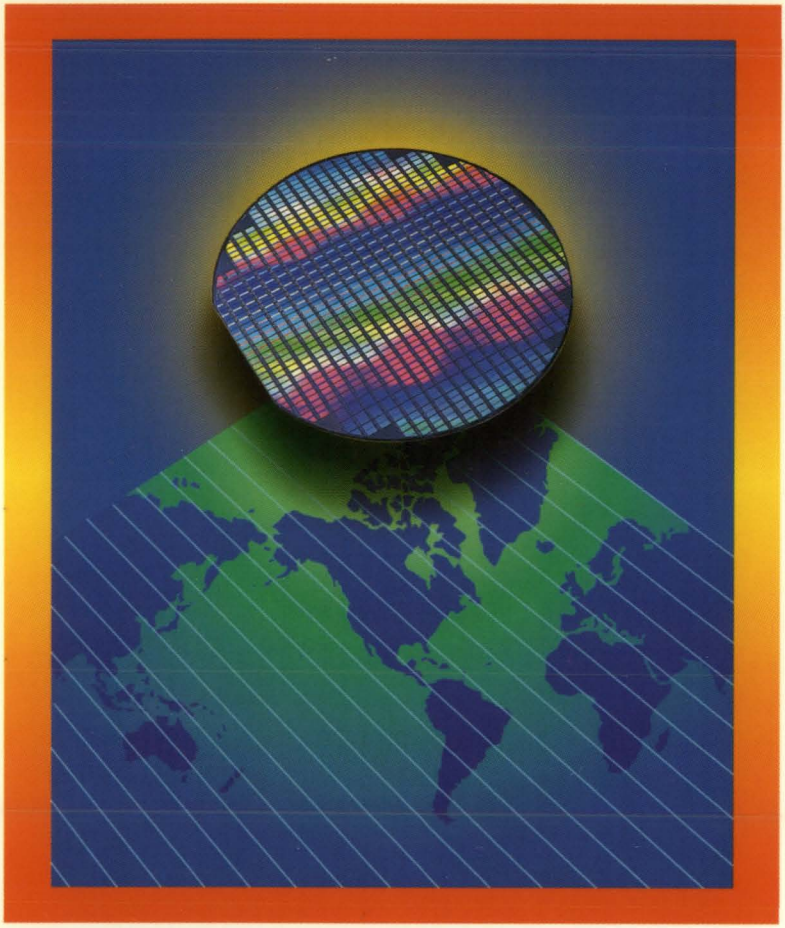
PIC16/17 MICROCONTROLLER DATA BOOK



THE EMERGING WORLD STANDARD™

1995/1996

MICROCHIP

MICROCHIP

Microchip PIC16/17 Microcontroller Data Book

SERVING A COMPLEX AND COMPETITIVE WORLD WITH FIELD-PROGRAMMABLE EMBEDDED CONTROL SYSTEM SOLUTIONS

1995 Microchip Technology Inc.

May 1995 /DS00158A

DATA SHEET MARKINGS

Microchip uses various data sheet markings to designate each document phase as it relates to the product development stage. The markings appear at the bottom of the data sheet, between the copyright and document and page numbers. The definitions for each marking are provided below for your use.

Marking	Description
Advanced Information	The information is on products in the design phase. Your designs should not be finalized with this information as revised information will be published when the product becomes available.
Preliminary	This is preliminary information on new products in production but not yet fully characterized. The specifications in these data sheets are subject to change without notice. Before you finalize your design, please ensure that you have the most current revision of the data sheet by contacting your Microchip sales office, representative or distributor.
No Marking	Information contained in the data sheet is on products in full production.

"Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Inc. with respect to the accuracy or use of such information, or infringement of patents arising from such use or otherwise. use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights."

PIC is a registered trademark of Microchip Technology Inc. in the U.S.A

PICMASTER and PICSTART are registered trademarks of Microchip Technology Inc.

PRO MATE, PICSEE, TrueGauge name and logo, *fuzzy*LAB, Smart Serial, Total Endurance, and UniMouse are trademarks of Microchip Technology Inc.

SQTP is a service mark of Microchip Technology Inc.

The Microchip logo and name, The Emerging World Standard, and The Embedded Control Solutions Company are trademarks of Microchip Technology Inc.

All rights reserved. Copyright © 1995, Microchip Technology, Inc.

ACCESS.bus is a trademark of the ACCESS.bus Industry Group (ABIG). *fuzzy*TECH is a registered trademark of Inform Software Corporation. I^2C is a trademark of Philips Corporation. IBM, IBM PC-AT, IBM PC/XT, and PS/2 are registered trademarks of International Business Machines Corp. Pentium is a trademark of Intel Corporation. MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows and Excel are trademarks of Microsoft Corporation. CompuServe is a registered trademark of CompuServe Incorporated. Apple and Macintosh are registered trademarks of Apple Computer, Inc. Apple Desktop Bus (ADB) is a trademark of Apple Computer Inc. Microwire is a trademark of National Semiconductor.

All other trademarks mentioned herein are property of their respective companies.

© 1995 Microchip Technology Inc

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

Table of Contents

© 1995 Microchip Technology Inc.

MICROCHIP

Table of Contents (Continued)

© 1995 Microchip Technology Inc.

SECTION 1
MICROCHIP TECHNOLOGY INC. COMPANY PROFILE

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

Microchip Technology Inc.

Company Profile

INTRODUCTION TO *THE EMBEDDED CONTROL SOLUTIONS COMPANY*™

Microchip Technology's mission is to offer leadership semiconductor products for embedded control system applications. To do this we have focusedSystem Management Bus our technology, engineering, manufacturing and marketing resources on two synergistic product lines: 8-bit PIC16/17 microcontrollers and Serial EEPROMS. These product lines provide the solutions to many of the problems facing designers of embedded control systems.

We publish the Microchip *Data Book* and *Embedded Control Handbook* to assist our customers, existing and new, in their efforts to design and produce state-of-the-art embedded control systems.

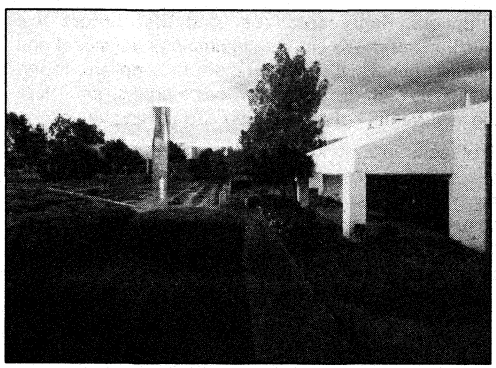
HIGHLIGHTS

Inside Microchip Technology you'll find:

- A focus on providing high-performance, cost-effective, field-programmable embedded control solutions
- An experienced executive team focused on innovation and committed to listening to our customers
- 8-bit RISC field-programmable microcontrollers and supporting logic products
- Serial and Parallel EEPROMs and EPROMs
- A variety of end-user Application-Specific Standard Products
- Fully integrated manufacturing capabilities
- A global network of manufacturing and customer support facilities
- A unique corporate culture dedicated to continuous improvement
- Distributor network support worldwide including certified distribution FAEs

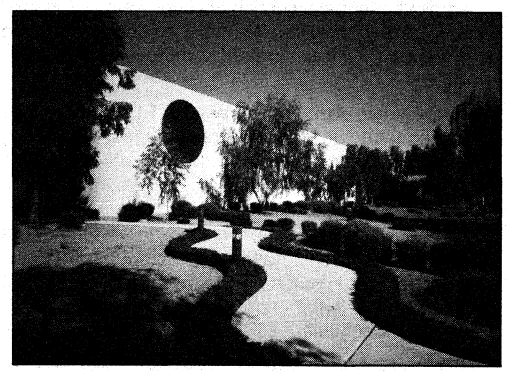
BUSINESS SCOPE

Microchip Technology Inc. manufactures and markets a variety of VLSI CMOS semiconductor components to support the market for cost-effective embedded control solutions. In particular, the company specializes in highly integrated, field-programmable RISC microcontrollers, application-specific standard products and related non-volatile memory products to meet growing market requirements for high performance, yet economical embedded control capability in products. Microchip's products feature the industry's most economical OTP (one-time programmable), reprogrammable EEPROM and ROM capability, along with the compact size, integrated functionality, ease of development and technical support so essential to timely and cost-effective product development by our customers.



Chandler, Arizona:

Company headquarters near Phoenix, Arizona; executive offices, R & D and wafer fabrication occupy this 142,000-square-foot facility. An additional 100,000 square foot adjacent facility is under construction with completion expected in mid-1995.



Tempe, Arizona:

Microchip's 170,000-square-foot wafer fabrication facility provides increased manufacturing capacity today and for the future.

© 1995 Microchip Technology Inc.

MICROCHIP

- Mission Statement -

Microchip Technology Incorporated is a leading supplier of field-programmable embedded control solutions by providing RISC microcontrollers and related non-volatile memory products. In order to contribute to the ongoing success of customers, shareholders and employees, our mission is to focus resources on high value, high quality products and to continuously improve all aspects of our business, providing a competitive return on investment.

- Guiding Values -

Customers Are Our Focus: We establish successful customer partnerships by exceeding customer expectations for products, services and attitude. We start by listening to our customers, earning our credibility by producing quality products, delivering comprehensive services and meeting commitments. We believe each employee must effectively serve their internal customers in order for Microchip's external customers to be properly served.

Quality Comes First: We will perform correctly the first time, maintain customer satisfaction and measure our quality against requirements. We practice effective and standardized improvement methods, such as statistical process control to anticipate problems and implement root cause solutions. We believe that when quality comes first, reduced costs follow.

Continuous Improvement Is Essential: We utilize the concept of "Vital Few" to establish our priorities. We concentrate our resources on continuously improving the Vital Few while empowering each employee to make continuous improvements in their area of responsibility. We strive for constructive and honest self-criticism to identify improvement opportunities.

Employees Are Our Greatest Strength: We design jobs and provide opportunities promoting employee teamwork, productivity, creativity, pride in work, trust, integrity, fairness, involvement, development and empowerment. We base recognition, advancement and compensation on an employee's achievement of excellence in team and individual performance. We provide for employee health and welfare by offering competitive and comprehensive employee benefits.

Products And Technology Are Our Foundation: We make ongoing investments and advancements in the design and development of our manufacturing process, device, circuit, system and software technologies to provide timely, innovative, reliable and cost effective products to support current and future market opportunities.

Total Cycle Times Are Optimized: We focus resources to optimize cycle times to our internal and external customers by empowering employees to achieve efficient cycle times in their area of responsibility. We believe that cycle time reduction is achieved by streamlining processes through the systematic removal of barriers to productivity.

Safety Is Never Compromised: We place our concern for safety of our employees and community at the forefront of our decisions, policies and actions. Each employee is responsible for safety.

Profits And Growth Provide For Everything We Do: We strive to generate and maintain competitive rates of company profits and growth as they allow continued investment for the future, enhanced employee opportunity and represent the overall success of Microchip.

Communication Is Vital: We encourage appropriate, honest, constructive, and ongoing communication in company, customer and community relationships to resolve issues, exchange information and share knowledge.

Suppliers, Representatives, And Distributors Are Our Partners: We strive to maintain professional and mutually beneficial partnerships with suppliers, representatives, and distributors who are an integral link in the achievement of our mission and guiding values.

Professional Ethics Are Practiced: We manage our business and treat customers, employees, shareholders, investors, suppliers, distributors, representatives, community and government in a manner that exemplifies our honesty, ethics and integrity. We recognize our responsibility to the community and are proud to serve as an equal opportunity employer.

© 1995 Microchip Technology Inc.

MARKET FOCUS

Microchip targets selected markets where our advanced designs, progressive process technology and industry-leading product performance enable us to deliver decidedly superior performance. The company has positioned itself to maintain a dominant role as a supplier of high-performance, field-programmable microcontrollers and associated memory and logic products for embedded control applications which are found throughout the consumer, automotive, telecommunication, office automation and industrial control markets.

FULLY INTEGRATED MANUFACTURING

Microchip delivers fast turnaround and consistent quality through total control over all phases of production. Research and development, design, mask making, wafer fabrication, and the major part of assembly and quality assurance testing are conducted at facilities wholly-owned and operated by Microchip. Our integrated approach to manufacturing along with rigorous use of advanced statistical process control (SPC) and a continuous improvement culture has resulted in high and consistent yields which have positioned Microchip as a quality leader in its global markets. Microchip's unique approach to SPC provides customers with excellent costs, quality, reliability and on-time delivery.

A GLOBAL NETWORK OF PLANTS AND FACILITIES

Microchip is a global competitor providing local service to the world's technology centers. The Company's focal point is its design and technology advancement facility in Chandler, Arizona. Product and technology development is located here, along with front-end wafer fabrication and wafer probe and sort.

In 1994, Microchip purchased a second wafer fabrication facility in Tempe, Arizona – thirteen miles from its Chandler, Arizona, headquarters. The additional 170,000 square foot facility is meeting production requirements beyond those which could be produced in Microchip's Chandler wafer facility. Assembly and test facilities predominantly located in Kaohsiung, Taiwan, and Bangkok, Thailand, house the technology and assembly and test equipment necessary for modern plastic and ceramic packaging.

Sales and application offices are located in key cities throughout the Americas, Asia/Pacific, Japan and Europe. Offices are staffed to meet the high quality expectations of our customers, and can be accessed for technical and business support.

EMBEDDED CONTROL OVERVIEW

Unlike "processor" applications such as personal computers and workstations, the computing or controlling elements of embedded control applications are buried inside the application. The user of the product is only concerned with the very top-level user interface (such as keypads, displays and high-level commands). Very rarely does an end-user know (or care to know) the embedded controller inside (unlike the conscientious PC users, who are intimately familiar not only with the processor type, but also its clock speed, DMA capabilities and so on).

It is, however, most vital for designers of embedded control products to select the most suitable controller and companion devices. Embedded control products are found in all market segments: consumer, commercial, PC peripherals, automotive, telecommunications (including fast-emerging personal telecommunication products) and industrial. Most often embedded control products must meet special requirements: cost-effectiveness, low power, small footprint and a high level of system integration.

Typically, most embedded control systems are designed around a microcontroller which integrates on-chip program memory, data memory (RAM) and various peripheral functions, such as timers and serial communication. In addition, these systems also usually require Serial EEPROM memories, display drivers, keypads, small displays, etc.

Microchip Technology has established itself as a leading supplier of field-programmable embedded control solutions. The combination of high-performance microcontrollers from the PIC17CXX, PIC16CXX and PIC16C5X families, along with industry leading nonvolatile memory products provides the basis for this leadership.

Microchip is committed to continuous innovation and improvement in design, manufacturing and technical support to provide the best possible embedded control solutions to you.

© 1995 Microchip Technology Inc.

MICROCONTROLLERS

PIC16/17 microcontrollers from Microchip combine high performance, low cost and small package size, offering the best price/performance ratio in the industry. Over 200 million of these devices have been used in cost-sensitive consumer products, computer peripherals, office automation, automotive control systems, security and telecommunication applications.

PIC16/17 MICROCONTROLLER OVERVIEW AND ROADMAP

Microchip offers three families of 8-bit microcontrollers to best fit your needs:

- PIC16C5X: Base-Line 8-bit Family
- PIC16CXX: Mid-Range 8-bit Family
- PIC17CXX: High-End 8-bit Family

All families offer One-Time-Programmable, low-voltage and low-power options, as well as various packaging options. Selected members are available in ROM and reprogrammable versions.

The widely-accepted PIC16C5X, PIC16CXX and PIC17CXX families are the industry's only 8-bit microcontrollers using a high-speed RISC architecture. Microchip pioneered the use of RISC architecture to obtain high speed and instruction efficiency.

PIC16C5X: BASE-LINE FAMILY

PIC16C5X is the well established base-line family offering the most cost-effective solution. This PIC16C5X products have a 12-bit wide instruction set and are currently offered in 18-, 20- or 28-pin packages. In SOIC and SSOP packaging options, these are the smallest footprint controllers. Low-voltage operation down to 2.0V makes this family ideal for battery operated applications.

The PIC16C5X base-line family is in high-volume production, shipping more than two million units per week, and has achieved more than twenty-five thousand design wins worldwide.

PIC16CXX: MID-RANGE FAMILY

PIC16CXX mid-range family offers a wide-range of options, from 18-pin to 44-pin packages as well as low to high level of peripheral integration. This family has a 14-bit wide instruction set, interrupt handling capability and deeper 8-level hardware stack. The PIC16CXX family provides the performance and versatility to meet the requirements of more demanding, yet cost-sensitive, mid-range 8-bit applications.

The PIC16CXX mid-range family is rapidly gaining acceptance with several of its members introduced: PIC16C61, PIC16C62, PIC16C63, PIC16C64, PIC16C65, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C73, PIC16C74 and PIC16C84.

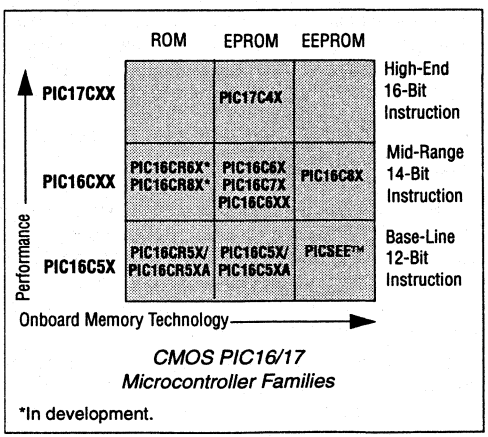
PIC17CXX: HIGH-END FAMILY

The PIC17CXX high-performance family offers the world's fastest execution performance of any 8-bit microcontroller family in the industry. The PIC17CXX family extends the PIC16/17 microcontroller's high-performance RISC architecture with a 16-bit instruction word, enhanced instruction set and powerful vectored interrupt handling capabilities. A powerful array of precise on-chip peripheral features provide the performance for the most demanding 8-bit applications.

Currently, two members of the PIC17CXX family have been announced. A third member will be available soon.

Current PIC16/17 microcontroller product families include advanced features such as sophisticated timers, embedded Analog-to-Digital Converter, extended instruction/data memory, inter-processor communication (I^2C™ bus, SPI and USARTs) and ROM, RAM, EPROM and EEPROM memories.

Both PIC16CXX and PIC17CXX families are supported by user-friendly development systems including assembler, software simulator, C Compiler, fuzzy logic development software, programmers and in-circuit emulators.



CMOS PIC16/17 Microcontroller Families

*In development.

© 1995 Microchip Technology Inc.

FIGURE 1: PIC16/17 MICROCONTROLLER MIGRATION PATH

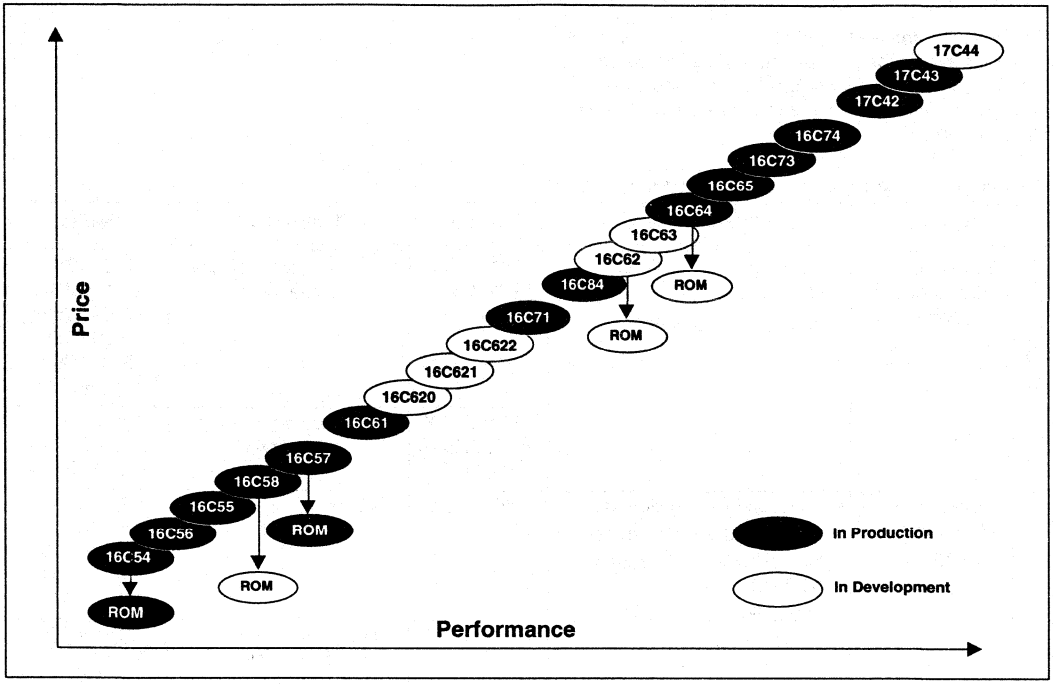


FIGURE 2: PIC16/17 SYNERGISTIC DEVELOPMENT TOOLS

Development Tool	Name	PIC16C5X	PIC16CXX	PIC17CXX
Assembler	MPASM	✔	✔	✔
Software Simulator	MPSIM	✔	✔	✔
C Compiler*	MP-C	✔	✔	✔
Entry Level Development Kit	PICSTART®	✔	✔	Planned
Universal Programmer	PRO MATE™	✔	✔	✔
Universal In-Circuit Emulator	PICMASTER®	✔	✔	✔
Fuzzy Logic Development Tool	fuzzyTECH®-MP	✔	✔	✔

* Available from Byte Craft Limited in Canada.

PIC16/17 NAMING CONVENTION

The PIC16/17 architecture offers users a wide range of cost/performance options of any 8-bit microcontroller family. In order to identify the families, the following naming conventions have been applied to the PIC16/17 microcontrollers.

TABLE 1: PIC16/17 NAMING CONVENTION

Family		Architectural Features	Name	Technology	Products
PIC16C5X	Base-Line 8-bit Microcontroller Family	• 12-bit wide instruction set • DC - 20 MHz clock speed • 200 ns instruction cycle	PIC16C5X PIC16C5XA (Note 1)	OTP program memory, digital only	PIC16C54 PIC16C54A PIC16C55 PIC16C56 PIC16C57 PIC16C58A
			PIC16CR5X PIC16CR5XA (Note 1)	ROM program memory, digital only	PIC16CR54 PIC16CR57A PIC16CR58A
PIC16CXX	Mid-Range 8-bit Microcontroller Family	• 14-bit wide instruction set • Internal/external interrupts • DC - 20 MHz clock speed (Note 3) • 200 ns instruction cycle (@ 20 MHz)	PIC16C6X	OTP program memory, digital	PIC16C61 PIC16C62 PIC16C63 PIC16C64 PIC16C65
			PIC16CR6X	ROM program memory, digital only	Planned
			PIC16C62X	OTP program memory with comparators	PIC16C620 PIC16C621 PIC16C622
			PIC16C7X	OTP program memory, with analog functions (e.g. A/D)	PIC16C71 PIC16C73 PIC16C74
			PIC16C8X	EEPROM program and data memory	PIC16C84
			PIC16CR8X	ROM program and EEPROM data memory	Planned
PIC17CXX	High-End 8-bit Microcontroller Family	• 16-bit wide instruction set • Internal/external interrupts • DC - 25 MHz clock speed • 160 ns instruction cycle	PIC17C4X	OTP program memory, digital only	PIC17C42 PIC17C43 PIC17C44
			PIC17CR4X	ROM program memory, digital only	Planned

Note 1: "A" designates a more advanced process technology, generally offering customers the benefits of lower power, higher speed, etc. (example: PIC16C54, PIC16C54A). Sometimes it designates additional functions such as the addition of Brown-out detect.

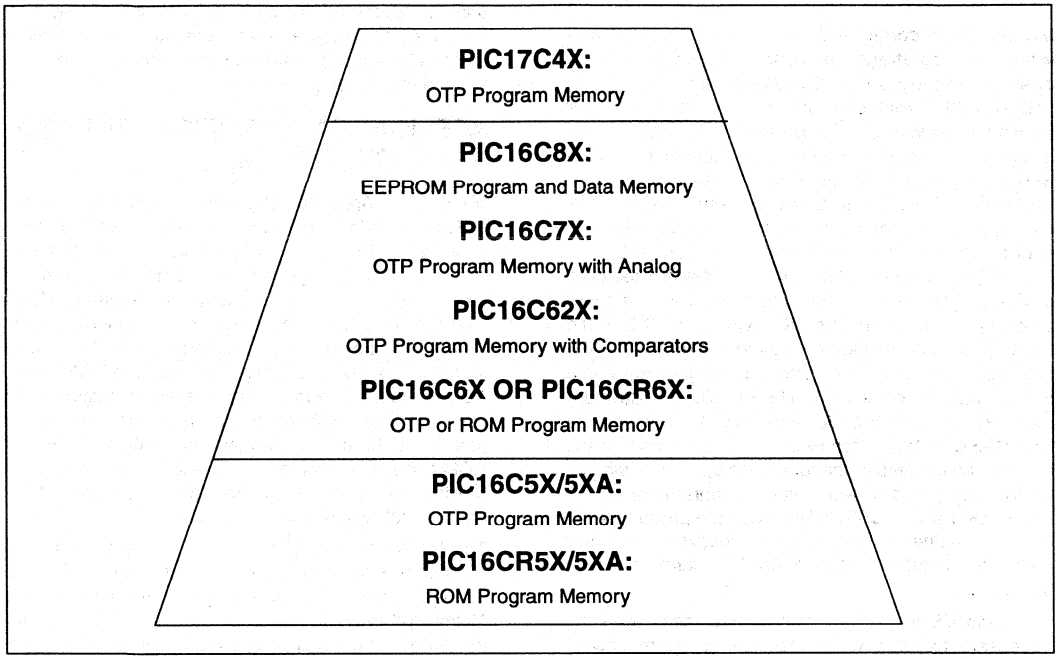
Note 2: The numbering system within each family is not necessarily significant.

Note 3: The maximum clock speed for some devices is less than 20 MHz.

Please check with your local Microchip distributor, sales representative or sales office for the latest product information.

© 1995 Microchip Technology Inc.

FIGURE 3: PIC16/17 8-BIT MICROCONTROLLER FAMILY



THE ADVANTAGE OF FIELD PROGRAMMABILITY

The PIC16/17 microcontroller family provides a unique combination of a high-performance RISC processor with cost-effective One-Time-Programmable (OTP) technology. Cost-effective OTP provides many benefits to the user at prices which can be comparable to competing ROM solutions. The benefits include: 1) quick time-to-market, 2) ease of code changes, 3) ability to provide adaptable solutions to end-customer requirements, 4) ability to meet upside potential via inventory positions at Microchip or worldwide distribution, 5) reduced scrappage in manufacturing, 6) reduced inventory in manufacturing, and 7) reduced work-in-process liability.

For most manufacturers, getting the product to market quickly has become the number one goal as global markets have become more competitive. Time-to-market puts pressure on all functions within the manufacturing process: development, purchasing, production, and marketing and sales. Field-programmable OTP technology streamlines the process for all stages in the product life cycle.

In the early product development stages, a programmable microcontroller allows much of the functionality to be implemented in software which can be modified more easily than hardware-only solutions.

In the manufacturing stage, the compression of the product life cycle curve puts pressure on the management of inventory and manufacturing cycle times. Minimizing inventory reduces the ability to meet upside demand. Using a traditional ROM-based microcontroller limits the ability to respond to the market with product enhancements or semi-customized products for specific customers. Using the standard OTP-based PIC16/17 microcontroller solves all these issues. Inventory can be managed effectively by using the same device in several systems. Costs can be reduced due to volume purchasing. Upsides can be met from either safety stock, directly from Microchip, or local distributors who regularly inventory all the PIC16/17 microcontroller devices. A sudden decline in demand means no work-in-process ROM-based inventory and any excess safety stock can be consumed by the other products using the same standard device.

OTP is the 'Flexible Manufacturing' technology of the microcontroller world. As competition intensifies, the demand for customer-specific products increases. Having the ability to change (for example, the appearance of LCD displays or add extra features in a timely manner) can be a key competitive advantage. Programming the OTP device on the manufacturing floor allows easy customizing and internal tracking of the devices for each specific customer. Customization can significantly increase the overall product life cycle to provide better return on investment and help minimize the threat of competition.

© 1995 Microchip Technology Inc.

DEVELOPMENT SYSTEMS

Microchip is committed to providing useful and innovative solutions to your embedded system designs. Among support products offered are the PICMASTER® real-time universal in-circuit emulator running under Windows™ environment. PICMASTER is designed to provide product development engineers with an optimized design tool for developing target applications. This universal in-circuit emulator provides a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. PRO MATE™, the full-featured device programmer, enables you to quickly and easily program user software into PIC16C5X, PIC16CXX and PIC17CXX CMOS microcontrollers. The PRO MATE operates as a stand-alone unit or in conjunction with a PC compatible host system. The PICSTART® development kit, a low-cost development system for the PIC16C5X/16CXX families of microcontrollers, includes an assembler for code development, a simulator for debug and a development programmer board. PICSEEKIT and PICSEESTART provide product development engineers with a cost-effective and timely design tool solution for the MTA8XXXX family of ASSP products.

The Serial EEPROM Designer's Kit includes everything necessary to read, write, erase or program special features of any Microchip Serial EEPROM products including *Smart Serials*™ and secure serials. The *Total Endurance*™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

The *TrueGauge*™ development tool supports system development with the MTA11200 TrueGauge Intelligent Battery Management IC.

SOFTWARE SUPPORT

Microchip's PIC16/17 microcontrollers families are supported by an assembler, compiler, software simulator and fuzzy logic development software. MPASM is a universal macro assembler supporting Microchip's entire product line of microcontrollers. MPSIM, a discrete event software simulator, is designed to imitate operation of PIC16C5X, PIC16CXX and PIC17CXX microcontrollers. It allows the user to debug software that will use any of these microcontrollers.

A full-featured C Compiler and Fuzzy Logic support are also available for all three microcontroller families.

Microchip endeavors at all times to provide the best service and responsiveness possible to its customers. The Microchip Systems Bulletin Board Service (BBS) is one service to facilitate this process. It's a multi-faceted tool that can provide you with information on a number of different topics. Special Interest Groups available through the BBS can provide you with the opportunity to discuss issues and topics of interest with others that share your interest or questions. The BBS is regularly used to distribute technical information, application notes, source codes, errata sheets, bug reports, interim patches for Microchip systems products and user contributed files for distribution.

APPLICATION-SPECIFIC STANDARD PRODUCTS (ASSPs)

Microchip's Application-Specific Standard Products (ASSP) provide value-added embedded control solutions by combining PIC16/17 microcontroller architecture, non-volatile memory and innovative software technology for vertical applications. These products incorporate technology that offers a complete solution that is both unique to the customer and standard in manufacture to Microchip. In addition, Microchip ASSPs reduce or remove the barriers for customers to use Microchip solutions in their products through the use of software embedded in secure OTP- or ROM-based microcontrollers. The family is packaged to provide the highest integration to the customer at the best overall system cost.

The MTA11200 family is the most accurate and most integrated battery management and charging solution available today. The TrueGauge family incorporates Microchip/SPAN patented technology which digitally integrates battery charge and discharge current to provide an accurate (>97% typical) state of charge indication. The family operates with NiCd and NiMH and lead acid battery packs from 3 VDC to 25 VDC. These products are ideal for portable PC, cellular phone and portable consumer product applications.

The MTA14000 programmable Intelligent Battery Management IC allows engineers to design intelligent controllers for smart batteries, battery chargers, battery status monitoring, uninterruptible power supplies, HVAC and other data acquisition and processing required for managing energy. The MTA14000's programmable 4K words of program memory and 192 bytes of RAM allows it to support any battery technology including Li Ion, NiMH, NiCd, Pb acid, Zinc Air. In addition, the products I^2C port enables any system OEM, battery pack VAR and battery manufacturer to design, build and market SBD-compliant products supporting the System Management Bus standard.

The MTE1122 Energy Management Controller combines Microchip's proprietary PIC16/17 8-bit RISC microcontroller technology with a unique, patent pending power management firmware algorithm in a single package. This device, by monitoring and controlling the supply requirements into an AC induction motor, effectively reduces the power consumed by the motor. The MTE1122 is available in both plastic DIP and space-saving SOIC packages, and operates over commercial and industrial ranges.

Ease-of-use, low voltage and low cost make the MTA41XXX mouse and trackball MCU firmware solutions ideal for implementing new designs for both PCs

© 1995 Microchip Technology Inc.

and Apple® computers. The products in the MTA41XXX family are 18-lead, low-power CMOS microcontroller ICs combined with application-specific software. By adding a few external components, the user can easily realize a complete mouse or trackball system.

The MTA8XXXX PICSEE™ family of cost-effective system solutions integrates PIC16/17 microcontrollers with EEPROM technology. These PICSEE™ devices are ideally suited for automotive security, keyless entry, remote control, data acquisition and telecommunication applications. The combined product assembly techniques provide the user the highest performance solution in a compact and cost-effective package.

Future ASSP products will include advanced features such as mixed analog and digital capability as well as an ever broadening family of turnkey software solutions for the embedded control market.

SERIAL EEPROM OVERVIEW

Microchip offers one of the broadest selections of CMOS Serial EEPROMs on the market for embedded control systems. Serial EEPROMs are available in a variety of densities, operating voltages, bus interface protocols, operating temperature ranges and space saving packages.

Densities:

Currently range from 1K to 64K with higher density devices in development.

Bus Interface Protocols:

All major protocols are covered: 2-wire, 3-wire and 4-wire.

Operating Voltages:

In addition to standard 5V devices there are two low voltage families. The "LC" devices operate down to 2.5V, while the breakthrough "AA" family operates, in both read and write mode, down to 1.8V, making these devices highly suitable for alkaline and NiCad battery powered applications.

Temperature Ranges:

Like all Microchip devices, Serial EEPROMs are offered in Commercial (0°C to 70°C), Industrial (-40°C to 85°C) and Automotive (-40°C to 125°C) operating temperature ranges.

Packages:

The focus is on small packages. Small footprint packages include: 8-lead DIP, 8-lead SOIC in JEDEC and EIAJ body widths, and 14-lead SOIC. The SOIC comes in two body widths; 150 mil and 207 mil.

In February 1995, Microchip announced its 10 million Erase/Write cycle guarantee - an endurance breakthrough unmatched by its competitors. The Company has also developed the world's first 64K Smart Serial EEPROM which provided four times the speed, four times the memory and four times the features of any competitive 2-wire Serial EEPROM. Device densities range from 256 bits up to 64K bits. Another first is the 24LC21, the only single chip DDC1/DDC2™-compatible solution for plug-and-play video monitors.

Microchip is a high-volume supplier of Serial EEPROMs to all the major markets worldwide including consumer, automotive, industrial, computer and communications. To date, more than 300 million units have been produced. Microchip continues to develop new Serial EEPROM solutions for embedded control applications.

Microchip's erase/write cycle endurance is among the best in the world, and only Microchip offers unique and powerful development tools such as the Total Endurance disk. This mathematical software model is an innovative tool used by system designers to optimize Serial EEPROM performance and reliability within the application.

PARALLEL EEPROM OVERVIEW

CMOS Parallel EEPROM devices from Microchip are available in 4K, 16K and 64K densities. The manufacturing process used for these EEPROMs ensures 10,000 to 100,000 write and erase cycles typical. Data retention is more than 10 years. Fast write times are less than 200 µs. These EEPROMs work reliably under demanding conditions and operate efficiently at temperatures from –40°C to +85°C. Microchip's expertise in advanced SOIC, TSOP and VSOP surface mount packaging supports our customers' needs in space-sensitive applications.

Typical applications include computer peripherals, engine control, pattern recognition and telecommunications.

OTP EPROM OVERVIEW

Microchip's CMOS EPROM devices are produced in densities from 64K to 512K. High-speed EPROMs have access times as low as 55 ns. Typical applications include computer peripherals, instrumentation and automotive devices. Microchip's expertise in surface mount Packaging on SOIC, TSOP and VSOP packages led to the development of the Surface Mount one-time-programmable (OTP) EPROM market where Microchip is a leading supplier today. Microchip is also a leading supplier of low-voltage EPROMs for battery powered applications.

© 1995 Microchip Technology Inc.

EASE OF PRODUCTION UTILIZING QUICK TURN PROGRAMMING (QTP) AND SERIALIZED QUICK TURN PROGRAMMING (SQTP℠)

Recognizing the needs of high-volume manufacturing operations, Microchip has developed two programming methodologies which make the OTP products as easy to use in manufacturing as they are efficient in the system development stage.

Quick Turn Programming allows factory programming of OTP product prior to delivery to the system manufacturing operation. PIC16/17, EPROM and Serial EEPROM products can be automatically programmed with the users program during the final stages of the test operation at Microchip's assembly and test operations in Philippine Islands, Taiwan and Thailand. This low-cost programming step allows the elimination of programming during system manufacturing and essentially allows the user to treat the PIC16/17 and memory products as custom ROM products. With one- to four-week lead times on QTP product, the user no longer needs to plan for the extended ROM masking lead times and masking charges associated with custom ROM products. This capability, combined with the off-the-shelf availability of standard OTP product, ensures the user of product availability and the ability to reduce his time-to-market once product development has been completed.

Unique in the 8-bit microcontroller market is Microchip's ability to enhance the QTP capability with Serialized Quick Turn Programming (SQTP). SQTP allows for the programming of devices with unique, random or serialized identification codes. As each PIC16/17 device is programmed with the customers program code, a portion of the program memory space can be programmed with a unique code, accessible from normal program memory, which will allow the user to provide each device with a unique identification. This capability is ideal for embedded systems applications where the transmission of key codes or identification of the device as a node within a network are essential. Taking advantage of this capability allows the system designer to eliminate the requirement for expensive off-chip code implementation using DIP switches or non-volatile memory components. The SQTP offering, pioneered by Microchip, provides the embedded systems designer with a low cost means of putting a unique and custom device into every system or node.

FUTURE PRODUCTS AND TECHNOLOGY

New process technology is constantly being developed for microcontroller, ASSP, EEPROM and high-speed EPROM products. Advanced process technology modules and products are being developed that will be integrated into present product lines to continue to achieve a range of compatible processes. Current production technology utilizes lithography dimensions down to 0.9 microns.

Microchip's research and development activities include exploring new process technologies and products that have industry leadership potential. Particular emphasis is placed on products that can be put to work in high-performance broad-based markets.

Equipment is continually updated to bring the most sophisticated process, CAD and testing tools online. Cycle times for new technology development are continuously reduced by using in-house mask generation, a high-speed pilot line within the manufacturing facility and continuously improving methodologies.

More advanced technologies are under development, as well as advanced CMOS RISC-based microcontroller, ASSP and CMOS EEPROM and EPROM products. Objective specifications for new products are developed by listening to our customers and by close cooperation with our many customer-partners worldwide.

© 1995 Microchip Technology Inc.

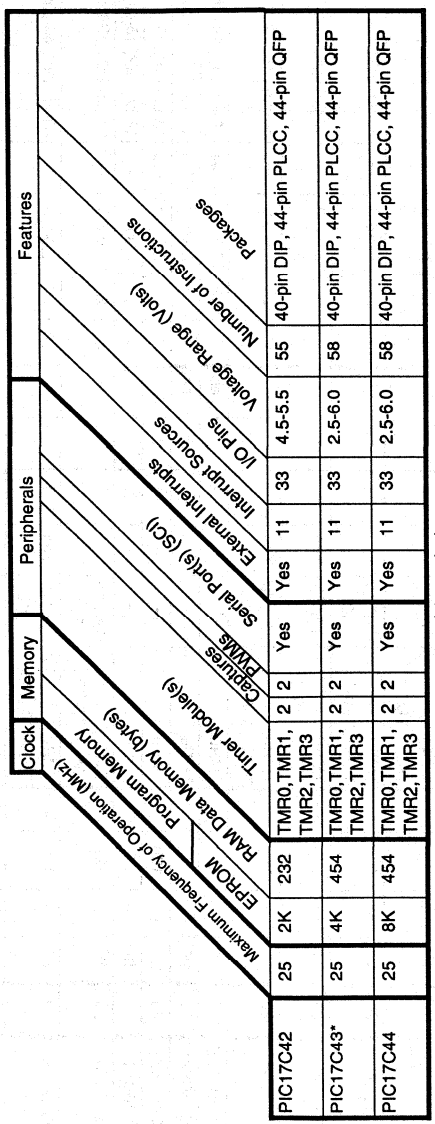
SECTION 2
PIC16/17 MICROCONTROLLER PRODUCT SPECIFICATIONS

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

PIC16/17 Family of 8-Bit Microcontrollers Cross-Reference Guide

PIC17CXX FAMILY OF DEVICES



	Clock	Memory			Peripherals					Features			
	Maximum Frequency of Operation (MHz)	Program Memory	EPROM	RAM Data Memory (bytes)	Timer Module(s)	Capture PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0,TMR1, TMR2,TMR3	2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller modes.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

	Clock	Memory				Peripherals							Features				
	Maximum Frequency of Operation (MHz)	Program Memory EPROM	Program Memory EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

PIC16C5X AND ENHANCED PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72		TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

PIN COMPATIBILITY

Devices that have the same package type; and VDD, VSS, and $\overline{MCLR}$ pin locations, are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

Pin Compatibile Devices

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

PIC16C5X

EPROM/ROM-Based 8-Bit CMOS Microcontroller Series

Devices Included in this Data Sheet

- PIC16C54
- PIC16CR54
- PIC16C55
- PIC16C56
- PIC16C57

High-Performance RISC-like CPU

- Only 33 single word instructions to learn
- All instructions are single cycle (200 ns) except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle

Device	Pins	I/O	EPROM/ ROM	RAM
PIC16C54	18	12	512	25
PIC16CR54	18	12	512	25
PIC16C56	18	12	1K	25
PIC16C55	28	20	512	24
PIC16C57	28	20	2K	72

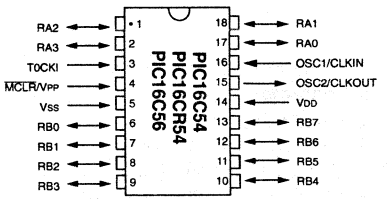
- 12-bit wide instructions
- 8-bit wide data path
- Seven or eight special function hardware registers
- Two-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

Peripheral Features

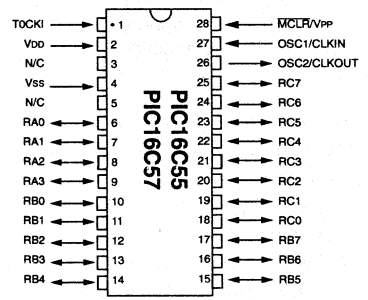
- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- Power-On Reset (POR)
- Device Reset Timer
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- EPROM/ROM selectable oscillator options:
  - RC: Low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High-speed crystal/resonator
  - LP: Power saving, low frequency crystal

PACKAGE TYPES

PDIP, SOIC, Windowed CERDIP



PDIP, SOIC, Windowed CERDIP



CMOS Technology

- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide-operating voltage range:
  - EPROM Commercial/Industrial 2.5V to 6.25V
  - ROM Commercial/Industrial 2.0V to 6.25V
  - EPROM/ROM Automotive 2.5V to 6.0V
- Low-power consumption
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 3V, 32 kHz
  - < 3 µA typical standby current (with WDT disabled) @ 3V, 0°C to 70°C

© 1995 Microchip Technology Inc.

Package Types

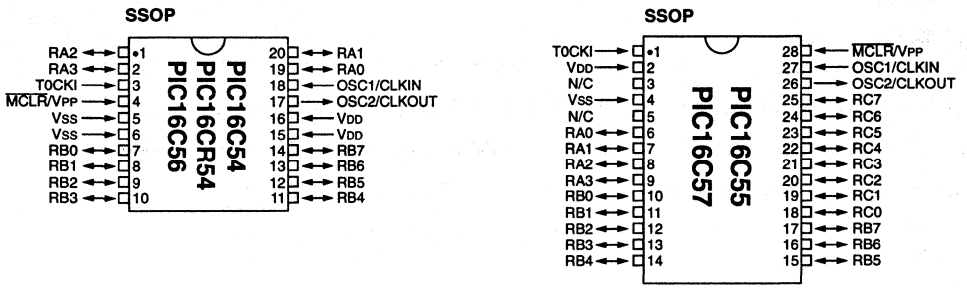


Table of Contents

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error from the previous version of this data sheet (PIC16C5X Data Sheet, Literature Number DS30015K), please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

© 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The PIC16C5X from Microchip Technology is a family of low-cost, high-performance, 8-bit, fully static, EPROM-based CMOS microcontrollers. This family is pin and software compatible with the Enhanced PIC16C5X family of devices. It employs a RISC-like architecture with only 33 single word/single cycle instructions. All instructions are single cycle (200 ns) except for program branches which take two cycles. The PIC16C5X delivers performance an order of magnitude higher than its competitors in the same price category. The 12-bit wide instructions are highly symmetrical resulting in 2:1 code compression over other 8-bit microcontrollers in its class. The easy to use and easy to remember instruction set reduces development time significantly.

The PIC16C5X products are equipped with special features that reduce system cost and power requirements. The Power-On Reset (POR) and Device Reset Timer eliminate the need for external reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost-saving RC oscillator. Power saving SLEEP mode, watchdog timer and code protection features improve system cost, power and reliability.

The UV-erasable CERDIP-packaged versions are ideal for code development, while the cost effective One Time Programmable (OTP) versions are suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in OTP microcontrollers while benefiting from the OTP's flexibility.

The PIC16C5X products are supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a 'C' compiler, fuzzy logic support tools, a low-cost development programmer, and a full featured programmer. All the tools are supported by IBM® PC and compatible machines.

1.1 Applications

The PIC16C5X series fits perfectly in applications ranging from high-speed automotive and appliance motor control to low-power remote transmitters/receivers, pointing devices and telecom processors. The EPROM technology makes customizing application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make this microcontroller series perfect for applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C5X series very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, replacement of "glue" logic in larger systems, co-processor applications).

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC16C5X FAMILY OF DEVICES

	Clock	Memory			Peripherals		Features		
	Maximum Frequency of Operation (MHz)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

2.0 PIC16C5X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in this section. When placing orders, please use the "PIC16C5X Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs

The UV erasable version can be erased and reprogrammed to any of the oscillator modes. Microchip's PICSTART™ and PRO MATE™ programmers both support programming of the PIC16C5X.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages permit the user to program them once. In addition to the program memory, the configuration fuses must be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and fuse options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround-Production (SQTP) Devices

Microchip offers the unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, giving the customer a low cost option for high volume, mature products.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C5X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C5X uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12-bits wide making it possible to have all single word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The PIC16C58 and PIC16C57 address 2K x 12 program memory, the PIC16C55 and PIC16C56 address 1K x 12, and the PIC16C54 addresses 512 x 12 of program memory. All program memory is internal.

The PIC16C5X can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory. The PIC16C5X has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16C5X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C5X device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the W register (working register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

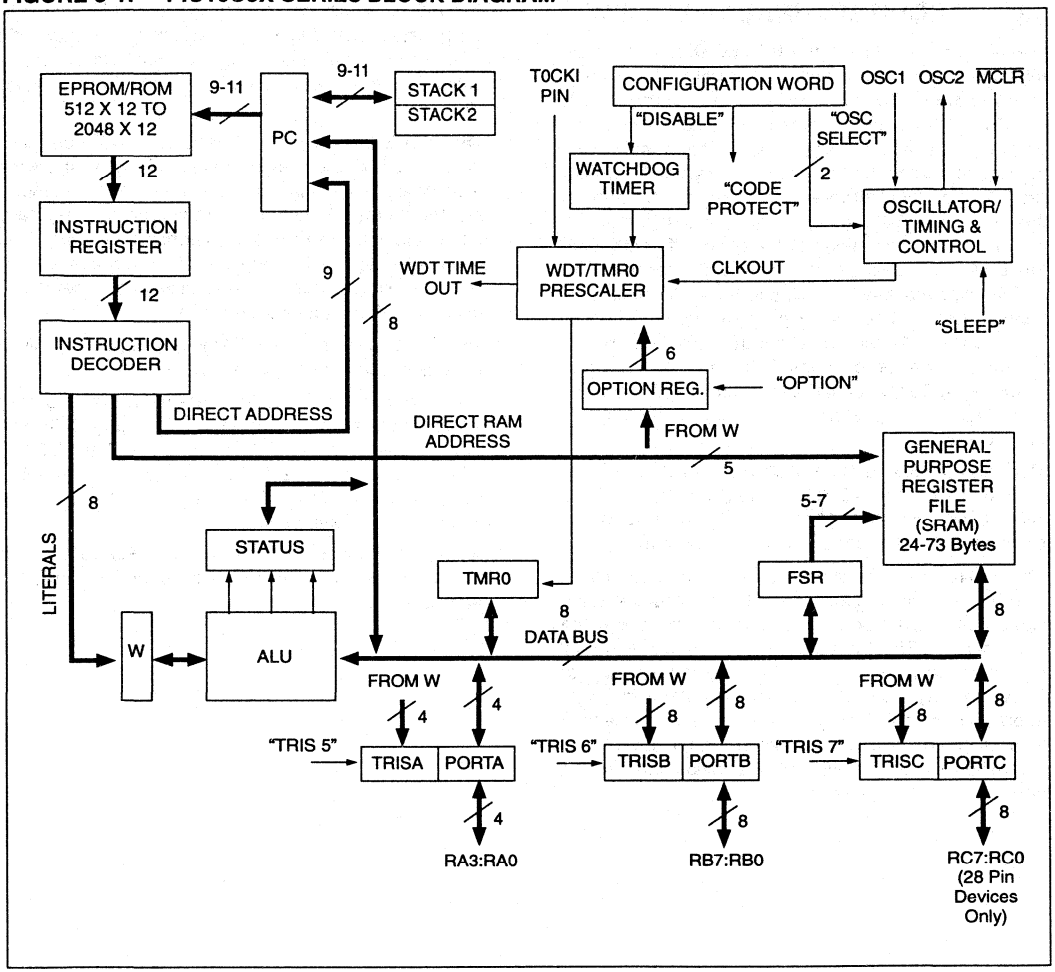
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBWF` and `ADDWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, with the corresponding device pins described in Table 3-1 and Table 3-2.

© 1995 Microchip Technology Inc.

FIGURE 3-1: PIC16C5X SERIES BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C54/CR54/C56 PINOUT DESCRIPTION

Name	DIP, SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
RA0	17	19	I/O	TTL	Bi-directional I/O port
RA1	18	20	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RB0	6	7	I/O	TTL	Bi-directional I/O port
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	
RB5	11	12	I/O	TTL	
RB6	12	13	I/O	TTL	
RB7	13	14	I/O	TTL	
T0CKI	3	3	I	ST	Clock input to TMR0 timer. Must be tied to Vss or VDD, if not in use, to reduce current consumption.
$\overline{MCLR}$/VPP	4	4	I	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device. Voltage on $\overline{MCLR}$/VPP must not exceed VDD to avoid unintended entering of test modes.
OSC1/CLKIN	16	18	I	ST	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	17	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
VDD	14	15,16	P	—	Positive supply for logic and I/O pins.
Vss	5	5,6	P	—	Ground reference for logic and I/O pins.

Legend: I=input, O=output, I/O=input/output, P=power, — =Not Used, TTL=TTL input, ST=Schmitt Trigger input

© 1995 Microchip Technology Inc.

TABLE 3-2: PIC16C55\C57 PINOUT DESCRIPTION

Name	DIP, SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
RA0	6	5	I/O	TTL	Bi-directional I/O port
RA1	7	6	I/O	TTL	
RA2	8	7	I/O	TTL	
RA3	9	8	I/O	TTL	
RB0	10	9	I/O	TTL	Bi-directional I/O port
RB1	11	10	I/O	TTL	
RB2	12	11	I/O	TTL	
RB3	13	12	I/O	TTL	
RB4	14	13	I/O	TTL	
RB5	15	15	I/O	TTL	
RB6	16	16	I/O	TTL	
RB7	17	17	I/O	TTL	
RC0	18	18	I/O	TTL	Bi-directional I/O port
RC1	19	19	I/O	TTL	
RC2	20	20	I/O	TTL	
RC3	21	21	I/O	TTL	
RC4	22	22	I/O	TTL	
RC5	23	23	I/O	TTL	
RC6	24	24	I/O	TTL	
RC7	25	25	I/O	TTL	
T0CKI	1	2	I	ST	Clock input to TMR0 register. Must be tied to Vss or VDD, if not in use, to reduce current consumption.
$\overline{MCLR}$/VPP	28	28	I	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device. Voltage on $\overline{MCLR}$/VPP must not exceed VDD to avoid unintended entering of test modes.
OSC1/CLKIN	27	27	I	ST	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	26	26	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
VDD	2	3,4	P	—	Positive supply for logic and I/O pins.
Vss	4	1,14	P	—	Ground reference for logic and I/O pins.
N/C	3,5	—	—	—	Unused, do not connect

Legend: I=input, O=output, I/O=input/output, P=power, — =Not Used, TTL=TTL input, ST=Schmitt Trigger input

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, and the instruction is fetched from program memory and latched into the instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

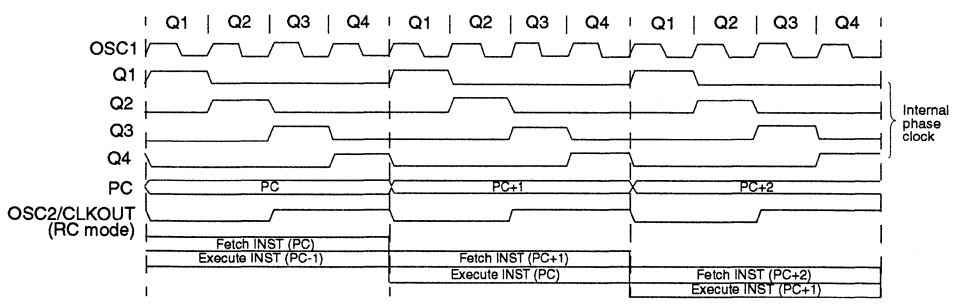
3.2 Instruction Flow/Pipelining

An Instruction Cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

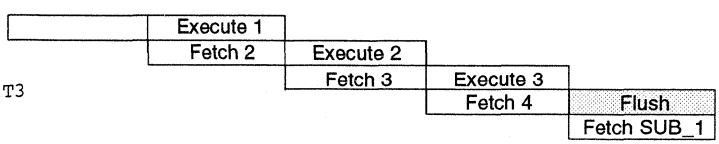
In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

```
1. MOVLW  55h
2. MOVWF  PORTB
3. CALL   SUB_1
4. BSF    PORTA, BIT3
```



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

Up to 512 words of 12-bit wide on-chip program memory (EPROM/ROM) can be directly addressed. Larger program memories can be addressed by selecting one of up to four available pages of 512 words each (Figure 4-1). Sequencing of instructions is controlled via the Program Counter (PC) which automatically increments to execute in-line programs. Program control operations supporting direct, indirect, and relative addressing modes, can be performed by bit test and skip instructions, call instructions, jump instructions or by loading computed addresses into the PC. In addition, an on-chip two-level stack is employed to provide easy to use subroutine nesting.

4.2 Data Memory Organization

The 8-bit data bus connects two basic functional elements together: the Register File composed of up to 80 addressable 8-bit registers including the I/O Ports, and an 8-bit wide Arithmetic Logic Unit. 32 bytes of RAM are directly addressable while a "banking" scheme, with banks of 16 bytes each, is employed to address larger data memories (Figure 4-2). Data can be addressed directly, or indirectly using the file select register. Immediate data addressing is supported by special "literal" instructions which load data from program memory into the W register.

The register file is divided into two functional groups: Special Function registers and General Purpose registers. The special function registers include the Timer0 (TMR0) register, the Program Counter (PC), the Status Register, the I/O registers (ports), and the File Select Register (FSR). The general purpose registers are used for data and control information under command of the instructions.

In addition, special purpose registers are used to control the I/O port configuration and prescaler options.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly or indirectly through the file select register FSR (Section 4.5).

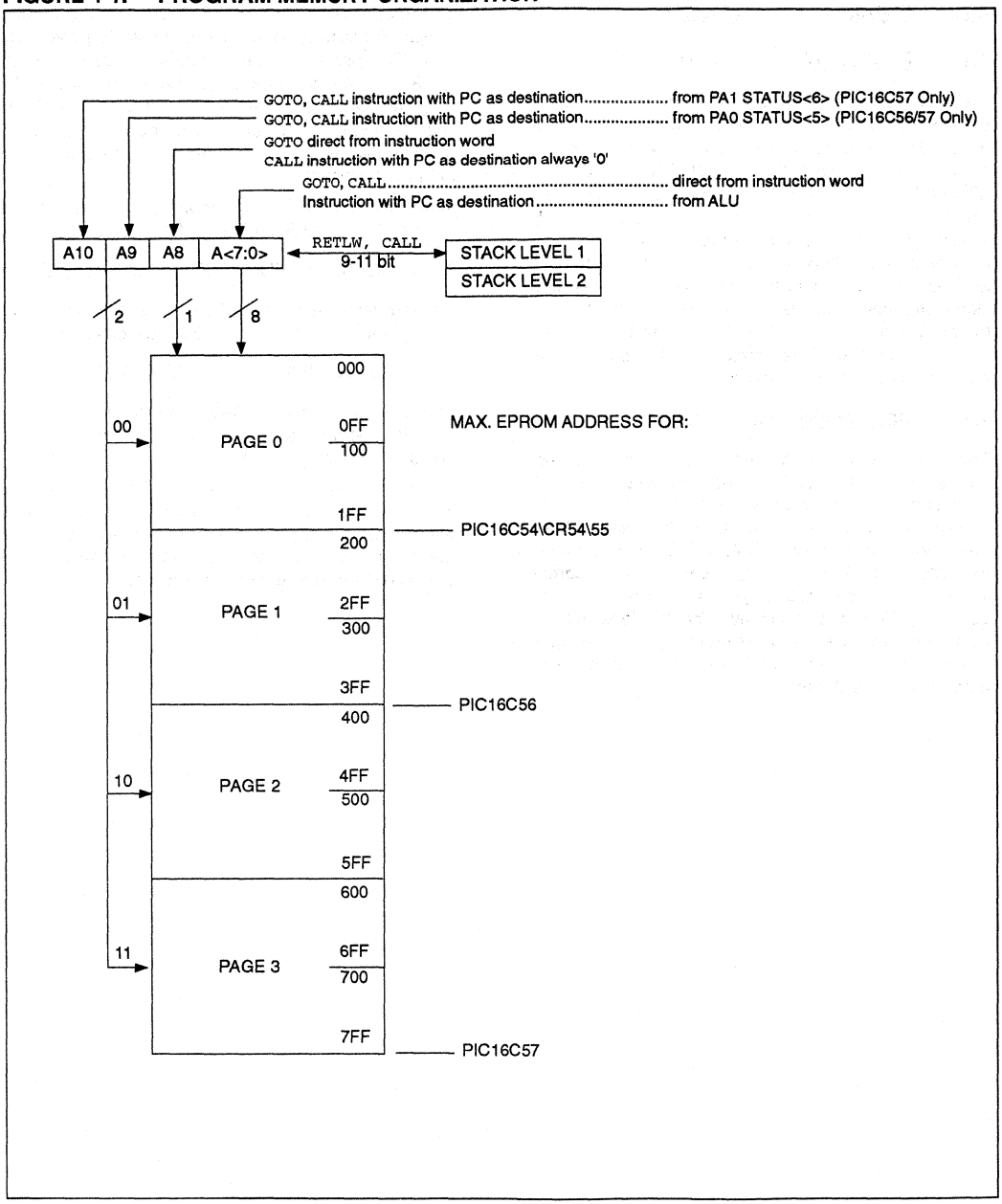
4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device (Table 4-1).

The special registers can be classified into two sets. The special registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

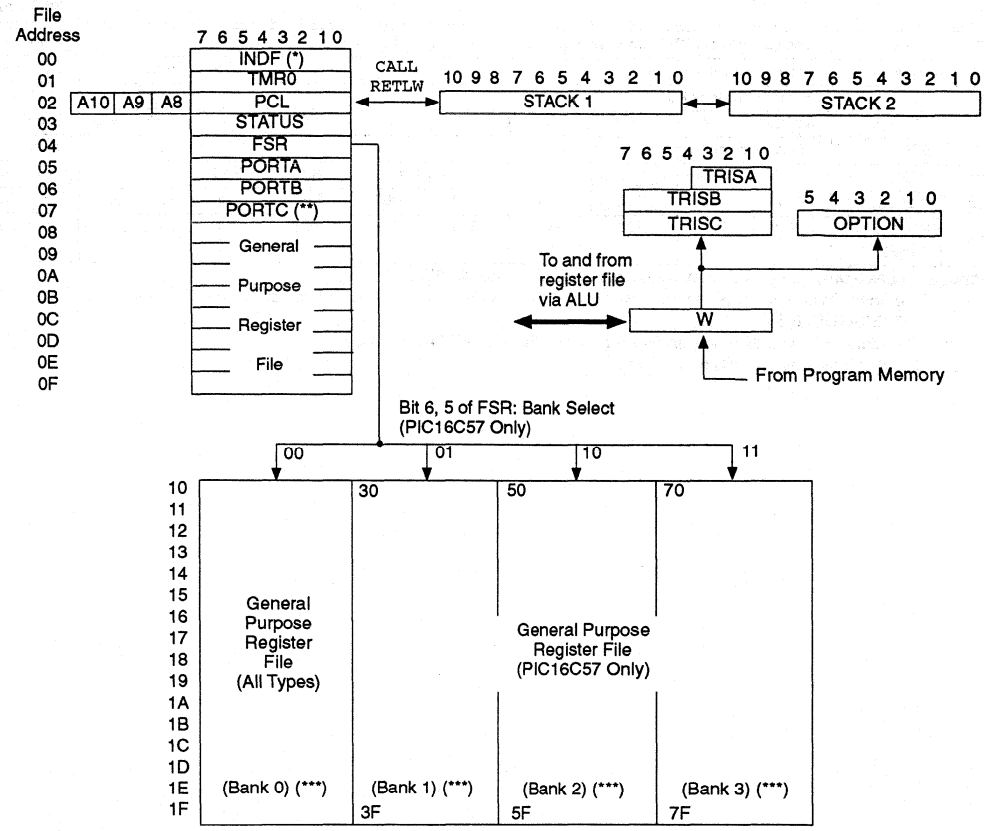
© 1995 Microchip Technology Inc.

FIGURE 4-1: PROGRAM MEMORY ORGANIZATION



© 1995 Microchip Technology Inc.

FIGURE 4-2: DATA MEMORY MAP



(*) Not a physically implemented register. See Section 4.5 for details.

(**) File address 7h is a general purpose register on the PIC16C54\CR54\C56 only.

(***) Bank 0 is available on all microcontrollers while Bank 1, Bank 2, and Bank 3 are only available on the PIC16C57. (Section 4.5).

© 1995 Microchip Technology Inc.

TABLE 4-1: PIC16C5X REGISTER FILE SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on $\overline{MCLR}$ and WDT Reset
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of PC								1111 1111	1111 1111
03h	STATUS	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	PORTC (Note 2)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = Unimplemented, Read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. The upper bits can be set or cleared by writing to PA1:PA0 (STATUS<6:5>).

2: File address 7h is a general purpose register on the PIC16C54\CR54\C56.

3: Shading indicates unimplemented bits.

© 1995 Microchip Technology Inc.

4.3 STATUS Register

This register contains the arithmetic status of the ALU, the RESET status, and the page preselect bits for program memories larger than 512 words .

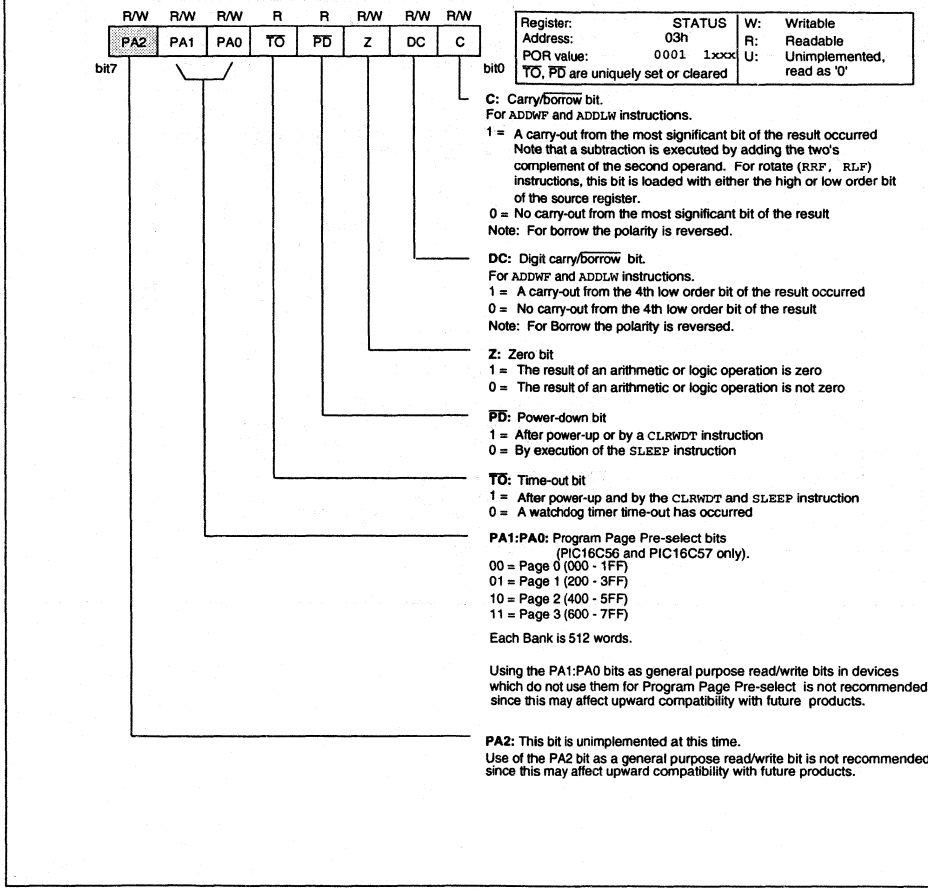
As with any other register, the STATUS register can be the destination for any instruction. However, the STATUS bits are set after the following write. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear all bits except for $\overline{TO}$ and $\overline{PD}$ and then set the Z bit and leave status register as `000u u100` (where `u` = unchanged).

Therefore, only `BCF`, `BSF` and `MOVWF` instructions should be used to alter the STATUS register because these instructions do not affect any STATUS bit.

For other instructions, which do affect STATUS bits, see Table 8-2, Instruction Set Summary.

FIGURE 4-3: STATUS REGISTER

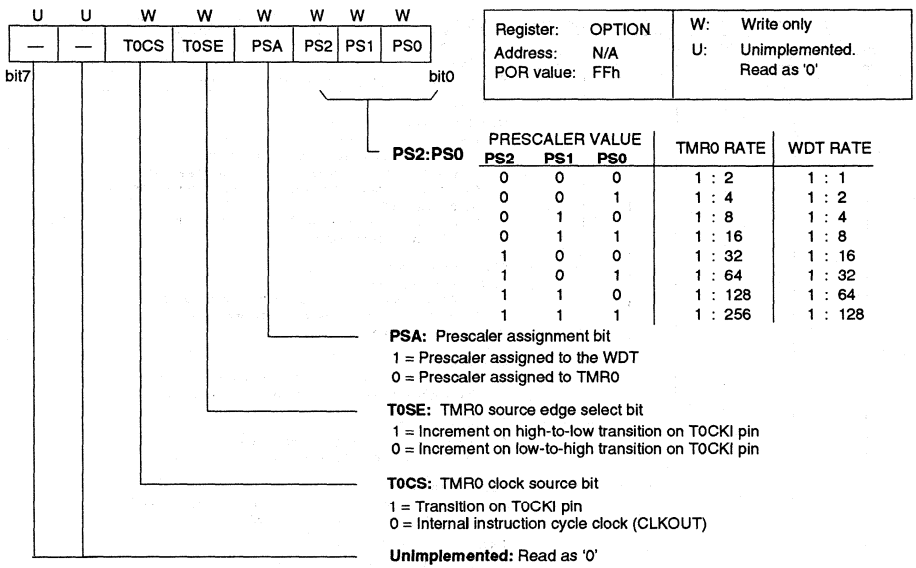


4.4 OPTION Register

The OPTION register is a 6-bit wide, write-only register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt and TMR0.

By executing the `OPTION` instruction, the contents of the W register will be transferred to the OPTION register. A RESET sets the OPTION register to all '1's.

FIGURE 4-4: OPTION REGISTER



© 1995 Microchip Technology Inc.

4.5 Indirect Data Addressing, INDF and FSR Registers

The INDF register is not a physical register and is used in conjunction with the FSR register to perform indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself (i.e., FSR = 0) indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
        movlw   0x10    ; Initialize pointer
        movwf   FSR     ; to RAM
Next    clrf    INDF    ; Clear loc
        incf    FSR     ; increment pntr
        btfsc   FSR,4   ; All done?
        goto    Next    ; No, clear next
        .               ; Location
        .
        .
```

4.5.1 FILE SELECT REGISTER (FSR)

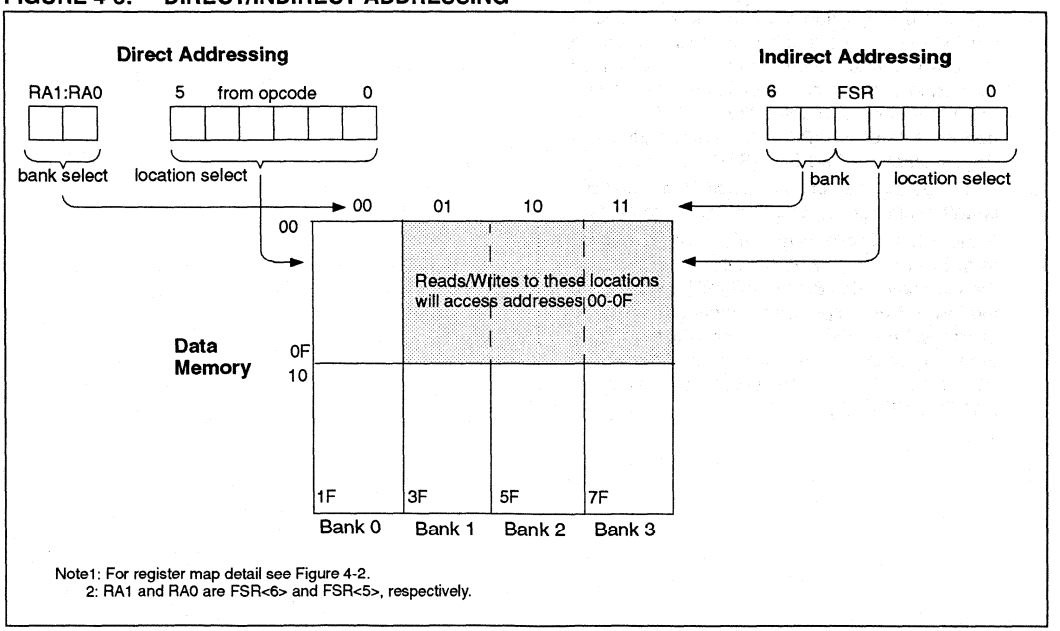
The FSR is either a 5-bit (PIC16C54\CR54\C55\C56) or 7-bit (PIC16CR57) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area. The FSR<4:0> bits are the pointer for data memory addresses 00h to 1Fh. FSR<4:0> toggles between the 16 lower (ooh-0Fh) and 16 upper (10h-1Fh) register files. When clear, FSR<4> points to the lower 16 register files and, when set, points to the upper 16 register files. FSR<3:0> provide the value to address the specific register file within each 16 file area.

When not performing indirect addressing, the FSR can be used as a 5-bit (FSR<4:0>) wide general purpose register. However, this is not recommended to help ensure future upward code compatibility.

PIC16C54\CR54\C55\C56: Does not use banking. FSR<7:5> are unimplemented and read as '1's.

PIC16CR57: FSR<6:5> are the bank select bits and are used to select the bank to be addressed (00 = bank 0, 01 = bank 1, 10 = bank 2, 11 = bank 3). The lower 16 register files for banks 1, 2 and 3 are mapped to bank 0, and are not accessible. In other words, the FSR<6:5> bits are ignored if FSR<4> is clear (= 0). FSR<7> is unimplemented and is always read as '1'.

FIGURE 4-5: DIRECT/INDIRECT ADDRESSING



Note1: For register map detail see Figure 4-2.
2: RA1 and RA0 are FSR<6> and FSR<5>, respectively.

4.6 Program Counter

The program counter generates the addresses for up to 2048 x 12 on-chip EPROM/ROM cells containing the program instruction words (Figure 4-1).

Depending on the device type, the program counter and its associated two-level hardware stack is 9 or 11-bits wide.

TABLE 4-2: PROGRAM COUNTER STACK WIDTH

Part #	PC width	Stack width
PIC16C54\CR54\C55	9-bit	9-bit
PIC16C56	10-bit	10-bit
PIC16C57	11-bit	11-bit

The program counter is set to all '1's upon RESET. During program execution it is auto-incremented with each instruction unless the result of that instruction changes the PC itself.

a) The GOTO instruction allows the direct loading of the lower nine program counter bits (PC<8:0>). In the case where the program memory is greater than 512 bytes, the upper two bits of PC (PC<10:9>) are loaded with page select bits PA1:PA0 (STATUS<6:5>). Thus, GOTO allows jumps to any location on any page.

b) The CALL instruction loads the lower 8-bits of the PC directly, while the ninth bit is cleared to '0'. The PC value, incremented by one, will be pushed into the stack. In cases where the program memory is greater than 512 bytes, the upper 2-bits of PC (PC<10:9>) are loaded with Page Select bits PA1:PA0 (STATUS<6:5>).

c) The RETLW instruction loads the program counter with the Top Of Stack (TOS) contents.

d) If the PC is the destination in any instruction (e.g. MOVWF PC, ADDWF PC, or BSF PC,5) then the computed 8-bit result will be loaded into the lower 8-bits of program counter. The ninth bit of PC will be cleared. In case where the program memory is greater than 512 bytes, PC<10:9> will be loaded with Page Select bits PA1:PA0 (STATUS<6:5>).

It should be noted that because bit8 (ninth bit) of the PC is cleared in the CALL instruction or any instruction which writes to the PC (e.g. MOVWF PC), all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

Incrementing the program counter when it is pointing to the last address of a selected memory page is also possible and will cause the program to continue in the next higher page. However, the page preselect bits in the STATUS register will not be changed, and the next GOTO, CALL, ADDWF PC, or MOVWF PC instruction will return to the previous page, unless the page preselect bits have been updated under program control. For example, an NOP at location 1FFh (page 0) increments the PC to 200h (page 1). A GOTO xxx at 200h will return the program to address xxxh on page 0 (assuming that PA1:PA0 is clear).

Upon a RESET condition, page 0 is pre-selected while the program counter addresses the last location in the last page. Thus, a GOTO instruction at this location will automatically cause the program to continue in page 0.

© 1995 Microchip Technology Inc.

4.7 W (Working) Register

The W register holds the second operand in two operand instructions and/or supports the internal data transfer.

TABLE 4-3: EVENTS AFFECTING $\overline{TO}/\overline{PD}$ STATUS BITS

Event	$\overline{TO}$	$\overline{PD}$	Remarks
Power-up	1	1	
WDT Timeout	0	x	No effect on $\overline{PD}$
SLEEP instruction	1	0	
CLRWDT instruction	1	1	

A WDT timeout will occur regardless of the status of the $\overline{TO}$ bit. A SLEEP instruction will be executed, regardless of the status of the $\overline{PD}$ bit. Table 4-4 reflects the status of $\overline{TO}$ and $\overline{PD}$ after the corresponding event.

4.7.1 TIME OUT AND POWER DOWN STATUS BITS ($\overline{TO}$, $\overline{PD}$)

The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be tested to determine if a RESET condition has been caused by a Watchdog Timer time-out, a power-up condition, or a wake-up from SLEEP by the Watchdog Timer or $\overline{MCLR}$ pin.

These STATUS bits are only affected by events listed in Table 4-3.

TABLE 4-4: $\overline{TO}/\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	= Low pulse on $\overline{MCLR}$ input

The $\overline{TO}$ and $\overline{PD}$ bits maintain their status (u) until an event of Table 4-3 occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

5.0 I/O PORTS

As with any other register, the I/O registers can be written and read under program control. However, read instructions (e.g., `MOVF PORTB,W`) always read the I/O pins independent of the pin's input/output modes. On RESET, all I/O ports are defined as input (outputs are at hi-impedance) as the I/O control registers (TRISA, TRISB, TRISC) are all set.

5.1 PORTA

PORTA is a 4-bit I/O register. Only the low order 4 bits are used (RA3:RA0). Bits 7-4 are unimplemented and read as '0's.

5.2 PORTB

PORTB is an 8-bit I/O register (PORTB<7:0>).

5.3 PORTC

PIC16C55/C57: 8-bit I/O register.

PIC16C54/CR54/C56: General purpose register.

5.4 TRIS Registers

The output driver control registers are loaded with the contents of the W register by executing the `TRIS f` instruction. A '1' from a TRIS register bit puts the corresponding output driver in a hi-impedance mode. A '0' puts the contents of the output data latch on the selected pins.

Note: A read of the ports reads the pins, not the output data latches, i.e., if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

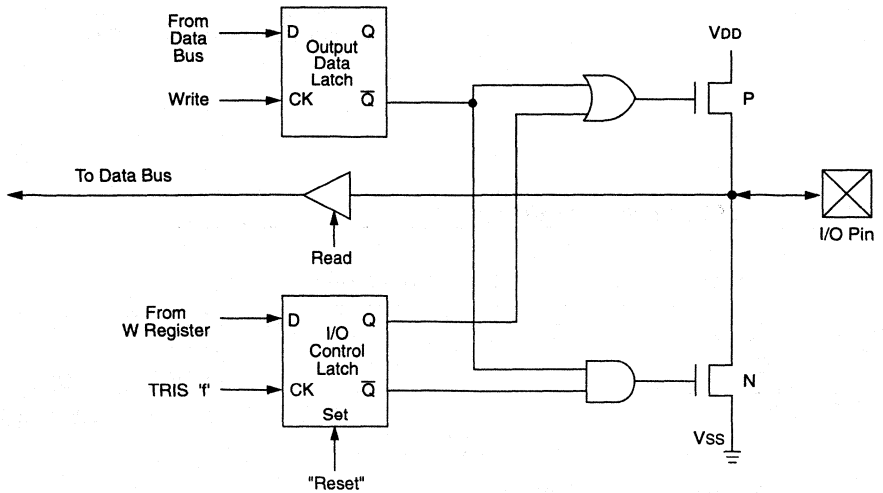
The TRIS registers are "write-only" and are set to all '1's (output drivers disabled) upon RESET.

5.5 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-1. All ports may be used for both input and output operations. For input operations these ports are non-latching. Any input must be present until read by an input instruction (e.g. `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit (in TRISA, TRISB, TRISC) must be cleared (= 0). For use as an input, the corresponding TRIS bit must be '1'. Any I/O pin can be programmed individually as input or output.

2

FIGURE 5-1: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



Note: I/O pins have protection diodes to VDD and VSS.

5.6 I/O Programming Considerations

5.6.1 BI-DIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-output the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and the PORTB value is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (say bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O PORT.

A pin actively outputting a HI or LO should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

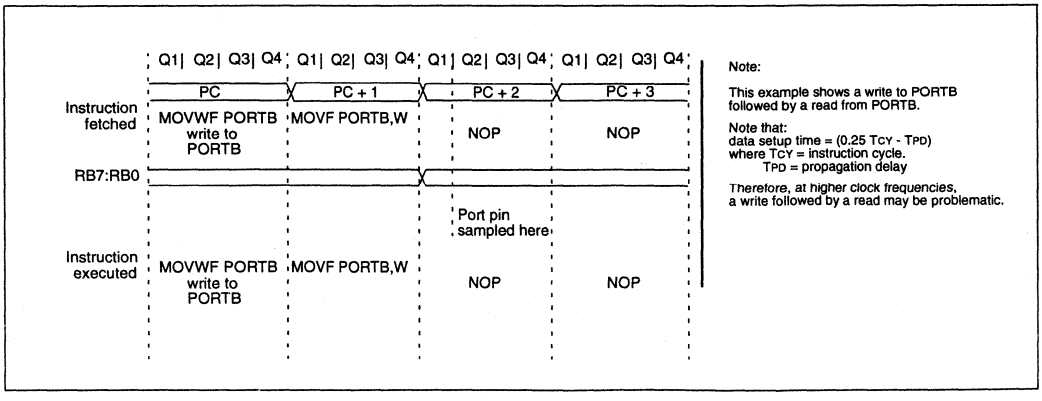
EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
; Initial PORT Settings
;   PORTB<7:4> Inputs
;   PORTB<3:0> Outputs
; PORTB<7:6> have external pull-ups and are
; not connected to other circuitry
;
;                        PORT latch  PORT pins
;                        ----------  ----------

    BCF   PORTB, 7     ; 01pp pppp   11pp pppp
    BCF   PORTB, 6     ; 10pp pppp   11pp pppp
    BSF   STATUS,RP0   ;
    MOVLW 03Fh         ;
    TRIS  PORTB        ; 10pp pppp   10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-2: SUCCESSIVE I/O OPERATION



© 1995 Microchip Technology Inc.

6.0 TIMER0 (TMR0) MODULE

The TMR0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Edge select for external clock

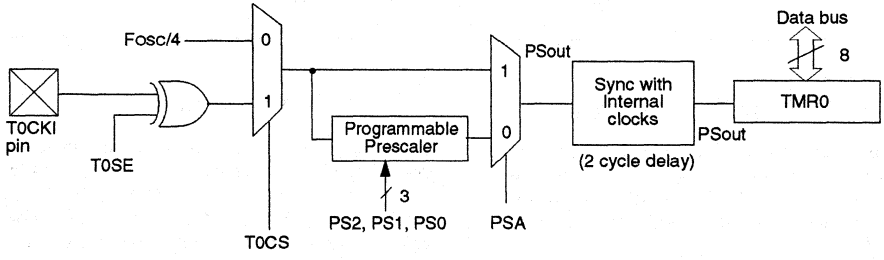
Figure 6-1 is a simplified block diagram of the TMR0 module, while Figure 6-2 shows the electrical structure of the TMR0 input.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, increment is inhibited for the following two cycles (Figure 6-3 and Figure 6-4). The user can work around this by writing an adjusted value to the TMR0 module.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin T0CKI. The incrementing edge is determined by the T0 source edge (T0SE) select bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.1.

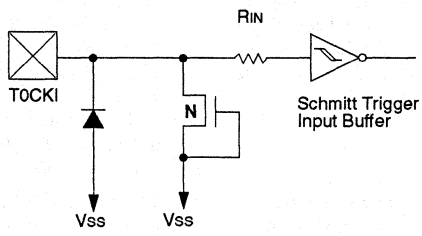
The prescaler is shared between the TMR0 module and the watchdog timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Setting the PSA bit will assign the prescaler to the WDT and cause the prescale for TMR0 to be 1:1. Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is neither readable nor writable. When the prescaler is assigned to the TMR0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 6.2 details the operation of the prescaler.

FIGURE 6-1: TMR0 BLOCK DIAGRAM



Note 1: Bits, T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.
2: The prescaler is shared with Watchdog Timer (Figure 6-6).

FIGURE 6-2: ELECTRICAL STRUCTURE OF T0CKI PIN



© 1995 Microchip Technology Inc.

FIGURE 6-3: TMR0 TIMING: INTERNAL CLOCK/NO PRESCALE

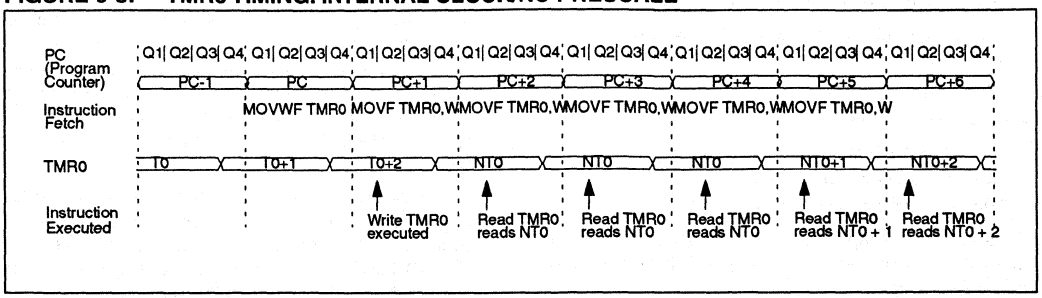
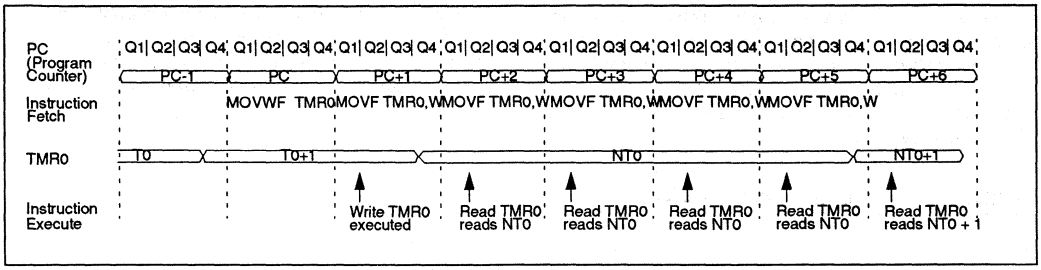


FIGURE 6-4: TMR0 TIMING: INTERNAL CLOCK/PRESCALE 1:2



6.1 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements to be able to synchronize with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of TMR0 after synchronization.

6.1.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. Synchronizing T0CKI with the internal phase clocks requires sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (plus a small RC delay) and low for at least 2 TOSC (plus a small RC delay). Refer to the appropriate electrical specification table.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler to ensure that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 TOSC (plus a small RC delay) divided by the prescaler value. The only limitation on T0CKI high and low time is that they are greater than the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the appropriate electrical specification section.

6.1.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

6.2 Prescaler

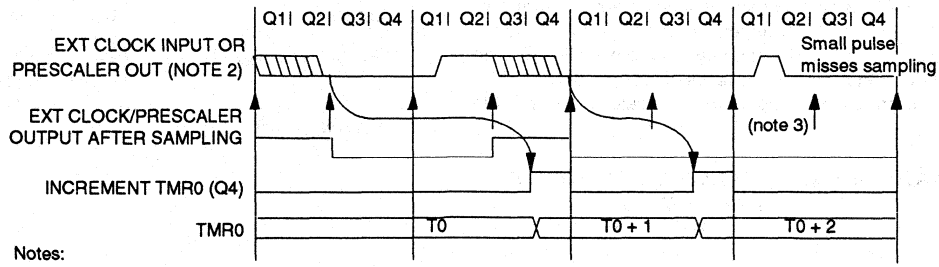
An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer, (Section 6.1.2). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g., `CLRF 1, MOVWF 1, BSF 1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is neither readable nor writable. On a RESET, the prescaler contains all '0's.

© 1995 Microchip Technology Inc.

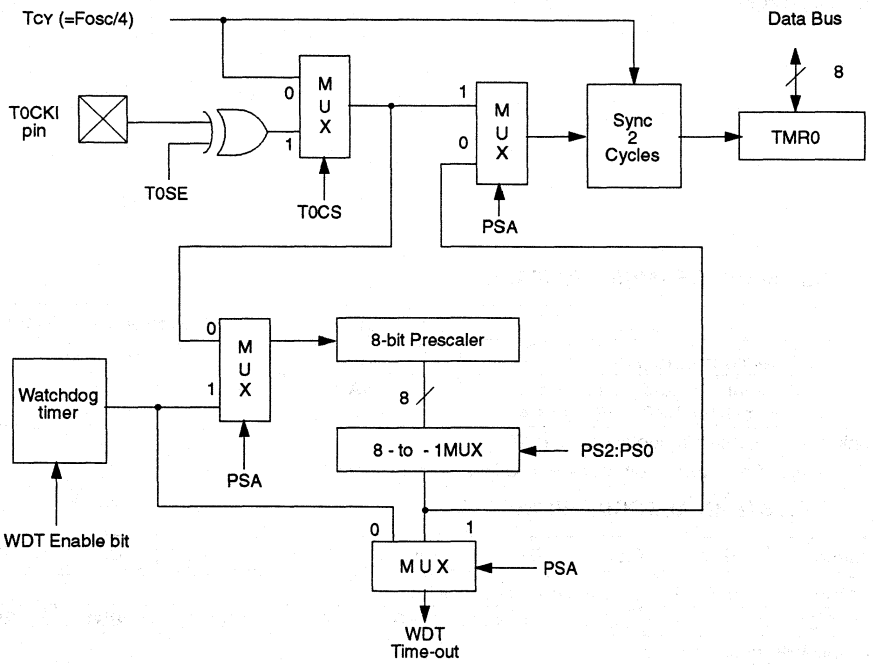
FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to TMR0 increment is 3 Tosc to 7 Tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 Tosc max.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

FIGURE 6-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0SE, T0CS, PSA, PS2:PS0 are bits in the OPTION register.

6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler from TMR0 to WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TMR0→WDT)

```
CLRF    TMR0            ;Clear TMR0
CLRWDT                  ;Clears WDT and
                        ;prescaler
MOVLW   'xxxx1xxx'b     ;Select new prescale
OPTION                  ;value
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled.

