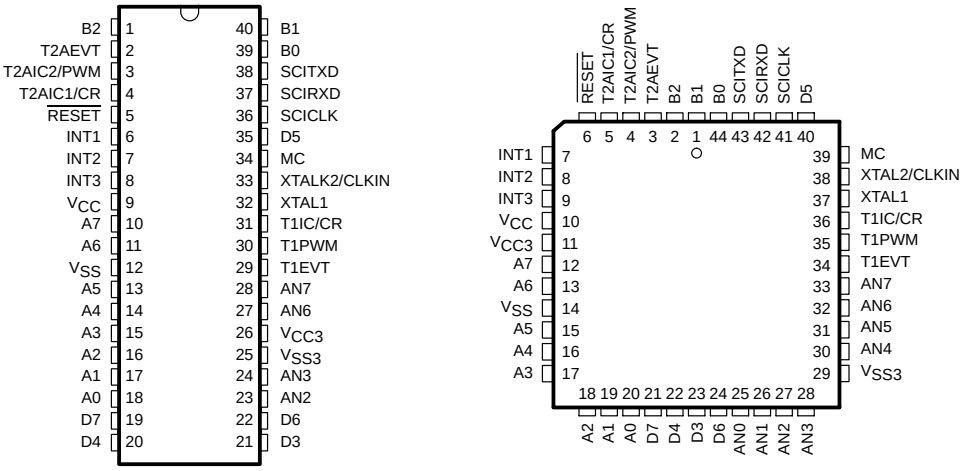
A. 44-Pin PLCC (FN)
B. 44-Pin CLCC (FZ)

Figure G–5. Pinouts for TMS370Cx36 Devices (Top View)



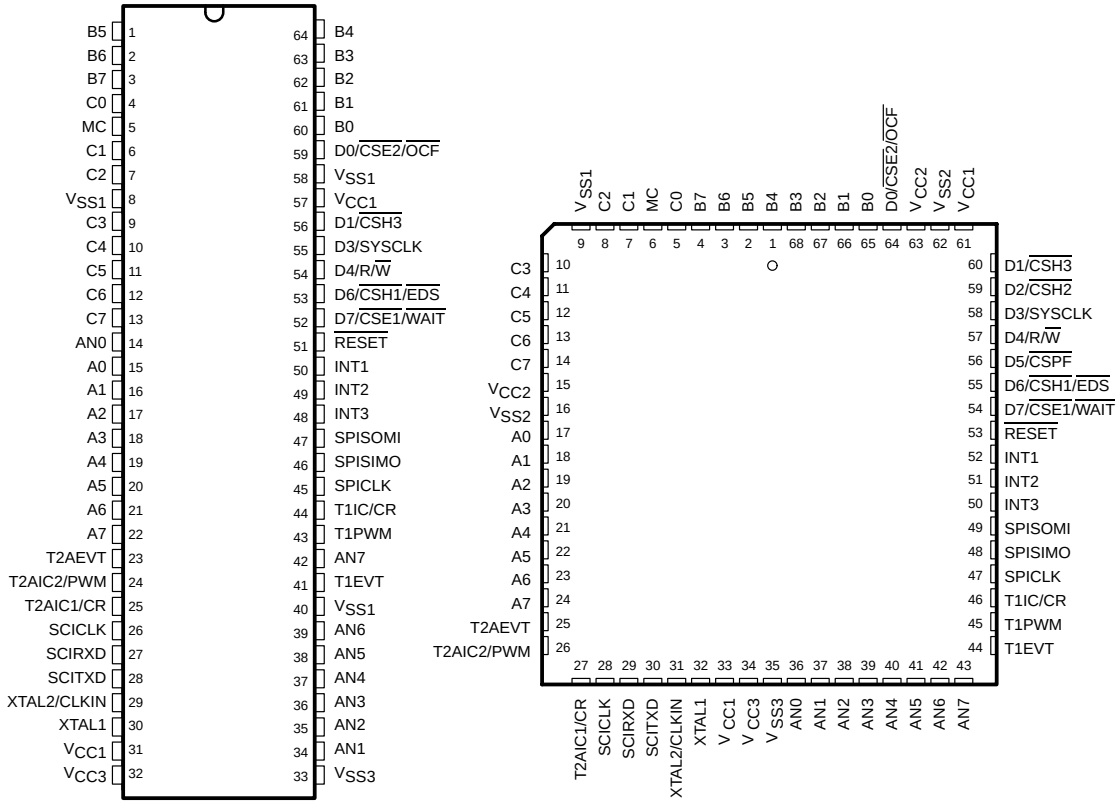
A. 44-Pin PLCC (FN)
B. 44-Pin CLCC (FZ)

Figure G–6. Pinouts for TMS370Cx4x Devices (Top View)



A. 40-Pin PDIP (N)
B. 40-Pin CDIP (JD)
C. 40-Pin PSDIP (NJ) formerly known as N2
D. 40-Pin CSDIP (JC)
E. 44-Pin PLCC (FN)
F. 44-Pin CLCC (FZ)

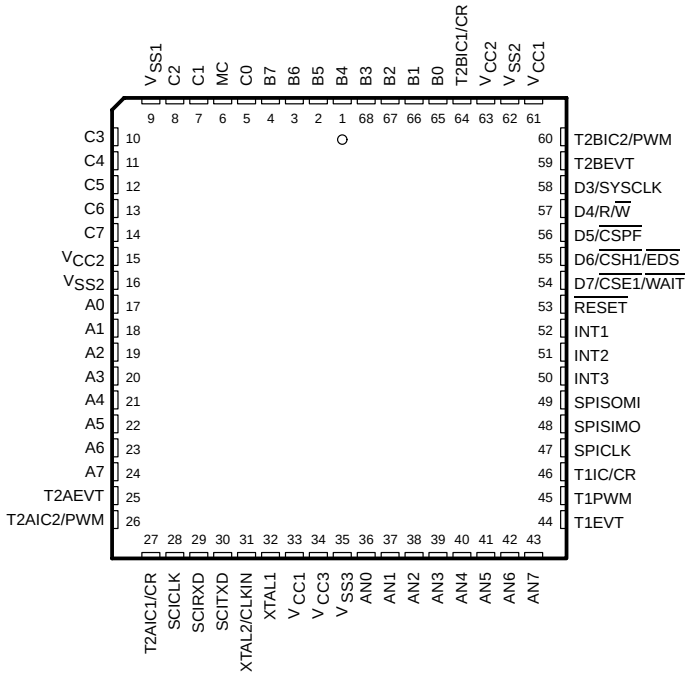
Figure G–7. Pinouts for TMS370Cx5x Devices (Top View)



A. 64-Pin PSDIP (NM)
B. 64-Pin CSDIP (JN)

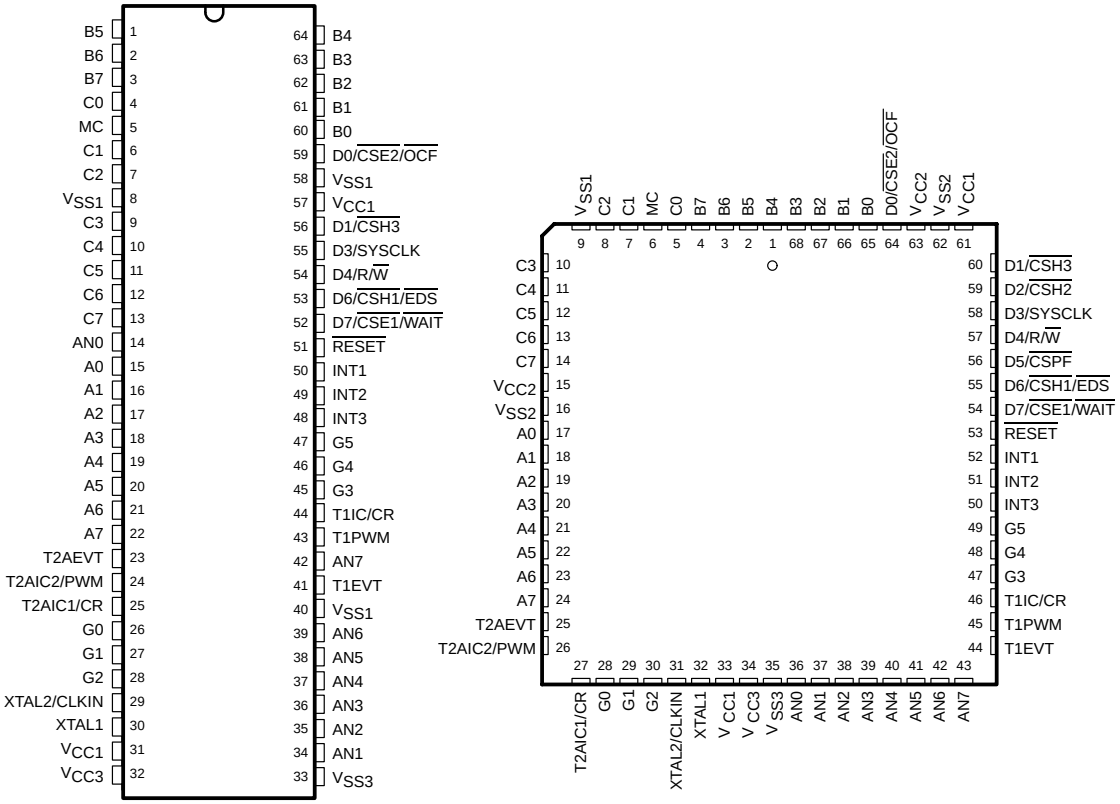
C. 68-Pin PLCC (FN)
D. 68-Pin CLCC (FZ)

Figure G–8. Pinouts for TMS370Cx6x Devices (Top View)



A. 68-Pin PLCC (FN)
B. 68-Pin CLCC (FZ)

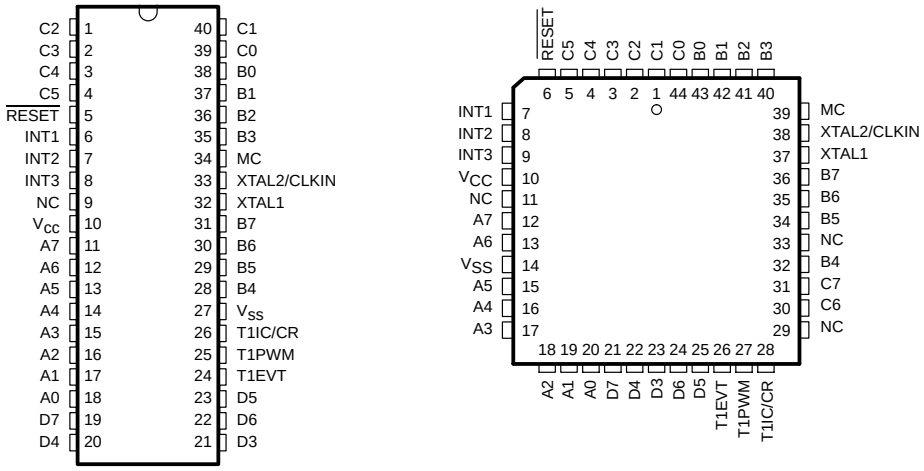
Figure G–9. Pinouts for TMS370Cx7x Devices (Top View)