Note that a `CLRWDT` instruction should be executed before switching the prescaler.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                  ;Clear WDT and
                        ;prescaler
MOVLW   'xxxx0xxx'b     ;Select TMR0, new
                        ;prescale value and
                        ;clock source
OPTION
```

TABLE 6-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. (Figure 4-4)	N/A	--11 1111

Legend: x = unknown, - = unimplemented, read as '0'.

Note: For reset values of registers in other reset situations, refer to Table 4-1.

TABLE 6-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01	TMR0	Timer0 - 8-bit real-time clock counter							
N/A	OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: — = Unimplemented, read as '0'.

Note: Shaded cells are not used by TMR0 module.

© 1995 Microchip Technology Inc.

7.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC16C5X family of microcontrollers has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- Oscillator selection
- Reset
- Power-On Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations

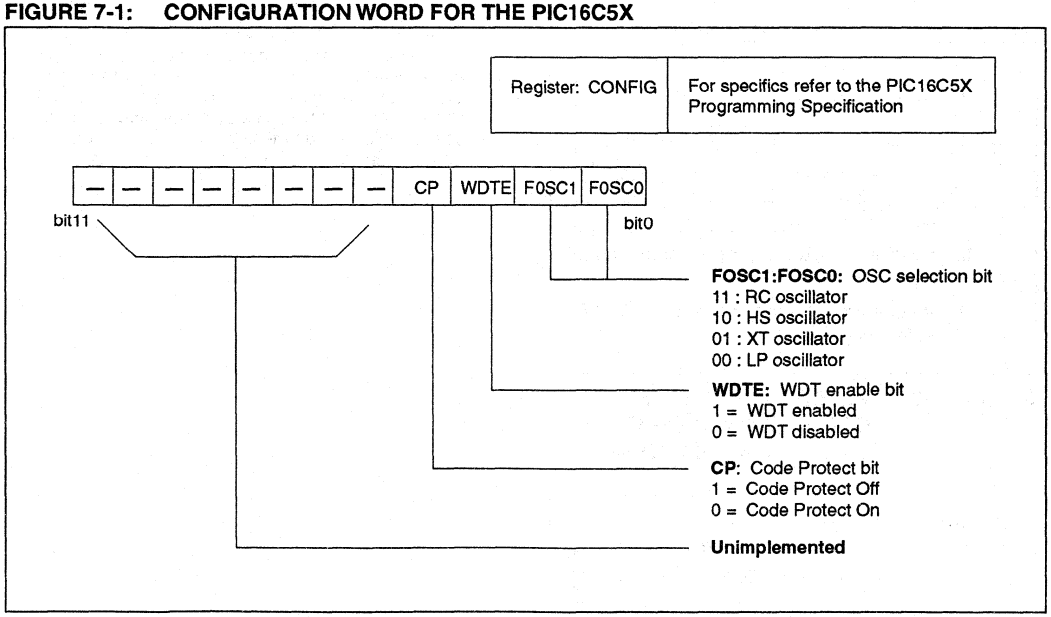
The PIC16C5X has a watchdog timer which can be shut off only through configuration bit WDTE. It runs off of its own RC oscillator for added reliability. There is an 18 ms delay provided by the Device Reset Timer (DRT), intended to keep the chip in reset until the crystal oscillator is stable. With this timer on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset or through a watchdog timer time-out. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

7.1 Configuration Bits

The configuration word consists of 4 or 12 bits, depending on the device configuration. Configuration bits can be programmed to select various device configurations. Two are for the selection of the oscillator type, one is the Watchdog Timer enable bit, and one is the code protection bit.

FIGURE 7-1: CONFIGURATION WORD FOR THE PIC16C5X



© 1995 Microchip Technology Inc.

7.2 Oscillator Configurations

7.2.1 OSCILLATOR TYPES

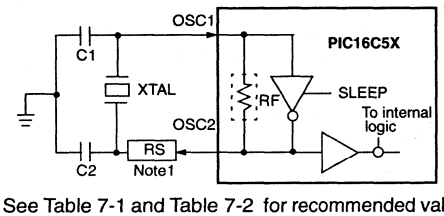
The PIC16C5X can be operated in four different oscillator modes. The user can choose one of following modes from the factory:

- LP: Low Power Crystal
- XT: Crystal/Resonator
- HS: High Speed Crystal/Resonator
- RC: Resistor/Capacitor

7.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 7-2). The PIC16C5X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 7-3).

FIGURE 7-2: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



See Table 7-1 and Table 7-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 7-3: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

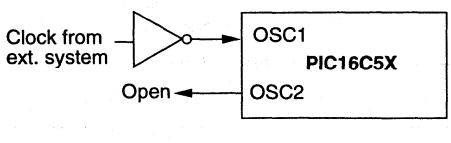


TABLE 7-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Osc Type	Resonator Freq	Cap. Range C1	Cap. Range C2
XT	455 kHz	68-100 pF	68-100 pF
	2.0 MHz	15-68 pF	15-68 pF
	4.0 MHz	10-100 pF	10-100 pF
HS	8.0 MHz	10-68 pF	10-68 pF
	16.0 MHz	10-10 pF	10-10 pF

These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

TABLE 7-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Resonator Freq	Cap.Range C1	Cap. Range C2
LP	32 kHz†	33-68 pF	33-68 pF
	200 kHz	15-33 pF	15-33 pF
XT	100 kHz	68-100 pF	68-100 pF
	2 MHz	10-22 pF	10-22 pF
	4 MHz	10-22 pF	10-22 pF
HS	8 MHz	22-47 pF	22-47 pF
	20 MHz	22-47 pF	22-47 pF

† For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.

These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

© 1995 Microchip Technology Inc.

7.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 7-4 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 7-4: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

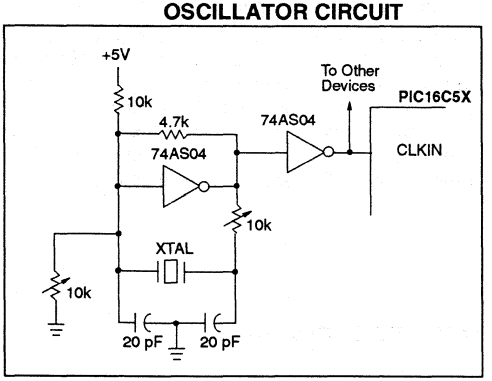
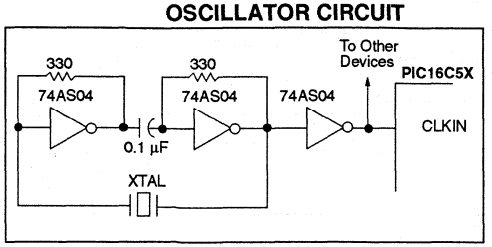


Figure 7-5 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 7-5: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



7.2.4 RC OSCILLATOR

For timing insensitive applications the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 7-6 shows how the R/C combination is connected to the PIC16C5X. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g., 1 MΩ) the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping Rext between 3 kΩ and 100 kΩ.

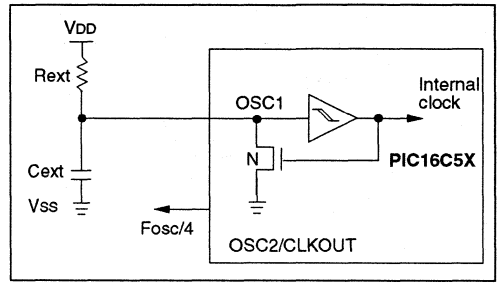
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 9.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 9.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic.

FIGURE 7-6: RC OSCILLATOR MODE



7.3 Reset

The PIC16C5X differentiates between various kinds of resets:

- Power-On Reset (POR)
- $\overline{MCLR}$ reset during normal operation
- $\overline{MCLR}$ reset during SLEEP
- WDT time-out reset

Some registers are not affected in any reset condition. Their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-On Reset (POR), $\overline{MCLR}$ or a WDT reset. Note that the PIC16C5X does not differentiate between a WDT reset during SLEEP, or during normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared depending on the different reset (Table 7-3). These bits may be used to determine the nature of the reset. See Table 7-5 for a full description of reset states of all registers.

Figure 7-7 shows a simplified block diagram of the on-chip reset circuit.

7.4 Power-On Reset (POR) and Device Reset Timer (DRT)

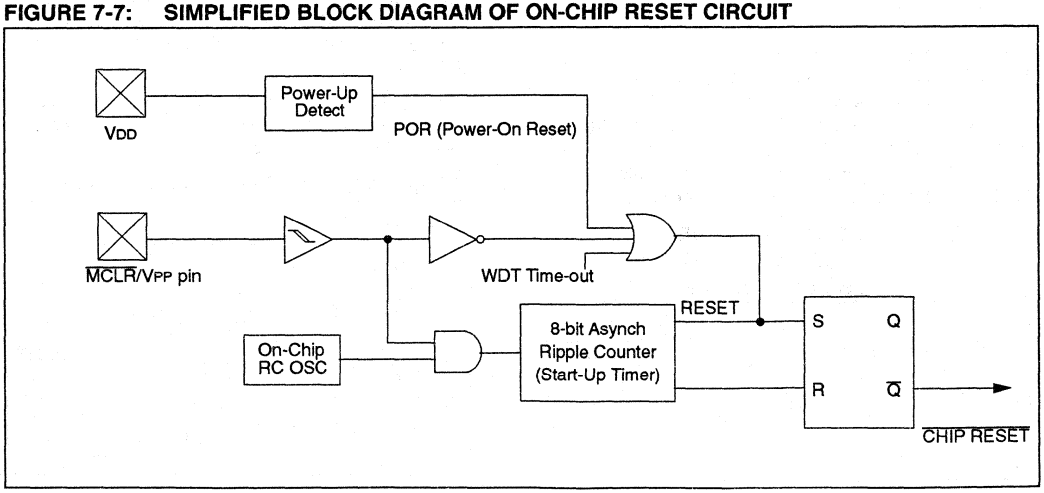
7.4.1 POWER-ON RESET (POR)

The PIC16C5X family incorporates an on chip Power-On Reset (POR) circuitry which provides an internal chip reset for most power-up situations. To use this feature the user merely needs to tie the $\overline{MCLR}$/VPP pin to VDD. Figure 7-14 shows the electrical structure of TMR0 inputs. A simplified block diagram of the on-chip Power-On Reset circuit is shown in Figure 7-7. The Power-On Reset circuit and the Device Reset Timer circuit are closely related. On power-up the reset latch is set and the DRT is reset. The DRT timer begins counting once it detects $\overline{MCLR}$ to be high. After the time-out period, which is typically 18 ms, it will reset the reset-latch and thus end the on-chip reset signal.

Figure 7-8 and Figure 7-9 show two power-up situations with relatively fast rise time on VDD. In Figure 7-8, VDD is allowed to rise and stabilize before bringing $\overline{MCLR}$ high. The chip will actually come out of reset TDRT msec after $\overline{MCLR}$ goes high. In Figure 7-9, the on-chip Power-On Reset feature is being used ($\overline{MCLR}$ and VDD are tied together). The VDD is stable before the start-up timer times out and there is no problem in getting a proper reset. Figure 7-10 depicts a potentially problematic situation where VDD rises too slowly. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip is, therefore, not guaranteed to function correctly.

To summarize, the on-chip POR is guaranteed to work if the rate of rise of VDD is no slower than 0.05V/ms and VDD starts from 0V. The on-chip POR time delay is too short for low frequency crystals which require much longer than 18 ms to start-up and stabilize. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times.

FIGURE 7-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



© 1995 Microchip Technology Inc.

7.4.2 DEVICE RESET TIMER (DRT)

The Device Reset Timer provides a fixed 18 ms nominal time-out on reset. The Device Reset Timer operates on an internal RC oscillator. The processor is kept in RESET as long as the DRT is active. The DRT delay allows the VDD to rise above VDD min., and for the oscillator to stabilize.

Oscillator circuits based on crystals or ceramic resonators require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a RESET condition for approximately 18 ms after the voltage on the $\overline{MCLR}$/VPP pin has reached a logic high (VIHMC) level. Thus, external RC networks connected to the $\overline{MCLR}$ input are not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications.

The Device Reset time delay will vary from chip to chip due to VDD, temperature, and process variation. See AC parameters for details.

The DRT will also be triggered upon a Watchdog Timer time-out. This is particularly important for applications using the WDT to wake the PIC16C5X from SLEEP mode automatically.

7.4.3 TIME-OUT SEQUENCE

Table 7-4 lists the reset conditions for the special function registers, while Table 7-5 lists the reset conditions for all the registers.

TABLE 7-3: $\overline{TO}$/$\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	= Low pulse on $\overline{MCLR}$ input

The $\overline{TO}$ and $\overline{PD}$ bit maintain their status (u) until a reset occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

TABLE 7-4: RESET CONDITIONS FOR SPECIAL REGISTERS

Condition	STATUS Addr: 03h	PCL Addr: 02h
Power-On Reset	0001 1xxx	1111 1111
$\overline{MCLR}$ reset during normal operation	000u uuuu(1)	1111 1111
$\overline{MCLR}$ reset during SLEEP	0001 0uuu	1111 1111
WDT reset during SLEEP	0000 0uuu	1111 1111
WDT reset during normal operation	0000 1uuu	1111 1111

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.
Note 1: $\overline{TO}$ and $\overline{PD}$ bits retain their last value until one of the other reset conditions occur.
2: The CLRWDT instruction will set the $\overline{TO}$ and $\overline{PD}$ bits.

TABLE 7-5: RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-On Reset	$\overline{MCLR}$ or WDT Reset
W	N/A	xxxx xxxx	uuuu uuuu
TRIS	N/A	1111 1111	1111 1111
OPTION	N/A	--11 1111	--11 1111
INDF	00h	—	—
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000? ?uuu (1)
FSR	04h	xxxx xxxx	1uuu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu
General Purpose register files	08-7Fh	xxxx xxxx	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented read as '0', ? = value depends on condition.
Note 1: See Table 7-4 for reset value for specific conditions.

FIGURE 7-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): CASE 1

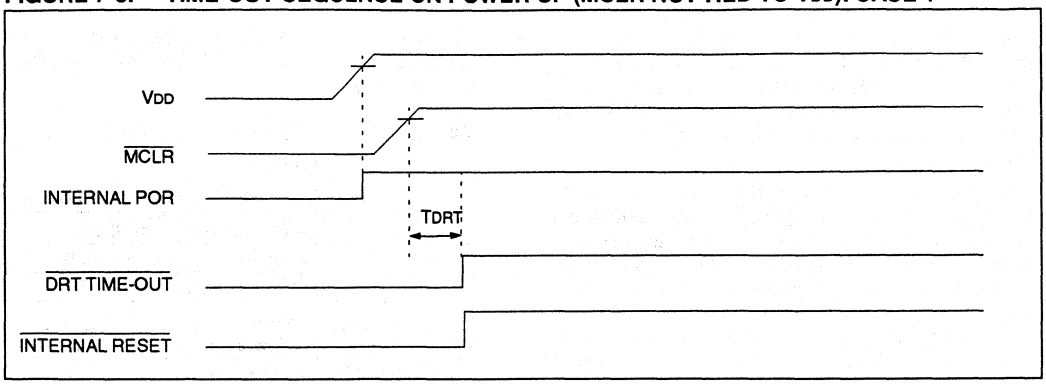


FIGURE 7-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD}): Case 2

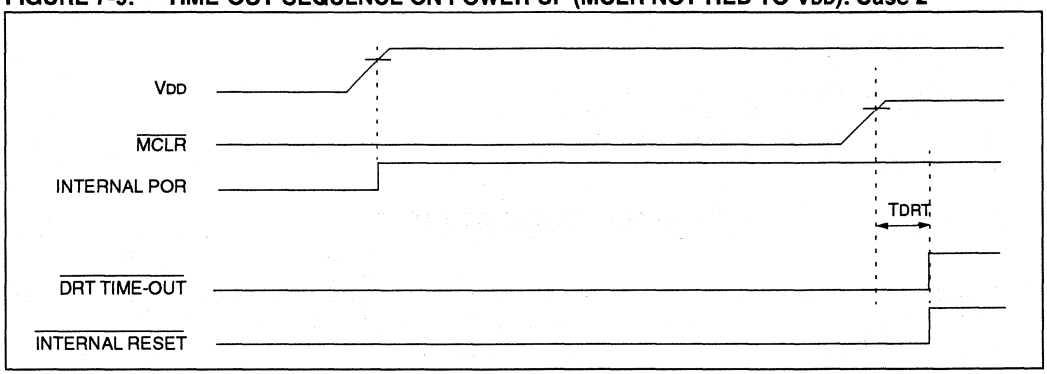
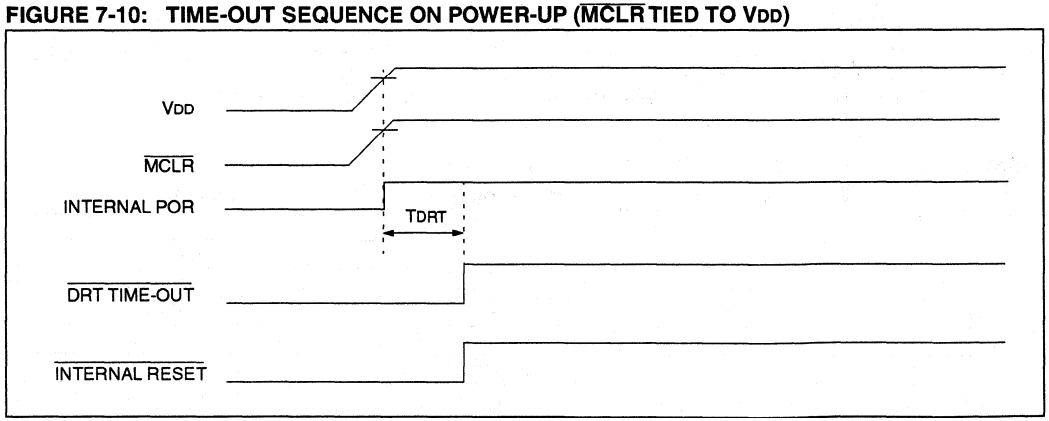
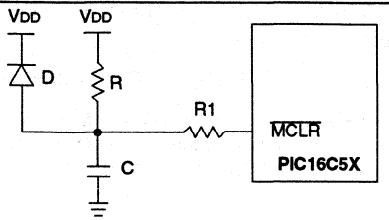


FIGURE 7-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD})



© 1995 Microchip Technology Inc.

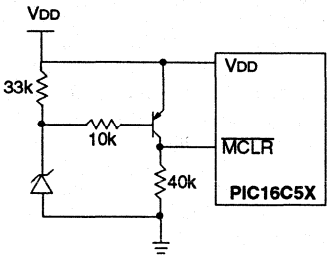
FIGURE 7-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note:

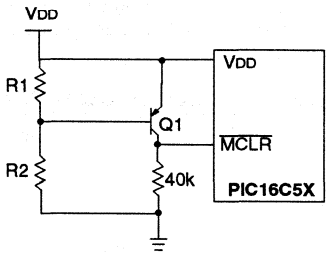
1. External Power-On Reset circuit is required only if VDD power-up is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$/VPP pin is 5 µA). A larger voltage drop will degrade VIH level on $\overline{MCLR}$/VPP pin.
3. R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$ pin breakdown due to

FIGURE 7-12: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when VDD goes below Vz + 0.7V (where Vz = Zener voltage).

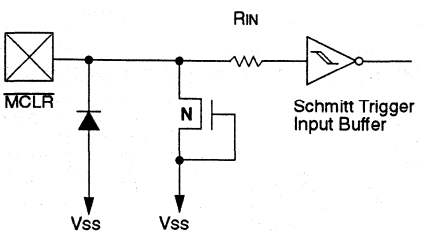
FIGURE 7-13: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V.$$

FIGURE 7-14: ELECTRICAL STRUCTURE OF $\overline{MCLR}$/VPP PIN



7.5 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins have been stopped, for example, by execution of a SLEEP instruction. During normal operation a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as a '0' (Section 7.1).

7.5.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). These periods vary with temperature, VDD and part to part process variations(see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

The $\overline{TO}$ bit (STATUS<4>) will be cleared upon a watchdog timer time-out.

7.5.2 WDT PROGRAMMING CONSIDERATIONS

Under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

FIGURE 7-15: WATCHDOG TIMER BLOCK DIAGRAM

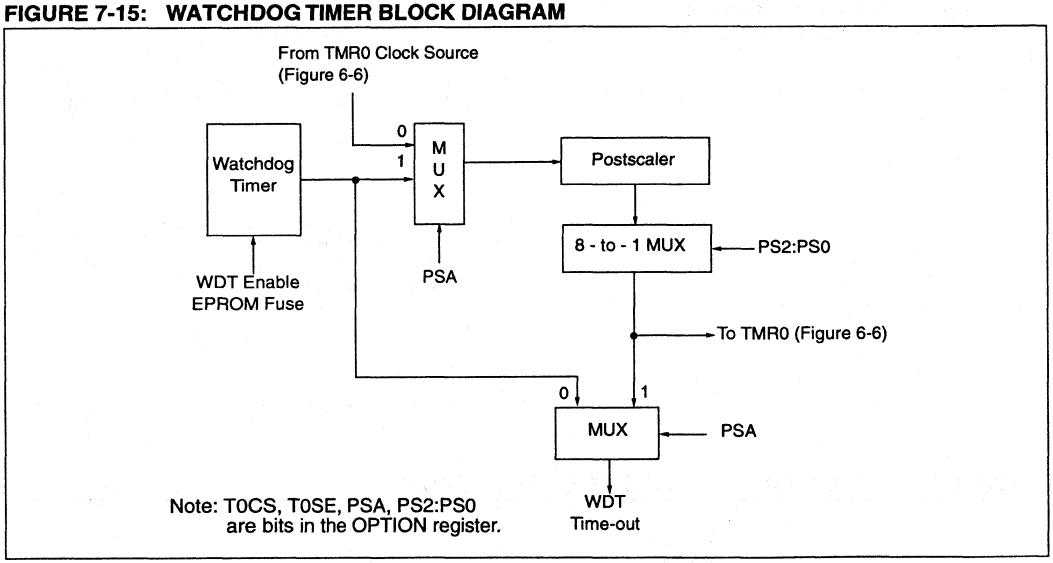


FIGURE 7-16: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Config. Word	—	—	—	—	CP	WDTE	FOSC1	FOSC0
OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0

Note 1: Shaded cells are not used by the Watchdog Timer.

© 1995 Microchip Technology Inc.

7.6 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low, or hi-impedance).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}$/VPP pin low.

For lowest current consumption while powered down, the T0CKI input should be at VDD or VSS and The $\overline{MCLR}$/VPP pin must be at a logic high level (VIHMC).

7.6.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. An external reset input on $\overline{MCLR}$/VPP pin.
2. A Watchdog timer time-out reset (if WDT was enabled).

Both of these events cause a device reset. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of device reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked.

The WDT is cleared when the device wakes from sleep, regardless of the wake-up source.

7.7 Code Protection

The code in the program memory can be protected by clearing the code protect bits.

In a code protected mode, the configuration word will not be protected, allowing reading of all bits.

Once code protected, all memory locations read out in a scrambled fashion. For EPROM devices, program memory locations 40h and above cannot be further programmed. However, the first 64 locations, 00h - 3Fh, may be programmed. These locations are not considered "secure".

7.8 ID Locations

Four memory locations are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify.

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '1's.

Note:	Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

8.0 INSTRUCTION SET SUMMARY

Each PIC16C5X instruction is a 12-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C5X instruction set summary in Table 8-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8- or 9-bit constant or literal value.

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

TABLE 8-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) d = 1 (store result in file register 'f') Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-Out bit
$\overline{PD}$	Power-Down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

© 1995 Microchip Technology Inc.

TABLE 8-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode				Status Affected	Notes
				MSb			LSb		
ADDWF	**f,d**	Add W and f	1	0001	11df	ffff		C,DC,Z	1,2,4
ANDWF	**f,d**	AND W with f	1	0001	01df	ffff		Z	2,4
CLRF	**f**	Clear f	1	0000	011f	ffff		Z	4
CLRW	**-**	Clear W	1	0000	0100	0000		Z	
COMF	**f, d**	Complement f	1	0010	01df	ffff		Z	
DECF	**f, d**	Decrement f	1	0000	11df	ffff		Z	2,4
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	0010	11df	ffff		None	2,4
INCF	**f, d**	Increment f	1	0010	10df	ffff		Z	2,4
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	0011	11df	ffff		None	2,4
IORWF	**f, d**	Inclusive OR W with f	1	0001	00df	ffff		Z	2,4
MOVF	**f, d**	Move f	1	0010	00df	ffff		Z	2,4
MOVWF	**f**	Move W to f	1	0000	001f	ffff		None	1,4
NOP	**-**	No Operation	1	0000	0000	0000		None	
RLF	**f, d**	Rotate left f through Carry	1	0011	01df	ffff		C	2,4
RRF	**f, d**	Rotate right f through Carry	1	0011	00df	ffff		C	2,4
SUBWF	**f, d**	Subtract W from f	1	0000	10df	ffff		C,DC,Z	1,2,4
SWAPF	**f, d**	Swap f	1	0011	10df	ffff		None	2,4
XORWF	**f, d**	Exclusive OR W with f	1	0001	10df	ffff		Z	2,4
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	0100	bbbf	ffff		None	2,4
BSF	**f, b**	Bit Set f	1	0101	bbbf	ffff		None	2,4
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	0110	bbbf	ffff		None	
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	0111	bbbf	ffff		None	
LITERAL AND CONTROL OPERATIONS									
ANDLW	**k**	AND literal with W	1	1110	kkkk	kkkk		Z	
CALL	**k**	Call subroutine	2	1001	kkkk	kkkk		None	1
CLRWDT	**k**	Clear watchdog timer	1	0000	0000	0100		$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Unconditional branch	2	101k	kkkk	kkkk		None	
IORLW	**k**	Inclusive OR literal with W	1	1101	kkkk	kkkk		Z	
MOVLW	**k**	Move Literal to W	1	1100	kkkk	kkkk		None	
OPTION	**k**	Load OPTION register	1	0000	0000	0010		None	
RETLW	**k**	Return, place literal in W	2	1000	kkkk	kkkk		None	
SLEEP	**-**	Go into standby mode	1	0000	0000	0011		$\overline{TO}$,$\overline{PD}$	
TRIS	**f**	Load TRIS register	1	0000	0000	0fff		None	3
XORLW	**k**	Exclusive OR Literal to W	1	1111	kkkk	kkkk		Z	

Note 1: the 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for `GOTO`. (Section 4.6)

2: When an I/O register is modified as a function of itself (e.g. `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

3: The instruction `TRIS f`, where f = 5, 6, or 7 causes the contents of the W register to be written to the tristate latches of PORTA, B or C, respectively. A '1' forces the pin to a hi-impedance state and disables the output buffers.

4: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared (if assigned to TMR0).

© 1995 Microchip Technology Inc.

ADDWF **Add W and f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

0001	11df	ffff

Description: Add the contents of the W register and register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

0001	01df	ffff

Description: The contents of the W register are AND'ed with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ANDWF   FSR,   1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

ANDLW **And literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

1110	kkkk	kkkk

Description: The contents of the W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example:
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

0100	bbbf	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example:
```
BCF     FLAG_REG,   7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

© 1995 Microchip Technology Inc.

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow (f<b>)$

Status Affected: None

Encoding:

0101	bbbf	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example:
```
BSF     FLAG_REG,   7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **Bit Test f, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

0110	bbbf	ffff

Description: If bit 'b' in register 'f' is 0 then the next instruction is skipped.
If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and an NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address (HERE)
After Instruction
if FLAG<1> = 0,
PC = address (TRUE);
if FLAG<1> = 1,
PC = address (FALSE)

BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: $0 \le f \le 127$
$0 \le b < 7$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

0111	bbbf	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and an NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSS   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0,
PC = address (FALSE);
if FLAG<1> = 1,
PC = address (TRUE)

© 1995 Microchip Technology Inc.

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC) + 1→ Top of Stack;
k → PC<8:0>;
(STATUS<6:5>) → PC<10:9>;
0 → PC<8>

Status Affected: None

Encoding:

1001	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eight bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example:
```
HERE    CALL    THERE
```
Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE)

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → (f);
1 → Z

Status Affected: Z

Encoding:

0000	011f	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example:
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W);
1 → Z

Status Affected: Z

Encoding:

0000	0100	0000

Description: The W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example:
```
CLRW
```
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0100

Description: The CLRWDT instruction resets the WDT. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example:
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

2

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: $(\bar{f}) \rightarrow (dest)$

Status Affected: Z

Encoding:

0010	01df	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13

After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – 1 → (dest)

Status Affected: Z

Encoding:

0000	11df	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
DECF    CNT,  1
```
Before Instruction
CNT = 0x01
Z = 0

After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – 1 → d; skip if result = 0

Status Affected: None

Encoding:

0010	11df	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE      DECFSZ    CNT, 1
          GOTO      LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address (HERE)

After Instruction
CNT = CNT - 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE+1)

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: k → PC<8:0>;
STATUS<6:5> → PC<10:9>

Status Affected: None

Encoding:

101k	kkkk	kkkk

Description: GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example:
```
GOTO THERE
```
After Instruction
PC = address (THERE)

© 1995 Microchip Technology Inc.

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

0010	10df	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
INCF    CNT,   1
```
Before Instruction
CNT = 0xFF
Z = 0

After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

0011	11df	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, then the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE      INCFSZ     CNT,  1
          GOTO       LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address (HERE)

After Instruction
CNT = CNT + 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE +1)

IORLW **Inclusive OR literal with W**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

1101	kkkk	kkkk

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example:
```
IORLW   0x35
```
Before Instruction
W = 0x9A

After Instruction
W = 0xBF

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (dest)

Status Affected: Z

Encoding:

0001	00df	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
IORWF          RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91

After Instruction
RESULT = 0x13
W = 0x93

© 1995 Microchip Technology Inc.

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

0010	00df	ffff

Description: The contents of register 'f' is moved to destination 'd'. If 'd' is 0, destination is the W register. If 'd' is 1, the destination is file register 'f'. 'd' is 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:
```
MOVF    FSR,  0
```
After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

1100	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into the W register. The don't cares will assemble as 0s.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

0000	001f	ffff

Description: Move data from the W register to register 'f'.

Words: 1

Cycles: 1

Example:
```
MOVWF   TEMP_REG
```
Before Instruction
TEMP_REG = 0xFF
W = 0x4F

After Instruction
TEMP_REG = 0x4F
W = 0x4F

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```


© 1995 Microchip Technology Inc.

OPTION — **Load OPTION Register**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

0000	0000	0010

Description: The content of the W register is loaded into the OPTION register.

Words: 1

Cycles: 1

Example

```
OPTION
```

Before Instruction
W = 0x07

After Instruction
OPTION = 0x07

RETLW — **Return, place literal in W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

1000	kkkk	kkkk

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

Words: 1

Cycles: 2

Example:

```
CALL TABLE  ;W contains
            ;table
TABLE       ;offset value
•           ;W now has table
              value•
•
ADDWF PC    ;W = offset
RETLW k1    ;Begin table
RETLW k2    ;
•
•
•
RETLW kn    ; End of table
```

Before Instruction
W = 0x07

After Instruction
W = value of k7

RLF — **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

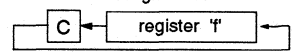
Operation: See description below

Status Affected: C

Encoding:

0011	01df	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example:

```
RLF     REG1,0
```

Before Instruction
REG1 = 1110 0110
C = 0

After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

© 1995 Microchip Technology Inc.

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

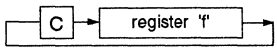
Operation: See description below

Status Affected: C

Encoding:

0011	00df	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RRF     REG1,0
```
Before Instruction
```
REG1  =  1110 0110
C     =  0
```
After Instruction
```
REG1  =  1110 0110
W     =  0111 0011
C     =  1
```

SLEEP **Enter SLEEP Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0011

Description: Time-out status bit ($\overline{TO}$) is set. The power down status bit ($\overline{PD}$) is cleared. The WDT and its prescaler are cleared.

The processor is put into SLEEP mode with the oscillator stopped. See section on SLEEP for more details.

Words: 1

Cycles: 1

Example:
```
SLEEP
```

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) → (dest)

Status Affected: C, DC, Z

Encoding:

0000	10df	ffff

Description: Subtract (2's complement method) the W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1, 1
```
Before Instruction
```
REG1  =  3
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  1
W     =  2
C     =  1    ; result is positive
```

Example 2:

Before Instruction
```
REG1  =  2
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  0
W     =  2
C     =  1    ; result is zero
```

Example 3:

Before Instruction
```
REG1  =  1
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  FF
W     =  2
C     =  0    ; result is negative
```


© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label*] SWAPF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f<3:0>) → (dest<7:4>);
(f<7:4>) → (dest<3:0>)

Status Affected: None

Encoding:

0011	10df	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example
```
SWAPF   REG1, 0
```
Before Instruction
REG1 = 0xA5
After Instruction
REG1 = 0xA5
W = 0X5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: 5 ≤ f ≤ 7

Operation: (W) → TRIS register f

Status Affected: None

Encoding:

0000	0000	0fff

Description: TRIS register 'f' (f = 5, 6, or 7) is loaded with the contents of the W register

Words: 1

Cycles: 1

Example
```
TRIS    PORTA
```
Before Instruction
W = 0XA5
After Instruction
TRISA = 0XA5

XORLW **Exclusive OR literal with W**

Syntax: [*label*] XORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

1111	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
XORLW 0xAF
```
Before Instruction
W = 0xB5
After Instruction
W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

0001	10df	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
XORWF   REG     1
```
Before Instruction
REG = 0xAF
W = 0xB5
After Instruction
REG = 0x1A
W = 0xB5

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

9.0 ELECTRICAL CHARACTERISTICS

9.1 Absolute Maximum Ratings†

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to V_{DD} + 0.6V
Total power dissipation (Note 1) ..... 800 mW
Max. current out of V_{SS} pin ..... 150 mA
Max. current into V_{DD} pin ..... 50 mA
Max. current into an input pin (T0CKI only) ..... ±500 µA
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{DD}$) ..... ±20 mA
Output clamp current, I_{OK} ($V_0 < 0$ or $V_0 > V_{DD}$) ..... ±20 mA
Max. output current sunk by any I/O pin ..... 25 mA
Max. output current sourced by any I/O pin ..... 20 mA
Max. output current sourced by a single I/O port (PORTA, B or C) ..... 40 mA
Max. output current sunk by a single I/O port (PORTA, B or C) ..... 50 mA

Note 1: Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$$

†NOTICE: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 9-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C5X-04	16C5X-10	16C5X-20	16LC5X-04
RC	V_{DD}: 3.0V to 6.25V I_{DD}: 2.4 mA Max. at 5.5V I_{PD}: 4 µA Max. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 2.5V to 6.25V I_{DD}: 0.5 mA typ. at 5.5V I_{PD}: 4 µA Max. at 2.5V WDT dis Freq: 2 MHz Max.
XT	V_{DD}: 3.0V to 6.25V I_{DD}: 2.4 mA Max. at 5.5V I_{PD}: 5 µA Max. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 2.5V to 6.25V I_{DD} 0.5 mA typ. at 5.5V I_{PD}: 4 µA Max. at 2.5V WDT dis Freq: 4 MHz Max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 0.3 µA typ. at 3.0V WDT dis Freq: 4 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 8 mA Max. at 5.5V I_{PD}: 4 µA Max. at 3.0V WDT dis Freq: 10 MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 16 mA Max. at 5.5V I_{PD}: 4 µA Max. at 3.0V WDT dis Freq: 20 MHz Max.	Do not use in HS mode
LP	V_{DD}: 4.5V to 5.5V I_{DD}: 17 µA typ. at 32 kHz, 3.0V I_{PD}: 0.3 µA typ. at 3.0V WDT dis Freq: 200 kHz typ.	Do not use in LP mode	Do not use in LP mode	V_{DD}: 2.5V to 6.25V I_{DD}: 11 µA typ. at 32 kHz, 2.5V I_{PD}: 0.25 µA typ. at 2.5V WDT dis Freq: 200 kHz typ.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

TABLE 9-2: DC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (COMMERCIAL)

DC Characteristics Power Supply Pins		Standard Operating Conditions Operating Temperature 0°C ≤ TA ≤ +70°C, unless otherwise stated. Operating Voltage VDD = 3.0V to 5.5V, unless otherwise stated.				
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5X-XT	VDD	3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5X-RC		3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5X-HS		4.5		5.5	V	FOSC = DC to 20 MHz
PIC16C5X-LP		2.5		6.25	V	FOSC = DC to 40 kHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR		VSS		V	See Section 7.4 for details on Power-On Reset
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*			V/ms	See Section 7.4 for details on Power-On Reset
Supply Current (Note 2)						
PIC16C5X-XT	IDD		1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5X-RC (Note 5)			1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5X-HS			4.8	10	mA	FOSC = 10 MHz, VDD = 5.5V
			9.0	20	mA	FOSC = 20 MHz, VDD = 5.5V
PIC16C5X-LP			15	32	µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 4)						
PIC16C5X	IPD		4	12	µA	VDD = 3.0V, WDT enabled
			0.6	9	µA	VDD = 3.0V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

5: Does not include current through Rext. The current through the resistor can be estimated by the formula: IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

TABLE 9-3: DC CHARACTERISTICS: PIC16C5XI-RC, XT, HS, LP (INDUSTRIAL)

DC Characteristics Power Supply Pins		Standard Operating Conditions Operating Temperature -40°C ≤ TA ≤ +85°C, unless otherwise stated. Operating Voltage VDD = 3.5V to 5.5V, unless otherwise stated.				
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5XI-XT	VDD	3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5XI-RC		3.0		6.25	V	FOSC = DC to 4 MHz
PIC16C5XI-HS		4.5		5.5	V	FOSC = DC to 20 MHz
PIC16C5XI-LP		2.5		6.25	V	FOSC = DC to 40 kHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR		VSS		V	See Section 7.4 for details on Power-On Reset
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*			V/ms	See Section 7.4 for details on Power-On Reset
Supply Current (Note 2)						
PIC16C5XI-XT	IDD		1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5XI-RC (Note 5)			1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5XI-HS			4.8	10	mA	FOSC = 10 MHz, VDD = 5.5V
			9.0	20	mA	FOSC = 20 MHz, VDD = 5.5V
PIC16C5XI-LP			19	40	µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 4)						
PIC16C5XI	IPD		5	14	µA	VDD = 3.0V, WDT enabled
			0.6	12	µA	VDD = 3.0V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

5: Does not include current through Rext. The current through the resistor can be estimated by the formula: IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

TABLE 9-4: DC CHARACTERISTICS: PIC16C5XE-RC, XT, HS, LP (AUTOMOTIVE)

DC Characteristics Power Supply Pins		Standard Operating Conditions Operating Temperature -40°C ≤ TA ≤ +125°C, unless otherwise stated. Operating Voltage VDD = 3.5V to 5.5V, unless otherwise stated.				
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Supply Voltage						
PIC16C5XE-XT	VDD	3.25		6.0	V	FOSC = DC to 4 MHz
PIC16C5XE-RC		3.25		6.0	V	FOSC = DC to 4 MHz
PIC16C5XE-HS		4.5		5.5	V	FOSC = DC to 20 MHz
PIC16C5XE-LP		2.5		6.0	V	FOSC = DC to 40 kHz
RAM Data Retention Voltage (Note 3)	VDR		1.5		V	Device in SLEEP mode
VDD Start Voltage to Guarantee Power-On Reset	VPOR		VSS		V	See Section 7.4 for details on Power-On Reset
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*			V/ms	See Section 7.4 for details on Power-On Reset
Supply Current (Note 2)						
PIC16C5XE-XT	IDD		1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5XE-RC (Note 5)			1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
PIC16C5XE-HS			4.8	10	mA	FOSC = 10 MHz, VDD = 5.5V
			9.0	20	mA	FOSC = 16 MHz, VDD = 5.5V
PIC16C5XE-LP			25	55	µA	FOSC = 32 kHz, VDD = 3.25V, WDT disabled
Power Down Current (Note 4)						
PIC16C5XE	IPD		5	22	µA	VDD = 3.25V, WDT enabled
			0.8	18	µA	VDD = 3.25V, WDT disabled

* These parameters are based on characterization and are not tested.

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

4: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

5: Does not include current through Rext. The current through the resistor can be estimated by the formula: IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

TABLE 9-5: DC CHARACTERISTICS:PIC16C5X-RC, XT, HS, LP (COMMERCIAL) PIC16C5XI-RC, XT, HS, LP (INDUSTRIAL)

DC Characteristics All Pins Except Power Supply Pins	Standard Operating Conditions Operating Temperature -40°C ≤ TA ≤ +85°C (for industrial) 0°C ≤ TA ≤ +70°C (for commercial) Operating Voltage VDD range is described in Section 9.1 and Section 9.2.					
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$ (Schmitt Trigger)		VSS		0.15 VDD	V	
T0CKI (Schmitt Trigger)		VSS		0.15 VDD	V	
OSC1 (Schmitt Trigger)		VSS		0.15 VDD	V	PIC16C5X-RC only (Note 5)
OSC1		VSS		0.3 VDD	V	PIC16C5X-XT, HS, LP
Input High Voltage						
I/O ports	VIH	0.45 VDD		VDD	V	For all VDD (Note 6)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 6)
		0.36 VDD		VDD	V	VDD > 5.5V
$\overline{MCLR}$ (Schmitt Trigger)		0.85 VDD		VDD	V	
T0CKI (Schmitt Trigger)		0.85 VDD		VDD	V	
OSC1 (Schmitt Trigger)		0.85 VDD		VDD	V	PIC16C5X-RC only (Note 5)
OSC1		0.7 VDD		VDD	V	PIC16C5X-XT, HS, LP
Input Leakage Current (Notes 3,4)						**For VDD ≤ 5.5V**
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS + 0.25V
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD
T0CKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD PIC16C5X-XT, HS, LP
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

Note 1: Data in the column labeled "Typical" is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

2: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

3: The leakage current on the $\overline{MCLR}$/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltage.

4: Negative current is defined as coming out of the pin.

5: For PIC16C5X-RC devices, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C5X be driven with external clock in RC mode.

6: The user may use the better of the two specifications.

© 1995 Microchip Technology Inc.

TABLE 9-6: DC CHARACTERISTICS: PIC16C5X-RC, XT, HS, LP (AUTOMOTIVE)

DC Characteristics All Pins Except Power Supply Pins		Standard Operating Conditions Operating Temperature -40°C ≤ TA ≤ +125°C Operating Voltage VDD range is described in Section 9.1 and Section 9.2.				
Characteristic	**Sym**	**Min**	**Typ (Note 1)**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.15 VDD	V	Pin at hi-impedance
MCLR (Schmitt Trigger)		VSS		0.15 VDD	V	
T0CKI (Schmitt Trigger)		VSS		0.15 VDD	V	
OSC1 (Schmitt Trigger)		VSS		0.15 VDD	V	PIC16C5X-RC only (Note 5)
OSC1		VSS		0.3 VDD	V	PIC16C5X-XT, HS, LP
Input High Voltage						
I/O ports	VIH	0.45 VDD		VDD	V	For all VDD (Note 6)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 6)
		0.36 VDD		VDD	V	VDD > 5.5V
MCLR (Schmitt Trigger)		0.85 VDD		VDD	V	
T0CKI (Schmitt Trigger)		0.85 VDD		VDD	V	
OSC1 (Schmitt Trigger)		0.85 VDD		VDD	V	PIC16C5X-RC only (Note 5)
OSC1		0.7 VDD		VDD	V	PIC16C5X-XT, HS, LP
Input Leakage Current (Notes 3,4)						**For VDD ≤ 5.5V**
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR		-5			µA	VPIN = VSS + 0.25V
MCLR			0.5	+5	µA	VPIN = VDD
T0CKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD PIC16C5X-XT, HS, LP
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (PIC16C5X-RC)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

Note 1: Data in the column labeled Typical is based on characterization results at 25°C. This data is for design guidance only and is not tested for, or guaranteed by Microchip Technology.

2: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

3: The leakage current on the MCLR/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltage.

4: Negative current is defined as coming out of the pin.

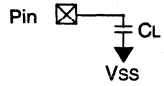
5: For PIC16C5X-RC devices, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C5X be driven with external clock in RC mode.

6: The user may use the better of the two specifications.

© 1995 Microchip Technology Inc.

9.2 Timing Diagrams and Specifications

FIGURE 9-1: LOAD CONDITIONS



C_L = 50 pF for all pins except OSC2
15 pF for OSC2 in XT, HS or LP modes when external clock is used to drive OSC1

© 1995 Microchip Technology Inc.

FIGURE 9-2: EXTERNAL CLOCK TIMING

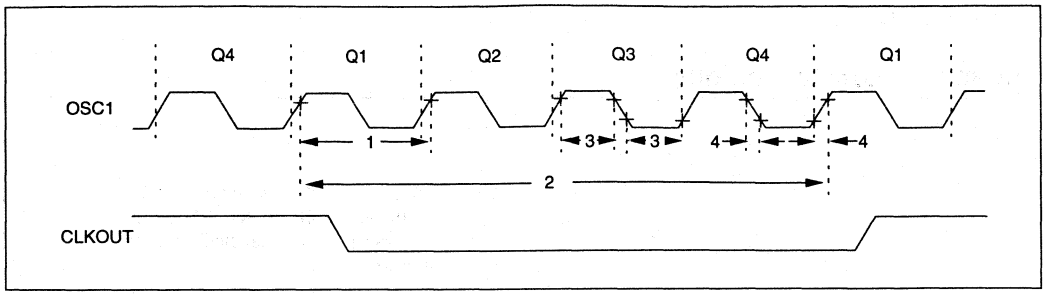


TABLE 9-7: EXTERNAL CLOCK TIMING REQUIREMENTS

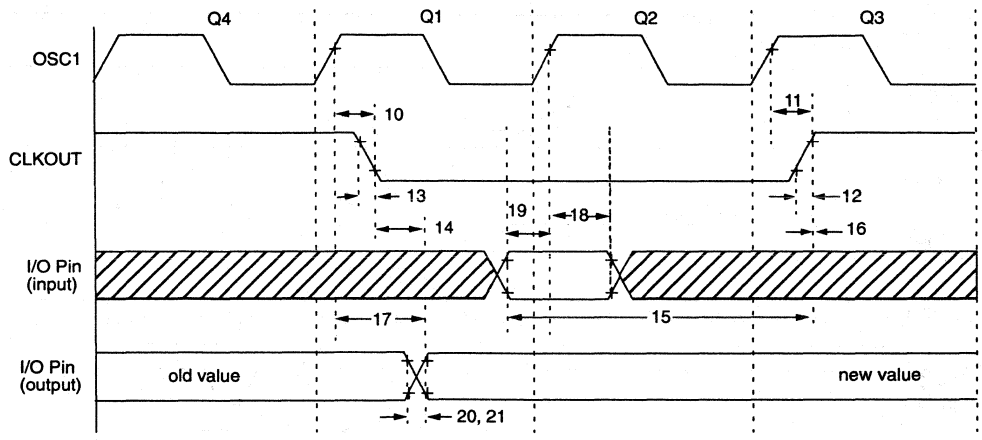
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			DC	—	4	MHz	XT osc mode
			DC	—	20	MHz	HS osc mode (Comm/Indust)
			DC	—	16	MHz	HS osc mode (Automotive)
			DC	—	40	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode (Comm/Indust)
			4	—	16	MHz	HS osc mode (Automotive)
			5	—	40	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	—	ns	XT osc mode
			50	—	—	ns	HS osc mode
			100	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			62.5	—	250	ns	HS osc mode (Comm/Indust)
			50	—	250	ns	HS osc mode (Automotive)
			100	—	200	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	1.0	—	DC	µs	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 1995 Microchip Technology Inc.

FIGURE 9-3: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 9-8: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5 TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	Note 2
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	Note 2
21	TioF	Port output fall time	—	10	25	ns	Note 2

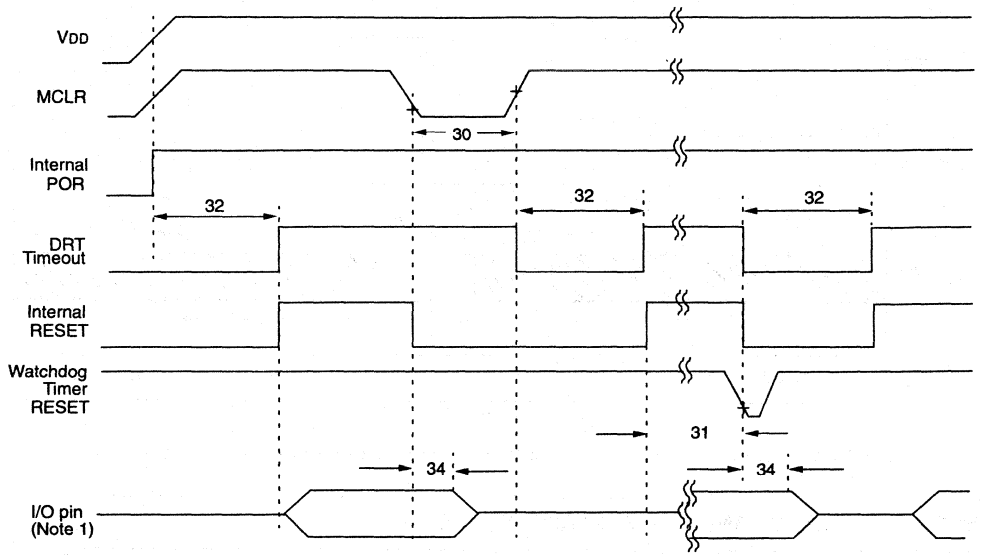
* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc.

2: See Figure 9-1 for loading conditions.

FIGURE 9-4: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of hi-impedance mode by enabling the output drivers in software.

TABLE 9-9: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	9*	18	30*	ms	VDD = 5V, -40°C to +125°C
32	TDRT	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5V, -40°C to +125°C
34	TioZ	I/O Hi-impedance from $\overline{\text{MCLR}}$ Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

FIGURE 9-5: TIMER0 CLOCK TIMINGS

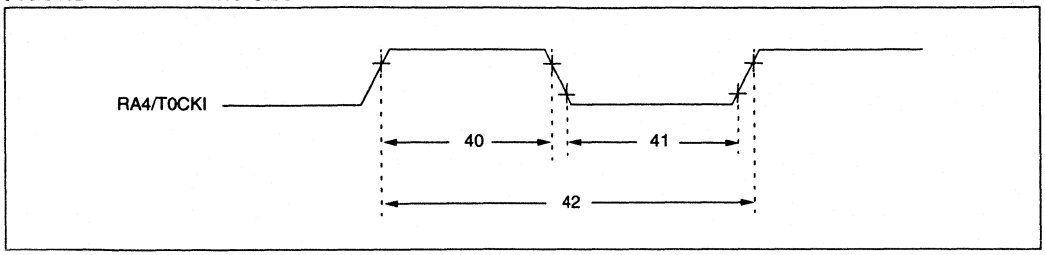


TABLE 9-10: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NOTES:

© 1995 Microchip Technology Inc.

10.0 DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

FIGURE 10-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

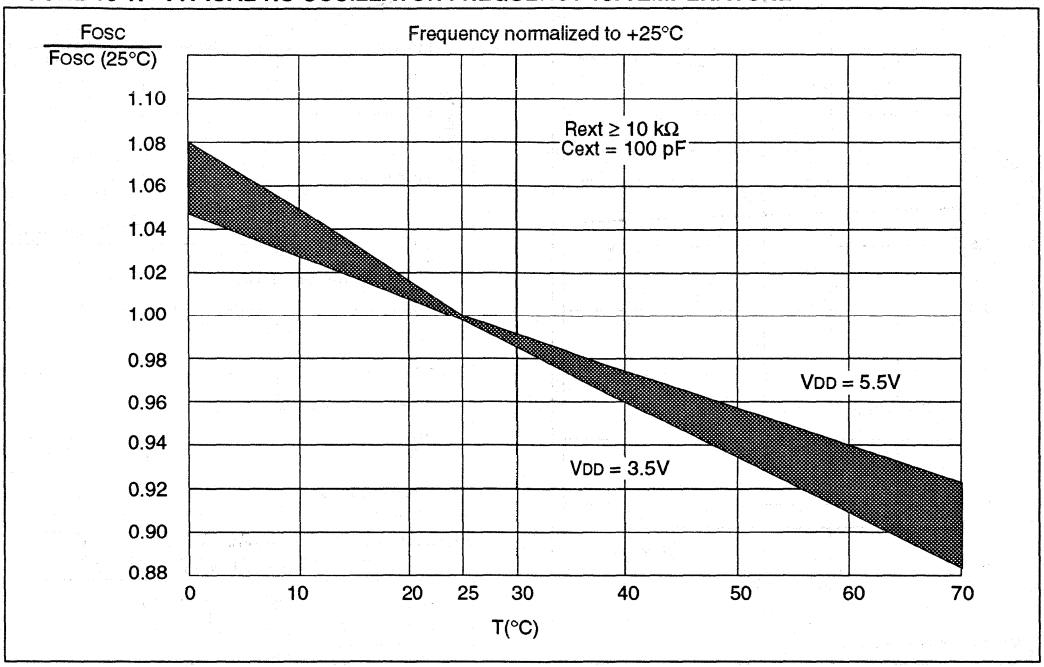


TABLE 10-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20 pF	3.3k	4.973 MHz	± 27%
	5k	3.82 MHz	± 21%
	10k	2.22 MHz	± 21%
	100k	262.15 kHz	± 31%
100 pF	3.3k	1.63 MHz	± 13%
	5k	1.19 MHz	± 13%
	10k	684.64 kHz	± 18%
	100k	71.56 kHz	± 25%
300 pF	3.3k	660.0 kHz	± 10%
	5.k	484.1 kHz	± 14%
	10k	267.63 kHz	± 15%
	160k	29.44 kHz	± 19%

The frequencies are measured on DIP packages.

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

FIGURE 10-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

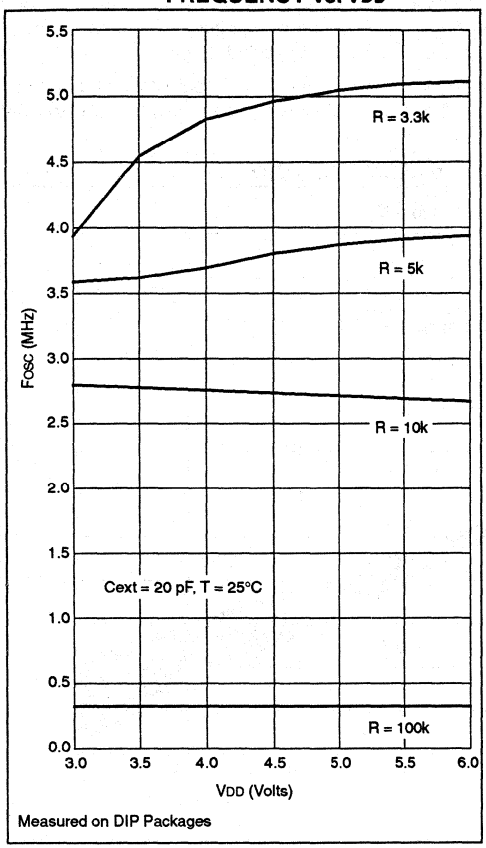


FIGURE 10-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

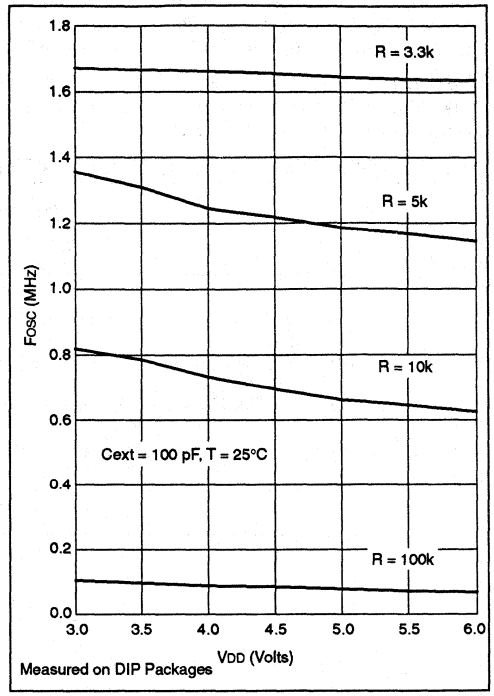
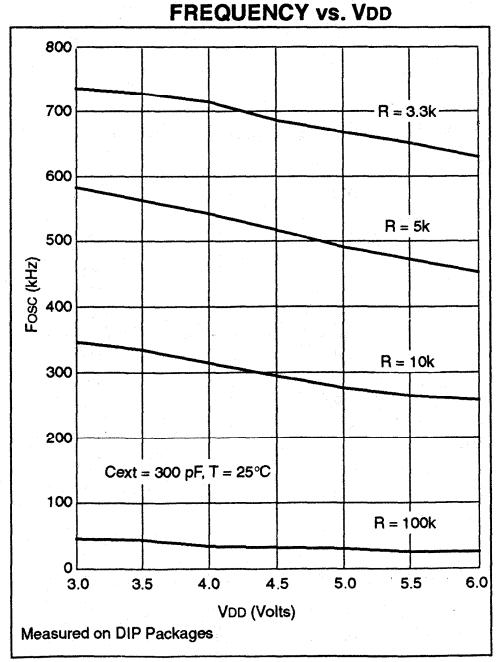


FIGURE 10-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 10-5: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

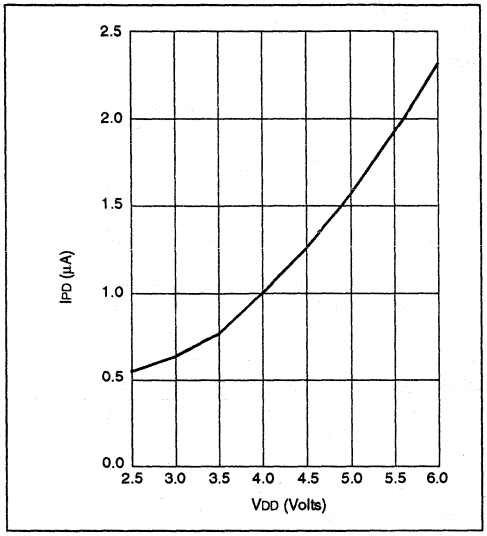


FIGURE 10-6: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

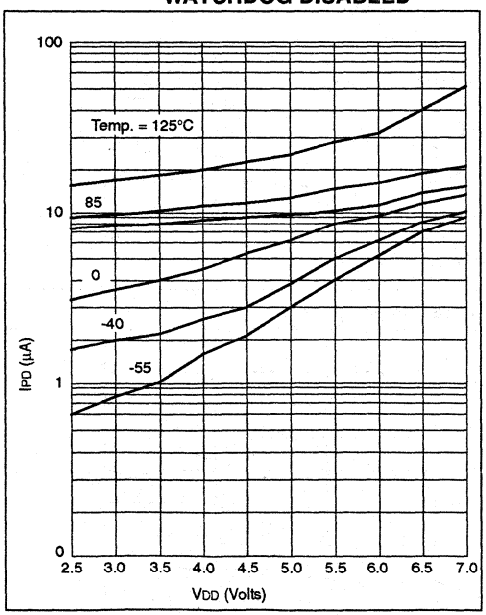


FIGURE 10-7: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

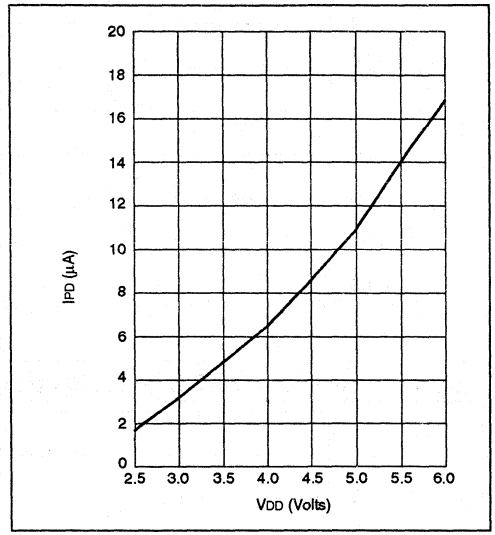
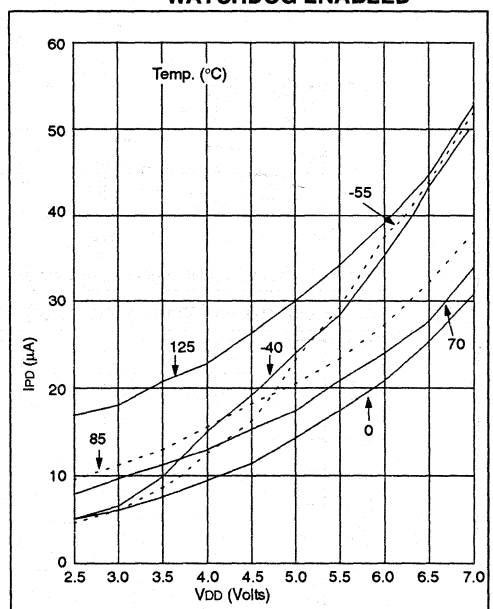


FIGURE 10-8: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



IPD, with WDT enabled, has two components: The leakage current which increases with higher temperature and the operating current of the WDT logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 10-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD

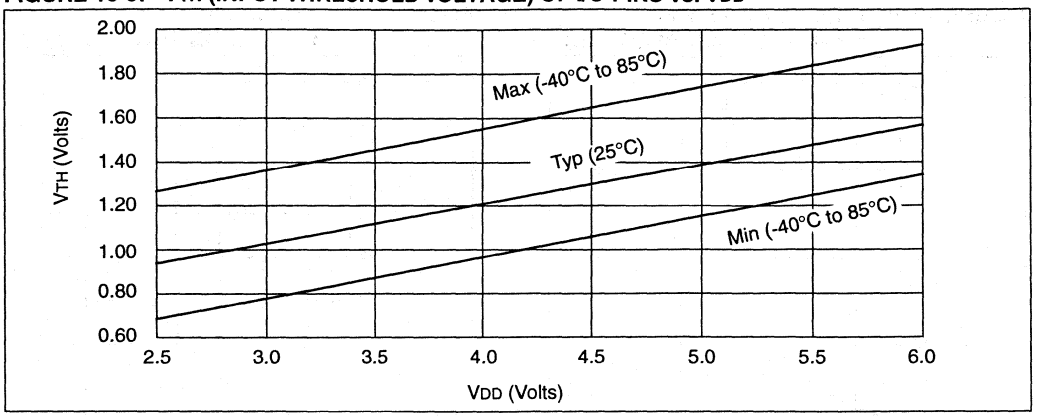
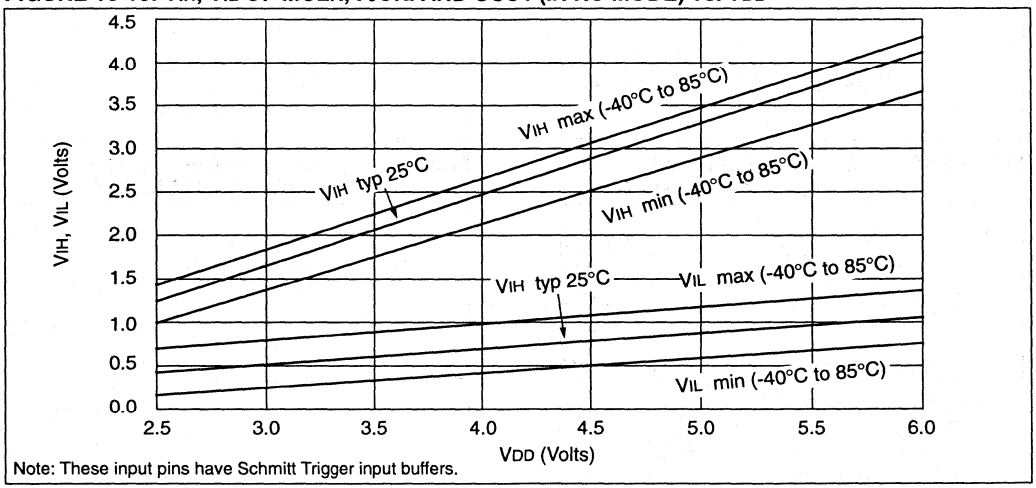
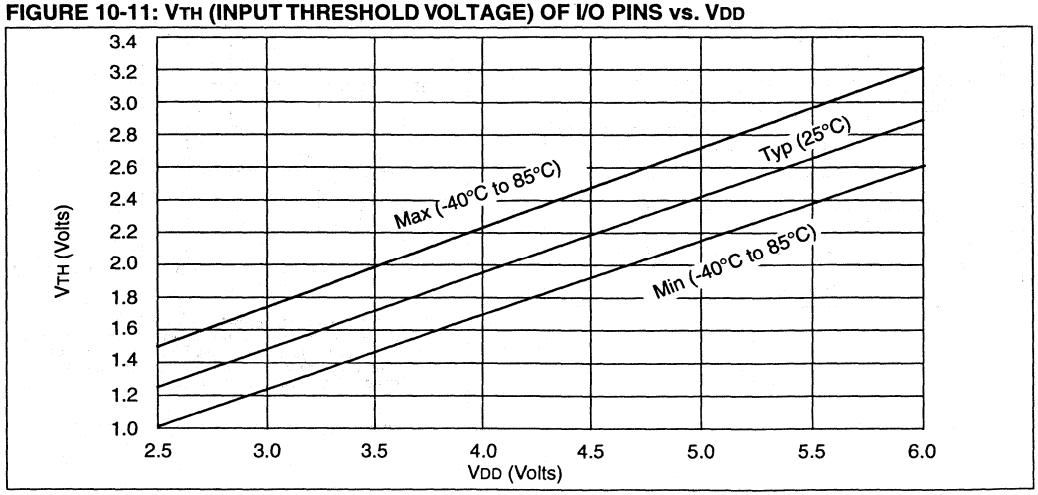


FIGURE 10-10: VIH, VIL OF $\overline{MCLR}$, T0CKI AND OSC1 (IN RC MODE) vs. VDD



Note: These input pins have Schmitt Trigger input buffers.

FIGURE 10-11: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 10-12: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

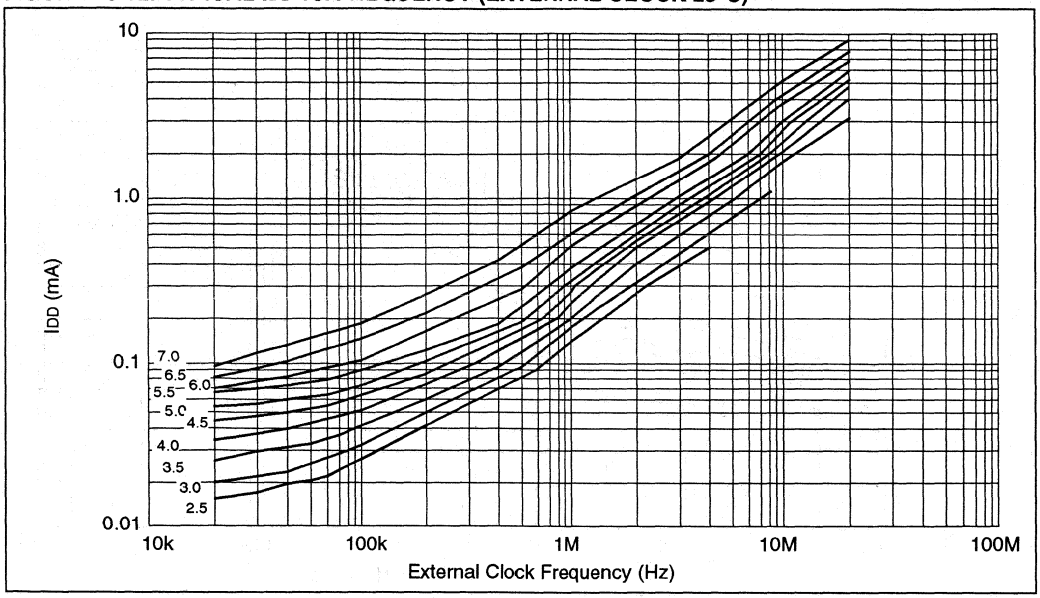


FIGURE 10-13: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -40°C TO +85°C)

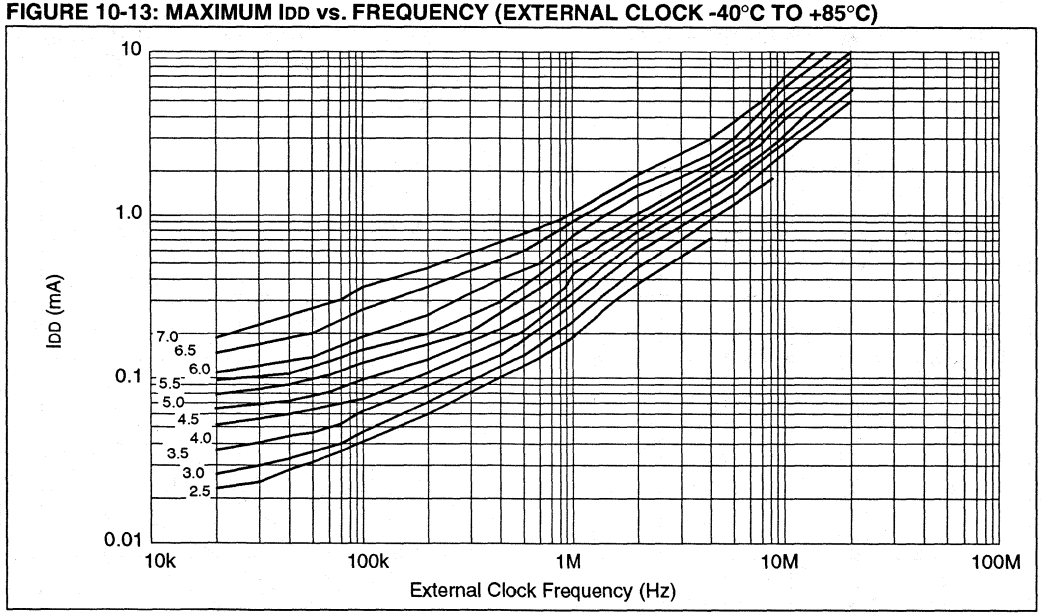


FIGURE 10-14: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -55°C TO +125°C)

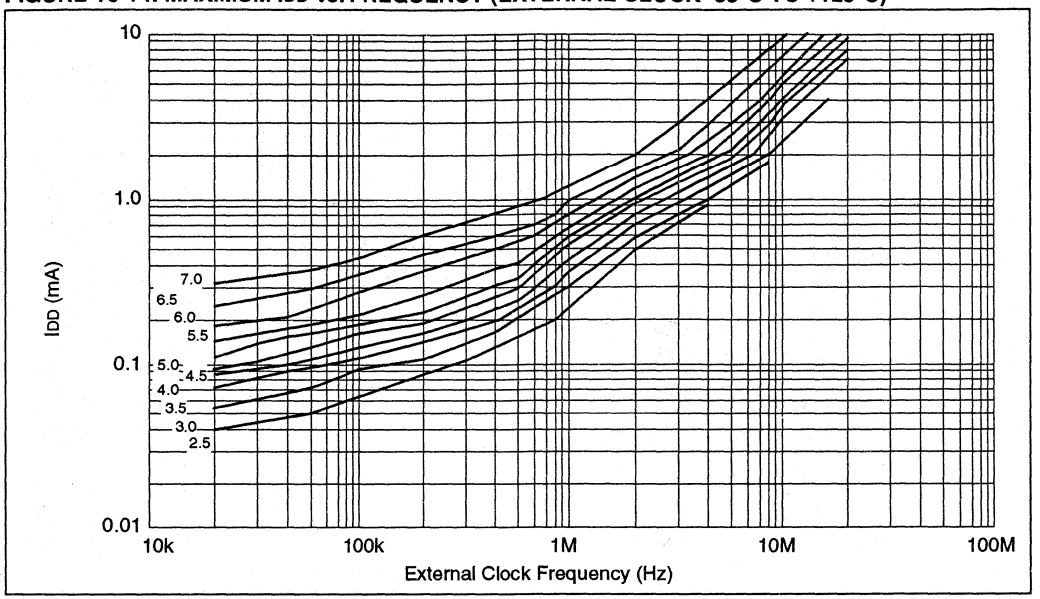


FIGURE 10-15: WDT TIMER TIME-OUT PERIOD vs. VDD

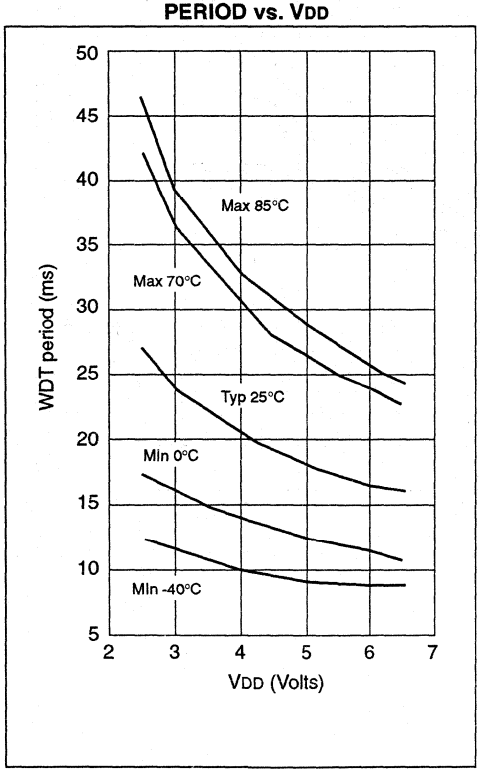
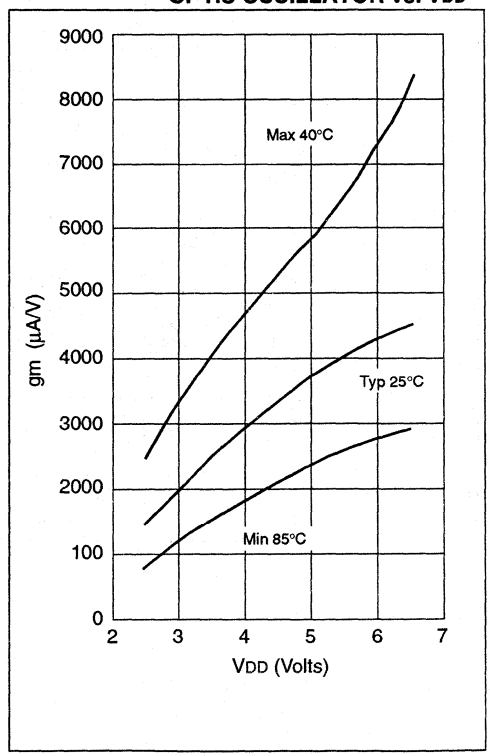


FIGURE 10-16: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 10-17: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. VDD

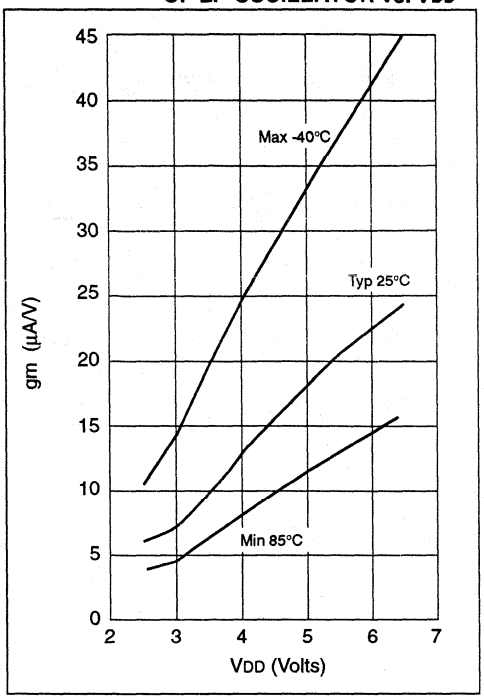


FIGURE 10-18: IOH vs. VOH, VDD = 3V

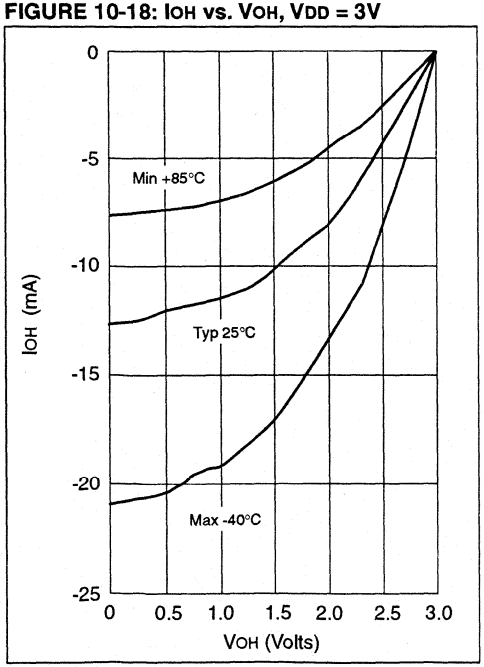


FIGURE 10-19: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

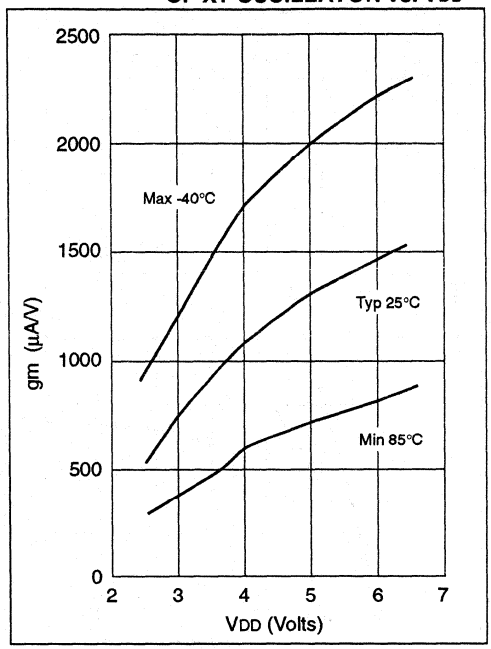


FIGURE 10-20: IOH vs. VOH, VDD = 5V

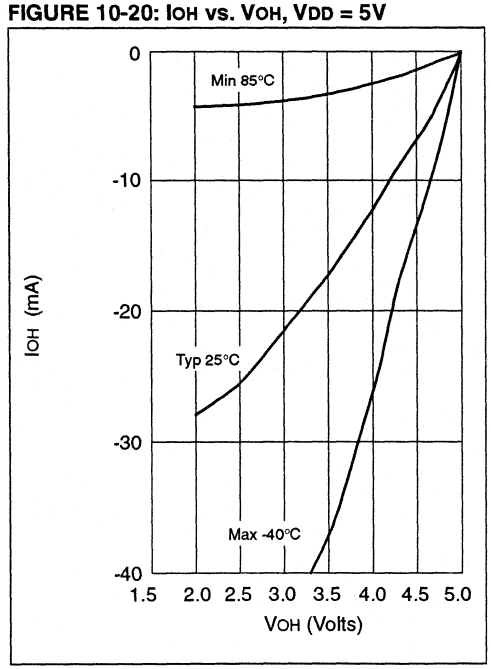


FIGURE 10-21: IOL vs. VOL, VDD = 3V

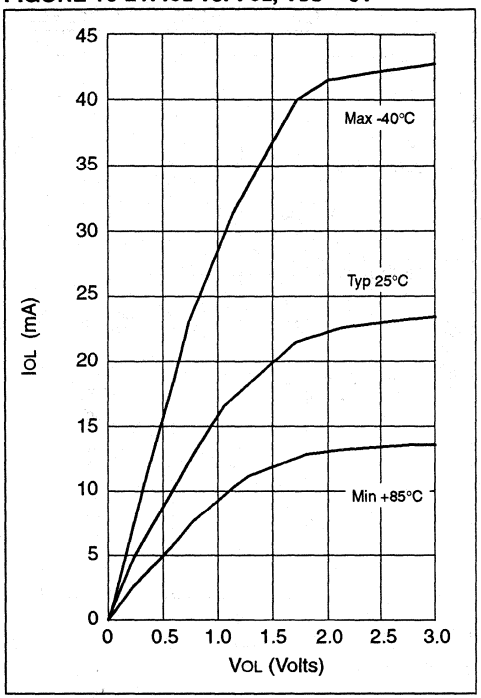
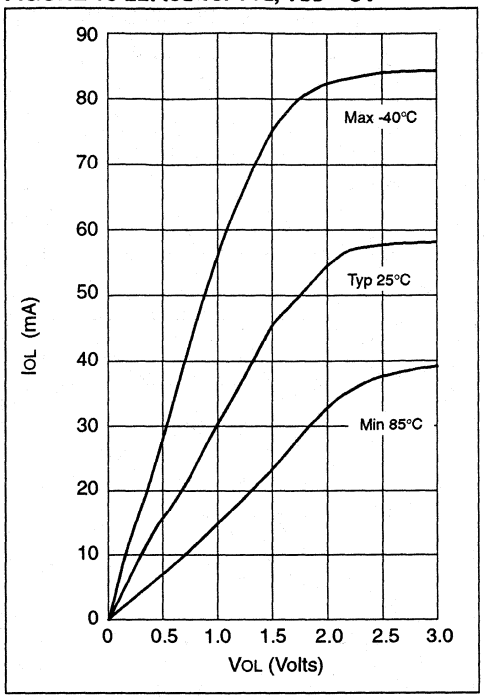


FIGURE 10-22: IOL vs. VOL, VDD = 5V



© 1995 Microchip Technology Inc.

11.0 DEVELOPMENT SUPPORT

11.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®-MP)

11.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 11-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 (and better) machines in the Microsoft Windows™ 3.x environment. Thus, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC-Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 11-1.

FIGURE 11-1: PICMASTER SYSTEM CONFIGURATION

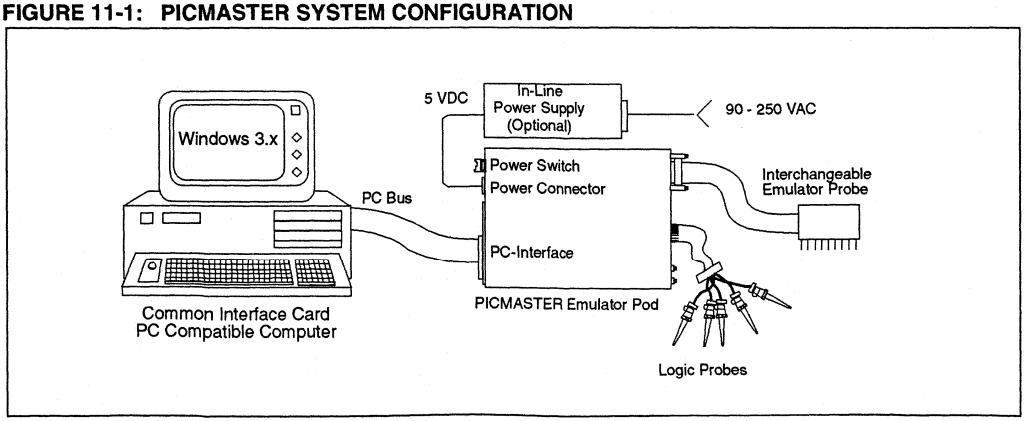


TABLE 11-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

11.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

11.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

11.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

11.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C63, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

© 1995 Microchip Technology Inc.

11.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically, i.e., by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and subtitles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

11.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

11.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

11.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP Edition, for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

11.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 11-2.

TABLE 11-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

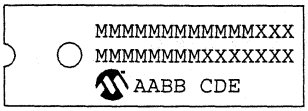
12.0 PACKAGING INFORMATION

For Package Dimensions,
please refer to the Packaging section of the Data Book

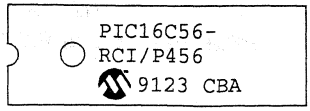
© 1995 Microchip Technology Inc.

12.1 Package Marking Information

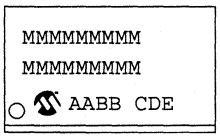
18-Lead PDIP



Example

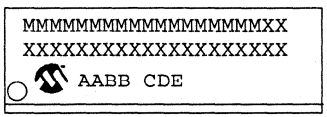


18-Lead SOIC

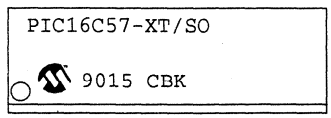


Example

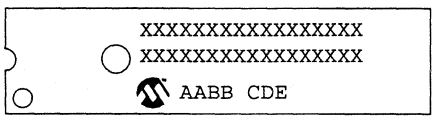
28-Lead SOIC



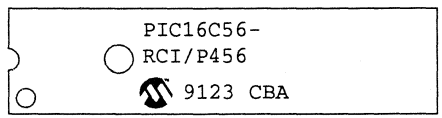
Example



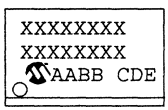
28-Lead PDIP (.300")



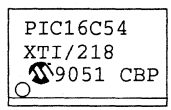
Example



20-Lead SSOP



Example

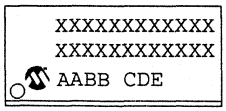


Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

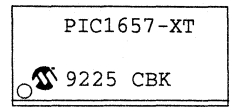
* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

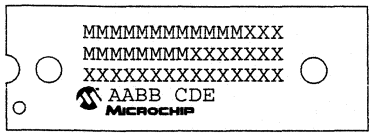
28-Lead SSOP



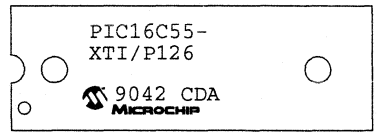
Example



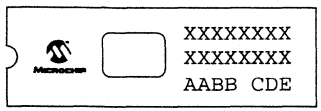
28-Lead PDIP (.600")



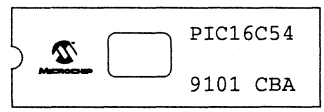
Example



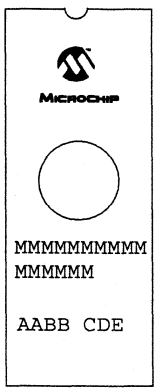
18-Lead CERDIP Windowed



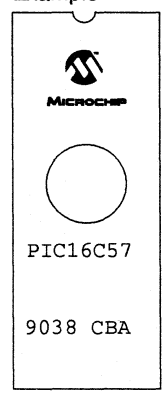
Example



28 Lead, CERDIP Windowed



Example



Legend:		
	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

APPENDIX A: COMPATIBILITY

To convert code written for PIC16CXX to PIC16C5X, the user should take the following steps:

1. Check any CALL, GOTO or instructions that modify the PC to determine if any program memory page select operations (PA2, PA1, PA0 bits) need to be made.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any special function register page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to proper value for processor used.
6. Remove any use of the ADDLW and SUBLW instructions.
7. Rewrite any code segments that use interrupts.

APPENDIX B: WHAT'S NEW

The format of this data sheet has been changed to be consistent with other product families. This ensures that important topics are covered across all PIC16/17 families. Here is an overview list of new features:

- Data Sheet Structure / Outline

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This is so that control bits that do the same function have the same name (regardless of processor family). Care must still be taken, since they may not be in the same special function register. The following shows the register and bit names that have been changed:

REGISTER NAME CHANGES

OLD NAME	NEW NAME
RTCC	TMR0

BIT NAME CHANGES

OLD NAME	NEW NAME
RTS	T0CS
RTE	T0SE

PIN NAME CHANGES

OLD NAME	NEW NAME
RTCC	T0CKI

© 1995 Microchip Technology Inc.

APPENDIX D: PIC16/17 MICROCONTROLLERS

TABLE D-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory			Peripherals					Features				
	Maximum Frequency of Operation (MHz)	EPROM	RAM Data Memory (bytes)	Program Memory	Timer Module(s)	Captures PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages	
PIC17C42	25	2K	232		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP	
PIC17C43*	25	4K	454		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP	
PIC17C44	25	8K	454		TMR0,TMR1, TMR2,TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP	

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller modes.

3: PORTB has software-configurable weak pull-ups.

TABLE D-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals								Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	ROM	Data Memory (bytes)	Program Memory (bytes)	Data EEPROM (bytes)	Timer Modules(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	—	4 ch	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE D-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	EPROM (words)	ROM (words)	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

D.1 Pin Compatibility

Devices that have the same package type; and VDD, VSS, and $\overline{MCLR}$ pin locations, are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE D-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

INDEX

© 1995 Microchip Technology Inc.

Z

LIST OF EXAMPLES

LIST OF FIGURES

LIST OF TABLES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except toll charge to CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks of Microchip Technology Incorporated.

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16C5X Literature Number: DS30015L

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject? _____

5. What deletions from the data sheet could be made without affecting the overall usefullness? _____

6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

© 1995 Microchip Technology Inc.

PIC16C5X Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)		
Package:	P	=	PDIP
	SO	=	SOIC (Gull Wing, 300 mil body)
	SP	=	PDIP (28L, 300 mil body)
	SS	=	SSOP (209 mill body)
	JW*	=	Windowed CERDIP
	S	=	Die in Waffle Pack
Temperature Range:	-	=	0°C to +70°C (T for tape/reel)
	I	=	–40°C to +85°C (S for tape/reel)
	E	=	–40°C to +125°C
Oscillator Type:	RC, HS		
	XT, LP		
Device:	PIC16C54		
	PIC16C55		
	PIC16C56		
	PIC16C57		

Examples:

a) PIC16C54 - XT/PXXX = "XT" oscillator, commercial temp., PDIP, QTP pattern

b) PIC16C55 - XTI/SO = "XT" oscillator, industrial temp., SOIC (OTP device)

c) PIC16C55 - JW = Commercial temp. CERDIP with WINDOW

d) PIC16C57 - RCI/S = "RC oscillator", industrial temp., dice in waffle pack

Note: * For UV erasable devices, the oscillator type is RC by default (=erased device). The user can select XT, HS or LP oscillators by programming the appropriate fuses.

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office (see below)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

ENHANCED PIC16C5X

EPROM/ROM-Based 8-Bit CMOS Microcontroller Series

Devices Included in this Data Sheet

- PIC16C54A
- PIC16C58A
- PIC16CR58A
- PIC16CR57A

High-Performance RISC-like CPU

- Only 33 single word instructions to learn
- All instructions are single cycle (200 ns) except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle

Device	Pins	I/O	EPROM/ ROM	RAM
PIC16C54A	18	12	512	25
PIC16C58A	18	12	2K	73
PIC16CR58A	18	12	2K	73
PIC16CR57A	28	20	2K	72

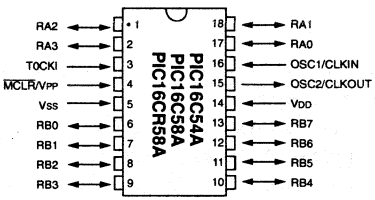
- 12-bit wide instructions
- 8-bit wide data path
- Seven or eight special function hardware registers
- Two-level deep hardware stack
- Direct, indirect and relative addressing modes for data and instructions

Peripheral Features

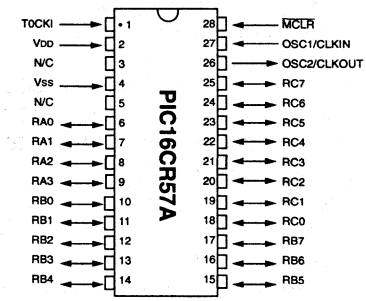
- 8-bit real time clock/counter (TMR0) with 8-bit programmable prescaler
- Power-On Reset (POR)
- Device Reset Timer
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- EPROM/ROM selectable oscillator options:
  - RC: Low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High-speed crystal/resonator
  - LP: Power saving, low frequency crystal

Package Types

PDIP, SOIC, Windowed CERDIP



PDIP, SOIC, CERDIP



CMOS Technology

- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide-operating voltage range:
  - EPROM Commercial/Industrial 2.5V to 6.25V
  - ROM Commercial/Industrial: 2.0V to 6.25V
- Low-power consumption
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 3V, 32 kHz
  - < 0.3 µA typical standby current (with WDT disabled) @ 3V, 0°C to 70°C

© 1995 Microchip Technology Inc.

Package Types

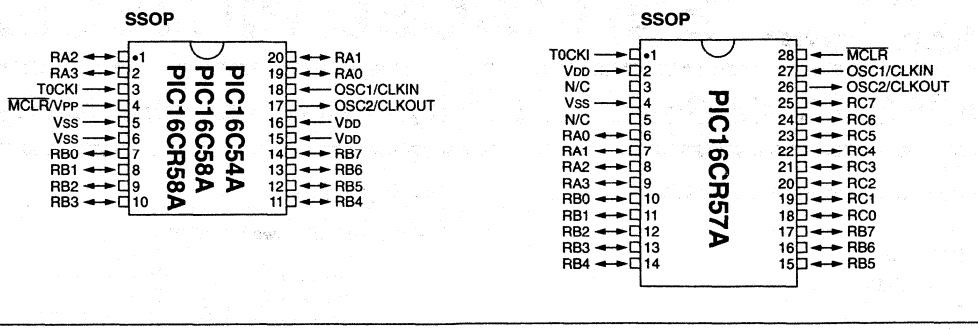


Table of Contents

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error from the previous version of this data sheet (Enhanced PIC16C5X Data Sheet, Literature Number DS30236A), please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

© 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The Enhanced PIC16C5X from Microchip Technology is a family of low-cost, high-performance, 8-bit, fully static, EPROM-based CMOS microcontrollers. This family is pin and software compatible with the PIC16C5X family of devices in a new enhanced process technology. It employs a RISC-like architecture with only 33 single word/single cycle instructions. All instructions are single cycle (200 ns) except for program branches which take two cycles. The Enhanced PIC16C5X delivers performance an order of magnitude higher than its competitors in the same price category. The 12-bit wide instructions are highly symmetrical resulting in a 2:1 code compression over other 8-bit microcontrollers in its class. The easy to use and easy to remember instruction set reduces development time significantly.

The Enhanced PIC16C5X products are equipped with special features that reduce system cost and power requirements. The Power-On Reset (POR) and Device Reset Timer eliminate the need for external reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost-saving RC oscillator. Power saving SLEEP mode, watchdog timer and code protection features improve system cost, power and reliability.

The UV-erasable CERDIP packaged versions are ideal for code development, while the cost-effective One-Time-Programmable (OTP) versions are suitable for production in any volume. The customer can take full advantage of Microchip's price leadership in OTP microcontrollers while benefiting from the OTP's flexibility.

The PIC16C5X products are supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a 'C' compiler, fuzzy logic support tools, a low-cost development programmer, and a full featured programmer. All the tools are supported by IBM® PC and compatible machines.

1.1 Applications

The PIC16C5X series fits perfectly in applications ranging from high-speed automotive and appliance motor control to low-power remote transmitters/receivers, pointing devices and telecom processors. The EPROM technology makes customizing application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make this microcontroller series perfect for applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C5X series very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, replacement of "glue" logic in larger systems, co-processor applications).

© 1995 Microchip Technology Inc.

TABLE 1-1: ENHANCED PIC16C5X FAMILY OF DEVICES

	Clock	Memory			Peripherals			Features	
	Maximum Frequency of Operation (MHz)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

2.0 PIC16C5X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in this section. When placing orders, please use the "PIC16C5X Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in a CERDIP package is optimal for prototype development and pilot programs

The UV erasable version can be erased and reprogrammed to any of the oscillator modes. Microchip's PICSTART™ and PRO MATE™ programmers both support programming of the PIC16C5X.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages permit the user to program them once. In addition to the program memory, the configuration fuses must be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and fuse options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround-Production (SQTP) Devices

Microchip offers the unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, giving the customer a low cost option for high volume, mature products.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C5X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C5X uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12-bits wide making it possible to have all single word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The PIC16C58A/CR58A and PIC16CR57A address 2K x 12 program memory, while the PIC16C54A addresses 512 x 12 of program memory. All program memory is internal.

The PIC16C5X can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory. The PIC16C5X has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16C5X simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16C5X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

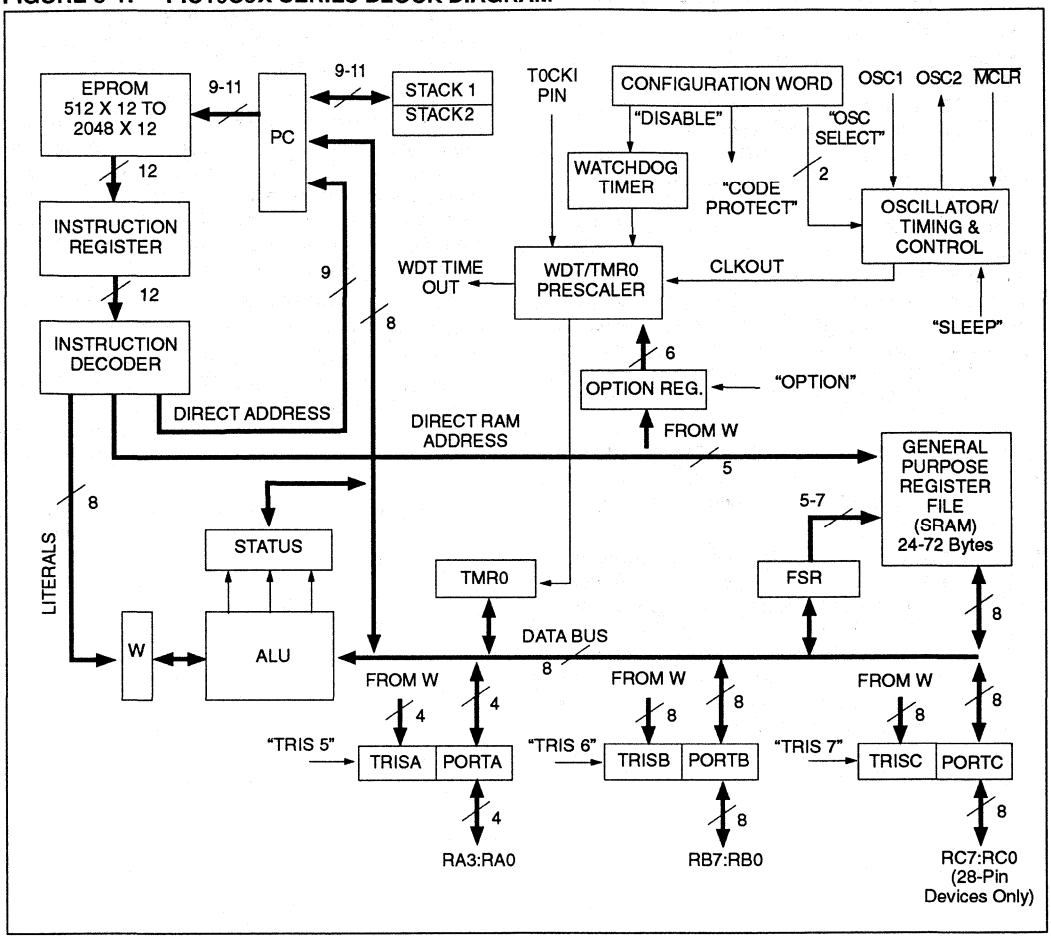
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBWF and ADDWF instructions for examples.

A simplified block diagram is shown in Figure 3-1, with the corresponding device pins described in Table 3-1, and Table 3-2.

© 1995 Microchip Technology Inc.

FIGURE 3-1: PIC16C5X SERIES BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C54A/C58A/CR58A PINOUT DESCRIPTION

Name	DIP, SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
RA0	17	19	I/O	TTL	Bi-directional I/O port
RA1	18	20	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RB0	6	7	I/O	TTL	Bi-directional I/O port
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	
RB5	11	12	I/O	TTL	
RB6	12	13	I/O	TTL	
RB7	13	14	I/O	TTL	
T0CKI	3	3	I	ST	Clock input to TMR0 timer. Must be tied to VSS or VDD, if not in use, to reduce current consumption.
$\overline{MCLR}$/VPP	4	4	I	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device. Voltage on the $\overline{MCLR}$/VPP pin must not exceed VDD to avoid unintended entering of test modes.
OSC1/CLKIN	16	18	I	ST	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	17	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
VDD	14	15,16	P	—	Positive supply for logic and I/O pins.
VSS	5	5,6	P	—	Ground reference for logic and I/O pins.

Legend: I = Input only; O = Output only; I/O = Input/Output; P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input.

TABLE 3-2: PIC16CR57A PINOUT DESCRIPTION

Name	DIP, SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
RA0	6	5	I/O	TTL	Bi-directional I/O port
RA1	7	6	I/O	TTL	
RA2	8	7	I/O	TTL	
RA3	9	8	I/O	TTL	
RB0	10	9	I/O	TTL	Bi-directional I/O port
RB1	11	10	I/O	TTL	
RB2	12	11	I/O	TTL	
RB3	13	12	I/O	TTL	
RB4	14	13	I/O	TTL	
RB5	15	15	I/O	TTL	
RB6	16	16	I/O	TTL	
RB7	17	17	I/O	TTL	
RC0	18	18	I/O	TTL	Bi-directional I/O port
RC1	19	19	I/O	TTL	
RC2	20	20	I/O	TTL	
RC3	21	21	I/O	TTL	
RC4	22	22	I/O	TTL	
RC5	23	23	I/O	TTL	
RC6	24	24	I/O	TTL	
RC7	25	25	I/O	TTL	
T0CKI	1	2	I	ST	Clock input to TMR0 register. Must be tied to VSS or VDD if not in use to reduce current consumption.
$\overline{MCLR}$	28	28	I	ST	Master clear (reset) input. This pin is an active low reset to the device. Voltage on the $\overline{MCLR}$ pin must not exceed VDD to avoid unintended entering of test modes.
OSC1/CLKIN	27	27	I	ST	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	26	26	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
VDD	2	3,4	P	—	Positive supply for logic and I/O pins.
VSS	4	1,14	P	—	Ground reference for logic and I/O pins.
N/C	3,5	—	—	—	Unused, do not connect

Legend: I = Input only; O = Output only; I/O = Input/Output; P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input.

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from the OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter is incremented every Q1, and the instruction is fetched from program memory and latched into instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

3.2 Instruction Flow/Pipelining

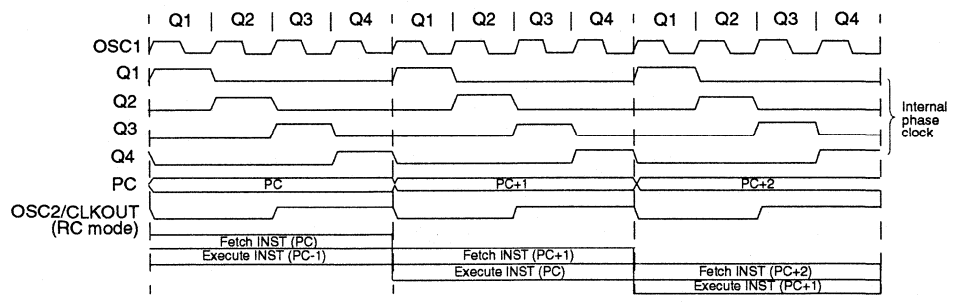
An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

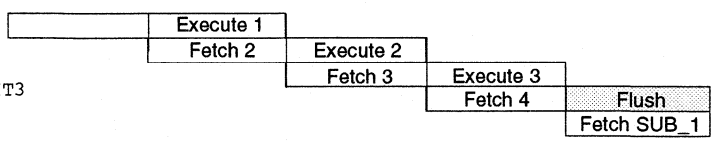
2

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