A. 64-Pin PSDIP (NM)
B. 64-Pin CSDIP (JN)

C. 68-Pin PLCC (FN)
D. 68-Pin CLCC (FZ)

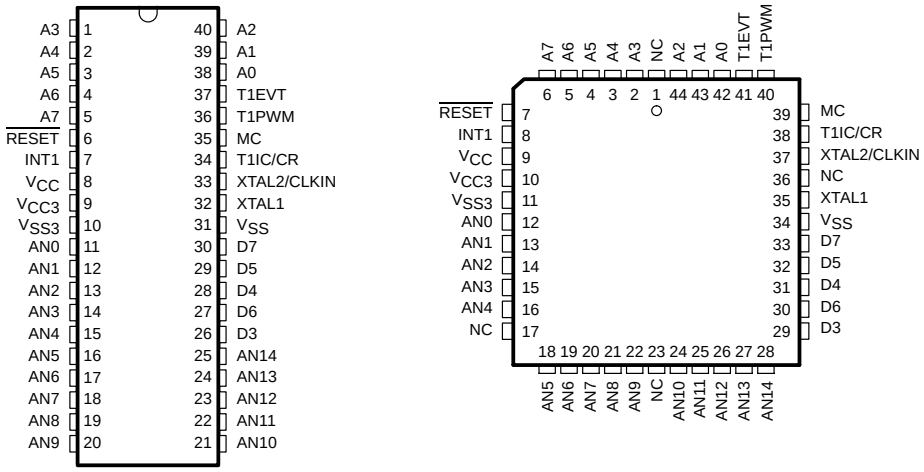
Figure G–10. Pinouts for TMS370Cx8x Devices (Top View)



A. 40-Pin PDIP (N)

B. 44-Pin PLCC (FN)
C. 44-Pin CLCC (FZ)

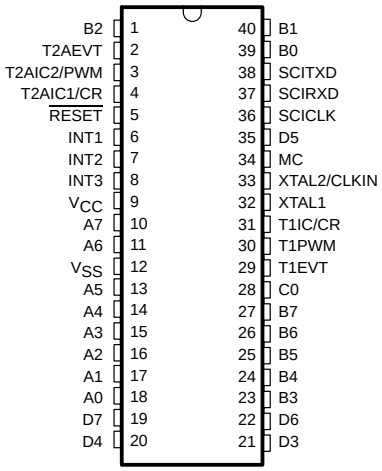
Figure G–11. Pinouts for TMS370Cx9x Devices (Top View)



A. 40-Pin PSDIP (NJ) formerly known as N2
B. 40-Pin CSDIP (JC)

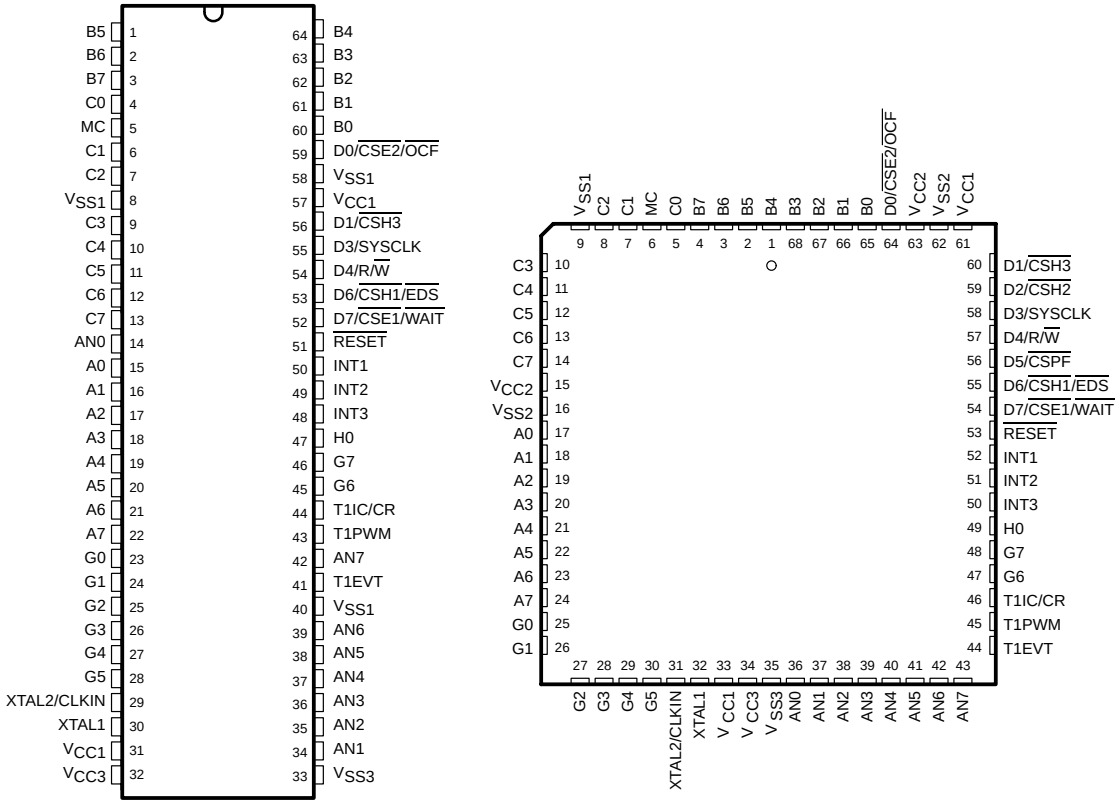
C. 44-Pin PLCC (FN)
D. 44-Pin CLCC (FZ)

Figure G–12. Pinouts for TMS370CxAx Devices (Top View)



A. 40-Pin PDIP (N)

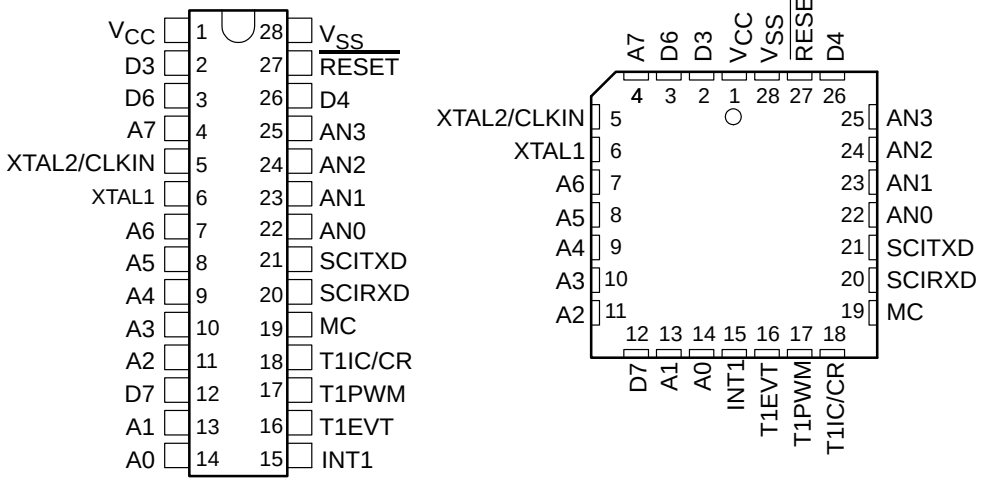
Figure G–13. Pinouts for TMS370CxBx Devices (Top View)



A. 64-Pin PSDIP (NM)

B. 68-Pin PLCC (FN)

Figure G–14. Pinouts for TMS370CxCx Devices (Top View)



A. 28-Pin PDIP (N)
B. 28-Pin CDIP (JD)
C. 28-Pin PLCC (FN)
D. 28-Pin CLCC (FZ)

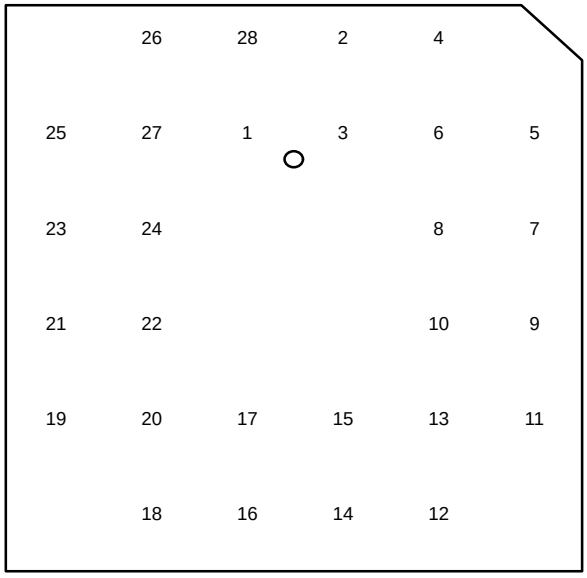
Appendix H

PLCC-to-PGA Socket Pinouts

This appendix shows the pinouts for the standard PLCC-to-PGA sockets that are commonly used in prototype and production applications. You can use these pinouts when you wirewrap your breadboard with a socket. These diagrams make constructing, debugging, and troubleshooting with the TMS370 family quicker and easier.