```
1. MOVLW  55h
2. MOVWF  PORTB
3. CALL   SUB_1
4. BSF    PORTA, BIT3
```



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

Up to 512 words of 12-bit wide on-chip program memory (EPROM/ROM) can be directly addressed. Larger program memories can be addressed by selecting one of up to four available pages of 512 words each (Figure 4-1). Sequencing of instructions is controlled via the Program Counter (PC) which automatically increments to execute in-line programs. Program control operations supporting direct, indirect, and relative addressing modes, can be performed by bit test, skip, call, and jump type instructions, or by loading computed addresses into the PC. In addition, an on-chip two-level stack is employed to provide easy to use subroutine nesting.

4.2 Data Memory Organization

The 8-bit data bus connects two basic functional elements together: the register file composed of up to 80 addressable 8-bit registers including the I/O ports, and an 8-bit wide Arithmetic Logic Unit (ALU). 32 bytes of RAM are directly addressable while a "banking" scheme, with banks of 16 bytes each, is employed to address larger data memories (Figure 4-2). Data can be addressed directly, or indirectly using the File Select Register (FSR). Immediate data addressing is supported by special "literal" instructions which load data from program memory into the W register.

The register file is divided into two functional groups: Special Function registers and General Purpose registers. The special function registers include the Timer0 (TMR0) register, the Program Counter (PC), the Status Register, the I/O registers (ports), and the File Select Register (FSR). The general purpose registers are used for data and control information under command of the instructions.

In addition, special purpose registers are used to control the I/O port configuration and prescaler options.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly or indirectly through the file select register FSR (Section 4.5).

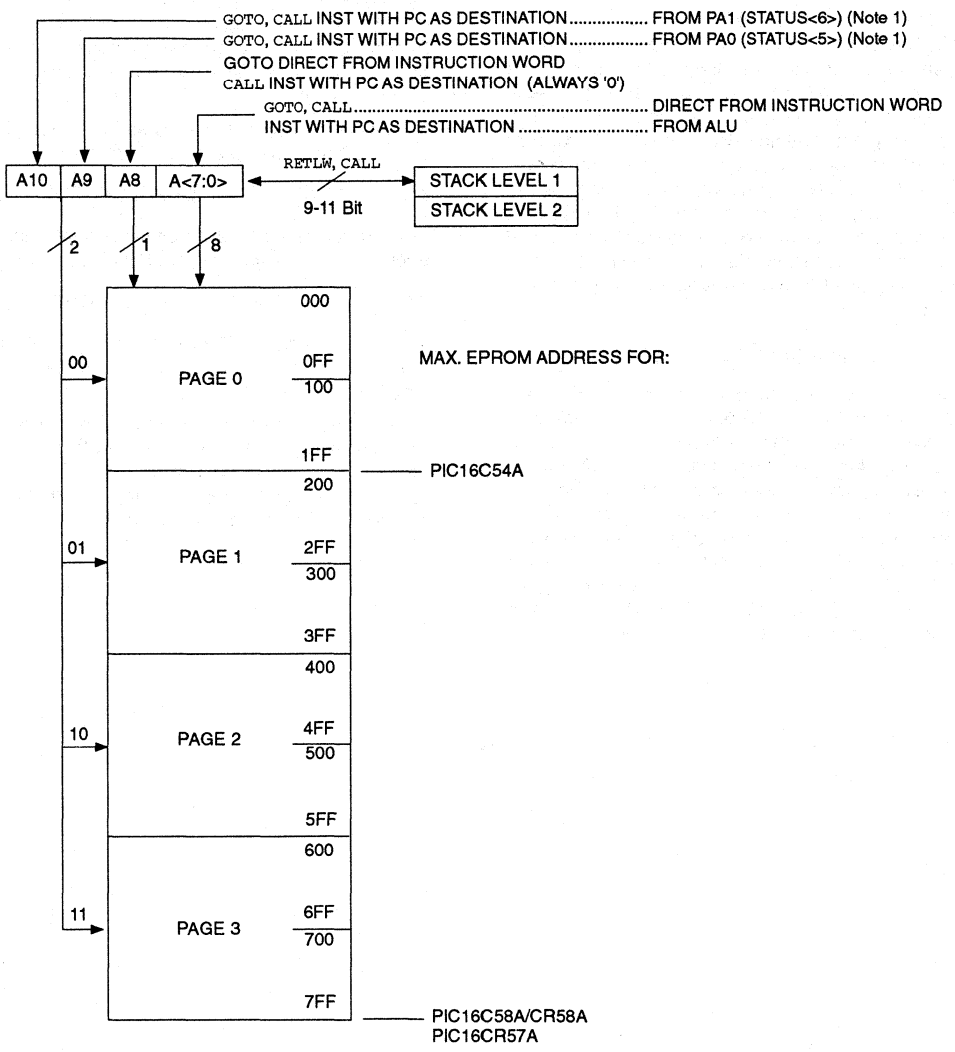
4.2.2 SPECIAL FUNCTION REGISTERS:

The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device (Table 4-1).

The special registers can be classified into two sets. The special registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

© 1995 Microchip Technology Inc.

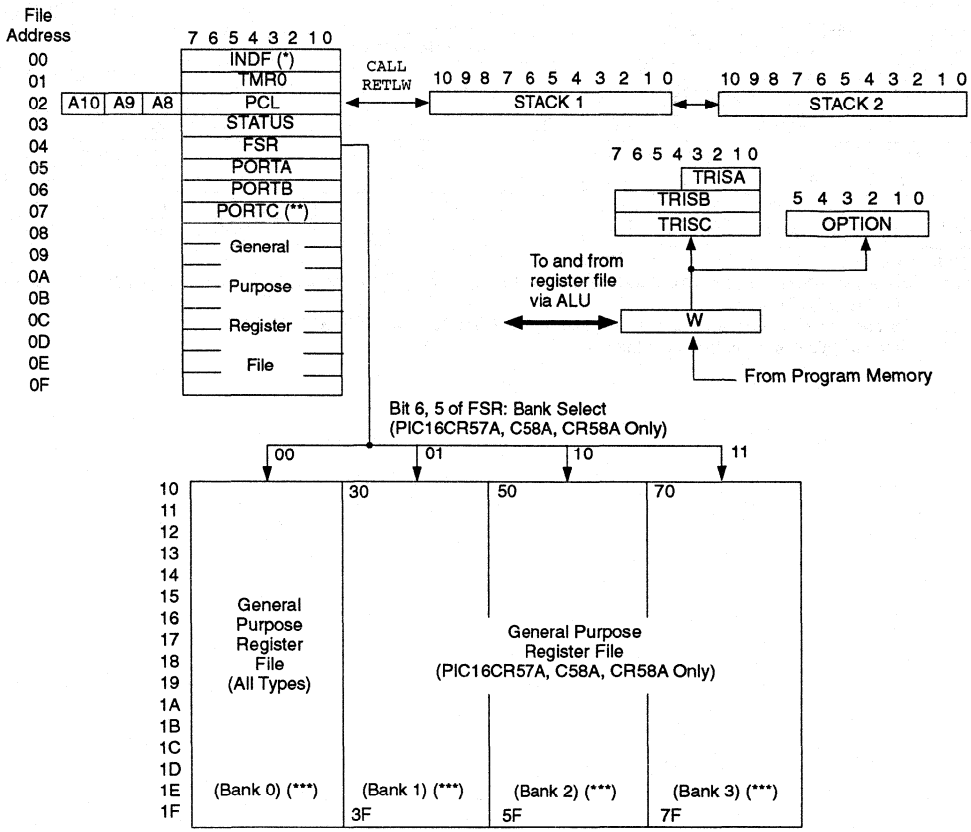
FIGURE 4-1: PROGRAM MEMORY ORGANIZATION



Note 1: PIC16CR57A and PIC16C58A/CR58A only.

© 1995 Microchip Technology Inc.

FIGURE 4-2: DATA MEMORY MAP



(*) Not a physically implemented register. See Section 4.5 for details.

(**) File address 7h is a general purpose register on the PIC16C54A, and PIC16C58A/CR58A.

(***) Bank 0 is available on all microcontrollers while Bank 1, Bank 2, and Bank 3 are only available on the PIC16CR57A, C58A, CR58A. (Section 4.5).

TABLE 4-1: PIC16C5X REGISTER FILE SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on $\overline{MCLR}$ and WDT resets
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of PC								1111 1111	1111 1111
03h	STATUS	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Indrect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	PORTC [2]	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged. - = unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. The upper bits can be set or cleared by writing to PA1:PA0 (STATUS<6:5>).

2: File address 7h is a general purpose register on the PIC16C54A and PIC16C58A/CR58A.

3: Shading indicates unimplemented bits.

© 1995 Microchip Technology Inc.

4.3 STATUS Register

This register contains the arithmetic status of the ALU, the RESET status, and the page preselect bits for program memories larger than 512 words.

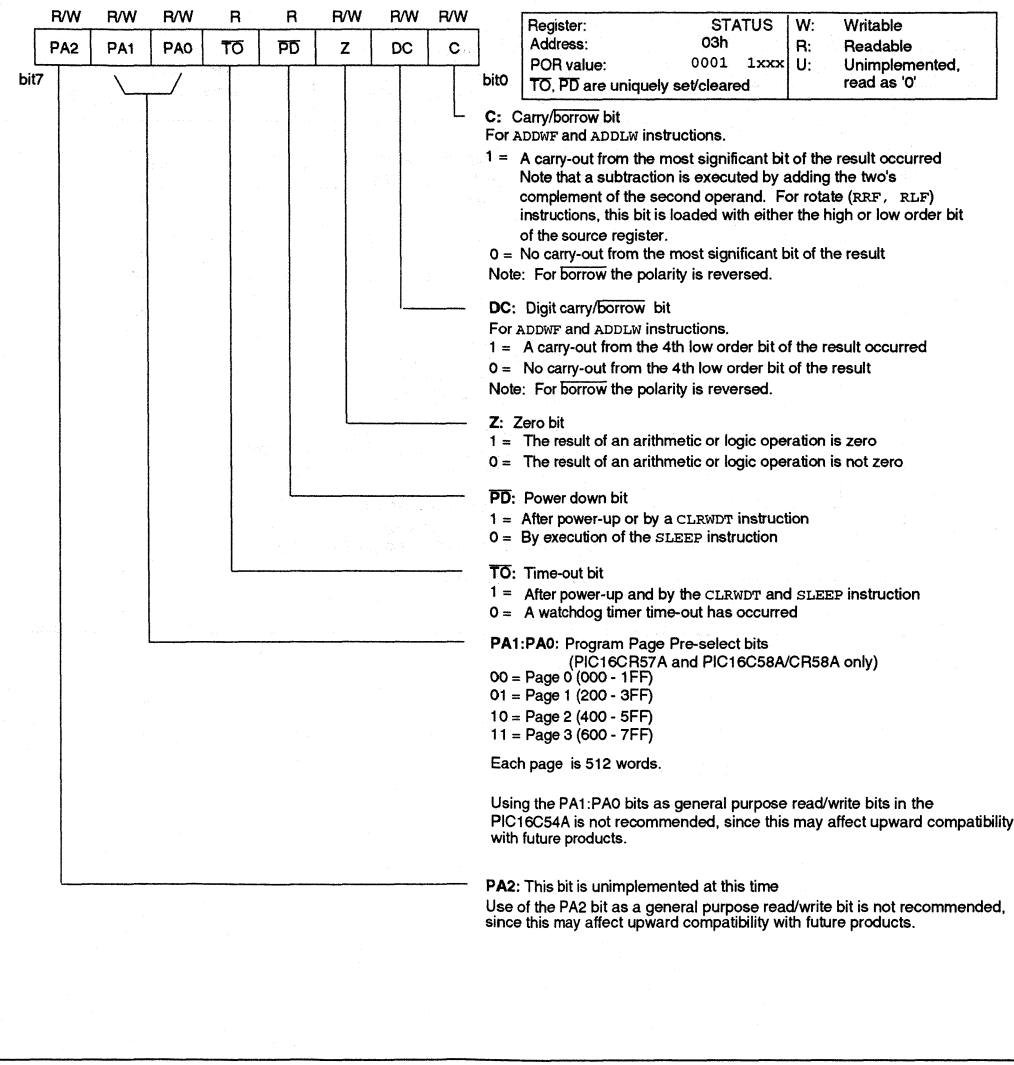
As with any other register, the STATUS register can be the destination for any instruction. However, the STATUS bits are set after the following write. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear all bits, except for $\overline{TO}$ and $\overline{PD}$, and then set the Z bit which leaves the STATUS register as `000u u100` (where `u` = unchanged).

Therefore, only `BCF`, `BSF` and `MOVWF` instructions should be used to alter the STATUS registers because these instructions do not affect any STATUS bit.

For the instructions that affect STATUS bits, see the "Instruction Set Summary" (Table 8-2).

FIGURE 4-3: STATUS REGISTER

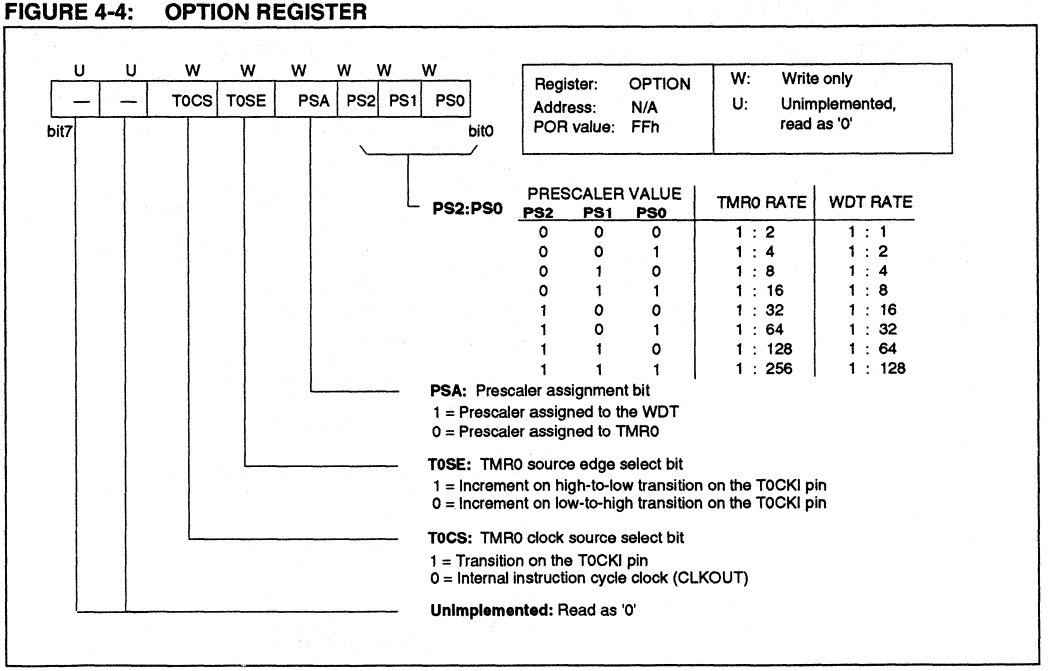


4.4 OPTION Register

The OPTION register is a 6-bit wide, write only register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, and TMR0.

By executing the `OPTION` instruction, the contents of the W register will be transferred to the OPTION register. A RESET sets the OPTION register to all '1's.

FIGURE 4-4: OPTION REGISTER



© 1995 Microchip Technology Inc.

4.5 Indirect Data Addressing, INDF and FSR Registers

The INDF register is not a physical register and is used in conjunction with the FSR register to perform indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself (i.e. FSR = 0) indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although STATUS bits may be affected).

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
        movlw   0x10    ; Initialize pointer
        movwf   FSR     ; to RAM
Next    clrf    INDF    ; Clear loc
        incf    FSR     ; increment pntr
        btfsc   FSR,4   ; All done?
        goto    Next    ; No, clear next
                        ; Location
        .
        .
        .
```

4.5.1 FILE SELECT REGISTER (FSR)

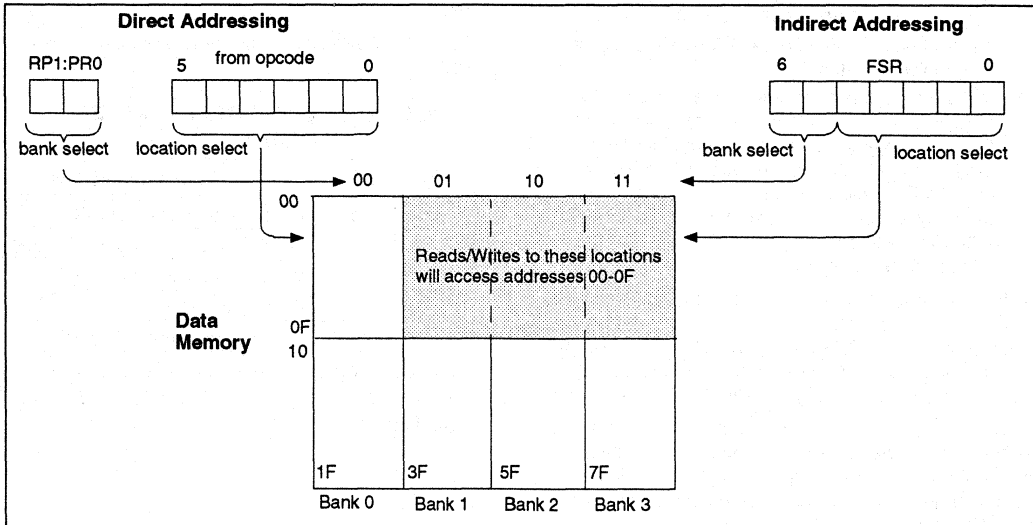
The FSR is either a 5-bit (PIC16C54A) or 7-bit (PIC16C57A/C58A/CR58A) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area. The FSR<4:0> bits are the pointer for data memory addresses 00h to 1Fh. FSR<4> toggles between the 16 lower (00h-0Fh) and 16 upper (10h-1Fh) register files. When clear, FSR<4> points to the lower 16 register files and, when set, points to the upper 16 files. FSR<3:0> provide the value to address the specific register file within each 16 file area.

When not performing indirect addressing, the FSR can be used as a 5-bit (FSR<4:0>) wide general purpose register. However, this is not recommended to help ensure future upward code compatibility.

PIC16C54A: Does not use banking. FSR<7:5> are unimplemented and read as '1's.

PIC16C57A/C58A/CR58A: FSR<6:5> are the bank select bits and are used to select the bank to be addressed (00 = bank 0, 01 = bank 1, 10 = bank 2, 11 = bank 3). The lower 16 register files for banks 1, 2 and 3 are mapped to bank 0, and are not accessible. In other words, the FSR<6:5> bits are ignored if FSR<4> is clear (= 0). FSR<7> is unimplemented and is always read as '1'.

FIGURE 4-5: DIRECT/INDIRECT ADDRESSING



Note 1: For register memory map detail Figure 4-2.
2: RP1 and RP0 are FSR<6> and FSR<5>, respectively.

4.6 Program Counter

The program counter generates the addresses for up to 2048 x 12 on-chip EPROM/ROM cells containing the program instruction words (Figure 4-1).

Depending on the device type, the program counter and its associated two-level hardware stack is 9 or 11-bits wide.

TABLE 4-2: PROGRAM COUNTER STACK WIDTH

Part #	PC width	Stack width
PIC16C54A	9-bit	9-bit
PIC16CR57A	11-bit	11-bit
PIC16C58A/CR58A	11-bit	11-bit

The program counter is set to all '1's upon RESET. During program execution it is auto incremented with each instruction unless the result of that instruction changes the PC itself.

a) GOTO instructions allow the direct loading of the lower nine program counter bits (PC<8:0>). In the case where the program memory is greater than 512 bytes, the upper two bits of PC (PC<10:9>) are loaded with page select bits PA1:PA0 (STATUS<6:5>). Thus, GOTO allows jumps to any location on any page.

b) CALL instructions load the lower 8-bits of the PC directly, while the ninth bit is cleared to '0'. The PC value, incremented by one, will be pushed into the stack. In cases where the program memory is greater than 512 bytes, the upper 2-bits of PC (PC<10:9>) are loaded with page select bits PA1:PA0 (STATUS<6:5>).

c) RETLW instructions load the PC with the Top of Stack (TOS) contents.

d) If the PC is the destination in any instruction (e.g. MOVWF PC, ADDWF PC, or BSF PC,5) then the computed 8-bit result will be loaded into the lower 8-bits of program counter. The ninth bit of PC will be cleared. In case where the program memory is greater than 512 bytes, PC<10:9> will be loaded with Page Select bits PA1, PA0 (STATUS<6:5>).

It should be noted that because bit8 (ninth bit) of the PC is cleared in the CALL instruction or any instruction which writes to the PC (e.g. MOVWF PC). All subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

Incrementing the PC when it is pointing to the last address of a selected memory page is also possible and will cause the program to continue in the next higher page. However, the page preselect bits in the STATUS register will not be changed, and the next GOTO, CALL, ADDWF PC, or MOVWF PC instruction will return to the previous page, unless the page preselect bits have been updated under program control. For example, an NOP at location 1FFh (page 0) increments the PC to 200h (page 1). A GOTO xxx at 200h will return the program to address "xxx" on page 0 (assuming that PA1 and PA0 are clear).

Upon RESET, page 0 is pre-selected while the PC addresses the last location in the last page. Thus, a GOTO instruction at this location will automatically cause the program to continue in page 0.

4.7 W (Working) Register

The W register holds the second operand in two operand instructions and/or supports the internal data transfer.

TABLE 4-3: EVENTS AFFECTING $\overline{TO}/\overline{PD}$ STATUS BITS

Event	$\overline{TO}$	$\overline{PD}$	Remarks
Power-up	1	1	
WDT Timeout	0	x	No effect on $\overline{PD}$
SLEEP instruction	1	0	
CLRWDT instruction	1	1	

A WDT timeout will occur regardless of the status of the $\overline{TO}$ bit. A SLEEP instruction will be executed, regardless of the status of the $\overline{PD}$ bit. Table 4-4 reflects the status of $\overline{TO}$ and $\overline{PD}$ after the corresponding event.

4.7.1 TIME-OUT AND POWER-DOWN STATUS BITS ($\overline{TO}$, $\overline{PD}$)

The $\overline{TO}$ and $\overline{PD}$ (STATUS<4:3>) can be tested to determine if a RESET condition has been caused by a Watchdog Timer time-out, a power-up condition, or a wake-up from SLEEP by the Watchdog Timer or $\overline{MCLR}$ /VPP pin.

These STATUS bits are only affected by events listed in Table 4-3.

TABLE 4-4: $\overline{TO}/\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	= Low pulse on $\overline{MCLR}$ input

The $\overline{TO}$ and $\overline{PD}$ bits maintain their status (u) until an event of Table 4-3 occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

© 1995 Microchip Technology Inc.

5.0 I/O PORTS

As with any other register the I/O registers can be written and read under program control. However, read instructions (e.g. `MOVF PORTB,W`) always read the I/O pins independent of the pin's input/output mode. On a RESET, all I/O ports are defined as input (outputs are at hi-impedance) as the I/O control registers (TRISA, TRISB, TRISC) are all set.

5.1 PORTA

PORTA is a 4-bit I/O register. Only the low order 4-bits are used (RA3:RA0). Bits 7:4 are unimplemented and read as '0's.

5.2 PORTB

PORTB is an 8-bit I/O register (PORTB<7:0>).

5.3 PORTC

PIC16CR57A: 8-bit I/O register.

PIC16C54A/C58A/CR58A: General purpose register.

5.4 TRIS Registers

The output driver control registers are loaded with the content of the W register by executing the `TRIS f` instruction. A '1' in any TRIS register bit puts the corresponding output driver in a hi-impedance mode. A '0' puts the contents of the output data latch on the selected pins.

Note: A read of the ports reads the pins, not the output data latches, i.e., if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

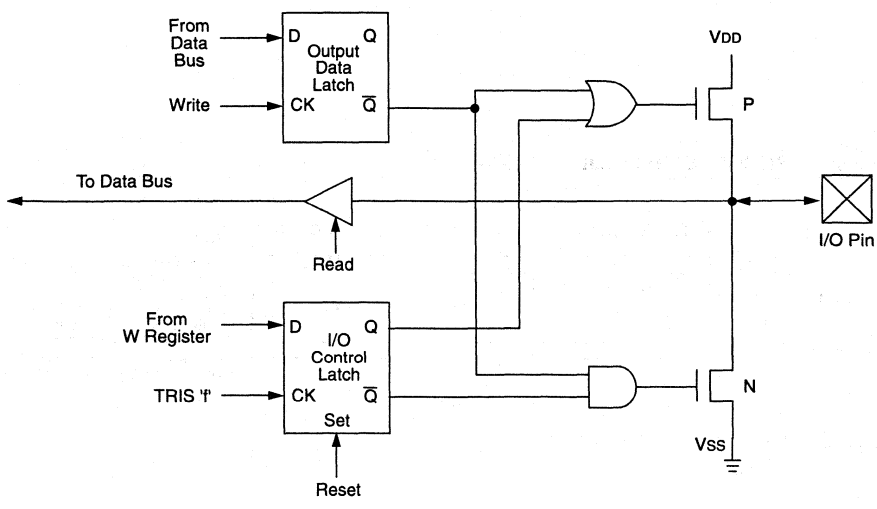
The TRIS registers are "write-only" and are set to all '1's (output drivers disabled) upon RESET.

5.5 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-1. All ports may be used for both input and output operations. For input operations these ports are non-latching. Any input must be present until read by an input instruction (e.g. `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit (in TRISA, TRISB, TRISC) must be cleared (= '0'). For use as an input, the corresponding TRIS bit must be set (= '1'). Any I/O pin can be programmed individually as input or output.

2

FIGURE 5-1: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



Note: I/O pins have protection diodes to VDD and VSS.

5.6 I/O Programming Considerations

5.6.1 BI-DIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and writes the result to the location specified. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and the PORTB value is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (say bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the contents of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF) on an I/O port.

A pin actively outputting a '0' or '1' should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

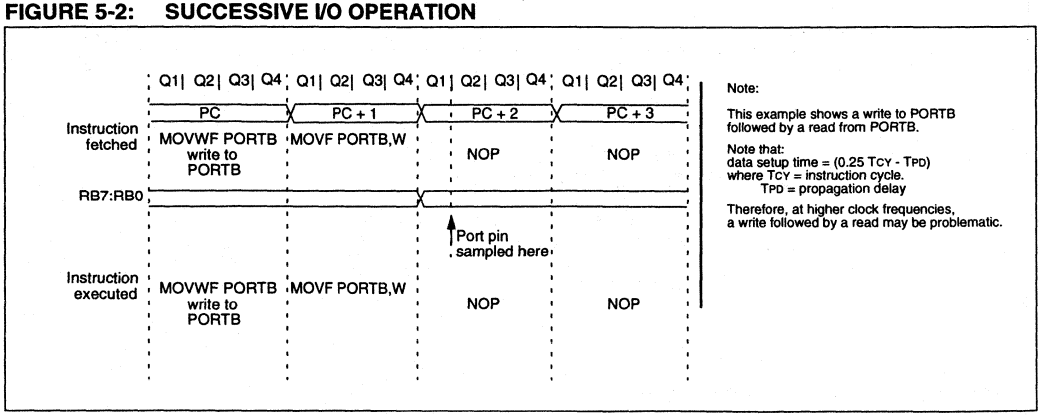
EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
; Initial PORT Settings
;   PORTB<7:4> Inputs
;   PORTB<3:0> Outputs
; PORTB<7:6> have external pull-ups and are not
; connected to other circuitry
;
;                        PORT latch   PORT pins
;                        ----------   ----------

    BCF   PORTB, 7     ; 01pp pppp    11pp pppp
    BCF   PORTB, 6     ; 10pp pppp    11pp pppp
    BSF   STATUS,RP0   ;
    MOVLW 03Fh         ;
    TRIS  PORTB        ; 10pp pppp    10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (see Figure 5-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with an NOP or another instruction not accessing this I/O port.

FIGURE 5-2: SUCCESSIVE I/O OPERATION



© 1995 Microchip Technology Inc.

6.0 TIMER0 (TMR0) MODULE

The TMR0 module has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the TMR0 module, while Figure 6-2 shows the electrical structure of the TMR0 input.

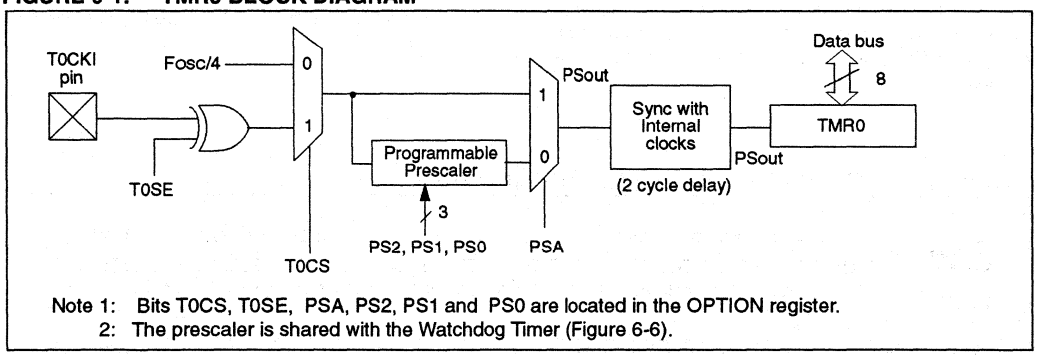
Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, increment is inhibited for the following two cycles (Figure 6-3 and Figure 6-4). The user can work around this by writing an adjusted value to the TMR0 module.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin T0CKI. The incrementing edge is determined by the T0 source edge select bit T0SE (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.1.

The prescaler is shared between the TMR0 module and the WDT. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Setting the PSA bit will assign the prescaler to the WDT and cause the prescale for TMR0 to be 1:1. Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is neither readable nor writable. When the prescaler is assigned to the TMR0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 6.2 details the operation of the prescaler.

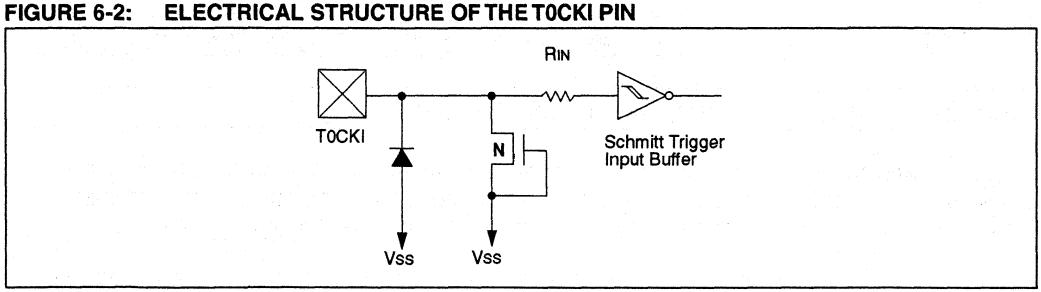
2

FIGURE 6-1: TMR0 BLOCK DIAGRAM



Note 1: Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.
2: The prescaler is shared with the Watchdog Timer (Figure 6-6).

FIGURE 6-2: ELECTRICAL STRUCTURE OF THE T0CKI PIN



© 1995 Microchip Technology Inc.

6.1 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements to be able to synchronize with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of TMR0 after synchronization.

6.1.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. Synchronizing T0CKI with the internal phase clocks requires sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (see Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (plus a small RC delay) and low for at least 2 TOSC (plus a small RC delay). Refer to the appropriate electrical specification table.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler to ensure that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 TOSC (plus a small RC delay) divided by the prescaler value. The only limitation on T0CKI high and low time is that they are greater than the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the appropriate electrical specification section.

6.1.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

6.2 Prescaler

An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer (Section 6.2.1). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g. `CLRF 1`, `MOVWF 1`, `BSF 1,x` ....etc.) will clear the prescaler. When assigned to the WDT, a `CLRWDT` instruction will clear the prescaler along with the WDT. The prescaler is neither readable or writable. On a reset, the prescaler contains all '0's.

FIGURE 6-3: TMR0 TIMING: INTERNAL CLOCK/NO PRESCALE

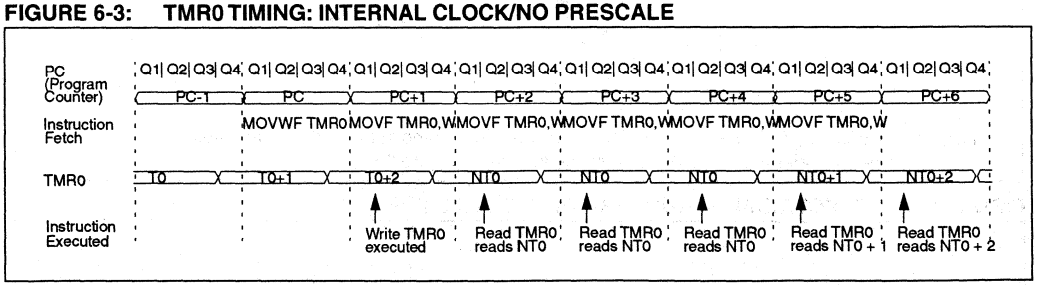
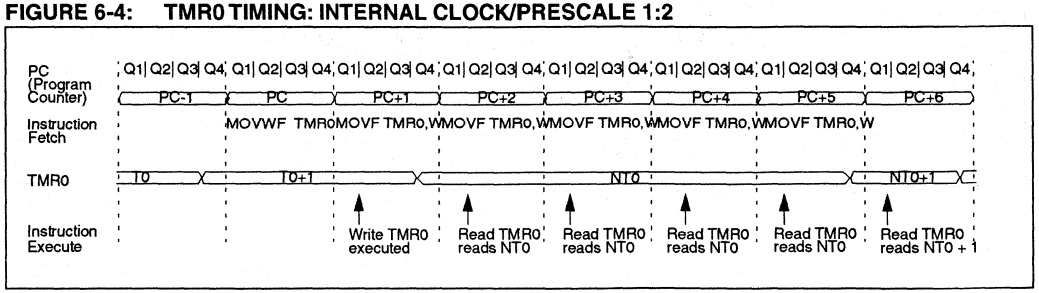
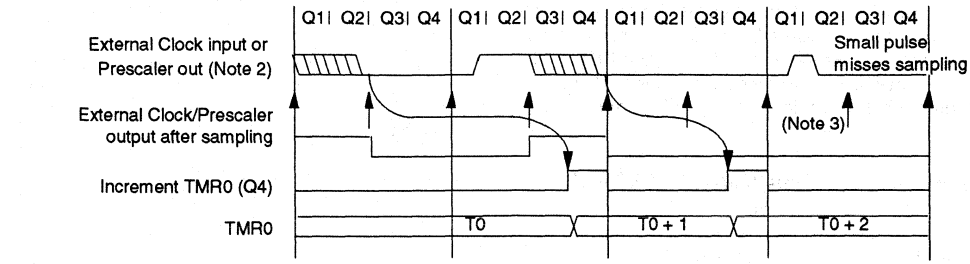


FIGURE 6-4: TMR0 TIMING: INTERNAL CLOCK/PRESCALE 1:2



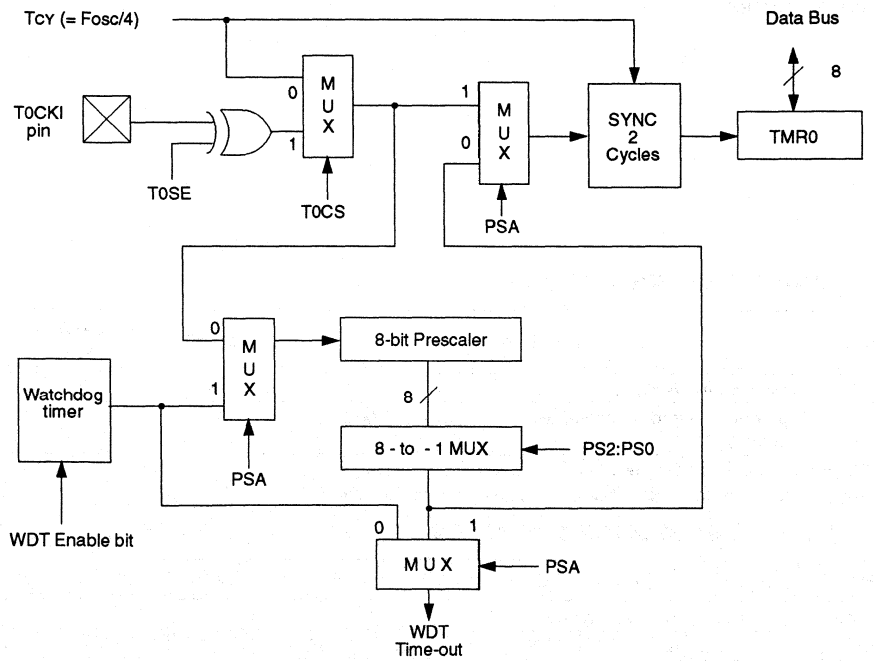
© 1995 Microchip Technology Inc.

FIGURE 6-5: TMR0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to TMR0 increment is 3 Tosc to 7 Tosc. (Duration of Q = Tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 Tosc max.
2: External clock if no prescaler selected, Prescaler output otherwise.
3: The arrows ↑ indicate the points in time where sampling occurs.

FIGURE 6-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0CS, T0SE, PSA, PS2:PS0 are bits OPTION<6:0>.

6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler from TMR0 to WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TMR0→WDT)

```
CLRF    TMR0            ;Clear TMR0
CLRWDT                  ;Clears WDT and
                        ;prescaler
MOVLW   'xxxx1xxx'b     ;Select new prescale
OPTION                  ;value
```

To change the prescaler from the WDT to the TMR0 module use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled.

Note that a `CLRWDT` instruction should be executed before switching the prescaler.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                  ;Clear WDT and
                        ;prescaler
MOVLW     'xxxx0xxx'b   ;Select TMR0, new
                        ;prescale value and
                        ;clock source
OPTION
```

TABLE 6-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-On Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. (Figure 4-4)	N/A	--11 1111

Legend: x = unknown, - = unimplemented, reads a '0'.
Note 1: For reset values of registers in other reset situations refer to Table 4-1.

TABLE 6-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01	TMR0	Timer0, 8-bit real time clock counter							
N/A	OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: — = Unimplemented locations, Read as '0'.
Note 1: Shaded cells are not used by the TMR0 module.

© 1995 Microchip Technology Inc.

7.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC16C5X family of microcontrollers has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- OSC selection
- Reset
- Power-On Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations

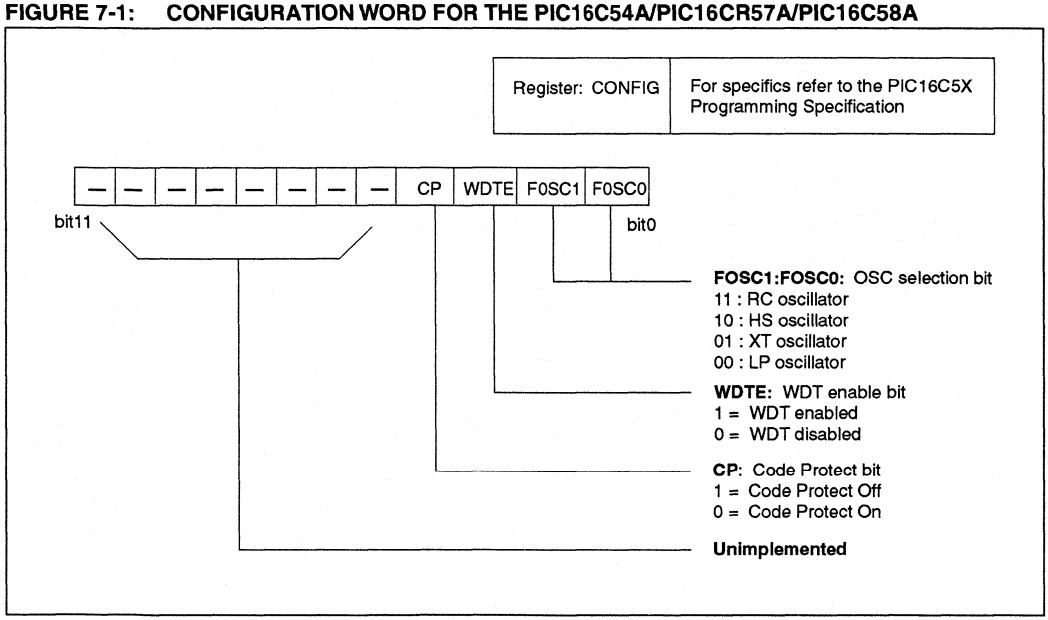
The PIC16C5X has a watchdog timer which can be shut off only through configuration bit WDTE. It runs off of its own RC oscillator for added reliability. There is an 18 ms Device Reset Timer (DRT), intended to keep the chip in reset until the crystal oscillator is stable. With the timer on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external resets or through a WDT time-out. Several oscillator options are also available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits (FOSC1:FOSC0) are used to select various options.

7.1 Configuration Bits

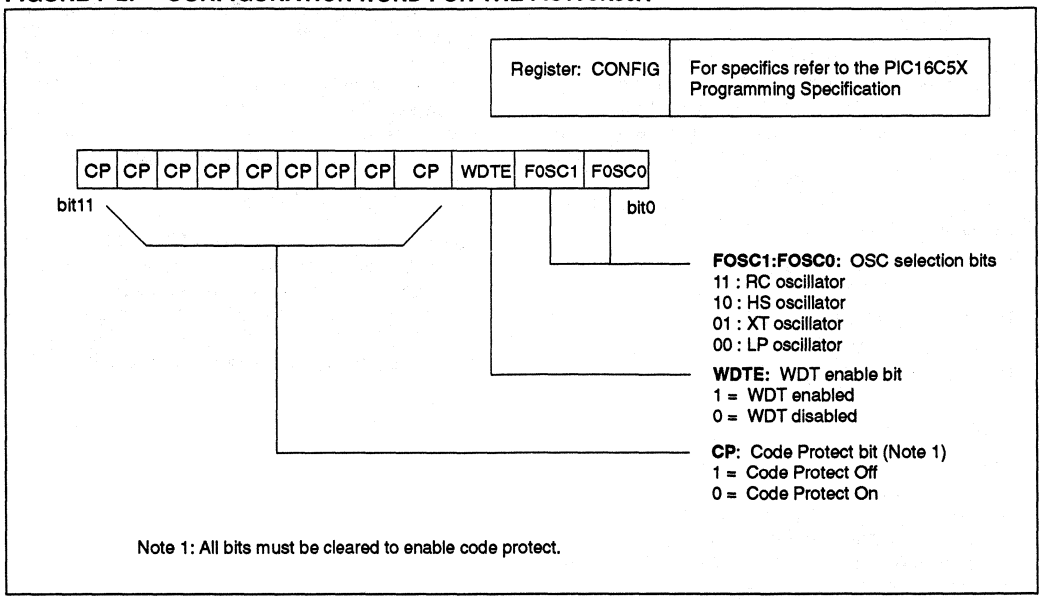
The configuration word consists of 4 or 12 bits depending on the device configuration. Configuration bits can be programmed to select various device configurations. Two provide for the selection of the oscillator type (FOSC1:FOSC0), one is the Watchdog Timer enable bit (WDTE), and one is the code protection bit (CP).

FIGURE 7-1: CONFIGURATION WORD FOR THE PIC16C54A/PIC16CR57A/PIC16C58A



© 1995 Microchip Technology Inc.

FIGURE 7-2: CONFIGURATION WORD FOR THE PIC16CR58A



Note 1: All bits must be cleared to enable code protect.

© 1995 Microchip Technology Inc.

7.2 Oscillator Configurations

7.2.1 OSCILLATOR TYPES

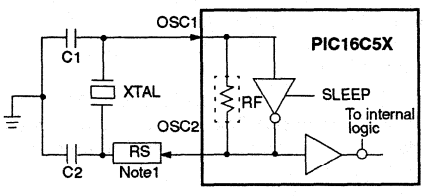
The PIC16C5X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

- LP: Low Power Crystal
- XT: Crystal/Resonator
- HS: High Speed Crystal/Resonator
- RC: Resistor/Capacitor

7.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 7-3). The PIC16C5X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin. (Figure 7-4).

FIGURE 7-3: CRYSTAL /CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



See Table 7-1 and Table 7-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 7-4: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC MODE)

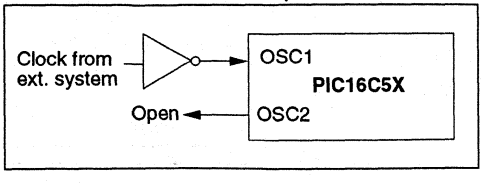


TABLE 7-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	455 kHz	22-100 pF	22-100 pF
	2.0 MHz	15-68 pF	15-68 pF
	4.0 MHz	15-68 pF	15-68 pF
HS	8.0 MHz	10-68 pF	10-68 pF
	16.0 MHz	10-22 pF	10-22 pF

These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

TABLE 7-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz†	33-68 pF	33-68 pF
	200 kHz	15-47 pF	15-47 pF
XT	100 kHz	47-100 pF	47-100 pF
	2 MHz	15-33 pF	15-33 pF
	4 MHz	10-33 pF	10-33 pF
HS	8 MHz	15-47 pF	15-47 pF
	20 MHz	15-47 pF	15-47 pF

† For V_{DD} > 4.5V, C1 = C2 ≈ 30 pF is recommended.

These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

© 1995 Microchip Technology Inc.

7.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

A prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 7-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 7-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

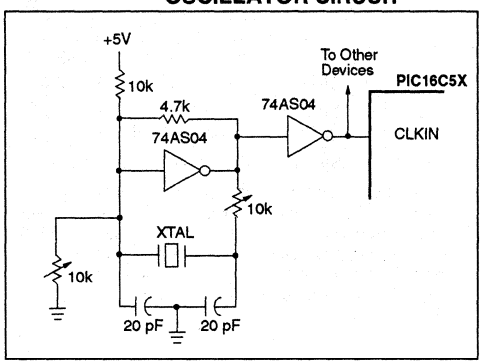
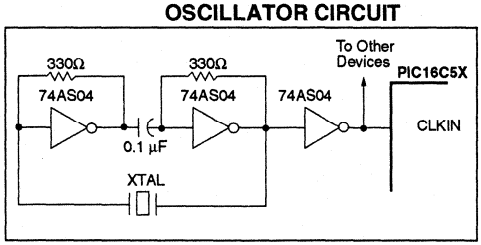


Figure 7-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 7-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



7.2.4 RC OSCILLATOR

For timing insensitive applications the RC device option offers additional cost savings. RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 7-7 shows how the R/C combination is connected to the PIC16C5X. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ) the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping Rext between 3 kΩ and 100 kΩ.

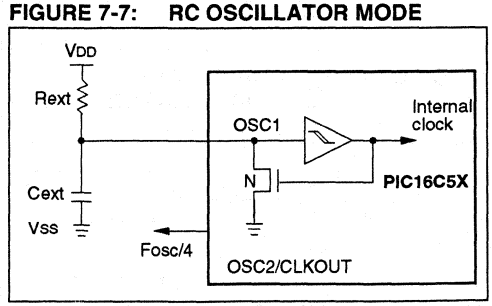
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 9.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 9.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic.

FIGURE 7-7: RC OSCILLATOR MODE



© 1995 Microchip Technology Inc.

7.3 Reset

The PIC16C5X differentiates between various kinds of resets:

- Power-On Reset (POR)
- $\overline{MCLR}$ reset during normal operation
- $\overline{MCLR}$ reset during SLEEP
- WDT time-out reset

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in all other resets. Most other registers are reset to a "reset state" on Power-On Reset (POR), $\overline{MCLR}$ or a WDT reset . Note that the PIC16C5X does not differentiate between a WDT reset during SLEEP or during normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared depending upon the reset situation (Table 7-3). These bits may be used to determine the nature of the reset. See Table 7-5 for a full description of reset states of all registers.

Figure 7-8 shows the simplified block diagram of the on-chip reset circuit.

7.4 Power-On Reset (POR) and Device-Reset Timer (DRT)

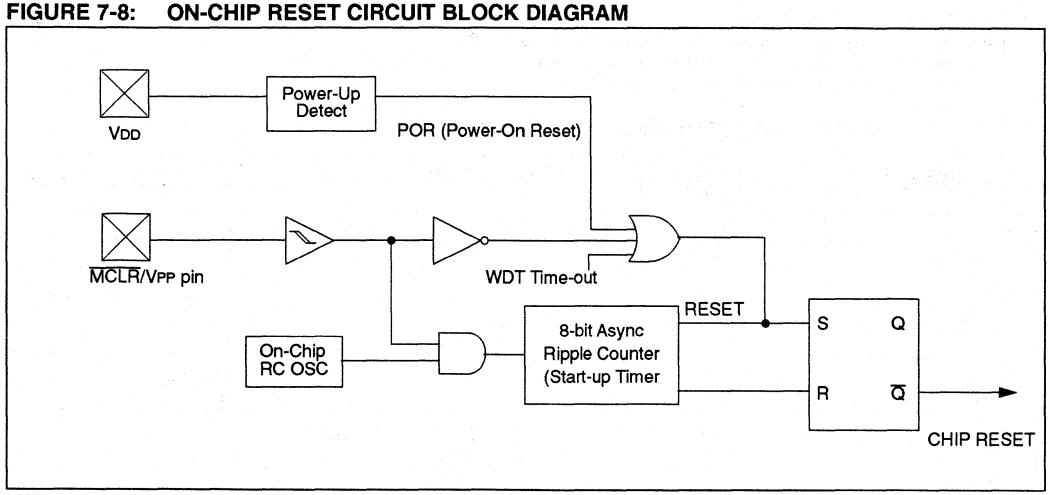
7.4.1 POWER-ON RESET (POR)

The PIC16C5X family incorporates an on-chip Power-On Reset (POR) circuitry which provides an internal chip reset for most power-up situations. To use this feature the user merely needs to tie the $\overline{MCLR}$/VPP pin to VDD. Figure 7-15 shows the electrical structure of TMR0 inputs. The Power-On Reset circuit and the Device Reset Timer circuit are closely related. On power-up the reset latch is set and the DRT is reset. The DRT timer begins counting once it detects $\overline{MCLR}$ to be high. After the time-out period, which is typically 18 ms, it will reset the reset-latch and thus end the on-chip reset signal.

Figure 7-9 and Figure 7-10 are two power-up situations with relatively fast rise time on VDD. In Figure 7-9, VDD is allowed to rise and stabilize before bringing $\overline{MCLR}$ high. The chip will actually come out of reset (TDRT msec) after $\overline{MCLR}$ goes high. In Figure 7-10, the on-chip Power-On Reset feature is being used ($\overline{MCLR}$ and VDD are tied together). VDD is stable before the start-up timer times out and there is no problem in getting a proper reset. Figure 7-11 depicts a potentially problematic situation where VDD rises too slowly. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip is, therefore, not guaranteed to function correctly.

To summarize, the on-chip POR is guaranteed to work if the rate of rise of VDD is no slower than 0.05V/ms, and VDD starts from 0V. The on-chip POR time delay is too short for low frequency crystals which require much longer than 18 ms to start-up and stabilize. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times.

FIGURE 7-8: ON-CHIP RESET CIRCUIT BLOCK DIAGRAM



7.4.2 DEVICE RESET TIMER (DRT)

The Device Reset Timer provides a fixed 18 ms nominal time-out on RESET. The Device Reset Timer operates with an internal RC oscillator. The processor is kept in RESET as long as the DRT is active. The DRT delay allows VDD to rise above VDD min., and allows the oscillator to stabilize.

Oscillator circuits based on crystals or ceramic resonators require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a RESET condition for approximately 18 ms after the voltage on the $\overline{MCLR}$/VPP pin has reached a logic high (VIHMC) level. Thus, external RC networks connected to the $\overline{MCLR}$ input are not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications.

The Device Reset time delay will vary from chip to chip and due to VDD, temperature, and process variation.

The DRT will also be triggered upon a WDT time-out. This is particularly important for applications using the WDT to waken the PIC16C5X from SLEEP automatically.

7.4.3 TIME-OUT SEQUENCE

Table 7-4 lists the reset conditions for the special function registers while Table 7-5 lists the reset conditions for all the registers.

TABLE 7-3: $\overline{TO}$/$\overline{PD}$ STATUS AFTER RESET

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT wake-up from SLEEP
0	1	WDT time-out (not during SLEEP)
1	0	$\overline{MCLR}$ wake-up from SLEEP
1	1	Power-up
u	u	= Low pulse on $\overline{MCLR}$ input

The $\overline{TO}$ and $\overline{PD}$ bits maintain their status (u) until a reset occurs. A low-pulse on the $\overline{MCLR}$ input does not change the $\overline{TO}$ and $\overline{PD}$ status bits.

TABLE 7-4: RESET CONDITIONS FOR SPECIAL REGISTERS

Condition	STATUS Addr: 03h	PCL Addr: 02h
Power-On Reset	0001 1xxx	1111 1111
$\overline{MCLR}$ reset during normal operation	000u uuuu(1)	1111 1111
$\overline{MCLR}$ reset during SLEEP	0001 0uuu	1111 1111
WDT reset during SLEEP	0000 0uuu	1111 1111
WDT reset during normal operation	0000 1uuu	1111 1111

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.
Note 1: The $\overline{TO}$ and $\overline{PD}$ bits retain their last value until one of the other reset conditions occur.
2: The CLRWDT instruction will set the $\overline{TO}$ and $\overline{PD}$ bits

TABLE 7-5: RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	$\overline{MCLR}$ or WDT Reset
W	N/A	xxxx xxxx	uuuu uuuu
TRIS	N/A	1111 1111	1111 1111
OPTION	N/A	--11 1111	--11 1111
INDF	00h	—	—
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000? ?uuu (1)
FSR	04h	xxxx xxxx	uuuu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu
General Purpose register files	08-7Fh	xxxx xxxx	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented read as '0', ? = value depends on condition.
Note 1: See Table 7-4 for reset value for specific conditions.

© 1995 Microchip Technology Inc.

FIGURE 7-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): Case 1

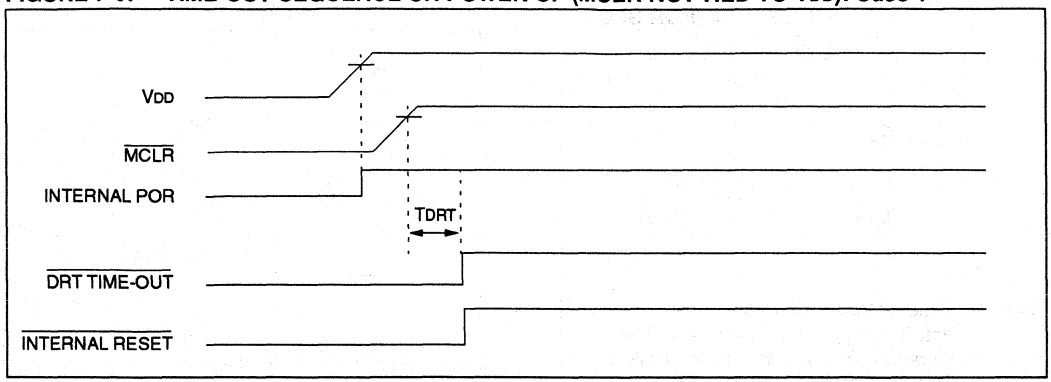


FIGURE 7-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): Case 2

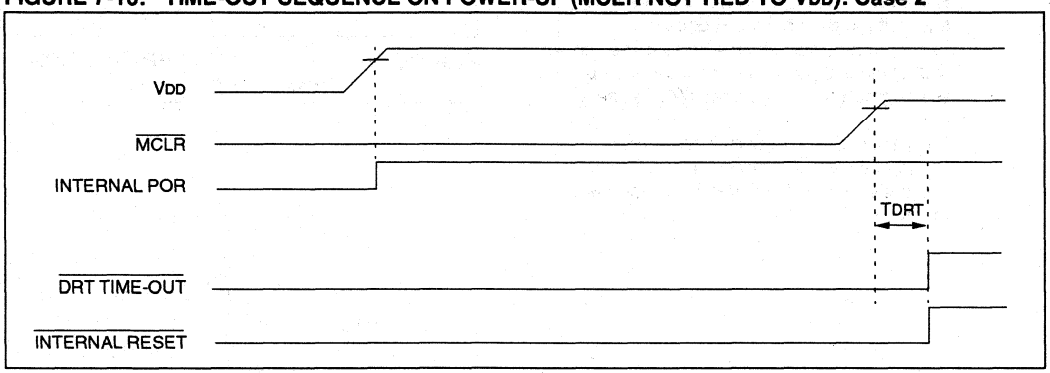
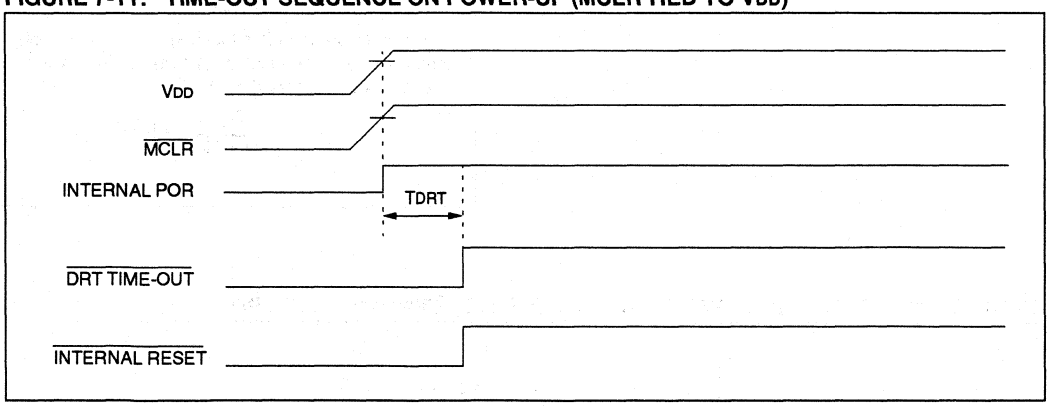
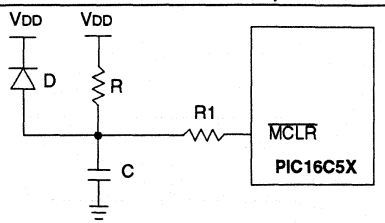


FIGURE 7-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD)



© 1995 Microchip Technology Inc.

FIGURE 7-12: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

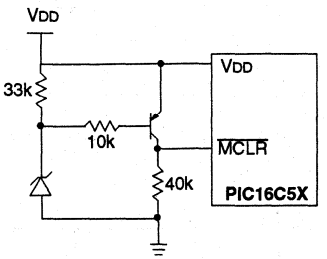


Note 1: External Power-On Reset circuit is required only if VDD power-up is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5 μA). A larger voltage drop will degrade VIH level on the $\overline{MCLR}$/VPP pin.

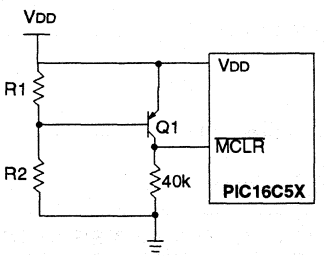
3: R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$/VPP pin breakdown due to ESD or EOS.

FIGURE 7-13: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when VDD goes below Vz + 0.7V (where Vz = Zener voltage).

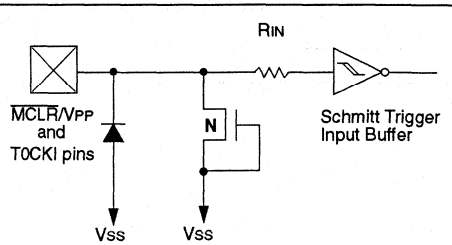
FIGURE 7-14: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \cdot \frac{R1}{R1 + R2} = 0.7V.$$

FIGURE 7-15: ELECTRICAL STRUCTURE OF THE $\overline{MCLR}$/VPP AND T0CKI PINS



© 1995 Microchip Technology Inc.

7.5 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device have been stopped (i.e., by executing the SLEEP instruction). During normal operation a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as a '0' (Section 7.1).

7.5.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). These periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out and generating a device RESET.

The $\overline{TO}$ bit (STATUS<4>) will be cleared upon a WDT time-out.

7.5.2 WDT PROGRAMMING CONSIDERATIONS

At worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

2

FIGURE 7-16: WATCHDOG TIMER BLOCK DIAGRAM

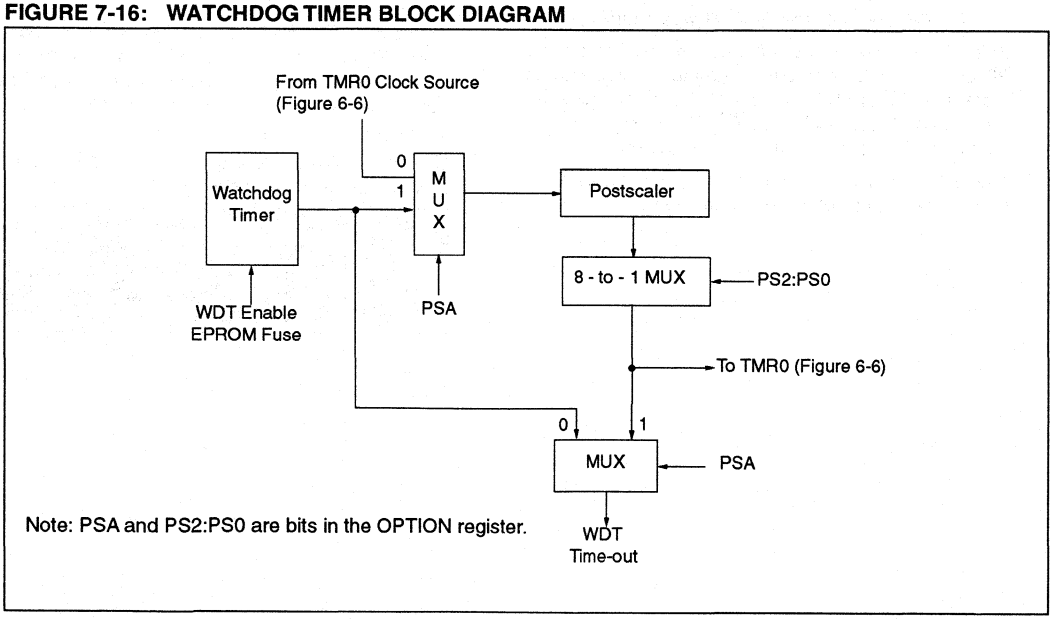


TABLE 7-6: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Config. Word	CP	CP	CP	CP	CP	WDTE	FOSC1	FOSC0
OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0

Note 1: CP7:CP4 are used by the PIC16CCR58A only. Unused in all other devices.
2: Shaded cells are not used by the Watchdog Timer.

© 1995 Microchip Technology Inc.

7.6 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared, and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low, or hi-impedance).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}$/VPP pin low.

For lowest current consumption while powered down, the T0CKI input should be at VDD or VSS and the $\overline{MCLR}$/VPP pin must be at a logic high level (VIHMC).

7.6.1 WAKE-UP FROM SLEEP

The device can wake from SLEEP through one of the following events:

1. An external reset input on the $\overline{MCLR}$/VPP pin.
2. A WDT time-out reset (if WDT was enabled).

Both of these events cause a device reset. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of device reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked.

The WDT is cleared when the device wakes from SLEEP, regardless of the wake-up source.

7.7 Code Protection

The code in the program memory can be protected by clearing the code protect bits.

In code protected mode, the configuration word will not be protected, allowing reading of all bits.

7.7.1 PIC16C54A, PIC16CR57A AND PIC16C58A

Once code protected, all memory locations read out in a scrambled fashion. For EPROM devices, program memory locations 40h and above cannot be further programmed. However, the first 64 locations, 00h - 3Fh, may be programmed. These locations are not considered secure.

7.7.2 PIC16CR58A

In a protected device, program memory locations 00h-3Fh read out normally. Locations 40h and higher cannot be read out.

7.8 ID Locations

Four memory locations are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify.

Use only the lower 4-bits of the ID locations and always program the upper 8-bits as '1's.

Note: Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

© 1995 Microchip Technology Inc.

8.0 INSTRUCTION SET SUMMARY

Each PIC16C5X instruction is a 12-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C5X instruction set summary in Table 8-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers is to be used by the instruction. (For the Enhanced PIC16CR57A, PIC16C58A and the PIC16CR58A, bits 6 and 5 in the FSR register determine the selected register bank.)

The destination designator specifies where the result of the operation is to be placed. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8- or 9-bit constant or literal value.

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

TABLE 8-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) d = 1 (store result in file register 'f') Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-Out bit
$\overline{PD}$	Power-Down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

2

TABLE 8-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	12-Bit Opcode MSb LSb	Status Affected	Notes
ADDWF	f,d	Add W and f	1	0001 11df ffff	C,DC,Z	1,2,4
ANDWF	f,d	AND W with f	1	0001 01df ffff	Z	2,4
CLRF	f	Clear f	1	0000 011f ffff	Z	4
CLRW	-	Clear W	1	0000 0100 0000	Z	
COMF	f, d	Complement f	1	0010 01df ffff	Z	
DECF	f, d	Decrement f	1	0000 11df ffff	Z	2,4
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	0010 11df ffff	None	2,4
INCF	f, d	Increment f	1	0010 10df ffff	Z	2,4
INCFSZ	f, d	Increment f, Skip if 0	1(2)	0011 11df ffff	None	2,4
IORWF	f, d	Inclusive OR W with f	1	0001 00df ffff	Z	2,4
MOVF	f, d	Move f	1	0010 00df ffff	Z	2,4
MOVWF	f	Move W to f	1	0000 001f ffff	None	1,4
NOP	-	No Operation	1	0000 0000 0000	None	
RLF	f, d	Rotate left f through Carry	1	0011 01df ffff	C	2,4
RRF	f, d	Rotate right f through Carry	1	0011 00df ffff	C	2,4
SUBWF	f, d	Subtract W from f	1	0000 10df ffff	C,DC,Z	1,2,4
SWAPF	f, d	Swap f	1	0011 10df ffff	None	2,4
XORWF	f, d	Exclusive OR W with f	1	0001 10df ffff	Z	2,4
BIT-ORIENTED FILE REGISTER OPERATIONS						
BCF	f, b	Bit Clear f	1	0100 bbbf ffff	None	2,4
BSF	f, b	Bit Set f	1	0101 bbbf ffff	None	2,4
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	0110 bbbf ffff	None	
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	0111 bbbf ffff	None	
LITERAL AND CONTROL OPERATIONS						
ANDLW	k	AND literal with W	1	1110 kkkk kkkk	Z	
CALL	k	Call subroutine	2	1001 kkkk kkkk	None	1
CLRWDT	k	Clear watchdog timer	1	0000 0000 0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Unconditional branch	2	101k kkkk kkkk	None	
IORLW	k	Inclusive OR literal with W	1	1101 kkkk kkkk	Z	
MOVLW	k	Move Literal to W	1	1100 kkkk kkkk	None	
OPTION	k	Load OPTION register	1	0000 0000 0010	None	
RETLW	k	Return, place literal in W	2	1000 kkkk kkkk	None	
SLEEP	-	Go into standby mode	1	0000 0000 0011	$\overline{TO}$,$\overline{PD}$	
TRIS	f	Load TRIS register	1	0000 0000 0fff	None	3
XORLW	k	Exclusive OR Literal to W	1	1111 kkkk kkkk	Z	

Note 1: the 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO. (Section 4.6)

2: When an I/O register is modified as a function of itself (e.g. `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

3: The instruction `TRIS f`, where f = 5, 6, or 7 causes the contents of the W register to be written to the tristate latches of PORTA, B or C, respectively. A '1' forces the pin to a hi-impedance state and disables the output buffers.

4: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared (if assigned to TMR0).

© 1995 Microchip Technology Inc.

ADDWF **Add W and f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

0001	11df	ffff

Description: Add the contents of the W register and register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDLW **And literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

1110	kkkk	kkkk

Description: The contents of the W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example:
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

0001	01df	ffff

Description: The contents of the W register are AND'ed with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ANDWF   FSR,    1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

0100	bbbf	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example:
```
BCF     FLAG_REG,   7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

© 1995 Microchip Technology Inc.

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow (f<b>)$

Status Affected: None

Encoding:

0101	bbbf	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example:
```
BSF     FLAG_REG,   7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **Bit Test f, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

0110	bbbf	ffff

Description: If bit 'b' in register 'f' is 0 then the next instruction is skipped.
If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and an NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address (HERE)
After Instruction
if FLAG<1> = 0,
PC = address (TRUE);
if FLAG<1> = 1,
PC = address (FALSE)

BTFSS **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: $0 \le f \le 127$
$0 \le b < 7$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

0111	bbbf	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and an NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSS   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0,
PC = address (FALSE);
if FLAG<1> = 1,
PC = address (TRUE)

© 1995 Microchip Technology Inc.

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC) + 1→ Top of Stack;
k → PC<8:0>;
(STATUS<6:5>) → PC<10:9>;
0 → PC<8>

Status Affected: None

Encoding:

1001	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eight bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example:
```
HERE    CALL    THERE
```
Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE)

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → (f);
1 → Z

Status Affected: Z

Encoding:

0000	011f	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example:
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W);
1 → Z

Status Affected: Z

Encoding:

0000	0100	0000

Description: The W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example:
```
CLRW
```
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0100

Description: The CLRWDT instruction resets the WDT. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example:
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

© 1995 Microchip Technology Inc.

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: $(\bar{f}) \rightarrow (dest)$

Status Affected: Z

Encoding:

0010	01df	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13

After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: $(f) - 1 \rightarrow (dest)$

Status Affected: Z

Encoding:

0000	11df	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
DECF    CNT,  1
```
Before Instruction
CNT = 0x01
Z = 0

After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: $(f) - 1 \rightarrow d$; skip if result = 0

Status Affected: None

Encoding:

0010	11df	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE      DECFSZ    CNT, 1
          GOTO      LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address (HERE)

After Instruction
CNT = CNT - 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE+1)

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: $k \rightarrow PC<8:0>$;
$STATUS<6:5> \rightarrow PC<10:9>$

Status Affected: None

Encoding:

101k	kkkk	kkkk

Description: GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example:
```
GOTO THERE
```
After Instruction
PC = address (THERE)

© 1995 Microchip Technology Inc.

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

0010	10df	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
INCF    CNT,    1
```
Before Instruction
CNT = 0xFF
Z = 0

After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

0011	11df	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, then the next instruction, which is already fetched, is discarded and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE      INCFSZ      CNT,  1
          GOTO        LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address (HERE)

After Instruction
CNT = CNT + 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE +1)

IORLW **Inclusive OR literal with W**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

1101	kkkk	kkkk

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example:
```
IORLW   0x35
```
Before Instruction
W = 0x9A

After Instruction
W = 0xBF

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (dest)

Status Affected: Z

Encoding:

0001	00df	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
IORWF           RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91

After Instruction
RESULT = 0x13
W = 0x93

© 1995 Microchip Technology Inc.

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

0010	00df	ffff

Description: The contents of register 'f' is moved to destination 'd'. If 'd' is 0, destination is the W register. If 'd' is 1, the destination is file register 'f'. 'd' is 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:
```
MOVF    FSR,    0
```
After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

1100	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into the W register. The don't cares will assemble as 0s.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

0000	001f	ffff

Description: Move data from the W register to register 'f'.

Words: 1

Cycles: 1

Example:
```
MOVWF   TEMP_REG
```
Before Instruction
TEMP_REG = 0xFF
W = 0x4F

After Instruction
TEMP_REG = 0x4F
W = 0x4F

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:
```
NOP
```


© 1995 Microchip Technology Inc.

OPTION — **Load OPTION Register**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

0000	0000	0010

Description: The content of the W register is loaded into the OPTION register.

Words: 1

Cycles: 1

Example

```
OPTION
```

Before Instruction
W = 0x07

After Instruction
OPTION = 0x07

RETLW — **Return, place literal in W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

1000	kkkk	kkkk

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

Words: 1

Cycles: 2

Example:

```
CALL TABLE  ;W contains
            ;table
            ;offset value
            ;W now has table
               value.
.
ADDWF PC    ;W = offset
RETLW k1    ;Begin table
RETLW k2    ;
.
.
.
RETLW kn    ; End of table
```

TABLE

Before Instruction
W = 0x07

After Instruction
W = value of k7

RLF — **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

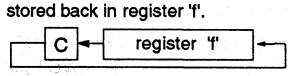
Operation: See description below

Status Affected: C

Encoding:

0011	01df	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example:

```
RLF     REG1,0
```

Before Instruction
REG1 = 1110 0110
C = 0

After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

© 1995 Microchip Technology Inc.

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

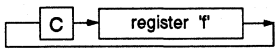
Operation: See description below

Status Affected: C

Encoding:

0011	00df	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RRF     REG1,0
```
Before Instruction
```
REG1  =  1110 0110
C     =  0
```
After Instruction
```
REG1  =  1110 0110
W     =  0111 0011
C     =  1
```

SLEEP **Enter SLEEP Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0011

Description: Time-out status bit ($\overline{TO}$) is set. The power down status bit ($\overline{PD}$) is cleared. The WDT and its prescaler are cleared.

The processor is put into SLEEP mode with the oscillator stopped. See section on SLEEP for more details.

Words: 1

Cycles: 1

Example:
```
SLEEP
```

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) → (dest)

Status Affected: C, DC, Z

Encoding:

0000	10df	ffff

Description: Subtract (2's complement method) the W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1, 1
```
Before Instruction
```
REG1  =  3
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  1
W     =  2
C     =  1     ; result is positive
```

Example 2:

Before Instruction
```
REG1  =  2
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  0
W     =  2
C     =  1     ; result is zero
```

Example 3:

Before Instruction
```
REG1  =  1
W     =  2
C     =  ?
```
After Instruction
```
REG1  =  FF
W     =  2
C     =  0     ; result is negative
```


© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<3:0>) → (dest<7:4>);
(f<7:4>) → (dest<3:0>)

Status Affected: None

Encoding:

0011	10df	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example
```
SWAPF   REG1, 0
```
Before Instruction
REG1 = 0xA5

After Instruction
REG1 = 0xA5
W = 0X5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: $5 \le f \le 7$

Operation: (W) → TRIS register f

Status Affected: None

Encoding:

0000	0000	0fff

Description: TRIS register 'f' (f = 5, 6, or 7) is loaded with the contents of the W register

Words: 1

Cycles: 1

Example
```
TRIS    PORTA
```
Before Instruction
W = 0XA5

After Instruction
TRISA = 0XA5

XORLW **Exclusive OR literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

1111	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
XORLW 0xAF
```
Before Instruction
W = 0xB5

After Instruction
W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

0001	10df	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
XORWF   REG     1
```
Before Instruction
REG = 0xAF
W = 0xB5

After Instruction
REG = 0x1A
W = 0xB5

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

9.0 ELECTRICAL CHARACTERISTICS FOR PIC16C54A

9.1 Absolute Maximum Ratings†

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... - 65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)
Total power dissipation (Note 1) ..... 800 mW
Max. current out of V_{SS} pin ..... 150 mA
Max. current into V_{DD} pin ..... 100 mA
Max. current into an input pin (T0CKI only) ..... ±500 µA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA
Max. output current sunk by any I/O pin ..... 25 mA
Max. output current sourced by any I/O pin ..... 20 mA
Max. output current sourced by a single I/O port (PORTA or B) ..... 40 mA
Max. output current sunk by a single I/O port (PORTA or B) ..... 50 mA

Note 1: Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows:

$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

† NOTICE: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 9-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C54A-04	16C54A-10	16C54A-20	16LC54A-04
RC	V_{DD}: 3.0V to 6.25V I_{DD}: 2.4 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 3.0V to 6.25V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 3.0V to 6.25V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 2.5V to 6.25V I_{DD}: 0.5 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 2.5V WDT dis Freq: 2 MHz max.
XT	V_{DD}: 3.0V to 6.25V I_{DD} 2.4 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 3.0V to 6.25V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 3.0V to 6.25V I_{DD}: 1.7 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 2.5V to 6.25V I_{DD}: 0.5 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 2.5V WDT dis Freq: 2 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 2.4 mA typ. at 5.5V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 8 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 10 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 16 mA max. at 5.5V I_{PD}: 4 µA max. at 3.0V WDT dis Freq: 20 MHz max.	Do not use in HS mode
LP	V_{DD}: 3.0V to 6.25V I_{DD}: 14 µA typ. at 32kHz, 3.0V I_{PD}: 0.25 µA typ. at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	Do not use in LP mode	V_{DD}: 2.5V to 6.25V I_{DD}: 27 µA max. at 32kHz, 2.5V I_{PD}: 4 µA max. at 2.5V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

TABLE 9-2: DC CHARACTERISTICS: PIC16C54A-04 (COMMERCIAL, INDUSTRIAL)

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 4.5	—	6.25 5.5	V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD	— — —	1.8 14 17	2.4	mA µA µA	XT and RC options (Note 4) FOSC = 4 MHz, VDD = 5.5V LP osc option, Commercial FOSC = 32 kHz, VDD = 3.0V, WDT disabled LP option, Industrial FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3) WDT enabled WDT disabled	IPD	— — — —	4 5 0.25 0.3	12 14 4 5	µA µA µA µA	VDD = 3.0V, Commercial VDD = 3.0V, Industrial VDD = 3.0V, Commercial VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

**TABLE 9-3: DC CHARACTERISTICS: PIC16C54A-10 (COMMERCIAL, INDUSTRIAL)
PIC16C54A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 4.5	-	6.25 5.5	V	XT, RC and LP options HS option
RAM Data Retention Voltage (Note 2)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 1)	IDD	— — —	1.7 2.4 4.5	3.3 8 16	mA mA mA	XT and RC options (Note 4) FOSC = 4 MHz, VDD = 5.5V HS option FOSC = 10 kHz, VDD = 5.5V FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3) WDT enabled WDT disabled	IPD	— — — —	4 5 0.25 0.3	12 14 4 5	µA µA µA µA	VDD = 3.0V, Commercial VDD = 3.0V, Industrial VDD = 3.0V, Commercial VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

TABLE 9-4: DC CHARACTERISTICS: PIC16LC54A-04 (COMMERCIAL, INDUSTRIAL)

DC CHARACTERISTICS POWER SUPPLY PINS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.5	—	6.25	V	XT, RC and LP options
RAM Data Retention Voltage (Note 2)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 1)	IDD					XT and RC options (Note 4)
		—	0.5		mA	FOSC = 2 MHz, VDD = 5.5V
						LP option, Commercial
		—	11	27	µA	FOSC = 32 kHz, VDD = 2.5V WDT disabled
						LP option, Industrial
			14	35	µA	FOSC = 32 kHz, VDD = 2.5V WDT disabled
Power Down Current (Note 3)	IPD					
WDT enabled		—	2.5	12	µA	VDD = 2.5V, Commercial
		—	3.5	14	µA	VDD = 2.5V, Industrial
WDT disabled		—	0.25	4	µA	VDD = 2.5V, Commercial
		—	0.3	5	µA	VDD = 2.5V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

**TABLE 9-5: DC CHARACTERISTICS: PIC16C54A-04 (COMMERCIAL, INDUSTRIAL)
PIC16LC54A-04 (COMMERCIAL, INDUSTRIAL)
PIC16C54A-10 (COMMERCIAL, INDUSTRIAL)
PIC16C54A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$		VSS		0.15 VDD	V	
T0CKI		VSS		0.15 VDD	V	
OSC1		VSS		0.15 VDD	V	RC option only (Note 4)
OSC1		VSS		0.3 VDD	V	XT, HS and LP options
Input High Voltage						
I/O ports	VIH	0.2 VDD+1V		VDD	V	For all VDD (Note 5)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 5)
$\overline{MCLR}$		0.85 VDD		VDD	V	
TOCKI		0.85 VDD		VDD	V	
OSC1		0.85 VDD		VDD	V	RC option only (Note 4)
OSC1		0.7 VDD		VDD	V	XT, HS and LP options
Input Leakage Current (Note 3)						For VDD ≤ 5.5V
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS +0.25V (Note 2)
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD (Note 2)
TOCKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD XT, HS and LP options
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

2: The leakage current on the $\overline{MCLR}$/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

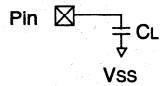
3: Negative current is defined as coming out of the pin.

4: In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. Do not drive the PIC16C54A with an external clock in RC mode.

5: The user may use better of the two specifications.

9.2 Timing Diagrams and Specifications

FIGURE 9-1: LOAD CONDITIONS



CL = 50 pF for all pins except OSC2
15 pF for OSC2 in XT, HS or
LP modes when external clock
is used to drive OSC1.

© 1995 Microchip Technology Inc.

FIGURE 9-2: EXTERNAL CLOCK TIMING

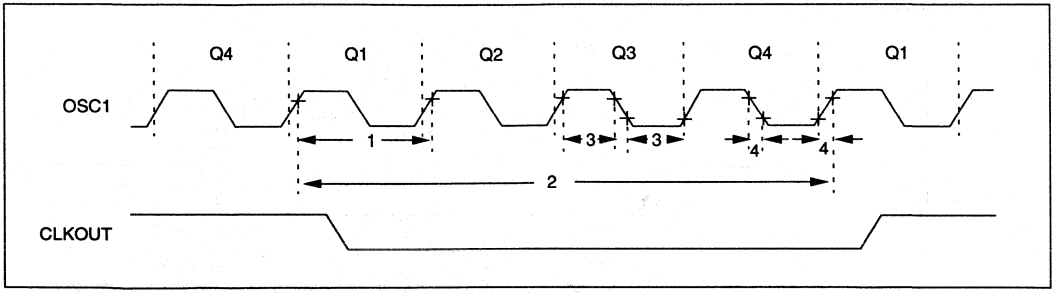


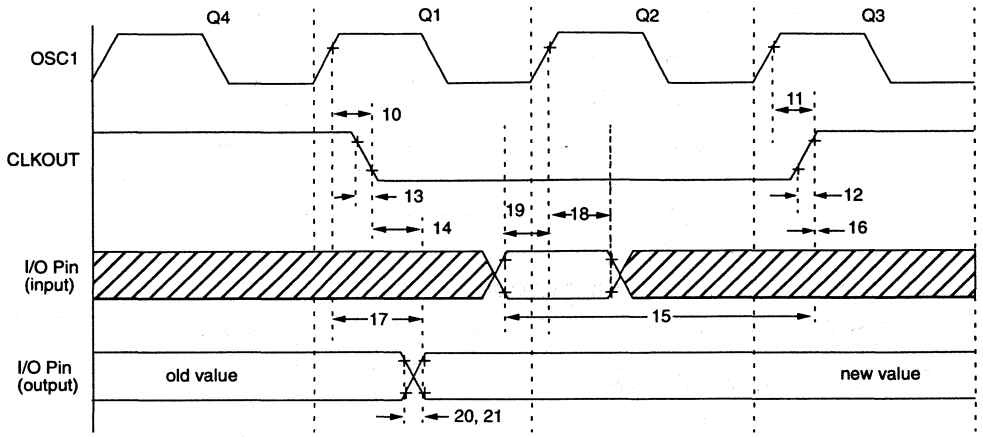
TABLE 9-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C5XA-04)
			DC	—	10	MHz	HS osc mode (PIC16C5XA-10)
			DC	—	20	MHz	HS osc mode (PIC16C5XA-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C5XA-04)
			4	—	10	MHz	HS osc mode (PIC16C5XA-10)
			4	—	20	MHz	HS osc mode (PIC16C5XA-20)
			5	—	200	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C5XA-04)
			100	—	—	ns	HS osc mode (PIC16C5XA-10)
			50	—	—	ns	HS osc mode (PIC16C5XA-20)
			5.0	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C5XA-04)
			100	—	250	ns	HS osc mode (PIC16C5XA-10)
			50	—	250	ns	HS osc mode (PIC16C5XA-20)
			5	—	200	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	1.0	—	DC	µs	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 9-3: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 9-7: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5 TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	Note 2
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	Note 2
21	TioF	Port output fall time	—	10	25	ns	Note 2

* These parameters are characterized but not tested.

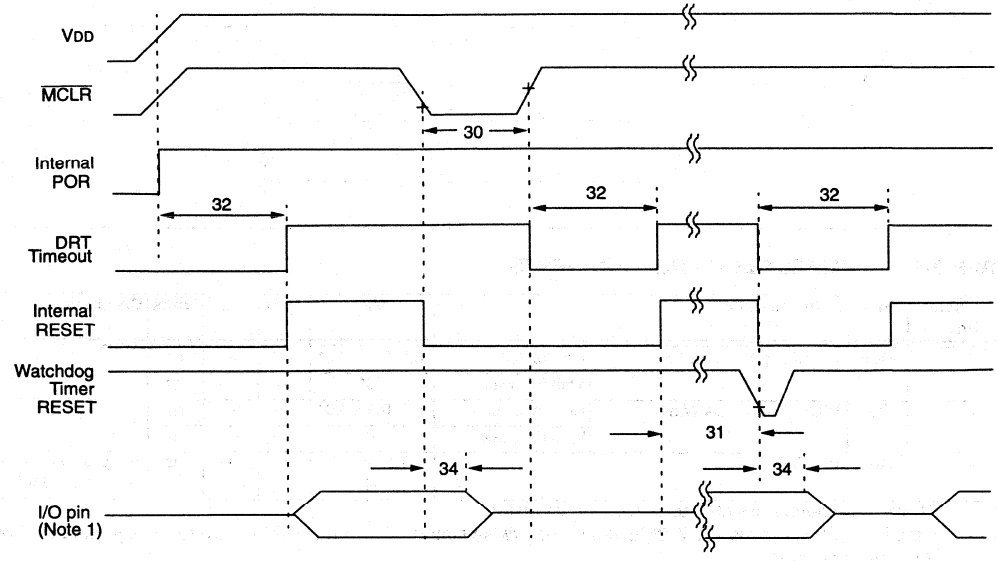
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

2: See Figure 9-1 for loading conditions.

© 1995 Microchip Technology Inc.

FIGURE 9-4: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of hi-impedance mode by enabling the output drivers in software.

TABLE 9-8: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	9*	18	30*	ms	VDD = 5V, -40°C to +125°C
32	TDRT	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5V, -40°C to +125°C
34	TioZ	I/O Hi-impedance from $\overline{MCLR}$ Low or WDT timeout			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 9-5: TIMER0 CLOCK TIMINGS

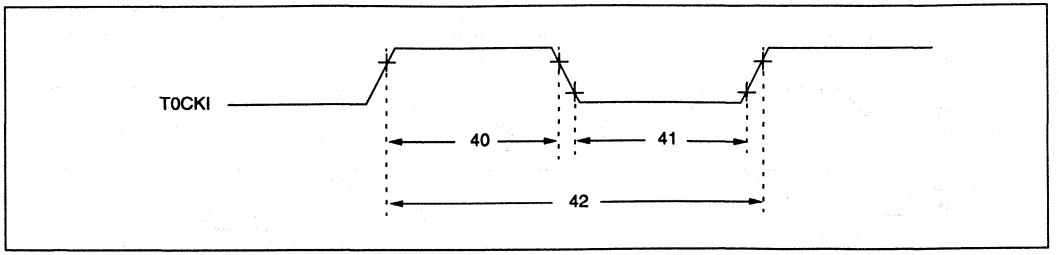


TABLE 9-9: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

10.0 ELECTRICAL CHARACTERISTICS FOR PIC16CR57A

10.1 Absolute Maximum Ratings†

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... - 65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)
Total power dissipation (Note 1) ..... 800 mW
Max. current out of V_{SS} pin ..... 150 mA
Max. current into V_{DD} pin ..... 100 mA
Max. current into an input pin (T0CKI only) ..... ±500 µA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA
Max. output current sunk by any I/O pin ..... 25 mA
Max. output current sourced by any I/O pin ..... 20 mA
Max. output current sourced by a single I/O port
  PORTA ..... 50 mA
  PORTB or C ..... 100 mA
Max. Output Current sunk by a single I/O port
  PORTA ..... 50 mA
  PORTB or C ..... 100 mA

Note 1: Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{(V_{DD}-V_{OH}) \times I_{OH}\} + \Sigma(V_{OL} \times I_{OL})$$

† NOTICE: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

© 1995 Microchip Technology Inc.

TABLE 10-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16CR57A-04	16CR57A-10	16CR57A-20	16LCR57A-04
RC	VDD: 2.5 V to 6.25 V IDD: 3.3 mA Max at 5.5 V IPD: 9 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 2.5V WDT dis Freq: 4 MHz Max
XT	VDD: 2.5V to 6.25V IDD: 3.3 mA max. at 5.5V IPD: 9 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 2.5V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 2.5V WDT dis Freq: 4 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 4.8 mA typ. at 5.5V IPD: 0.6 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V IPD: 9 µA max. at 3.0V WDT dis Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 20 mA max. at 5.5V IPD: 9 µA max. at 3.0V WDT dis Freq: 20 MHz Max	Do not use in HS mode
LP	VDD: 2.5V to 6.25V IDD: 15 µA typ. at 32 kHz, 3.0V IPD: 0.6 µA typ.at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	Do not use in LP mode	VDD: 2.5V to 6.25V IDD: 32 µA max. at 32 kHz, 2.5V IPD: 9 µA max. at 2.5V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

© 1995 Microchip Technology Inc.

**TABLE 10-2: DC CHARACTERISTICS: PIC16CR57A-04 (COMMERCIAL, INDUSTRIAL)
PIC16CR57A-10 (COMMERCIAL, INDUSTRIAL)
PIC16CR57A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.5 4.5	-	6.25 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					XT and RC options (Note 4)
			1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V
						HS option
			4.8	10	mA	FOSC = 10 MHz, VDD = 5.5V
			9.0	20	mA	FOSC = 20 MHz, VDD = 5.5V
						LP osc option, Commercial
			15	32	µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled
						LP option, Industrial
			19	40	µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3)	IPD					
WDT enabled		—	4	12	µA	VDD = 3.0V, Commercial
		—	5	14	µA	VDD = 3.0V, Industrial
WDT disabled		—	0.6	9	µA	VDD = 3.0V, Commercial
		—	0.8	12	µA	VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

TABLE 10-3: DC CHARACTERISTICS: PIC16LCR57A-04 (COMMERCIAL, INDUSTRIAL)

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.5	—	6.25	V	XT, RC and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					XT and RC options (Note 4)
			1.8	3.3	mA	FOSC = 4 MHz, VDD = 5.5V LP osc option, Commercial
			15	32	µA	FOSC = 32 kHz, VDD = 2.5V, WDT disabled LP option, Industrial
			19	40	µA	FOSC = 32 kHz, VDD = 2.5V, WDT disabled
Power Down Current (Note 3)						
WDT enabled	IPD	—	4	12	µA	VDD = 2.5V, Commercial
		—	5	14	µA	VDD = 2.5V, Industrial
WDT disabled		—	0.6	9	µA	VDD = 2.5V, Commercial
		—	0.8	12	µA	VDD = 2.5V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

**TABLE 10-4: DC CHARACTERISTICS: PIC16CR57A-04 (COMMERCIAL, INDUSTRIAL)
PIC16LCR57A-04 (COMMERCIAL, INDUSTRIAL)
PIC16CR57A-10 (COMMERCIAL, INDUSTRIAL)
PIC16CR57A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$		VSS		0.15 VDD	V	
T0CKI		VSS		0.15 VDD	V	
OSC1		VSS		0.15 VDD	V	RC option only (Note 4)
OSC1		VSS		0.3 VDD	V	XT, HS and LP options
Input High Voltage						
I/O ports		0.45 VDD		VDD	V	For all VDD (Note 5)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 5)
		0.36 VDD		VDD	V	VDD > 5.5V
$\overline{MCLR}$		0.85VDD		VDD	V	
T0CKI		0.85 VDD		VDD	V	RC option only (Note 4)
OSC1		0.85 VDD		VDD	V	XT, HS and LP options
OSC1		0.7 VDD		VDD	V	
Input Leakage Current (Note 3)						**For VDD ≤ 5.5V**
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS +0.25V (Note 2)
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD (Note 2)
T0CKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD XT, HS and LP options
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

2: The leakage current on the $\overline{MCLR}$/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

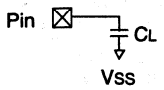
3: Negative current is defined as coming out of the pin.

4: In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. Do not drive the PIC16C54A with an external clock in RC mode.

5: The user may use better of the two specifications.

10.2 Timing Diagrams and Specifications

FIGURE 10-1: LOAD CONDITIONS



C_L = 50 pF for all pins except OSC2
15 pF for OSC2 in XT, HS or
LP modes when external clock
is used to drive OSC1.

© 1995 Microchip Technology Inc.

FIGURE 10-2: EXTERNAL CLOCK TIMING

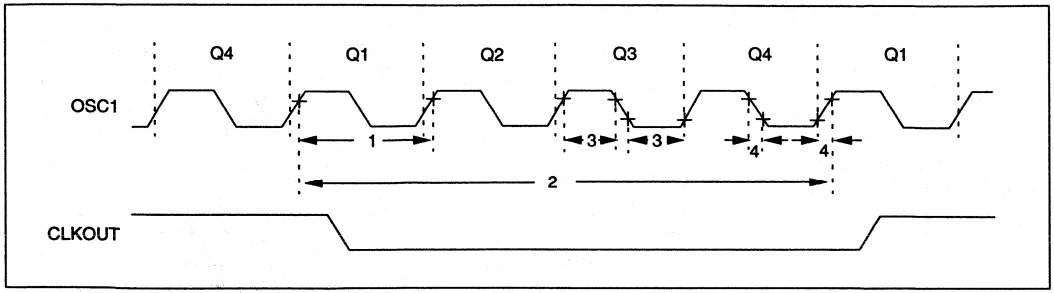


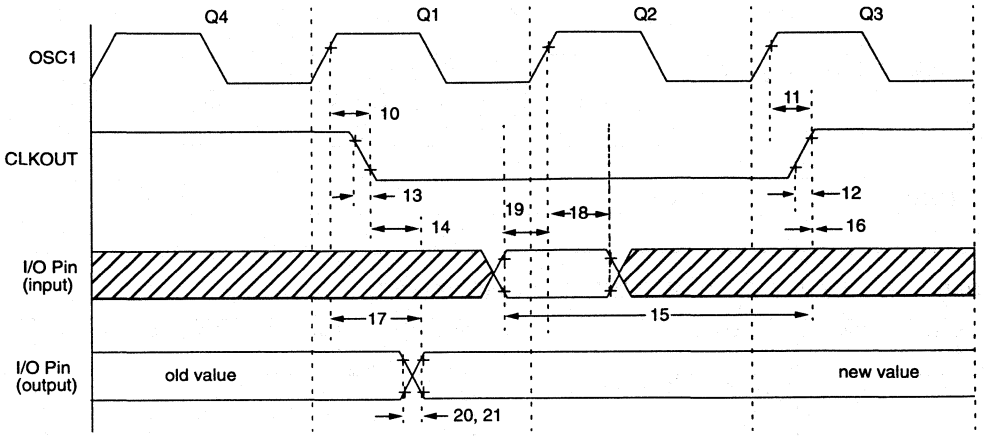
TABLE 10-5: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C5XA-04)
			DC	—	10	MHz	HS osc mode (PIC16C5XA-10)
			DC	—	20	MHz	HS osc mode (PIC16C5XA-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C5XA-04)
			4	—	10	MHz	HS osc mode (PIC16C5XA-10)
			4	—	20	MHz	HS osc mode (PIC16C5XA-20)
			5	—	200	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C5XA-04)
			100	—	—	ns	HS osc mode (PIC16C5XA-10)
			50	—	—	ns	HS osc mode (PIC16C5XA-20)
			5.0	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C5XA-04)
			100	—	250	ns	HS osc mode (PIC16C5XA-10)
			50	—	250	ns	HS osc mode (PIC16C5XA-20)
			5	—	200	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	1.0	—	DC	µs	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 10-3: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 10-6: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5 TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	Note 2
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	Note 2
21	TioF	Port output fall time	—	10	25	ns	Note 2

* These parameters are characterized but not tested.

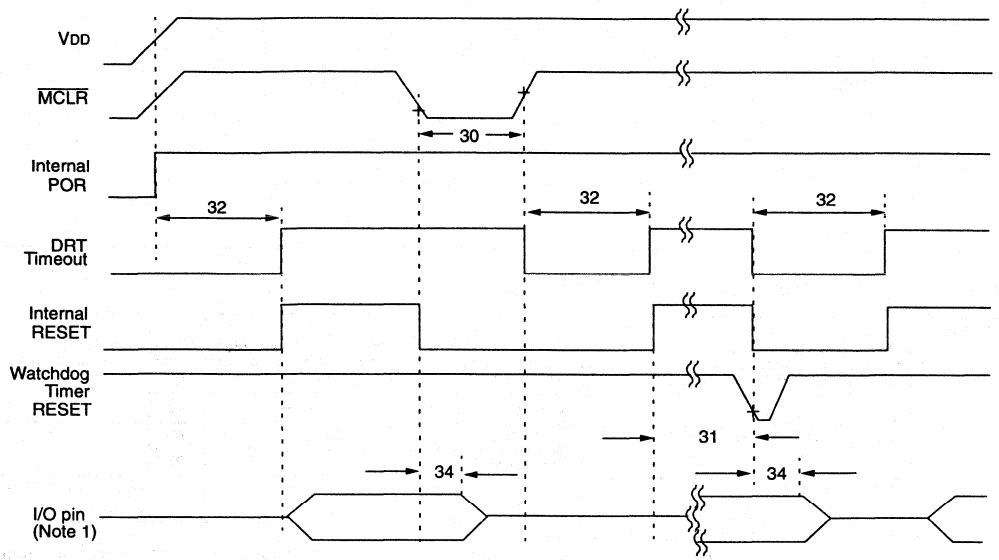
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

2: See Figure 10-1 for loading conditions.

© 1995 Microchip Technology Inc.

FIGURE 10-4: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of hi-impedance mode by enabling the output drivers in software.

TABLE 10-7: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	9*	18	30*	ms	VDD = 5V, -40°C to +125°C
32	TDRT	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O Hi-impedance from MCLR Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 10-5: TIMER0 CLOCK TIMINGS

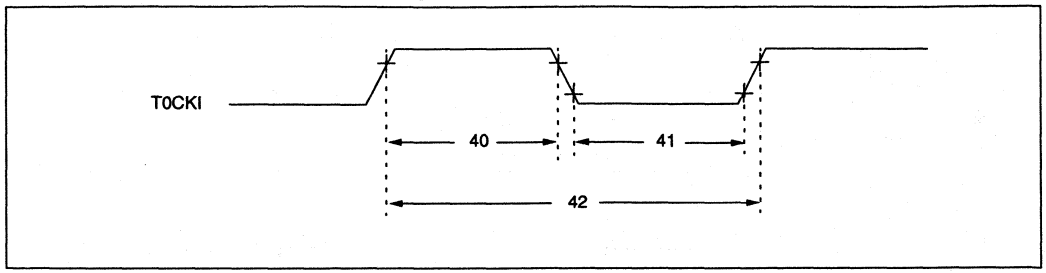


TABLE 10-8: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

11.0 ELECTRICAL CHARACTERISTICS FOR PIC16C58A/CR58A

11.1 Absolute Maximum Ratings†

Ambient temperature Under Bias ..... - 55°C to +125°C
Storage temperature ..... - 65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)
Total power dissipation (Note 1) ..... 800 mW
Max. current out of V_{SS} pin ..... 150 mA
Max. current into V_{DD} pin ..... 100 mA
Max. current into an input pin (T0CKI only) ..... ±500 µA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA
Max. output current sunk by any I/O pin ..... 25 mA
Max. output current sourced by any I/O pin ..... 20 mA
Max. output current sourced by a single I/O port
PORTA ..... 50 mA
PORTB ..... 100 mA
Max. Output Current sunk by a single I/O port
PORTA ..... 50 mA
PORTB ..... 100 mA

Note 1: Total power dissipation should not exceed 800 mW for the package. Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \Sigma I_{OH}\} + \Sigma \{(V_{DD}-V_{OH}) \times I_{OH}\} + \Sigma(V_{OL} \times I_{OL})$$

† NOTICE: Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 11-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C58A-04	16C58A-10	16C58A-20	16LC58A-04
RC	VDD: 3.0V to 6.25V IDD: 2.5 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 2.5V to 6.25V IDD: 0.5 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 2 MHz max.
XT	VDD: 3.0V to 6.25V IDD: 2.5 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 2.5V to 6.25V IDD: 0.5 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 2 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 1.9 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 8 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 17 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 20 MHz max.	Do not use in HS mode
LP	VDD: 3.0V to 6.25V IDD: 15 µA typ. at 32 kHz, 3.0V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	Do not use in LP mode	VDD: 2.5V to 6.25V IDD: 28 µA max. at 32 kHz, 2.5V IPD: 4 µA max. at 2.5V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

TABLE 11-2: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES) CON'T

OSC	16CR58A-04	16CR58A-10	16CR58A-20	16LCR58A-04
RC	VDD: 3.0V to 6.25V IDD: 2.5 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 2.5V to 6.25V IDD: 0.5 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 2 MHz max.
XT	VDD: 3.0V to 6.25V IDD: 2.5 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 3.0V to 6.25V IDD: 1.8 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz Max	VDD: 2.5V to 6.25V IDD: 0.5 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 2 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 1.9 mA typ. at 5.5V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 8 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 10 MHz max.	VDD: 4.5V to 5.5V IDD: 17 mA max. at 5.5V IPD: 4 µA max. at 3.0V WDT dis Freq: 20 MHz max.	Do not use in HS mode
LP	VDD: 3.0V to 6.25V IDD: 15 µA typ. at 32 kHz, 3.0V IPD: 0.25 µA typ. at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	Do not use in LP mode	VDD: 2.5V to 6.25V IDD: 28 µA max. at 32 kHz, 2.5V IPD: 4 µA max. at 2.5V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

© 1995 Microchip Technology Inc.

TABLE 11-3: DC CHARACTERISTICS: PIC16C58A-04 (COMMERCIAL, INDUSTRIAL) PIC16CR58A-04 (COMMERCIAL, INDUSTRIAL)

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0	—	6.25	V	XT, RC and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					XT and RC options (Note 4)
			1.9	2.5	mA	FOSC = 4 MHz, VDD = 5.5V LP osc option, Commercial
			15		µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled LP option, Industrial
			18		µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3)	IPD					
WDT enabled		—	4	12	µA	VDD = 3.0V, Commercial
		—	5	14	µA	VDD = 3.0V, Industrial
WDT disabled		—	0.25	4	µA	VDD = 3.0V, Commercial
		—	0.3	5	µA	VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula: IR = VDD/2Rext (mA) with Rext in kOhm.

**TABLE 11-4: DC CHARACTERISTICS: PIC16C58A-10 (COMMERCIAL, INDUSTRIAL)
PIC16C58A-20 (COMMERCIAL, INDUSTRIAL)
PIC16CR58A-10 (COMMERCIAL, INDUSTRIAL)
PIC16CR58A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 4.5	—	6.25 5.5	V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD		1.8 2.5 4.7	 8 17	mA mA mA	XT and RC options (Note 4) FOSC = 4 MHz, VDD = 5.5V HS option FOSC = 10 MHz, VDD = 5.5V FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3) WDT enabled WDT disabled	IPD	 — — — —	 4 5 0.25 0.3	 12 14 4 5	 µA µA µA µA	 VDD = 3.0V, Commercial VDD = 3.0V, Industrial VDD = 3.0V, Commercial VDD = 3.0V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are: OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

**TABLE 11-5: DC CHARACTERISTICS: PIC16LC58A-04 (COMMERCIAL, INDUSTRIAL)
PIC16LCR58A-04 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage: VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.5	-	6.25	V	XT, RC and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					XT and RC options (Note 4)
			0.5		mA	FOSC = 2 MHz, VDD = 5.5V
						LP option, Commercial
			12	28	µA	FOSC = 32 kHz, VDD = 2.5V, WDT disabled
						LP option, Industrial
			15	37	µA	FOSC = 32 kHz, VDD = 2.5V, WDT disabled
Power Down Current (Note 3)	IPD					
WDT enabled		—	2.5	12	µA	VDD = 2.5V, Commercial
		—	3.5	14	µA	VDD = 2.5V, Industrial
WDT disabled		—	0.25	4	µA	VDD = 2.5V, Commercial
		—	0.3	5	µA	VDD = 2.5V, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, bus rate, internal code execution pattern, and temperature also have an impact on the current consumption.

a) The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
b) For stand-by current measurements, the conditions are the same, except that the device is in SLEEP mode.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: RC mode does not include current through Rext. The current through the resistor can be estimated by the formula:
IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

**TABLE 11-6: DC CHARACTERISTICS: PIC16C58A-04 (COMMERCIAL, INDUSTRIAL)
PIC16LC58A-04 (COMMERCIAL, INDUSTRIAL)
PIC16C58A-10 (COMMERCIAL, INDUSTRIAL)
PIC16C58A-20 (COMMERCIAL, INDUSTRIAL)
PIC16CR58A-04 (COMMERCIAL, INDUSTRIAL)
PIC16LCR58A-04 (COMMERCIAL, INDUSTRIAL)
PIC16CR58A-10 (COMMERCIAL, INDUSTRIAL)
PIC16CR58A-20 (COMMERCIAL, INDUSTRIAL)**

DC CHARACTERISTICS POWER SUPPLY PINS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C for automotive, -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL	VSS		0.2 VDD	V	Pin at hi-impedance
$\overline{MCLR}$		VSS		0.15 VDD	V	
T0CKI		VSS		0.15 VDD	V	
OSC1		VSS		0.15 VDD	V	RC option only (Note 4)
OSC1		VSS		0.3 VDD	V	XT, HS and LP options
Input High Voltage						
I/O ports		0.45 VDD		VDD	V	For all VDD (Note 5)
		2.0		VDD	V	4.0V < VDD ≤ 5.5V (Note 5)
		0.36 VDD		VDD	V	VDD > 5.5V
$\overline{MCLR}$		0.85 VDD		VDD	V	
T0CKI		0.85 VDD		VDD	V	RC option only (Note 4)
OSC1		0.85 VDD		VDD	V	XT, HS and LP options
OSC1		0.7 VDD		VDD	V	
Input Leakage Current (Note 3)						**For VDD ≤ 5.5V**
I/O ports	IIL	-1	0.5	+1	µA	VSS ≤ VPIN ≤ VDD Pin at hi-impedance
$\overline{MCLR}$		-5			µA	VPIN = VSS +0.25V (Note 2)
$\overline{MCLR}$			0.5	+5	µA	VPIN = VDD (Note 2)
T0CKI		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD
OSC1		-3	0.5	+3	µA	VSS ≤ VPIN ≤ VDD XT, HS and LP options
Output Low Voltage						
I/O ports	VOL			0.6	V	IOL = 8.7 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)				0.6	V	IOL = 1.6 mA, VDD = 4.5V
Output High Voltage						
I/O ports (Note 4)	VOH	VDD-0.7			V	IOH = -5.4 mA, VDD = 4.5V
OSC2/CLKOUT (RC option only)		VDD-0.7			V	IOH = -1.0 mA, VDD = 4.5V

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total power dissipation as stated under absolute maximum ratings must not be exceeded.

2: The leakage current on the $\overline{MCLR}$/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

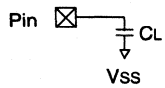
4: In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. Do not drive the PIC16C54A with an external clock in RC mode.

5: The user may use better of the two specifications.

© 1995 Microchip Technology Inc.

11.2 Timing Diagrams and Specifications

FIGURE 11-1: LOAD CONDITIONS



CL = 50 pF for all pins except OSC2
15 pF for OSC2 in XT, HS or
LP modes when external clock
is used to drive OSC1.

© 1995 Microchip Technology Inc.

FIGURE 11-2: EXTERNAL CLOCK TIMING

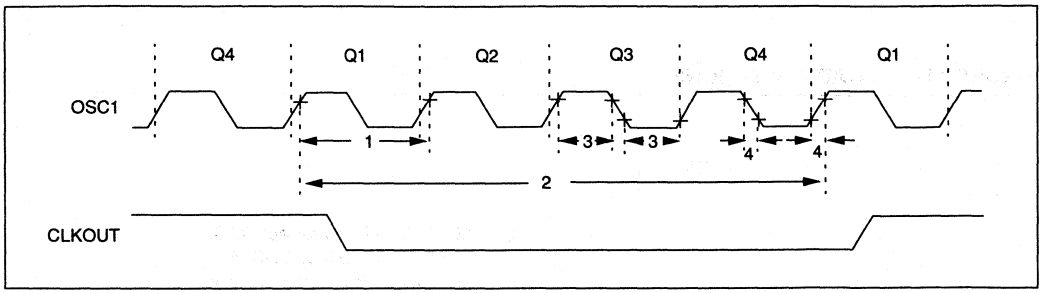


TABLE 11-7: EXTERNAL CLOCK TIMING REQUIREMENTS

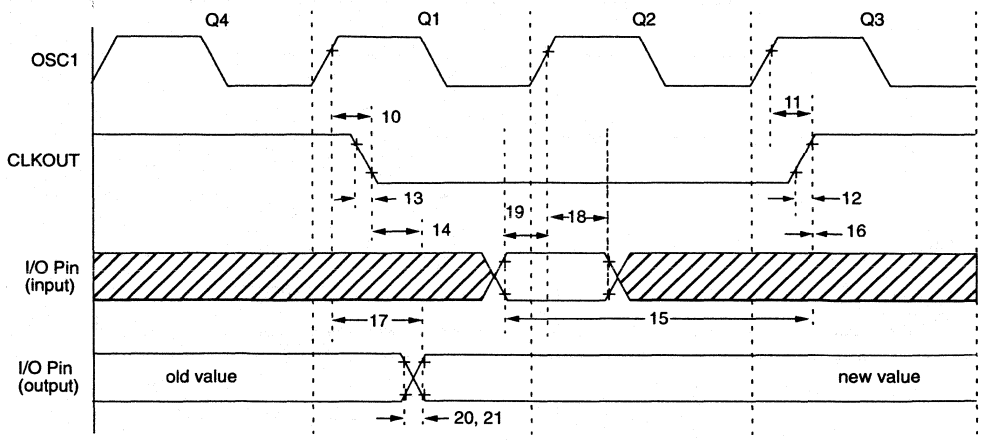
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C58A-04)
			DC	—	10	MHz	HS osc mode (PIC16C58A-10)
			DC	—	20	MHz	HS osc mode (PIC16C58A-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C58A-04)
			4	—	10	MHz	HS osc mode (PIC16C58A-10)
			4	—	20	MHz	HS osc mode (PIC16C58A-20)
			5	—	200	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C58A-04)
			100	—	—	ns	HS osc mode (PIC16C58A-10)
			50	—	—	ns	HS osc mode (PIC16C58A-20)
			5.0	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C58A-04)
			100	—	250	ns	HS osc mode (PIC16C58A-10)
			50	—	250	ns	HS osc mode (PIC16C58A-20)
			5	—	200	µs	LP osc mode
2	TCY	Instruction Cycle Time (Note 1)	0.2	—	DC	µs	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50	—	—	ns	XT oscillator
			2.0	—	—	µs	LP oscillator
			20	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			25	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 1995 Microchip Technology Inc.

FIGURE 11-3: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 11-8: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5 TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	Note 2
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	Note 2
21	TioF	Port output fall time	—	10	25	ns	Note 2

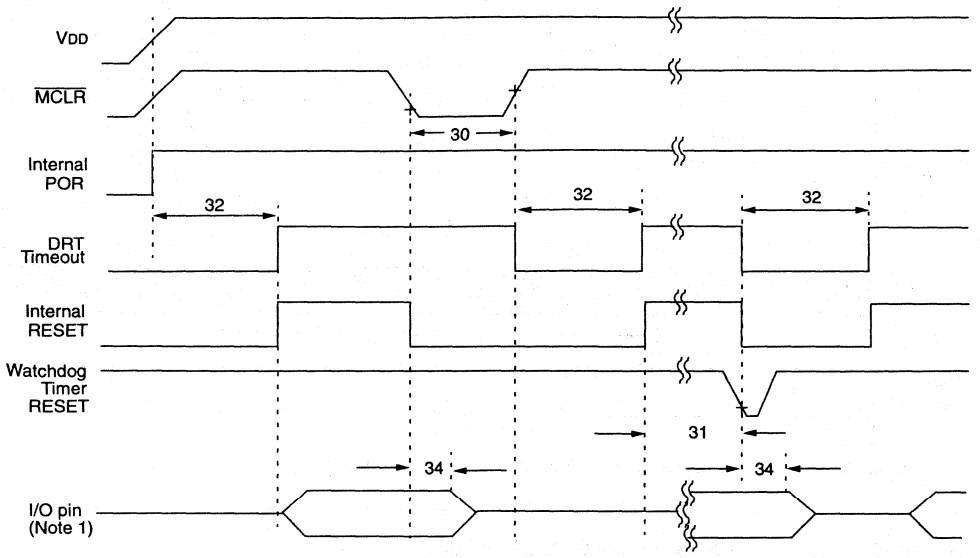
* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

2: See Figure 11-1 for loading conditions.

FIGURE 11-4: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of hi-impedance mode by enabling the output drivers in software.

TABLE 11-9: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C (PIC16C58A)
			1	—	—	ns	VDD = 5V, -40°C to +125°C (PIC16CR58A)
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	9*	18	30*	ms	VDD = 5V, -40°C to +125°C
32	TDRt	Device Reset Timer Period	9*	18*	30*	ms	VDD = 5V, -40°C to +125°C
34	Tioz	I/O Hi-impedance from $\overline{MCLR}$ Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

FIGURE 11-5: TIMER0 CLOCK TIMINGS

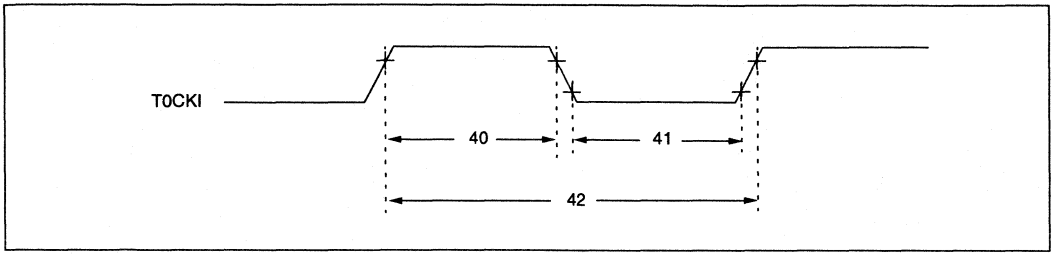


TABLE 11-10: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NOTES:

© 1995 Microchip Technology Inc.

12.0 DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

FIGURE 12-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

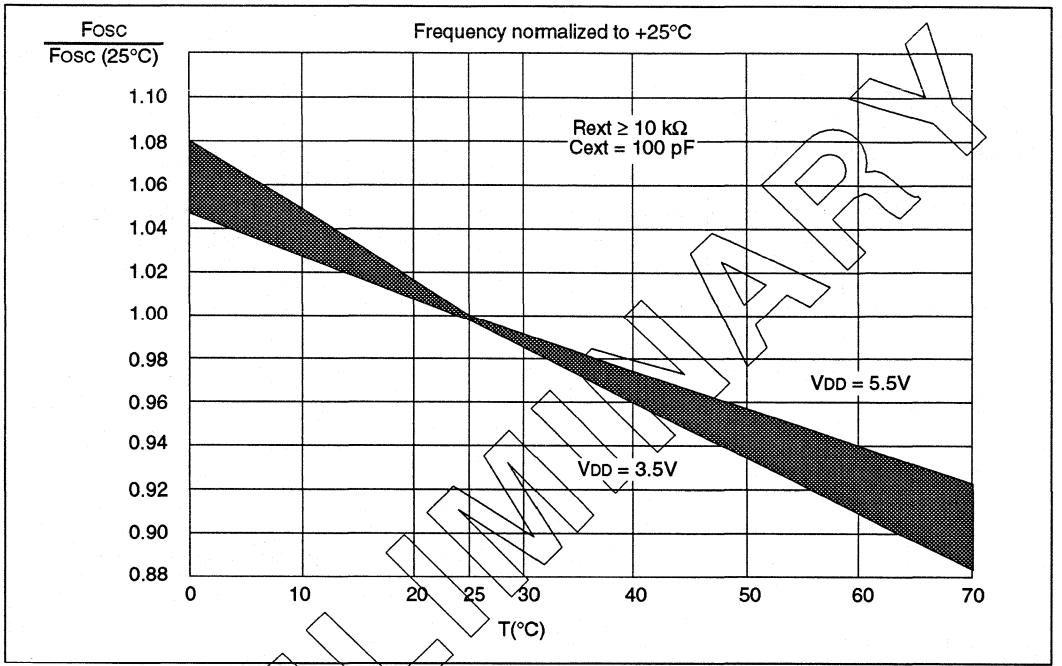


TABLE 12-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20 pF	3.3k	4.973 MHz	± 27%
	5k	3.82 MHz	± 21%
	10k	2.22 MHz	± 21%
	100k	262.15 kHz	± 31%
100 pF	3.3k	1.63 MHz	± 13%
	5k	1.19 MHz	± 13%
	10k	684.64 kHz	± 18%
	100k	71.56 kHz	± 25%
300 pF	3.3k	660.0 kHz	± 10%
	5.k	484.1 kHz	± 14%
	10k	267.63 kHz	± 15%
	160k	29.44 kHz	± 19%

The frequencies are measured on DIP packages.

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

PRELIMINARY

FIGURE 12-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

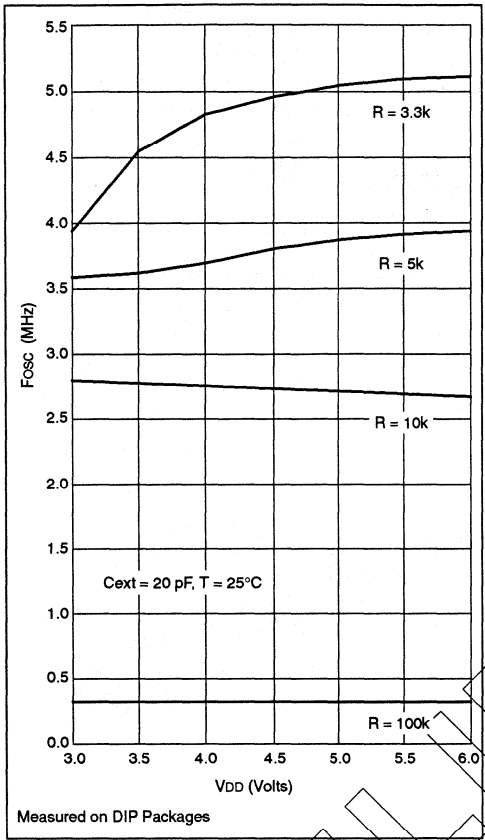


FIGURE 12-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

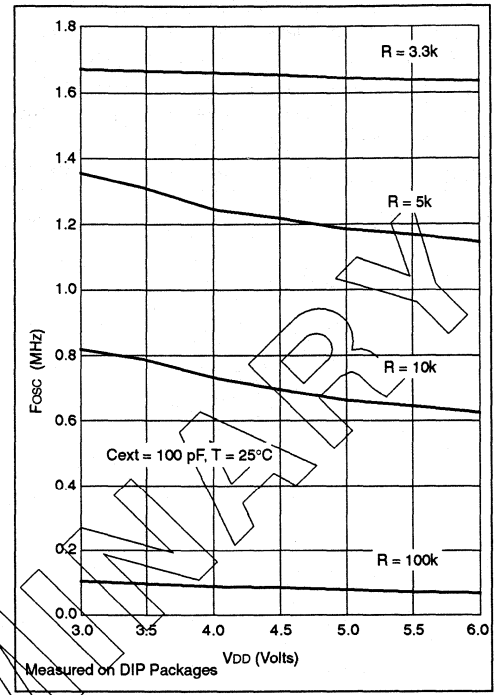
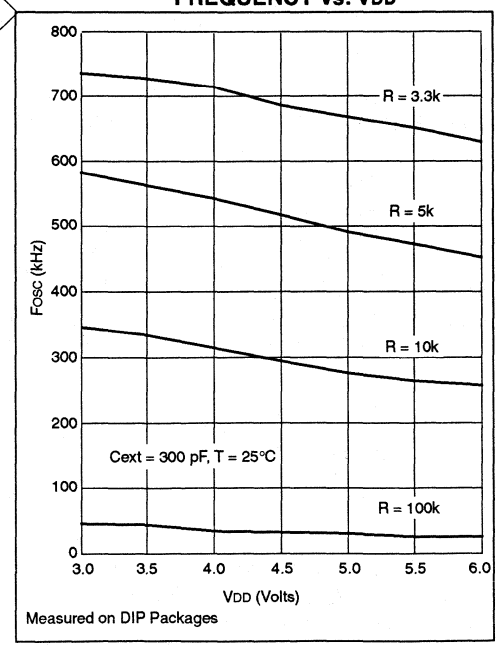


FIGURE 12-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



PRELIMINARY

 © 1995 Microchip Technology Inc.

FIGURE 12-5: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

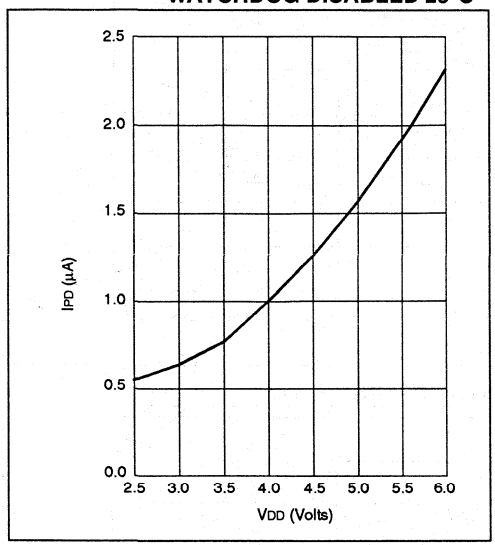


FIGURE 12-6: MAXIMUM IPD vs. VDD WATCHDOG DISABLED

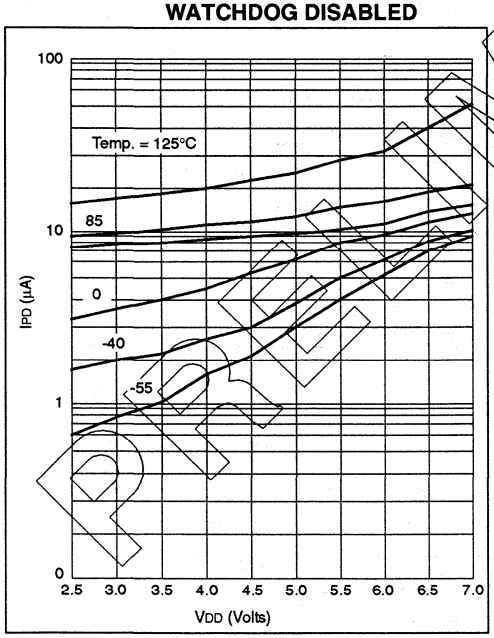


FIGURE 12-7: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

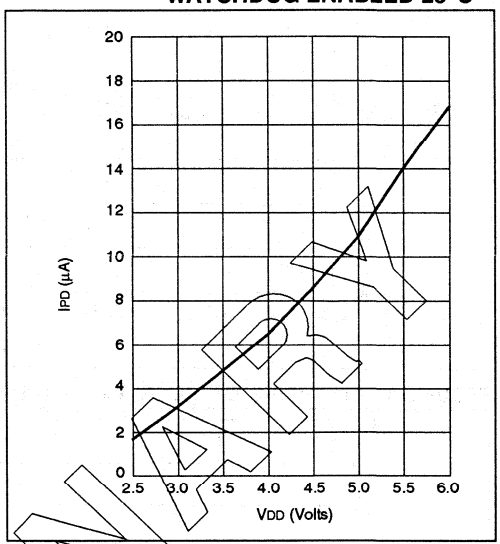
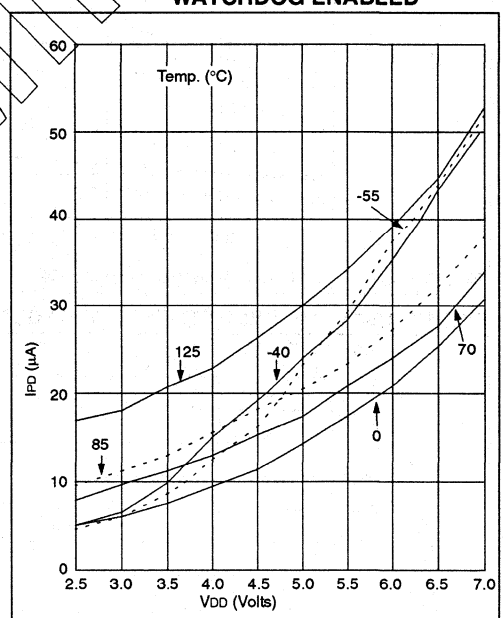


FIGURE 12-8: MAXIMUM IPD vs. VDD WATCHDOG ENABLED



IPD, with WDT enabled, has two components:
The leakage current which increases with higher temperature, and the operating current of the WDT logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

Preliminary

FIGURE 12-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD

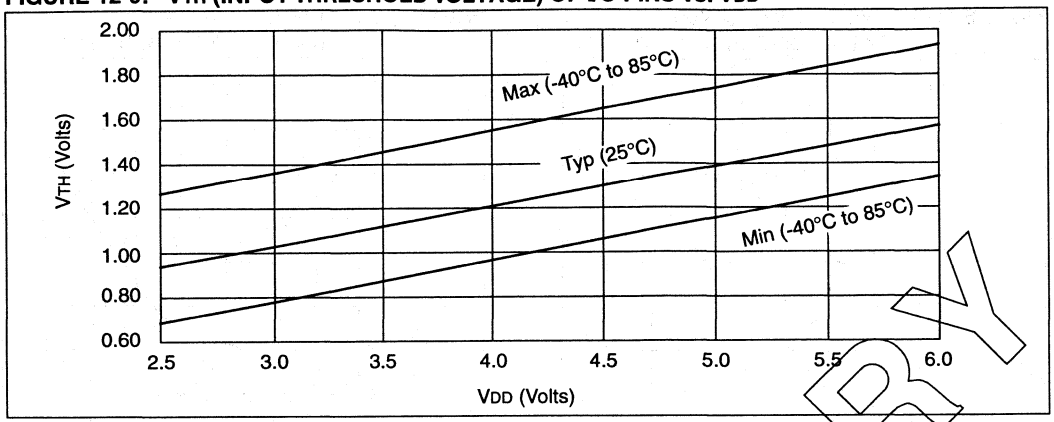
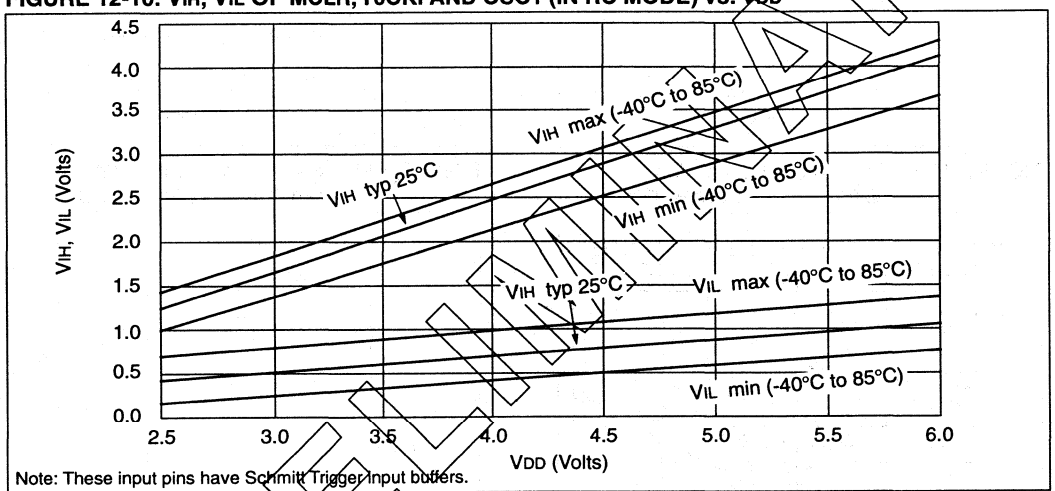
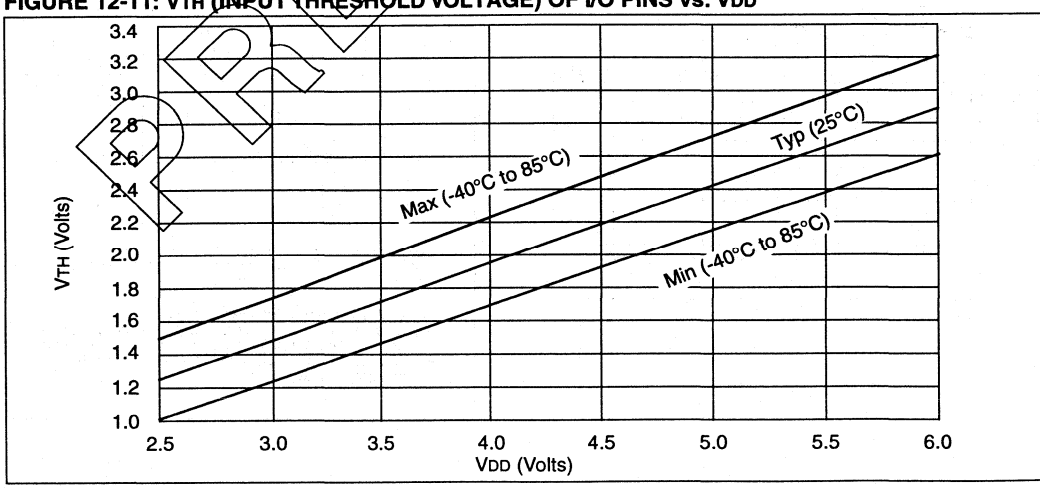


FIGURE 12-10: VIH, VIL OF MCLR, T0CKI AND OSC1 (IN RC MODE) vs. VDD



Note: These input pins have Schmitt Trigger input buffers.

FIGURE 12-11: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD



PRELIMINARY

 © 1995 Microchip Technology Inc.

FIGURE 12-12: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

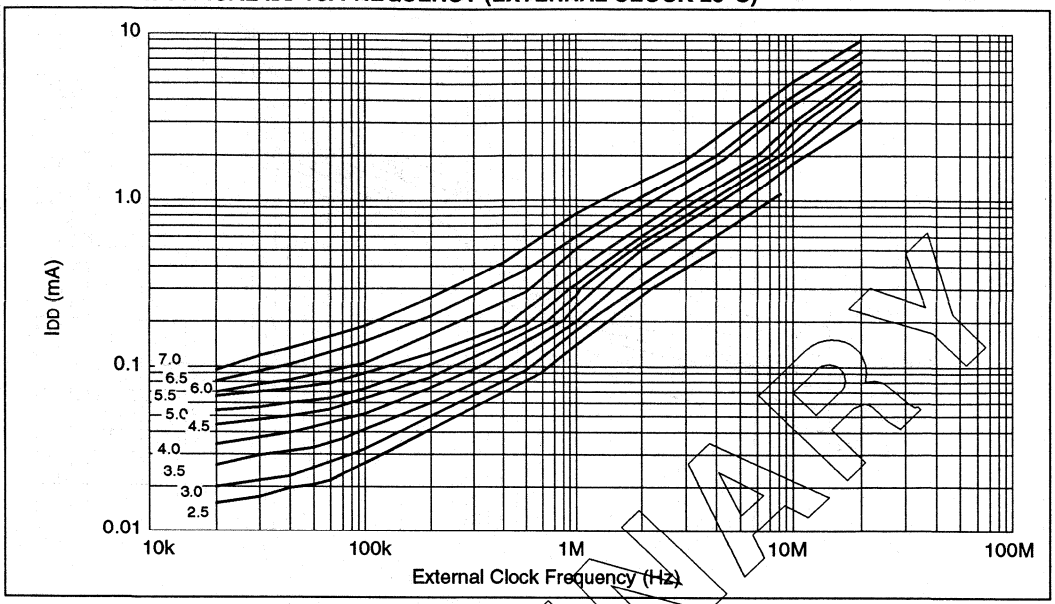
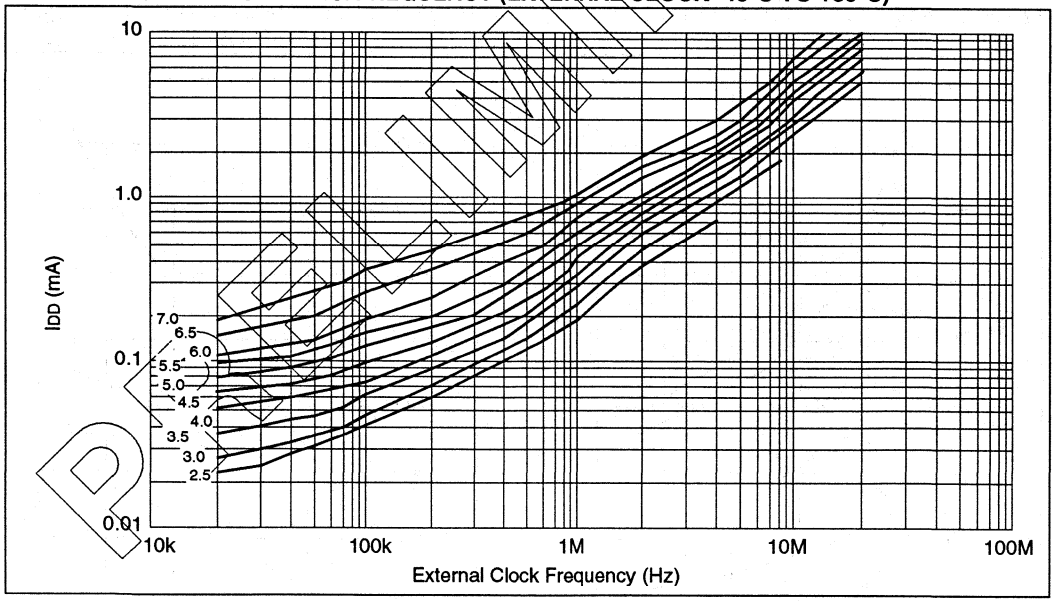


FIGURE 12-13: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -40°C TO +85°C)



PRELIMINARY

FIGURE 12-14: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK -55°C TO +125°C)

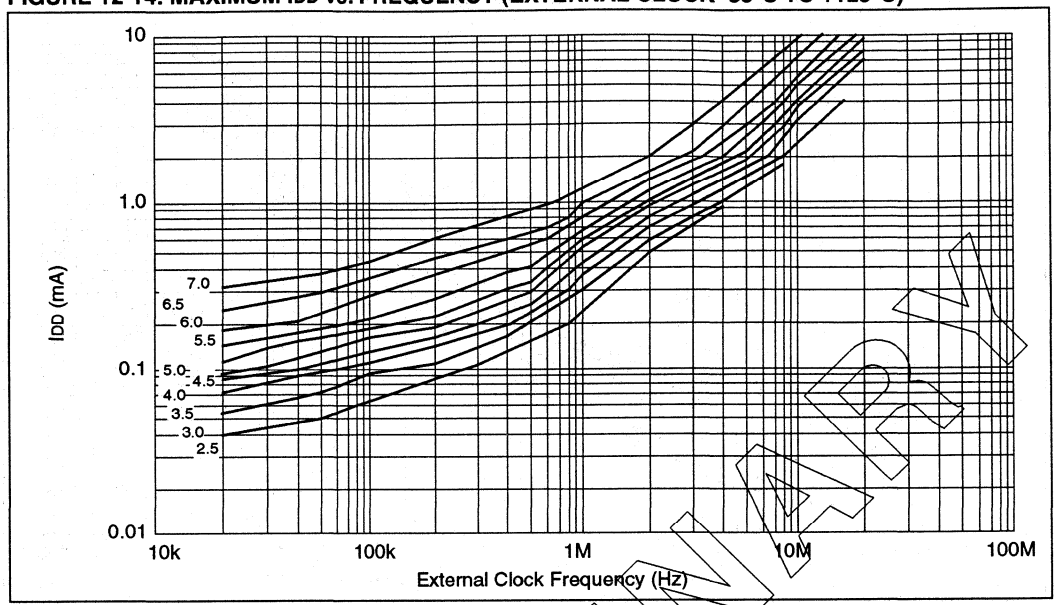


FIGURE 12-15: WDT TIMER TIME-OUT PERIOD vs. VDD

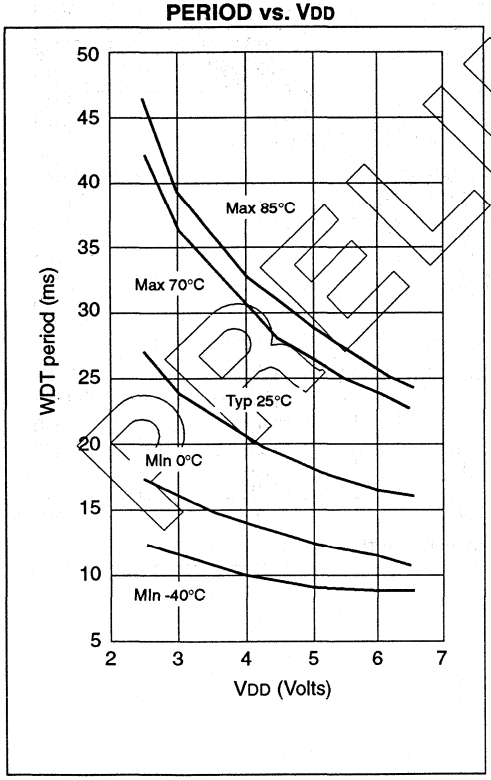
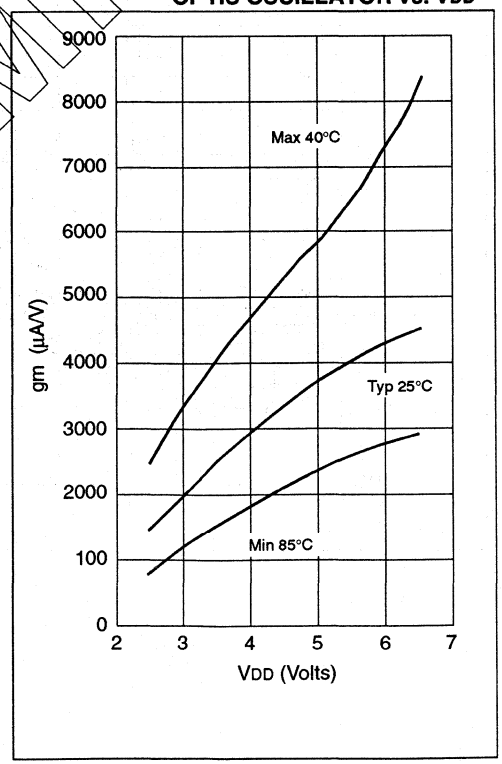


FIGURE 12-16: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD



PRELIMINARY

 © 1995 Microchip Technology Inc.

FIGURE 12-17: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. VDD

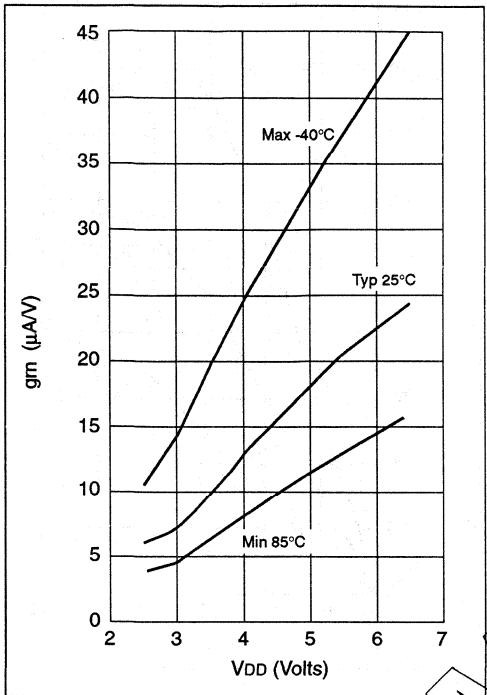


FIGURE 12-18: IOH vs. VOH, VDD = 3V

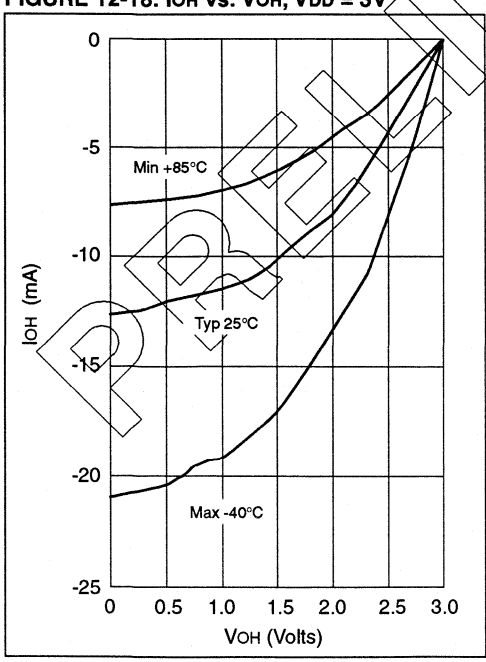


FIGURE 12-19: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

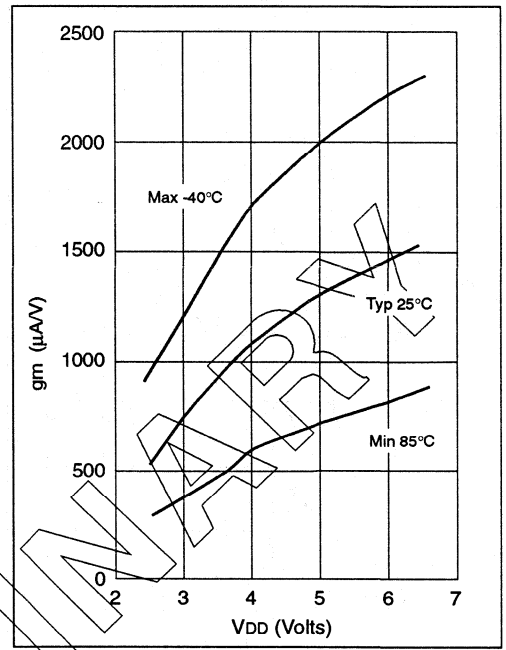
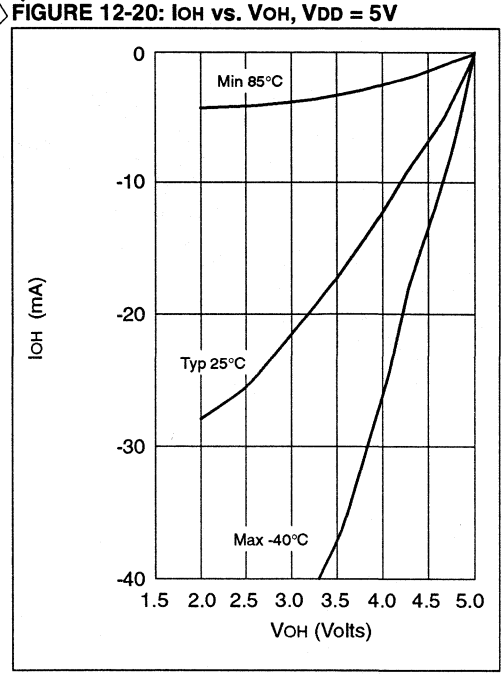


FIGURE 12-20: IOH vs. VOH, VDD = 5V



PRELIMINARY

FIGURE 12-21: IOL vs. VOL, VDD = 3V

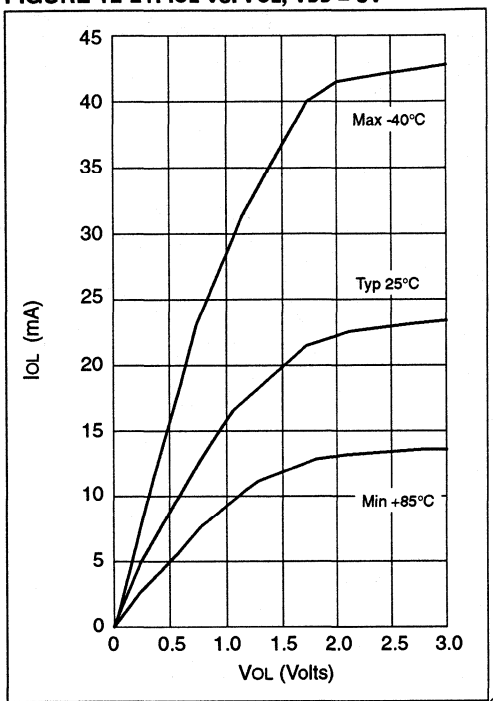
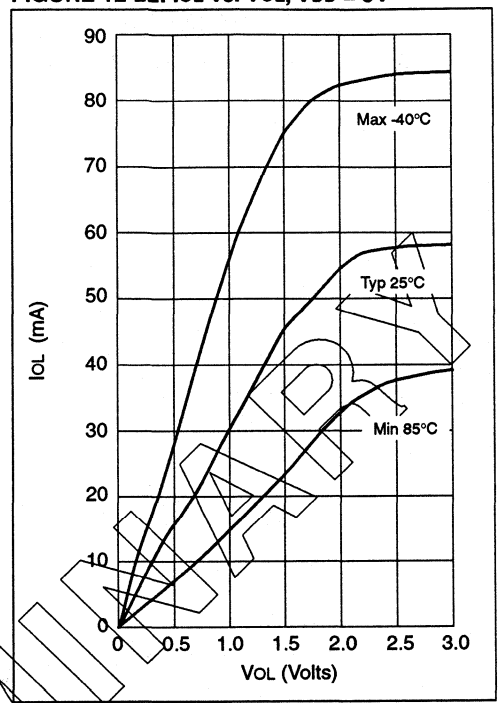


FIGURE 12-22: IOL vs. VOL, VDD = 5V



PRELIMINARY

 © 1995 Microchip Technology Inc.

13.0 DEVELOPMENT SUPPORT

13.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

13.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 13-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 (and better) machines in the Microsoft Windows™ 3.x environment. Thus, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC-Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 13-1.

FIGURE 13-1: PICMASTER SYSTEM CONFIGURATION

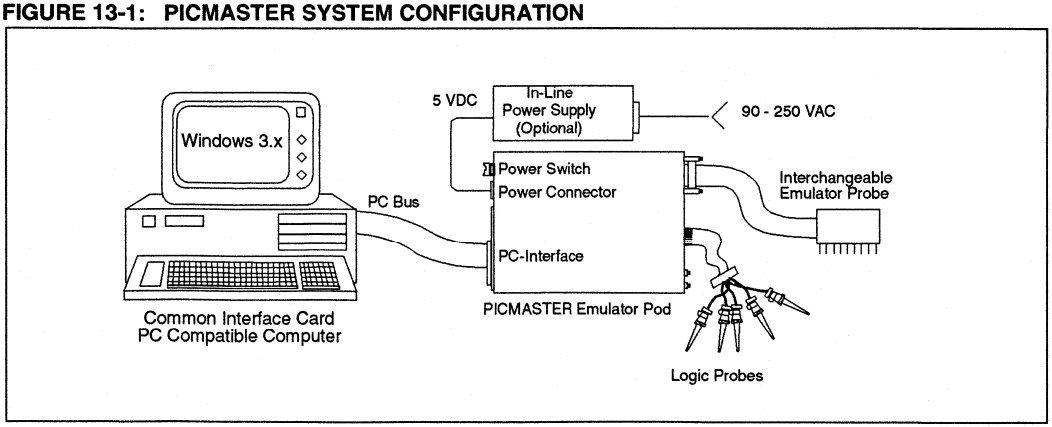


TABLE 13-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

13.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

13.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

13.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

13.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C63, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

© 1995 Microchip Technology Inc.

13.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically, i.e., by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and subtitles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

13.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

13.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

13.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP Edition, for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

13.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 13-2.

TABLE 13-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

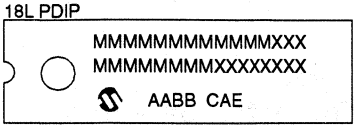
14.0 PACKAGING INFORMATION

For Package Dimension,
please refer to the Packaging Section of the Data Book.

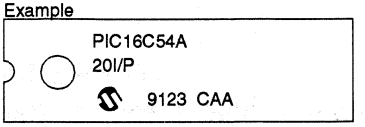
© 1995 Microchip Technology Inc.

14.1 Package Marking Information

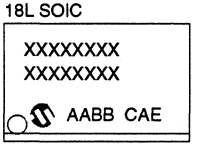
18L PDIP



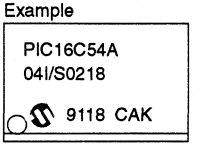
Example



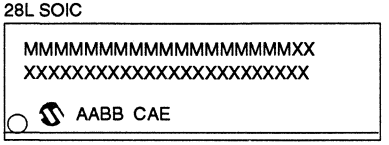
18L SOIC



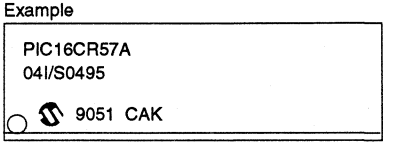
Example



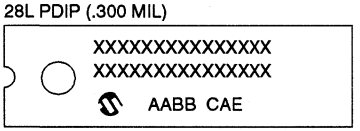
28L SOIC



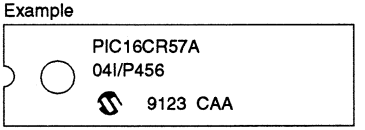
Example



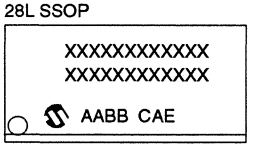
28L PDIP (.300 MIL)



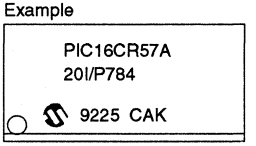
Example



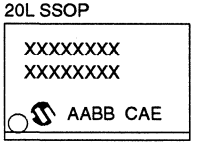
28L SSOP



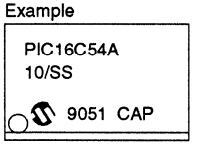
Example



20L SSOP



Example



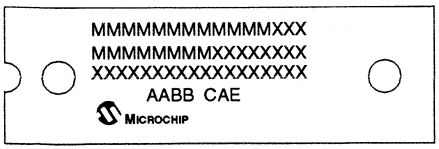
Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

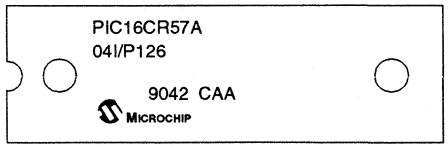
© 1995 Microchip Technology Inc.

14.2 Package Marking Information (Cont.)

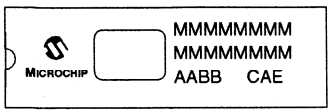
28L PDIP (.600 mil)

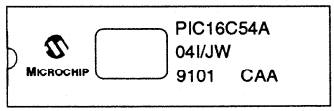


Example



18L CERDIP





Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

NOTES

© 1995 Microchip Technology Inc.

APPENDIX A: COMPATIBILITY

To convert code written for PIC16CXX to PIC16C5X, the user should take the following steps:

1. Check the CALL, GOTO instructions or any instruction that modifies the PC to determine if any program memory page select operations (PA2, PA1, PA0 bits) need to be made.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any special function register page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to proper value for processor used.
6. Remove any use of the ADDLW and SUBLW instructions.
7. Rewrite any code segments that use interrupts.

APPENDIX B: WHAT'S NEW

The format of this data sheet has been changed to be consistent with other product families. This ensures that important topics are covered across all PIC16/17 families. Here is an overview list of new features:

- Data Sheet Structure / Outline
- Addition of PIC16C58A

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This is so that control bits that do the same function have the same name (regardless of processor family). Care must still be taken, since they may not be in the same special function register. The following shows the register and bit names that have been changed:

REGISTER NAME CHANGES

OLD NAME	NEW NAME
RTCC	TMR0

BIT NAME CHANGES

OLD NAME	NEW NAME
RTS	T0CS
RTE	T0SE

PIN NAME CHANGES

OLD NAME	NEW NAME
RTCC	T0CKI

© 1995 Microchip Technology Inc.

APPENDIX D: PIC16/17 MICROCONTROLLERS

TABLE D-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory			Peripherals				Features			
	Maximum Frequency of Operation (MHz)	Program Memory	EPROM	RAM Data Memory (bytes)	Timer Module(s)	Captures	PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0,TMR1, TMR2,TMR3	2	2	Yes	1	1	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	1	1	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	1	1	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller modes.

3: PORTB has software-configurable weak pull-ups.

TABLE D-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory					Peripherals						Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
	Program Memory																
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE D-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals	Features		
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72		TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2

D.1 Pin Compatibility

Devices that have the same package type; and VDD, VSS, and $\overline{MCLR}$ pin locations, are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE D-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

INDEX

2

© 1995 Microchip Technology Inc.

W

Z

LIST OF EXAMPLES

LIST OF FIGURES

LIST OF TABLES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS**.

There is **no charge** for connecting to the BBS, except toll charge to CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks of Microchip Technology Incorporated.

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: Enhanced PIC16C5X Literature Number: DS30236A

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject? _____

5. What deletions from the data sheet could be made without affecting the overall usefullness? _____

6. Is there any incorrect or misleading information (what and where)? _____

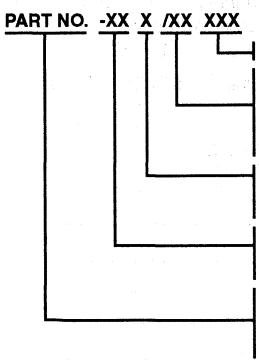
7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

© 1995 Microchip Technology Inc.

Enhanced PIC16C5X Product Identification System

To order or obtain information, e.g., on pricing or delivery refer to the factory or the listed sales office.



Pattern:	QTP, SQTP, ROM Code or Special Requirements	
Package:	P	= PDIP
	SO	= SOIC (Gull Wing, 300 mil body)
	SS	= SSOP
	JW	= Windowed CERDIP
	SP	= Skinny SDIP (300 mil)
Temperature Range:	-	= 0°C to +70°C
	I	= -40°C to +85°C
	E	= -40°C to +125°C (PIC16C54A only)
Frequency Range:	04	= 4 MHz
	10	= 10 MHz
	20	= 20 MHz
Device	PIC16C5XAT: :Standard VDD range (Tape/Reel)	
	PIC16LC5XA :Extended VDD range	
	PIC16CR5XA :ROM Version, Standard VDD range	
	PIC16LCR5XA:ROM Version, Standard VDD range	

Examples

a) PIC16C54A - 104/p 301: Commercial Temp., PDIP Package, 4 MHz, normal VDD limits, QTP pattern #301.

b) PIC16LC58A - 04I/SO Industrial Temp., SOIC package, 4 MHz, extended VDD limits

c) PIC16CR57A - 10E/P35k: ROM program memory, Automotive Temp., PDIP package, 10 MHz, normal VDD limits.

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.
For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

PIC16C62X

EPROM-Based 8-Bit CMOS Microcontroller

Devices included in this data sheet are:

- PIC16C620
- PIC16C621
- PIC16C622

FEATURES

High Performance RISC-like CPU

- Only 35 instructions to learn
- All single cycle instructions (200ns), except for program branches which are two-cycle
- Operating speed:
  - DC - 20MHz clock input
  - DC - 200ns instruction cycle

Device	Program Memory	Data Memory
PIC16C620	512	80
PIC16C621	1K	80
PIC16C622	2K	128

- Interrupt capability
- 16 special function hardware registers
- 8 level deep hardware stack
- Direct, Indirect and relative addressing modes

Peripheral Features

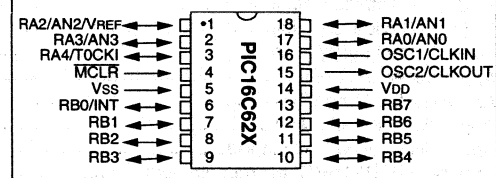
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - 2 analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs can be output signals
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features

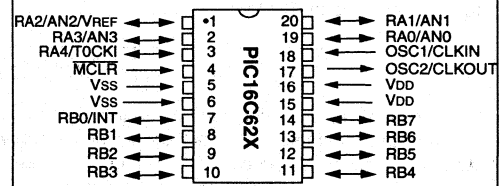
- Power-on-reset (POR)
- Power-up Timer (PWRT) and Oscillator start-up timer (OST)
- Brown-out protection
- Watchdog timer (WDT) with its own on-chip RC oscillator for reliable operation

PACKAGE TYPE

PDIP, SOIC, Windowed CERDIP



SSOP



Special Microcontroller Features (Continued)

- Programmable code protection
- Power saving SLEEP mode
- Fuse selectable oscillator options
- Serial in-system programming (via two pins)
- 4 user programmable ID locations

CMOS Technology

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide operating voltage range
  - 3.0V to 6.0 V
- Commercial and industrial temperature range
- Low power consumption
  - < 2 mA @ 5V, 4 MHz
  - 15 µA typical @ 3V, 32 KHz
  - < 1 µA typical standby current @ 3V

TABLE OF CONTENTS

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

 © 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The PIC16C62X are 18-Pin EPROM-based members of the versatile PIC16CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC-like architecture. The PIC16C62X have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C62X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in its class.

The PIC16C620 and PIC16C621 have 80 bytes of RAM. The PIC16C622 has 128 bytes of RAM. Each device has 13 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler. In addition, the PIC16C62X add two analog comparators with a programmable on-chip voltage reference module. The comparator module is ideally suited for applications requiring a low-cost analog interface, e.g. battery chargers, threshold detectors, white goods controllers etc.

The PIC16C62X have special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake up the chip from SLEEP through several external and internal interrupts and reset.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lockup.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

Table 1-1 shows the features of the PIC16C620, PIC16C621 and PIC16C622 as well as the other members of the PIC16CXX mid-range microcontroller family.

A simplified block diagram of the PIC16C62X is shown in Figure 3-1.

The PIC16C62X series fit perfectly in applications ranging from battery chargers to low-power remote sensors. The EPROM technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C62X very versatile.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to PIC16C62X family of devices (see Appendix B).

1.2 Development Support

The PIC16C62X family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals							Features				
	Maximum Frequency of Operation (MHz)	Program Memory EPROM	Program Memory EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

2.0 PIC16C62X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements the proper device option can be selected using the information in the PIC16C62X Product Identification System section at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the oscillator modes etc. Microchip's PICSTART® and PRO MATE™ programmers both support programming of the PIC16C62X.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages permit the user to program them once. In addition to the program memory, the configuration bits must be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who chose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround-Production (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

© 1995 Microchip Technology Inc.

NOTES:

 © 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C62X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C62X uses a Harvard architecture, in which, program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. Instruction op-codes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (see Example 3-1). Consequently, all instructions (35) execute in a single cycle (200ns @ 20MHz) except for program branches.

The PIC16C620 addresses 512 x 14 on-chip program memory, all on chip. The PIC16C621 addresses 1K x 14 program memory, all on-chip. The PIC16C622 addresses 2K x 14 program memory, also on-chip. Program execution is in internal memory only for all devices.

The PIC16C62X can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16C62X have an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16C62X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C62X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

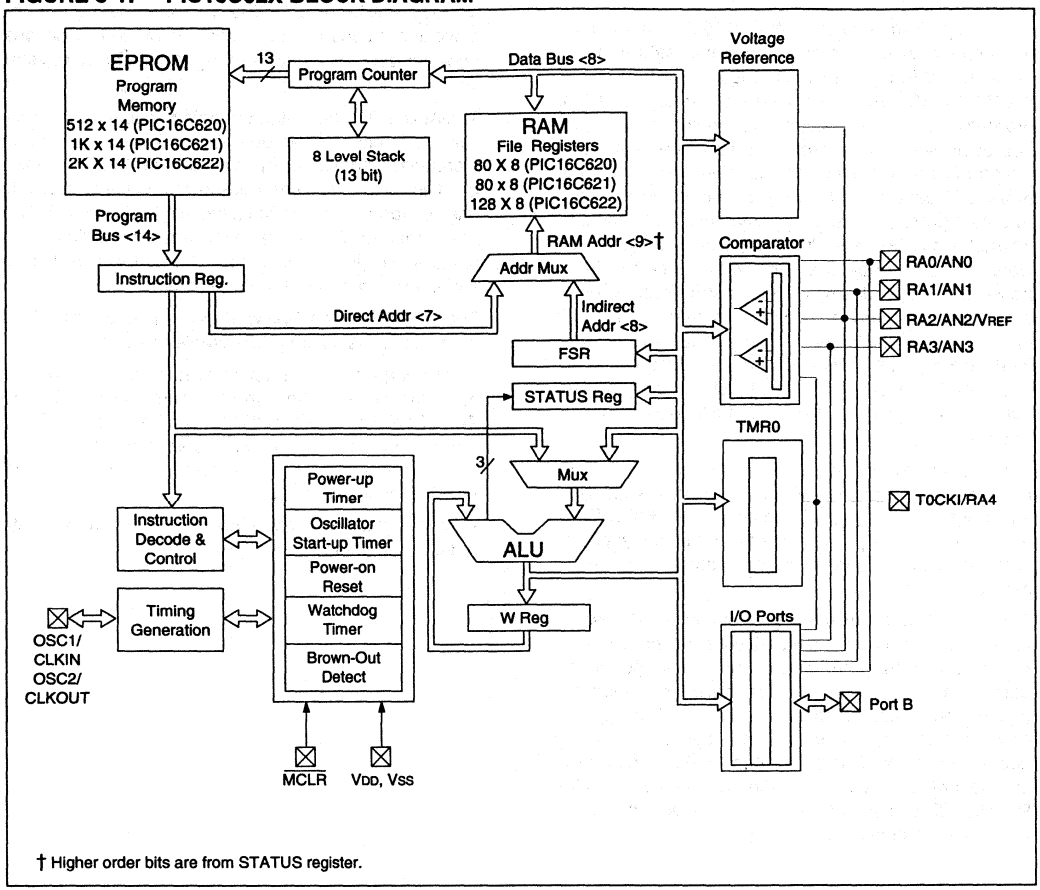
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, bit in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram is shown in Figure 3-1, with a description of the device pins in Table 3-1.

© 1995 Microchip Technology Inc.

FIGURE 3-1: PIC16C62X BLOCK DIAGRAM



† Higher order bits are from STATUS register.

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C62X PINOUT DESCRIPTION

Name	DIP SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	18	I	ST/CMOS	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	17	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bidirectional I/O port.
RA0/AN0	17	19	I/O	ST	Analog comparator input
RA1/AN1	18	20	I/O	ST	Analog comparator input
RA2/AN2/VREF	1	1	I/O	ST	Analog comparator input or VREF output
RA3/AN3	2	2	I/O	ST	Analog comparator input /output
RA4/T0CKI	3	3	I/O	ST	Can be selected to be the clock input to the TMR0 timer/counter or a comparator output. Output is open drain type.
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	7	I/O	ST	RB0/INT can also be selected as an external interrupt pin.
RB1	7	8	I/O	ST	
RB2	8	9	I/O	ST	
RB3	9	10	I/O	ST	
RB4	10	11	I/O	ST	Interrupt on change pin.
RB5	11	12	I/O	ST	Interrupt on change pin.
RB6	12	13	I/O	ST	Interrupt on change pin. Serial programming clock.
RB7	13	14	I/O	ST	Interrupt on change pin. Serial programming data.
VSS	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: O = output I/O = input/output P = power
— = Not used I = Input ST = Schmitt trigger input

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

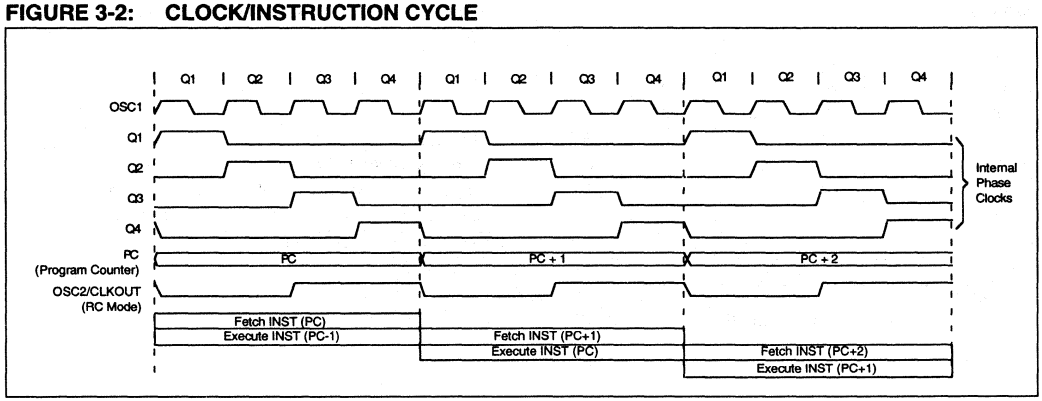
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-1).

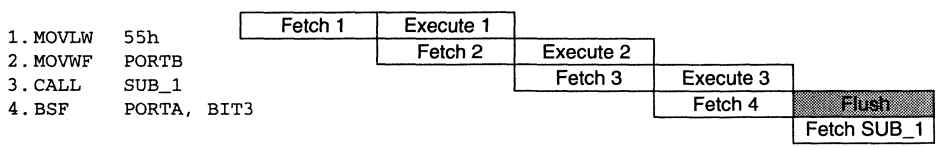
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetched instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

 © 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C62X has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16C620, 1K x 14 (0000h - 03FFh) for the PIC16C621 and the first 2K x 14 (0000h - 07FFh) for the PIC16C622 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 space (PIC16C620) or 1K x 14 space (PIC16C621) or 2K x 14 space (PIC16C622). The reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 4-1, Figure 4-2, Figure 4-3).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C620

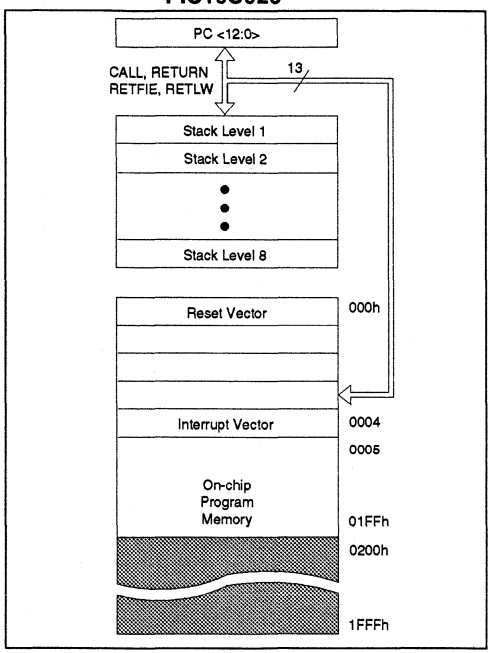


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C621

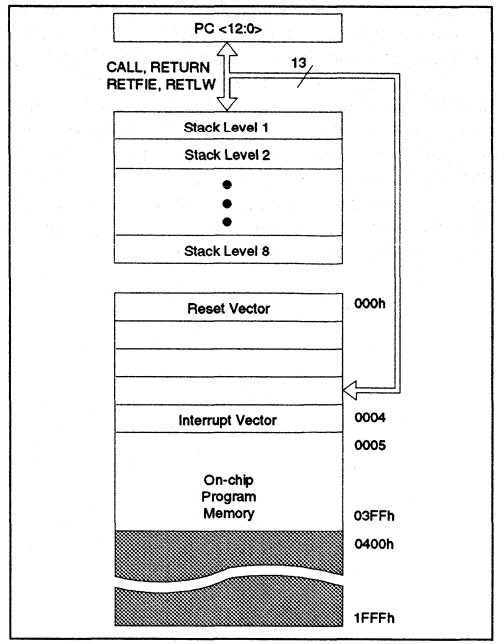
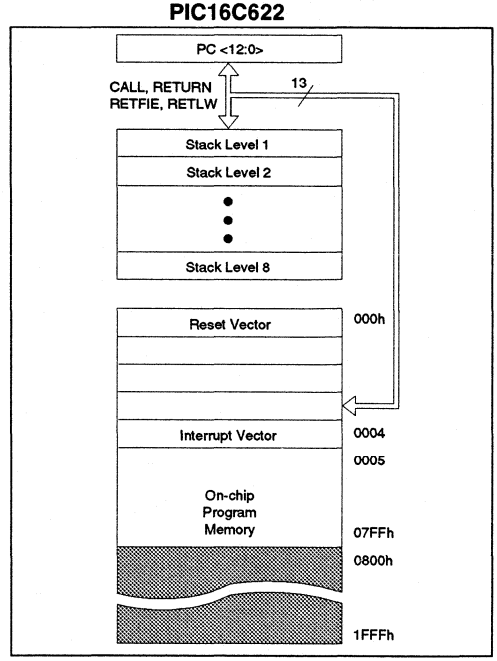


FIGURE 4-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16C622



4.2 Data Memory Organization

The data memory (see Figure 4-4 and Figure 4-5) is partitioned into two Banks which contain the general purpose registers and the special function registers. Bank 0 is selected when the RP0 bit in the STATUS register is cleared. Bank 1 is selected when the RP0 bit in the STATUS register is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations 20h-6Fh (Bank 0) on the PIC16C620/621 and 20h-7Fh (Bank 0) and A0h-BFh (Bank 1) on the PIC16C622 are general purpose registers implemented as static RAM. Some special purpose registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 80 x 8 in the PIC16C620/621 and 128 x 8 in the PIC16C622. Each is accessed either directly or indirectly through the file select register FSR (see Section 4.4).

 © 1995 Microchip Technology Inc.

FIGURE 4-4: DATA MEMORY MAP FOR THE PIC16C620/621

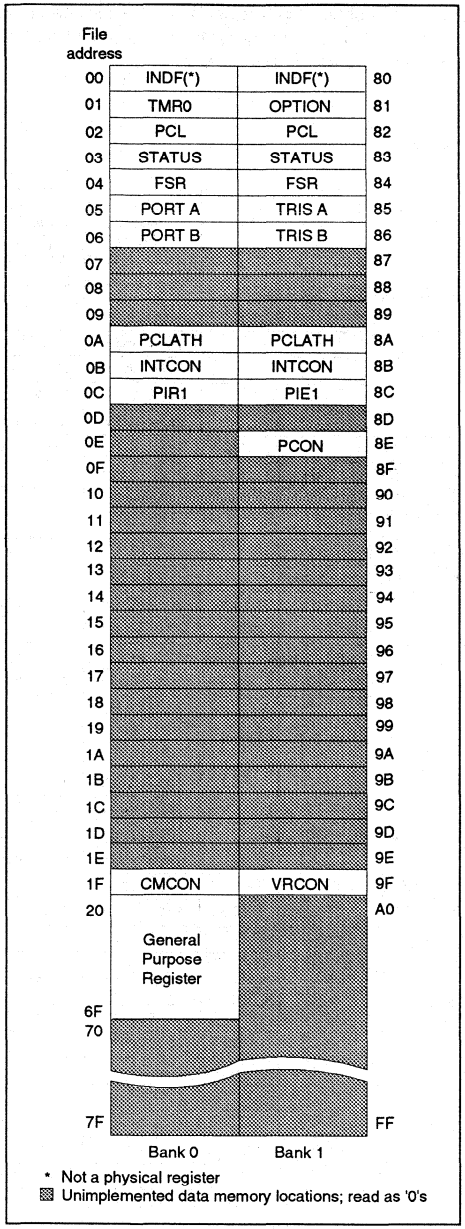
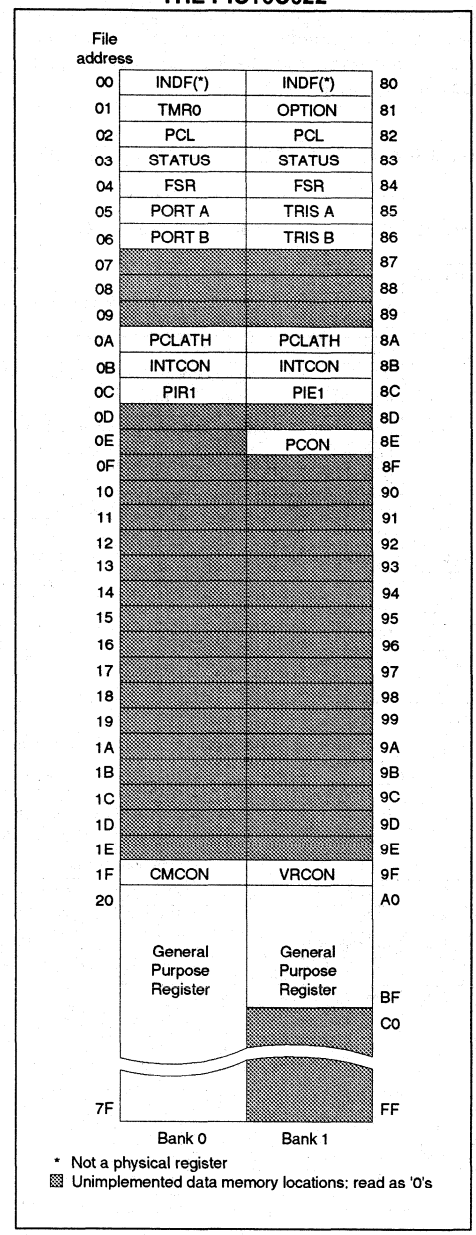


FIGURE 4-5: DATA MEMORY MAP FOR THE PIC16C622



2

4.2.2 SPECIAL FUNCTION REGISTERS

The special function registers are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (see Table 4-1). These registers are static RAM.

The special registers can be classified into two sets. The special registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C62X

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 1)
Bank 0											
00h	INDF (indirect address)	Addressing this location uses contents of FSR to address data memory (not a physical register)								—	—
01h	TMR0	Timer0								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	Unimplemented									—	—
08h	Unimplemented									—	—
09h	Unimplemented									—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	-0-- ----	-0-- ----
0Dh-1Eh	Unimplemented									—	—
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
Bank 1											
80h	INDF (indirect address)	Addressing this location uses contents of FSR to address data memory (not a physical register)								—	—
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
87h	Unimplemented									—	—
88h	Unimplemented									—	—
89h	Unimplemented									—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	-0-- ----	-0-- ----
8Dh	Unimplemented									—	—
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BO}$	---- --0x	---- --u?
8Fh-9Eh	Unimplemented									—	—
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

Legend: — = Unimplemented locations. Read as '0', u = unchanged, x = unknown, ? = value depends on condition

Note 1: Other (non power-up) resets include $\overline{MCLR}$ reset, Brown-out Detect reset and Watchdog Timer time-out during normal operation.

 © 1995 Microchip Technology Inc.

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 4-6, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

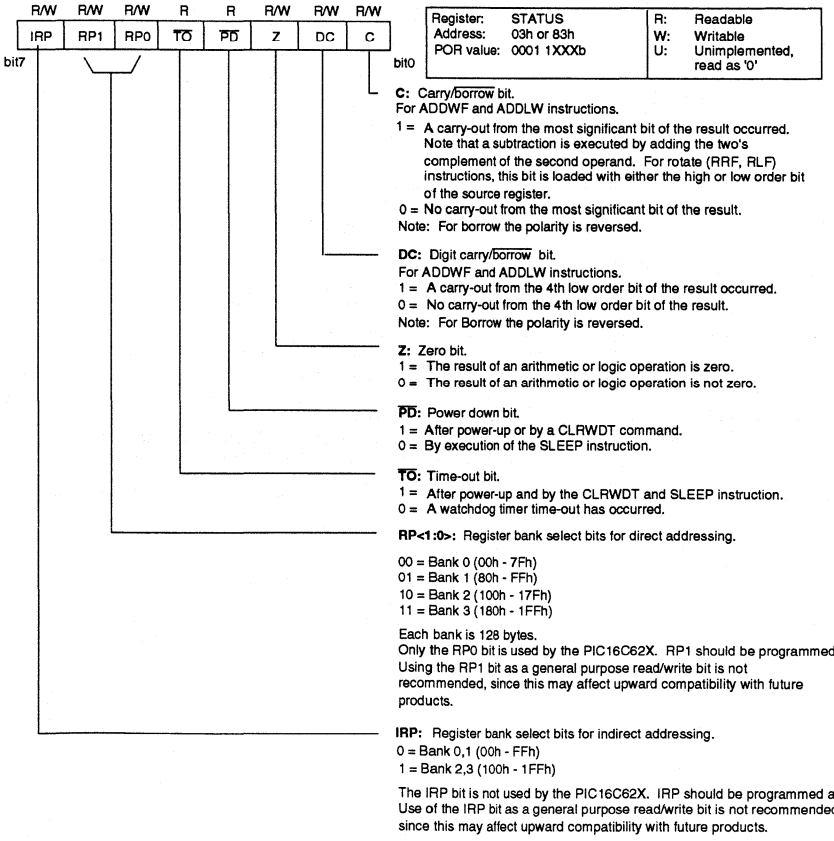
For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the status register as 000UU1UU (where U = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the status registers because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the "Instruction Set Summary".

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C62X and should be programmed as '0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

FIGURE 4-6: STATUS REGISTER



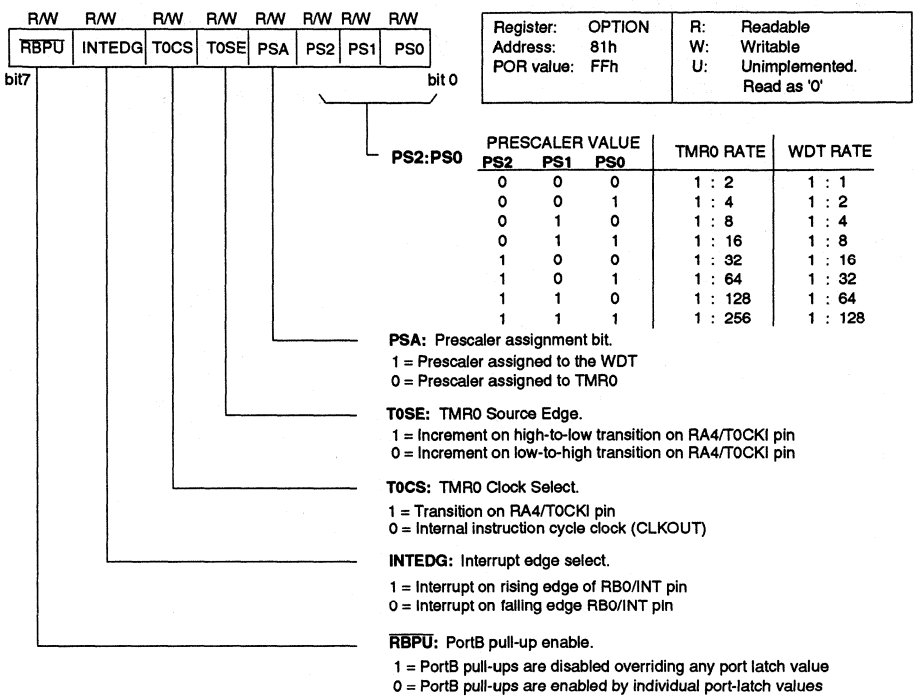
© 1995 Microchip Technology Inc.

4.2.2.2 OPTION REGISTER

The OPTION register, shown in Figure 4-7, is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note 1: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1).

FIGURE 4-7: OPTION REGISTER



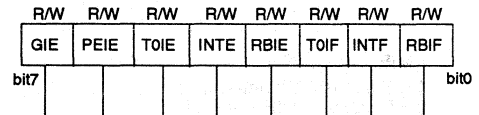
 © 1995 Microchip Technology Inc.

4.2.2.3 INTCON REGISTER

The INTCON register, shown in Figure 4-8, is a readable and writable register which contains the various enable and flag bits for all interrupt sources except the comparator module. See Section 4.2.2.4 and Section 4.2.2.5 for a description of the comparator enable and flag bits.

Note: The T0IF, INTF, or RBIF will be set by the specified condition even if the corresponding Interrupt Enable Bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled).

FIGURE 4-8: INTCON REGISTER



Register:	INTCON	R:	Readable
Address:	0Bh or 8Bh	W:	Writable
POR value:	0000 000xb	U:	Unimplemented, read as '0'

RBIF: RB port change interrupt flag.
1 = When at least one of the RB<7:4> inputs change. Must be cleared by software.
0 = None of the RB<7:4> inputs have changed.

INTF: External interrupt flag.
1 = The external INT interrupt has occurred. Must be cleared by software.
0 = The INT interrupt did not occur

T0IF: TMR0 overflow interrupt flag.
1 = The TMR0 has overflowed. Must be cleared by software.
0 = TMR0 did not overflow.

RBIE: RB port change interrupt enable bit.
1 = Enables RBIF interrupt
0 = Disables RBIF interrupt

INTE: INT interrupt enable bit.
1 = Enables INTF interrupt
0 = Disables INTF interrupt

T0IE: T0IF interrupt enable bit.
1 = Enables T0IF interrupt
0 = Disables T0IF interrupt

PEIE: Peripheral interrupt enable bit.
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

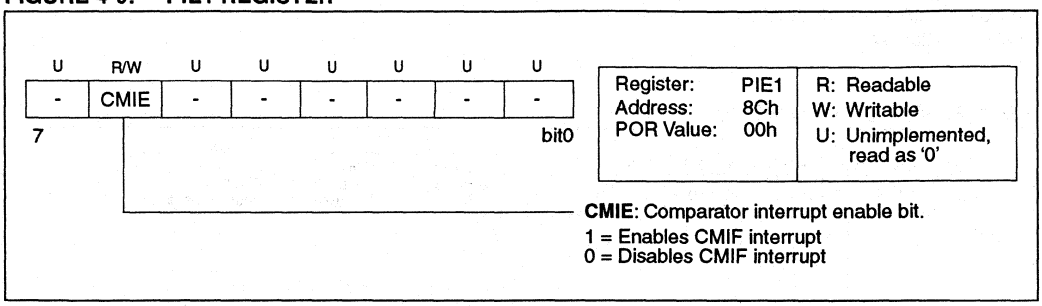
GIE: Global interrupt enable.
1 = Enables all un-masked interrupts
0 = Disables all interrupts

© 1995 Microchip Technology Inc.

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bit for the comparator interrupt.

FIGURE 4-9: PIE1 REGISTER

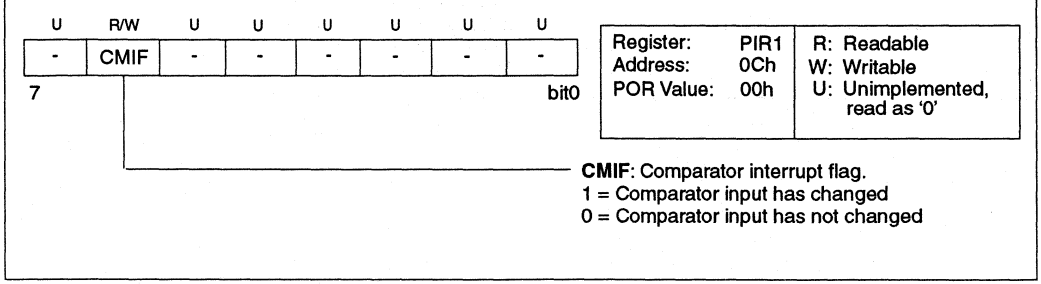


4.2.2.5 PIR1 REGISTER

This register contains the individual flag bit for the comparator interrupt.

Note: This bit will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

FIGURE 4-10: PIR1 REGISTER



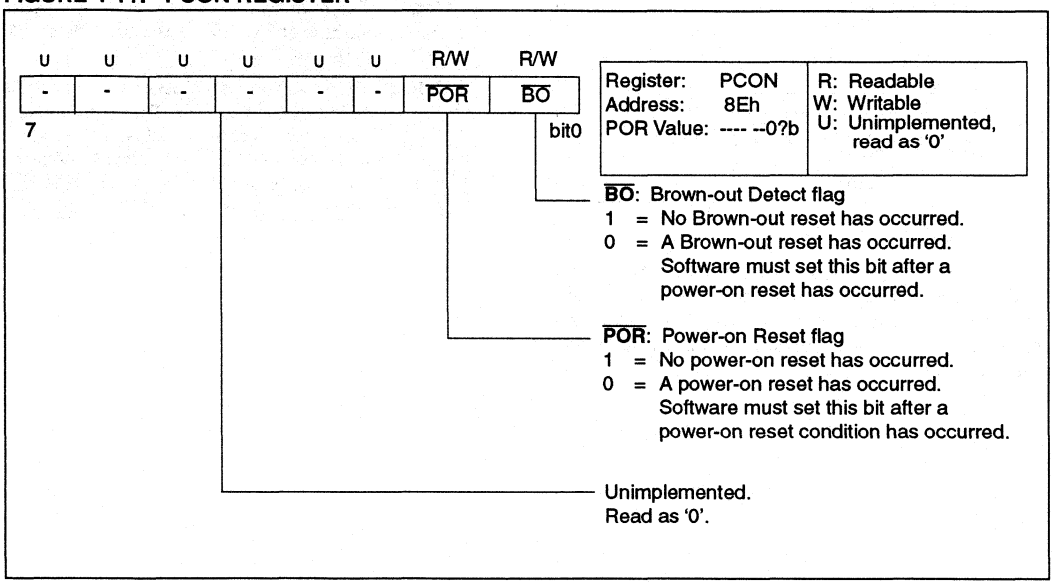
 © 1995 Microchip Technology Inc.

4.2.2.6 PCON REGISTER

The PCON register, shown in Figure 4-11, contains flag bits to differentiate between a Power-on Reset, an external $\overline{MCLR}$ reset, WDT reset or a Brown-out Detect reset.

Note: $\overline{BO}$ is unknown on power-on reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BO}$ is cleared, indicating a brown-out has occurred. The $\overline{BO}$ status bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled (by programming BODEN bit in the Configuration word).

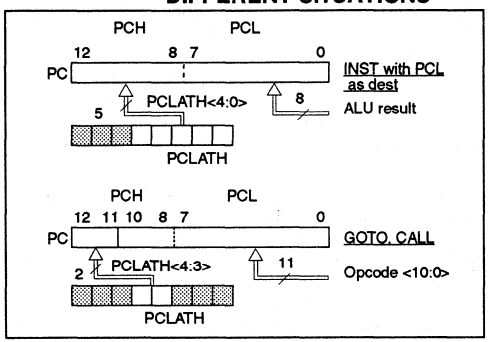
FIGURE 4-11: PCON REGISTER



4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte, PCL, is a readable and writable register. The high byte of the PC (PCH) is not directly readable or writable. The high byte of the PC can be written through the PCLATH register. When the PC is loaded with a new value during a CALL, GOTO or a write to PCL, the high bits of PC are loaded from PCLATH as shown in Figure 4-12.

FIGURE 4-12: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note "Table Read Using the PIC16CXX" (AN556).

4.3.2 STACK

The PIC16CXX has an 8 deep x 13-bit wide hardware stack (see Figure 4-2 and Figure 4-3). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed in the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is POPped in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer. This means that after the stack has been "PUSHed" eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflows or stack underflow conditions.

Note 2: There are no instructions mnemonics called PUSH nor POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address.

 © 1995 Microchip Technology Inc.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-13. However, IRP is not used in the PIC16C62X.

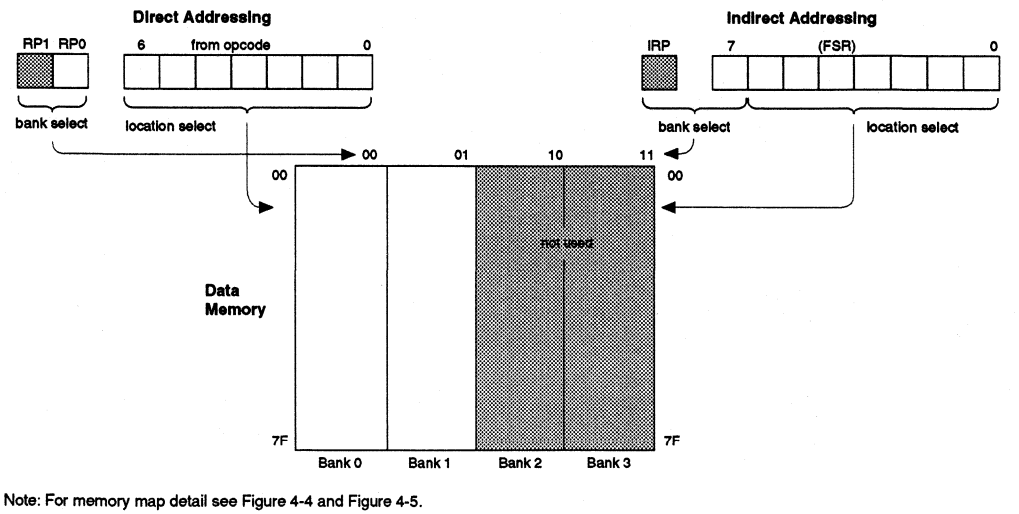
A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
        movlw   0x20    ;initialize pointer
        movf    FSR     ;to RAM
NEXT    clrf    INDF    ;clear INDF register
        incf    FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;no clear next
                        ;yes continue
CONTINUE:
```