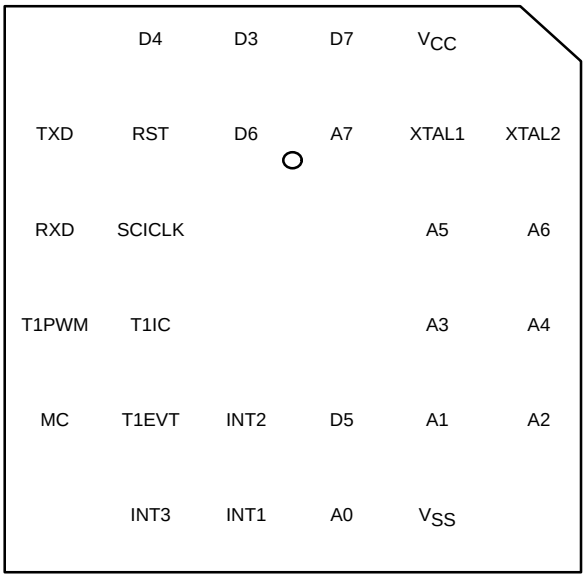
The figures shown in this appendix are for sockets that correspond to the following package types:

Figure H–1. 28-Pin PGA Pinout



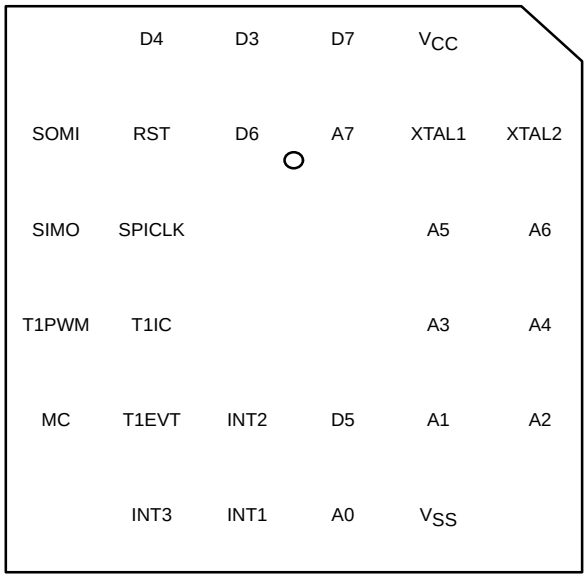
Bottom View

Figure H–2. TMS370Cx0x Device PGA Pinout



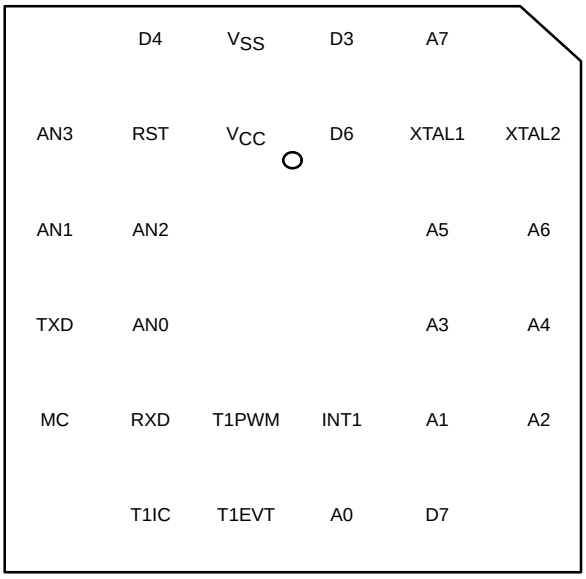
Bottom View

Figure H–3. TMS370Cx1x Device PGA Pinout



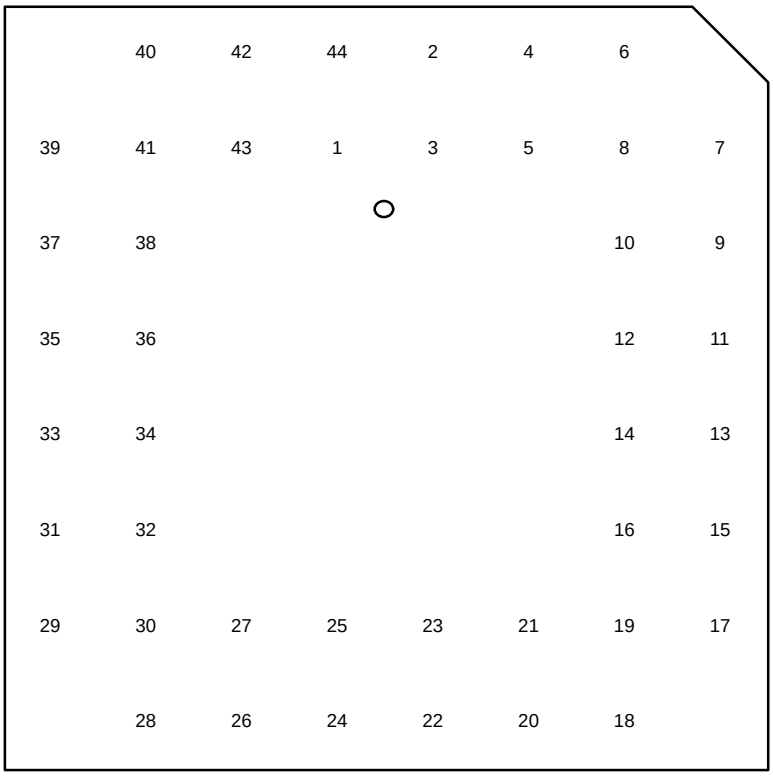
Bottom View

Figure H–4. TMS370CxCx Device PGA Pinout



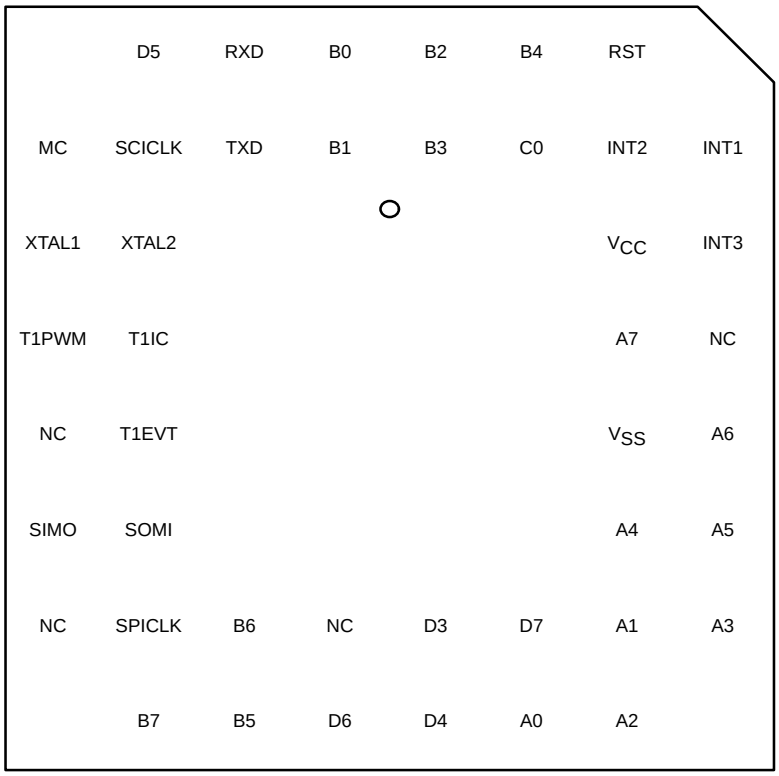
Bottom View

Figure H–5. 44-Pin PGA Pinout



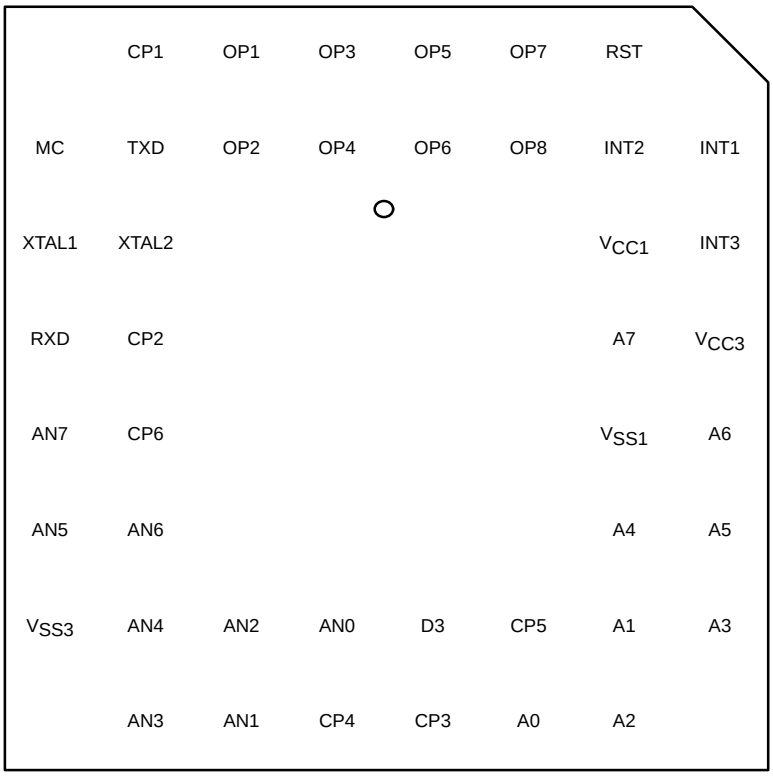
Bottom View

Figure H–6. TMS370Cx2x Device PGA Pinout



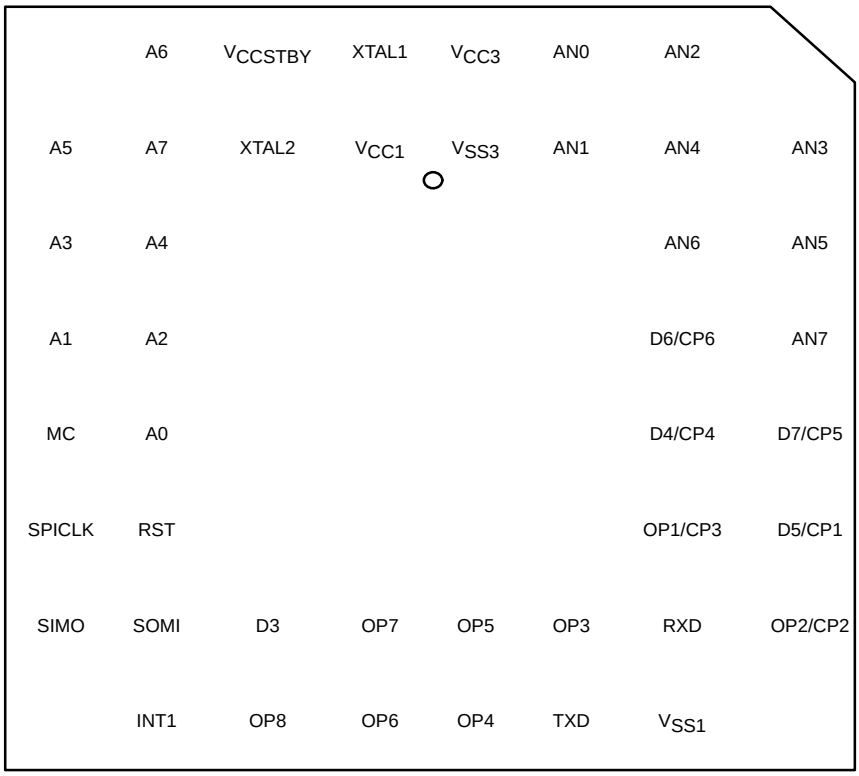
Bottom View

Figure H–7. TMS370Cx32 Device PGA Pinout



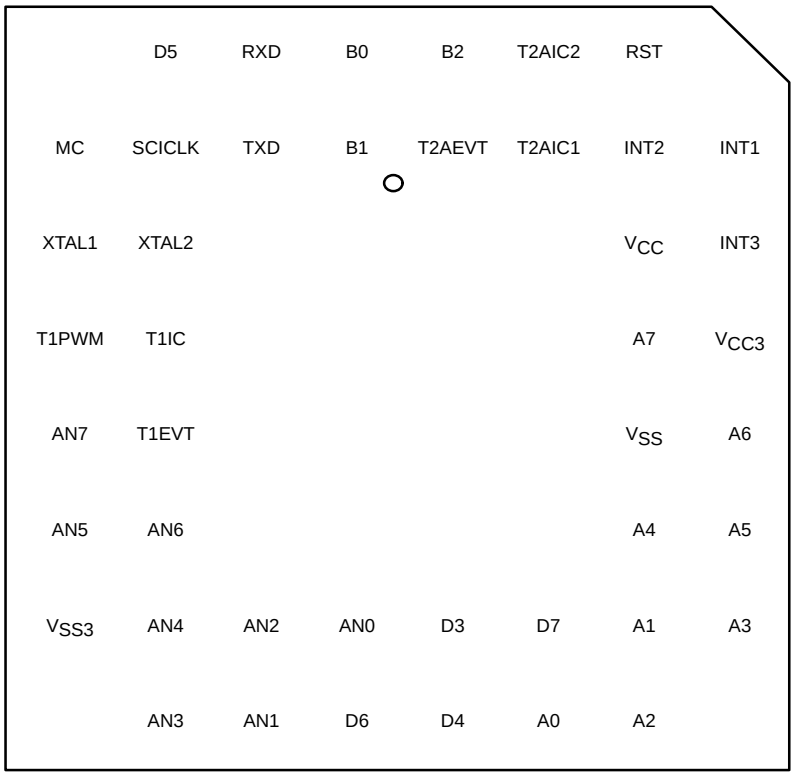
Bottom View

Figure H–8. TMS370Cx36 Device PGA Pinout



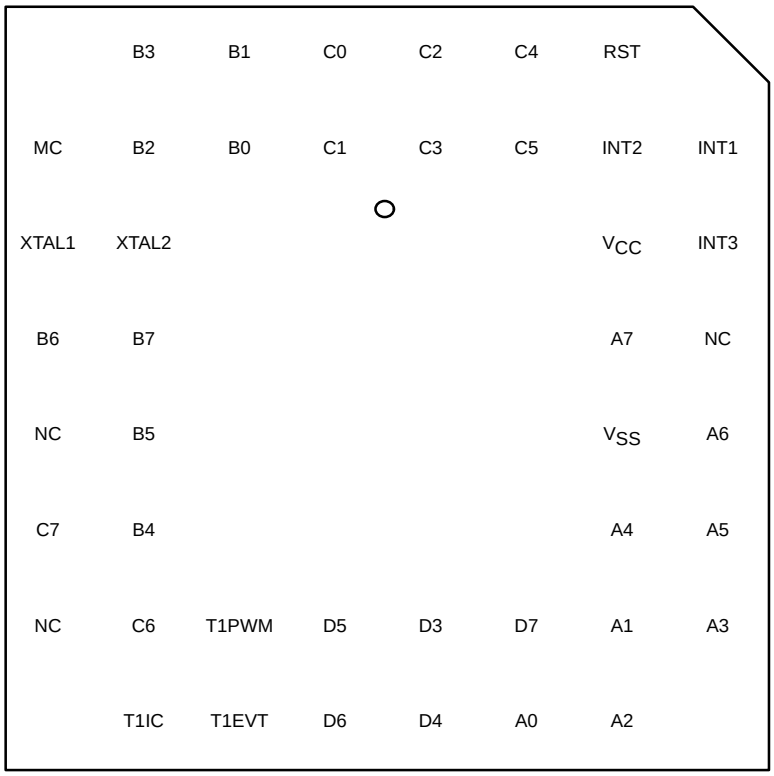
Bottom View

Figure H–9. TMS370Cx4x Device PGA Pinout



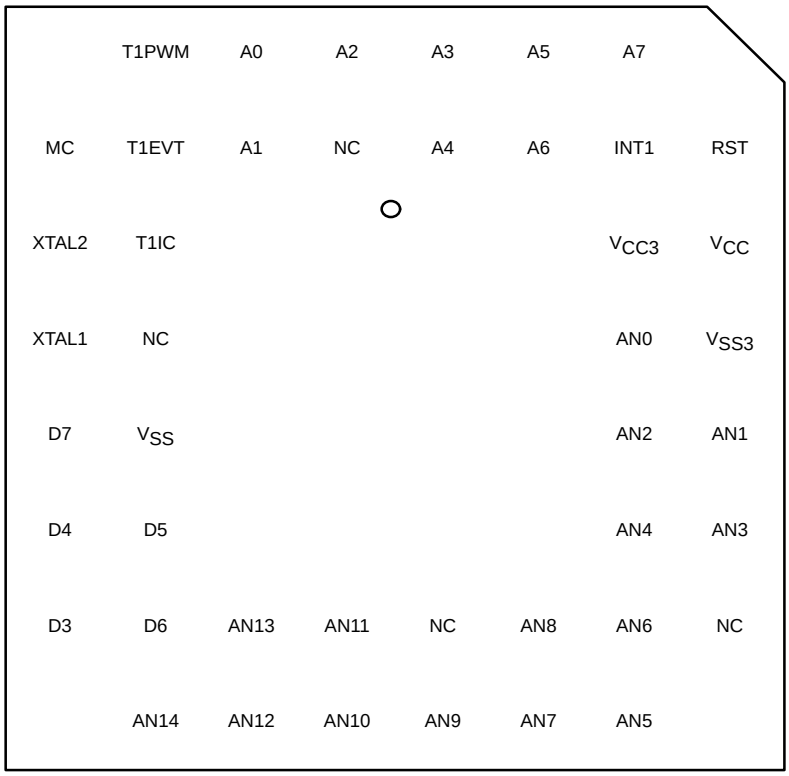
Bottom View

Figure H–10. TMS370Cx8x Device PGA Pinout



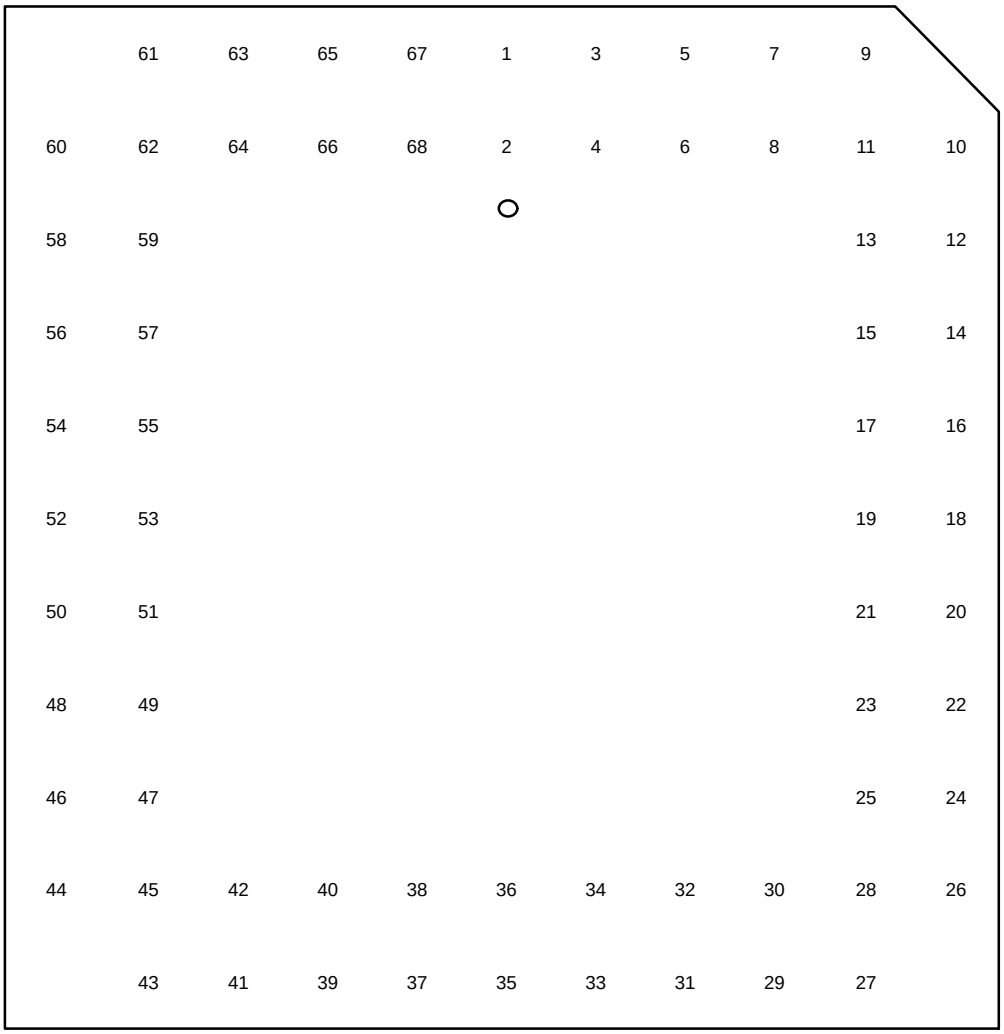
Bottom View

Figure H–11. TMS370Cx9x Device PGA Pinout



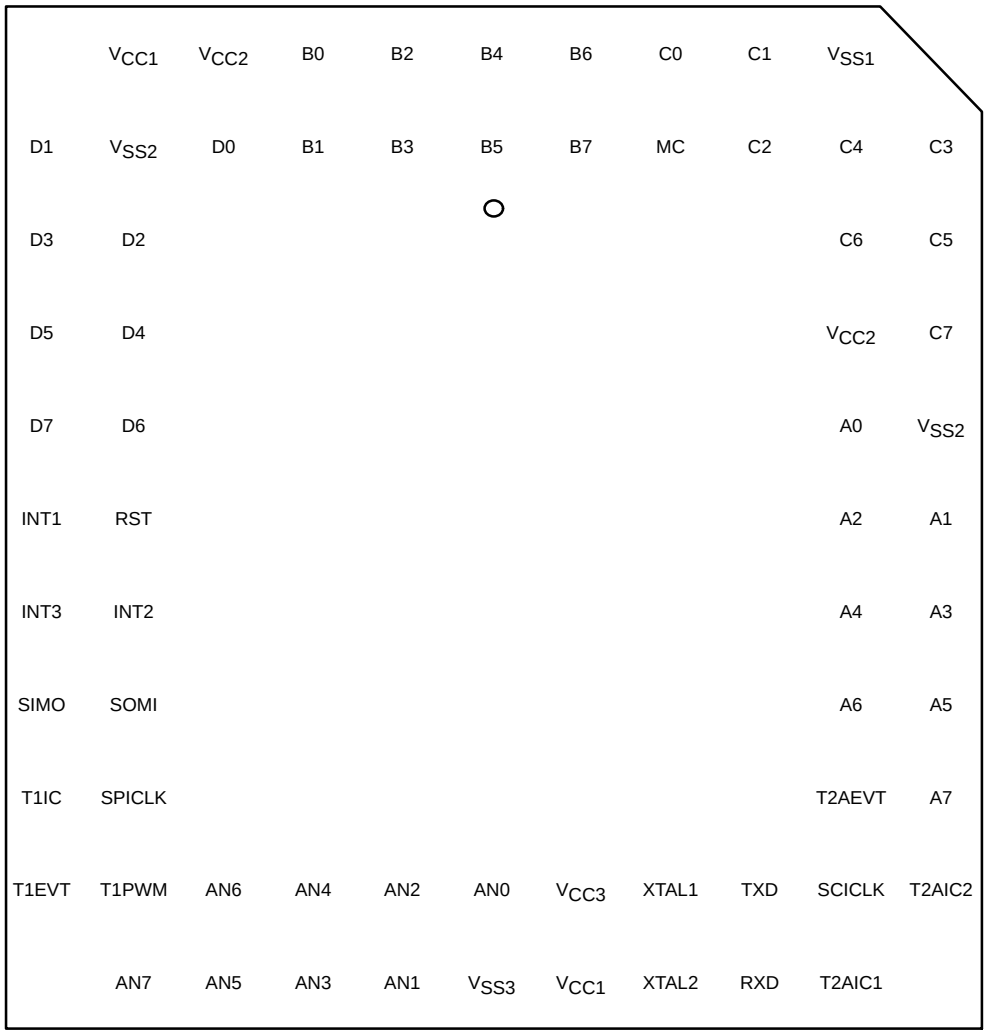
Bottom View

Figure H–12. 68-Pin PGA Pinout



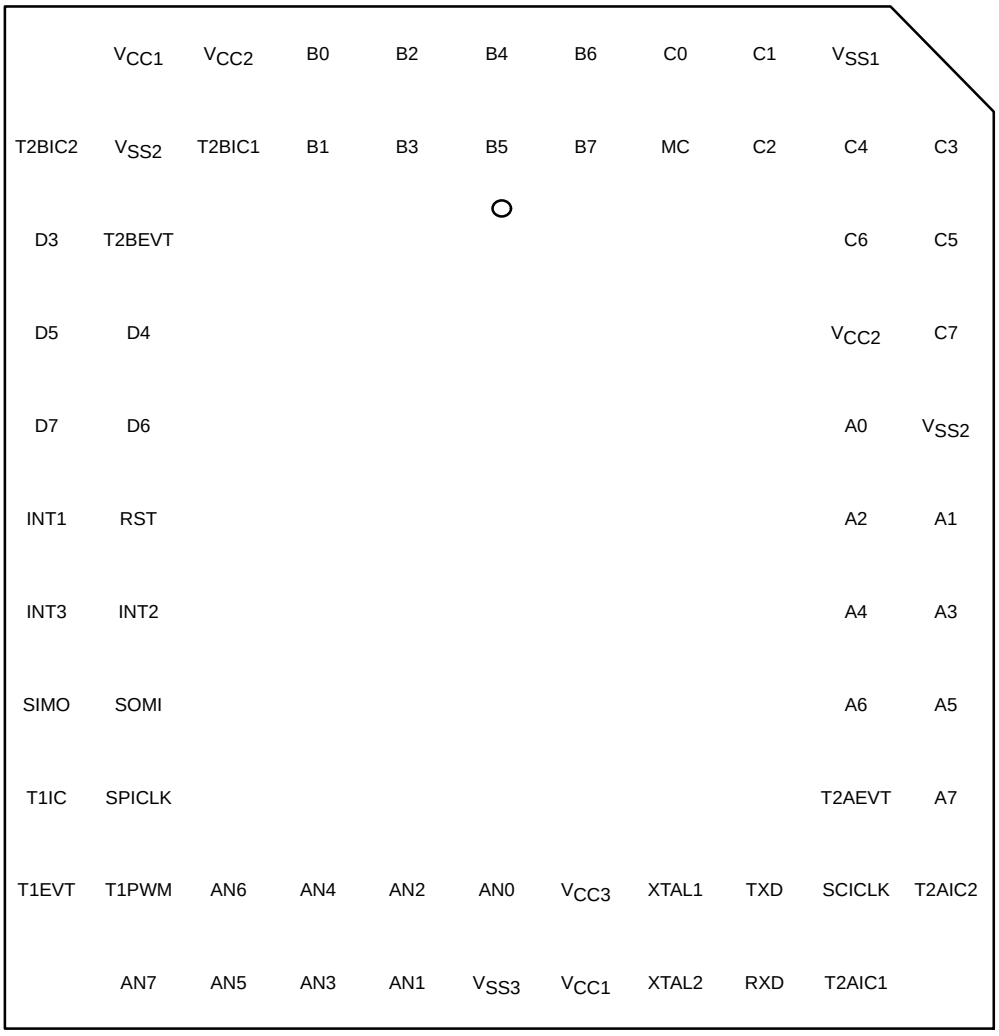
Bottom View

Figure H–13. TMS370Cx5x Device PGA Pinout



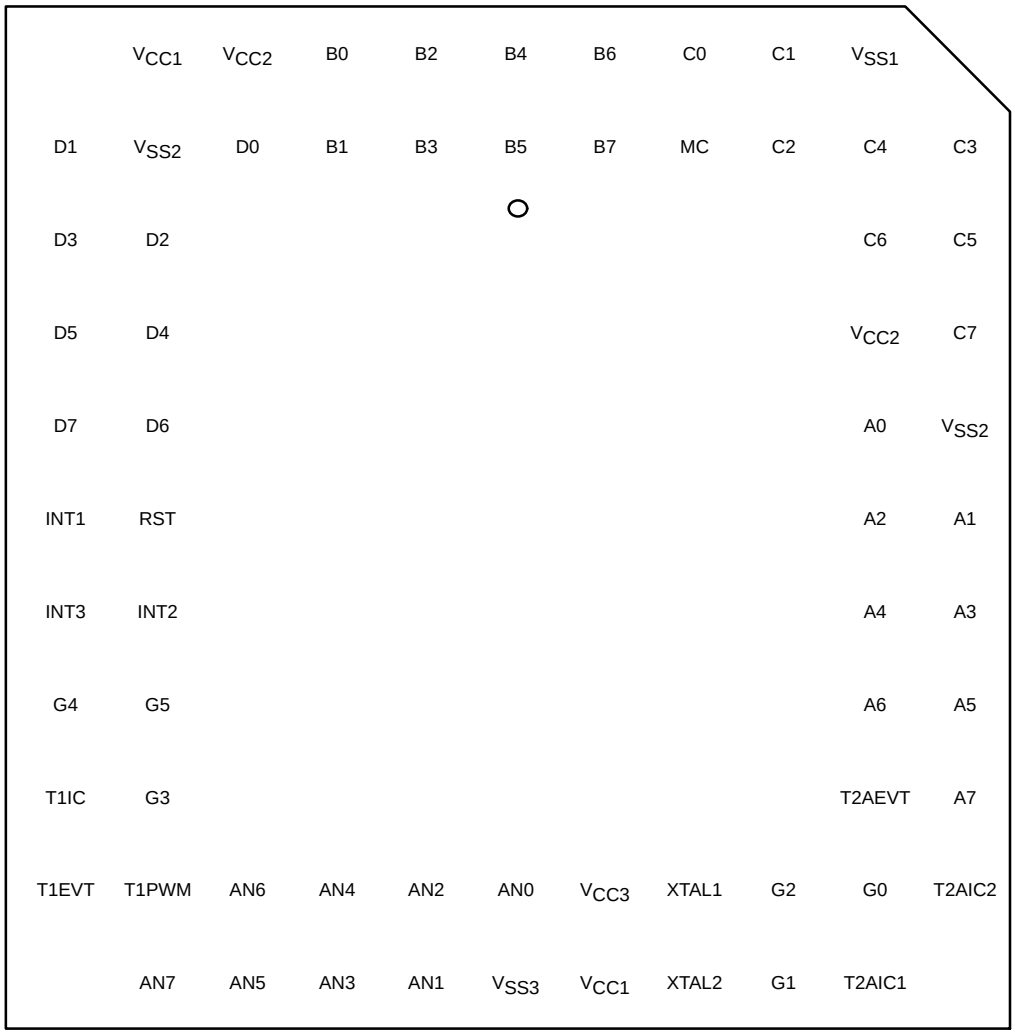
Bottom View

Figure H–14. TMS370Cx6x Device PGA Pinout



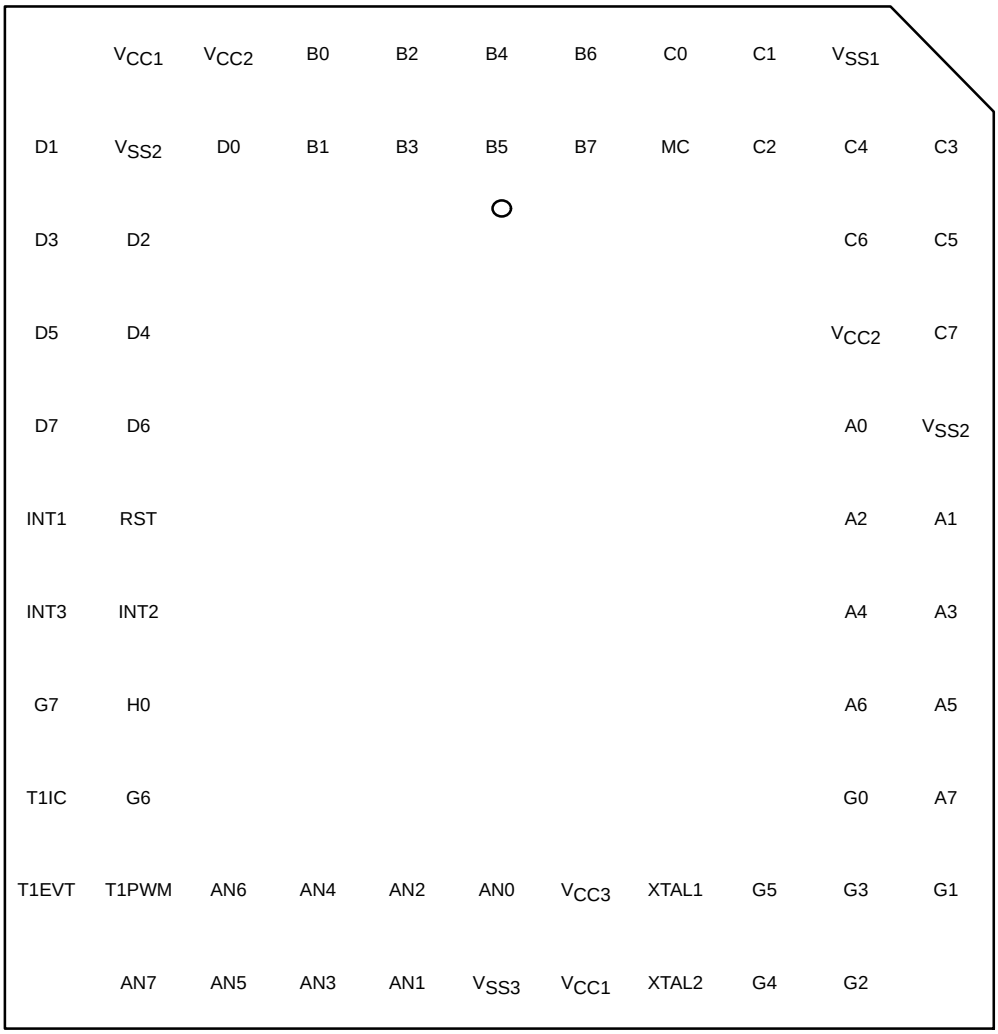
Bottom View

Figure H–15. TMS370Cx7x Device PGA Pinout



Bottom View

Figure H–16. TMS370CxBx Device PGA Pinout



Bottom View

Appendix I

PACT.H MACROS

The macros defined in the PACT.H file make it easier to define the initial values for the PACT command/definition area. You can obtain the latest version of this file from the microcontroller bulletin board. This appendix describes how to use these macros.

Topic **Page**

I.1 General Comments About Macros

The following subsections discuss generalities about macros including how to address commands and definitions in dual-port RAM, how to define output pins, and how to define actions.

I.1.1 Addressing Commands and Definitions in Dual-Port RAM

The initial value of the command/definition area is usually defined in program memory and then copied to dual-port RAM during the initialization routine. If the values in a command or definition are to be read from or written to while the PACT module is running, the runtime location of that command or definition must be known. Each of the six macros allows for an optional parameter (a register label) that, if passed as a symbol, equates the symbol to the register containing the least significant byte of the command or definition. For example, you can use the MOVW instruction with the register label parameter to change the compare value of a standard compare command from its initial value of 80h to 100h.

```
stdcmp   80h,op2,enable|opp_act,pwm1len ;if the command
                                        ;looks like this
movw     #100h,pwm1len                  ;this modifies
                                        ;the compare value
```

Often, the code that reads from or writes to the command/definition area is in a separate file from the code that initializes this area. You can make references to specific commands or definitions by declaring the register label parameter as .globreg. Bytes other than the least significant byte in a command or definition can be referenced as offsets from the least significant byte. Likewise, individual bits can be referenced according to the definition of the least significant byte. For example, the byte containing the output pin value of the standard compare command and the bit that enables that command can be referenced in this way:

```
         .globreg pwm1len           ;PWM 1 compare value
pwm1pin  .equ pwm1len–2             ;PWM 1 output pin byte
pwm1en   .dbit 3,pwm1len–3          ;PWM 1 enable bit
```

The assembler requires that you declare the global register symbols used in equates are .globreg before using them. You can do this by creating a file that has all of the .globreg symbols at the top, followed by the equates and the bit definitions. This file then can be included at the beginning of the file that defines the command/definition area and again at the beginning of each file that references that area. Using this technique, you have only to reassemble the module that defines the command/definition area. If this area changes, modules that reference specific commands or definitions are corrected at link time.

Before the macro is invoked, define the symbols cmd.stand table, so that the macros calculate the final destination of the command or definition.

I.1.2 Defining Output Pins

The symbols OP1 through OP8 are defined in the PACT.H file to make it easier to read the output pin selected for a specific command. You can use the numbers one through eight or any previously defined symbol that equates to a value in this range. The macros automatically subtract one and then shift the bits into their proper place for the requested command.

I.1.3 Defining Actions

Twenty-three possible actions are equated to numeric values at the beginning of the PACT.H file. The commands and definitions, for which each action is valid, are shown in the comment section beside the equate statement. The | operator concatenates multiple actions. The numeric value assigned to each action is valid only when used in the macro expansion. These symbols should not be used to set or clear the action bits while the PACT module is running.

I.2 Comments About Specific Macros

This section provides specific information about the following commands and definitions:

- Standard compare command
- Conditional compare command
- Virtual timer definition
- Baud timer definition

I.2.1 Standard Compare Command