2

FIGURE 4-13: DIRECT/INDIRECT ADDRESSING PIC16C62X



Note: For memory map detail see Figure 4-4 and Figure 4-5.

NOTES:

 © 1995 Microchip Technology Inc.

5.0 I/O PORTS

The PIC16C62X have two ports, PORTA and PORTB. These port pins may be multiplexed with an alternate function for the peripheral features on the device.

5.1 PORTA and TRISA Registers

PORTA is a 5-bit wide latch. RA4 is a Schmitt trigger input and an open drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

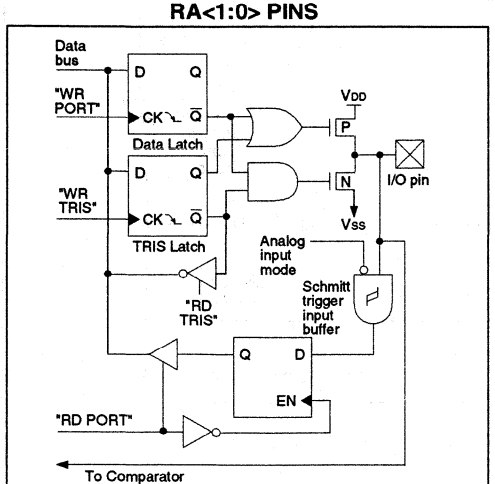
A '1' in the TRISA register puts the corresponding output driver in a high impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (comparator control register) register and the VRCON (voltage reference control register) register. When selected as a comparator input, these pins will read as '0's.

Note: On reset, the TRISA register is set to all inputs. The digital inputs are disabled and the comparator inputs are forced to ground to reduce excess current consumption.

FIGURE 5-1: BLOCK DIAGRAM OF RA<1:0> PINS



Note: I/O pins have protection diodes to VDD and VSS.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

The RA2 pin will also function as the output for the voltage reference. When in this mode, the Vref pin is a very high impedance output. The user must configure TRISA<2> bit as an input and use high impedance loads.

In one of the comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be set to enable outputs to use this function.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF   PORTA        ;Initialize PORTA by setting
                    ;output data latches
MOVLW  0X07         ;Turn comparators off and
MOVWF  CMCON        ;enable pins for I/O
                    ;functions
BSF    STATUS, RP0  ;Select Bank1
MOVLW  0x1F         ;Value used to initialize
                    ;data direction
MOVWF  TRISA        ;Set RA<4:0> as inputs
                    ;TRISA<7:5> are always
                    ;read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA2 PIN

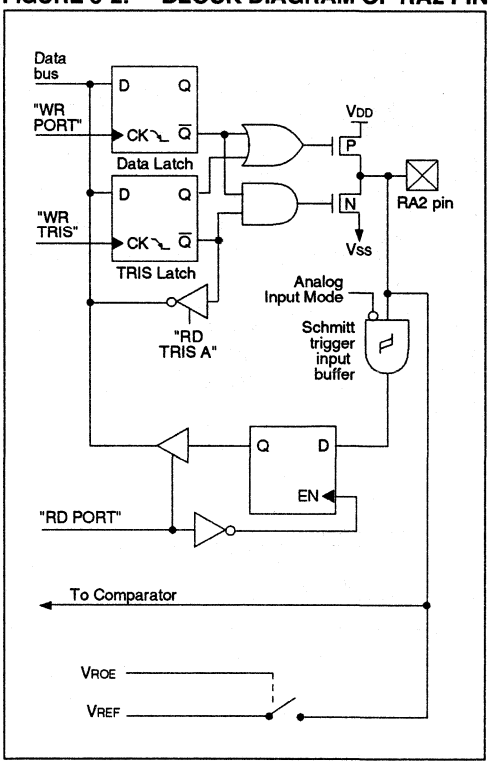


FIGURE 5-3: BLOCK DIAGRAM OF RA3 PIN

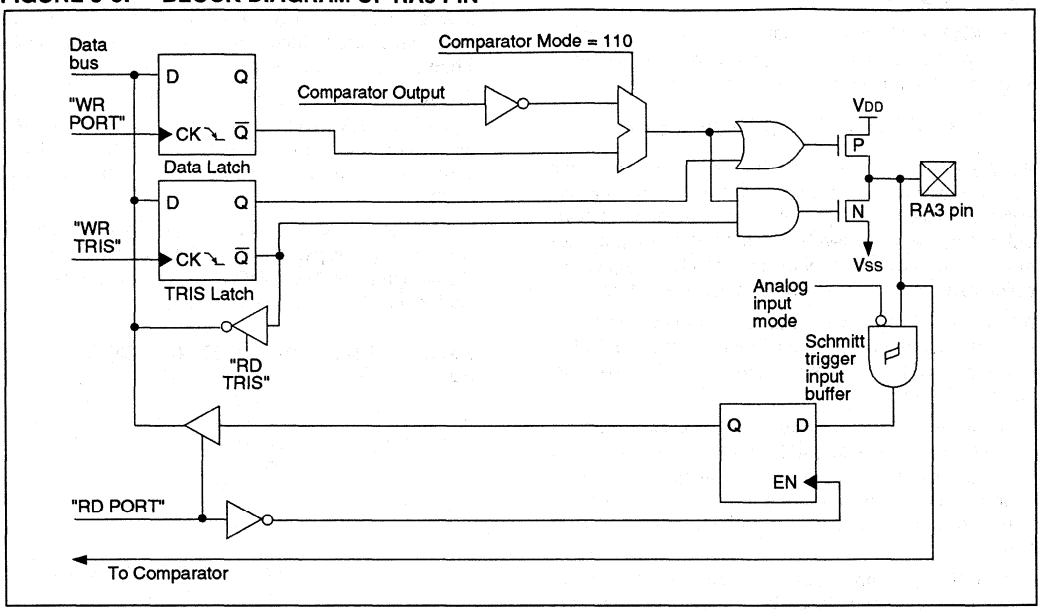
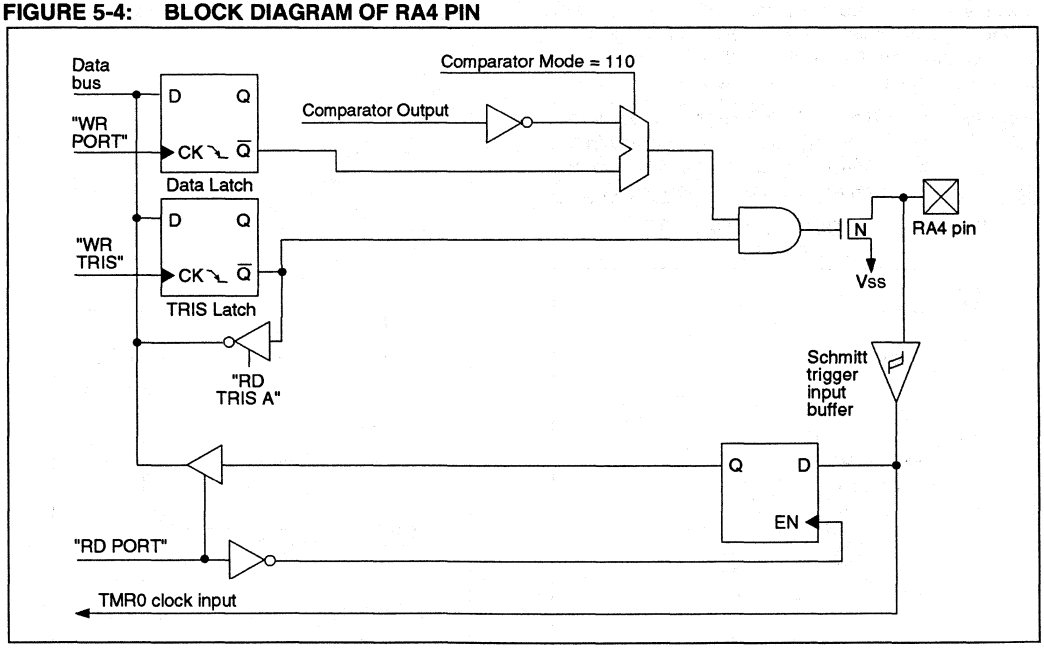


FIGURE 5-4: BLOCK DIAGRAM OF RA4 PIN



 © 1995 Microchip Technology Inc.

TABLE 5-1: PORTA FUNCTIONS

Name	Bit	Buffer Type	Function
RA0/AN0	bit0	ST	Input/output or comparator input
RA1/AN1	bit1	ST	Input/output or comparator input
RA2/AN2/VREF	bit2	ST	Input/output or comparator input or VREF output
RA3/AN3	bit3	ST	Input/output or comparator input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0 or comparator output. Output is open drain type.

Legend: ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF PORTA REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTA	PORTA pins when read. PORTA latch when written.	05h	---x xxxx
TRISA	PORTA data direction register. 0 = output, 1 = input	85h	---1 1111
CMCON	Comparator control register configures PORTA pins.	1Fh	00-- 0000
VRCON	Voltage Reference control register configures VREF output on RA2.	9Fh	00-0 0000

Note 1: X = unknown, - = unimplemented, reads as '0'.

Note 2: For reset values of registers in other reset situations refer to Table 9-6.

5.2 PORTB and TRISB Registers

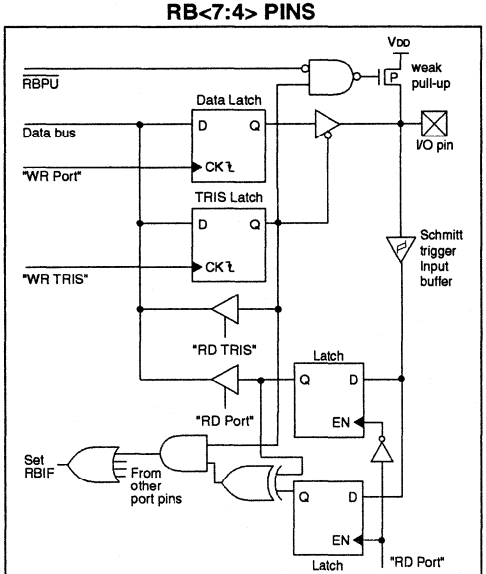
PORTB is an 8-bit wide bidirectional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a high impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up (~200 µA typical). A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on power-on reset.

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7–RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7–RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7–RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

FIGURE 5-5: BLOCK DIAGRAM OF RB<7:4> PINS



Note 1: TRISB = 1 enables weak pull-up if $\overline{\text{RPBU}}$ = 0 in OPTION register
2: I/O pins have diode protection to VDD and Vss

This interrupt can wake the device up from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

a) Disable the interrupt by clearing RBIE (INTCON<3>) bit.
b) Read PORTB. This will end mismatch condition. Then, clear the RBIF bit.

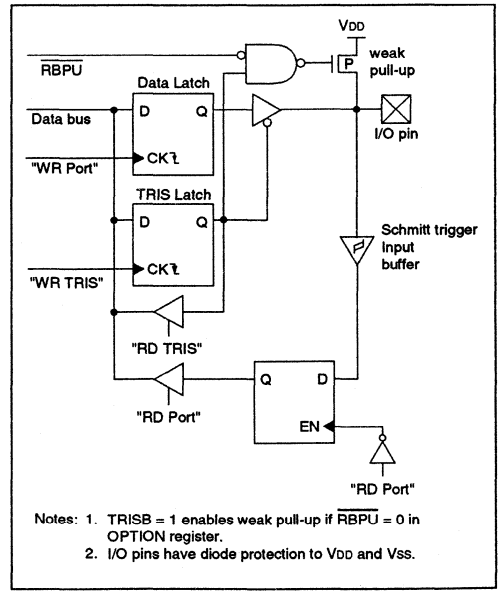
A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the Microchip *Embedded Control Handbook*).

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-6: BLOCK DIAGRAM OF RB<3:0> PINS



Notes: 1. TRISB = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = 0 in OPTION register.
2. I/O pins have diode protection to VDD and Vss.

 © 1995 Microchip Technology Inc.

TABLE 5-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/INT	bit0	ST	Input/output or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	ST	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	ST	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	ST	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock pin.
RB7	bit7	ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data pin.

Legend: ST = Schmitt Trigger

TABLE 5-4: SUMMARY OF PORTB REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTB	PORTB pins when read PORTB latch when written	06h	xxxx xxxx
TRISB	PORTB data direction register 0 = output, 1 = input	86h	1111 1111
$\overline{\text{OPTION}}$	Weak pull-up on/off control ($\overline{\text{RBPU}}$ bit)	81h	1111 1111

Legend: x = unknown

Note: For reset values of registers in other reset situations refer to Table 9-6.

5.3 I/O Programming Considerations

5.3.1 BIDIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit 5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit 5 and PORTB is written to the output latches. If another bit of PORTB is used as a bidirectional I/O pin (e.g., bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit 0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading the PORT register, reads the values of the PORT pins. Writing to the PORT register writes the value to the PORT latch. When using read modify write instructions (ex. BCF, BSF, etc.) on a PORT, the value of the PORT pins is read, the desired operation is done to this value, and this value is then written to the PORT latch.

Example 5-2 shows the effect of two sequential read modify write instructions (ex. BCF, BSF, etc.) on an I/O PORT.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

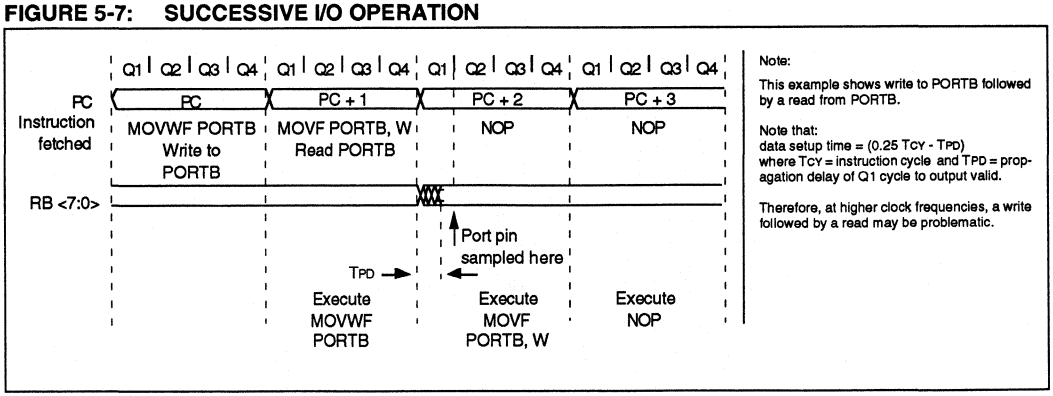
EXAMPLE 5-2: READ MODIFY WRITE INSTRUCTIONS ON AN I/O PORT

```
; Initial PORT settings:  PORTB<7:4> Inputs
;
;                         PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are not
; connected to other circuitry
;
;                         PORT latch  PORT pins
;                         ----------  ----------

    BCF PORTB, 7        ; 01pp pppp   11pp pppp
    BCF PORTB, 6        ; 10pp pppp   11pp pppp
    BSF STATUS,RP0      ;
    BCF TRISB, 7        ; 10pp pppp   11pp pppp
    BCF TRISB, 6        ; 10pp pppp   10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (see Figure 5-7). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-7: SUCCESSIVE I/O OPERATION



 © 1995 Microchip Technology Inc.

6.0 TIMER0 (TMR0) MODULE

The TMR0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the TMR0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two cycles (see Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 module.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the T0 source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

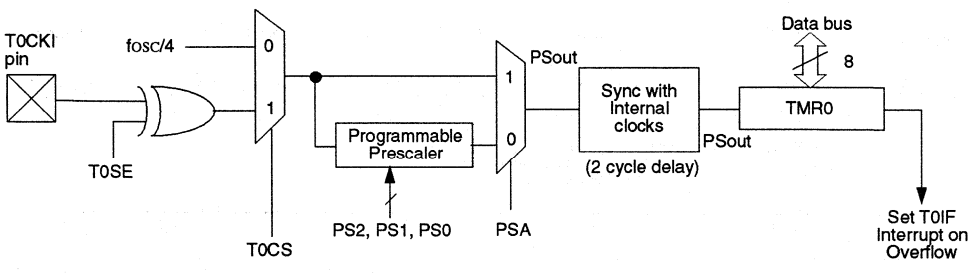
The prescaler is shared between the TMR0 module and the watchdog timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is not readable or writable. When the prescaler is assigned to the TMR0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 6.3 details the operation of the prescaler.

6.1 TIMER0 (TMR0) Interrupt

TMR0 interrupt is generated when the TMR0 module timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the TMR0 module interrupt service routine before re-enabling this interrupt. The TMR0 module interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. See Figure 6-4 for TMR0 interrupt timing.

2

FIGURE 6-1: TIMER0 (TMR0) BLOCK DIAGRAM



Note 1: Bits, T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.
Note 2: The prescaler is shared with Watchdog Timer (refer to Figure 6-6 for detailed diagram)

FIGURE 6-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALE

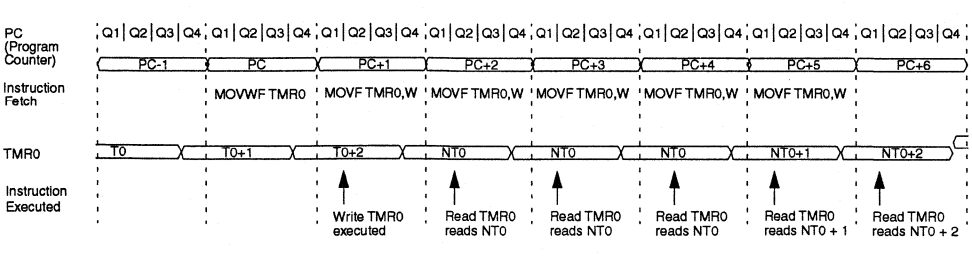


FIGURE 6-3: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/PRESCALE 1:2

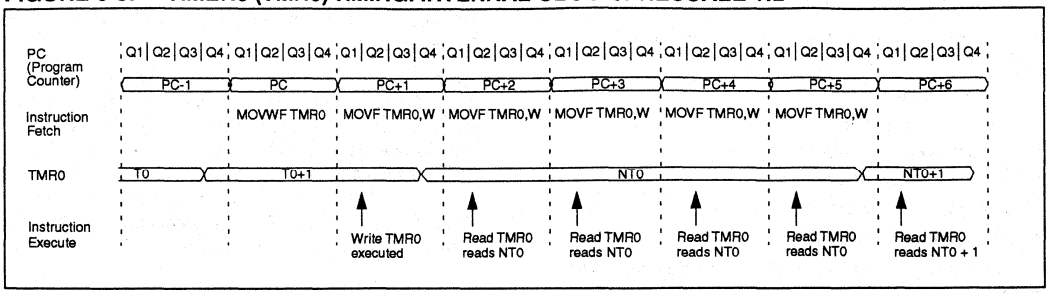
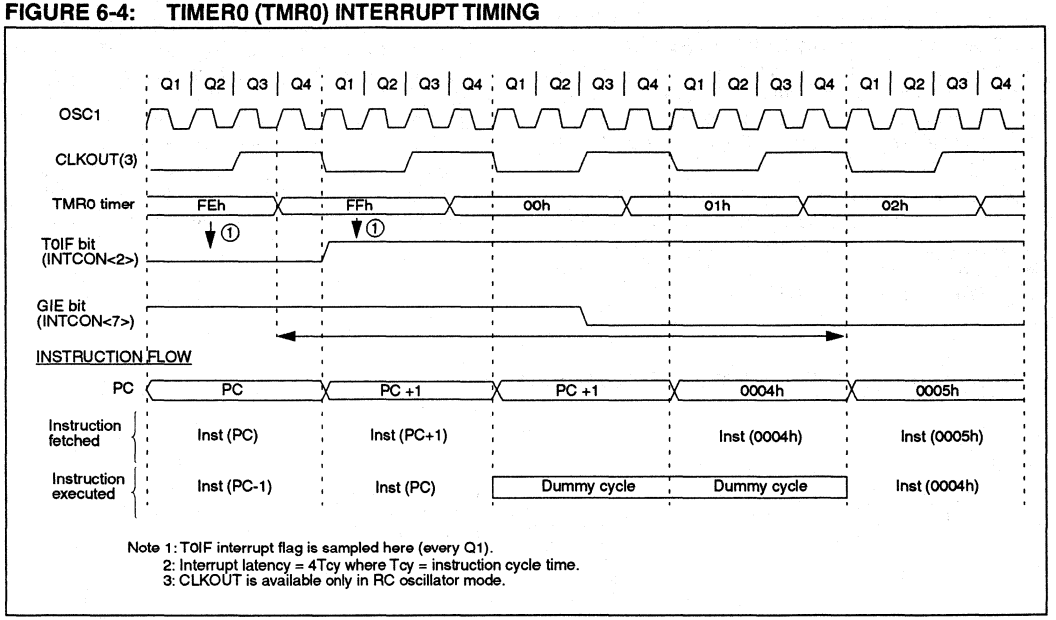


FIGURE 6-4: TIMER0 (TMR0) INTERRUPT TIMING



Note 1: T0IF interrupt flag is sampled here (every Q1).
2: Interrupt latency = 4Tcy where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

 © 1995 Microchip Technology Inc.

6.2 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (see Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20ns) and low for at least 2Tosc (and a small RC delay of 20ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 Tosc (and a small RC delay of 40ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

6.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

6.3 Prescaler

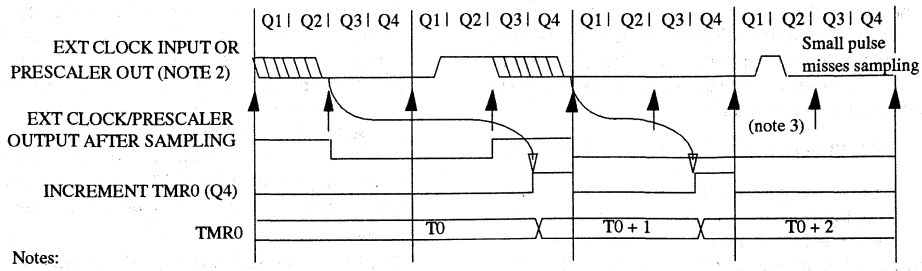
An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer, respectively (see Figure 6-6). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2-PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g. `CLRF 1, MOVWF 1, BSF 1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

2

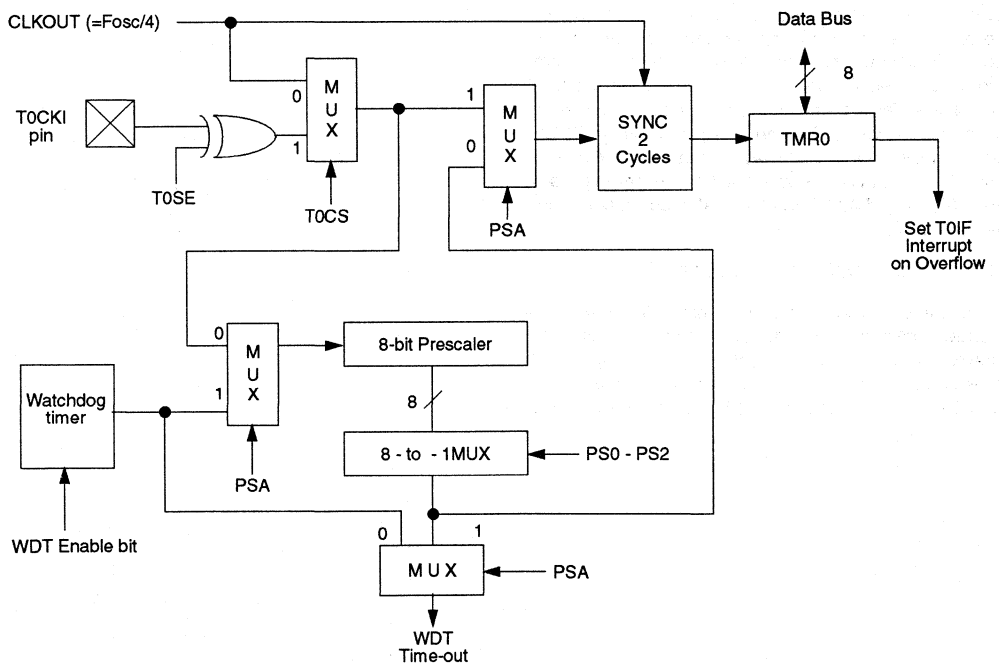
FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to TMR0 increment is 3 tosc to 7 tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 tosc max.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

FIGURE 6-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

 © 1995 Microchip Technology Inc.

6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (shown in Example 6-1) must be executed when changing the prescaler assignment from TMR0 to WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TMR0→WDT)

```
BCF     STATUS,  RP0 ;Bank 0
CLRF    TMR0         ;Clear TMR0 & Prescaler
BSF     STATUS,  RP0 ;Bank 1
CLRWDT               ;Clears WDT and
                     ;
MOVLW   B'xxxx1xxx'  ;Select new prescaler
MOVWF   OPTION       ;value
BCF     STATUS,  RP0;Bank 0
```

To change prescaler from the WDT to the TMR0 module use the sequence shown in Example 6-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
BSF     STATUS, RP0
MOVLW   B'xxxx0xxx'   ;Select TMR0, new
                      ;prescale value and
                      ;clock source
MOVWF   OPTION
BCF     STATUS, RP0
```

2

TABLE 6-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. See Figure 6-5.	81h	1111 1111
INTCON	TMR0 overflow interrupt flag and mask bits. See Figure 6-6	0Bh	0000 000x

Legend: x = unknown, - = unimplemented, reads as a '0'.
Note: For reset values of registers in other reset situations refer to Table 14-8.

TABLE 6-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	TMR0	TMR0							
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
85h	TRISA	—	—	TRISA 5[1]	TRISA 4	TRISA 3	TRISA 2	TRISA 1	TRISA 0

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by TMR0 module
Note 1: This bit is not available in the PIC16C61

NOTES:

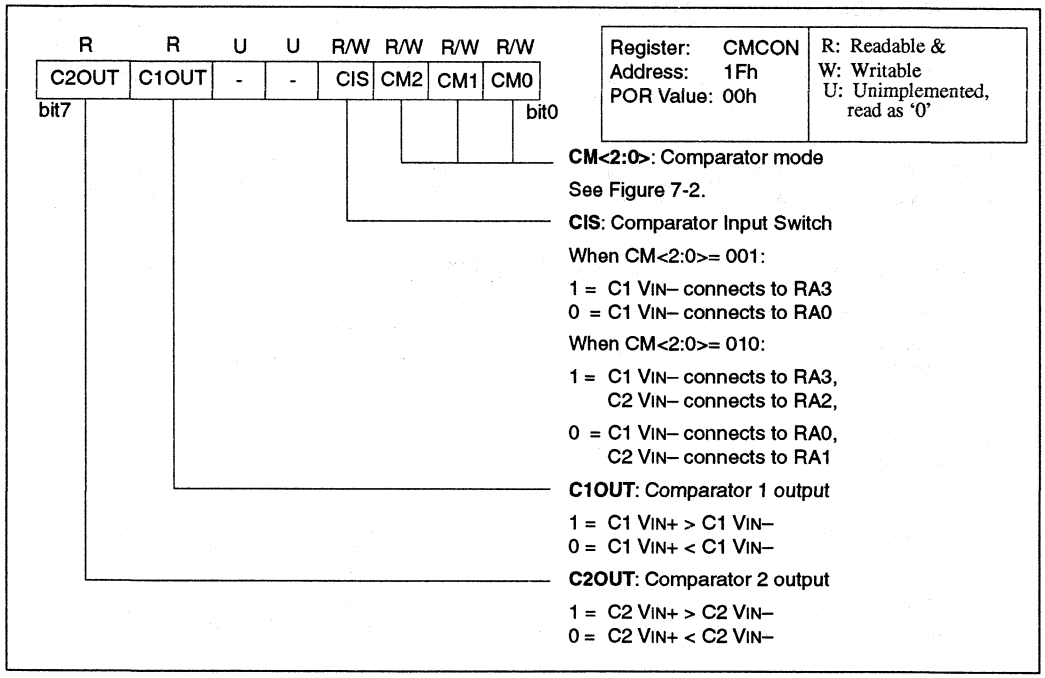
 © 1995 Microchip Technology Inc.

7.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the RA0 through RA3 pins. The on-chip Voltage Reference (see Section 8.0) can also be an input to the comparators.

The CMCON register, shown in Figure 7-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 7-2.

FIGURE 7-1: CMCON REGISTER

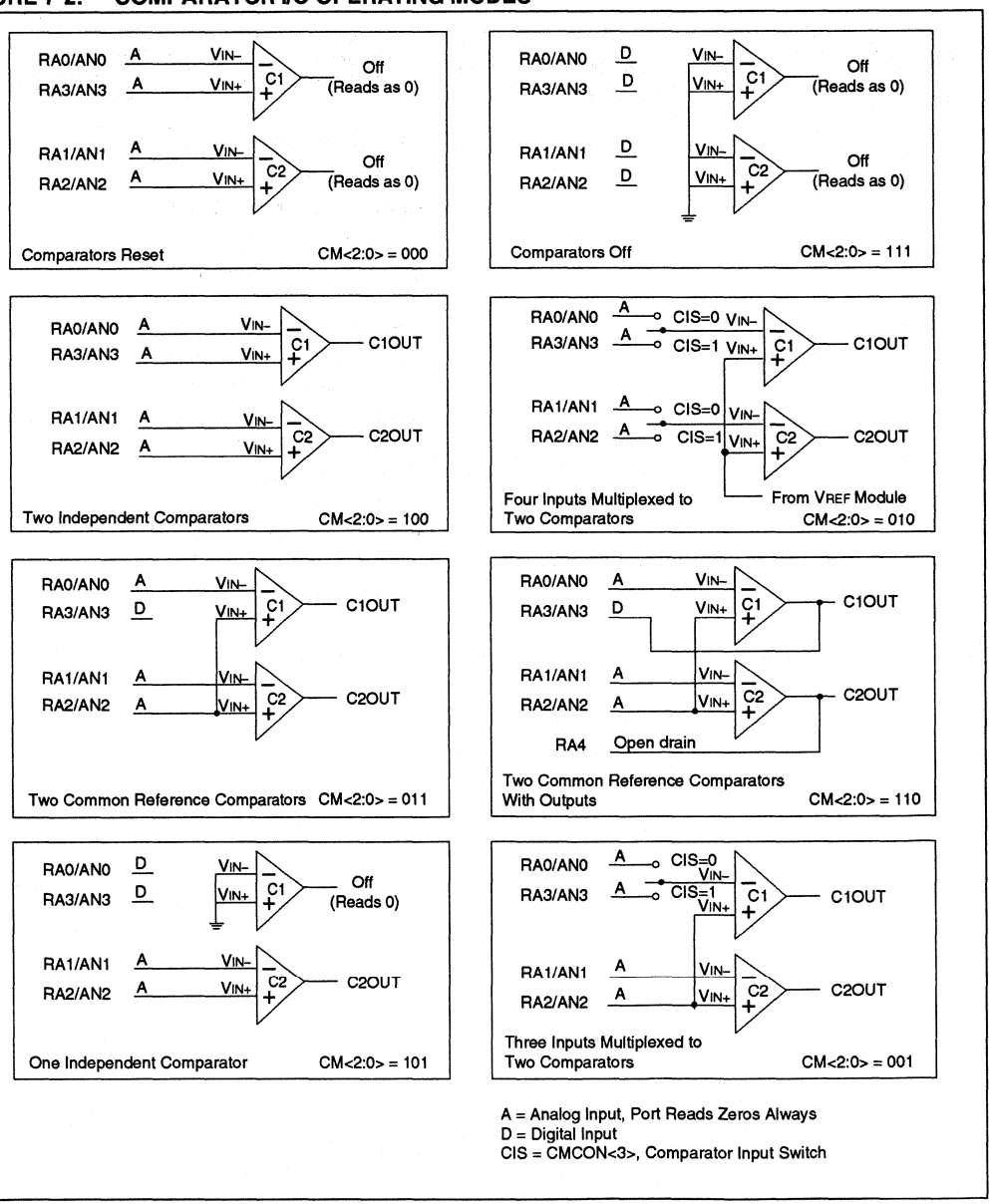


7.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. Figure 7-2 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 12-4.

Note: Comparator interrupts should be disabled during a comparator mode change otherwise a false interrupt may occur.

FIGURE 7-2: COMPARATOR I/O OPERATING MODES



A = Analog Input, Port Reads Zeros Always
D = Digital Input
CIS = CMCON<3>, Comparator Input Switch

 © 1995 Microchip Technology Inc.

The code example in Example 7-1 depicts the steps required to configure the comparator module. RA3 and RA4 are configured as digital output. RA0 and RA1 are configured as the V- inputs and RA2 as the V+ input to both comparators.

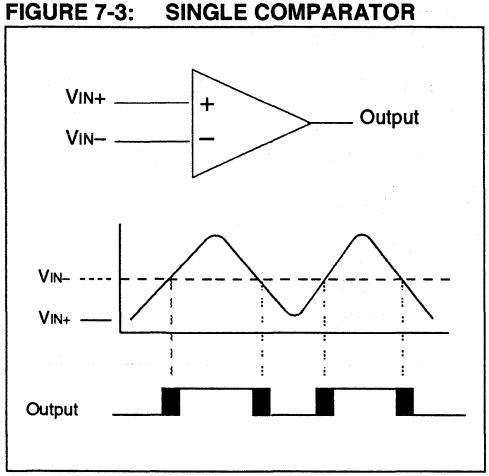
EXAMPLE 7-1: INITIALIZING COMPARATOR MODULE

```
FLAG_REG  EQU      0X20
CLRF      FLAG_REG ;Init flag register
CLRF      PORTA    ;Init PORTA
ANDLW     0xC0     ;Mask comparator bits
IORWF     FLAG_REG,F ;Store bits in flag register
MOVLW     0x03     ;Init comparator mode
MOVWF     CMCON    ;CM<2:0> = 011
BSF       STATUS,RP0 ;Select Bank1
MOVLW     0x07     ;Initialize data direction
MOVWF     TRISA    ;Set RA<2:0> as inputs
                   ;RA<4:3> as outputs
                   ;TRISA<7:5> always read '0'
BCF       STATUS,RP0 ;Select Bank 0
CALL      DELAY 10 ; 10µ delay
MOVF      CMCON,F  ;Read CMCON to end change condition
BCF       PIR1,CMIF ;Clear pending interrupts
BSF       STATUS,RP0 ;Select Bank 1
BSF       PIE1,CMIE ;Enable comparator interrupts
BCF       STATUS,RP0' ;Select Bank 0
BSF       INTCON,PEIE ;Enable peripheral interrupts
BSF       INTCON,GIE ;Global interrupt enable
```

7.2 Comparator Operation

A single comparator is shown in Figure 7-3 along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN–, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN–, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 7-3 represent the uncertainty due to input offsets and response time.

FIGURE 7-3: SINGLE COMPARATOR



7.3 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal that is present at VIN– is compared to the signal at VIN+, and the digital output of the comparator is adjusted accordingly (see Figure 7-3).

7.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between VSS and VDD, and can be applied to either pin of the comparator(s).

7.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. Section 8.0 contains a detailed description of the Voltage Reference Module that provides this signal. The internal reference signal is used when the comparators is in mode CM<2:0>=010 (see Figure 7-2). In this mode, the internal voltage reference is applied to the VIN+ pin of both comparators.

7.4 Comparator Response Time

Response time is the minimum time after selecting a new reference voltage or input source, before the comparator output is guaranteed to have a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise the maximum delay of the comparators should be used. (See Table 12-4 and Table 12-5)

7.5 Comparator Outputs

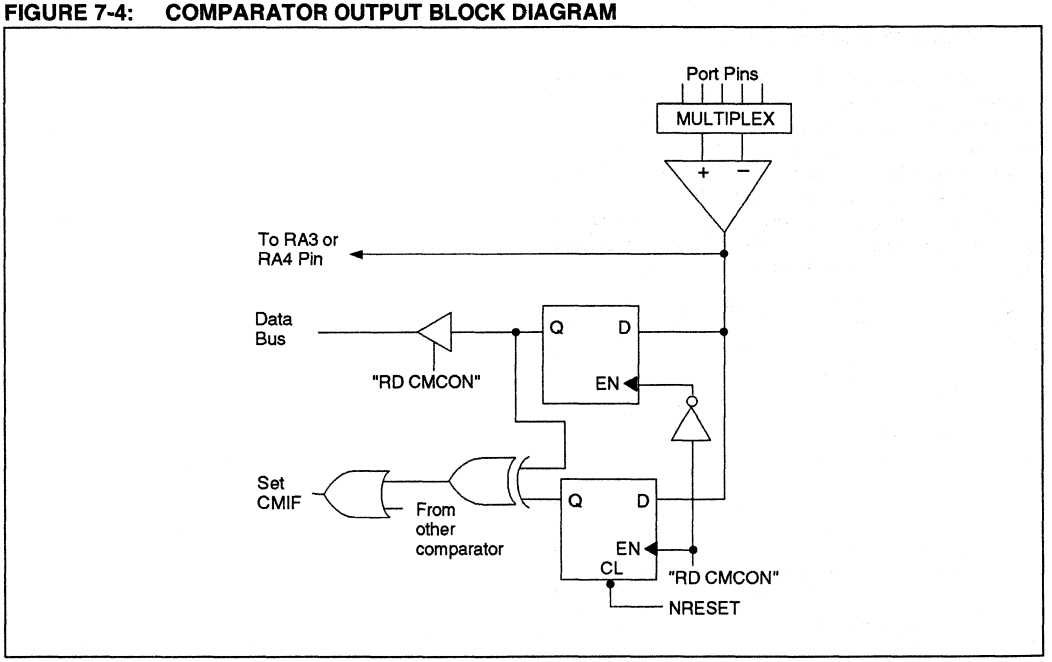
The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When the CM<2:0> = 110, multiplexors in the output path of the RA3 and RA4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 7-4 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/disable for the RA3 and RA4 pins while in this mode.

Note 1: When reading the PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt trigger input specification.

Note 2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

FIGURE 7-4: COMPARATOR OUTPUT BLOCK DIAGRAM



 © 1995 Microchip Technology Inc.

7.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit, PIR1<6>, is the comparator interrupt flag. The CMIF bit must be reset by setting it to a '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit and the PEIE bit must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

Note: If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<6>) interrupt flag may not be set.

7.7 Comparator Operation During SLEEP

When a comparator is active and the device is placed in sleep mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake up the device from sleep mode when enabled. While the comparator is powered-up, higher sleep currents than shown in the power down current specification will occur. Each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in sleep mode, turn off the comparators, CM<2:0> = 111, before entering sleep. If the device wakes-up from sleep, the contents of the CMCON register are not affected.

7.8 Effects of a RESET

A device reset forces the CMCON register to its reset state. This forces the comparator module to be in the comparator reset mode, CM<2:0> = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at reset time. The comparators will be powered-down during the reset interval.

7.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 7-5. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10KΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 7-5: ANALOG INPUT MODEL

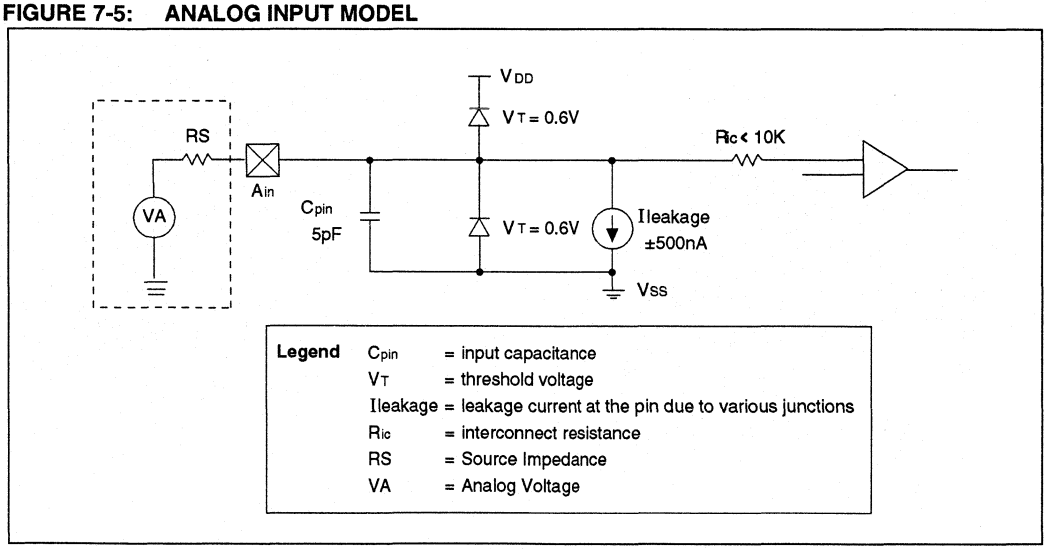


TABLE 7-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
0Bh	INCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	—	CMIF	—	—	—	—	—	—
8Ch	PIE1	—	CMIE	—	—	—	—	—	—
85h	TRISA	—	—	—	TRIS4	TRISA3	TRISA2	TRISA1	TRISA0

 © 1995 Microchip Technology Inc.

8.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference is not being used. The VRCON register controls the operation of the reference as shown in Figure 8-1. The block diagram is given in Figure 8-2.

FIGURE 8-1: VRCON REGISTER

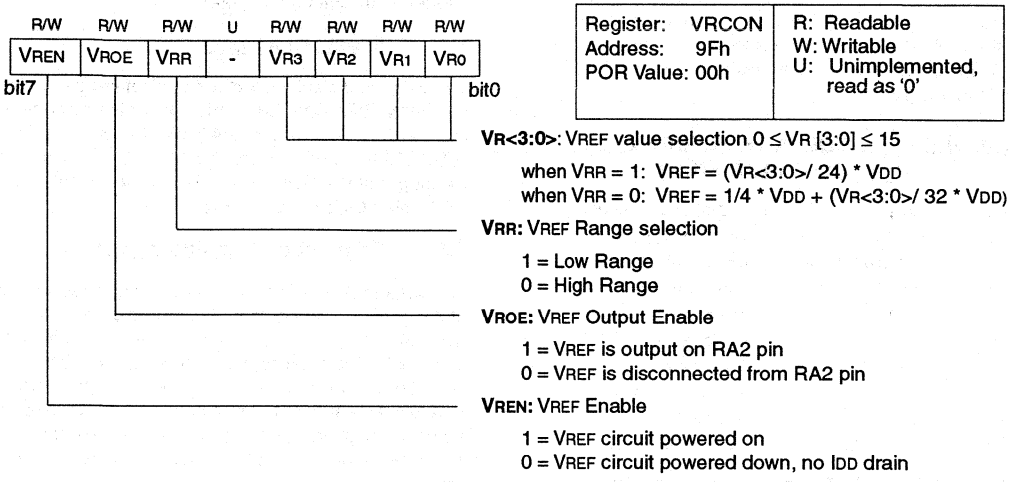
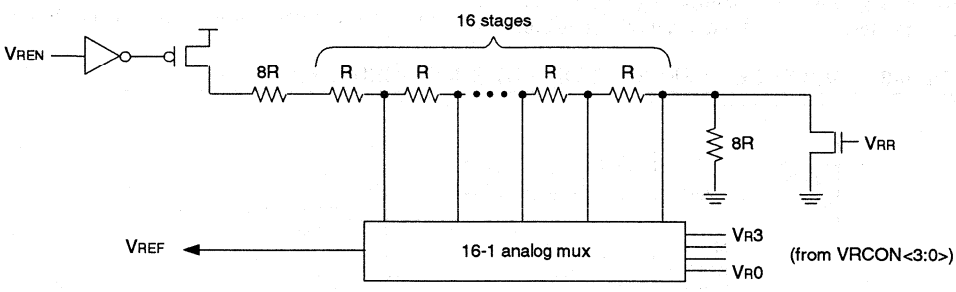


FIGURE 8-2: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Table 12-5.

8.1 Configuring the Voltage Reference

The Voltage Reference can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

if VRR = 1: VREF = (VR<3:0>/24) x VDD

if VRR = 0: VREF = (VDD x 1/4) + (VR<3:0>/32) x VDD

The setting time of the Voltage Reference must be considered when changing the VREF output (see Table 12-5). Example 8-1 shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with VDD = 5.0V.

EXAMPLE 8-1: VOLTAGE REFERENCE CONFIGURATION

```
MOVLW   0x02        ; 4 Inputs Muxed
MOVWF   CMCON       ; to 2 comps.
BSF     STATUS,RP0  ; go to Bank 1
MOVLW   0x07        ; RA3-RA0 are
MOVWF   TRISA       ; outputs
MOVLW   0xA6        ; enable VREF
MOVWF   VRCON       ; low range
                    ; set VR<3:0>=6
BCF     STATUS,RP0  ; go to Bank 0
CALL    DELAY10     ; 10µ delay
```

8.2 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (see Figure 8-2) keep VREF from approaching VSS or VDD. The Voltage Reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The absolute accuracy of the Voltage Reference can be found in Table 12-5.

8.3 Operation During Sleep

When the device wakes up from sleep through an interrupt or a Watchdog Timer timeout, the contents of the VRCON register are not affected. To minimize current consumption in sleep mode, the Voltage Reference should be disabled.

8.4 Effects of a Reset

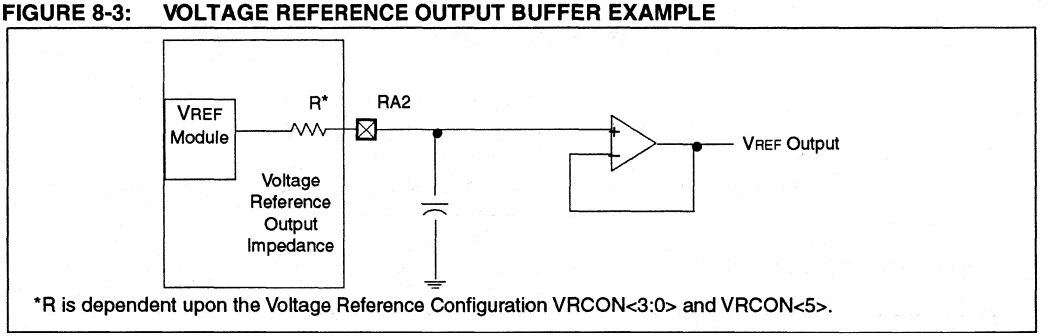
A device reset disables the Voltage Reference by clearing VREN (VRCON<7>). This reset also disconnects the reference from the RA2 pin by clearing VROE (VRCON<6>) and selects the high voltage range by clearing VRR (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

8.5 Connection Considerations

The Voltage Reference Module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit is set and the VROE bit, VRCON<6>, is set. Enabling the Voltage Reference output onto the RA2 pin with an input signal present will increase current consumption. Connecting RA2 as a digital output with VREF enabled will also increase current consumption.

The RA2 pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the Voltage Reference output for external connections to VREF. Figure 8-3 shows an example buffering technique.

FIGURE 8-3: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



*R is dependent upon the Voltage Reference Configuration VRCON<3:0> and VRCON<5>.

TABLE 8-1: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
8Ch	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0

 © 1995 Microchip Technology Inc.

9.0 SPECIAL FEATURES OF THE CPU

What sets apart a microcontroller from other processors are special circuits to deal with the needs of real time applications. The PIC16C62X family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

1. OSC selection
2. Reset
   - Power-On Reset (POR)
   - Power-Up Timer (PWRT)
   - Oscillator Start-Up Timer (OST)
   - Brown-out Detect (BOD)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID Locations
8. In-circuit serial programming

The PIC16C62X has a watchdog timer which is controlled by configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the oscillator start-up timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the power-up timer (PWRT), which provides a fixed delay of 72ms (nominal) on power up only, designed to keep the part in reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs which provides at least a 72 ms reset. With these three functions on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset, watchdog timer time-out or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

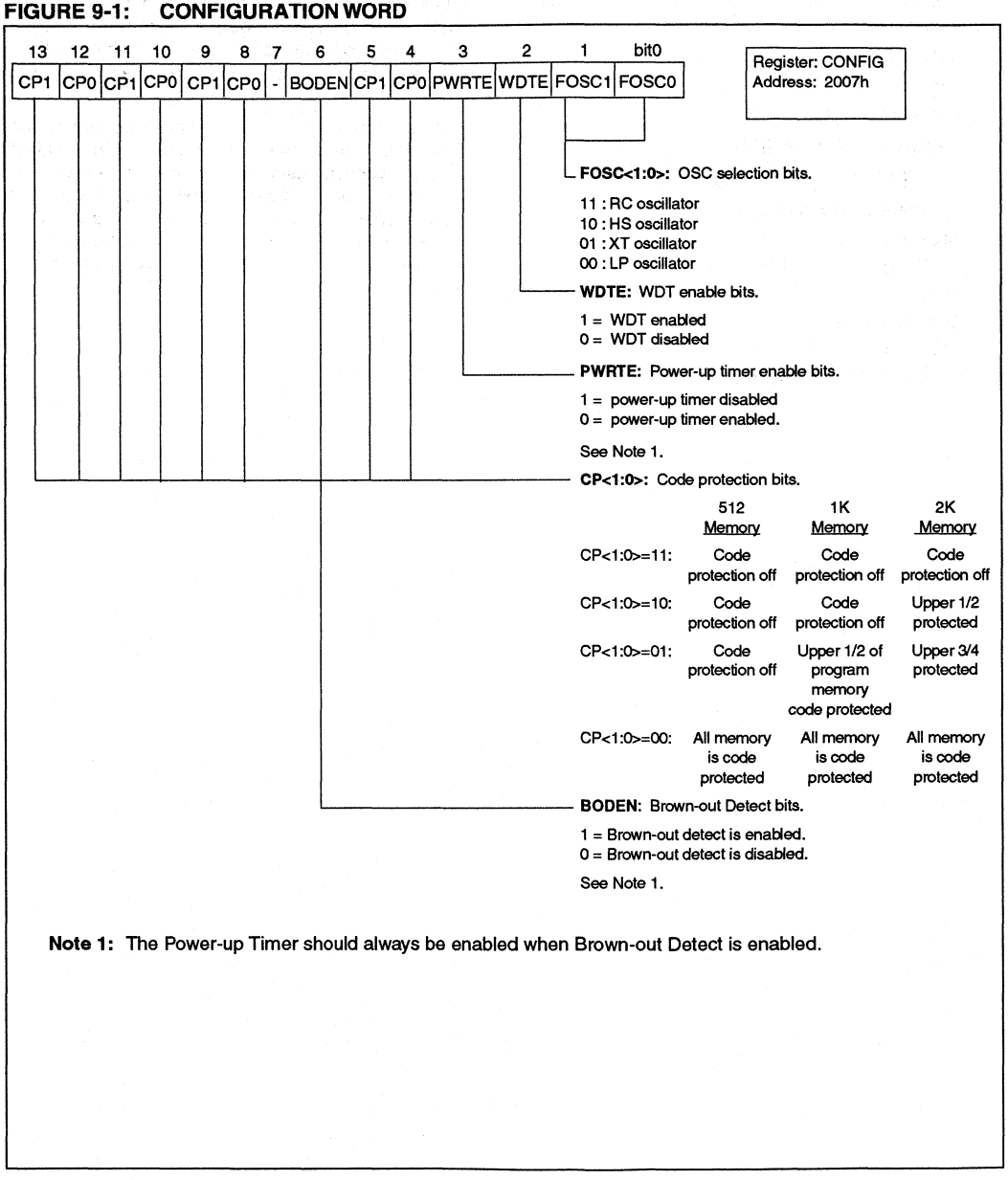
© 1995 Microchip Technology Inc.

9.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

FIGURE 9-1: CONFIGURATION WORD



Note 1: The Power-up Timer should always be enabled when Brown-out Detect is enabled.

 © 1995 Microchip Technology Inc.

9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

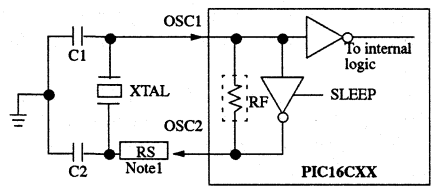
The PIC16CXX can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 9-2). The PIC16CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin. This is shown in Figure 9-3.

FIGURE 9-2: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



See Table 9-1 and Table 9-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 9-3: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

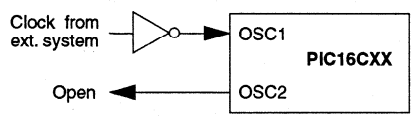


TABLE 9-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (PRELIMINARY)

Oscillator Type	Resonator Frequency	Capacitor Range C1 = C2
XT	455 kHz 2.0 MHz 4.0 MHz	22-100pF 15-68pF 15-68pF
HS	8.0MHz 16.0 MHz	10-68pF 10-22pF

Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:

455 kHz	Panasonic EFO-A455K04B +/- 0.3%
2.0 MHz	Murata Erie CSA2.00MG +/- 0.5%
4.0 MHz	Murata Erie CSA4.00MG +/- 0.5%
8.0 MHz	Murata Erie CSA8.00MT +/- 0.5%
16.0 MHz	Murata Erie CSA16.00MX +/- 0.5%

All resonators used did not have built-in capacitors.

TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (PRELIMINARY)

Osc Type	Freq	C1	C2
LP	32 kHz 200 kHz	68-100pF 15-30pF	68-100pF 15-30pF
XT	100 kHz 2 MHz 4 MHz	68-150pF 15-30pF 15-30pF	150-200pF 15-30pF 15-30pF
HS	4 MHz 10 MHz 20 MHz	15-30pF 15-30pF 15-30pF	15-30pF 15-30pF 15-30pF

Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

© 1995 Microchip Technology Inc.

9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 9-4 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 9-4: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

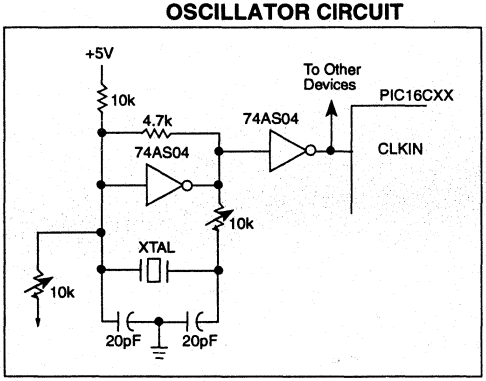
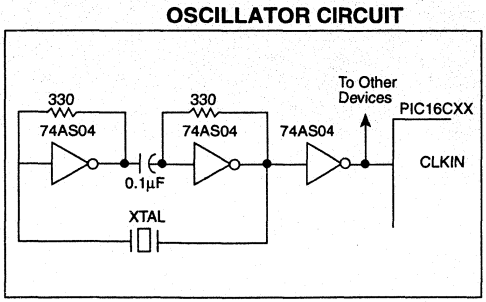


Figure 9-5 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330-W resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 9-5: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



9.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-6 shows how the R/C combination is connected to the PIC16CXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

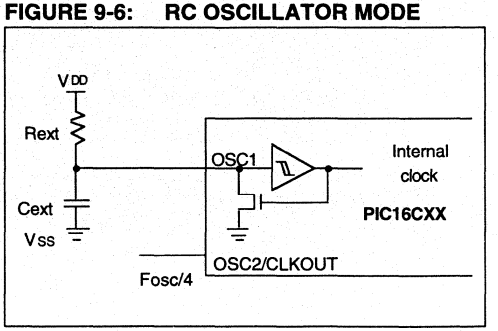
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 13.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 13.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-2 for waveform).

FIGURE 9-6: RC OSCILLATOR MODE



 © 1995 Microchip Technology Inc.

9.3 Reset

The PIC16CXX differentiates between various kinds of reset:

a) Power-on reset (POR)
b) $\overline{MCLR}$ reset during normal operation
c) $\overline{MCLR}$ reset during SLEEP
d) WDT time-out reset during normal operation
e) WDT time-out reset during SLEEP
f) Brown-out Detect

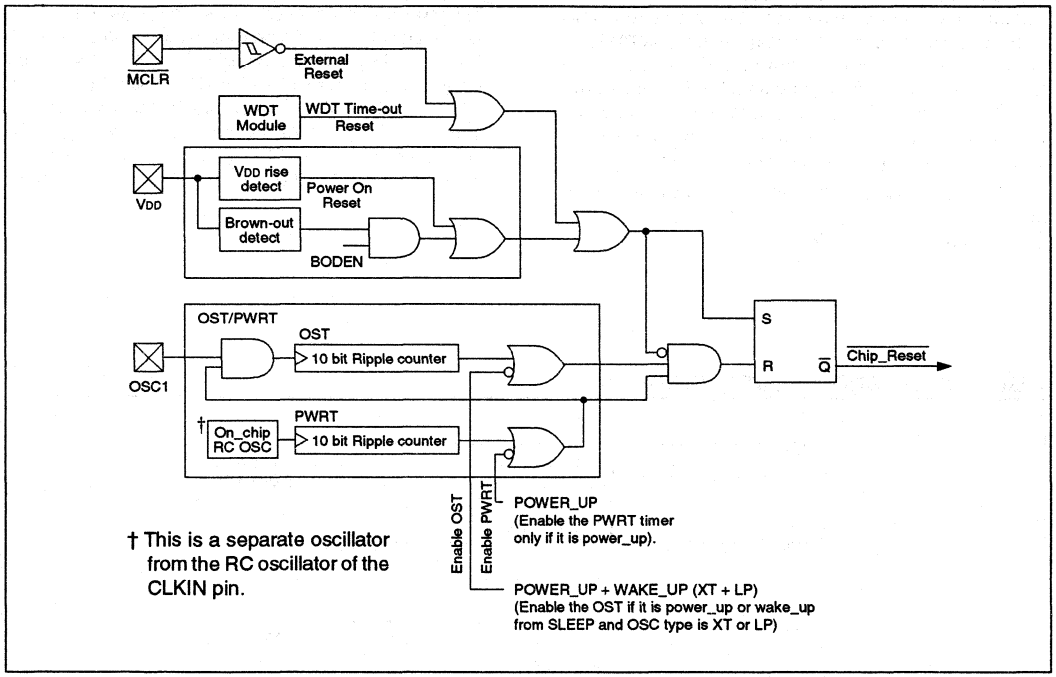
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on power-on reset (POR), on $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 9-4. These bits are used in software to determine the nature of reset. See Table 9-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 9-7.

The $\overline{MCLR}$ reset path has a noise filter to detect and ignore small pulses. See Table 12-6 for pulse width specification.

2

FIGURE 9-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



9.4 Power-On Reset (POR), Power-Up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Detect (BOD)

9.4.1 POWER-ON RESET (POR)

A Power-On Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.6V - 1.8V). To take advantage of the POR, just tie $\overline{MCLR}$ pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-On Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

The POR circuit does not produce internal reset when VDD declines.

9.4.2 POWER-UP TIMER (PWRT)

The Power-Up Timer provides a fixed 72ms (nominal) time-out on power-up only, from POR or Brown-out Detect. The power-up timer operates on an internal RC oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, PWRTE can disable (if set) or enable (if cleared or programmed) the power-up timer. The Power-up Timer should always be enabled when Brown-out Detect is enabled.

The Power-Up Time delay will vary from chip to chip and due to VDD, temperature and process variation. See DC parameters for details.

9.4.3 OSCILLATOR START-UP TIMER (OST)

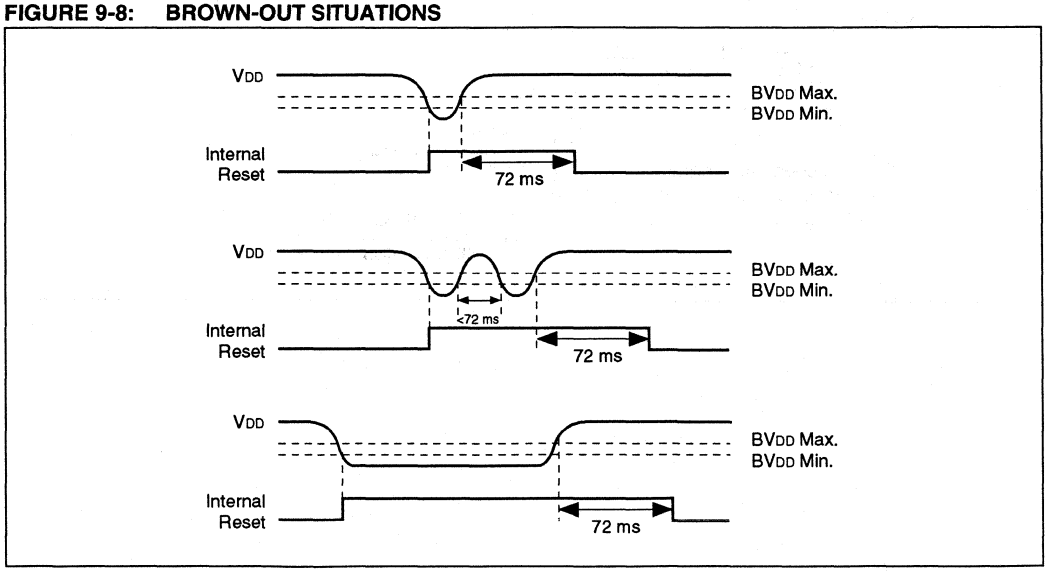
The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This guarantees that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on power-on reset or wake-up from SLEEP.

9.4.4 BROWN-OUT DETECT (BOD)

The PIC16C62X members have on-chip brown-out detection circuitry. A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Detect reset circuitry. If VDD falls below 4.0V (3.8V - 4.2V range) for greater than parameter 35 in Table 12-6, the brown-out situation will reset the chip. A reset is not guaranteed to occur if VDD falls below 4.0V for less than parameter 35. The chip will remain in Brown-out Detect reset until VDD rises above BVDD. The Power-Up Timer will now be invoked and will keep the chip in reset an additional 72 ms. If VDD drops below BVDD while the Power-Up Timer is running, the chip will go back into a Brown-out Detect reset and the Power-Up Timer will be initialized. Once VDD rises above BVDD, the Power-Up Timer will execute a 72 ms reset. The Power-up Timer should always be enabled when Brown-out Detect is enabled. Figure 9-8 shows typical Brown-out situations.

FIGURE 9-8: BROWN-OUT SITUATIONS



 © 1995 Microchip Technology Inc.

9.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in RC mode with PWRTE bit erased (PWRT disabled), there will be no time-out at all. Figure 9-9, Figure 9-10 and Figure 9-11 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately (see Figure 9-10). This is useful for testing purposes or to synchronize more than one PIC16C62X device operating in parallel.

Table 9-5 shows the reset conditions for some special registers, while Table 9-6 shows the reset conditions for all the registers.

9.4.6 POWER CONTROL/STATUS REGISTER (PCON)

The power control/status register, PCON (address 8Eh) has two bits.

Bit 0 is $\overline{BO}$ (Brown-out). $\overline{BO}$ is unknown on power-on-reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BO}$ = 0 indicating that a brown-out has occurred. The $\overline{BO}$ status bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled (by setting BODEN bit = 0 in the Configuration word).

Bit 1 is $\overline{POR}$ (Power-on-reset). It is a '0' on power-on-reset and unaffected otherwise. The user must write a '1' to this bit following a power-on-reset. On a subsequent reset if $\overline{POR}$ is '0', it will indicate that a power-on-reset must have occurred (V_{DD} may have gone too low).

2

TABLE 9-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power Up		Brown-out	Wake-up from SLEEP
	PWRTE = 0	PWRTE = 1		
XT, HS, LP	72 ms + 1024 tosc	1024 tosc	72 ms + 1024 tosc	1024 tosc
RC	72 ms	—	72 ms	—

TABLE 9-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{BO}$	$\overline{TO}$	$\overline{PD}$	
0	X	1	1	Power-on-reset
0	X	0	X	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	X	X	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	X	X	Brown-out reset
1	1	0	1	WDT reset during normal operation
1	1	0	0	WDT time-out wakeup from SLEEP
1	1	1	1	$\overline{MCLR}$ reset during normal operation
1	1	1	0	$\overline{MCLR}$ reset during SLEEP

TABLE 9-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

	PCL Addr: 02h	STATUS Addr: 03h	PCON Addr: 8Eh
Power on reset	000h	0001 1xxx	---- --0x
$\overline{MCLR}$ reset during normal operation	000h	0001 1uuu	---- --uu
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu	---- --uu
WDT reset during normal operation	000h	0000 1uuu	---- --uu
WDT during SLEEP	PC + 1	uuu0 0uuu	---- --uu
Brown-out reset	000h	0001 1uuu	---- --u0
Interrupt wake-up from SLEEP	PC + 1 [1]	uuu1 0uuu	---- --uu

Legend: Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.
Note 1: When wake-up is due to an interrupt (GIE = 1), the next instruction after the SLEEP instruction is executed, and then a branch to 004h occurs.

TABLE 9-6: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	• Power-on Reset	• $\overline{MCLR}$ Reset during normal operation • $\overline{MCLR}$ Reset during SLEEP • WDT Reset • Brown-out Reset (1)	• Wake up from SLEEP through interrupt • Wake up from SLEEP through WDT time-out
W	-	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	-	-	-
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000 0000	0000 0000	PC + 1 (3)
STATUS	03h	0001 1xxx	000? ?uuu (4)	uuu? ?uuu (4)
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CMCON	1Fh	00-- 0000	00-- 0000	uu-- uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000x	uuuu uuuu (2)
PIR1	0Ch	-0-- ----	-0-- ----	-u-- ---- (2)
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	-0-- ----	-0-- ----	-u-- ----
PCON	8Eh	---- --0x	---- --u? (1)	---- --uu
VRCON	9Fh	000- 0000	000- 0000	uuu- uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', ? = value depends on condition.

Note 1: If V_{DD} goes too low, power on reset will be activated and registers will be affected differently.
Note 2: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).
Note 3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
Note 4: See Table 9-5 for reset value for specific condition.

 © 1995 Microchip Technology Inc.

FIGURE 9-9: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

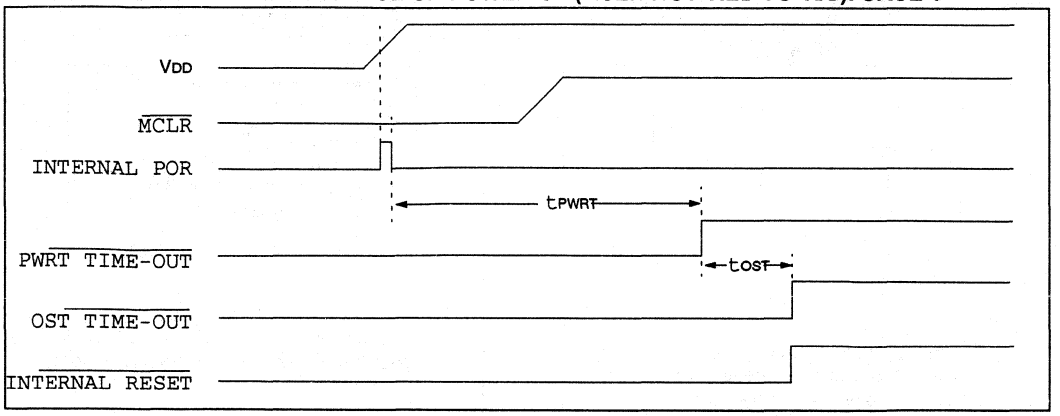


FIGURE 9-10: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

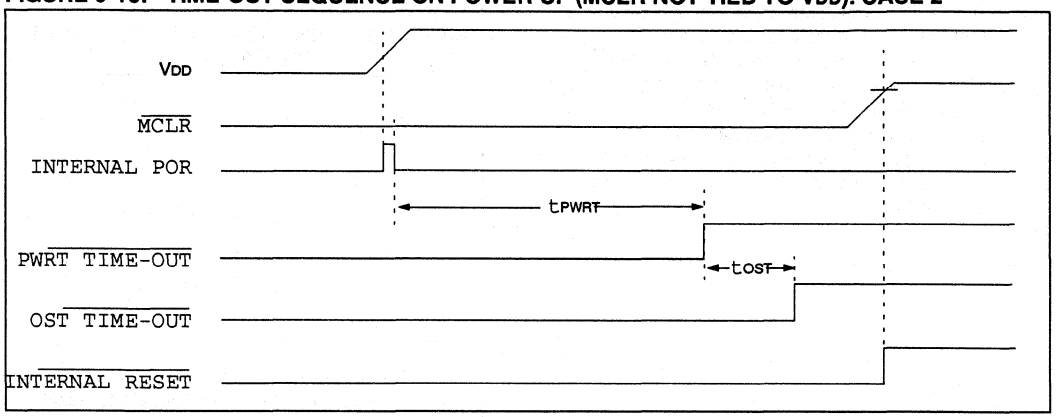


FIGURE 9-11: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

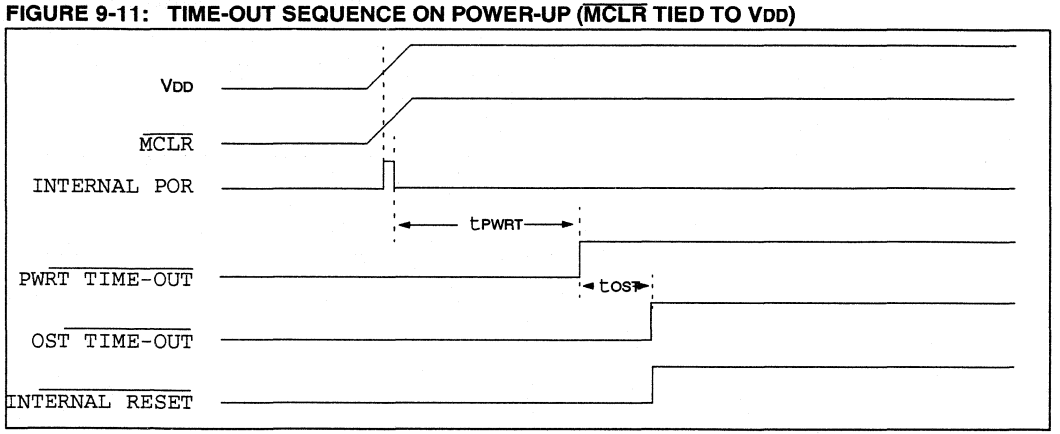
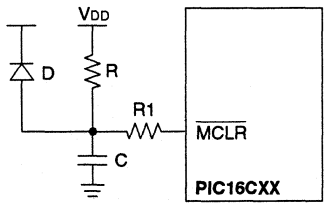


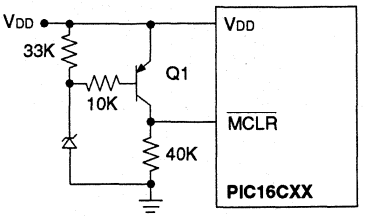
FIGURE 9-12: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note:

1. External power-on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40KΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5μA). A larger voltage drop will degrade VIH level on $\overline{MCLR}$ pin.
3. R1 = 100Ω to 1KΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$ pin breakdown due to ESD or EOS.

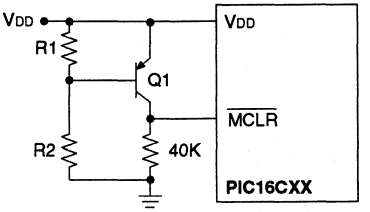
FIGURE 9-13: BROWN-OUT PROTECTION CIRCUIT 1



Note:

1. This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 9-14: BROWN-OUT PROTECTION CIRCUIT 2



Note:

1. This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \cdot \frac{R1}{R1 + R2} = 0.7V.$$

 © 1995 Microchip Technology Inc.

9.5 Interrupts

The PIC16C62X has 4 sources of interrupt:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PortB change interrupts (pins RB<7:4>)
- Comparator interrupt

The interrupt control register (INTCON, addr 0Bh/8BH) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The "return from interrupt" instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flag is contained in the special register PIR1. The corresponding interrupt enable bit is contained in special registers PIE1.

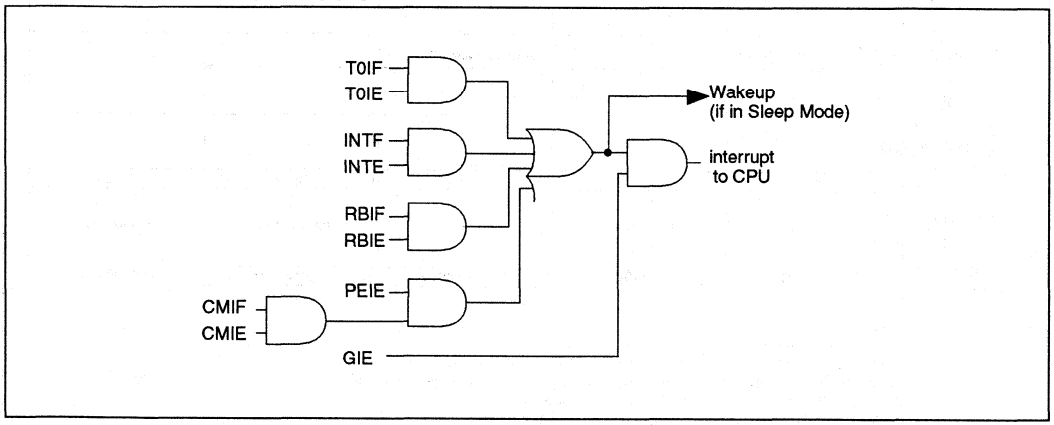
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (see Figure 9-16). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a NOP in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

FIGURE 9-15: INTERRUPT LOGIC



9.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the INT pin, the INTF bit (INTCON<1>) is set . This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 9.8 for details on SLEEP and Figure 9-19 for timing of wake-up from SLEEP through INT interrupt.

9.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the TMR0 module, see Section 6.0.

9.5.3 PORT RB INTERRUPT

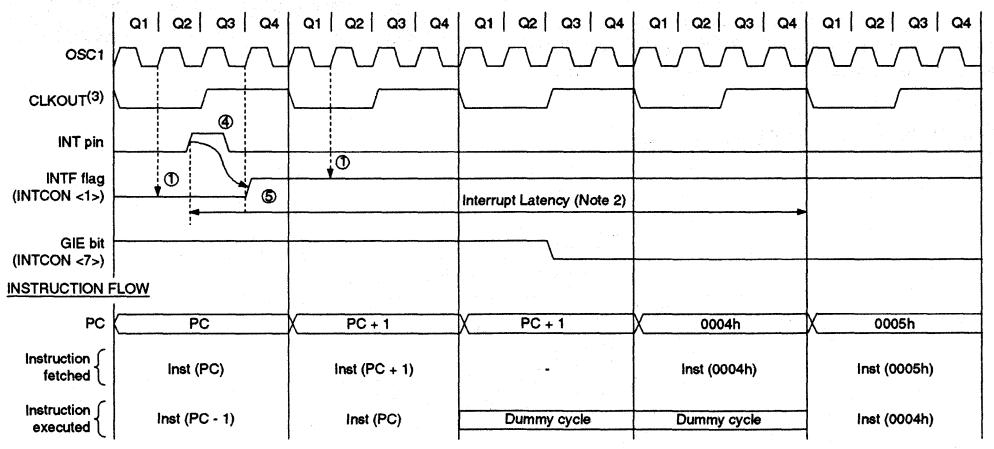
An input change on PortB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PortB, see Section 5.2.

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

9.5.4 COMPARATOR INTERRUPT

See Section 7.6 for complete description of comparator interrupts.

FIGURE 9-16: INT PIN INTERRUPT TIMING



Notes:
1. INTF flag is sampled here (every Q1)
2. Interrupt latency = 3-4 Tcy where Tcy = instruction cycle time.
   Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3. CLKOUT is available only in RC oscillator mode.
4. For minimum width spec of INT pulse, refer to AC specs.
5. INTF is enabled to be set anytime during the Q4-Q1 cycles.

 © 1995 Microchip Technology Inc.

9.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt e.g. W register and STATUS register. This will have to be implemented in software.