It is not necessary to specify the compare value or the output pin if the enable action is not specified. For example, you can use the standard compare command as a dummy command so that a definition may follow as shown below.

```
stdcmp ,,nxt_def ;dummy command, next line is a definition
```

I.2.2 Conditional Compare Command

The time compare value that is passed to this macro is reduced by two before it is encoded into the command. This allows the value that is passed to the macro to more accurately reflect the time delay until the specified actions occur. The time compare value must be greater than or equal to two.

I.2.3 Virtual Timer Definition

The virtual timer period value passed to the macro is reduced by two so that the desired period is achieved. This value must be able to be represented in the maximum value format described in subsection 15.5.2 on page 15-17, *Virtual Timers*. If you want to have the macro truncate the value to fit into the maximum value format without generating an error, comment out the appropriate error lines in the PACT.H file.

The initial timer value is optional; if it is used, it must be an even number.

I.2.4 Baud Timer Definition

The maximum count value for this definition is derived by the equation given in 15.9 on page 15-32. The value obtained must then meet the maximum value format, or an error will be generated.

The initial timer value is optional; if it is used, it must be an even number.

I.3 PACT.H Macros

This section describes a file that contains macro definitions for all PACT commands.

Example I–1. Macro Definitions

```
;This file contains macro definitions for all PACT commands and definitions.
;All the actions desired in each of the commands/definitions must be passed
;in the macro as they are defined in the following equate table. All the
;actions are passed as one parameter in the macro. These actions are
;concatenated by '|' to form one parameter. These actions can be defined in
;any order.
;
;NOTE: If an action, which is not a valid action for a particular command
;      or definition, is used in that command, incorrect assembly may occur
;      without flagging an error.
;
;If you want to use different action names, the equate table must be
;modified.
;
;                                          Version 1.00
;OUTPUT PINS
op1       .EQU  1
op2       .EQU  2
op3       .EQU  3
op4       .EQU  4
op5       .EQU  5
op6       .EQU  6
op7       .EQU  7
op8       .EQU  8
```

Example I–1. Macro Definitions (Continued)

```
; ACTIONS            VTD BRD OTD SCC CCC DEC
clr_pin    .EQU 0   ;                x   x       Default condition
clr_evt1   .EQU 0   ;                        x   Default condition
nxt_def    .EQU 1   ;                x   x   x   Next entry is a def
int_cmp    .EQU 2   ;                x   x       Interrupt on compare =
int_evt1   .EQU 2   ;                        x   Interrupt on event 1
int_trst   .EQU 4   ;  x              x           Interrupt on timer = 0
enable     .EQU 8   ;  x        x    x       x   Enable timer or pin
rst_def_tmr  .EQU 10h ;              x           Reset def tmr on evt max
rst_def_ev2  .EQU 10h ;                      x   Reset def tmr on evt 2
set_pin    .EQU 20h ;                x   x       Set output pin on =
set_evt1   .EQU 20h ;                        x   Set output pin on evt1
step       .EQU 40h ;            x   x       x   Go to half resolution
int_evt    .EQU 80h ;            x               Interrupt on each event
int_max_evt  .EQU 100h;          x               Interrupt on max event
opp_act    .EQU 200h;                x           xOpp action on timer rst
int_evt2   .EQU 400h;                        x   Int on event 2
tx         .EQU 800h;      x                     Use as TX bit rate
rx         .EQU 1000h;     x                     Use as RX bit rate
vir_cap    .EQU 2000h;           x               Cap virt timer each evt
cap_def_ev1 .EQU 2000h;                      x   Cap def timer on event 1
def_cap    .EQU 4000h;           x               Cap def timer on evt max
cap_def_ev2 .EQU 4000h;                      x   Cap def timer on event 2
evt_plus1  .EQU 8000h;                   x       Action on event plus 1
```

Example I–1. Macro Definitions (Continued)

```
;STANDARD COMPARE COMMAND
; stdcmp <compare value>,<pin>,<actions>,<register label>
; compare value: 16–bit timer compare value
; pin: Output pin selection. (D18–D20)
; Possible actions:enable,set_pin,clr_pin,int_cmp,step,
;                  nxt_def,int_trst,opp_act
; register label:  a symbol to be equated to the register containing the
;                  least significant byte of this command


STDCMP     .MACRO cmpval,pin,actions,lab
           .var   b1,b2,b3,b4
           .if    ((pin.v<1)|(pin.v>8))&((actions.v&enable)=enable)
** ERROR, pin selection is illegal **
           .endif
           .if    (actions.v&0FD90h)!=0
** ERROR, illegal action specified **
           .endif
           .asg   cmpval.v&0FFh,b1.v
           .asg   (cmpval.v>>8)&0FFh,b2.v
           .if    (pin.v<1)|(pin.v>8)
           .asg   1,pin.v
           .endif
           .asg   pin.v–1,pin.v
           .asg   actions.v&63h|pin.v<<2,b3.v
           .asg   actions.v&0Ch|actions.v>>8&2h,b4.v
           .byte  b1.v,b2.v,b3.v,b4.v
           .if    lab.l!=0
           .asg   cmd_st–$+table+4,b1.v
:lab:      .equ   r:b1.v:
           .endif
           .ENDM


;CONDITIONAL COMPARE COMMAND
; CONCMP <event compare value>,<time compare value>,<pin>,<actions>,
;          <register label>
;
; event compare value: 8–bit value compared to the event counter
; time compare value: 16–bit value compared to the referred timer
; pin: Output pin  (only pin 1–7 are valid)
; Possible actions: nxt_def,int_cmp,set_pin,clr_pin,evt_plus1
; register label: a symbol to be equated to the register containing the
;                 least significant byte of this command
```