Example 9-1 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e. W_TEMP is defined at 0x20 in Bank 0 and it must also be defined at 0xA0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 9-1:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

EXAMPLE 9-1: SAVING THE STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP          ;copy W to temp register,
                        ;could be in either bank
SWAPF   STATUS,W        ;swap status to be saved into W
BCF     STATUS,RP0      ;change to bank 0 regardless
                        ;of current bank
MOVWF   STATUS_TEMP     ;save status to bank 0
                        ;register
    :
    :   (ISR)
    :
SWAPF   STATUS_TEMP,W   ;swap STATUS_TEMP register
                        ;into W, sets bank to original
                        ;state
MOVWF   STATUS          ;move W into STATUS register
SWAPF   W_TEMP,F        ;swap W_TEMP
SWAPF   W_TEMP,W        ;swap W_TEMP into W
```

9.7 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT timeout causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as a '0' (Section 9.1).

9.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

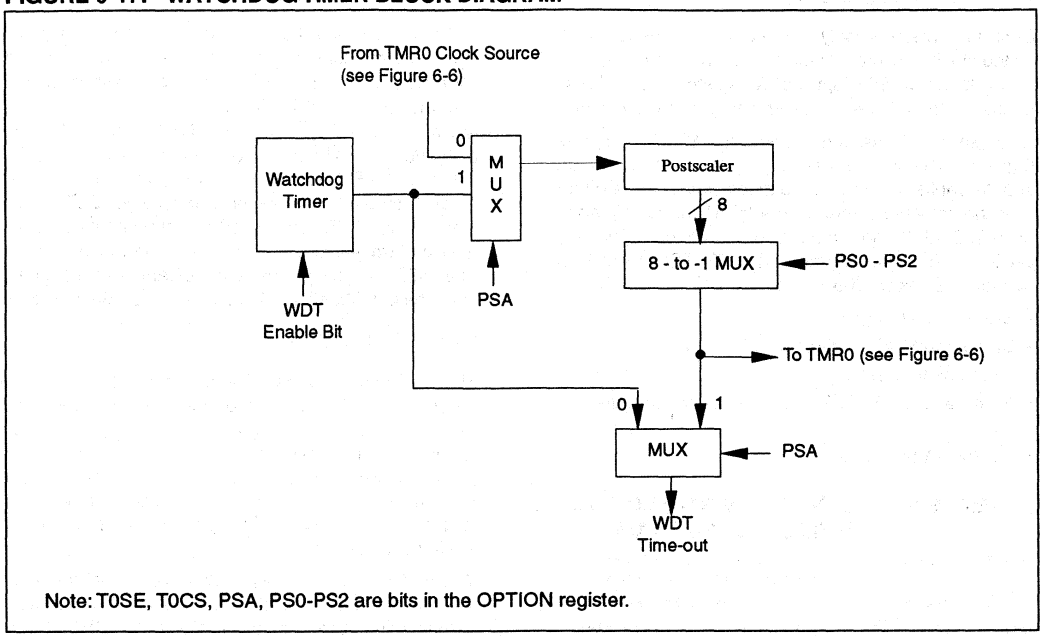
The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a watchdog timer time-out.

9.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

FIGURE 9-17: WATCHDOG TIMER BLOCK DIAGRAM



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

FIGURE 9-18: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. Bits	–	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

 © 1995 Microchip Technology Inc.

9.8 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP command was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin and the comparators and VREF should be disabled. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on chip pull-ups on PortB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a RESET generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

9.8.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin
2. Watchdog timer time-out reset (if WDT was enabled)
3. Interrupt from INT pin, RB port change, or the Peripheral Interrupt (Comparator).

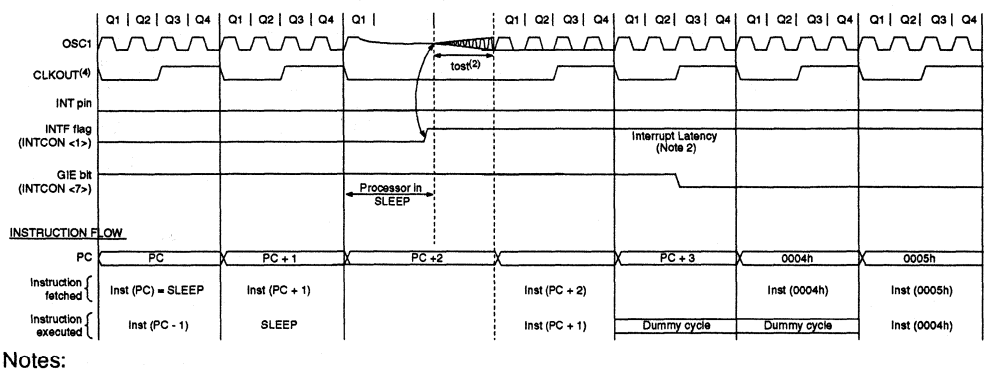
The first event will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. $\overline{PD}$ bit, which is set on power-up is cleared when SLEEP is invoked. $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

Note: If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wakeup from sleep. The sleep instruction is completely executed.

The WDT is cleared when the device wakes-up from sleep, regardless of the source of wake-up.

FIGURE 9-19: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Notes:

1. XT, HS or LP oscillator mode assumed.
2. tost = 1024 tosc (drawing not to scale) This delay will not be there for RC osc mode.
3. GIE = 1 assumed. In this case after wake up processor jumps to interrupt routine. If GIE = 0, execution will continue in line.
4. CLKOUT is not available in these osc modes, but shown here for timing reference.

9.9 Code Protection

The code in the program memory can be protected by programming the code protect bits (CP<1:0>).

Note: Code protection is not supported on windowed devices.

The microcontroller code protection scheme allows the user to selectively code protect portions of the program memory. Refer to Figure 9-1 for the bit assignments to enable code protection. Once a program segment has been code protected, those memory locations cannot be further programmed. Unprotected segments will read normally and may be reprogrammed.

The configuration word and ID locations are not code protected

9.10 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. Only the least significant 4 bits of the ID locations are useable.

9.11 In-Circuit Serial Programming

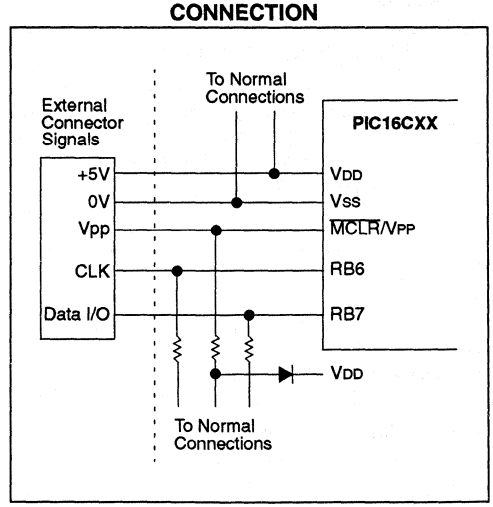
The PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

A typical in-system serial programming connection is shown in Figure 9-20.

FIGURE 9-20: TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION



 © 1995 Microchip Technology Inc.

10.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 10-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 10-1 shows the opcode field descriptions.

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be utilized by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 10-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8 bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable Bit
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-out Bit
$\overline{PD}$	Power-down Bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte oriented operations
- Bit oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

Table 10-2 lists the instructions recognized by the MPASM assembler.

Figure 10-1 shows the three general formats that the instructions can have.

Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS

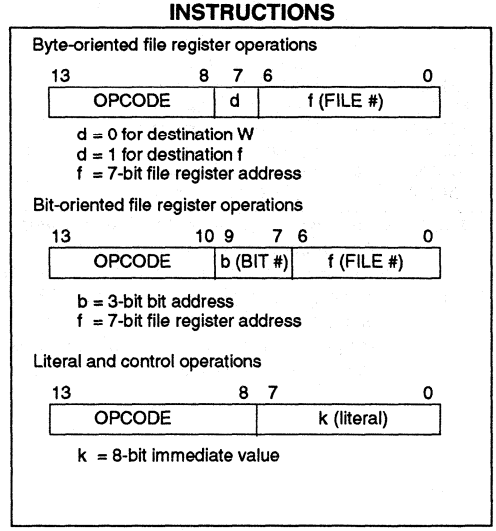


TABLE 10-2: PIC16CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode msb			lsb	Status Affected	Notes
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W and f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W and f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate left through carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W and f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal to W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal to W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear watchdog timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal to W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Excl. OR literal to W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g. MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

Note 2: If this instruction is executed on the TMR0 register (and, where applicable, d=1), the prescaler will be cleared if assigned to the TMR0.

Note 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

 © 1995 Microchip Technology Inc.

10.1 Instruction Descriptions

ADDLW **Add Literal to W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → W

Status Affected: C, DC, Z

Encoding:

11	111x	kkkk	kkkk

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ADDL   0x15
W
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

ADDWF **ADD W to f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0111	dfff	ffff

Description: Add the contents of the W register to register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDLW **And Literal and W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → W

Status Affected: Z

Encoding:

11	1001	kkkk	kkkk

Description: The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

00	0101	dfff	ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

© 1995 Microchip Technology Inc.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow f<b>$

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example

```
BCF     FLAG_REG,  7
```

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow f<b>$

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example

```
BSF     FLAG_REG,  7
```

Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **BIT Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is skipped.
If bit 'b' is '0' then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address TRUE
if FLAG<1>=1,
PC=address FALSE

 © 1995 Microchip Technology Inc.

BTFSS **Bit Test, skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address FALSE
if FLAG<1>=1,
PC=address TRUE

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL   THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE +1

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → f
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF    FLAG_REG
```

Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W Register**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0XXX	XXXX

Description: W registered is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description: CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

00	1001	dfff	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f)-1 → (dest)

Status Affected: Z

Encoding:

00	0011	dfff	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → d; skip if result = 0

Status Affected: None

Encoding:

00	1011	dfff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

 © 1995 Microchip Technology Inc.

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: k → PC<10:0>
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example
```
GOTO THERE
```
After Instruction
PC = Address THERE

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
INCF    CNT, 1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is decremented. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE     INCFSZ    CNT, 1
         GOTO      LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE +1

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
IORLW   0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (W)

Status Affected: Z

Encoding:

00	0100	dfff	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
IORWF          RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

Description: The contents of register f is moved to destination d. If d=0, destination is W register. If d=1, the destination is file register f itself. d=1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example
```
MOVF     FSR, 0
```
After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

11	00XX	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example
```
MOVLW    0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

00	0000	1fff	ffff

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example
```
MOVWF       OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

 © 1995 Microchip Technology Inc.

NOP — **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

00	0000	0XX0	0000

Description: No operation.

Words: 1

Cycles: 1

Example

```
NOP
```

OPTION — **Load Option Register**

Syntax: [*label*] OPTION

Operands: None

Operation: W → OPTION;

Status Affected: None

Encoding:

00	0000	0110	0010

Description: The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

RETFIE — **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE;

Status Affected: None

Encoding:

00	0000	0000	1001

Description: Return from Interrupt. Stack is popped and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting the Global Interrupt Enable (GIE) bit. GIE is the global interrupt enable bit (INTCON<7>). This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

RETLW — **Return Literal to W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W; TOS → PC;

Status Affected: None

Encoding:

11	01XX	kkkk	kkkk

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
CALL TABLE ;W contains table
           ;offset value
           ;W now has table value
           .
           .
           .
TABLE      ADDWF PC    ;W = offset
           RETLW k1    ;Begin table
           RETLW k2    ;
           .
           .
           .
           RETLW kn    ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k7

2

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

00	0000	0000	1000

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETURN
```

After Interrupt
PC = TOS

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

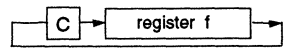
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example

```
RRF             REG1,0
```

Before Instruction
REG1 = 11100110
C = 0
After Instruction
REG1 = 11100110
W = 01110011
C = 1

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

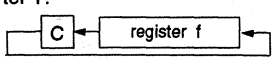
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example

```
RLF             REG1,0
```

Before Instruction
REG1 = 11100110
C = 0
After Instruction
REG1 = 11100110
W = 11001100
C = 1

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power down status bit ($\overline{PD}$) is cleared. Time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped. See Section 9.8 for more details.

Words: 1

Cycles: 1

Example: SLEEP

 © 1995 Microchip Technology Inc.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1:
```
SUBL    0x02
W
```
Before Instruction
```
W = 1
C = ?
```
After Instruction
```
W = 1
C = 1; result is positive
```

Example 2: Before Instruction
```
W = 2
C = ?
```
After Instruction
```
W = 0
C = 1; result is zero
```

Example 3: Before Instruction
```
W = 3
C = ?
```
After Instruction
```
W = FF
C = 0; result is negative
```

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: f - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement methodize W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBW    REG1,1
F
```
Before Instruction
```
REG1 = 3
W    = 2
C    = ?
```
After Instruction
```
REG1 = 1
W    = 2
C    = 1; result is positive
```

Example 2: Before Instruction
```
REG1 = 2
W    = 2
C    = ?
```
After Instruction
```
REG1 = 0
W    = 2
C    = 1; result is zero
```

Example 3: Before Instruction
```
REG1 = 1
W    = 2
C    = ?
```
After Instruction
```
REG1 = FF
W    = 2
C    = 0; result is negative
```


© 1995 Microchip Technology Inc.

11.0 DEVELOPMENT SUPPORT

11.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

11.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator provides the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 11-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 and better machines. The development software runs in the Microsoft Windows™ 3.x environment, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC Host Emulation Control Software

The Windows 3.1 operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host emulation control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously on the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes currently meet the specifications shown in Table 11-1.

FIGURE 11-1: PICMASTER SYSTEM CONFIGURATION

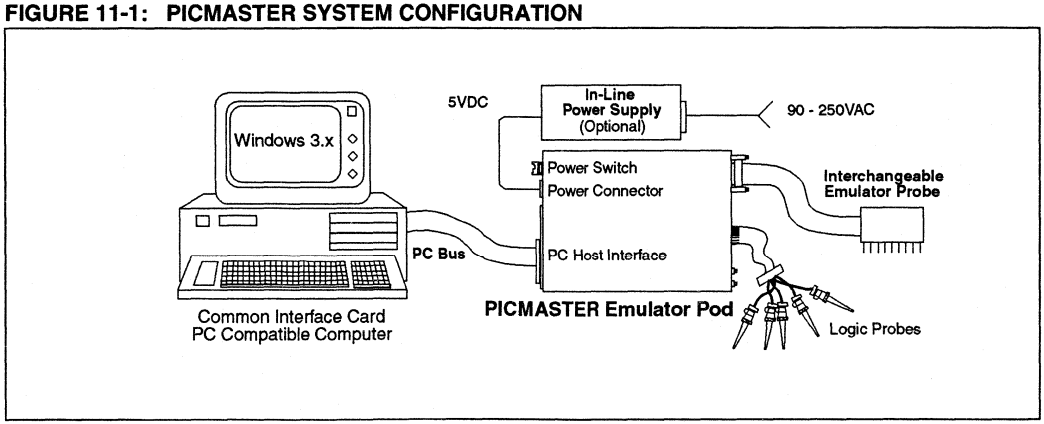


TABLE 11-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

*PROBE-16F indirectly supports the PIC16C65

 © 1995 Microchip Technology Inc.

11.3 PRO MATE™: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC-based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program, and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

11.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

11.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58), PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

11.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

© 1995 Microchip Technology Inc.

11.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications:

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro Assembly Capability
- Produces Object, Listing, Symbol and special files required for debugging with one of the Microchip Emulator systems.
- Supports Hex (default), Decimal and Octal source, and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically, by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and sub-titles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

11.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in single step, execute until break or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

11.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

11.10 Fuzzy Logic Development System (*fuzzy*TECH®-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions: a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design, and a full-featured *fuzzy*TECH-MP Edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

11.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 11-2.

TABLE 11-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator with PRO MATE Programmer, Assembler, Software Simulator, Samples, and your choice of Target Probe, .
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

 © 1995 Microchip Technology Inc.

12.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -40 to+ 125°C
Storage Temperature ..... -65°C to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD} and $\overline{MCLR}$) ..... -0.6V to V_{DD} +0.6V
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5 V
Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) ..... 0 to +14 V
Total power Dissipation (Note 1) ..... 1.0 W
Maximum Current out of V_{SS} pin 300mA
Maximum Current into V_{DD} pin 250mA
Input clamp current, I_{IK} (V_I<0 or V_I> V_{DD}) ..... ±20mA
Output clamp current, I_{OK} (V_0 <0 or V_0>V_{DD}) ..... ±20mA
Maximum Output Current sunk by any I/O pin 25mA
Maximum Output Current sourced by any I/O pin 25mA
Maximum Current sunk by PORTA and PORTB 200mA
Maximum Current sourced by PORTA and PORTB 200mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 12-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C62X-04	PIC16C62X-20	PIC16LC62X-04
RC	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3mA Max. at 5.5V I_{PD}: 20µA Max. at 4V WDT dis Freq: 4MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8mA typ. at 5.5V I_{PD}: 1.0µA typ. at 4.5V WDT dis Freq: 4MHz Max.	V_{DD}: 3.0V to 6.0V I_{DD}: 1.4mA typ. at 3.0V I_{PD}: 0.7µA typ. at 3V WDT dis Freq: 4MHz Max.
XT	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3mA Max. at 5.5V I_{PD}: 20µA Max. at 4V WDT dis Freq: 4MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8mA typ. at 5.5V I_{PD}: 1.0µA typ. at 4V WDT dis Freq: 4MHz Max.	V_{DD}: 3.0V to 6.0V I_{DD}: 1.4mA typ. at 3.0V I_{PD}: 0.7µA typ. at 3V WDT dis Freq: 4MHz Max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 9mA typ. at 5.5V I_{PD}: 1.0µA typ. at 4.0V WDT dis Freq: 4MHz Max.	V_{DD}: 4.5V to 5.5V I_{DD}: 20mA Max. at 5.5V I_{PD}: 1.0µA typ. at 4.5V WDT dis Freq: 20MHz Max.	Do not use in HS mode
LP	V_{DD}: 4.0V to 6.0V I_{DD}: 35µA typ. at 32 KHz, 3.0V I_{PD}: 1.0µA typ. at 4.0V WDT dis Freq: 200KHz typ.	Do not use in LP mode	V_{DD}: 3.0V to 6.0V I_{DD}: 32µA Max. at 32 KHz, 3.0V I_{PD}: 9µA Max. at 3.0V WDT dis Freq: 200KHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

12.1 DC CHARACTERISTICS: PIC16C62X-04 (Commercial, Industrial) PIC16C62X-20 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym.**	**Min.**	**Typ.†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	–	1.5	–	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	–	VSS	–	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	–	–	V/ms	See section on power-on reset for details
Brown-out Detect Voltage	BVDD	3.8	4.0	4.2	V	BODEN configuration fuse is erased
Supply Current (Note 2)	IDD	–	1.8	3.3	mA	XT and RC osc configuration FOSC = 4 MHz, VDD = 5.5V, WDT disabled (Note 4)
		–	35	70	µA	LP osc configuration, PIC16C62X-04 only FOSC = 32 KHz, VDD = 4.0V, WDT disabled
		–	9.0	20	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V, WDT disabled
WDT Current (Note 5)	ΔIWDT	–	6.0	20	µA	VDD=4.0V
Brown-out Detect Current (Note 5)	ΔIBOD	–	300	375	µA	$\overline{BOD}$ enabled, VDD = 5.0V
Comparator Current for each Comparator (Note 5)	ΔICOMP	–		100	µA	VDD = 4.0V
VREF Current (Note 5)	ΔIVREF	–		300	µA	VDD = 4.0V
Power Down Current (Note 3)	IPD	–	1.0	8	µA	VDD=4.0V, WDT disabled
WDT Current (Note 5)	ΔIWDT	–	6.0	20	µA	VDD=4.0V
Brown-out Detect Current (Note 5)	ΔIBOD	–	300	375	µA	$\overline{BOD}$ enabled, VDD = 5.0V
Comparator Current for each Comparator (Note 5)	ΔICOMP	–		100	µA	VDD = 4.0V
VREF Current (Note 5)	ΔIVREF	–		300	µA	VDD = 4.0V

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tri-stated™, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

Note 3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

Note 4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

Note 5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD measurement.

 © 1995 Microchip Technology Inc.

12.2 DC CHARACTERISTICS: PIC16LC62X-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0	-	6.0	V	XT, RC and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	–	1.5	–	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	–	VSS	–	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	–	–	V/ms	See section on power-on reset for details
Brown-out Detect Voltage	BVDD	3.8	4.0	4.2	V	BODEN configuration fuse is erased
Supply Current (Note 2)	IDD	–	1.4	2.5	mA	XT and RC osc configuration FOSC = 2 MHz, VDD = 3.0V, WDT disabled (Note 4)
		–	26	53	µA	LP osc configuration FOSC = 32 KHz, VDD = 3.0V, WDT disabled
WDT Current (Note 5)	ΔIWDT	–	6.0	15	µA	VDD=3.0V
Brown-out Detect Current (Note 5)	ΔIBOD	–	300	375	µA	$\overline{BOD}$ enabled, VDD = 5.0V
Comparator Current for each Comparator (Note 5)	ΔICOMP	–		100	µA	VDD = 3.0V
VREF Current (Note 5)	ΔIVREF	–		300	µA	VDD = 3.0V
Power Down Current (Note 3)	IPD	–	0.7	8	µA	VDD=3.0V, WDT disabled
WDT Current (Note 5)	ΔIWDT	–	6.0	15	µA	VDD=3.0V
Brown-out Detect Current (Note 5)	ΔIBOD	–	300	375	µA	$\overline{BOD}$ enabled, VDD = 5.0V
Comparator Current for each Comparator (Note 5)	ΔICOMP	–		100	µA	VDD = 3.0V
VREF Current (Note 5)	ΔIVREF	–		300	µA	VDD = 3.0V

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

Note 2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

Note 3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

Note 4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

Note 5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD measurement.

12.3 DC CHARACTERISTICS: PIC16C62X (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec Table 12-1 and Table 12-4				
Characteristic	**Sym**	**Min**	**Typ †**	**Max**	**Units**	**Conditions**
Input Low Voltage	VIL					
I/O ports						
		VSS	-	0.2VDD	V	
$\overline{MCLR}$, RA4/T0CKI,OSC1 (in RC mode)		Vss	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
Input High Voltage	VIH					
I/O ports			-			
		0.85VDD	-	VDD		
$\overline{MCLR}$ RA4/T0CKI		0.85VDD	-	VDD	V	
OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	200	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)	IIL					
I/O ports (Except RA)				±1	µA	VSS ≤ VPIN ≤ VDD, pin at high-impedance
RA		-	-	±0.5	µA	Vss ≤ VPIN ≤ VDD, Pin at high-impedance
RA4/T0CKI		-	-	±1	µA	Vss ≤ VPIN ≤ VDD
OSC1, $\overline{MCLR}$		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage	VOL					
I/O ports		-	-	0.6	V	IOL = 8.5mA, VDD=4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD=4.5V, -40°C to +85°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD=4.5V, -40°C to +85°C
Output High Voltage	VOH					
I/O ports (Except RA4) (Note 3)		VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +85°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +85°C
Open-Drain High Voltage	VOD			14	V	RA4 pin
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C62X be driven with external clock in RC mode.

Note 2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

Note 3: Negative current is defined as coming out of the pin.

 © 1995 Microchip Technology Inc.

TABLE 12-2: COMPARATOR SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 6.0V, -40°C < TA < 85°C, unless otherwise stated. Current consumption is specified in Table 12-1.

Characteristics	Sym	Min	Typ	Max	Units	Comments
Input offset voltage			+/- 5	+/- 10	mV	
Input common mode voltage		0		VDD - 1.5	V	
CMRR		-35*			db	
Response Time †				400*	ns	VDD = 5.0V
Comparator Mode Change to Output Valid				10*	µs	

* These parameters are characterized but not tested.
† Response time measured with one comparator input at (VDD — 1.5)/2 while other transitions from VSS to VDD.

TABLE 12-3: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 3.0V < VDD < 6.0V, -40°C < TA < 85°C, unless otherwise stated. Current consumption is specified in Table 12-1.

Characteristics	Sym	Min	Typ	Max	Units	Comments
Resolution		VDD/24		VDD/32	LSB	
Absolute Accuracy				1/4 LSB	V	
Unit Resistor Value (R)				2K*	Ω	See Figure 8-2.
Settling Time †				10	µs	

* These parameters are characterized but not tested.
† Settling time measured while VRR=1 and VR<3:0> transitions from 0000 to 1111.

12.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
ck	CLKOUT	osc	OSC1
io	I/O port	t0	T0CKI
mc	$\overline{\text{MCLR}}$		

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedence)	V	Valid
L	ow	Z	High Impedence

 © 1995 Microchip Technology Inc.

12.5 Timing Diagrams and Specifications

FIGURE 12-1: EXTERNAL CLOCK TIMING

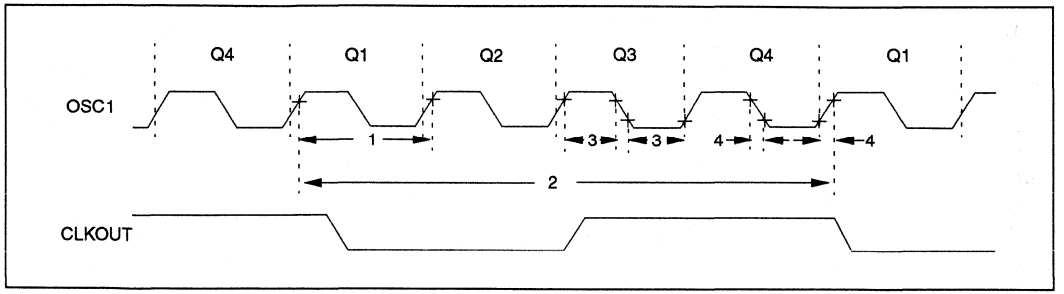


TABLE 12-4: EXTERNAL CLOCK TIMING REQUIREMENTS

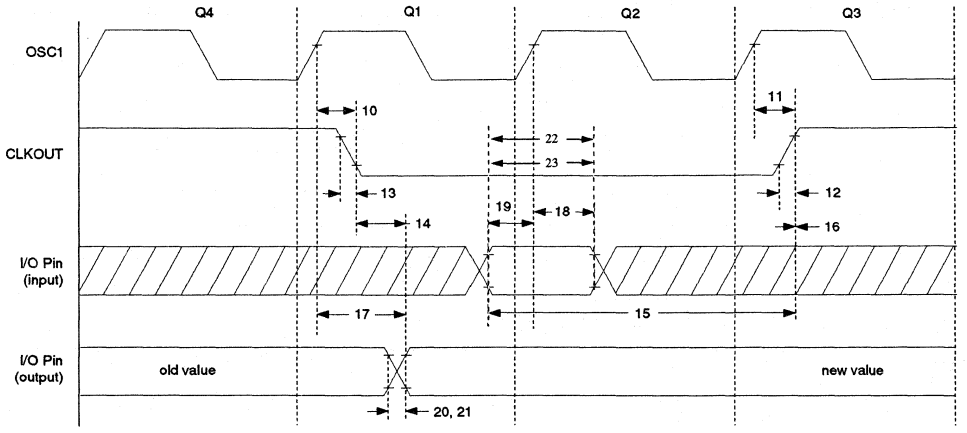
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fos	**External CLKIN Frequency (Note 1)**	DC	—	4	MHz	XT and RC osc mode, VDD=5V
			DC	—	20	MHz	HS osc mode
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode, VDD=5V
			0.1	—	4	MHz	XT osc mode
			1	—	20	MHz	HS osc mode
			DC	–	200	KHz	LP osc mode
1	Tosc	**External CLKIN Period (Note 1)**	250	—	—	ns	XT and RC osc mode
			50	—	—	ns	HS osc mode
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			50	—	1,000	ns	HS osc mode
			5	—	—	µs	LP osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	1.0	4/Fosc	DC	µs	
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 1995 Microchip Technology Inc.

FIGURE 12-2: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT

TABLE 12-5: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	25	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	TBD	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22	Tinp	INT pin high or low time	20	—	—	ns	
23	Trbp	RB<7:4> change interrupt high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc

 © 1995 Microchip Technology Inc.

FIGURE 12-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

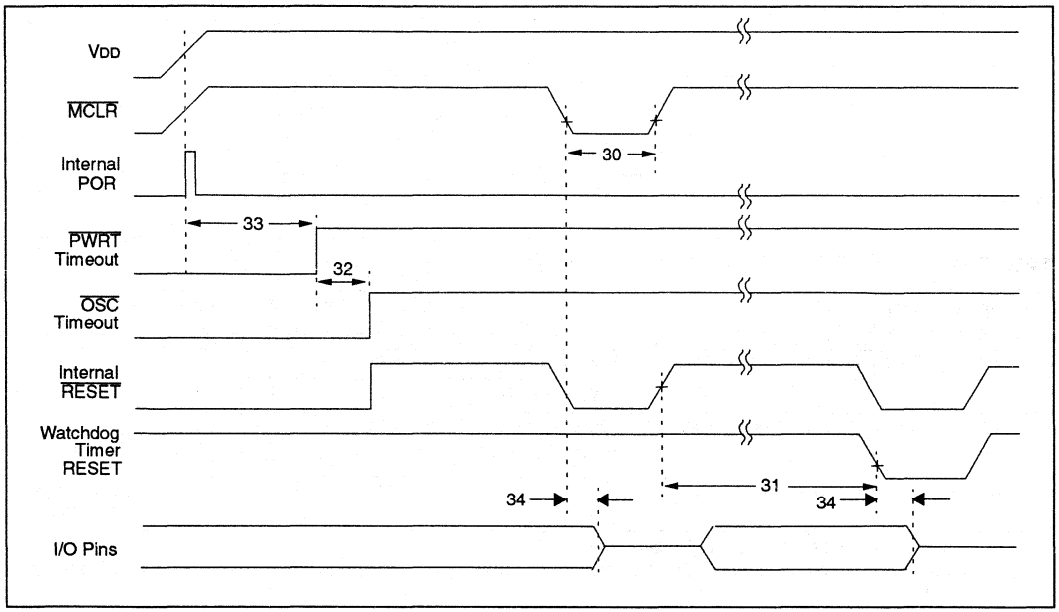


FIGURE 12-4: BROWN-OUT DETECT TIMING

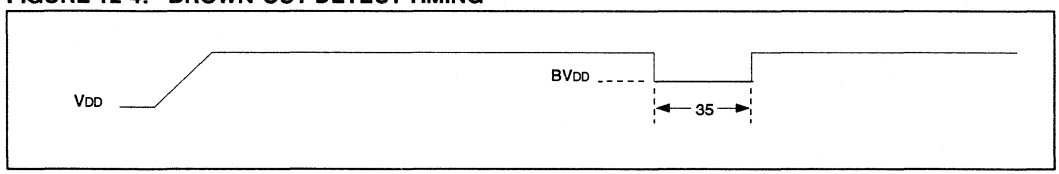


TABLE 12-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	1000	—	—	ns	VDD = 5V, -40°C to +85°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +85°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O high impedance from MCLR low			100	ns	
35	TBOD	Brown-out Detect Pulse Width	100			µs	3.8V ≤ VDD ≤ 4.2V

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 12-5: TIMER0 CLOCK TIMING

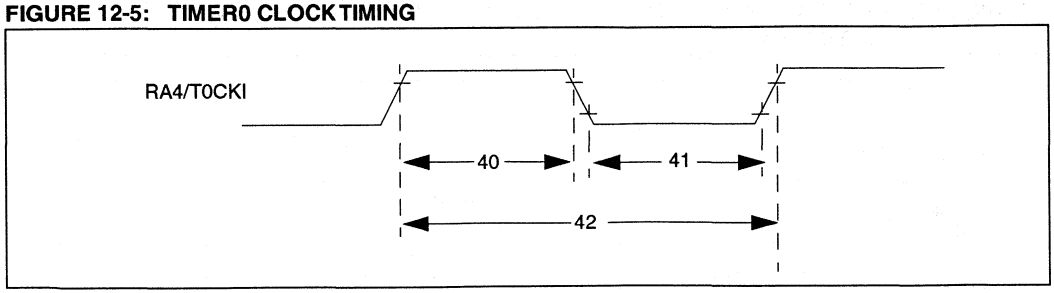


TABLE 12-7: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

13.0 DEVICE CHARACTERIZATION INFORMATION

Not Available at this time.

NOTES:

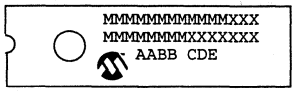
 © 1995 Microchip Technology Inc.

14.0 PACKAGING INFORMATION

For Package Dimensions please refer to the Packaging Section of the Data Book.

14.1 Package Marking Information

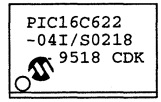
18L PDIP



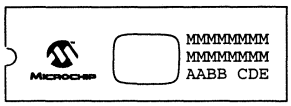
Example

18L SOIC

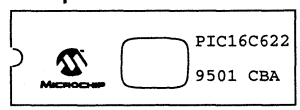
Example



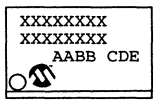
18L CERDIP



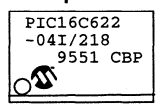
Example



20L SSOP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A.
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

 © 1995 Microchip Technology Inc.

APPENDIX A:

The following are the list of enhancements over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (up to 128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from status register.
3. Data memory paging is slightly redefined. Status register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers, oscillator start-up timer (OST) and power-up timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PortB has weak pull-ups and interrupt on change feature.
13. RTCC pin is also a port pin (RA4) and has a TRIS bit.
14. FSR is made a full eight bit register.
15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, /VPP, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on-Reset (POR) status bit and a Brown-out Detect status bit ($\overline{BO}$).
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. All inputs are now Schmitt trigger type.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S NEW

The format of certain sections of this data sheet have been changed to be consistent with other product families. These sections are:

6.0 Timer0 (TMR0) Module

11.0 Development Support

APPENDIX D: WHAT'S CHANGED

1. Table 1-1 has been upgraded.
2. A prescalar assignment note was added to Section 4.2.2.2 Option Register.
3. Figures 5-1, 5-2, 5-3 and 5-4 were modified to include edge symbols on the CK inputs of the data latches.
4. The instruction MOVF CMCON,F was added to Example 7-1.
5. Ceramic resonator tolerances were added to Table 9-1.
6. A note was added to Figure 9-7.
7. The MS oscillator configuration was deleted from Table 12-2 PC Characteristics.
8. The VIM specification for I/O posrts, $\overline{\text{MCLR}}$ and RA4/T0CKI was changed from 0.8VDD to 0.85VDD.
9. The CLKOUT waveform in Table 12-2 was modified.

 © 1995 Microchip Technology Inc.

APPENDIX E: PIC16/17 MICROCONTROLLERS

TABLE E-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals			Features			
	Maximum Frequency of Operation (MHz)	EPROM	RAM Data Memory (bytes)	Program Memory (bytes)	Timer Module(s)	Captures	PWMs	Serial Port(s) (SCI)	External Interrupts	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

TABLE E-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals							Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Program Memory (bytes)	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

 © 1995 Microchip Technology Inc.

TABLE E-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

E.1 Pin Compatibility

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin (20 pin)
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

 © 1995 Microchip Technology Inc.

INDEX

© 1995 Microchip Technology Inc.

LIST OF EXAMPLES

LIST OF FIGURES

 © 1995 Microchip Technology Inc.

LIST OF TABLES

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks, and SQTP is a service mark of Microchip Technology Incorporated.

ACCESS.bus is a trademark of the ACCESS.bus Industry Group (ABIG).

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

 © 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16C620/621/622 Literature Number: DS30235B

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject matter: _____

5. What deletions from the data sheet could be made without affecting the overall usefulness? _____

6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

PIC16C62X Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)		
Package:	P	=	PDIP
	SO	=	SOIC (Gull Wing, 300 mil body)
	SS	=	SSOP (209 mil)
	JW	=	Windowed CERDIP
Temperature Range:	-	=	0°C to +70°C
	I	=	–40°C to +85°C)
Frequency Range:	04	=	200kHz (LP osc)
	04	=	4 MHz (XT and RC osc)
	10	=	20 MHz (HS osc)
Device:	PIC16C62X :VDD range 4.0V to 6.0V PIC16C62T :VDD range 4.0V to 6.0V (Tape and Reel) PIC16LC62X:VDD range3.0V to 6.0V PIC16LC62XT:VDD range3.0V to 6.0V (Tape and Reel)		

Examples:

a) PIC16C62X - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301

b) PIC16LC62X- 04I/SO = Industrial temp., SOIC package, 200kHz, extended VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

 © 1995 Microchip Technology Inc.

MICROCHIP

PIC16C6X

8-Bit CMOS Microcontrollers

Devices included in this Data Sheet:
- PIC16C65
- PIC16C64
- PIC16C61

High-Performance RISC-like CPU
- Only 35 single word instructions to learn
- All single cycle instructions (200ns) except for program branches which are two-cycle
- Operating speed: DC - 20MHz clock input
  DC - 200ns instruction cycle

Device	Program Memory	Data Memory	I/O
PIC16C65	4K	192	33
PIC16C64	2K	128	33
PIC16C61	1K	36	13

- Interrupt capability
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes

Peripheral Features of all PIC16C6X
- TMR0: 8-bit timer/counter with 8-bit prescaler
- High I/O current sink/source capability (25 mA/20 mA)

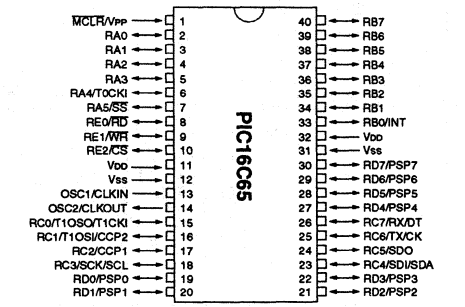
Peripheral Features of the PIC16C65/64 only
- One pin that can be configured as capture input, PWM output, or compare output
- Capture is 16-bit, max resolution 12.5ns, compare is 16-bit, max resolution 200ns, 10-bit PWM resolution
- TMR1: 16-bit timer/counter (time-base for capture/compare). TMR1 can be incremented during sleep via external crystal/clock (for real-time clock)
- TMR2: 8-bit timer/counter with 8-bit period register (time-base for PWM), prescaler and postscaler
- Parallel Slave Port (PSP): 8-bit wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (microprocessor bus interface)
- Synchronous serial port (SSP) with SPI and I^2C™/ACCESS.bus™

Peripheral Features of the PIC16C65 only
- Additional pin that can be configured as capture input, PWM output, or compare output
- Serial Communications Interface (SCI)/USART

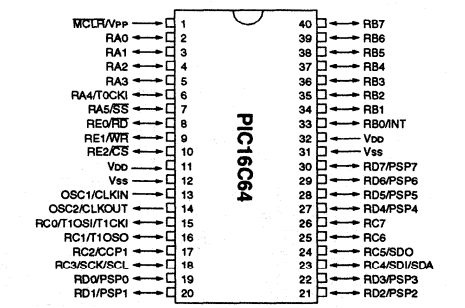
Special Microcontroller Features
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Fuse selectable oscillator options
- Serial in-system programming (via two pins)

PACKAGE TYPES

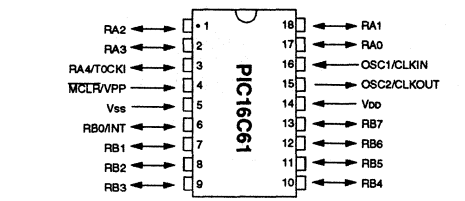
PDIP, Windowed CERDIP



PDIP, Windowed CERDIP



PDIP, SOIC, Windowed CERDIP

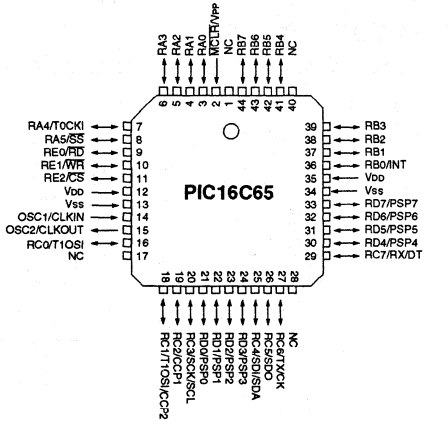


CMOS Technology
- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range - 3.0V to 6.0V
- Commercial, Industrial, and Automotive Temp. Range
- Low-power consumption- < 2mA @ 5V, 4 MHz
  - 15μA typical @ 3V, 32 KHz
  - <1μA typical standby current

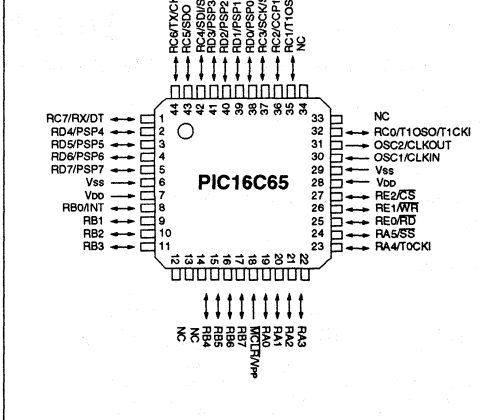
© 1995 Microchip Technology Inc.

PACKAGE TYPES

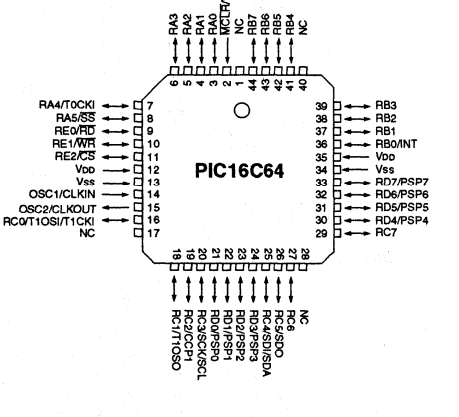
PLCC



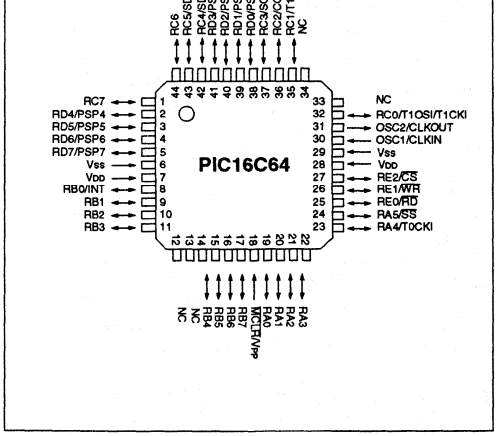
MQFP



PLCC



MQFP



© 1995 Microchip Technology Inc.

TABLE OF CONTENTS

2

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The PIC16C6X is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC-like architecture. The PIC16CXX has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC16C65 device has 192 bytes of RAM and 33 I/O pins. In addition several peripheral features are available including: three timer/counters, two capture/compare/PWM modules and two serial ports. The synchronous serial port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I^2C) bus. The Serial Communications Interface (SCI) can be configured as either synchronous or asynchronous (USART). An 8-bit Parallel Slave Port is also provided.

The PIC16C64 device has 128 bytes of RAM and up to 33 I/O pins. In addition several peripheral features are available including: three timer/counters, one capture/compare/PWM module and one serial port. The synchronous serial port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I^2C) bus. An 8-bit Parallel Slave Port is also provided.

The PIC16C61 device has 36 bytes of RAM and up to 13 I/O pins. In addition a timer/counter is available.

The PIC16C6X device has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake up the chip from SLEEP through several external and internal interrupts and reset.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

Table 1-1 shows the features of the PIC16C6X family.

Table 1-2 shows information on which sections apply to the specific devices.

Simplified block diagrams of the PIC16C65, PIC16C64 and PIC16C61 are shown in Figure 3-1, Figure 3-2, and Figure 3-3 respectively.

The PIC16C6X family fits perfectly in applications ranging from high speed automotive and appliance control to low-power remote sensors, keyboards and telecom processors. The EPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease-of-use and I/O flexibility make the PIC16C6X very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, serial communication, capture and compare, PWM functions and co-processor applications).

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to PIC16CXX family of devices (see Appendix B).

1.2 Development Support

The PIC16CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals							Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Data Memory (bytes)	Program Memory	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	1	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	—	4 ch	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	5 ch	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	8 ch	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE 1-2: PIC16C6X FUNCTION SECTIONS

FUNCTION	SECTION		
	PIC16C65	PIC16C64	PIC16C61
I/O Ports	5.0	5.0	5.0
PORTA	5.1	5.1	5.1
PORTB	5.2	5.2	5.2
PORTC	5.3	5.3	–
PORTD	5.4	5.4	–
PORTE	5.5	5.5	–
I/O Programming	5.6	5.6	5.6
Parallel Slave Port	5.7	5.7	–
Timer Module Overview	6.0	6.0	6.0
Timer0	7.0	7.0	7.0
Timer1	8.0	8.0	–
Timer2	9.0	9.0	–
Capture/Compare/PWM	10.0	10.0	–
Synchronous Serial Port (SPI/I^2C)	11.0	11.0	–
Serial Communications Interface (USART)	12.0	–	–

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

2.0 PIC16C6X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C6X Product Selection System section at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PICSTART® and PRO MATE™ programmers both support programming of the PIC16C6X. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory using the same bus. Separating program and data busses further allows instructions to be sized differently than 8-bit wide data word. Instruction op-codes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (see Example 3-1). Consequently, all instructions execute in a single cycle (200ns @ 20MHz) except for program branches.

The PIC16C65 addresses 4K x 14 program memory. The PIC16C64 addresses 2K x 14 program memory and the PIC16C61 addresses 1K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of "special optimal situations" makes programming with the PIC16CXX simple yet efficient. In addition, the learning curve is significantly reduced.

The PIC16CXX device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

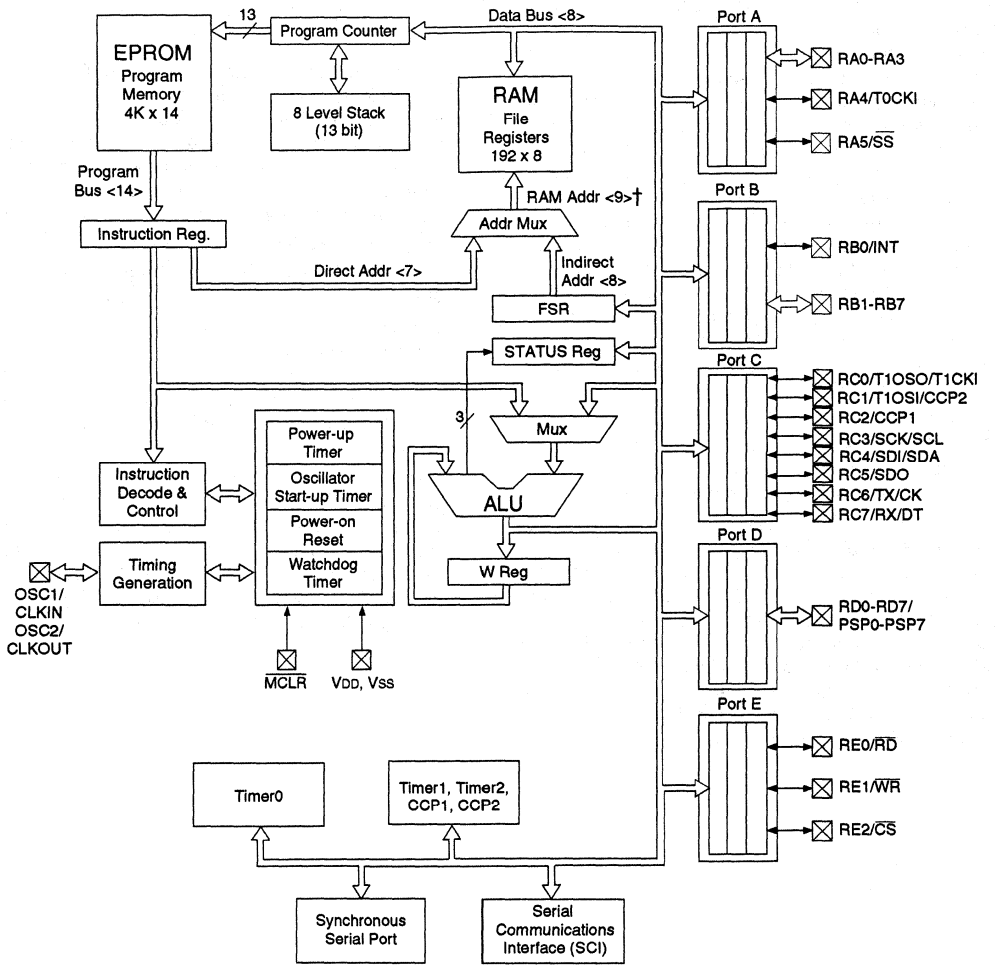
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending upon the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram for the PIC16C65 is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1, a simplified block diagram for the PIC16C64 is shown in Figure 3-2, its corresponding pin description is shown in Table 3-1, and a simplified block diagram for the PIC16C61 is shown in Figure 3-3, its corresponding pin description is shown in Table 3-1.

© 1995 Microchip Technology Inc.

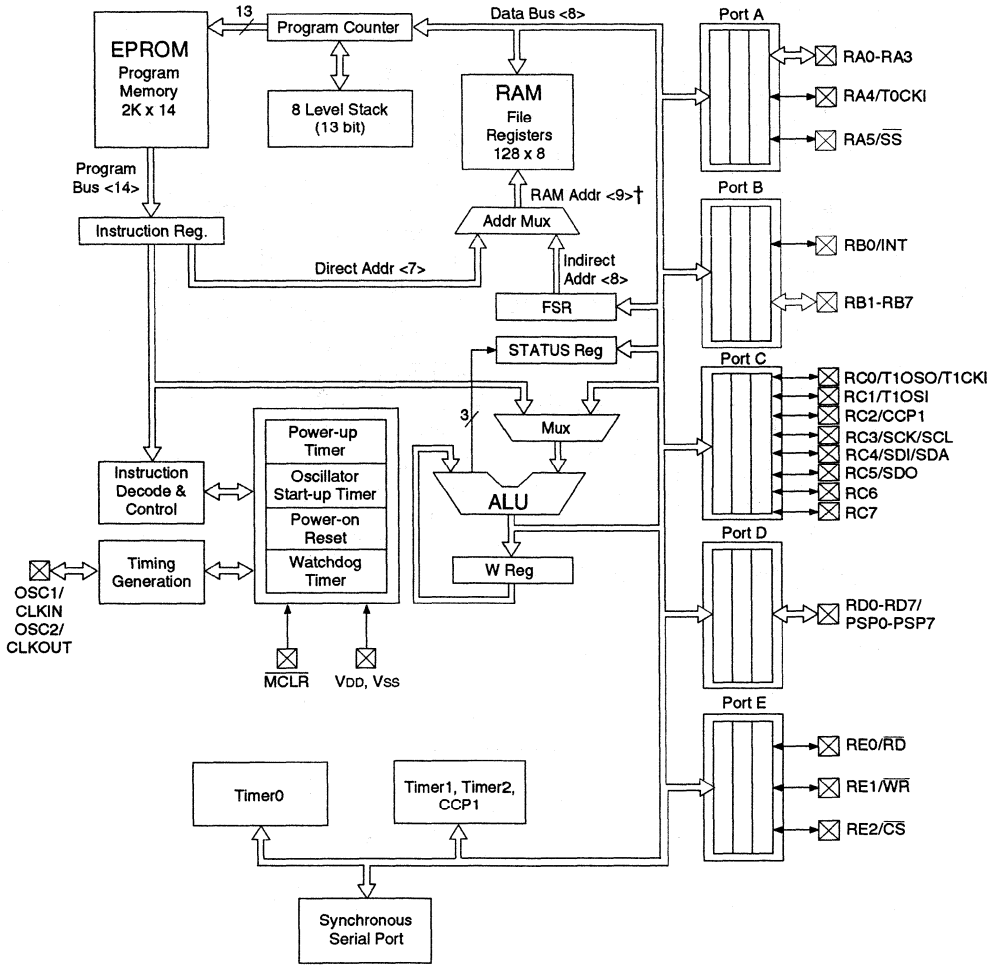
FIGURE 3-1: PIC16C65 BLOCK DIAGRAM



† Higher order bits are from STATUS register.

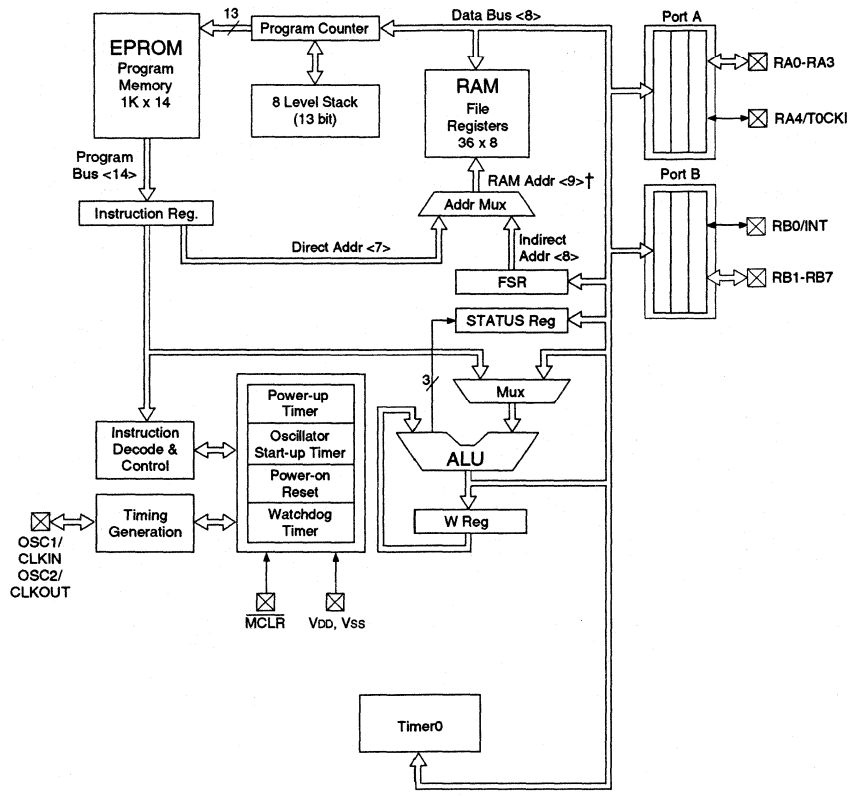
© 1995 Microchip Technology Inc.

FIGURE 3-2: PIC16C64 BLOCK DIAGRAM



† Higher order bits are from STATUS register.

FIGURE 3-3: PIC16C61 BLOCK DIAGRAM



† Higher order bits are from STATUS register.

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C65 PINOUT DESCRIPTION

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS[4]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
						PORTA is a bidirectional I/O port.
RA0	2	3	19	I/O	TTL	
RA1	3	4	20	I/O	TTL	
RA2	4	5	21	I/O	TTL	
RA3	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
RA5/$\overline{SS}$	7	8	24	I/O	TTL	Slave select for the synchronous serial port.
						PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	Interrupt on change pin.
RB5	38	42	15	I/O	TTL	Interrupt on change pin.
RB6	39	43	16	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	40	44	17	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0/T1OSO/T1CKI can also be selected as a Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1/T1OSI/CCP2 can also be selected as a Timer1 oscillator input or Capture 2, input/Compare 2 output/PWM 2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2/CCP1 can also be selected as a capture1 input/compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5/SDO can also be selected as the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6/TX/CK can also be selected as Asynchronous Transmit or SCI Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7/RX/DT can also be selected as the Asynchronous Receive or SCI Synchronous Data.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C65 PINOUT DESCRIPTION (CONT.)

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
						PORTD is a bidirectional I/O port or parallel slave or interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL[3]	
RD1/PSP1	20	22	39	I/O	ST/TTL[3]	
RD2/PSP2	21	23	40	I/O	ST/TTL[3]	
RD3/PSP3	22	24	41	I/O	ST/TTL[3]	
RD4/PSP4	27	30	2	I/O	ST/TTL[3]	
RD5/PSP5	28	31	3	I/O	ST/TTL[3]	
RD6/PSP6	29	32	4	I/O	ST/TTL[3]	
RD7/PSP7	30	33	5	I/O	ST/TTL[3]	
						PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$	8	9	25	I/O	ST/TTL[3]	RE0/$\overline{RD}$ read control for parallel slave port5.
RE1/$\overline{WR}$	9	10	26	I/O	ST/TTL[3]	RE1/$\overline{WR}$ write control for parallel slave port6.
RE2/$\overline{CS}$	10	11	27	I/O	ST/TTL[3]	RE2/$\overline{CS}$ select control for parallel slave port.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-2: PIC16C64 PINOUT DESCRIPTION

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS[4]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
						PORTA is a bidirectional I/O port.
RA0	2	3	19	I/O	TTL	
RA1	3	4	20	I/O	TTL	
RA2	4	5	21	I/O	TTL	
RA3	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
RA5/$\overline{SS}$	7	8	24	I/O	TTL	Slave select for the synchronous serial port.
						PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	Interrupt on change pin.
RB5	38	42	15	I/O	TTL	Interrupt on change pin.
RB6	39	43	16	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	40	44	17	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
						PORTC is a bidirectional I/O port.
RC0/T1OSI/T1CKI	15	16	32	I/O	ST	RC0/T1OSI/T1CKI can also be selected as a Timer1 oscillator input/Timer1 clock input.
RC1/T1OSO	16	18	35	I/O	ST	RC1/T1OSI can also be selected as a Timer1 oscillator output.
RC2/CCP1	17	19	36	I/O	ST	RC2/CCP1 can also be selected as a capture1 input/compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5/SDO can also be selected as the SPI Data Out (SPI mode).
RC6	25	27	44	I/O	ST	
RC7	26	29	1	I/O	ST	

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-2: PIC16C64 PINOUT DESCRIPTION (CONT.)

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
						PORTD is a bidirectional I/O port or parallel slave or interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL[3]	
RD1/PSP1	20	22	39	I/O	ST/TTL[3]	
RD2/PSP2	21	23	40	I/O	ST/TTL[3]	
RD3/PSP3	22	24	41	I/O	ST/TTL[3]	
RD4/PSP4	27	30	2	I/O	ST/TTL[3]	
RD5/PSP5	28	31	3	I/O	ST/TTL[3]	
RD6/PSP6	29	32	4	I/O	ST/TTL[3]	
RD7/PSP7	30	33	5	I/O	ST/TTL[3]	
						PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$	8	9	25	I/O	ST/TTL[3]	RE0/$\overline{RD}$ read control for parallel slave port5.
RE1/$\overline{WR}$	9	10	26	I/O	ST/TTL[3]	RE1/$\overline{WR}$ write control for parallel slave port6.
RE2/$\overline{CS}$	10	11	27	I/O	ST/TTL[3]	RE2/$\overline{CS}$ select control for parallel slave port.
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-3: PIC16C61 PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS[3]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bidirectional I/O port.
RA0	17	17	I/O	TTL	
RA1	18	18	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
					PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
VSS	5	5	P	—	Ground reference for logic and I/O pins.
VDD	14	14	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clock and instruction execution flow is shown in Figure 3-4.

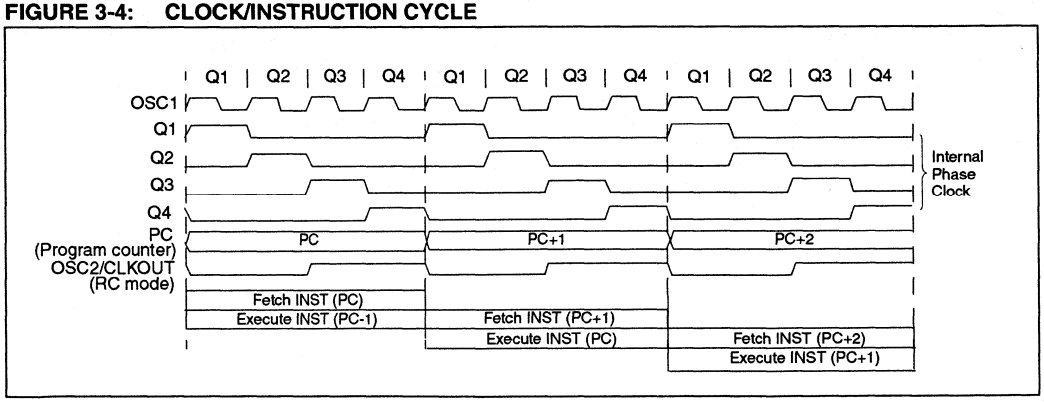
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-1).

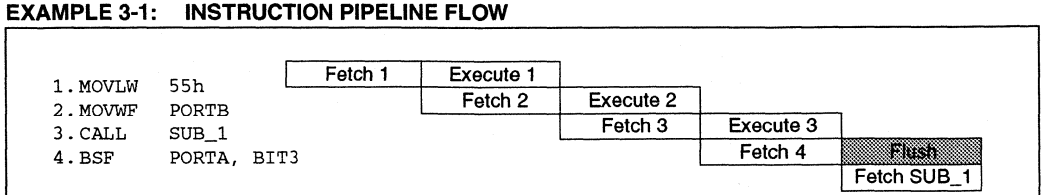
A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-4: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetched instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C6X family has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C65, only the first 4K x 14 (0000-0FFFh) are physically implemented. For the PIC16C64, only the first 2K x 14 (0000-07FFh) are physically implemented and for the PIC16C61, only the first 1K x 14 (0000-03FFh) is physically implemented. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 4-1, Figure 4-2 and Figure 4-3).

FIGURE 4-1: PIC16C65 PROGRAM MEMORY MAP AND STACK

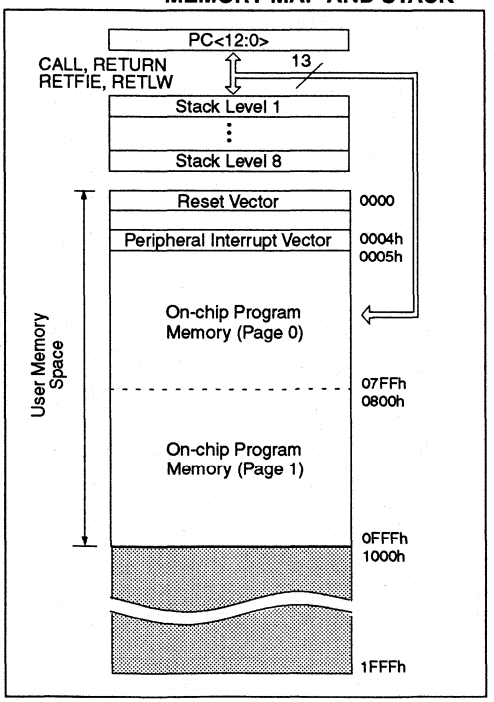


FIGURE 4-2: PIC16C64 PROGRAM MEMORY MAP AND STACK

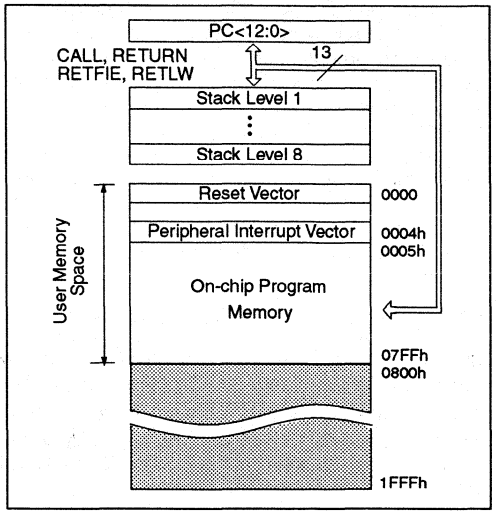
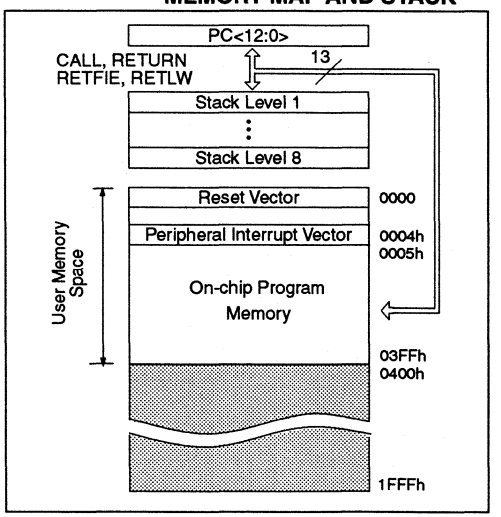


FIGURE 4-3: PIC16C61 PROGRAM MEMORY MAP AND STACK



2

4.2 Data Memory Organization

The data memory (see Figure 4-4, Figure 4-5 and Figure 4-6) is partitioned into two Banks which contain the general purpose registers and the special function registers. Bank 0 is selected when the RP0 bit in the STATUS register is cleared. Bank 1 is selected when the RP0 bit in the STATUS register is set. Each Bank extends up to 7Fh (128 bytes). The lower locations of each Bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers implemented as static RAM. Some Special Function Registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly, or indirectly through the file select register FSR (see Section 4.4).

The general purpose register locations 8Ch–AFh of Bank 1 on the PIC16C61 are not physically implemented. These locations are mapped into 0Ch – 2Fh of Bank 0.

FIGURE 4-4: PIC16C65 REGISTER FILE MAP

File Address	Bank 0	Bank 1	
00	Indirect addr.(*)	Indirect addr.(*)	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07	PORTC	TRISC	87
08	PORTD	TRISD	88
09	PORTE	TRISE	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C	PIR1	PIE1	8C
0D	PIR2	PIE2	8D
0E	TMR1L	PCON	8E
0F	TMR1H		8F
10	T1CON		90
11	TMR2		91
12	T2CON	PR2	92
13	SSPBUF	SSPADD	93
14	SSPCON	SSPSTAT	94
15	CCPR1L		95
16	CCPRIH		96
17	CCP1CON		97
18	RCSTA	TXSTA	98
19	TXREG	SPBRG	99
1A	RCREG		9A
1B	CCPR2L		9B
1C	CCPR2H		9C
1D	CCP2CON		9D
1E			9E
1F			9F
20 – 7F	General Purpose Register	General Purpose Register	A0 – FF

* Not a physical register

▨ Unimplemented data memory location; reads as '0's.

© 1995 Microchip Technology Inc.

FIGURE 4-5: PIC16C64 REGISTER FILE MAP

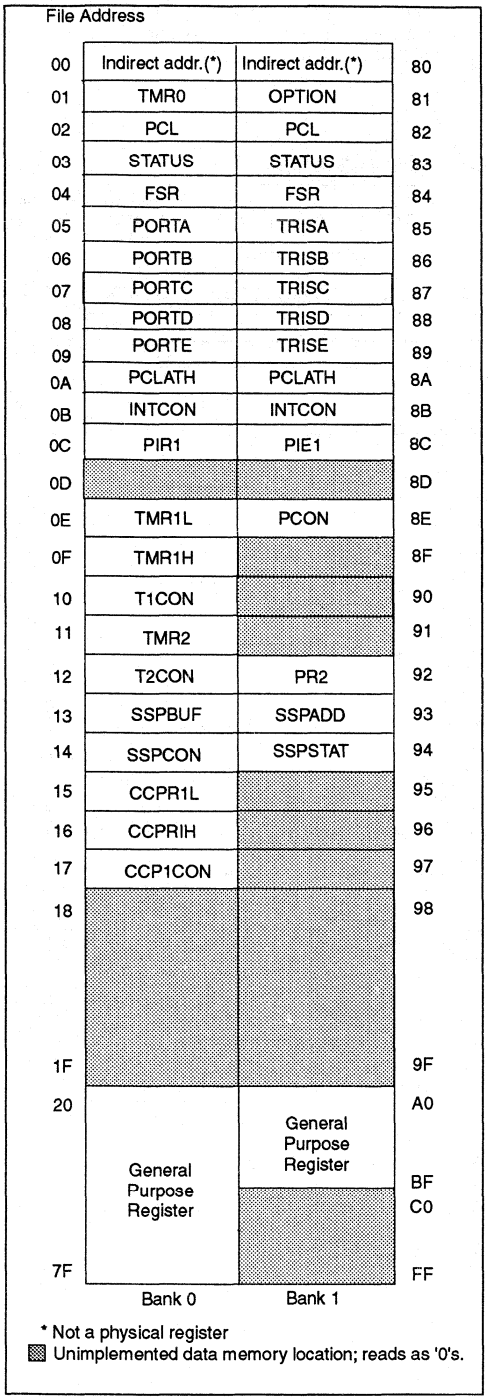
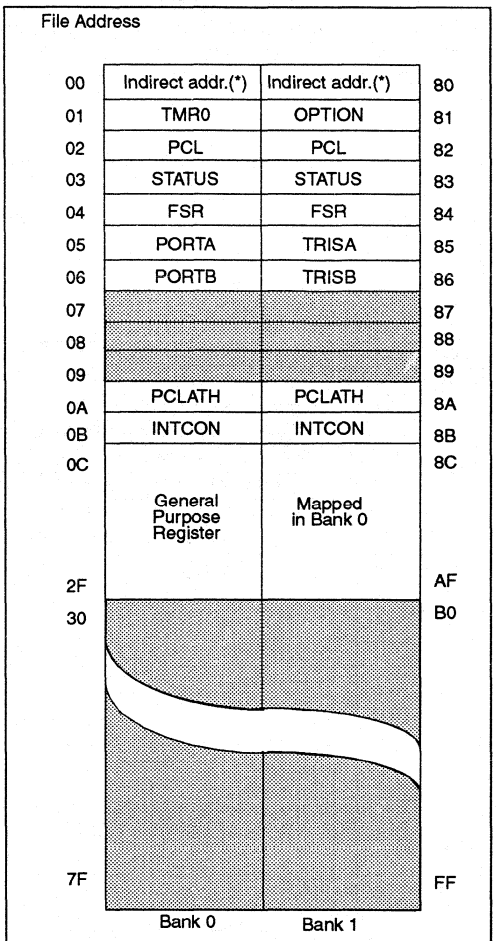


FIGURE 4-6: PIC16C61 REGISTER FILE MAP



2

4.2.2 SPECIAL FUNCTION REGISTERS:

The Special Function Registers are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (see Table 4-2 and Table 4-2). These registers are static RAM.

The special registers can be classified into two sets. The special registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C65 AND PIC16C64

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
Bank 0											
00†	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01	TMR0	Timer0								xxxx xxxx	uuuu uuuu
02†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
03†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--xx xxxx	--uu uuuu
06	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09	PORTE	PORTE Data Latch when written: PORTE pins when read								---- -xxx	---- -uuu
0A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the (PC)[1]					---0 0000	---0 0000
0B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0C	PIR1	PSPIF	—	RCIF††	TXIF††	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0D††	PIR2	—	†††	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
0E	TMR1L	Timer1 Least Significant Byte								xxxx xxxx	uuuu uuuu
0F	TMR1H	Timer1 Most Significant Byte								xxxx xxxx	uuuu uuuu
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11	TMR2	Timer2								0000 0000	0000 0000
12	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15	CCPR1L	Capture/Compare/Duty Cycle Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16	CCPR1H	Capture/Compare/Duty Cycle Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18††	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00x
19††	TXREG	SCI Transmit Data Register								0000 0000	0000 0000
1A††	RCREG	SCI Receive Data Register								0000 0000	0000 0000
1B††	CCPR2L	Capture/Compare/Duty Cycle Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1C††	CCPR2H	Capture/Compare/Duty Cycle Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1D††	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	xx00 0000	xx00 0000

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

††: These registers or bits are not physically implemented on the PIC16C64 and are read as "0".

†††These bits are reserved on the PIC16C65.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

© 1995 Microchip Technology Inc.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C65 AND PIC16C64 (CONT.)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
1E-1F	—	Unimplemented								---- ----	---- ----
Bank 1											
80†	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								–	–
81	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
83†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89	TRISE	IBF	OBF	IBOV	PSP-MODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
8A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the (PC)[1]					---0 0000	---0 0000
8B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8C	PIE1	PSPIE	†††	RCIE††	TXIE††	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8D††	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
8E	PCON	—	—	—	—	—	—	POR	—	---- --0-	---- --u-
8F	—	Unimplemented								---- ----	---- ----
90	—	Unimplemented								---- ----	---- ----
91	—	Unimplemented								---- ----	---- ----
92	PR2	Timer2 Period Register								1111 1111	1111 1111
93	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	--00 0000	--00 0000
95	—	Unimplemented								---- ----	---- ----
96	—	Unimplemented								---- ----	---- ----
97	—	Unimplemented								---- ----	---- ----
98††	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8	0000 -010	0000 -010
99††	SPBRG	Baud Rate Register								0000 0000	0000 0000
9A	—	Unimplemented								---- ----	---- ----
9B	—	Unimplemented								---- ----	---- ----
9C	—	Unimplemented								---- ----	---- ----
9D	—	Unimplemented								---- ----	---- ----
9E	—	Unimplemented								---- ----	---- ----
9F	—	Unimplemented								---- ----	---- ----

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

††: These registers or bits are not physically implemented on the PIC16C64 and are read as "0".

†††These bits are reserved on the PIC16C65.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

TABLE 4-2: SPECIAL REGISTERS FOR THE PIC16C61

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
Bank 0											
00†	INDF (indirect address)	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01	TMR0	Timer0								xxxx xxxx	uuuu uuuu
02†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
03†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05	PORTA	—	—	—	PORTA Data Latch when written: PORTA pins when read					---x xxxx	---u uuuu
06	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07	—	Unimplemented								---- ----	---- ----
08	—	Unimplemented								---- ----	---- ----
09	—	Unimplemented								---- ----	---- ----
0A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the (PC)[1]					---0 0000	---0 0000
0B†	INTCON	GIE	—	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80†	INDF (indirect address)	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
83†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85	TRISA	—	—	—	PORTA Data Direction Register					---1 1111	---1 1111
86	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87	—	Unimplemented								---- ----	---- ----
88	—	Unimplemented								---- ----	---- ----
89	—	Unimplemented								---- ----	---- ----
8A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter (PC) (Note 1)					---0 0000	---0 0000
8B†	INTCON	GIE	—	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

© 1995 Microchip Technology Inc.

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 4-7, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the status register as 000UU1UU (where U = unchanged).

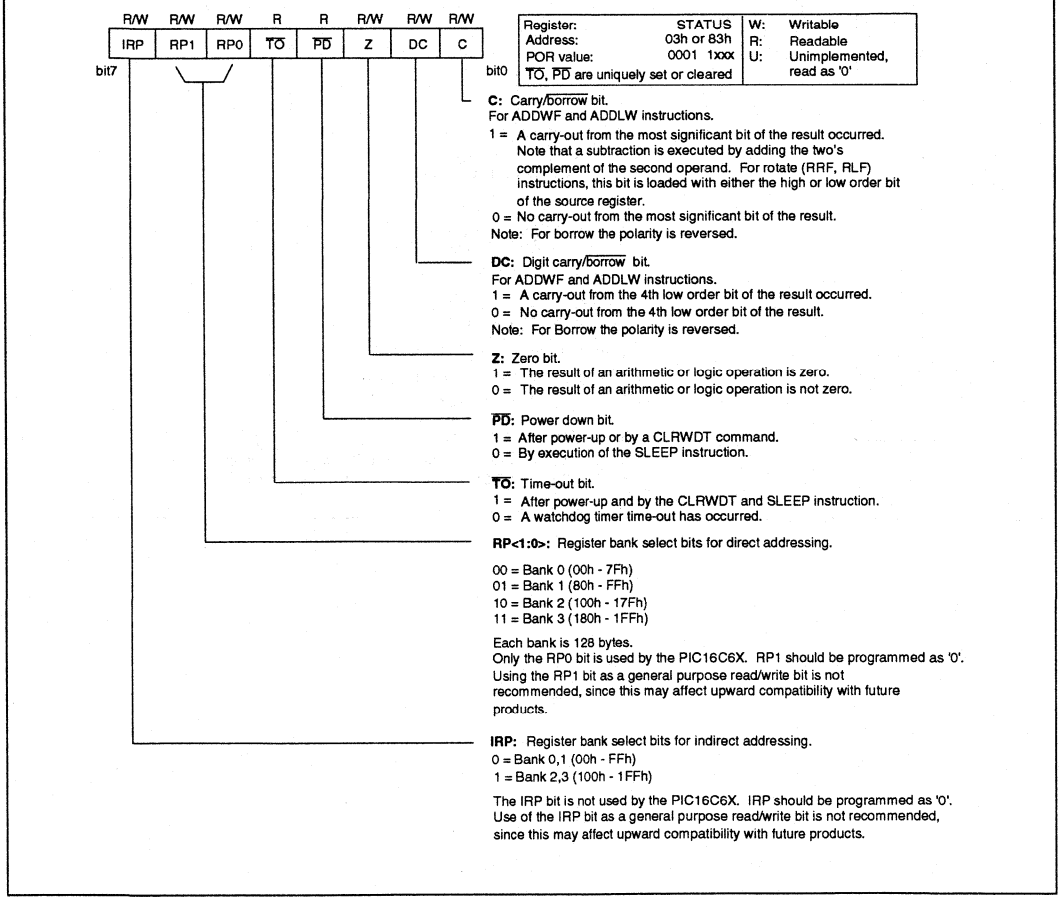
It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the status registers because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the "Instruction Set Summary

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C6X and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

2

FIGURE 4-7: STATUS REGISTER

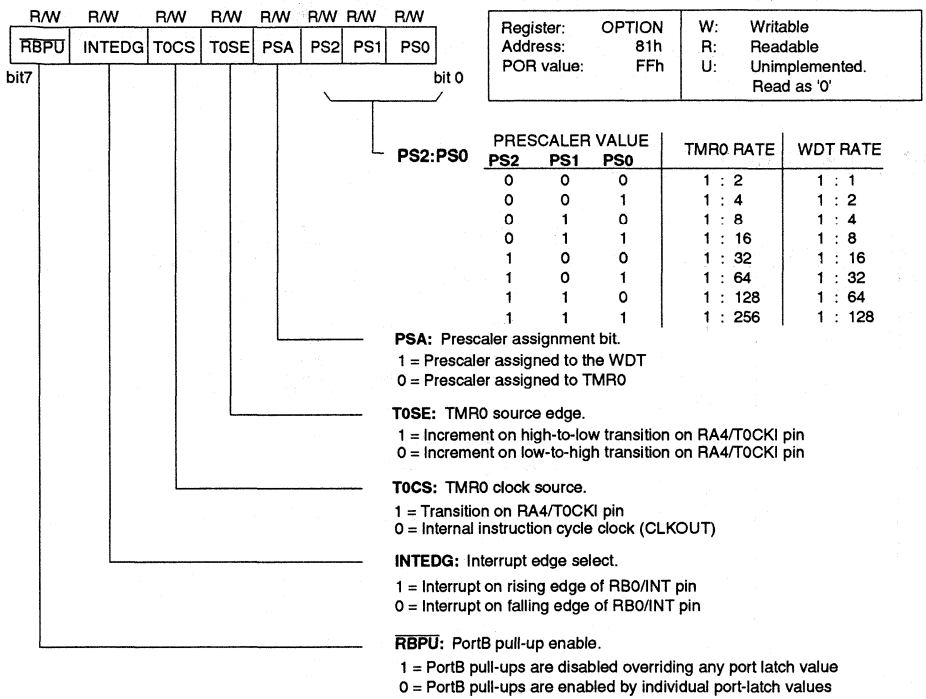


4.2.2.2 OPTION REGISTER

The OPTION register, shown in Figure 4-8, is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for Timer0, assign the prescaler to the WDT (PSA=1)

FIGURE 4-8: OPTION REGISTER



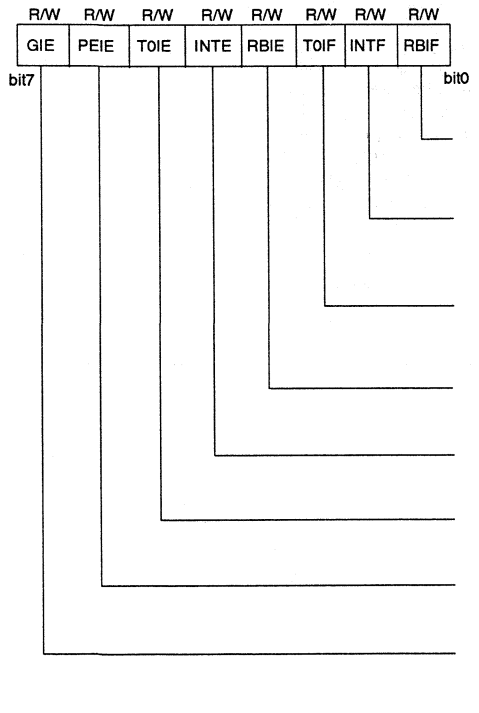
© 1995 Microchip Technology Inc.

4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains the various enable and flag bits for the Timer0 overflow, RB port change and external INT pin interrupts. Figure 4-9 shows the bits for the INTCON Register for the PIC16C65 and PIC16C64. Figure 4-10 shows the INTCON register for the PIC16C61.

Note: The T0IF, INIF, or RBIF will be set by the specified condition even if the corresponding Interrupt Enable Bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled).

FIGURE 4-9: INTCON REGISTER FOR PIC16C65 AND PIC16C64 ONLY



Register:	INTCON	W:	Writable
Address:	0Bh or 8Bh	R:	Readable
POR value:	0000 000xb	U:	Unimplemented, read as '0'

RBIF: RB port change interrupt flag.
1 = When at least one of the RB<7:4> inputs change. Must be cleared by software.
0 = None of the RB<7:4> inputs have changed.

INTF: External interrupt flag.
1 = The external INT interrupt has occurred. Must be cleared by software.
0 = The INT interrupt did not occur

T0IF: TMR0 overflow interrupt flag.
1 = The TMR0 has overflowed. Must be cleared by software.
0 = TMR0 did not overflow.

RBIE: RB port change interrupt enable bit.
1 = Enables RBIF interrupt
0 = Disables RBIF interrupt

INTE: INT interrupt enable bit.
1 = Enables INTF interrupt
0 = Disables INTF interrupt

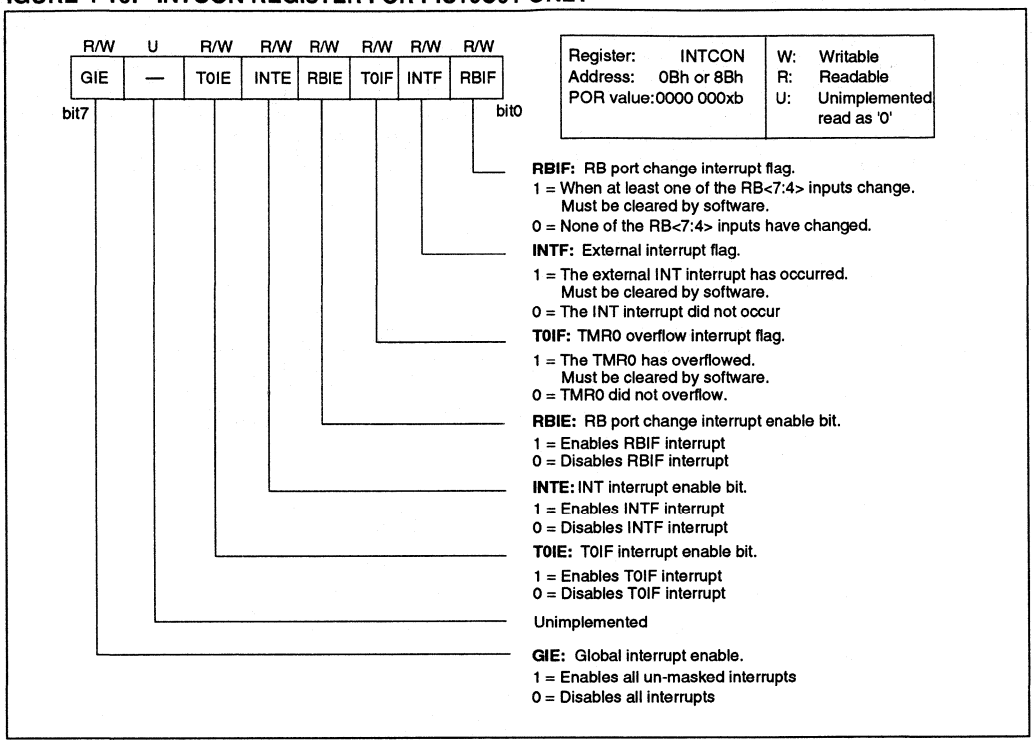
T0IE: T0IF interrupt enable bit.
1 = Enables T0IF interrupt
0 = Disables T0IF interrupt

PEIE: Peripheral interrupt enable bit.
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

GIE: Global interrupt enable.
1 = Enables all un-masked interrupts
0 = Disables all interrupts

© 1995 Microchip Technology Inc.

FIGURE 4-10: INTCON REGISTER FOR PIC16C61 ONLY



© 1995 Microchip Technology Inc.

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the Peripheral interrupts. Figure 4-11 shows the PIE1 register for the PIC16C65 and Figure 4-12 shows the PIE1 register for the PIC16C64.

Note: INTCON<6> must be enabled to enable any interrupt in PIE1.

FIGURE 4-11: PIE1 REGISTER FOR PIC16C65 ONLY

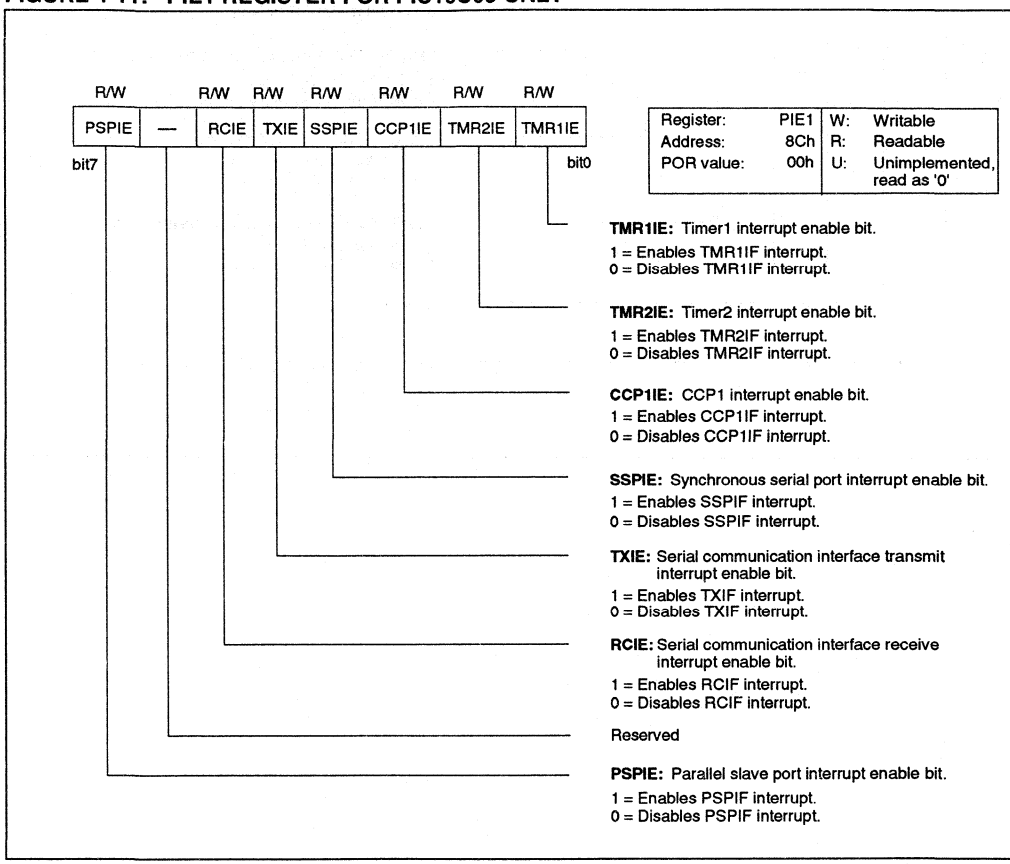
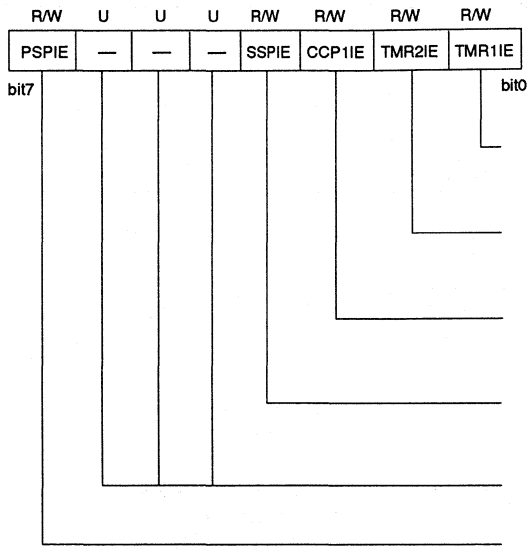


FIGURE 4-12: PIE1 REGISTER FOR PIC16C64 ONLY



Register:	PIE1	W:	Writable
Address:	8Ch	R:	Readable
POR value:	00h	U:	Unimplemented, read as '0'

TMR1IE: Timer1 interrupt enable bit.
1 = Enables TMR1IF interrupt.
0 = Disables TMR1IF interrupt.

TMR2IE: Timer2 interrupt enable bit.
1 = Enables TMR2IF interrupt.
0 = Disables TMR2IF interrupt.

CCP1IE: CCP1 interrupt enable bit.
1 = Enables CCP1IF interrupt.
0 = Disables CCP1IF interrupt.

SSPIE: Synchronous serial port interrupt enable bit.
1 = Enables SSPIF interrupt.
0 = Disables SSPIF interrupt.

Unimplemented

PSPIE: Parallel slave port interrupt enable bit.
1 = Enables PSPIF interrupt.
0 = Disables PSPIF interrupt.

© 1995 Microchip Technology Inc.

4.2.2.5 PIR1 REGISTER

This register contains the individual flag bits for the Peripheral interrupts. Figure 4-13 shows the PIR1 register for the PIC16C65 and Figure 4-14 shows the PIR1 register for the PIC16C64.

Note: These bits will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

FIGURE 4-13: PIR1 REGISTER FOR PIC16C65 ONLY

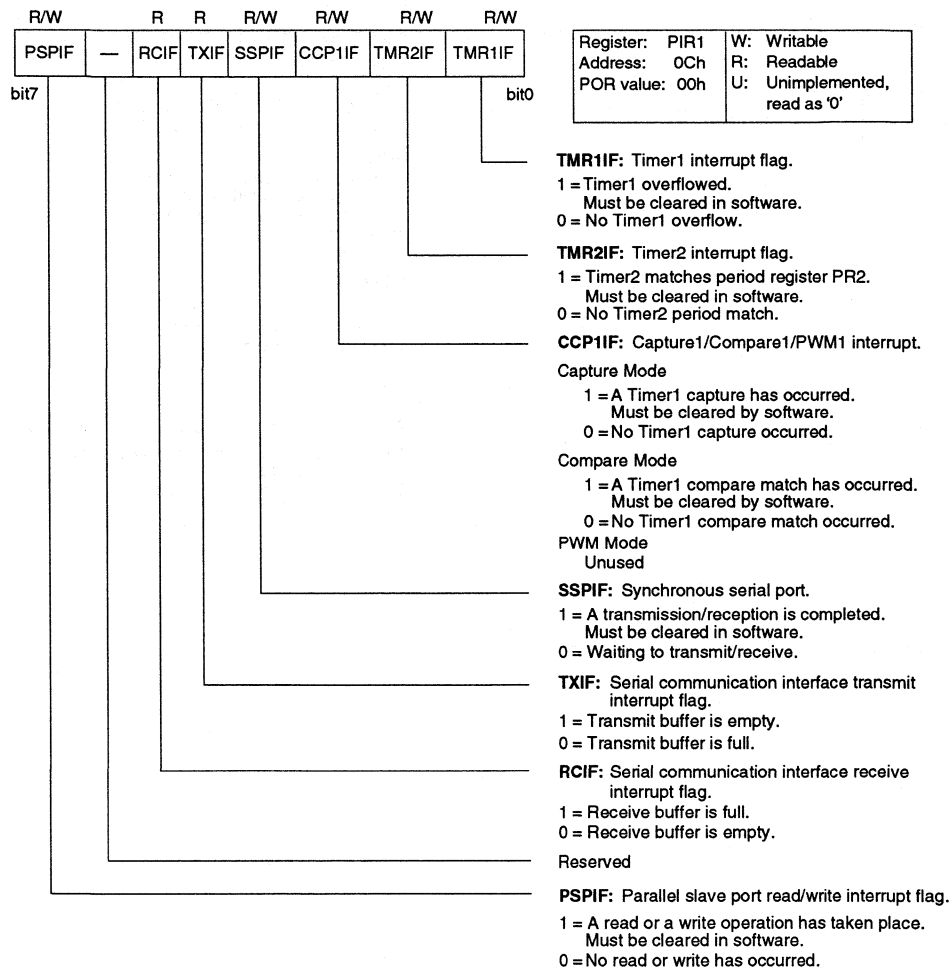
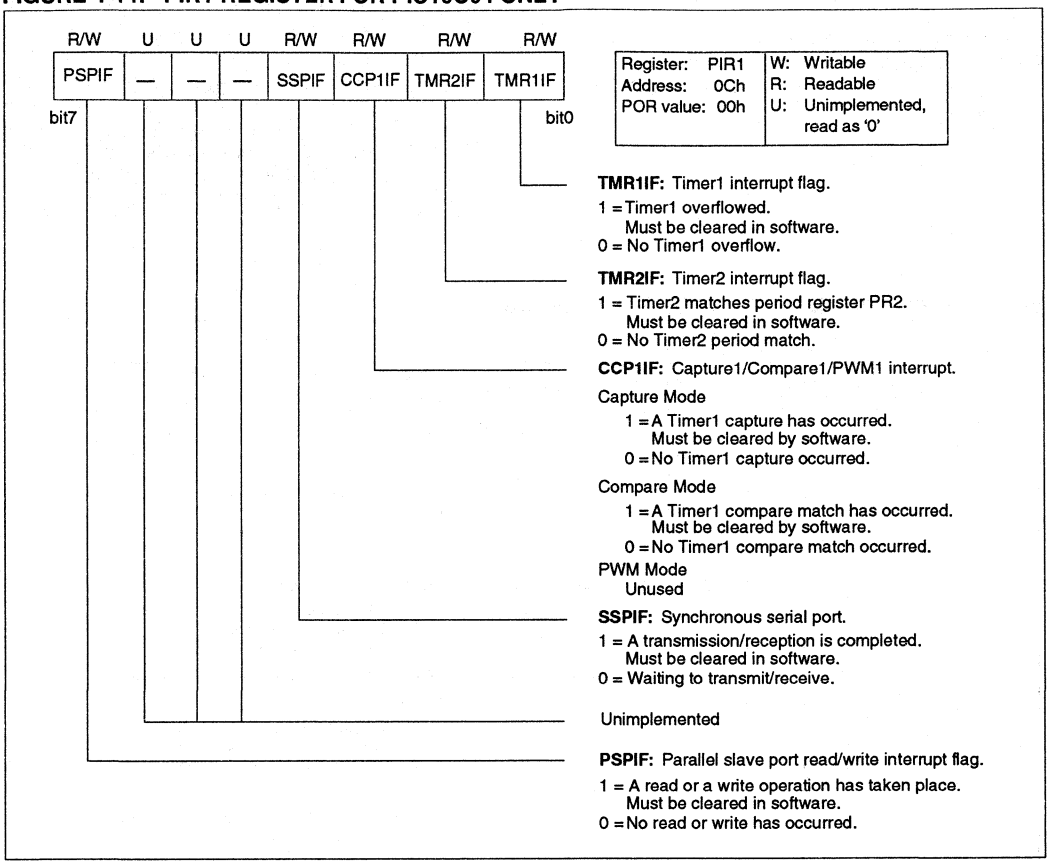


FIGURE 4-14: PIR1 REGISTER FOR PIC16C64 ONLY

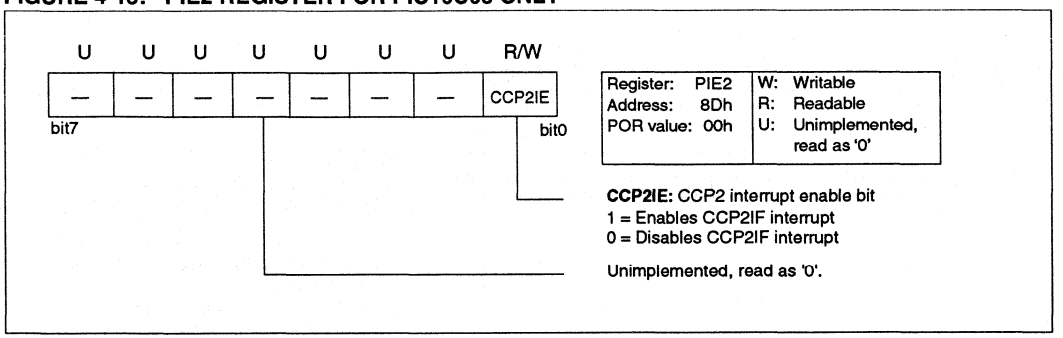


© 1995 Microchip Technology Inc.