Example I–1. Macro Definitions (Continued)

```
CONCMP     .MACRO evcmpval,cmpval,pin,actions,lab
           .var   b1,b2,b3,b4
           .if    (cmpval.v=0)|(cmpval.v=1)
** ERROR, compare value must be greater than 1 **
           .endif

           .asg   cmpval.v–2,cmpval.v
           .if    (pin.v>7)|(pin.v<0)
** ERROR, pin selection is illegal **
           .endif
           .if    (actions.v&07FDCh)!=0
** ERROR, illegal action specified **
           .endif
           .if    (evcmpval.v>255)|(evcmpval.v<0)
** ERROR, Event counter compare value out of range **
           .endif
           .asg   cmpval.v&0FFh,b1.v
           .asg   (cmpval.v>>8)&0FFh,b2.v
           .if    pin.v=0
           .asg   7,pin.v
           .else
           .asg   pin.v–1,pin.v
           .endif
           .asg   80h|actions.v&23h|pin.v<<2|actions.v>>9&40h,b3.v
           .asg   evcmpval.v,b4.v
           .byte  b1.v,b2.v,b3.v,b4.v
           .if    lab.l!=0
           .asg   cmd_st–$+table+4,b1.v
:lab:      .equ   r:b1.v:
           .endif
           .ENDM

;DOUBLE EVENT COMMAND
; DEVCMP <event value 1>,<event value 2>,<output pin>,<actions>,
;          <register label>
;
; event value 1: 8–bit value compared to the event counter
; event value 2: 8–bit value compared to the event counter
; pin: Output pin
; Possible actions: nxt_def,int_evt1,set_pin,clr_pin,step,opp_act,int_evt2
;                rst_def_ev2,cap_def_ev1,cap_def_ev2,enable,
; register label: a symbol to be equated to the register containing the
;                least significant byte of this command


DEVCMP     .MACRO e1cmpval,e2cmpval,pin,actions,lab
           .var   b1,b2,b3,b4
           .if    (e1cmpval.v>255)|(e1cmpval.v<0)
** ERROR, Event compare 1 value out of range **
           .endif
```

Example I–1. Macro Definitions (Continued)

```
          .if   (e2cmpval.v>255)|(e2cmpval.v<0)
** ERROR, Event compare 2 value out of range **
          .endif
          .asg  e1cmpval.v,b1.v
          .asg  e2cmpval.v,b2.v
          .if   (pin.v<1)|(pin.v>8)
          .asg  1,pin.v
** ERROR, pin selection is illegal **
          .endif
          .asg  pin.v–1,pin.v
          .if   (actions.v&09984h)!=0
** ERROR, illegal action specified **
          .endif
          .asg  actions.v&063h|pin.v<<2,b3.v
          .asg  actions.v&18h|actions.v>>8&66h|1,b4.v
          .byte b1.v,b2.v,b3.v,b4.v
          .if   lab.l!=0
          .asg  cmd_st–$+table+4,b1.v
:lab:     .equ  r:b1.v:
          .endif
          .ENDM


;VIRTUAL TIMER DEFINITION
; virtmr <period>,<actions>,<initial timer value>,<register label>
;
; period: The period of the virtual timer, the maximum count plus 1
; Possible actions: enable,int_trst
; initial timer value: 16–bit virtual timer initial value.
; register label: a symbol to be equated to the register containing the
;                 least significant byte of this definition


VIRTMR    .MACRO period,actions,tmrval,lab
          .var  b1,b2,b3,b4
          .if   (period.v=0)|(period.v=1)
** Error, Max Timer value must be greater than 2 **
          .endif
          .if   (actions.v&0FFF3h)!=0
** ERROR, illegal action specified **
          .endif
          .asg  period.v–2,period.v
          .asg  tmrval.v&0FEh,b1.v
          .asg  (tmrval.v>>8)&0FFh,b2.v
          .if   ((period.v>>8)&0FFh) > 1Fh
          .asg  (period.v>>9)&70h|(period.v<<3)&80h|08h,b3.v
          .if   (period.v&0Fh)!=0
** ERROR, Max. Timer value truncated in last 4 bits **
          .endif
          .else
```

Example I–1. Macro Definitions (Continued)

```
          .asg  (period.v<<3)&0F0h|(actions.v&0Ch)>>1,b3.v
          .if   period.v&01h!=0
** ERROR, Max. Timer value truncated in last bit **
          .endif
          .endif
          .if   tmrval.v&01h!=0
** ERROR, Timer value truncated in last bit **
          .endif
          .asg  b3.v|(actions.v&0Ch)>>1,b3.v
          .asg  (period.v>>5)&0FFh,b4.v
          .byte b1.v,b2.v,b3.v,b4.v
          .if   lab.l!=0
          .asg  cmd_st–$+table+4,b1.v
:lab:     .equ  r:b1.v:
          .endif
          .ENDM


;BAUD TIMER DEFINITION
; BRTMR <maximum count>,<actions>,<initial timer value>,<register label>
;
; maximum count: number that determines the baud rate
; initial timer value: 16–bit virtual timer initial value
; Possible actions: RX,TX
; register label: a symbol to be equated to the register containing the
;               least significant byte of this definition


BRTMR     .MACRO maxcount,actions,tmrval,lab
          .var  b1,b2,b3,b4
          .if   ((actions.v&0E7FFh)!=0)
** ERROR, illegal action specified **
          .endif
          .asg  tmrval.v&0FEh,b1.v
          .asg  (tmrval.v>>8)&0FFh,b2.v
          .if   ((maxcount.v>>8)&0FFh) > 1Fh
          .asg  (maxcount.v>>9)&70h|(maxcount.v<<3)&80h|08h,b3.v
          .if   maxcount.v&0Fh!=0
** ERROR, Max. Timer value truncated in last 4 bits **
          .endif
          .else
          .asg  (maxcount.v<<3)&0F0h,b3.v
          .if   maxcount.v&01h!=0
** ERROR, Max. Timer value truncated in last bit **
          .endif
          .endif
          .if   tmrval.v&01h!=0
** ERROR, Timer value truncated in last bit **
          .endif
          .asg  (maxcount.v>>5)&0FFh,b4.v
```

Example I–1. Macro Definitions (Continued)

```
   .asg  b3.v|((actions.v&1800h)>>10)|1,b3.v
   .byte b1.v,b2.v,b3.v,b4.v
   .if lab.l!=0
   .asg  cmd_st–$+table+4,b1.v
:lab: .equ  r:b1.v:
   .endif
   .ENDM


;OFFSET TIMER DEFINITION
; OFSTMR <max event count>,<actions>,<initial value>,<register label>
;
; max event count: The maximum value the event counter may reach before
;     being reset.
; Possible actions: step,int_max_evt,enable,rst_def_tmr,
;     vir_cap,def_cap,int_evt
; initial value: 16–bit initial timer value
; register label: a symbol to be equated to the register containing the
;     least significant byte of this definition


OFSTMR .MACRO maxcount,actions,tmrval,lab
   .var  b1,b2,b3,b4
   .if(maxcount.v>255)|(maxcount.v<0)
** ERROR, Maximum event value out of range **
   .endif
   .if((actions.v&09E27h)!=0)
** ERROR, illegal action specified **
   .endif
   .asg  (tmrval.v&0FFh|1),b1.v
   .asg  (tmrval.v>>8)&0FFh,b2.v
   .asg  (actions.v&090h)|((actions.v&8)>>1)|(actions.v&40h)>>6,b3.v
   .asg  b3.v|((actions.v&100h)>>7)|((actions.v>>8)&60h),b3.v
   .asg  maxcount.v&0FFh,b4.v
   .byte b1.v,b2.v,b3.v,b4.v
   .if lab.l!=0
   .asg  cmd_st–$+table+4,b1.v
:lab: .equ  r:b1.v:
   .endif
   .ENDM
```

Appendix J

Glossary

A

absolute addressing mode: An addressing mode in which code or operands produce the actual address.

addressing mode: The method by which an instruction calculates the location of its required data.

analog-to-digital converter (ADC): An 8-bit successive-approximation converter with internal sample-and-hold circuitry. Also designated as ADC1 (8-channels except the 40-pin TMS370Cx4x devices which have only 4 channels), ADC2 (4-channels), and ADC3 (15-channels).

ANSI C: A version of the C programming language that conforms to the C standards defined by the *American National Standards Institute.*

archiver: A software program that allows you to collect several individual files into a single file called an archive library. The archiver also allows you to add, delete, extract, or replace members of the archive library.

assembler: A software program that creates a machine-language program from a source file containing assembly language instructions, directives, and macro directives. The assembler substitutes absolute operation codes for symbolic operation codes, and absolute or relocatable addresses for symbolic addresses.

assembly language: A symbolic language that describes the binary machine code in a more readable form. Each of the 73 unique instructions of the TMS370 family converts to one machine operation.

asynchronous communications mode: An SCI mode that needs no synchronizing clock. This format consists of a start bit followed by data bits, an optional parity bit, and a stop bit. This format is commonly used with RS-232-C communications and PC serial ports.

B

baud: The communication speed for serial ports; equivalent to bits per second.

BCD: *Binary coded decimal.* Each 4-bit nibble expresses a digit from 0–9 and usually packs two digits to a byte, giving a range of 0–99.

breakpoint, trace, and timing (BTT) features: A set of features supported by the BTT board (included with the XDS/22 emulation system). These features allow you to set hardware breakpoints, collect trace samples, and perform timing analysis.

buffer pointer: A 5-bit register in the PACT module peripheral frame that points to the next available location in the circular capture buffer.

byte: A sequence of 8 adjacent bits operated upon as a unit.

C

C: A high-level, general-purpose programming language that is useful for writing compilers and operating systems and for programming microprocessors.

C compiler: A program that translates C source statements into assembly language source statements.

capture register: A timer 2 register that is loaded with the 16-bit counter value on the occurrence of an external input transition. Either edge of the external input can be configured to trigger the capture.

chip select: For some blocks of the TMS370 memory map, the most significant bits of the address are pre-decoded to activate chip-select signals. These chip-select signals allow the TMS370 to access external addresses with a minimum of external logic and to perform memory bank selection under software control.

circular buffer: A variable length area in the PACT module dual-port RAM that stores the value of a PACT timer when a capture request is made. As new values are captured, they are put into successive locations in the buffer. When the buffer is full, the oldest captures are replaced with newer captures.

code conversion utility: A software program that translates a COFF object file into one of several standard ASCII hexadecimal formats suitable for loading into an EPROM programmer.

COFF: *Common Object File Format.* An implementation of the object file format of the same name developed by AT&T. The TMS370 compiler, assembler, and linker use and generate COFF files.

command/definition area: A variable length area in the PACT module dual-port RAM that is used to define the actions taken by the PACT module.

comment: A source statement (or portion of a source statement) that documents or improves the readability of a source file. Comments are not compiled, assembled, or linked; they have no effect on the object file.

CDT370: A low-cost code-development tool that is external to the target system and provides direct control over the TMS370 processor that is on the target system.

compare register: A timer 1 or timer 2 register that contains a value that is compared to the counter value. The compare function triggers when the counter matches the contents of the compare register.

constant: A value that does not change during execution.

CPU: An 8-bit register-oriented processor with a status register, program counter, and stack pointer. The TMS370 CPU uses the register file, accessed in one bus cycle, as working registers.

D

debugger: A window-oriented software interface that helps you to debug ’370 programs running on an emulator, CDT370, or design kit.

dedicated capture registers: An area in the PACT module dual-port RAM that stores the value of the default timer at the time of a specified edge on one of the PACT input capture pins. Unlike the circular buffer, the location of the dedicated capture register does not change.

default timer (also hardware timer): A 20-bit hardware counter in the PACT module that is incremented by the PACT prescaled clock.

design kit: A low-cost tool that allows you to analyze the hardware and software capabilities of the TMS370 family.

dual-port RAM: An area in RAM that can be read from and written to by both the TMS370 CPU and the PACT module.

E

edge detection: A type of circuitry that senses an active pulse transition on a given timer input and provides appropriate output transitions to the rest of the module. The active transition can be configured to be low-to-high or high-to-low.

EEPROM: *Electrically erasable programmable read-only memory.* Memory that can be programmed and erased under direct program control.

EPROM: *Erasable programmable read-only memory.* Memory that can be programmed under direct program control.

G

gang programmer: An interactive, menu-driven system that provides programming support for on-chip EEPROM or EPROM of the TMS370 microcontrollers in a production environment.

H

halt mode: An operating mode that reduces operating power by stopping the internal clock, which stops processing in all the modules. This is the lowest-power mode in which all register contents are preserved.

I

idle mode: An operating mode in which the CPU stops processing and waits for the next interrupt. It is not a low-power mode.

immediate operand: An operand whose actual constant value is specified in the instruction and placed after the opcode in the machine code.

index: An 8-bit unsigned number added to a base address to give a final address.

instruction: The basic unit of programming that causes the execution of one operation; consists of an opcode and operands, along with optional labels and comments.

interrupt: A signal to the CPU that stops the flow of a program and forces the CPU to execute instructions at an address corresponding to the source of the interrupt. When the interrupt is finished, the CPU resumes execution at the point where it was interrupted.

isosynchronous communications mode: An SCI mode in which data transmission is synchronized by a clock signal (SCICLK) common to both the sender and receiver. The format is identical to the asynchronous mode and consists of a start bit, data bits, an optional parity bit, and a stop bit. The term “isochronous” is interchangeable with isosynchronous.

L

label: A symbol that begins in column 1 of a source statement and corresponds to the address of that statement.

linker: A software tool that combines object files to form an object module that can be allocated into system memory and executed by the devices.

low-power mode: An operating mode that reduces operating power by reducing or stopping the activity of various modules. There are two low-power modes: halt and standby.

LSB: *Least significant bit.*

LSbyte: *Least significant byte.*

M

machine code: The actual bytes read by the CPU during an instruction execution; usually read by a programmer as hexadecimal bytes.

MC pin: *Mode control pin.* The pin that determines the operating mode of the TMS370 device, depending on the voltage applied to the pin. Twelve volts on the MC pin after reset places the processor in the write-protection override (WPO) mode.

memory map: A description of the addresses of the various sections and features of the TMS370 processor. The map depends on the operating mode.

microcomputer mode with external expansion: An operating mode in which the address, control, and data memory extend off-chip to access external memory or peripherals.

microcomputer single-chip mode: An operating mode in which the device uses only on-chip memory.

microcontroller programmer: An interactive, menu-driven system that provides a method of programming TMS370 family devices and EPROMs directly or through an XDS.

microprocessor mode with internal program memory: An operating mode in which the on-chip program memory is available to the processor.

microprocessor mode without internal program memory: An operating mode in which the on-chip program memory is not available to the processor. The processor must have external memory.

mini-SCI: The mini-UART function available in the PACT module.

mnemonic: A symbol that represents the opcode part of an assembly language instruction.

MSB: *Most significant bit.*

MSbyte: *Most significant byte.*

multiprocessor communications: An SCI format option that enables one processor to efficiently send blocks of data to other processors on the same serial link.

N

NCRF: New Code Release Form. A form that provides TI with information on the custom features of the device (e.g., prototype and production quantities and dates, exceptions to standard electrical specifications, part numbers, symbology, package type, etc.).

nested interrupts: The ability of an interrupt to suspend the service routine of a prior interrupt. Nested interrupts are implemented in TMS370 devices by executing an interrupt service routine that uses the EINT, EINTL, or EINTH instructions to set the global interrupt enable bits in the status register.

New Code Release Form: See NCRF.

nonmaskable interrupt (NMI): An interrupt that causes the processor to execute the NMI routine. On TMS370 devices, INT1 can be configured as an NMI.

NRZ (nonreturn to zero) format: A communication format in which the inactive state is a logic state.

O

offset: A signed value that is added to the base operand to give the final address.

opcode: *Operation code.* The first byte of the machine code that describes to the CPU the type of operation and combination of operands. Some TMS370 instructions use 16-bit opcodes.

operand: The part of an instruction that tells the programmer where the CPU will fetch or store data.

OTP: *One-time programmable*. A memory that can be programmed only once (compared to an EPROM that is erasable and can be reprogrammed).

P

PACT: *Programmable acquisition and control timer module.* A timer coprocessor module for the TMS370 microcontroller family.

peripheral file (PF): The 128 or 256 bytes of memory, starting at 1000h, that contain the registers that control the on-board peripherals and system configuration.

peripheral file frames: A set of sixteen contiguous peripheral file registers, usually related by function.

PPM: *Pulse-position modulation.* A serial signal in which the information is contained in the frequency of a signal with a constant pulse width. By using the timer compare features, a TMS370 device can output a PPM signal with a constant duty cycle without any program intervention.

prescaler: A circuit that slows the rate of a clocking source to the counter. The timer 1 prescaler can slow the clocking source by a factor of 4, 16, 64, or 256.

privilege mode: A mode immediately following reset in which the program can alter the privileged registers and bits. Once the privilege mode is disabled, these registers cannot be changed before another reset. This mode does not affect the EEPROM or the watchdog registers.

program counter (PC): A CPU register that identifies the current statement in the program.

prototyping device: A device used before mask-ROM devices are available that has identical functions, pinout, size, and timings to those of the actual device. Programmable memory such as EEPROM or EPROM is used in place of the masked ROM.

pulse accumulation: A timer 1 or timer 2 mode that keeps a cumulative count of SYSCLK pulses gated by the T1EVT or T2EVT signal.

PWM: *Pulse-width modulation.* A serial signal in which the information is contained in the width of a pulse of a constant frequency signal. By using the timer compare features, a TMS370 device can output a PWM signal with a constant duty cycle without any program intervention.

R

RAM: *Random access memory.*

ratiometric conversion: An analog-to-digital conversion in which the conversion value is a ratio of the V_{REF} source to the analog input. As V_{REF} is increased, the input voltage needed to give a certain conversion value changes; however, all conversion values keep the same relationship to V_{REF}.

referred timer: The timer that a PACT command uses for time comparisons. This is the last timer defined in the PACT command/definition area before the command was encountered, or if no timer has been defined, it is the least significant 16 bits of the hardware timer.

register file (RF): The first 128 or 256 bytes of memory that can be accessed by the majority of the instructions.

relative addressing mode: An operating mode in which operands and code produce an absolute address at some distance from the current location.

$\overline{\text{RESET}}$ pin: A pin that, when held low, starts hardware initialization and insures an orderly software startup. If the MC pin is low when the $\overline{\text{RESET}}$ signal returns high, then the processor enters the microcomputer mode. If the MC pin is high when the $\overline{\text{RESET}}$ signal returns high, then it enters the microprocessor mode.

ROM security: Inhibits the reading of the ROM data using any programmer.

S

serial communications interface (SCI): SCI is referred to as SCI1 or SCI2. SCI1 is a built-in serial interface that can be programmed to be asynchronous or isosynchronous. SCI2 is also a built-in serial interface that can only be programmed to be asynchronous. Many timing, data format, and protocol factors are programmable and controlled by the SCI module in operation.

serial peripheral interface (SPI): A built-in serial interface that facilitates communication between networked master and slave CPUs. As in the SCI, the SPI is set up by software; from then on, the CPU takes no part in timing, data format, or protocol.

signed integer: A number system used to express positive and negative integers.

stack: The part of the register file used as last-in, first-out memory for temporary variable storage. The stack is used during interrupts and calls to store the current program status. The area occupied by the stack is determined by the stack pointer and by the application program.

stack pointer (SP): An 8-bit CPU register that points to the last entry or top of the stack. The SP is automatically incremented before data is pushed onto the stack and decremented after data is popped from the stack.

standby mode: A power reduction mode in which the CPU stops processing, but the on-chip oscillator remains active. Timers remain active and can cause the CPU to exit the standby mode.

status register (ST): A CPU register that monitors the operation of the instructions and contains the global interrupt enable bits.

symbolic debugging: The ability of a software tool to retain symbolic information so that it can be used by a debugging tool such as an XDS/22, a design kit, or a CDT370.

symbol table: A portion of a COFF object file that contains information about the symbols that are defined and used by the file.

T

time slots: The internal cycles in which the PACT module can make a 32-bit access to the dual-port RAM. Each command or definition requires one, two, or three time slots. The number of time slots available is a function of the PACT prescaled clock and the frequency of access to the dual-port RAM by the CPU.

U

unsigned integer: A number system used to express positive integers.

V

virtual timer: An entry in the PACT command/definition area that creates an independent time base that is incremented by the PACT prescaled clock and cleared upon reaching a maximum value that is set by this definition.

W

WAIT pin: The pin that allows an external device to cause the processor to wait an indefinite number of clock cycles. When the wait line is released, the processor resynchronizes with the rising edge of the clockout signal and continues with the program.

wait states, automatic: Extra clock cycles inserted automatically on every external memory access to accommodate peripherals or expansion memory with slower access time than the TMS370 processor. These wait states are governed by two control bits: PF AUTOWAIT (SCCR0.5) and AUTOWAIT DISABLE (SCCR1.4).

watchdog timer: A timer option that can be programmed to generate an interrupt when it times out. This function serves as a hardware monitor over the software to prevent a “lost” program and is available in both the timer 1 and PACT modules. If timer 1 does not need a watchdog, this timer can be used as a general-purpose timer.

write protect override (WPO): The only mode in which a TMS370 device can modify the on-board EEPROM. The WPO mode is entered when external circuitry applies 12 volts to the MC pin after the device has been reset into one of its normal operating modes.

X

XDS/22: A code-development tool that is external to the target system and provides direct control over the TMS370 processor that is on the target system.

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

V

W

X

Z