4.2.2.6 PIE2 REGISTER

This register, implemented on the PIC16C65 only, contains the individual enable bit for the Peripheral Interrupts (see Figure 4-15).

FIGURE 4-15: PIE2 REGISTER FOR PIC16C65 ONLY



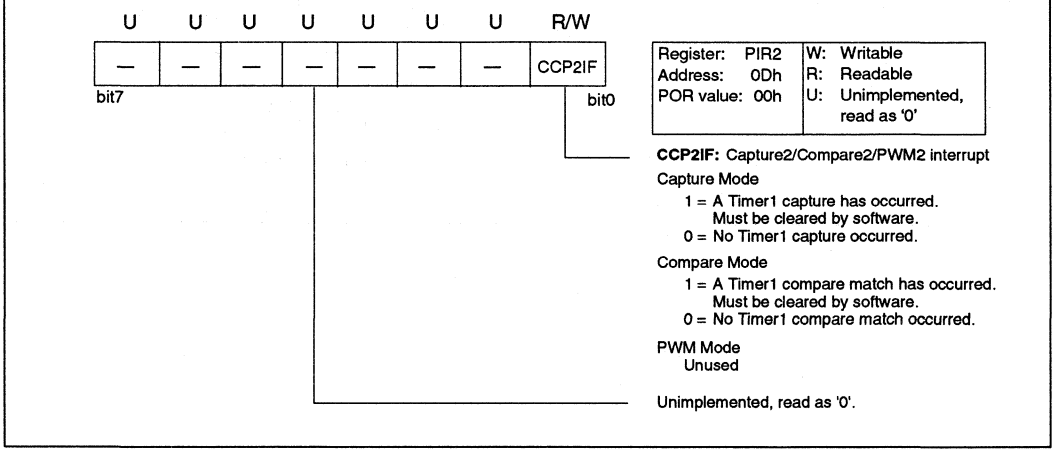
© 1995 Microchip Technology Inc.

4.2.2.7 PIR2 REGISTER

This register, implemented on the PIC16C65 only, contains the individual flag bit for the Peripheral interrupts (see Figure 4-16).

Note: This bit will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

FIGURE 4-16: PIR2 REGISTER FOR PIC16C65 ONLY

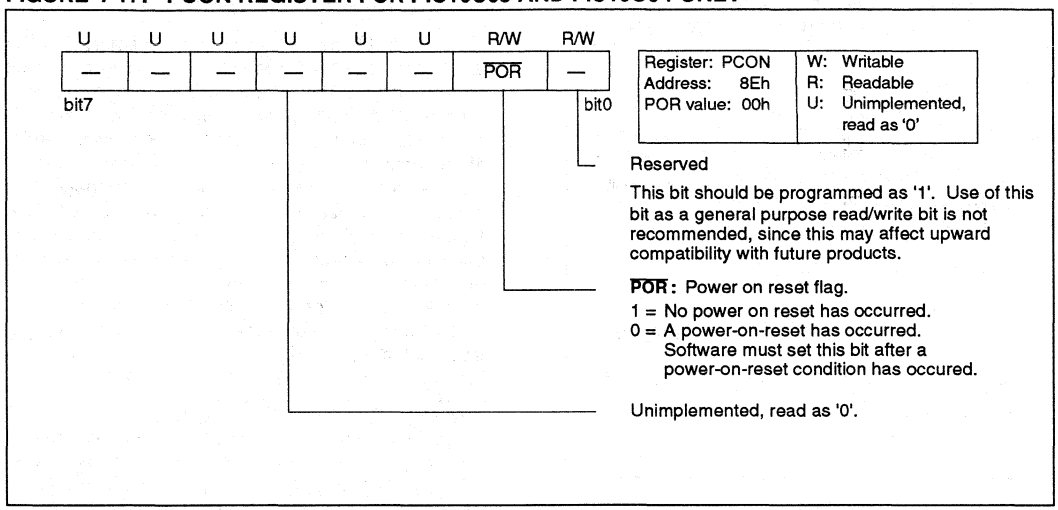


© 1995 Microchip Technology Inc.

4.2.2.8 PCON REGISTER

The Power Control (PCON) register, implemented on the PIC16C65 and PIC16C64 only, contains a flag bit to allow differentiation between a Power-on Reset to an external $\overline{MCLR}$ reset or WDT reset (see Figure 4-17).

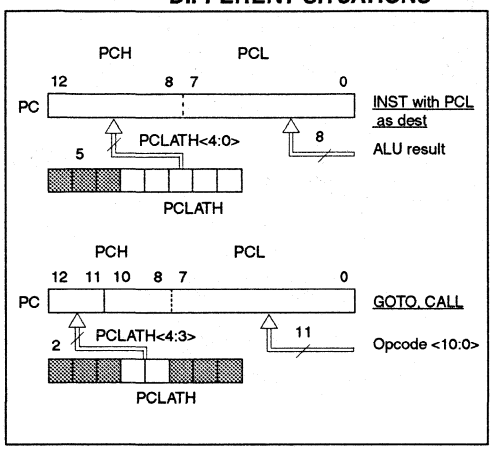
FIGURE 4-17: PCON REGISTER FOR PIC16C65 AND PIC16C64 ONLY



4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte, PCL, is a readable and writable register. The high byte of the PC (PCH) is not directly readable or writable. PCLATH is a holding register for PC<15:8> where contents are transferred to the upper byte of the program counter. When the PC is loaded with a new value during a CALL, GOTO or a write to PCL, the high bits of PC are loaded from PCLATH as shown in Figure 4-18.

FIGURE 4-18: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note, *"Table Read Using the* PIC16CXX*"* (AN556).

4.3.2 STACK

The PIC16CXX has an 8 deep x 13-bit wide hardware stack (see Figure 4-1, Figure 4-2 and Figure 4-3). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed in the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is POPped in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer. This means that after the stack has been "PUSHed" eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflows or stack underflow conditions.

Note 2: There are no instructions mnemonics called PUSH nor POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address

4.3.3 PROGRAM MEMORY PAGING

The PIC16C65 has 4K of program memory, but the CALL and GOTO instructions only have an 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size. To allow CALL and GOTO instructions to address the entire 4K program memory address range, there must be another bit to specify the program memory page. This paging bit comes from the PCLATH<3> bit (see Figure 4-18). When doing a CALL or GOTO instruction, the user must ensure that this page bit (PCLATH<3>) is programmed to the desired program memory page. If a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> is not required for the return instructions (which pops the PC from the stack).

Note: The PIC16C6X ignores the PCLATH<4> bit, which is used for program memory pages 2 and 3 (1000h - 1FFFh). The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.

The PIC16C64 and PIC16C61 ignore the PCLATH<3> bit, which is used for program memory page 1 (0800h-0FFFh). The use of PCLATH<3> as a general purpose read/write bit for the PIC16C64 and PIC16C61 is not recommended since this may affect upward compatibility with future products.

Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that the PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

© 1995 Microchip Technology Inc.

EXAMPLE 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
ORG 0X500
BSF     PCLATH, 3 ; Select page 1 (800h-FFFh)
CALL    SUB1_P1   ; Call subroutine in
        :         ; page 1 (800h-FFFh)
        :
        :
ORG     0X900
SUB1 P1 :         ; called subroutine
        :         ; page 1 (800h-FFFh)
        :
RETURN            ; return to page 0
                  ; (000h-7FFh)
```

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

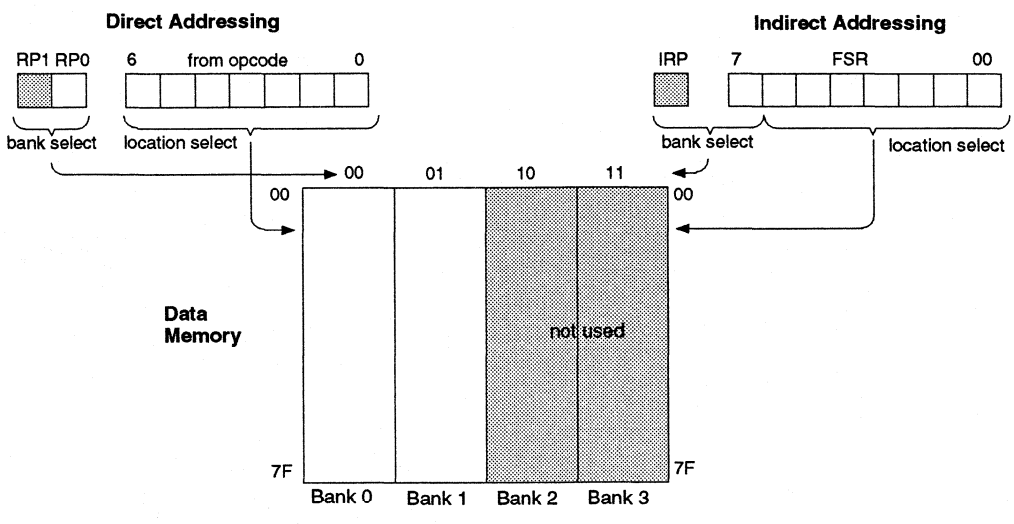
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-19. However, IRP is not used in the PIC16C6X.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

```
        movlw   0x20    ;initialize pointer
        movf    FSR     ;to RAM
NEXT    clrf    INDF    ;clear INDF register
        inc     FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;no clear next
                        ;yes continue
CONTINUE:
```

FIGURE 4-19: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail see Figure 4-4, Figure 4-5 and Figure 4-6.

NOTES:

© 1995 Microchip Technology Inc.

5.0 I/O PORTS

The PIC16C65 and PIC16C64 have five ports, PORTA through PORTE and the PIC16C61 has two ports, PORTA and PORTB. These port pins may be multiplexed with an alternate function for the peripheral features on the device.

5.1 PORTA and TRISA Registers

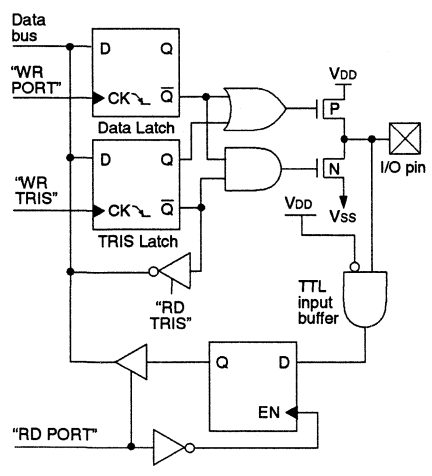
PORTA is a 6-bit wide latch for the PIC16C65 and PIC16C64 and is a 5-bit wide latch for the PIC16C61. RA4 is a Schmitt trigger input and an open collector output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

A '1' in the TRISA register puts the corresponding output driver in a high impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Port RA4 is multiplexed with TMR0 clock input.

FIGURE 5-1: BLOCK DIAGRAM OF RA<3:0> (AND RA<5> ON THE PIC16C65/PIC16C64) PINS

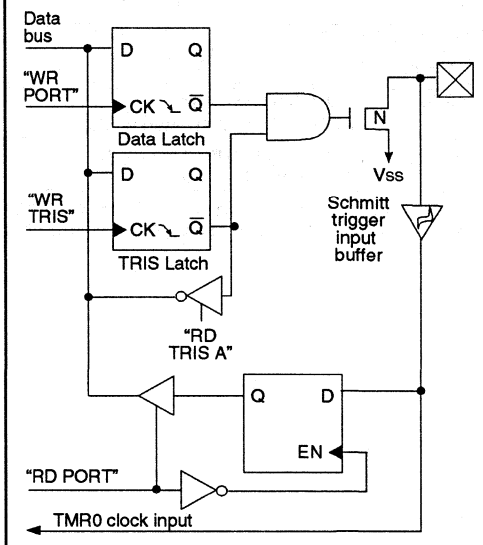


Note: I/O pins have protection diodes to VDD and VSS

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF   PORTA        ;Initialize PORTA by setting
                    ; output data latches
BSF    STATUS, RP0  ;Select Bank1
MOVLW  0xCF         ;Value used to initialize
                    ;data direction
MOVWF  TRISA        ;Set RA<3:0> as inputs
                    ;RA<5:4> as outputs
                    ;TRISA<7:6> are always
                    ;read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN



Note: I/O pin has protection diodes to VSS only.

TABLE 5-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open collector type.
RA5/$\overline{SS}$ [1]	bit5	TTL	Input/output, slave select input for synchronous serial port.

Legend: TTL = TTL input, ST = Schmitt trigger input

TABLE 5-2: SUMMARY OF PORTA REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTA	PORTA pins when read PORTA data latch when written	05h	--xx xxxx[1]
TRISA	PORTA data direction register 0 = output, 1 = input	85h	--11 1111[1]

Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 13-8.
Note 1: The PIC16C61 does not have PORTA bit 5 or TRISA bit 5, read as '0'.

© 1995 Microchip Technology Inc.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bidirectional port (file register address 06h). The corresponding data direction register is TRISB (address 86h). A '1' in the TRISB register puts the corresponding output driver in a high impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

EXAMPLE 5-2: INITIALIZING PORTB

```
CLRF    PORTB       ; Initialize PORTB data
                    ;   latches before setting
                    ;   the data direction
                    ;   register
BSF     STATUS, RP0 ; Select Bank1
MOVLW   0xCF        ; Value used to initialize
                    ; data direction
MOVWF   TRISB       ; Set RB<3:0> as inputs
                    ;   RB<5:4> as outputs
                    ;   RBZZ<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on power-on reset.

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7–RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7–RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7–RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

This interrupt can wake the device up from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

a) Disable the interrupt by clearing RBIE (INTCON<3>) bit.

b) Read PORTB. This will end mismatch condition. Then, clear the RBIF bit.

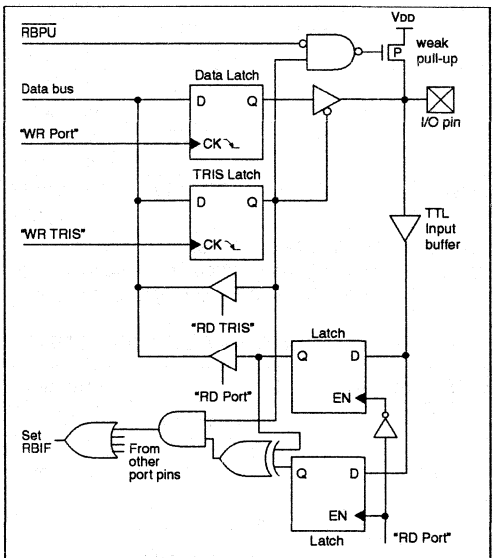
A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the *Embedded Control Handbook*).

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

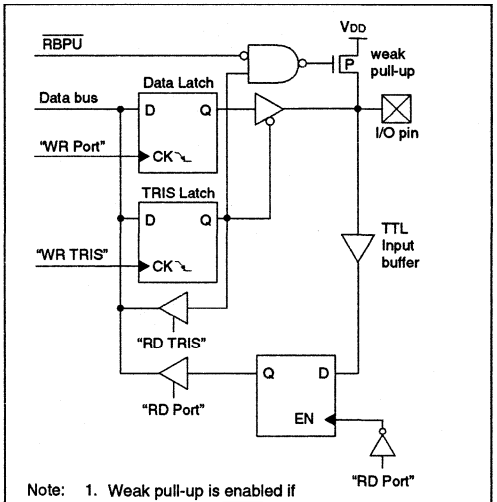
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-3: BLOCK DIAGRAM OF RB<7:4> PINS



Note: 1. Weak pull-up is enabled if TRISB = 1 and $\overline{\text{RBPU}}$, Option <7>, = 0
2. I/O pins have diode protection to VDD and VSS

FIGURE 5-4: BLOCK DIAGRAM OF RB<3:0> PINS



Note: 1. Weak pull-up is enabled if TRISB = 1 and $\overline{\text{RBPU}}$, Option <7>, = 0
2. I/O pins have diode protection to VDD and VSS.
3. For RB0/INT pin, the INT input comes through a Schmitt trigger input buffer.

TABLE 5-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/INT	bit0	TTL/ST†	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger

† This buffer is a Schmitt trigger input when configured as the external interrupt.

‡ This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4: SUMMARY OF PORTB REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTB	PORTB pins when read PORTB data latch when written	06h	xxxx xxxx
TRISB	PORTB data direction register 0 = output, 1 = input	86h	1111 1111
OPTION	Weak pull-up on/off control ($\overline{RBPU}$ bit)	81h	1111 1111

Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 13-8.

© 1995 Microchip Technology Inc.

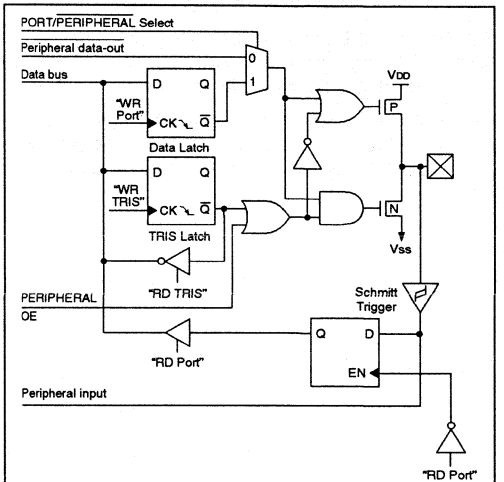
5.3 PORTC and TRISC Registers

PORTC is an 8-bit bidirectional port available in the PIC16C65 and PIC16C64 only. Each pin is individually configurable as input and output through the TRISC register. PORTC is multiplexed with several peripheral functions (see Table 5-5). PORTC pins have Schmitt trigger input buffers.

EXAMPLE 5-3: INITIALIZING PORTC

```
CLRF   PORTC        ; Initialize PORTC data
                    ;    latches before setting
                    ;    the data direction
                    ;    register
BSF    STATUS, RPO  ; Select Bank1
MOVLW  0xCF         ; Value used to initialize
                    ; data direction
MOVWF  TRISC        ; Set RC<3:0> as inputs
                    ;    RC<5:4> as outputs
                    ;    RC<7:6> as inputs
```

FIGURE 5-5: PORTC BLOCK DIAGRAM



Note 1: Port/Peripheral select signal selects between port data and peripheral output.
2: Peripheral OE (output enable) is only activated if peripheral select is active.
3: I/O pins have diode protection to VDD and VSS.

TABLE 5-5: PORTC FUNCTIONS

Name	Bit	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input on the PIC16C65
RC0/T1OSI/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator input/Timer1 clock input on the PIC16C64
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin, Timer1 oscillator input, Capture 2 input/Compare 2 output/PWM 2 output on the PIC16C65
RC1/T1OSO	bit1	ST	Input/output port pin or Timer1 oscillator output on the PIC16C64
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL	bit3	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous serial port data output
RC6/TX/CK[1]	bit6	ST	Input/output port pin, SCI Asynchronous Transmit, or SCI Synchronous Clock
RC7/RX/DT[1]	bit7	ST	Input/output port pin SCI Asynchronous Receive, or SCI Synchronous Data

Legend: ST = Schmitt Trigger Input
Note 1: TX/CK and RX/DT are not implemented on the PIC16C64.

TABLE 5-6: SUMMARY OF PORTC REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTC	PORTC pins when read PORTC data latch when written	07h	xxxx xxxx
TRISC	PORTC data direction register 0 = output, 1 = input	87h	1111 1111

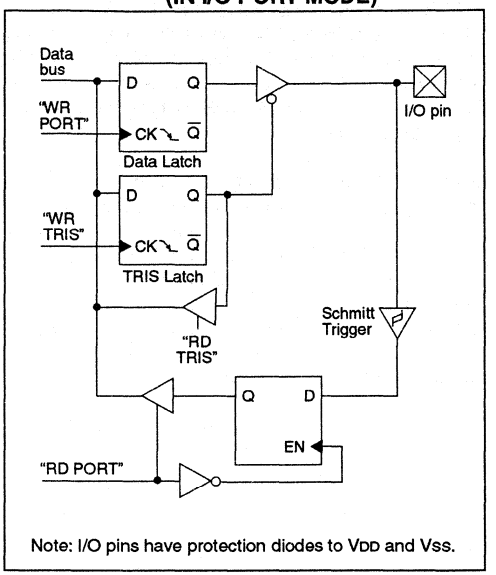
Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 13-8.

5.4 PORTD and TRISD Registers

PORTD is an 8-bit port with Schmitt trigger input buffers available in the PIC16C65 and PIC16C64 only. Each pin is individually configurable as input or output.

PORTD can be configured as an 8-bit wide microprocessor port (or parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

FIGURE 5-6: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and Vss.

TABLE 5-7: PORTD FUNCTIONS

Name	Bit	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL	Input/output port pin or parallel slave port bit 0
RD1/PSP1	bit1	ST/TTL	Input/output port pin or parallel slave port bit 1
RD2/PSP2	bit2	ST/TTL	Input/output port pin or parallel slave port bit 2
RD3/PSP3	bit3	ST/TTL	Input/output port pin or parallel slave port bit 3
RD4/PSP4	bit4	ST/TTL	Input/output port pin or parallel slave port bit 4
RD5/PSP5	bit5	ST/TTL	Input/output port pin or parallel slave port bit 5
RD6/PSP6	bit6	ST/TTL	Input/output port pin or parallel slave port bit 6
RD7/PSP7	bit7	ST/TTL	Input/output port pin or parallel slave port bit 7

Legend: ST = Schmitt Trigger Input when configured for general purpose I/O, TTL = TTL input when configured for Parallel Slave Port (PSP).

TABLE 5-8: SUMMARY OF PORTD REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTD	PORTD pins when read PORTD data latch when written	08h	xxxx xxxx
TRISD	PORTD data direction register 0 = output, 1 = input	88h	1111 1111

Legend: x = unknown, -= unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 13-8.

© 1995 Microchip Technology Inc.

5.5 PORTE and TRISE Register

PORTE is available on the PIC16C65 and PIC16C64 only and has three pins RE0, RE1 and RE2, which are individually configurable as inputs or outputs. These have Schmitt trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when the PSPMODE bit (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set. In this mode the input buffers are TTL.

Figure 5-8 shows the TRISE register, which also controls the parallel slave port operation.

TRISE controls the direction of the RE pins.

FIGURE 5-7: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)

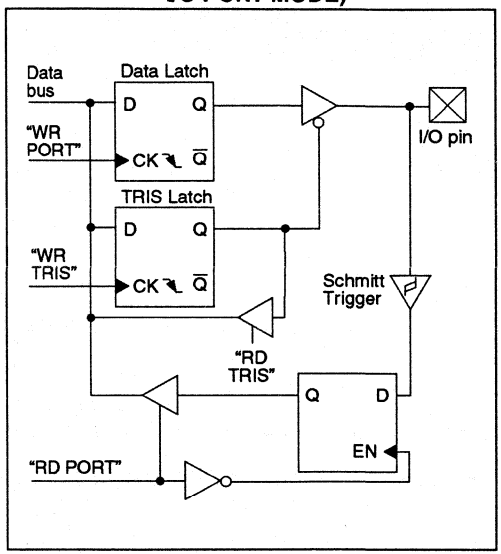


FIGURE 5-8: TRISE REGISTER

R	R	R/W	R/W	U	R/W	R/W	R/W
IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
bit7							bit0

Register: TRISE	W: Writable
Address: 89h	R: Readable
POR value: 07h	U: Unimplemented, read as '0'

TRISE0: Direction control bit for port pin RE0
1 = Input
0 = Output

TRISE1: Direction control bit for port pin RE1
1 = Input
0 = Output

TRISE2: Direction control bit for port pin RE2
1 = Input
0 = Output

Unimplemented.
Read as '0'

PSPMODE: Selects parallel slave port mode for ports RD and RE.
1 = Parallel slave port mode.
0 = General purpose I/O.

IBOV: Input buffer overflow in microprocessor mode.
1 = A write occurred when a previous input word has not been read.
Must be cleared in software.
0 = No overflow has occurred.

OBF: Output buffer full.
1 = The output buffer still holds a previously written word.
0 = Output buffer has been read.

IBF: Input buffer full.
1 = A word has been received and waiting to be read by the CPU.
0 = No word has been received.

TABLE 5-9: PORTE FUNCTIONS

Name	Bit	Buffer Type	Function
RE0/$\overline{RD}$	bit0	ST/TTL	Input/output port pin, Read control input in parallel slaveport mode $\overline{RD}$ 1 = Not a read operation 0 = Read operation. The system reads the PIC16C65/64 PORTD register (if chip selected)
RE1/$\overline{WR}$	bit1	ST/TTL	Input/output port pin, Write control input in parallel slaveport mode $\overline{WR}$ 1 = Not a write operation 0 = Write operation. The system writes to the PIC16C65/64 PORTD register (if chip selected)
RE2/$\overline{CS}$	bit2	ST/TTL	Input/output port pin, Chip select control input in parallel slave port mode $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger Input when configured for general purpose I/O, TTL = TTL input for Parallel Slave Port (PSP).

TABLE 5-10: SUMMARY OF PORTE REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTE	PORTE pins when read PORTE data latch when written	09h	---- -xxx
TRISE	PORTE data direction control bits and PORTD mode control	89h	0000 -111

Legend: x = unknown, -= unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 13-8.

© 1995 Microchip Technology Inc.

5.6 I/O Programming Considerations

5.6.1 BIDIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit 5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit 5 and PORTB is written to the output latches. If another bit of PORTB is used as a bidirectional I/O pin (e.g., bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit 0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading the PORT register, reads the values of the PORT pins. Writing to the PORT register writes the value to the PORT latch. When using read modify write instructions (ex. BCF, BSF, etc.) on a PORT, the value of the PORT pins is read, the desired operation is done to this value, and this value is then written to the PORT latch.

Example 5-4 shows the effect of two sequential read modify write instructions (ex. BCF, BSF, etc.) on an I/O PORT.

EXAMPLE 5-4: READ MODIFY WRITE INSTRUCTIONS ON AN I/O PORT

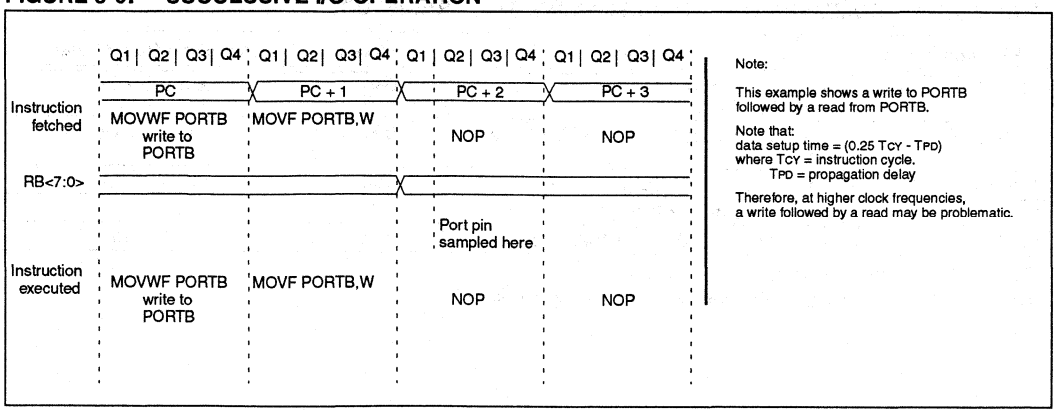
```
;Initial PORT settings:  PORTB<7:4> Inputs
;                        PORTB<3:0> Outputs
;PORTB<7:6> have external pull-up and are not
;connected to other circuitry
;                        PORT latch  PORT pins
;                        ----------  ----------

   BCF PORTB, 7         ;01pp pppp   11pp pppp
   BCF PORTB, 6         ;10pp pppp   11pp pppp
   BSF STATUS,RP0       ;
   BCF TRISB, 7         ;10pp pppp   11pp pppp
   BCF TRISB, 6         ;10pp pppp   10pp pppp
;
;Note that the user may have expected the pin
;values to be 00pp pppp. The 2nd BCF caused
;RB7 to be latched as the pin value (High).
```

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (see Figure 5-9). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-9: SUCCESSIVE I/O OPERATION



5.7 Parallel Slave Port

PORTD operates as an 8-bit wide parallel slave port, or microprocessor port when control bit PSPMODE (TRISE<4>) is set. In slave mode it is asynchronously readable and writable by the external world through $\overline{RD}$ control input (RE0/$\overline{RD}$) and $\overline{WR}$ control input (RE1/$\overline{WR}$).

It can directly interface to an 8-bit microprocessor data bus. The microprocessor can read or write the PORTD latch as an 8-bit latch. Setting PSPMODE enables the port pin RE0 to be the $\overline{RD}$ input, RE1 to be the $\overline{WR}$ input and RE2 to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set).

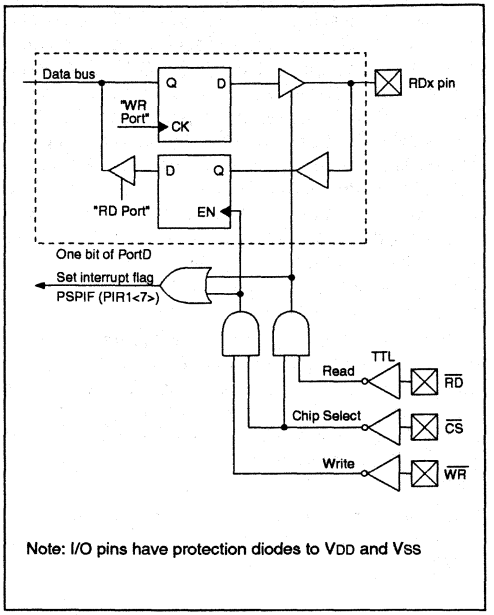
There are actually two 8-bit latches, one for data-out (from the PIC16/17) and one for data input. The user writes 8-bit data to PORTD data latch and reads data from the port pin latch (note that they have the same address). In this mode the TRISD register is ignored, since the microprocessor is controlling the direction of data flow.

Status flag IBF, Input Buffer Full (TRISE<7>), is set if a received word is waiting to be read by the CPU. Once the PORTD input latch is read by the PIC16C65/64, IBF is cleared. IBF is a read only status bit. Status flag OBF, Output Buffer Full (TRISE<6>), is set if a word written to PORTD latch is waiting to be read by the external bus. Once the PORTD output latch is read by the microprocessor, OBF is cleared. Status flag IBOV, Input Buffer Overflow (TRISE<5>), is set if a second word is written to the microprocessor port when the previous word has not been read by the CPU. It is a read/write bit and must be cleared by the CPU.

When not in PSPMODE, IBF and OBF bits are held clear. However, if the IBOV flag was previously set, it must be cleared in the software.

An interrupt is generated and latched into control bit PSPIF (PIR1<7>) when a read or a write operation is completed. The PSPIF interrupt flag must be cleared by the CPU and the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

FIGURE 5-10: PORTD AND PORTE AS A PARALLEL SLAVE PORT



Note: I/O pins have protection diodes to VDD and VSS

TABLE 5-11: SUMMARY OF PARALLEL SLAVE PORT REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTD	Parallel slave port Read/Write Data	08h	xxxx xxxx
PORTE	Parallel slave port Read/Write/Chip Select signals	09h	---- -xxx
TRISE	Control bits for PORTE peripheral	89h	0000 -111
PIR1	Interrupt register (PSPIF bit)	0Ch	0000 0000
PIE1	Interrupt Enable register (PSPIE bit)	8Ch	0000 0000

Legend: x = unknown, - = unimplemented, read as a '0'.
Note: For reset values of registers in other reset situations refer to Table 13-8.

© 1995 Microchip Technology Inc.

TABLE 5-12: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
08h	PORTD	PORTD							
09h	PORTE	PORTE							
89h	TRISE	IBF	OBF	IBOV	PSPMODE	–	TRISE 2	TRISE 1	TRISE 0
0Ch	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
8Ch	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE

Legend: – = unimplemented locations, read as a '0'.
Note: Shaded boxes are not used by Parallel Slave Port module.
Note 1: These bits are not implemented on the PIC16C64.
†These bits are reserved on the PIC16C65.

2

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

6.0 OVERVIEW OF TIMER MODULES

The PIC16C65 and PIC16C64 have three timer modules. The PIC16C61 has one timer module. Each module can generate an interrupt to indicate that an event has occurred (i.e. timer overflow). Each of these modules is explained in full detail in the following sections. The timer modules are:

- Timer0 (TMR0) module (see Section 7.0)
- Timer1 (TMR1) module (see Section 8.0)
- Timer2 (TMR2) module (see Section 9.0)

For enhanced time-based functionality, two additional modules can be used with either of the TMR1 or TMR2 modules. There are:

- Capture/Compare/PWM1 (CCP1) module (see Section 10.0)
- Capture/Compare/PWM2 (CCP2) module (see Section 10.0)

6.1 Timer0 (TMR0) Overview

The TMR0 module (previously known as RTCC) is a simple 8-bit overflow counter. The clock source can be either the internal system clock (OSC/4) or an external clock. When the clock source is an external clock, the TMR0 module can be selected to increment on either the rising or falling edge.

The TMR0 module also has a programmable prescaler option. This prescaler can be assigned to either the TMR0 module or the Watchdog timer. The PSA bit (OPTION<3>) assigns the prescaler, and the PS2 -PS0 (OPTION<2:0>) determines the prescaler value. The TMR0 can increment at the following rates: 1:1 (when prescaler assigned to Watchdog timer), 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

6.2 Timer1 (TMR1) Overview

Timer1 (TMR1) is a 16-bit timer/counter. The clock source can be either the internal system clock (OSC/4), an external clock, or an external crystal. TMR1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows TMR1 to operate during sleep, which is useful for applications that require a real time clock as well as the power savings of sleep mode.

TMR1 also has a prescaler option which allows the TMR1 to increment at the following rates: 1:1, 1:2, 1:4, 1:8. TMR1 can be used in conjunction with the Capture / Compare / PWM (CCP1 or CCP2) module. When used with the CCP 1or CCP2 module, TMR1 is the timebase for 16-bit capture or the 16-bit compare. When using the TMR1 module with the CCP1 or CCP2 module, TMR1 must be synchronized to the device.

6.3 Timer2 (TMR2) Overview

Timer2 (TMR2) is an 8-bit timer. TMR2 has both a programmable prescaler and postscaler, as well as an 8-bit period register (PR2). TMR2 can be used with the CCP1 module as well as the baud rate generator for the Synchronous Serial Port (SSP). The prescaler option which allows the TMR2 to increment at the following rates: 1:1, 1:4, 1:16.

The postscaler allows TMR2 to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

6.4 CCP1 and CCP2 Overview

The CCP modules can operate in one of these three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM).

Capture mode, captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or the sixteenth rising edge of the CCPx pin.

Compare mode, compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs an interrupt can be generated, and the output pin CCP1 can be forced to given state (High or Low) and TMR1 can be reset (CCP1). This depends on the control bits CCPxM3 - CCPxM0.

PWM mode, compares TMR2 to a 10-bit duty cycle register as well as to an 8-bit period register (PR2). When the TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high. When the TMR2 = Duty Cycle register, the CCPx pin will be forced low.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

7.0 TIMER0 (TMR0) MODULE

The TMR0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 7-1 is a simplified block diagram of the TMR0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two cycles (see Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 module.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the T0 source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

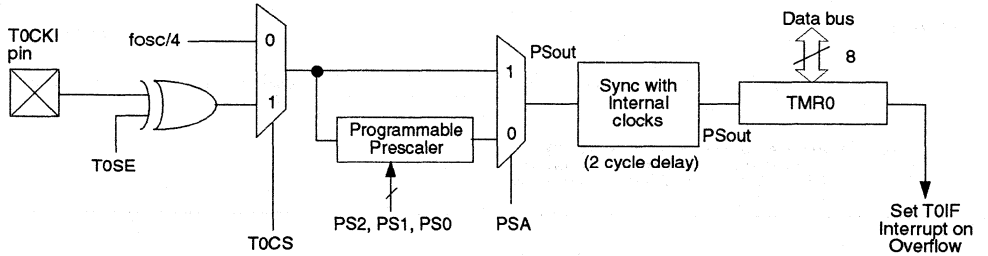
The prescaler is shared between the TMR0 module and the watchdog timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is not readable or writable. When the prescaler is assigned to the TMR0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

7.1 TIMER0 (TMR0) Interrupt

TMR0 interrupt is generated when the TMR0 module timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the TMR0 module interrupt service routine before re-enabling this interrupt. The TMR0 module interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. See Figure 7-4 for TMR0 interrupt timing.

2

FIGURE 7-1: TIMER0 (TMR0) BLOCK DIAGRAM



Note 1: Bits, T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.
Note 2: The prescaler is shared with Watchdog Timer (refer to Figure 7-6 for detailed diagram)

FIGURE 7-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALE

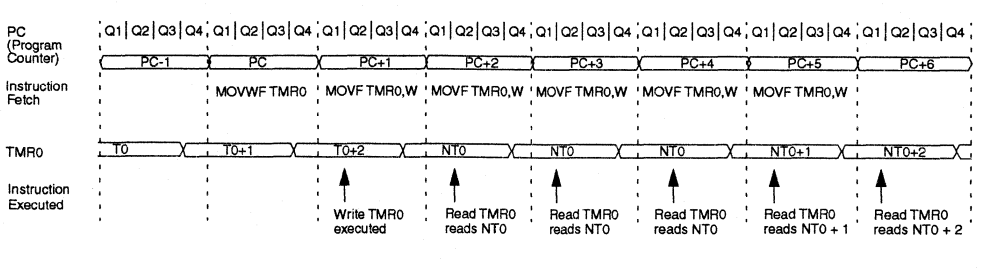


FIGURE 7-3: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/PRESCALE 1:2

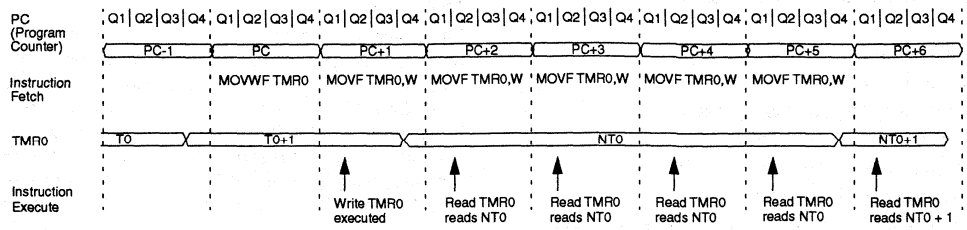
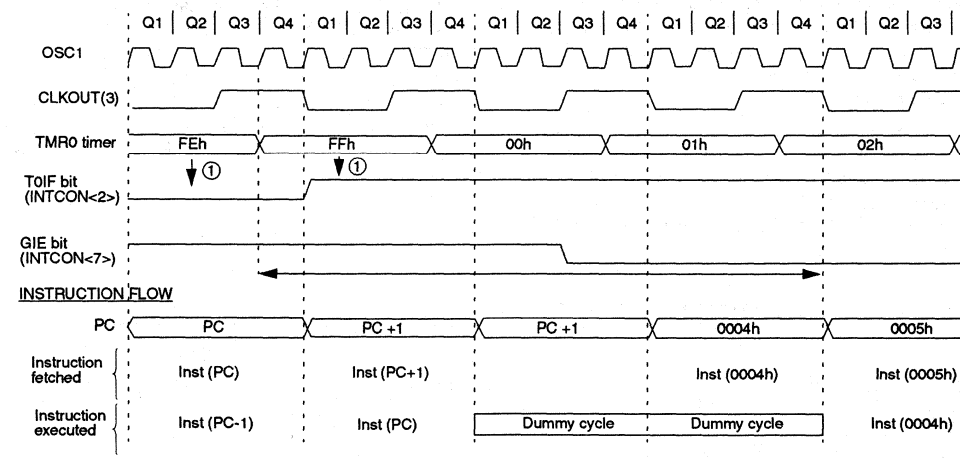


FIGURE 7-4: TIMER0 (TMR0) INTERRUPT TIMING



Note 1: T0IF interrupt flag is sampled here (every Q1).
2: Interrupt latency = 4Tcy where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

© 1995 Microchip Technology Inc.

7.2 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (see Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20ns) and low for at least 2Tosc (and a small RC delay of 20ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 Tosc (and a small RC delay of 40ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

7.3 Prescaler

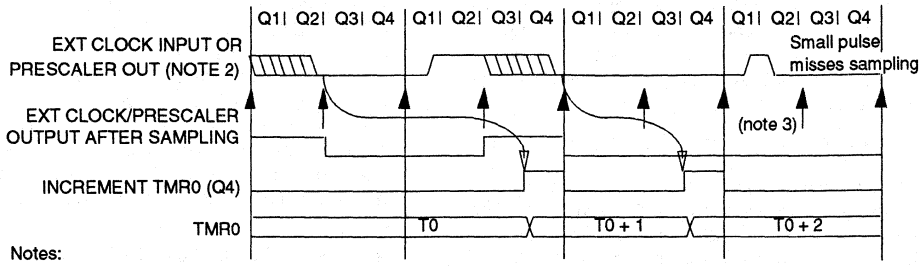
An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer, respectively (see Figure 7-6). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2-PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g. `CLRF 1, MOVWF 1, BSF 1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

© 1995 Microchip Technology Inc.

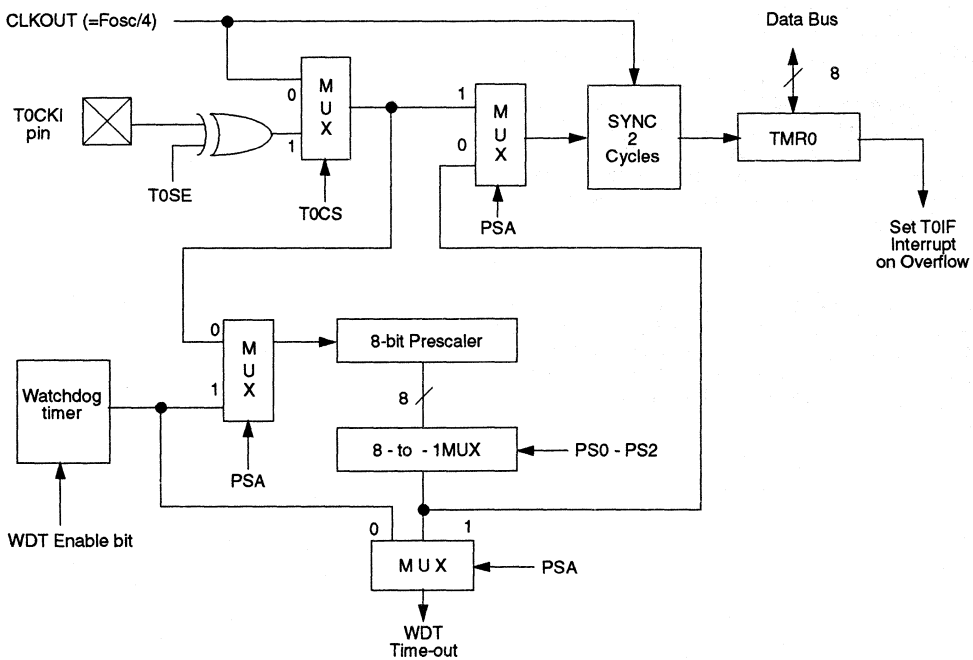
FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to TMR0 increment is 3 tosc to 7 tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 tosc max.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

FIGURE 7-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

© 1995 Microchip Technology Inc.

7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from TMR0 to WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TMR0→WDT)

```
BCF     STATUS, RP0   ;Bank 0
CLRF    TMR0          ;Clear TMR0 & Prescaler
BSF     STATUS, RP0   ;Bank 1
CLRWDT                ;Clears WDT
MOVLW   B'xxxx1xxx'   ;Select new prescaler
MOVWF   OPTION        ;value
BCF     STATUS, RP0   ;Bank 0
```

To change prescaler from the WDT to the TMR0 module use the sequence shown in Example 7-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
BSF     STATUS, RP0
MOVLW   B'xxxx0xxx'   ;Select TMR0, new
                      ;prescale value and
                      ;clock source
MOVWF   OPTION
BCF     STATUS, RP0
```

2

TABLE 7-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. See Figure 7-5.	81h	1111 1111
INTCON	TMR0 overflow interrupt flag and mask bits. See Figure 7-6	0Bh	0000 000x

Legend: x = unknown, - = unimplemented, reads as a '0'.
Note: For reset values of registers in other reset situations refer to Table 13-8.

TABLE 7-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	TMR0	TMR0							
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
85h	TRISA	—	—	TRISA 5[1]	TRISA 4	TRISA 3	TRISA 2	TRISA 1	TRISA 0

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by TMR0 module
Note 1: This bit is not available in the PIC16C61

NOTES:

© 1995 Microchip Technology Inc.

8.0 TIMER1 (TMR1) MODULE

TMR1 is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L) which are readable and writable. TMR1 increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled or disabled using the TMR1 interrupt enable bit TMR1IE (PIE1<0>).

TMR1 can operate in one of two modes:

- As a timer
- As a counter

This is determined by the clock select bit, TMR1CS (T1CON<1>).

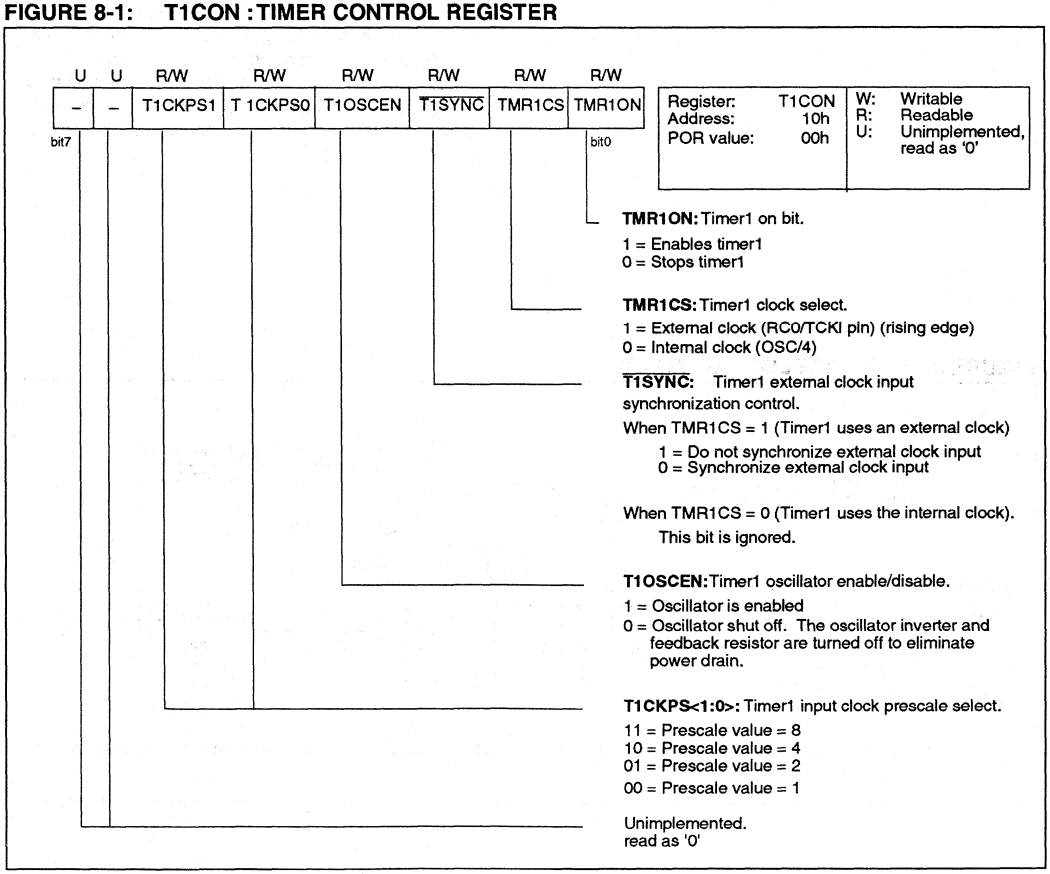
In timer mode, TMR1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input on RC0/T1OSO/T1CKI.

TMR1 can be turned on or off using the control bit TMR1ON (T1CON<0>).

TMR1 also has an internal "reset input". This reset can be generated by CCP1 or CCP2 (Capture/compare/PWM) module. See Section 10.0 for details. Figure 8-1 shows the Timer1 control register.

When the TMR1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 pin on the PIC16C65 or the RC0/T1OSI/T1CKI pin on the PIC16C64, becomes an input. That is, the TRISC<1> value is ignored. The RC0/T1OSO/T1CKI pin on the PIC16C65 or the RC1/T1OSO pin on the PIC16C64 should normally be configured as an input (for external clock). However, this pin can be configured as an output if self-clocking (through the output pin) is desired.

FIGURE 8-1: T1CON : TIMER CONTROL REGISTER



8.1 TMR1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is OSC/4. The synchronize control bit T1INSYNC (T1CON<2>) has no effect since the internal clock is always in sync.

8.2 TMR1 Operation in Synchronized Counter Mode

Counter mode is selected by setting the TMR1CS bit. In this mode the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2 (when T1OSCEN is set) or the RC0/T1OSO/T1CKI.

If T1SYNC is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during SLEEP mode, TMR1 will not increment even if external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

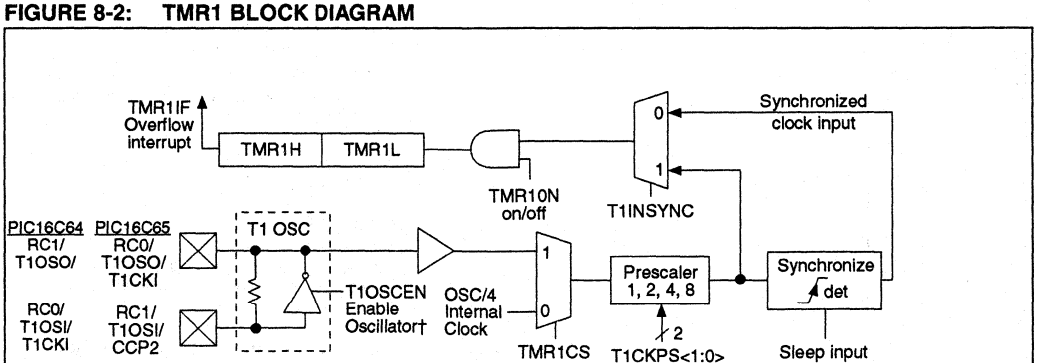
8.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for TMR1 in synchronized counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T1CKI to be high for at least 2Tosc (and a small RC delay of 20ns) and low for at least 2Tosc (and a small RC delay of 20ns). Refer to Figure 16-5, Figure 18-5 and Figure 20-5, parameters 45 and 46.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least 4Tosc (and a small RC delay of 40ns) divided by the prescaler value. The only requirement on T1CKI high and low time is that they do not violate the minimum pulse width requirements of 10ns). Refer to Figure 16-5 and Figure 18-5, parameters 45, 46, and 47.

FIGURE 8-2: TMR1 BLOCK DIAGRAM



† When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

© 1995 Microchip Technology Inc.

8.3 TMR1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{T1SYNC}}$ is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and generate an interrupt on overflow which will wake up the processor. However, special precautions in software are needed to read/write the timer (see Section 8.3.2).

In asynchronous counter mode, Timer1 can not be used as timebase for capture or compare operations.

8.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet a certain minimum high time and low time requirements, as specified in timing parameter.

8.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will guarantee a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Example 8-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

EXAMPLE 8-1: READING A 16-BIT FREE-RUNNING TIMER

```
;  All Interrupts are disabled
    MOVF    TMR1H, W        ;Read high byte
    MOVWF   TMPH            ;
    MOVF    TMR1L, W        ;Read low byte
    MOVWF   TMPL            ;
    MOVF    TMR1H, W        ;Read high byte
    SUBWF   TMPH,  W        ;Sub 1st read
                            ;with 2nd read
    BTFSC   STATUS,Z        ;is result = 0
    GOTO    CONTINUE        ;Good 16-bit read
;
; TMR1L may have rolled over between the read
; of the high and low bytes. Reading the high
; and low bytes now will read a good value.
;
    MOVF    TMR1H, W        ;Read high byte
    MOVWF   TMPH            ;
    MOVF    TMR1L, W        ;Read low byte
    MOVWF   TMPL            ;
;   Re-enable Interrupt (if required)
CONTINUE                    ;Continue with
    :                       ;your code
```

8.4 Timer1 Oscillator

A crystal oscillator circuit is built in between T1OSI pin (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200KHz. It will continue to run during SLEEP. It is primarily intended for a 32KHz crystal. Table 8-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must allow software time-out to ensure proper oscillator start-up.

TABLE 8-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Osc Type	Freq	C1	C2
LP	32 KHz§	15 pF	15 pF
	100 KHz	15 pF	15 pF
	200 KHz	0 - 15 pF	0 - 15 pF
Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. §For V_{DD} > 4.5V, C1 = C2 ≈ 30pf is recommended.			

8.5 Resetting Timer1 using a CCP Trigger Output

If CCP1 or CCP2 module is configured in compare mode to generate a "special event" trigger (CCP1M<3:0> = 1011), this signal will reset timer1.

Timer1 must be configured for timer or synchronized counter mode operation to take advantage of this feature. If the Timer1 is running in asynchronous counter mode, this reset operation may not work.

In the event that a write to Timer1 coincides with a reset trigger from CCP1 or CCP2, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL registers pair effectively becomes the period register for the timer1.

8.6 Resetting of Timer1 Registers

TMR1H and TMR1L registers are not reset on POR or any other reset except by the CCP1 special reset trigger.

T1CON register is reset to 00h on Power-on Reset. In any other reset, the register is unaffected.

8.7 TMR1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers

TABLE 8-2: REGISTERS ASSOCIATED WITH TMR1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
8C	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR21E	TMR1IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer1 Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1INSYNC	TMR1CS	TMR1ON

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by Timer1 module.
Note 1: These bits are not implemented on the PIC16C64.
† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

9.0 TIMER2 (TMR2) MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It is especially suitable as PWM time-base (for PWM mode of CCP modules). TMR2 is a readable and writable register, and is cleared on any device reset.

The input clock (OSC/4) has a prescale option of 1, 4 or 16 (selected by control bits T2CKPS1, T2CKPS0, of register T2CON).

Timer2 has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is set to all 1s during a reset.

The overflow (or match) output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling) to generate a Timer2 interrupt (latched in TMR2IF bit, PIR<1>).

TMR2 can be shut off using TMR2ON (T2CON<2>) control bit to minimize power consumption.

Figure 9-2 shows the TMR2 control register.

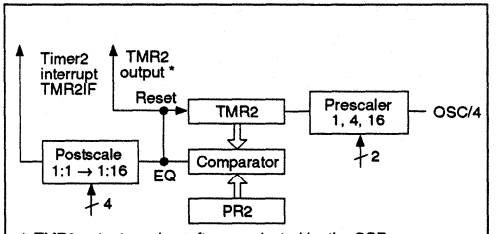
9.1 TMR2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs: a write to the TMR2 register, a write to the T2CON register, or any device reset (Power-on Reset, $\overline{MCLR}$ reset, or Watchdog Timer reset). TMR2 will not clear when T2CON is written, only for a WDT, $\overline{POR}$, and $\overline{MCLR}$ reset.

9.2 Output of TMR2

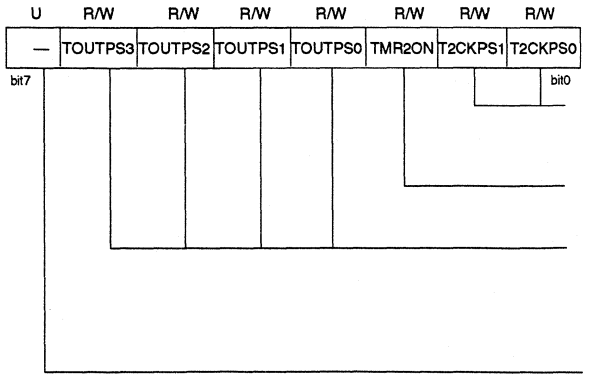
The output of TMR2 (before the postscaler) is fed to the synchronous serial port module which optionally uses it to generate shift clock.

FIGURE 9-1: TIMER2 BLOCK DIAGRAM



* TMR2 output can be software selected by the SSP module as baud clock.

FIGURE 9-2: T2CON: TIMER2 CONTROL REGISTER



Register: T2CON	W:	Writable
Address: 12h	R:	Readable
POR value: 00h	U:	Unimplemented, read as '0'

T2CKPS<1:0>: Timer2 clock prescaler select.
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

TMR2ON: Timer2 on/off control.
1 = TMR2 is on
0 = TMR2 is off

TOUTPS<3:0>: Timer2 output postscale select
0000 = Postscaler is 1
0001 = Postscaler is 2
•
•
1111 = Postscaler is 16

Unimplemented.
Read as '0'

TABLE 9-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
8Ch	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR21E	TMR1IE
11h	TMR2	Timer2							
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKS1	T2CKPS0
92h	PR2	Timer2 period Register.							

Legend: — = Unimplemented locations, read as '0'
Note: Shaded boxes are not used by Timer2 module.
Note 1: These bits are not implemented on the PIC16C64.
† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

10.0 CAPTURE/COMPARE/PWM MODULE

The PIC16C6X has two Capture/Compare/PWM (CCP) modules consisting of a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM output. (Both the CCP1 and CCP2 modules are identical in operation, with the exception of the special trigger.) In the following sections, the operation of a CCP module is described with respect to CCP1. Please note that CCP2 is similar to CCP1, except where noted.

CCP1 module:

Capture/compare/PWM register1 (CCPR1) is made up of two 8-bit sections: low byte, CCPR1L and high byte, CCPR1H. Both are readable and writable.

CCP2 module:

Capture/compare/PWM register2 (CCPR2) is made up of two 8-bit sections: low byte, CCPR2L and high byte, CCPR2H. Both are readable and writable. The interaction of multiple CCP modules is discussed in application note AN594.

10.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of TMR1 when an event occurs on pin RC2/CCP1. An event is defined as:

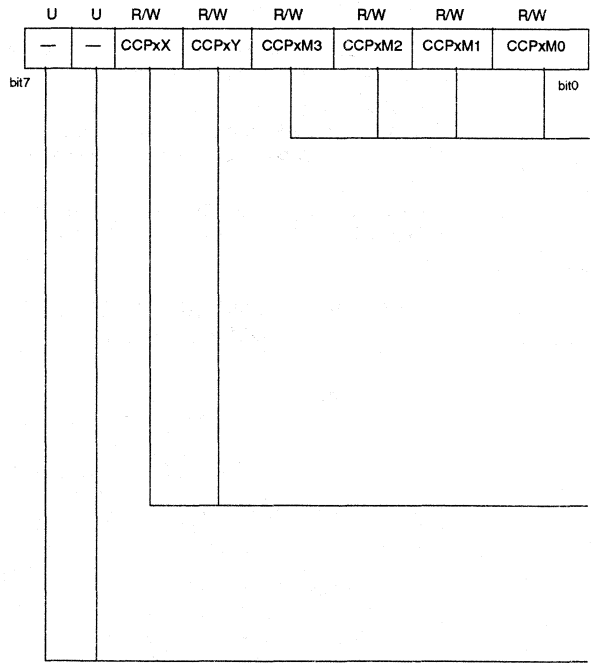
1. A falling edge
2. A rising edge
3. Every 4 rising edges
4. Every 16 rising edges

One of these is selected by the control bits CCP1M <3:0> in register CCP1CON. When a capture is made the interrupt request flag, CCP1IF bit (PIR1<2>) is set. It must be reset in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost. In capture mode, the RC2/CCP1 pin should be configured as an input through its corresponding TRIS bit.

Note: If the RC2/CCP1 is configured as an output, a write to the port can cause a capture condition.

When the capture mode is changed, a false capture interrupt may be generated. The user should keep CCP1IE clear to avoid false interrupts and should clear the CCP1IF bit following any such change in operating mode.

FIGURE 10-1: CCP1CON/CCP2CON REGISTER



Register: CCP1CON Address: 17h Register: CCP2CON Address: 1Dh POR value: 00h	W: Writable R: Readable U: Unimplemented

CCPxM <3:0>: CCPX mode select.
0000 = Capture/compare/PWM off (Resets CCPX module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set). CCPX pin is unaffected
1011 = Compare mode, trigger special event
- CCP1 resets TMR1
- CCP2 resets TMR1

11XX = PWM mode

CCPxX:CCPxY: Two low order bits.
Capture mode Unused
Compare mode Unused
PWM mode Write the two low order bits in high resolution (10-bit) mode. May be kept constant (at '0') if only 8-bit resolution (in standard resolution mode) is desired.

Unimplemented, read as '0'.

10.1.1 PRESCALER

There are four prescaler settings, specified by the CCP1M3-CCP1M0 bits. Whenever the CCP module is turned off, or the CCP module is not in capture mode, the prescaler counter is cleared. This means that any reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, and therefore the first capture may be from a non-zero prescaler. Example 10-1 shows the recommended way to switch between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

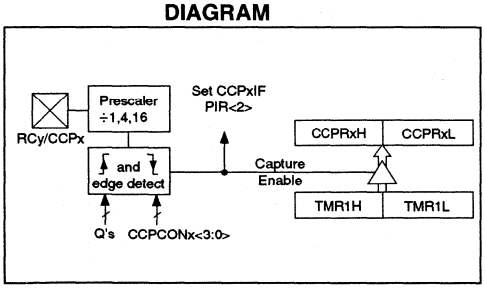
EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON      ; Turn CCP module off
MOVLW   NEW_CAPT_PS ; Load the W reg with
                     ; the new prescaler
                     ; mode value and CCP ON
MOVWF   CCP1CON      ; Load CCP1CON with
                     ; this value
```

10.1.2 TMR1 MODE SELECTION

TMR1 must be running in timer mode or synchronized counter mode for the CCP1 modules to use the capture feature. In asynchronous counter mode the capture operation may not work.

FIGURE 10-2: CAPTURE MODE OPERATION BLOCK DIAGRAM



10.2 Compare Mode

In compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1. When a match occurs, the RC2/CCP1 pin is:

- Driven High
- Driven Low
- Remains Unchanged

The action on the pin is based on the control bits CCP1M <3:0> in register CCP1CON. At the same time, a compare interrupt is also generated. The user must set the RC2/CCP1 pin as an output through the TRISC<2> bit.

Note: Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the data latch.

10.2.1 TIMER1 MODE SELECTION

Timer1 must be running in timer mode or synchronized counter mode if the CCP1 module is using the compare feature. In asynchronous counter mode, the compare operation may not work.

10.2.2 SOFTWARE INTERRUPT MODE

Another compare mode is software interrupt mode in which the CCP1 pin is not affected. Only CCP1IF interrupt is generated.

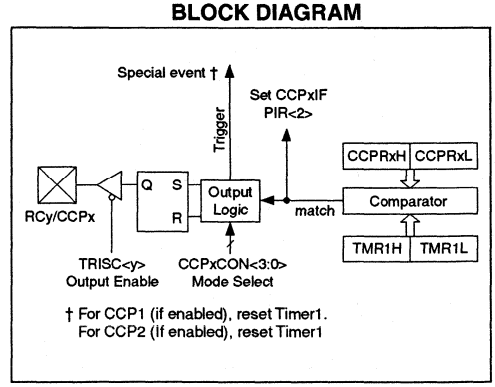
10.2.3 SPECIAL TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The special trigger output of CCP1 and CCP2 resets the TMR1. This allows the CCPR1 and CCPR2 registers to effectively be 16-bit programmable period registers for Timer1.

For compatability issues, the special trigger output of CCP2 on PIC16C7X devices also starts an A/D conversion.

FIGURE 10-3: COMPARE MODE OPERATION BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

10.3 PWM Mode

In Pulse Width Modulation (PWM) mode, the RC2/CCP1 produces up to 10-bit resolution PWM output. This pin must be configured as an output through the TRISC<2> bit. The pin is multiplexed with the data latch. In PWM mode, the user writes the 8-bit duty-cycle value to the low byte of the CCPR1 register, namely CCPR1L. The high-byte, CCPR1H is used as the slave buffer to the low byte. The 8-bit data is transferred from the master to the slave when the PWM1 output is set (i.e. at the beginning of the duty cycle). This double buffering is essential for glitchless PWM output. In PWM mode, CCPR1H is readable but not writable. The period of the PWM is determined by the Timer2 period register (PR2).

PWM period is =

$$[(PR2) + 1] \bullet 4\ T_{OSC} \bullet (\text{TMR2 prescale value})$$

PWM duty cycle =

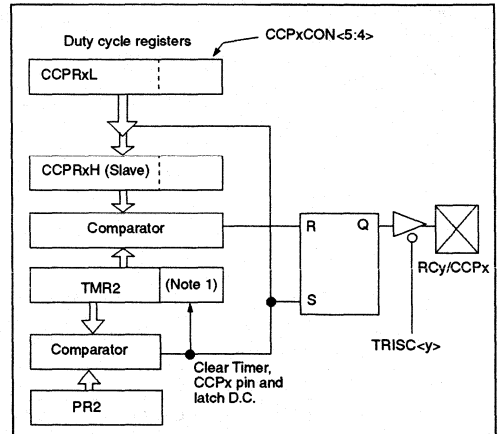
$$(DC1) \bullet T_{OSC} \bullet (\text{TMR2 prescale value})$$

where DC1 = 10 bit value from CCPRxL and CCPxCON<5:4> concantenated.

The PWM output resolution is therefore programmable up to a maximum of 10-bit.

Note: Clearing the CCP1CON register will force the RC2/CCP1 PWM output latch to the default low level. This is not the I/O data latch. The TMR2 postscaler is not used in the determination of the PWM frequency. The use of the postscaler could be used to have a servo update rate at a different frequency than the PWM output.

FIGURE 10-4: SIMPLIFIED PWM BLOCK DIAGRAM



Note: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

TABLE 10-1: PWM FREQUENCY VS RESOLUTION AT 20 MHZ

Max Resolution (High Resolution Mode)	Frequency		
	TMR2 Prescale=1	TMR2 Prescale=4	TMR2 Prescale=16
10 bit	19.53 KHz	4.88 KHz	1.22 KHz
9 bit	39.06 KHz	9.77 KHz	2.44 KHz
8 bit	78.13 KHz	19.53 KHz	4.88 KHz

TABLE 10-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHZ

PWM Frequency	1.22 KHz	4.88 KHz	19.53 KHz	78.12 KHz	156.3 KHz	208.3 KHz
Timer Prescaler (1,4,16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x5F
Resolution (High-resolution mode)	10-bit	10-bit	10-bit	8-bit	7-bit	6.5-bit
Resolution (Standard-resolution mode)†	8-bit	8-bit	8-bit	6-bit	5-bit	4.5-bit

† Standard resolution mode has the CCPIX:CCPIY bit constant (or '0'), and only compares the TMR2 against the PR2. The Q-cycles are not used.

TABLE 10-3: REGISTERS ASSOCIATED WITH TIMER1 AND CAPTURE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
15	CCPR1L	Timer1 Capture Register (LSb)							
16	CCPR1H	Timer1 Capture Register (MSb)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L[1]	Timer1 Capture Register (LSb)							
1C	CCPR2H[1]	Timer1 Capture Register (MSb)							
1D	CCP2CON[1]	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0'

Note: Shaded boxes are not used in this mode.

Note 1: The bits or registers are not implemented on the PIC16C64.

† These bits are reserved on the PIC16C65.

TABLE 10-4: REGISTERS ASSOCIATED WITH TIMER1 AND COMPARE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
15	CCPR1L	Timer1 Capture Register (LSb)							
16	CCPR1H	Timer1 Capture Register (MSb)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L[1]	Timer1 Capture Register (LSb)							
1C	CCPR2H[1]	Timer1 Capture Register (MSb)							
1D	CCP2CON[1]	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0'

Note: Shaded boxes are not used in this mode.

Note 1: The bits or registers are not implemented on the PIC16C64.

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

TABLE 10-5: REGISTERS ASSOCIATED WITH TIMER2 AND PWM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
11	TMR2	Timer2							
92	PR2	Timer2 period Register							
12	T2CON	—	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	T2CKPS1	T2CKPS0
15	CCPR1L	Timer2 Duty Cycle Register							
16	CCPR1H	Timer2 Duty Cycle Register (Slave)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L[1]	Timer2 Duty Cycle Register							
1C	CCPR2H[1]	Timer2 Duty Cycle Register (Slave)							
1D	CCP2CON[1]	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0'

Note: Shaded boxes are not used in this mode.

Note 1: The bits or registers are not implemented on the PIC16C64.

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

NOTES:

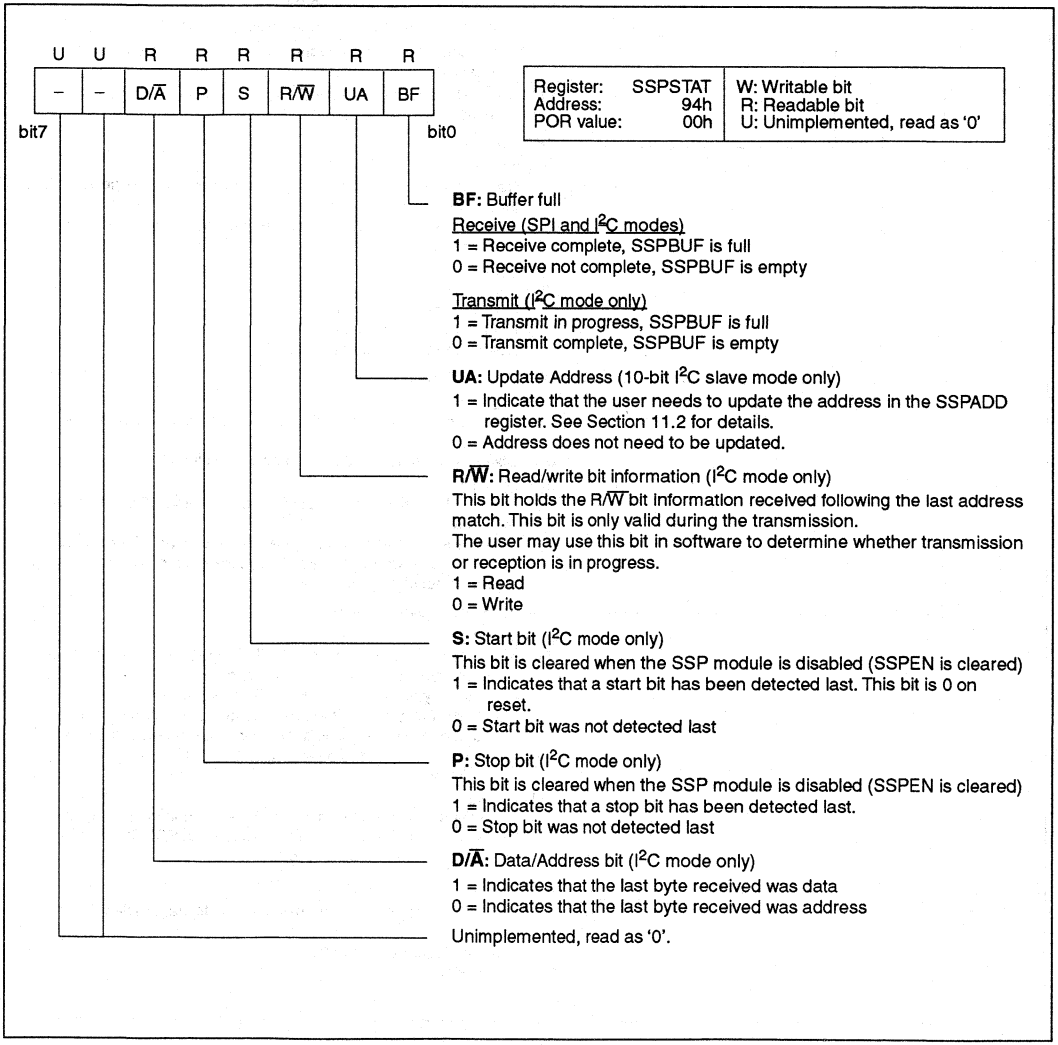
© 1995 Microchip Technology Inc.

11.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

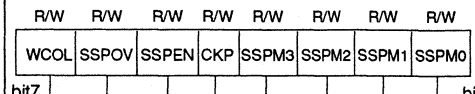
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

FIGURE 11-1: SSPSTAT: SYNCHRONOUS SERIAL PORT STATUS REGISTER



© 1995 Microchip Technology Inc.

FIGURE 11-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER



Register:	SSPCON	W: Writable bit
Address:	14h	R: Readable bit
POR value:	00h	U: Unimplemented, read as '0'

SSPM<3:0>: Synchronous serial port mode select

0000 = SPI master mode, clock = OSC/4
0001 = SPI master mode, clock = OSC/16
0010 = SPI master mode, clock = OSC/64
0011 = SPI master mode, clock = (TMR2 output/2)
0100 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
0110 = I^2C slave mode, 7-bit address
0111 = I^2C slave mode, 10-bit address
1011 = I^2C master mode support enabled (slave idle)
1110 = I^2C slave mode, 7-bit address with master mode support enabled
1111 = I^2C slave mode, 10-bit address with master mode support enabled

CKP: Clock polarity select.

In SPI modes:
1 = Transmit happens on falling edge, receive on rising edge. Idle state for clock is a High level.
0 = Transmit happens on rising edge, receive on falling edge. Idle state for clock is a Low level.

In I^2C modes:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch)
Note: Used to ensure data setup time

SSPEN: Sync serial port enable

In SPI modes:
1 = Enables serial port and configures SCK, SDO and SDI as serial port pins.
0 = Disables serial port and configures these pins as I/O port pins.

In I^2C modes:
1 = Enables the serial port and configures SDA and SCL pins as serial port pins.
0 = Disables serial port and configures these pins as I/O port pins.
In both modes, when enabled, these pins must be properly configured as input or output.

SSPOV: Receive overflow flag.

In SPI modes:
1 = A new byte is received while SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode overflow bit is not set since each new reception (and transmission) is initiated by writing to SSPBUF.
0 = No overflow

In I^2C modes:
1 = A byte is received while the SSPBUF is still holding the previous byte. SSPOV is a don't care in transmit mode. SSPOV must be cleared in software in either mode.

WCOL: Write collision detect.

1 = the SSPBUF register is written while it is still transmitting the previous word.
Must be cleared in software.
0 = No collision

© 1995 Microchip Technology Inc.

11.1 SPI Mode

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI
- Serial Clock (SCK) RC3/SCK

Additionally a fourth pin may be used when in a slave mode of operation:

- Slave Select ($\overline{SS}$) RA5/$\overline{SS}$

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bit in the SSPCON register (SSPCON<5:0>). These control bits allow the following to be specified:

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

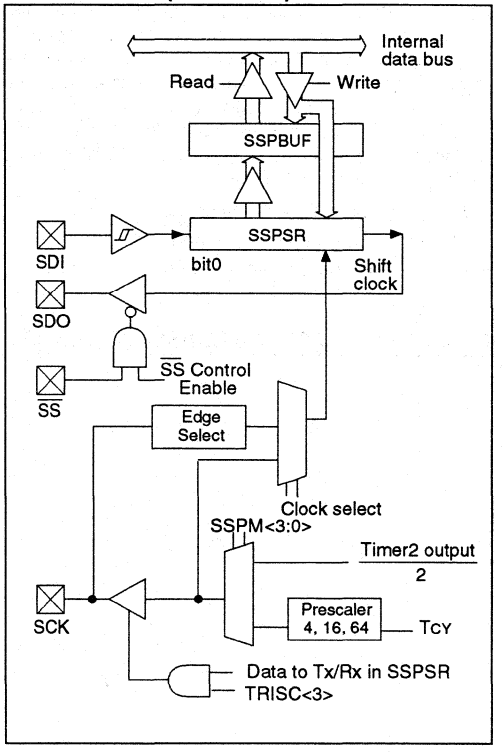
The SSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSB first, while the SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that information is moved to the SSPBUF register, the Buffer Full (BF) bit (SSPSTAT <0>) and the SSPIF bit are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect (WCOL) bit (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF completed successfully. When the application software is expecting to receive valid data, the SPPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full (BF) bit (SSPSTAT<0>) indicates when the SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF can then be read (if data is meaningful) and/or the SSPBUF (SSPSR) can be written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-1 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOPBSF   STATUS, RP0  ;Specify Bank 1
  BTFSS   SSPSTAT, BF  ;Has data been
                       ;received
                       ;(transmit
                       ;complete)?
  GOTO    LOOP         ;No
  BCF     STATUS, RP0  ;Specify Bank 0
  MOVF    SSPBUF, W    ;W reg = contents
                       ;of SSPBUF
  MOVWF   RXDATA       ;Save in user RAM
  MOVF    TXDATA, W    ;W reg = contents of '
                       ; TXDATA
  MOVWF   SSPBUF       ;New data to xmit.
```

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)



To enable the serial port, the SSP enable bit (SSPEN) must be set. To reset or reconfigure SPI mode, clear SPEN, re-initialize SSPCON, and then set SSPEN. This configures the SDI, SDO, SCK, and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and $\overline{SS}$ could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-4 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2) wishes to broadcast data by the software protocol.

In master mode the data is transmitted/received as soon as the SSPBUF is written to. If the SPI is only going to receive, the SCK output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor" mode.

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag (SSPIF) is set (PIR1<3>).

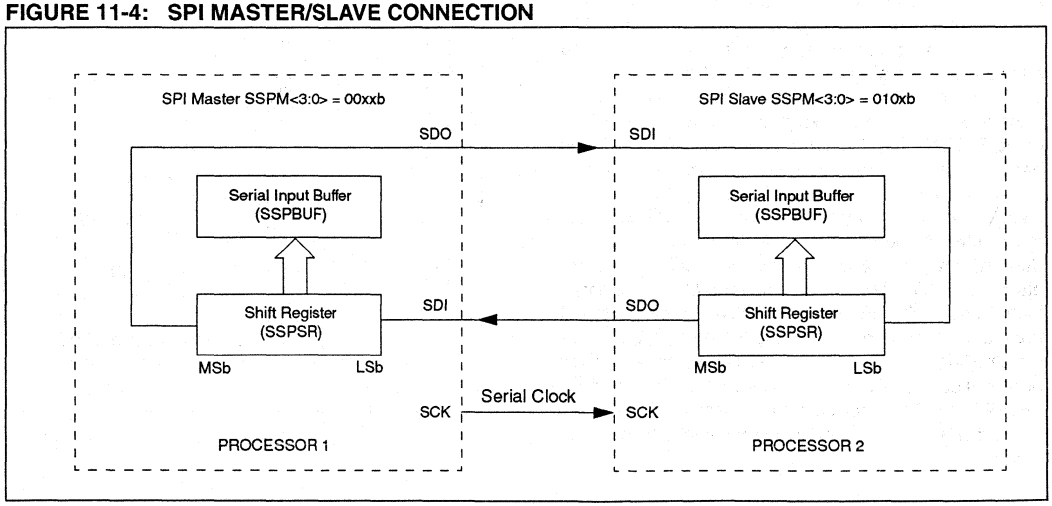
The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-5 and Figure 11-6 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- OSC / 4 (or TCY)
- OSC / 16 (or 4 • TCY)
- OSC / 64 (or 16 • TCY)
- Timer2 output / 2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times (see Table 16-7).

In sleep mode, the slave can transmit and receive data and wake-up the device from sleep.

FIGURE 11-4: SPI MASTER/SLAVE CONNECTION



© 1995 Microchip Technology Inc.

The $\overline{SS}$ pin allows a synchronous slave mode. The SPI must be in slave mode (SSPCON<3:0> = 04h) and the TRISA<5> bit must be set the for the synchronous slave mode to be enabled. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/ pull-down resisters may be desirable, depending on the application.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 11-5: SPI MODE TIMING (MASTER MODE OR SLAVE MODE W/O $\overline{SS}$ CONTROL)

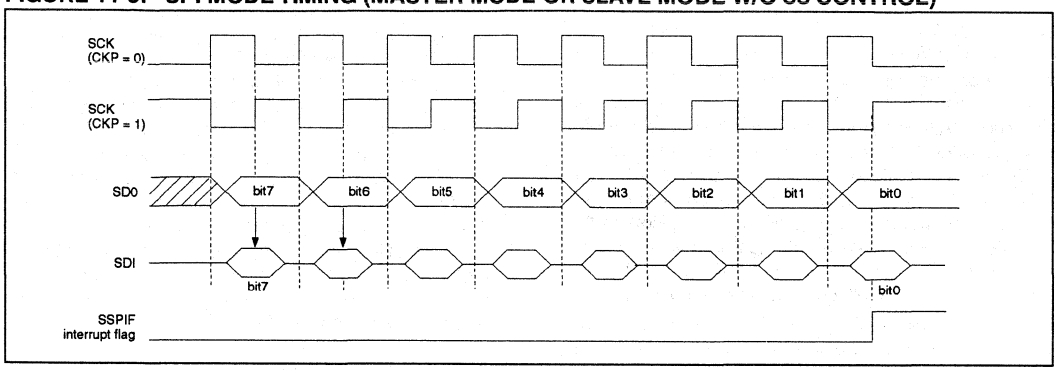


FIGURE 11-6: SPI MODE TIMING (SLAVE MODE WITH $\overline{SS}$ CONTROL)

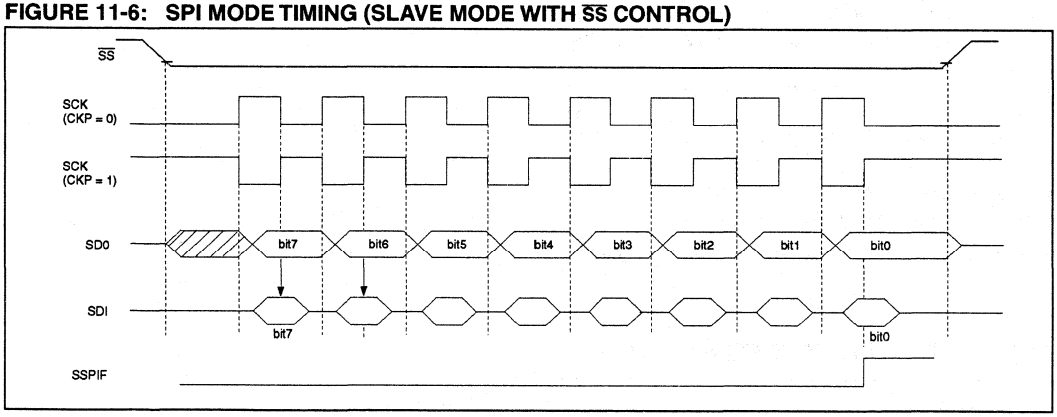


TABLE 11-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
8C	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE
13	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
14	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
94	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by SSP module in SPI mode.
Note 1: These bits are not implemented on the PIC16C64.
† These bits are reserved on the PIC16C65.

11.2 I²C Overview

This section provides an overview of the Inter-IC (I^2C) bus, with Section 11.3 discussing the operation of the SSP module in I^2C mode. The I^2C bus is a two-wire serial interface developed by the Philips Corporation. The original specification, or standard mode, was for data transfers of up to 100 Kbps. An enhanced specification, or fast mode, supports data transmission up to 400 Kbps. Both standard mode and fast mode devices will inter-operate if attached to the same bus.

The I^2C interface employs a comprehensive protocol to ensure reliable transmission and reception of data. When transmitting data, one device is the "master" (generates the clock), while the other device(s) acts as the "slave." All portions of the slave protocol are implemented in the SSP module's hardware, while portions of the master protocol need to be addressed in the PIC16CXX software. Table 11-2 defines some of the I^2C-bus terminology. For additional information on the I^2C interface specification, refer to the Philips Corporation document *"The I^2C-bus and how to use it."* The order number for this document is 98-8080-575.

In the I^2C interface protocol each device has an address. When a master wishes to initiate a data transfer, it first transmits the address of the device that it wishes to "talk" to. All devices "listen" to see if this is their address. Within this address, a bit specifies if the master wishes to read / write from / to the slave device. The master and slave are always in opposite modes (transmitter/receiver) of operation during a data transfer. That is they can be thought of operating in either of these two relations:

- Master-transmitter and Slave-receiver
- Slave-transmitter and Master-receiver

In both cases the master generates the clock signal.

The output stages of the clock (SCL) and data (SDA) lines must have an open-drain or open-collector in order to perform the wired-AND function of the bus. External pull-up resistors are used to ensure a high level, when no device is pulling the line down. The number of devices that may be attached to the I^2C bus is limited only by the maximum bus loading specification of 400 pF.

11.2.1 INITIATING AND TERMINATING DATA TRANSFER

During times of no data transfer (idle time), both the clock line (SCL) and the data line (SDA) are pulled high through the external pull-up resistors. The START and STOP conditions determine the start and stop of data transmission. The START condition is defined as a high to low transition of the SDA when the SCL is high. The STOP condition is defined as a low to high transition of the SDA when the SCL is high. Figure 11-7 shows the START and STOP conditions. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP conditions, when data is being transmitted the SDA line can only change state when the SCL line is low.

FIGURE 11-7: START AND STOP CONDITIONS

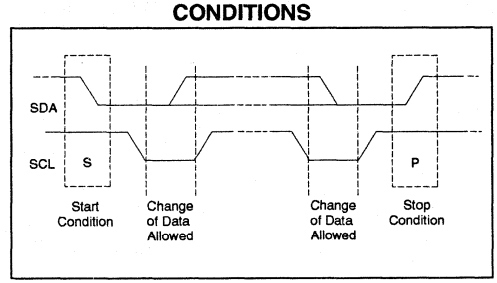


TABLE 11-2: I²C-BUS TERMINOLOGY

Term	Description
Transmitter	The device that sends the data to the bus
Receiver	The device that receives the data from the bus
Master	The device which initiates the transfer, generates the clock and terminates the transfer
Slave	The device addressed by a master
Multi-master	More than one master device in a system. These masters can attempt to control the bus at the same time without corrupting the message.
Arbitration	Procedure that ensures that only one of the master devices will control the bus. This ensure that the transfer data does not get corrupted.
Synchronization	Procedure where the clock signals of two or more devices are synchronized.

© 1995 Microchip Technology Inc.

11.2.2 ADDRESSING I²C DEVICES

There are two address formats. The simplest is the 7-bit address format with a R/$\overline{W}$ bit (see Figure 11-8). The more complex is the 10-bit address with a R/$\overline{W}$ bit (see Figure 11-9). For 10-bit address format, two bytes must be transmitted with the first five bits specifying this to be a 10-bit address.

FIGURE 11-8: 7-BIT ADDRESS FORMAT

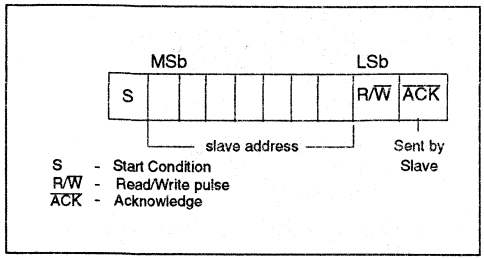
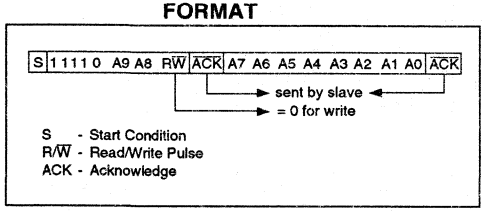


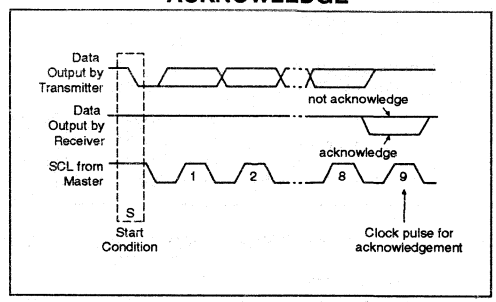
FIGURE 11-9: I²C 10-BIT ADDRESS FORMAT



11.2.3 TRANSFER ACKNOWLEDGE

All data must be transmitted per byte, with no limit to the number of bytes transmitted per data transfer. After each byte, the slave-receiver generates an acknowledge bit ($\overline{ACK}$). This is shown in Figure 11-10. When a slave-receiver doesn't acknowledge the slave address or received data, the master must abort the transfer. The slave must leave SDA high so that the master can generate the STOP condition (see Figure 11-7).

FIGURE 11-10: SLAVE-RECEIVER ACKNOWLEDGE



If the master is receiving the data (master-receiver), it generates an acknowledge signal for each received byte of data, except for the last byte. To signal the end of data to the slave-transmitter, the master does not generate an acknowledge (not acknowledge). The slave then releases the SDA line so the master can generate the STOP condition. The master can also generate the STOP condition during the acknowledge pulse for valid termination of data transfer.

If the slave needs to delay the transmission of the next byte, holding the SCL line low will force the master into a wait state. Data transfer continues when the slave releases the SCL line. This allows the slave to move the received data or fetch the data it needs to transfer before allowing the clock to start. This wait state technique can also be implemented at the bit level. Figure 11-11 shows a data transfer waveform.

FIGURE 11-11: A DATA TRANSFER

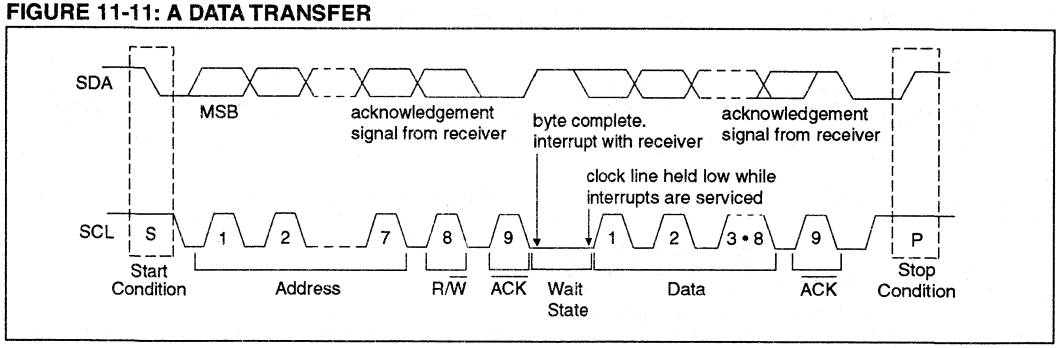


Figure 11-12 and Figure 11-13 show Master-transmitter and Master-receiver data transfer sequences.

When a master does not wish to relinquish the bus (by generating a STOP condition), a repeated START condition (Sr) must be generated. This condition is identical to the start condition (SDA goes high-to-low while SCL is high), but occurs after a data transfer acknowledge pulse (not the bus-free state). This allows a master to send "commands" to the slave and then receive the requested information or to address a different slave device. This sequence is shown in Figure 11-14.

FIGURE 11-12: MASTER-TRANSMITTER SEQUENCE

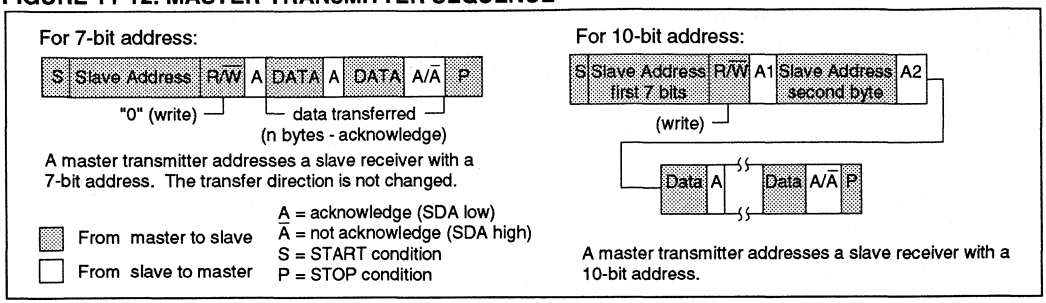


FIGURE 11-13: MASTER-RECEIVER SEQUENCE

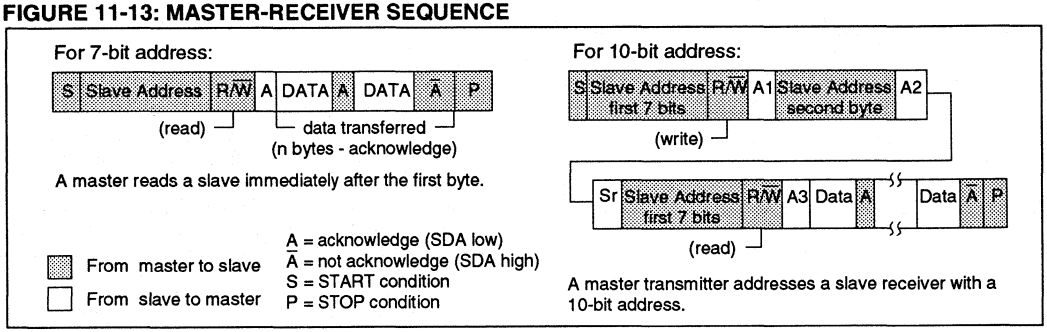
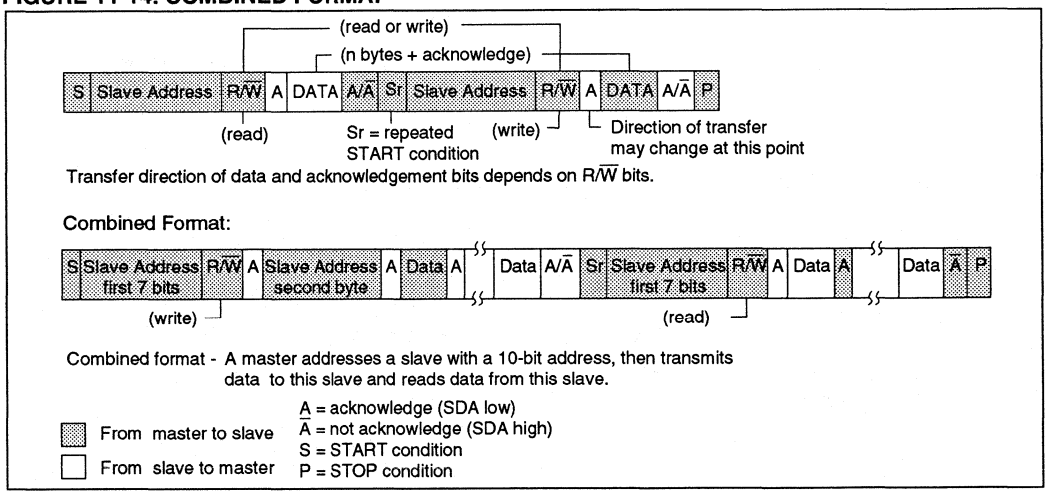


FIGURE 11-14: COMBINED FORMAT



© 1995 Microchip Technology Inc.

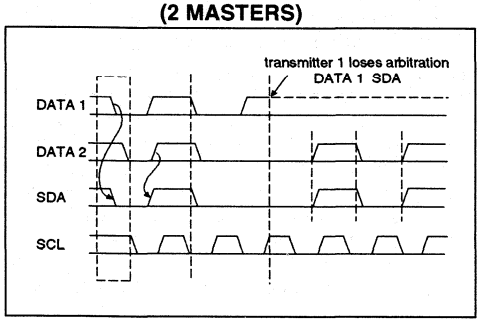
11.2.4 MULTI-MASTER

The I^2C protocol allows a system to have more than one master. This is called multi-master. When two or more masters try to transfer data at the same time, arbitration and synchronization occur.

11.2.4.1 ARBITRATION

Arbitration takes place on the SDA line, while the SCL line is high. The master which transmits a high when the other master transmits a low loses arbitration (see Figure 11-15), and turns off its data output stage. A master which lost arbitration can generate clock pulses until the end of the data byte where it lost arbitration. When the master devices are addressing the same device, arbitration continues into the data.

FIGURE 11-15: MULTI-MASTER ARBITRATION (2 MASTERS)



Masters that also incorporate the slave function, and have lost arbitration must immediately switch over to slave-receiver mode. This is because the winning master-transmitter may be addressing it.

Arbitration is not allowed between:

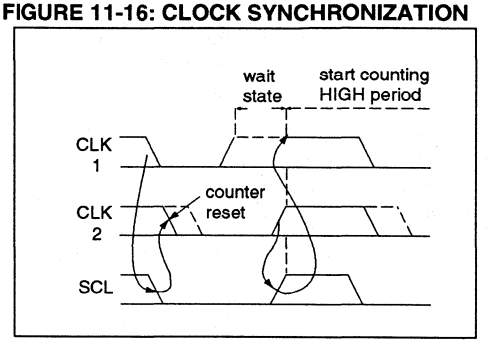
- A repeated START condition
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Care needs to be taken to ensure that these conditions do not occur.

11.2.4.2 Clock Synchronization

Clock synchronization occurs after the devices have started arbitration. This is performed using a wired-AND connection to the SCL line. A high to low transition on the SCL line causes the concerned devices to start counting off their low period. Once a device clock has gone low, it will hold the SCL line low until its SCL high state is reached. The low to high transition of this clock may not change the state of the SCL line, if another device clock is still within its low period. The SCL line is held low by the device with the longest low period. Devices with shorter low periods enter a high wait-state, until the SCL line comes high. When the SCL line comes high, all devices start counting off their high periods. The first device to complete its high period will pull the SCL line low. The SCL line high time is determined by the device with the shortest high period. This is shown in Figure 11-16.

FIGURE 11-16: CLOCK SYNCHRONIZATION

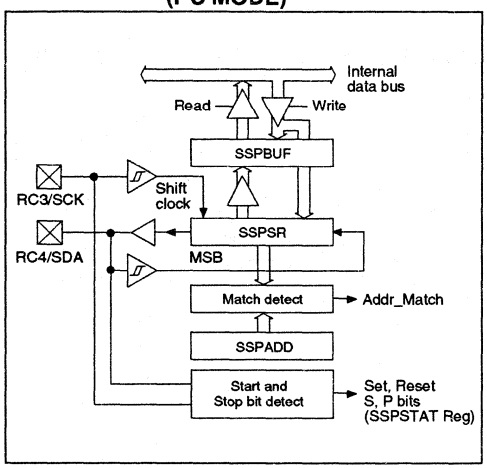


© 1995 Microchip Technology Inc.

11.3 SSP I²C Operation

The SSP module in I²C mode fully implements all slave functions, and provides support in hardware to facilitate software implementations of the master functions. The SSP module implements the standard and fast mode specifications as well as 7-bit and 10-bit addressing. Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits. A block diagram of the SSP module in I²C mode is shown in Figure 11-17. The SSP module functions are enabled by setting the SSP Enable (SSPEN) bit (SSPCON<5>).

FIGURE 11-17: SSP BLOCK DIAGRAM (I²C MODE)



The SSP module has five registers for I²C operation. These are the:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive / Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- Address Register(SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allows one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with master-mode support enabled
- I²C Slave mode (10-bit address), with master-mode support enabled
- I²C Master mode, support enabled slave is idle

Selection of any I²C mode and with the SSPEN bit set, forces the SCL and SDA pins to be open collector, provided these pins are set to inputs through the TRISC bits.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer. The SSPSTAT register is read only.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF and the SSPIF is set. If another complete byte is received before the SSPBUF is read, a receiver overflow has occurred and the SSPOV bit (SSPCON<6> is set.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1 1 1 1 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7 - A0).

© 1995 Microchip Technology Inc.

11.3.1 SLAVE MODE

In slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched or the data transfer from an address match is received, the hardware automatically will generate the acknowledge ($\overline{ACK}$) pulse, and then load the SSPBUF with the received value in the SSPSR.

There are certain conditions that will cause the SSP module not to give this $\overline{ACK}$ pulse. These are if either (or both):

- The Buffer Full (BF) bit was set before the transfer was received.
- The Overflow (SSPOV) bit was set before the transfer was received.

In this case, the SSPSR value is not loaded into the SSPBUF, but the SSPIF bit is set. Table 11-3 shows what happens when a data transfer byte is received, given the status of the BF and SSPOV bits. The shaded boxes shows the condition where user software did not properly clear the overflow condition. The BF flag is cleared by reading the SSPBUF register while the SSPOV bit is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I^2C specification as well as the requirement of the SSP module is shown in timing parameter #100 and parameter #101.

11.3.1.1 ADDRESSING

Once the SSP module has been enabled, The SSP waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR. All incoming bits are sampled with the rising edge of the clock (SCL) line. The SSPSR<7:1> is compared to the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following things happen:

- SSPSR loaded into SSPBUF
- Buffer Full (BF) bit is set
- $\overline{ACK}$ pulse is generated
- SSP Interrupt Flag (SSPIF) is set (interrupt is generated if enabled) - on falling edge of ninth SCL pulse

In 10-bit address mode, two address bytes need to be received by the slave (see Figure 11-9). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. The R/$\overline{W}$ bit (bit 0) must specify a write, so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1 1 1 1 0 A9 A8 0', where A9 and A8 are the two MSbs of the address. The sequence of events for 10-bit address are as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (SSPIF, BF and UA are set)
2. Update SSPADD with second (low) byte of Address (clears UA and releases SCL line)
3. Read SSPBUF (clears BF) and clear SSPIF
4. Receive second (low) byte of Address (SSPIF, BF and UA are set)
5. Update SSPADD with first (high) byte of Address (clears UA, if match releases SCL line)
6. Read SSPBUF (clears BF) and clear SSPIF
7. Receive Repeated START condition
8. Receive first (high) byte of Address (SSPIF and BF are set)
9. Read SSPBUF (clears BF) and clear SSPIF

TABLE 11-3: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received				
BF	**SSPOV**	**SSPSR →SPBUF**	**Generate $\overline{ACK}$ Pulse**	**Set SSPIF bit (SSP Interrupt if Enabled)**
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

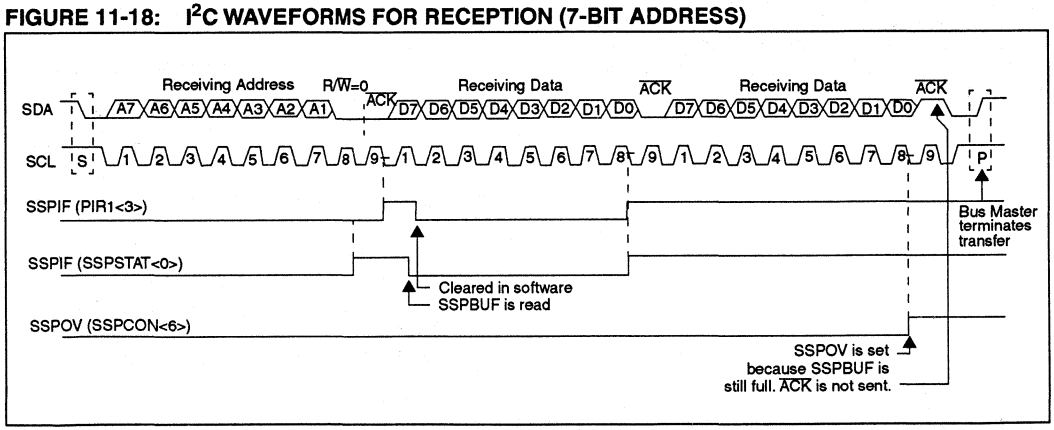
11.3.1.2 RECEPTION

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF.

When the address byte overflow condition exists then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either the BF bit (SSPSTAT<0>) is set or the SSPOV bit (SSPCON<6>) is set.

A SSPIF interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software, and the SSPSTAT register is used to determine the status of the byte.

FIGURE 11-18: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)



© 1995 Microchip Technology Inc.

11.3.1.3 TRANSMISSION

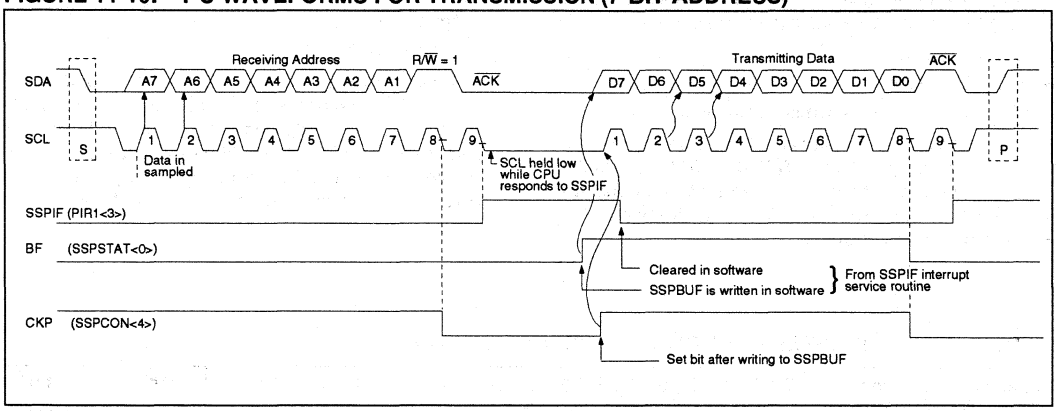
When the R/$\overline{W}$ bit of the address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF. The $\overline{ACK}$ pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit (SSPCON<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (see Figure 11-19).

A SSPIF interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. The slave then monitors for another occurrence of the START bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit (SSPCON<4>).

2

FIGURE 11-19: I^2C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



11.3.2 MASTER MODE

Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are cleared

In master mode the SCL and SDA lines are manipulated by changing the corresponding TRISC<4:3> bit(s) to an output (cleared). The output level is always low, irrespective of the value(s) in PORTB<4:3>. So when transmitting data, a "1" data bit must have the TRISC<4> bit set (input) and a "0" data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit.

The following events will cause the SSP Interrupt Flag (SSPIF) to be set (SSP Interrupt if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received

Master mode of operation can be done with either the slave mode idle (SSPM3 - SSPM0 = 1011) or with the slave active. When both master and slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

11.3.3 MULTI-MASTER MODE

In multi-master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are cleared. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the stop condition occurs.

In multi-master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, the device may be being addressed. If addressed an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

TABLE 11-4: REGISTERS ASSOCIATED WITH I^2C OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	PSPIF	†	RCIF[1]	TXIF[1]	SSPIF	CCP1IF	TMR2IF	TMR1IF
8Ch	PIE1	PSPIE	†	RCIE[1]	TXIE[1]	SSPIE	CCP1IE	TMR2IE	TMR1IE
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
93h	DSSPADD	Synchronous Serial Port (I^2C mode) Address Register							
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
94h	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF

Legend: — = Unimplemented locations, Read as '0'.

Note: Shaded boxes are not used by the SSP module in I^2C mode.

Note 1: These bits are not implemented on the PIC16C64.

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

FIGURE 11-20: OPERATION OF THE I²C MODULE IN IDLE_MODE, RCV_MODE OR XMIT_MODE

```
IDLE_MODE (7-bit):
if (Addr_match)                 {       Set interrupt;
                                        if (R/W = 1)    {       Send ACK = 0;
                                                                set XMIT_MODE;
                                                        }
                                        else if (R/W = 0) set RCV_MODE;
                                }
RCV_MODE:
if ((SSPBUF=Full) OR (SSPOV = 1))
        {       Set SSPOV;
                Do not acknowledge;
        }
else    {       transfer SSPSR → SSPBUF;
                send ACK = 0;
        }
Receive 8-bits in SSPSR;
Set interrupt;
XMIT_MODE:
While ((SSPBUF = Empty) AND (CKP=0)) Hold SCL Low;
Send byte;
Set interrupt;
if ( ACK Received = 1)          {       End of transmission;
                                        Go back to IDLE_MODE;
                                }
else if ( ACK Received = 0)  Go back to XMIT_MODE;
IDLE_MODE (10-Bit):
If (High_byte_addr_match AND (R/W = 0))
        {       PRIOR_ADDR_MATCH = FALSE;
                Set interrupt;
                if ((SSPBUF = Full) OR ((SSPOV = 1))
                        {       Set SSPOV;
                                Do not acknowledge;
                        }
                else    {       Set UA = 1;
                                Send ACK = 0;
                                While (SSPADD not updated) Hold SCL low;
                                Clear UA = 0;
                                Receive Low_addr_byte;
                                Set interrupt;
                                Set UA = 1;
                                If (Low_byte_addr_match)
                                        {       PRIOR_ADDR_MATCH = TRUE;
                                                Send ACK = 0;
                                                while (SSPADD not updated) Hold SCL low;
                                                Clear UA = 0;
                                                Set RCV_MODE;
                                        }
                        }
        }
else if (High_byte_addr_match AND (R/W = 1)
        {       if (PRIOR_ADDR_MATCH)
                        {       send ACK = 0;
                                set XMIT_MODE;
                        }
        else PRIOR_ADDR_MATCH = FALSE;
        }
```

NOTES:

© 1995 Microchip Technology Inc.

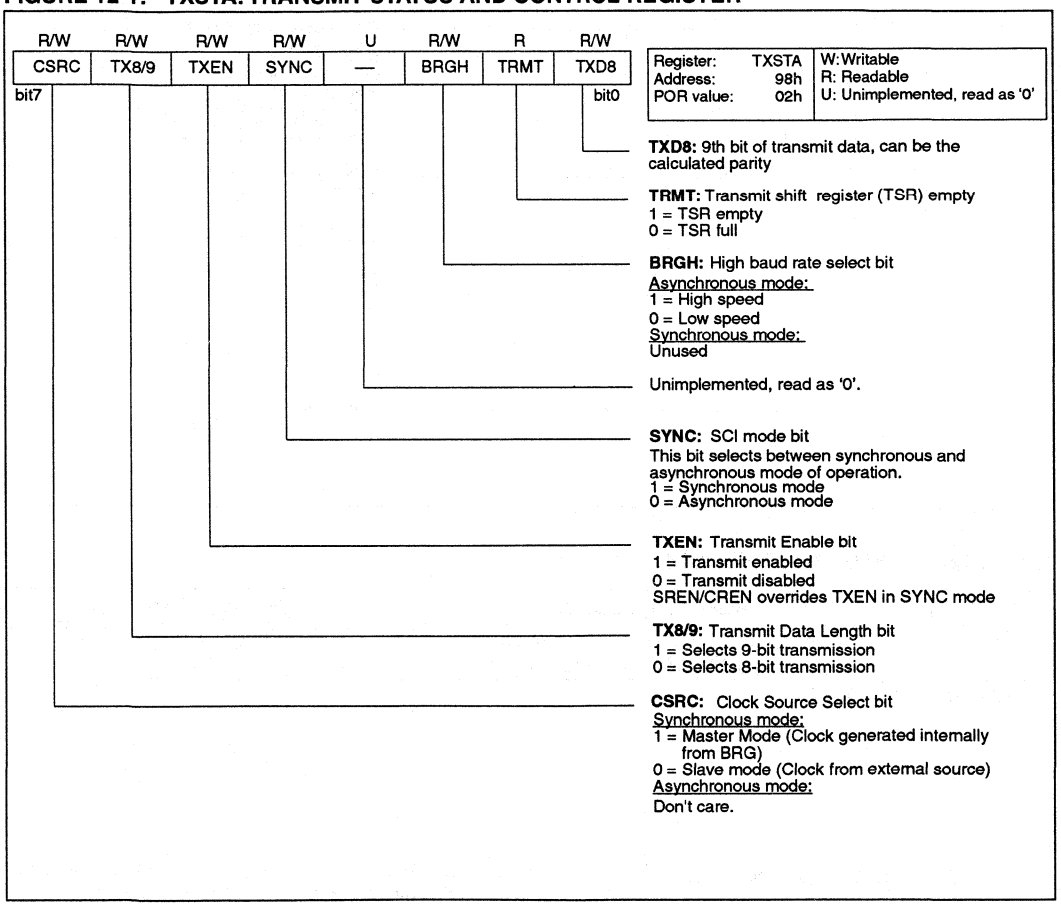
12.0 SERIAL COMMUNICATION INTERFACE (SCI) MODULE

The Serial Communication Interface (SCI) module, is one of the two serial I/O modules. The SCI can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc. The SCI can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

The SPEN (RCSTA<7>) bit has to be set in order to configure RC6 and RC7 as the Serial Communication Interface.

FIGURE 12-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER



© 1995 Microchip Technology Inc.

FIGURE 12-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

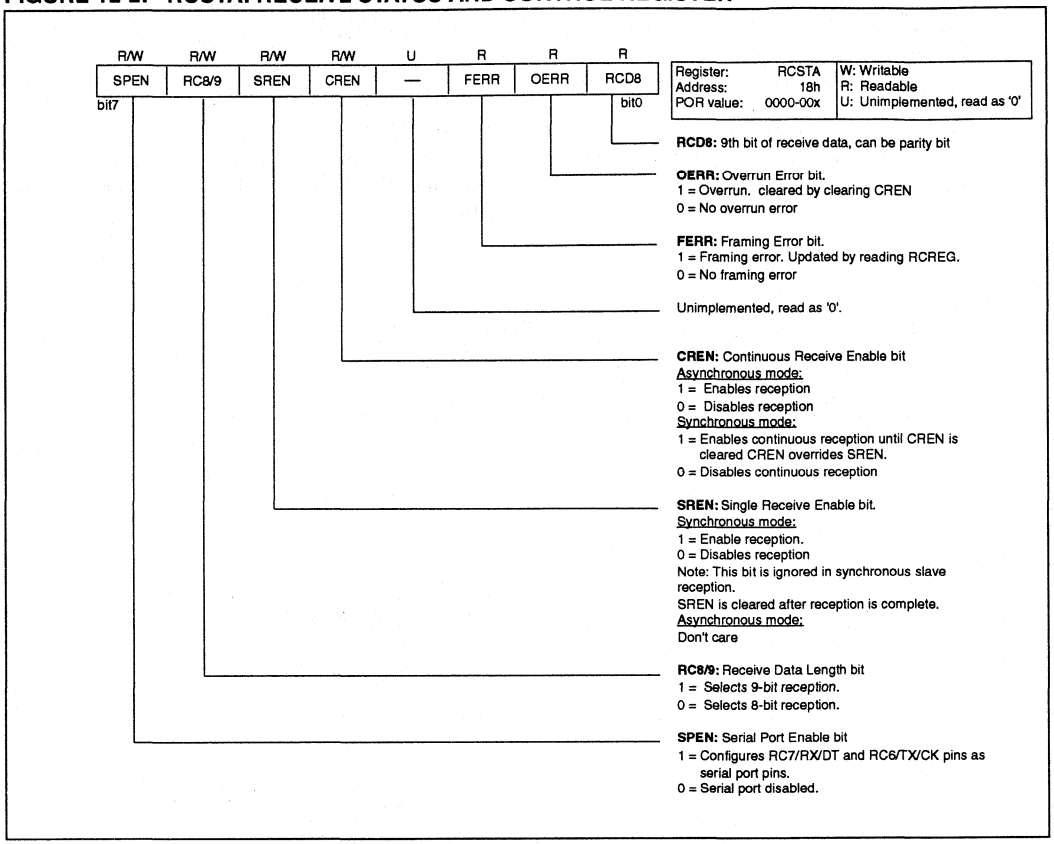


TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH=0 (Low Speed)	BRGH=1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

X = value in SPBRG (0 to 255)

12.1 SCI Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the SCI. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode the BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different SCI modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for SPBRG can be calculated using the formula in Table 12-1. From this, the error in baud rate can be determined.

© 1995 Microchip Technology Inc.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz
Desired Baud Rate = 9600
BRGH = 0
SYNC = 0

EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

Desired Baud rate=Fosc / (64 (X + 1))

9600 = 16000000 /(64 (X + 1))

X = ⌊25.042⌋ = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

Error = $\frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}}$

= (9615 - 9600) / 9600

= 0.16%

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the Fosc/(16(x + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPREG, causes the BRG timer to be reset (or cleared), this guarantees that the BRG does not wait for a timer overflow before outputting the new baud rate.

2

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0X98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	-	BRGH	TRMT	TXD8
0X99h	SPBRG	Baud Rate Register							

TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 20MHZ			16MHZ			10MHZ			7.15909MHZ		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	Fosc = 5.0688MHZ			3.579545MHZ			1MHZ			32.768KHZ		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

© 1995 Microchip Technology Inc.

TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH=0)

BAUD RATE (K)	Fosc = 20MHZ KBAUD	%ERROR	SPBRG value (decimal)	16MHZ KBAUD	%ERROR	SPBRG value (decimal)	10MHZ KBAUD	%ERROR	SPBRG value (decimal)	7.15909MHZ KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	Fosc = 5.0688MHZ KBAUD	%ERROR	SPBRG value (decimal)	3.579545MHZ KBAUD	%ERROR	SPBRG value (decimal)	1MHZ KBAUD	%ERROR	SPBRG value (decimal)	32.768KHZ KBAUD	%ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH=1)

BAUD RATE (K)	Fosc = 20MHZ KBAUD	%ERROR	SPBRG value (decimal)	16MHZ KBAUD	%ERROR	SPBRG value (decimal)	10MHZ KBAUD	%ERROR	SPBRG value (decimal)	7.16MHZ KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (K)	Fosc = 5.068MHZ KBAUD	%ERROR	SPBRG value (decimal)	3.579MHZ KBAUD	%ERROR	SPBRG value (decimal)	1MHZ KBAUD	%ERROR	SPBRG value (decimal)	32.768KHZ KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.6	0	32	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-

© 1995 Microchip Technology Inc.

12.1.1 SAMPLING

The data on the RX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. If BRGH is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a 16x clock (see Figure 12-3). If BRGH is set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a 4x clock (see Figure 12-4 and Figure 12-5).

FIGURE 12-3: RX PIN SAMPLING SCHEME (BRGH=0)

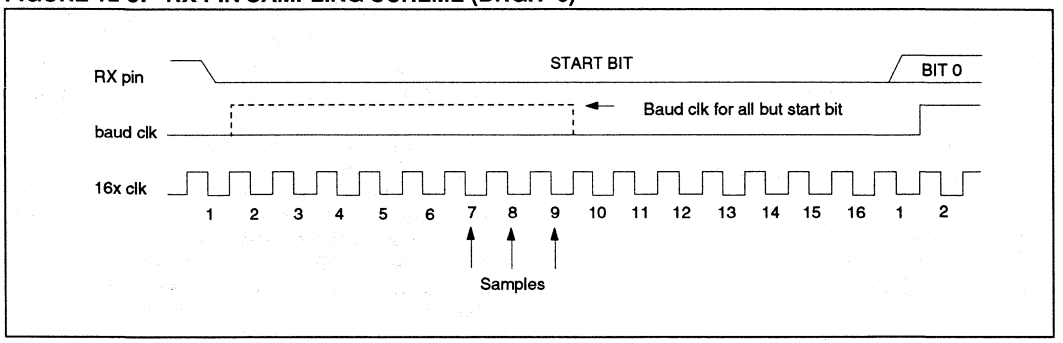


FIGURE 12-4: RX PIN SAMPLING SCHEME (BRGH=1)

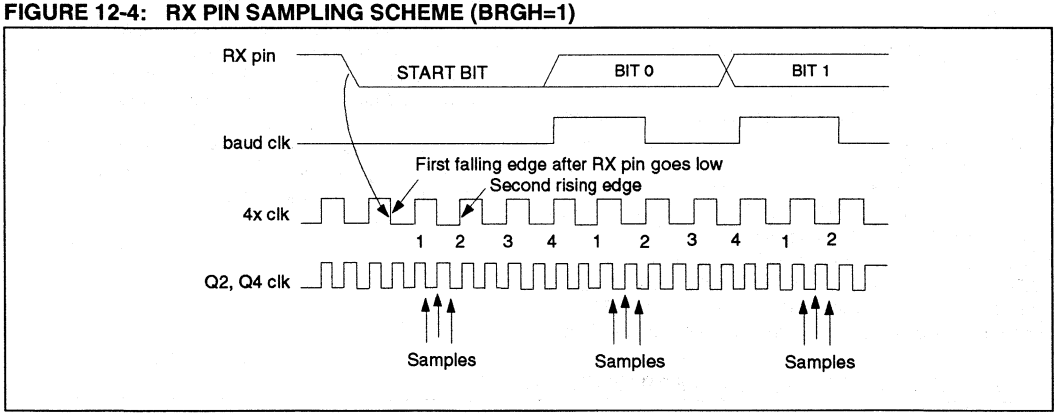
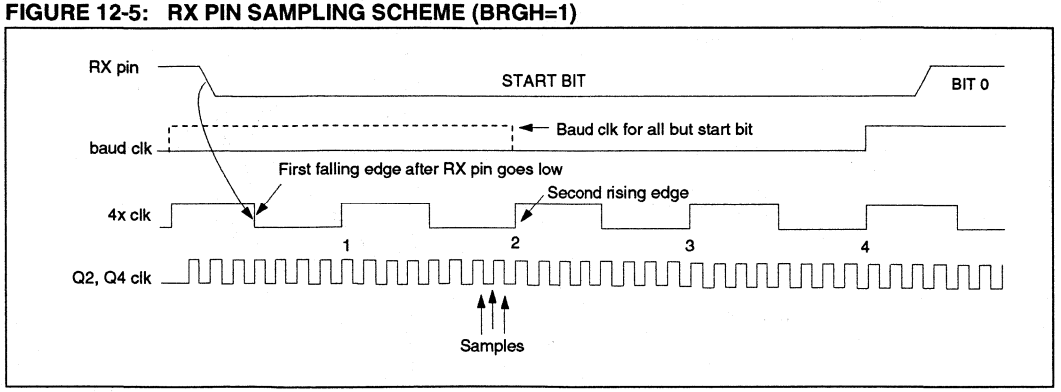


FIGURE 12-5: RX PIN SAMPLING SCHEME (BRGH=1)



12.2 SCI Asynchronous Mode

In this mode, the SCI used standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The SCI transmits and receives the LSB bit first. The SCI's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock either 16x or 64x of the bit shift rate, depending on the BRGH (TXSTA<2>) bit. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The asynchronous mode is selected by resetting the SYNC bit (TXSTA<4>).

The SCI Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

12.2.1 SCI ASYNCHRONOUS TRANSMITTER

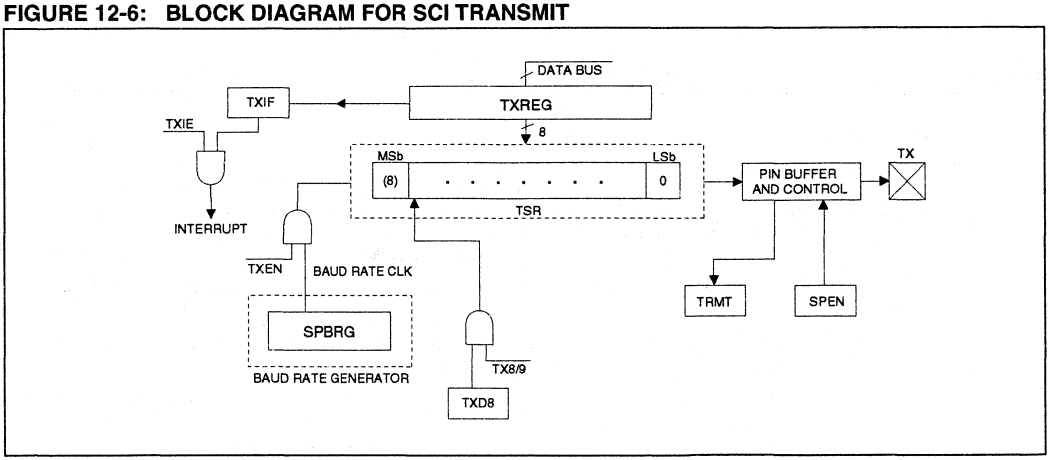
The SCI transmitter block diagram is shown in Figure 12-6. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). The TXREG is loaded with data in software. The TSR is not loaded until the stop bit has been transmitted from the previous load. As soon as the stop bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG transfers the data to the TSR(occurs in one TCY), the TXREG is empty and a interrupt bit, TXIF (PIR1<4>) is set. This interrupt can be enabled or disabled by the TXIE bit (PIE1<4>). TXIF will be set regardless of TXIE and cannot be reset in software. It will reset only when new data is loaded into the TXREG. While TXIF indicated the status of the TXREG, another bit TRMT(TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

Note: The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until the TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 12-6). The transmission can also be started by first loading the TXREG and then setting TXEN. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 12-8). Clearing TXEN during a transmission, will cause the transmission to be aborted and will reset the transmitter as a result the TX pin will revert to hi-impedance.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to the TXD8(TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). In such a case, an incorrect ninth data bit maybe loaded in the TSR.

FIGURE 12-6: BLOCK DIAGRAM FOR SCI TRANSMIT



© 1995 Microchip Technology Inc.

Steps to follow when setting up a Asynchronous Transmission:

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set BRGH to 1. See Section 12.1 for details.
- Enable the asynchronous serial port by configuring the bits SYNC = 0 and SPEN = 1.
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9 bit should be set.
- Enable the transmission by setting TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Load data to the TXREG (starts transmission).

FIGURE 12-7: ASYNCHRONOUS MASTER TRANSMISSION

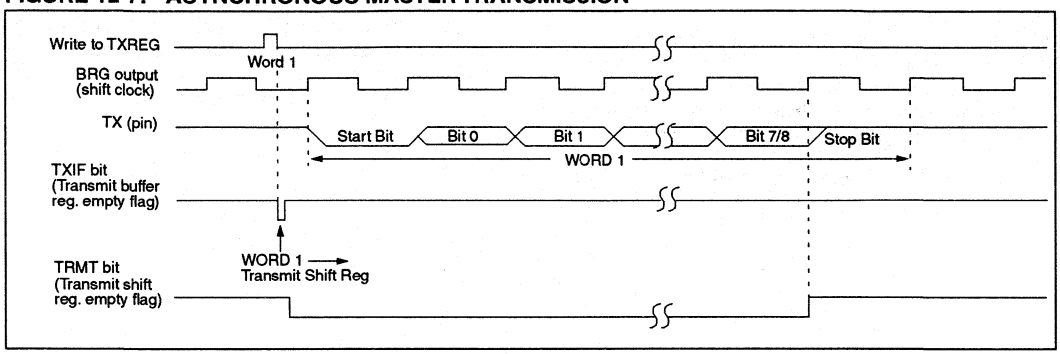
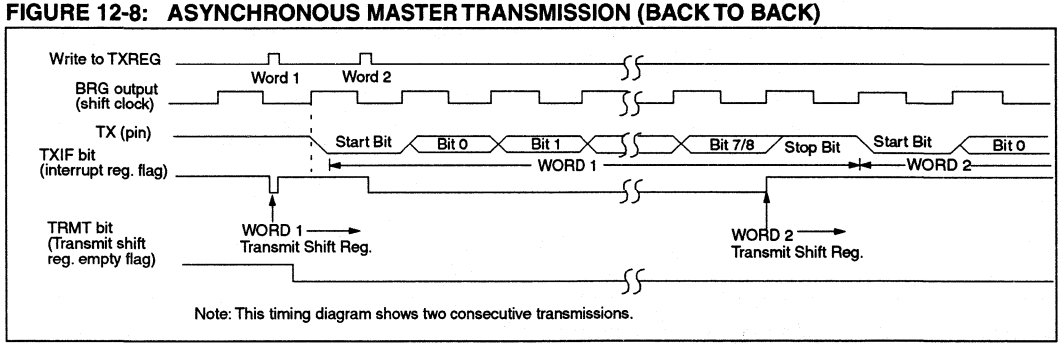


FIGURE 12-8: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION:

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD6
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	PSPIE	†	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

12.2.2 SCI ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 12-9. The data comes in the RX pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once the asynchronous mode is selected, reception is enabled by setting CREN(RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the stop bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled or disabled by the RCIE(PIE1<5>) bit. The RCIF is a read only bit which is reset by the hardware. It is cleared when the RCREG has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On the detection of the stop bit of the third byte, if the RCREG is still full then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software and this is done by resetting the receive logic (CREN is set). If the OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR if it is set. The framing error bit FERR(RCSTA<2>) is set if a stop bit is detected as a 0. The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG, will load the RCD8 and the FERR with new values, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RCD8 information.

FIGURE 12-9: BLOCK DIAGRAM FOR SCI RECEIVE

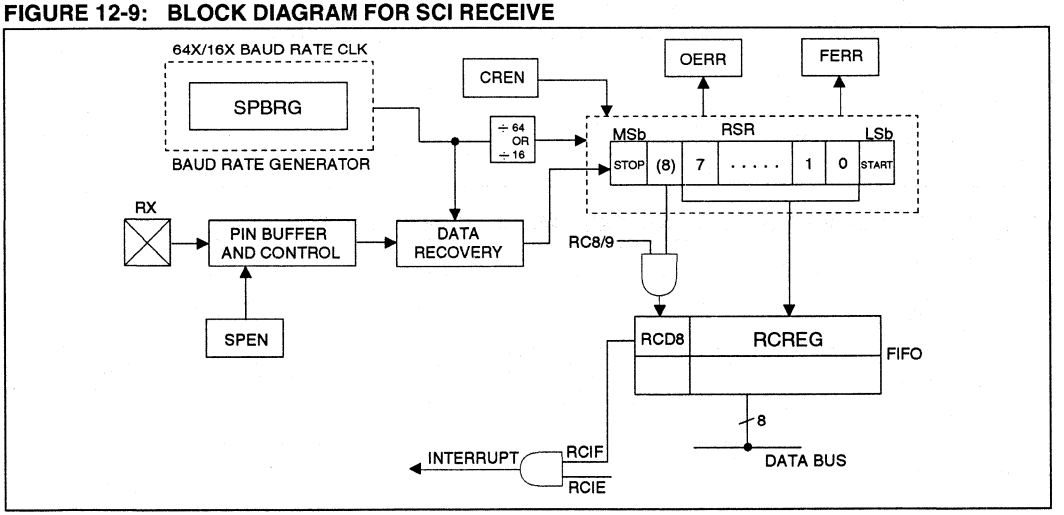
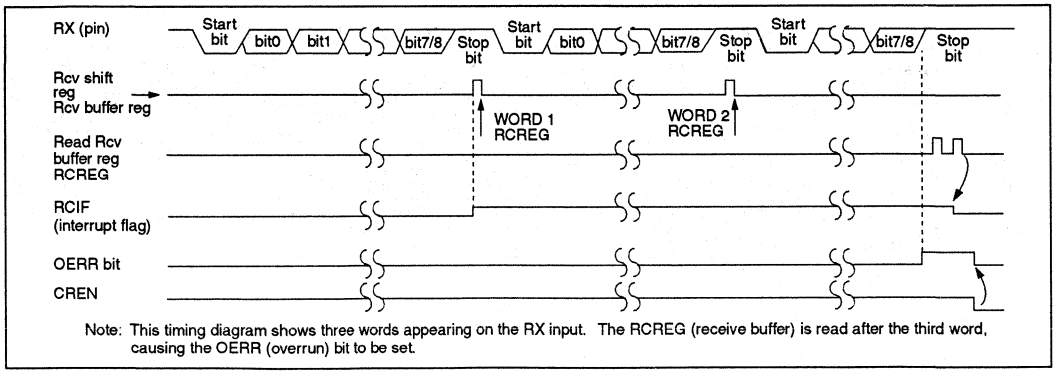


FIGURE 12-10: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

© 1995 Microchip Technology Inc.

Steps to follow when setting up a Asynchronous Reception:

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set BRGH. See Section 12.1 for details.
- Enable the asynchronous serial port by configuring the SYNC = 0 and SPEN = 1.
- If interrupts are desired, then the RCIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- Enable the reception by setting CREN.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit(if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0x18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD6
0x1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
0x8Ch	PIE1	PSPIE	†	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
0x98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
0x99h	SPBRG	Baud Rate Register							

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

12.3 SCI Synchronous Master Mode

In Master Synchronous mode the data is transmitted in a half-duplex manner i.e. transmission and reception do not occur at the same time, when transmitting data, the reception is inhibited and vice versa. The synchronous mode is entered by setting the SYNC (TXSTA<4>) bit. In addition the SPEN (RCSTA<7>) bit is set in order to configure the RC6 and RC7 I/O ports to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC (TXSTA<7>) bit.

12.3.1 SCI SYNCHRONOUS MASTER TRANSMISSION

The SCI transmitter block diagram is shown in Figure 12-6. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer TXREG. The TXREG is loaded with data in software. The TSR is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG transfers the data to the TSR(occurs in one Tcycle), the TXREG is empty and a interrupt bit, TXIF (PIR1<4>) is set. This interrupt can be enabled or disabled by the TXIE bit (PIE1<4>). TXIF will be set regardless of TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG. While TXIF indicates the status of the TXREG, another bit TRMT(TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until the TXREG has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on CK. Data out is stable around the falling edge of the synchronous clock (Figure 12-11). The transmission can also be started by first loading the TXREG and then setting TXEN. This is advantageous when slow baud rates are selected, since the BRG is kept in reset when TXEN=CREN=SREN = 0. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing TXEN during a transmission, will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either CREN or SREN are set, during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if CSRC=1(internal clock). The transmitter logic however is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear TXEN. If the SREN bit is set (to interrupt an on going transmission and receive a single word), then after the single word is received, the SREN will = 0 and the serial port will revert back to transmitting since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, TXEN should be cleared.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to TXD8(TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the "new" TXD8, the "present" value of TXD8 is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

- Initialize the SPBRG register for the appropriate baud rate (see Section 12.1 for details)
- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9-bit should be set.
- Enable the transmission by setting TXEN to 1.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Start transmission by loading data to the TXREG.

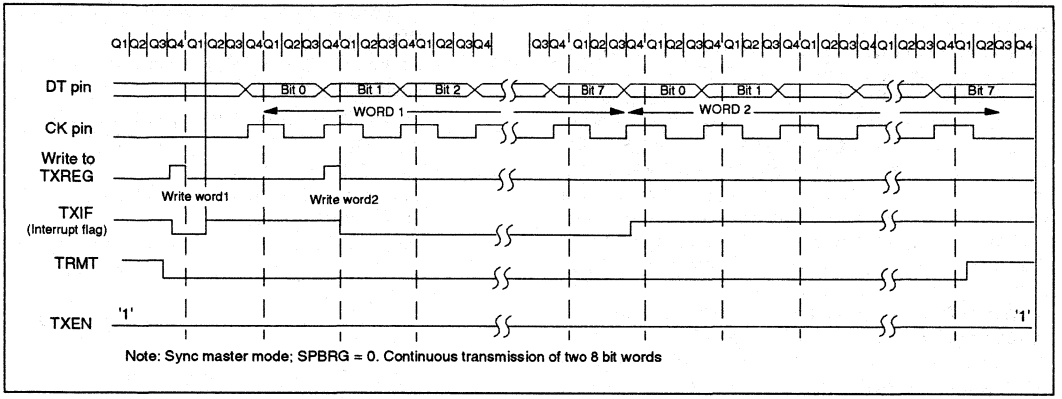
© 1995 Microchip Technology Inc.

TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION:

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	PSPIE	†	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

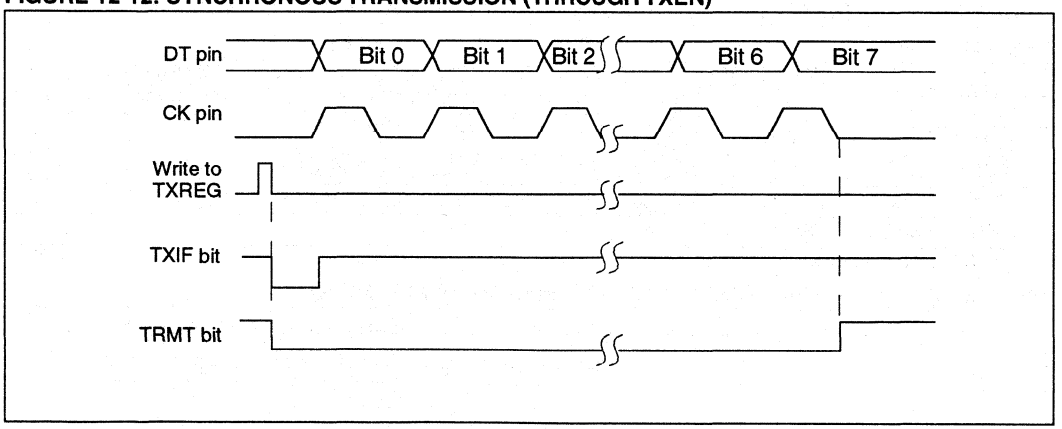
† These bits are reserved on the PIC16C65.

FIGURE 12-11: SYNCHRONOUS TRANSMISSION



Note: Sync master mode; SPBRG = 0. Continuous transmission of two 8 bit words

FIGURE 12-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



12.3.2 SCI SYNCHRONOUS MASTER RECEPTION

Once the synchronous mode is selected, reception is enabled by setting either the SREN(RCSTA<5>) bit or the CREN(RCSTA<4>). Data is sampled on the DT pin on the falling edge of the clock. If SREN is set, then only a single word is received, if CREN is set, the reception is continuous until CREN is reset. If both the bit are set then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register(RSR) is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled or disabled by the RCIE(PIE1<5>) bit. The RCIF is a read only bit which is reset by the hardware. In this case it is reset when the RCREG has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR. On the clocking of the last bit of the third byte, if the RCREG is still full then the overrun error bit, OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software and this is done by clearing CREN. If the OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG, will load the RCD8 with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RCD8 information.

Steps to follow when setting up a Synchronous Master Reception:

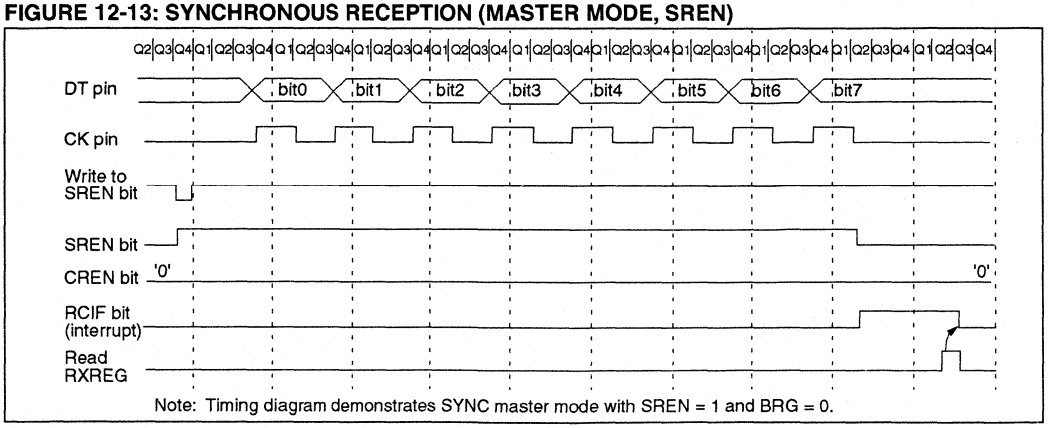
- Initialize the SPBRG register for the appropriate baud rate. See section 12.1 for details.
- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
- CREN= SREN = 0
- If interrupts are desired, then the RXIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- If a single reception is required, set SREN. For continuous reception set CREN.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION:

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD6
19h	PCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
8Ch	PIE1	PSPIE	††	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† These bits are reserved on the PIC16C65.

FIGURE 12-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates SYNC master mode with SREN = 1 and BRG = 0.

© 1995 Microchip Technology Inc.

12.4 SCI Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC(TXSTA<7>) bit.

12.4.1 SCI SYNCHRONOUS SLAVE TRANSMIT

The operation of the sync master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction executed, the following will occur. The first word will immediately transfer to the TSR and transmit. The second word will remain in TXREG. The TXIF will not be set. When the first word has been shifted out of TSR, the TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If the TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
- Make CREN = SREN = 0
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9-bit should be set.
- Enable the transmission by setting TXEN to 1.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Start transmission by loading data to the TXREG.

12.4.2 SCI SYNCHRONOUS SLAVE RECEPTION

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to the RCREG and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
- If interrupts are desired, then the RCIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- To enable reception, set CREN = 1.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit(if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	PSPIE	†	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† These bits are reserved on the PIC16C65.

TABLE 12-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	PSPIF	†	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
8Ch	PIE1	PSPIE	†	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† These bits are reserved on the PIC16C65.

© 1995 Microchip Technology Inc.

13.0 SPECIAL FEATURES OF THE CPU

What sets apart a microcontroller from other processors are special circuits to deal with the needs of real time applications. The PIC16CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

1. OSC selection
2. Reset
   - Power-On Reset (POR)
   - Power-Up Timer (PWRT)
   - Oscillator Start-Up Timer (OST)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID locations
8. In-circuit serial programming

The PIC16CXX has a watchdog timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the oscillator start-up timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the power-up timer (PWRT), which provides a fixed delay of 72ms (nominal) on power up only, designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

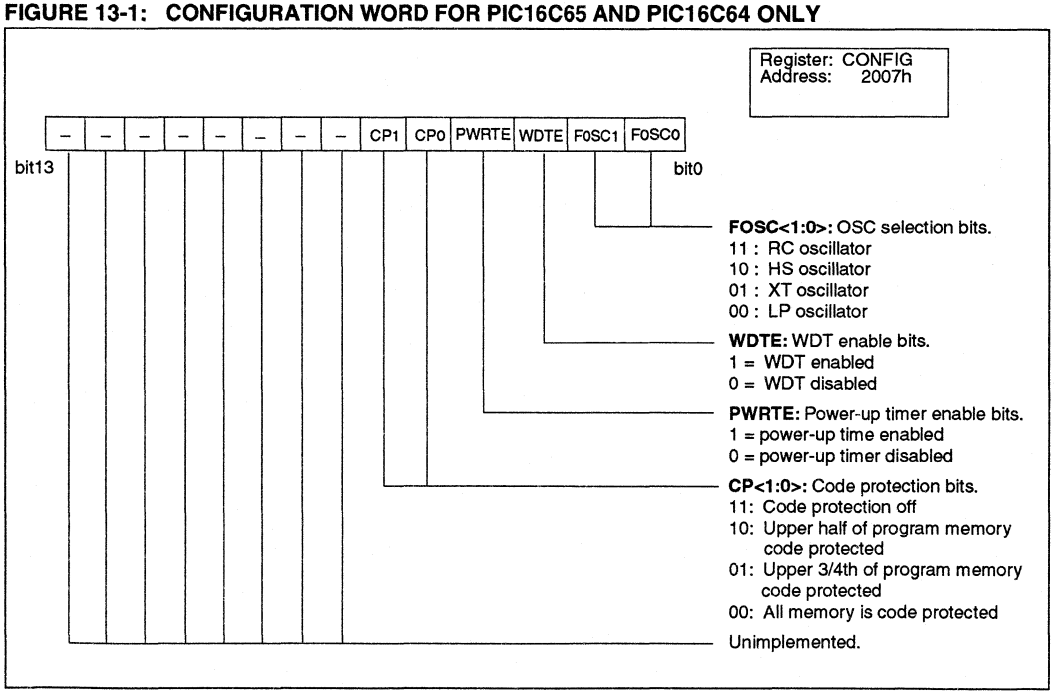
The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset, watchdog timer time-out or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

13.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

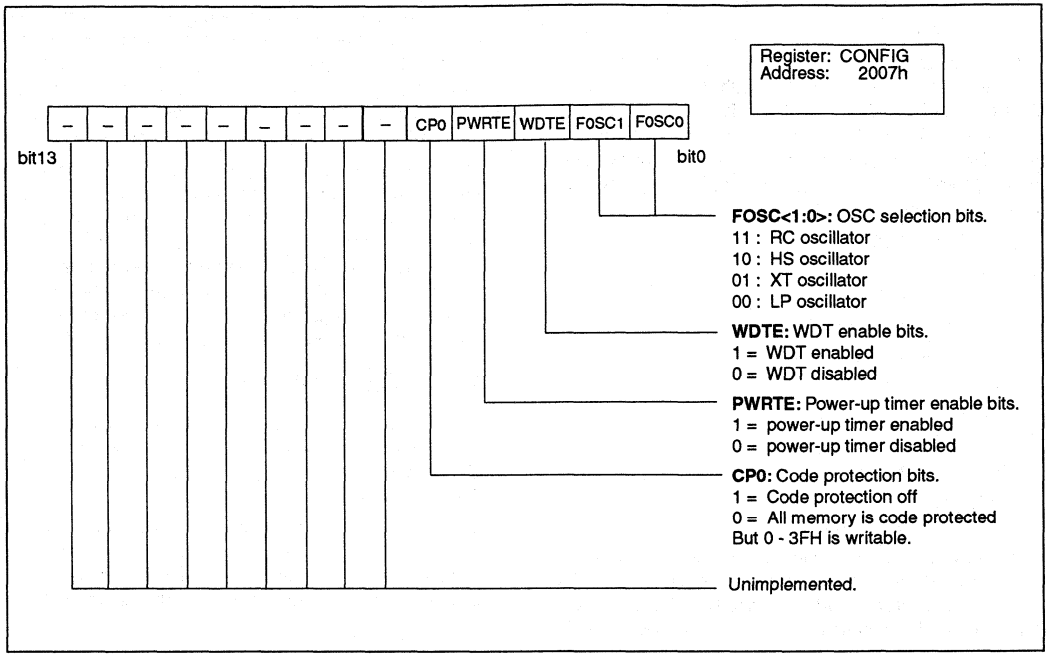
The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

FIGURE 13-1: CONFIGURATION WORD FOR PIC16C65 AND PIC16C64 ONLY



© 1995 Microchip Technology Inc.

FIGURE 13-2: CONFIGURATION WORD FOR PIC16C61 ONLY



© 1995 Microchip Technology Inc.

13.2 Oscillator Configurations

13.2.1 OSCILLATOR TYPES

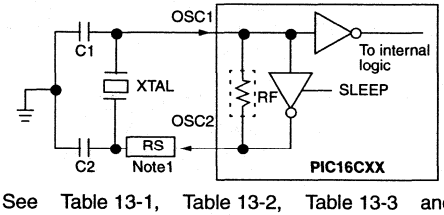
The PIC16CXX can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

13.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 13-3). The PIC16CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin. This is shown in Figure 13-4.

FIGURE 13-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



See Table 13-1, Table 13-2, Table 13-3 and Table 13-4 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 13-4: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

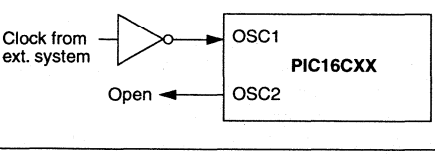


TABLE 13-1: CERAMIC RESONATORS PIC16C65/PIC16C64

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455kHZ	68 - 100 pF	68 - 100 pF
	2.0MHz	15 - 68 pF	15 - 68 pF
	4.0MHz	15 - 68 pF	15 - 68 pF
HS	8.0MHz	10 - 68 pF	10 - 68 pF
	16.0MHz	10 - 22 pF	10 - 22 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.
Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:		
455kHz	Panasonic EFO-A455K04B	+/-.3%
2.0MHz	Murata Erie CSA2.00MG	+/-.5%
4.0MHz	Murata Erie CSA4.00MG	+/-.5%
8.0MHz	Murata Erie CSA8.00MT	+/-.5%
16.0MHz	Murata Erie CSA16.00MX	+/-.5%
All resonators used did not have built-in capacitors.		

TABLE 13-2: CERAMIC RESONATORS PIC16C61

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455kHZ	68 - 100 pF	68 - 100 pF
	2.0MHz	15 - 68 pF	15 - 68 pF
	4.0MHz	15 - 68 pF	15 - 68 pF
HS	8.0MHz	10 - 68 pF	10 - 68 pF
	16.0MHz	10 - 22 pF	10 - 22 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.
Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:		
455kHz	Panasonic EFO-A455K04B	+/-.3%
2.0MHz	Murata Erie CSA2.00MG	+/-.5%
4.0MHz	Murata Erie CSA4.00MG	+/-.5%
8.0MHz	Murata Erie CSA8.00MT	+/-.5%
16.0MHz	Murata Erie CSA16.00MX	+/-.5%
All resonators used did not have built-in capacitors.		

© 1995 Microchip Technology Inc.

TABLE 13-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C65 AND PIC16C64

Mode	Freq	OSC1	OSC2
LP	32kHz§	33 - 68 pF	33 - 68 pF
	200kHz	15 - 47 pF	15 - 47 pF
XT	100kHz	47 - 100 pF	47 - 100 pF
	500kHz	20 - 68 pF	20 - 68 pF
	1MHz	15 - 68 pF	15 - 68 pF
	2MHz	15 - 47 pF	15 - 47 pF
	4MHz	15 - 33 pF	15 - 33 pF
HS	8MHz	15 - 47 pF	15 - 47 pF
	20MHz	15 - 47 pF	15 - 47 pF

Note: Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

§ For VDD > 4.5V, C1 = C2 ≈ 30pf is recommended.

TABLE 13-4: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C61 ONLY

Mode	Freq	OSC1	OSC2
LP	32kHz§	33 - 68 pF	33 - 68 pF
	200kHz	15 - 47 pF	15 - 47 pF
XT	100kHz	47 - 100 pF	47 - 100 pF
	500kKHz	20 - 68 pF	20 - 68 pF
	1MHz	15 - 68 pF	15 - 68 pF
	2MHz	15 - 47 pF	15 - 47 pF
	4MHz	15 - 33 pF	15 - 33 pF
HS	8MHz	15 - 47 pF	15 - 47 pF
	20MHz	15 - 47 pF	15 - 47 pF

Note: Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

§ For VDD > 4.5V, C1 = C2 ≈ 30pf is recommended.

13.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 13-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 13-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

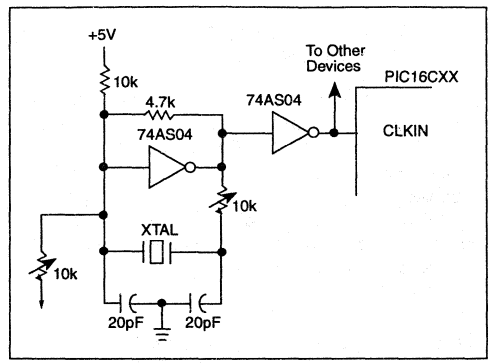
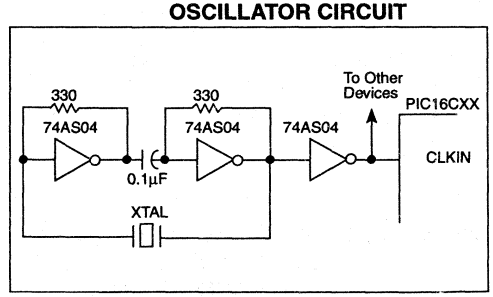


Figure 13-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 13-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



13.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low

© 1995 Microchip Technology Inc.

Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 13-7 shows how the R/C combination is connected to the PIC16CXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

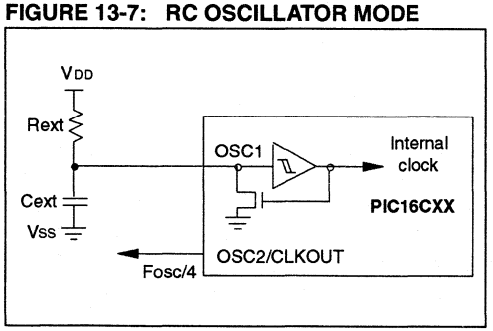
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See characterization data for desired device for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characterization data for desired device for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-4 for waveform).

FIGURE 13-7: RC OSCILLATOR MODE



13.3 Reset

The PIC16CXX differentiates between various kinds of reset:

a) Power-on reset ($\overline{POR}$)
b) $\overline{MCLR}$ reset during normal operation
c) $\overline{MCLR}$ reset during SLEEP
d) WDT time-out reset during normal operation
e) WDT time-out reset during SLEEP

Some registers are not affected in any reset condition; their status is unknown on $\overline{POR}$ and unchanged in any other reset. Most other registers are reset to a "reset state" on power-on reset ($\overline{POR}$), on $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 13-6 and Figure 13-7. These bits are used in software to determine the nature of reset. See Table 13-9 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 13-8.

2

© 1995 Microchip Technology Inc.

13.4 Power-On Reset (POR), Power-Up-Timer (PWRT) and Oscillator Start-up Timer (OST)

13.4.1 POWER-ON RESET (POR)

A Power-On Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.6V - 1.8V). To take advantage of the POR, just tie MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-On Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

The POR circuit does not produce internal reset when VDD declines.

13.4.2 POWER-UP TIMER (PWRT)

The Power-Up Timer provides a fixed 72ms nominal time-out on power-up only, from POR. The power-up timer operates on an internal RC oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, PWRTE can enable (if set) or disable (if cleared or programmed) the power-up timer.

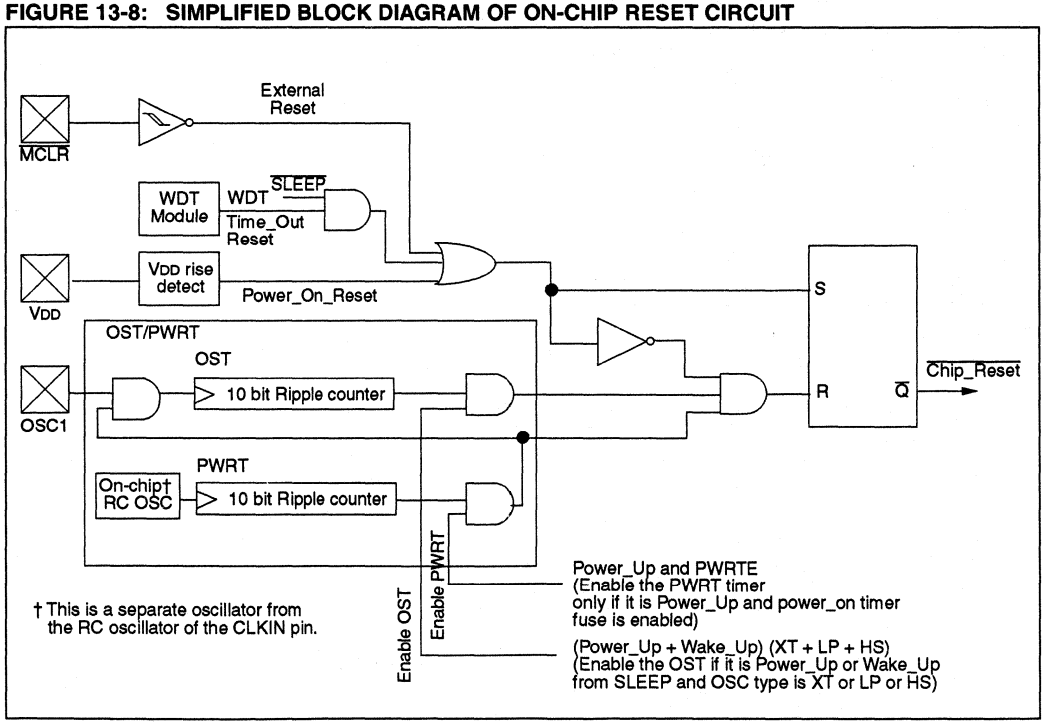
The Power-Up Time delay will vary from chip to chip and due to VDD, temperature, and process variation. See DC parameters for details.

13.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This guarantees that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on power-on reset or wake-up from SLEEP.

FIGURE 13-8: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



© 1995 Microchip Technology Inc.

13.4.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after $\overline{POR}$ has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in RC mode with PWRTE bit cleared (PWRT disabled), there will be no time-out at all. Figure 13-9, Figure 13-10, and Figure 13-11 depict time-out sequences on power up.

TABLE 13-5: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake up from SLEEP
	PWRTE=1	PWRTE=0	
XT, HS, LP	72 ms + 1024TOSC	1024TOSC	1024 TOSC
RC	72 ms	—	—

Since the time-outs occur from the $\overline{POR}$ pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately (see Figure 13-10). This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 13-8 shows the reset conditions for some special registers, while Table 13-6 shows the reset conditions for all the registers.

13.4.5 POWER CONTROL/STATUS REGISTER (PCON)

The power control/status register, PCON (address 8Eh) has only one bit.

Bit1 is $\overline{POR}$ (Power-on-reset). It is cleared on power-on-reset and unaffected otherwise. The user must set this bit following power-on-reset. On a subsequent reset if $\overline{POR}$ is cleared, it will indicate that a Power-On Reset must have occurred VDD may have gone too low).

TABLE 13-6: STATUS BITS AND THEIR SIGNIFICANCE FOR PIC16C65 AND PIC16C64

$\overline{POR}$	$\overline{TO}$	$\overline{PD}$	
0	1	1	Power-on reset
0	0	X	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	X	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	WDT reset during normal operation
1	0	0	WDT timeout wakeup from SLEEP
1	1	1	$\overline{MCLR}$ reset during normal operation
1	1	0	$\overline{MCLR}$ reset during SLEEP or interrupt wake-up from SLEEP

TABLE 13-7: STATUS BITS AND THEIR SIGNIFICANCE FOR PIC16C61

$\overline{TO}$	$\overline{PD}$	
1	1	Power-on reset
0	1	WDT reset during normal operation
0	0	WDT timeout wakeup from SLEEP
u	u	$\overline{MCLR}$ reset during normal operation
1	0	$\overline{MCLR}$ reset during SLEEP or interrupt wake-up from SLEEP

Legend: u = unchanged

TABLE 13-8: RESET CONDITION FOR SPECIAL REGISTERS

	PCL Addr: 02h	STATUS Addr: 03h	PCON on PIC16C65 and PIC16C64 only Addr: 8Eh
Power-On Reset	000h	0001 1xxx	---- --0-
$\overline{MCLR}$ reset during normal operation	000h	0001 1uuu	---- --u-
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu	---- --u-
WDT reset during normal operation	000h	0000 1uuu	---- --u-
WDT during SLEEP	PC + 1	uuu0 0uuu	---- --u-
Interrupt wake-up from SLEEP	PC + 1 (Note1)	uuu1 0uuu	---- --u-

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

© 1995 Microchip Technology Inc.

TABLE 13-9: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Device	Power-on Reset	MCLR Reset during: – normal operation – SLEEP WDT timeout during normal operation	Wake up from SLEEP through interrupt Wake up from SLEEP through WDT timeout
W	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	6X	-	-	-
TMR0	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	6X	0000h	0000h	PC + 1 (2)
STATUS	6X	0001 1xxx	000? ?uuu (3)	uuu? ?uuu (3)
FSR	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	61	---u uuuu	---u uuuu	---u uuuu
	Other 6X	--xx xxxx	--uu uuuu	--uu uuuu
PORTB	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	6X	---- -xxx	---- -uuu	---- -uuu
PCLATH	6X	---0 0000	---0 0000	---u uuuu
INTCON	6X	0000 000x	0000 000u	uuuu uuuu(1)
PIR1	6X	0000 0000	0000 0000	uuuu uuuu(1)
PIR2	65	---- ---0	---- ---0	---- ---u(2)
TMR1L	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	6X	--00 0000	--uu uuuu	--uu uuuu
TMR2	6X	0000 0000	0000 0000	uuuu uuuu
T2CON	6X	-000 0000	-000 0000	-uuu uuuu
SSPBUF	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	6X	0000 0000	0000 0000	uuuu uuuu
CCPR1L	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	6X	--00 0000	--00 0000	--uu uuuu
RCSTA	65	0000 -00x	0000 -00x	uuuu uuuu
TXREG	65	0000 0000	0000 0000	uuuu uuuu
RCREG	65	0000 0000	0000 0000	uuuu uuuu
CCPR2L	65	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	65	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	65	0000 0000	0000 0000	uuuu uuuu
OPTION	6X	1111 1111	1111 1111	uuuu uuuu
TRISA	61	---1 1111	---1 1111	---u uuuu
	Other 6X	--11 1111	--11 1111	--uu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', ? = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 13-8 for reset value for specific condition.

© 1995 Microchip Technology Inc.

TABLE 13-9: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Device	Power-on Reset	$\overline{MCLR}$ Reset during: – normal operation – SLEEP WDT timeout during normal operation	Wake up from SLEEP through interrupt Wake up from SLEEP through WDT timeout
TRISB	6X	1111 1111	1111 1111	uuuu uuuu
TRISC	6X	1111 1111	1111 1111	uuuu uuuu
TRISD	6X	1111 1111	1111 1111	uuuu uuuu
TRISE	6X	0000 -111	0000 -111	uuuu -uuu
PIE1	6X	0000 0000	0000 0000	uuuu uuuu
PIE2	65	---- ---0	---- ---0	---- ---u
PCON	6X	---- --0-	---- --u-	---- --u-
PR2	6X	1111 1111	1111 1111	1111 1111
SSPADD	6X	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	6X	--00 0000	--00 0000	--uu uuuu
TXSTA	65	0000 -010	0000 -010	uuuu -uuu
SPBRG	65	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', ? = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 13-8 for reset value for specific condition.

© 1995 Microchip Technology Inc.

FIGURE 13-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 1

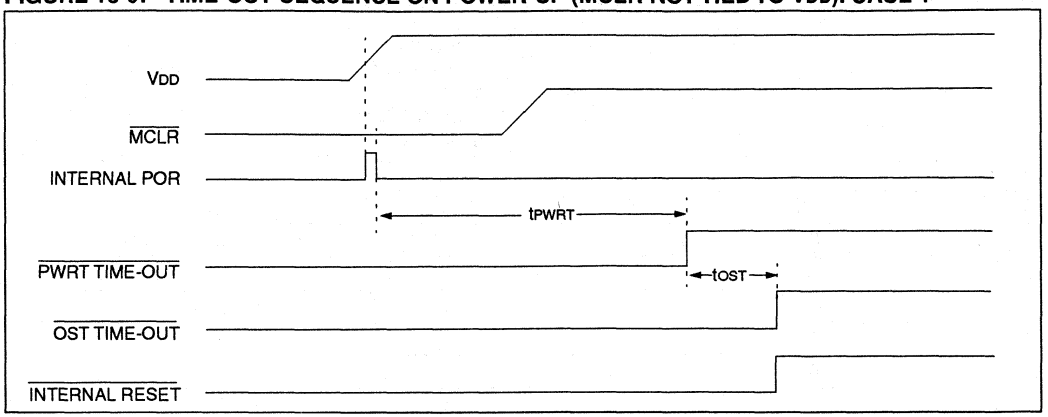


FIGURE 13-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 2

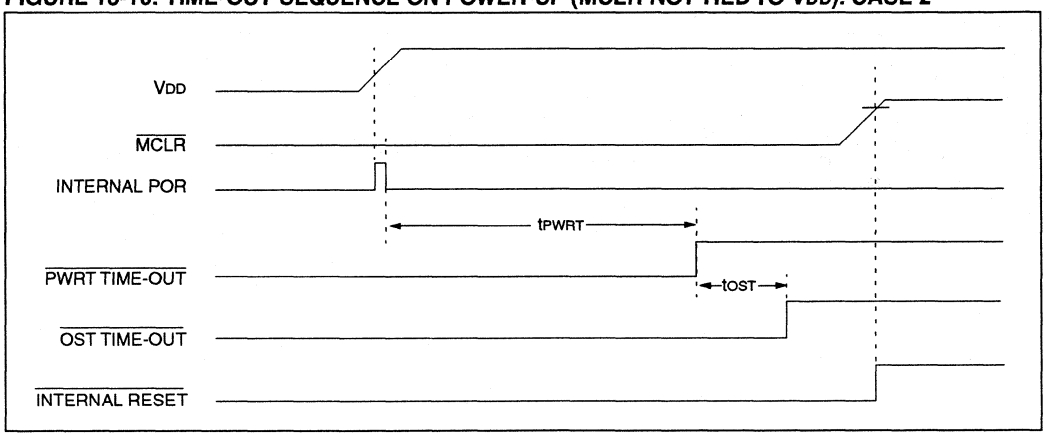
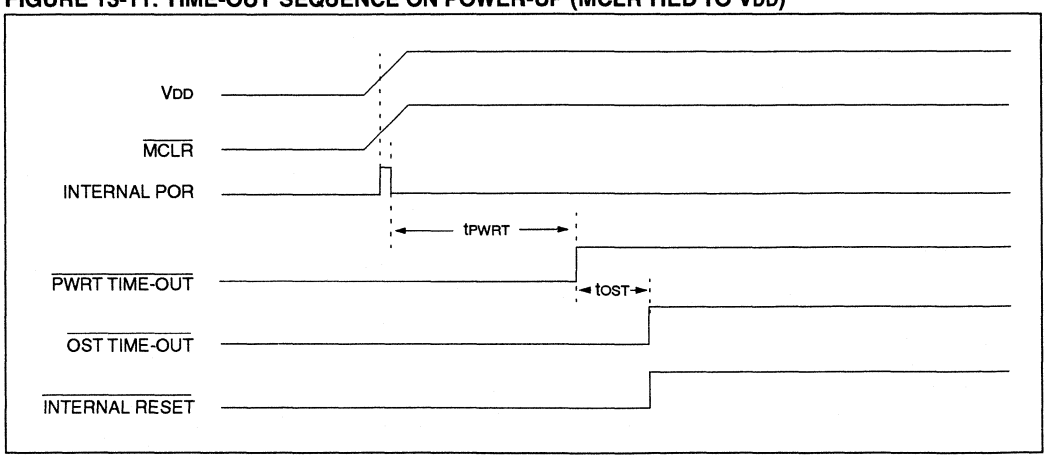
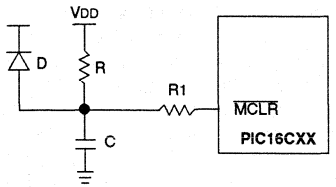


FIGURE 13-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD)



© 1995 Microchip Technology Inc.

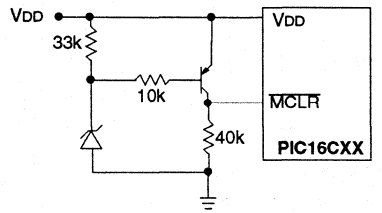
FIGURE 13-12: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note:

1. External power-on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40KΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5μA). A larger voltage drop will degrade VIH level on $\overline{MCLR}$ pin.
3. R1 = 100Ω to 1KΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$ pin breakdown due to ESD or EOS.

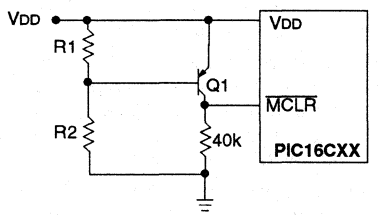
FIGURE 13-13: BROWN-OUT PROTECTION CIRCUIT 1



Note:

1. This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 13-14: BROWN-OUT PROTECTION CIRCUIT 2



Note:

1. This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V.$$

© 1995 Microchip Technology Inc.

13.5 Interrupts

The PIC16C6X family has up to 12 sources of interrupt:

Interrupt sources:	Device:		
	PIC16C65	PIC16C64	PIC16C61
External interrupt RB0/INT	X	X	X
TMR0 overflow interrupt	X	X	X
PORTB change interrupts (pins RB<7:4>)	X	X	X
TMR1 overflow interrupt	X	X	
TMR2 matches period interrupt	X	X	
CCP1 interrupt	X	X	
CCP2 interrupt	X		
SCI asynchronous transmit and receive	X		
Synchronous serial port interrupt	X	X	
Parallel slave port read/write interrupt	X	X	

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The "return from interrupt" instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special register PIR1. The corresponding interrupt enable bits are contained in special register PIE1 and the peripheral interrupt enable bit is contained in special register INTCON.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (see Figure 13-17). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged
2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GIE bit was cleared by the instruction, as shown in the following code:

```
LOOP    BCF INTCON,GIE      ; Disable Global
                            ; Interrupts
        BTFSC INTCON,GIE    ; Global Interrupts
                            ; Disabled?
        GOTO   LOOP;        ; NO, try again
        :                   ; Yes, continue
                            ; with program
                            ; flow
```


© 1995 Microchip Technology Inc.

FIGURE 13-15: INTERRUPT LOGIC FOR PIC16C65 AND PIC16C64

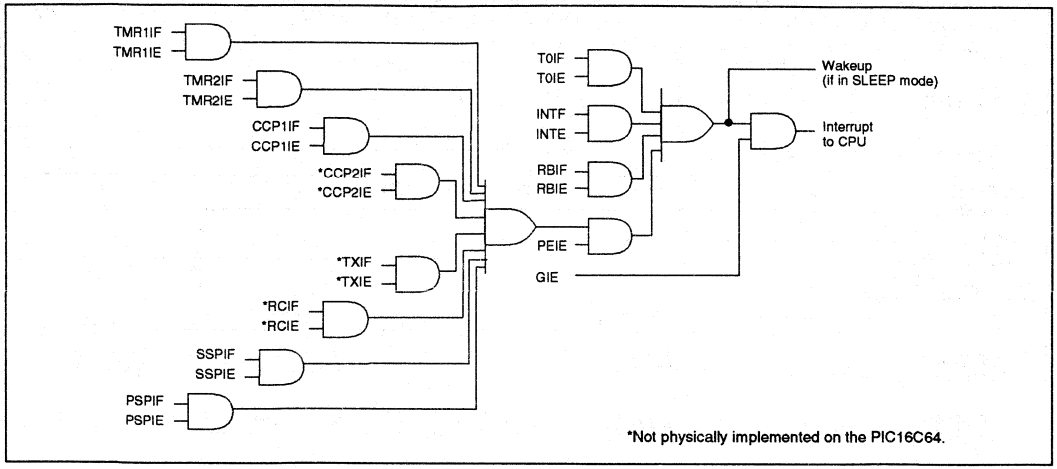


FIGURE 13-16: INTERRUPT LOGIC FOR PIC16C61

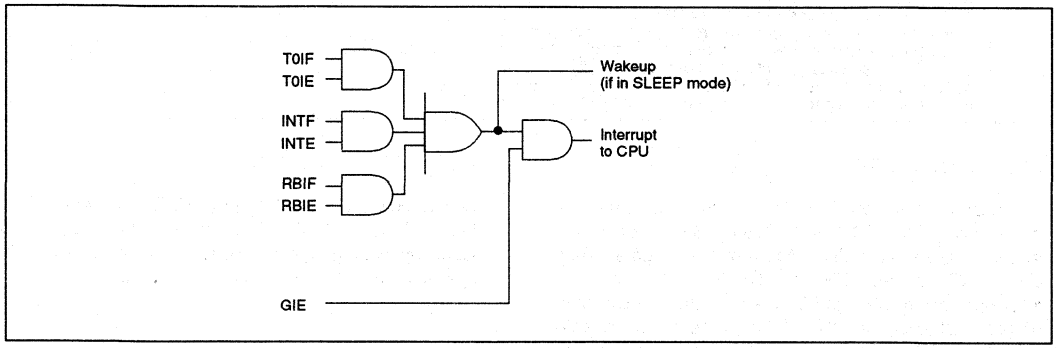
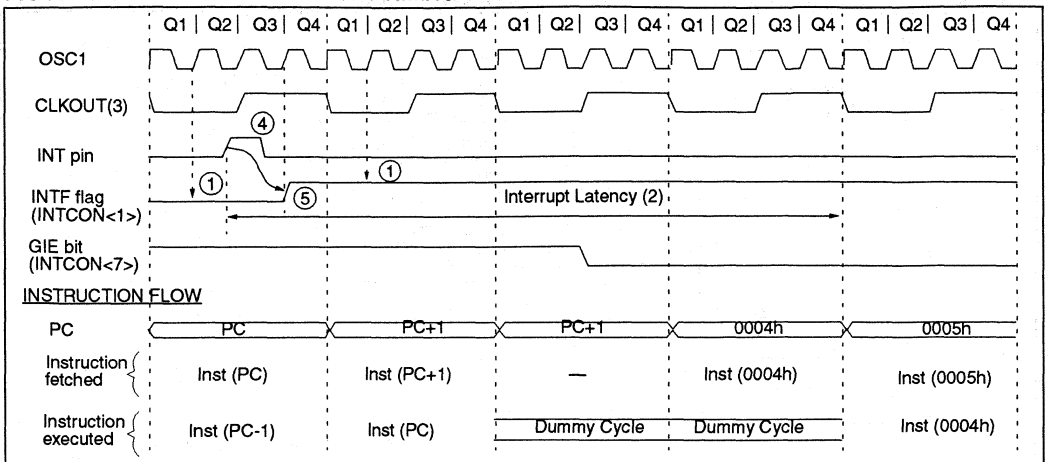


FIGURE 13-17: INT PIN INTERRUPT TIMING



Notes:
1. INTF flag is sampled here (every Q1)
2. Interrupt latency = 3-4Tcy where Tcy = instruction cycle time.
   Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3. CLKOUT is available only in RC oscillator mode.
4. For minimum width spec of INT pulse, refer to AC specs.
5. INTF is enabled to be set anytime during the Q4-Q1 cycles.

13.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the INT pin, the INTF bit (INTCON<1>) is set . This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 13.8 for details on SLEEP mode.

13.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the TMR0 module, see Section 7.0.

© 1995 Microchip Technology Inc.

13.5.3 PORTB INTCON CHANGE

An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB, see Section 5.2.

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

13.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt e.g. W register and STATUS register. This will have to be implemented in software.

Example 13-1 and Example 13-2 store and restore the STATUS and W registers. For PIC16C65 and PIC16C64, the register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1). For PIC16C65 and PIC16C64, the user register, STATUS_TEMP, must be defined in bank 0.

The example:

- Stores W register
- Stores STATUS register in bank 0
- Executes ISR code
- Restores STATUS (and bank select bit) register
- Restores W register

EXAMPLE 13-1: SAVING STATUS AND W REGISTERS IN RAM (PIC16C65 AND PIC16C64)

```
MOVWF   W_TEMP          ;Copy W to TEMP register, could be bank one or zero
SWAPF   STATUS,W        ;Swap status to be saved into W
BCF     STATUS,RP0      ;Change to bank zero, regardless of current bank
MOVWF   STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
:       (ISR)
:
SWAPF   STATUS_TEMP,W   ;Swap STATUS_TEMP register into W (sets bank to
                        ;original state)
MOVWF   STATUS          ;Move W into STATUS register
SWAPF   W_TEMP,F        ;Swap W_TEMP
SWAPF   W_TEMP,W        ;Swap W_TEMP into W
```

EXAMPLE 13-2: SAVING STATUS AND W REGISTERS IN RAM (PIC16C61)

```
MOVWF   W_TEMP          ;Copy W to TEMP register
SWAPF   STATUS,W        ;Swap status to be saved into W
MOVWF   STATUS_TEMP     ;Save status to STATUS_TEMP register
:
:       (ISR)
:
SWAPF   STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
MOVWF   STATUS          ;Move W into STATUS register
SWAPF   W_TEMP,F        ;Swap W_TEMP
SWAPF   W_TEMP,W        ;Swap W_TEMP into W
```

© 1995 Microchip Technology Inc.

13.7 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT timeout causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration fuse WDTE as a '0' (Section 13.1).

13.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

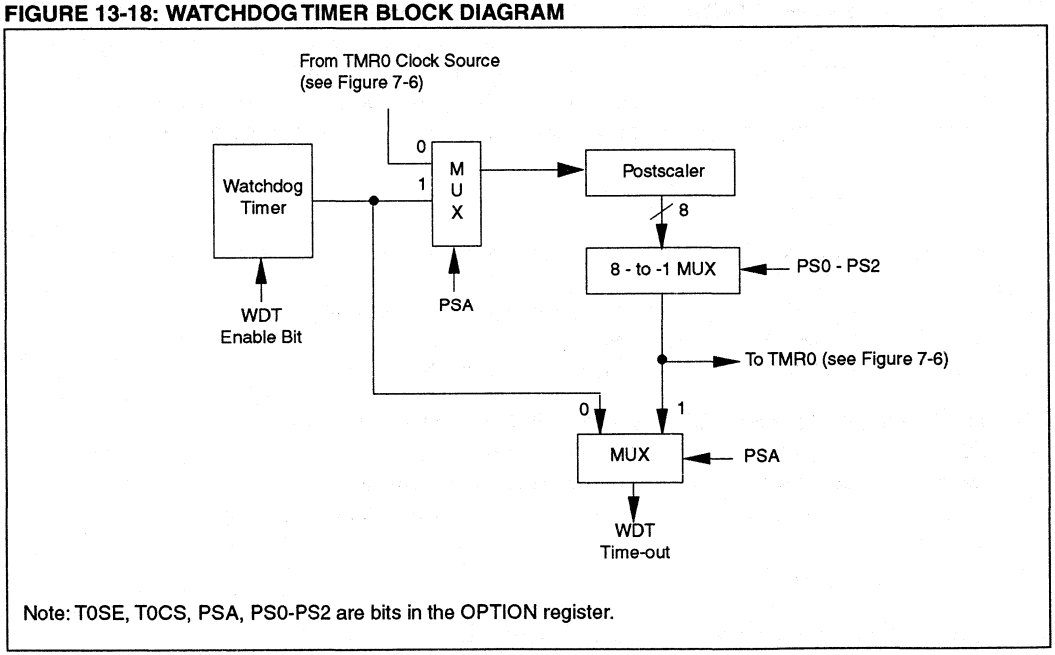
The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a watchdog timer time-out.

13.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

FIGURE 13-18: WATCHDOG TIMER BLOCK DIAGRAM



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

FIGURE 13-19: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	–	–	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

© 1995 Microchip Technology Inc.

13.8 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP command was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin, disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a RESET generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

13.8.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin
2. Watchdog timer time-out reset (if WDT was enabled)
3. Interrupt from INT pin, RB port change, or some Peripheral Interrupts.

The following peripheral interrupts can wake-up from SLEEP:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. SSP (Start/Stop) bit detect interrupt.
3. CCP capture mode interrupt.
4. Parallel Slave port read or write.

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

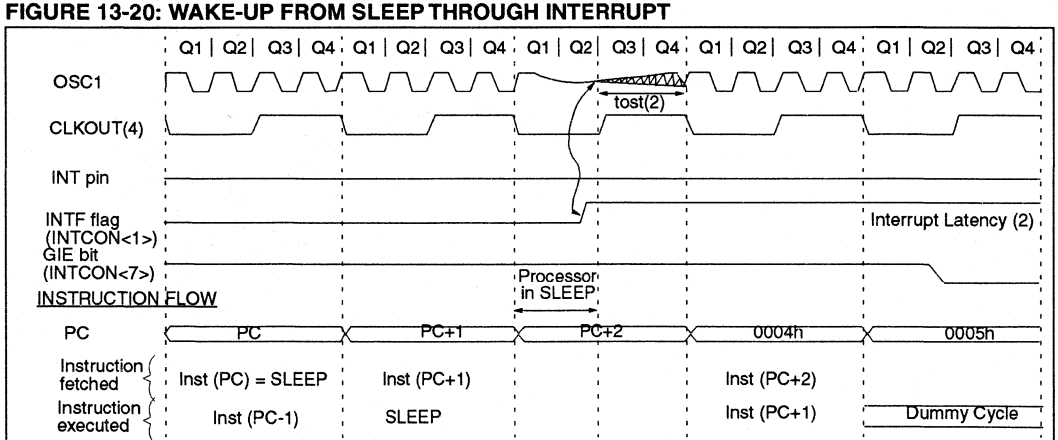
The first event will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. $\overline{PD}$ bit, which is set on power-up is cleared when SLEEP is invoked. $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

Note: If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wakeup from sleep. The sleep instruction is completely executed.

The WDT is cleared when the device wakes-up from sleep, regardless of the source of wake-up.

FIGURE 13-20: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT or LP oscillator mode assumed.
2: tost = 1024tosc (drawing not to scale). This delay will not be there for RC osc mode.
3: When GIE = 0 processor jumps to interrupt routine after wake-up. If GIE = 1, execution will continue in line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

13.9 Code Protection

The code in the program memory can be protected by programming the code protect bits.

The PIC16C65 and PIC16C64 each have two code protect bits (CP<1:0>) and the PIC16C61 has one code protect bit (CP<0>).

The PIC16C65 and PIC16C64 code protection scheme allows the user to selectively protect portions of the program memory. Refer to Figure 13-1 for code protection bit assignments for the PIC16C65 and PIC16C64. Once a segment has been code protected, those memory locations cannot be further programmed. Unprotected segments can be read and reprogrammed.

The PIC16C61 has one code protection bit, CP0 (refer to Figure 13-2). When code protection is enabled, all locations 40h and above cannot be reprogrammed.The first 64 locations, 00h – 3Fh, can be reprogrammed.

The configuration word and ID locations are not code protected for all devices.

13.10 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. Only the 4 least significant bits of ID location are usable.

13.11 In-Circuit Serial Programming

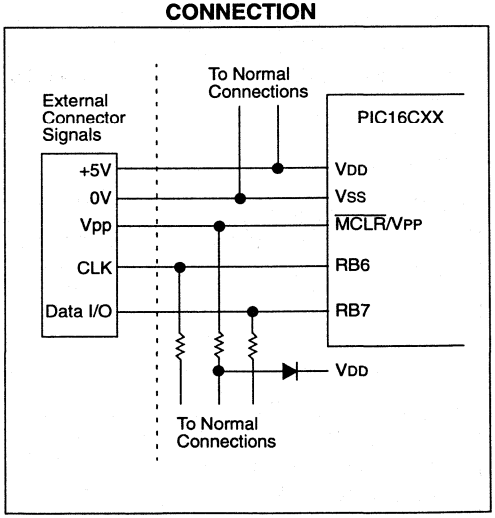
The PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

A typical in-system serial programming connection is shown in Figure 13-21.

FIGURE 13-21: TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION



© 1995 Microchip Technology Inc.

14.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 14-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 14-1 shows the opcode field descriptions.

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be utilized by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 14-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8 bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable Bit
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-out Bit
$\overline{PD}$	Power-down Bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte oriented operations
- Bit oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

Table 14-2 lists the instructions recognized by the MPASM assembler.

Figure 14-1 shows the three general formats that the instructions can have.

Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 14-1: GENERAL FORMAT FOR INSTRUCTIONS

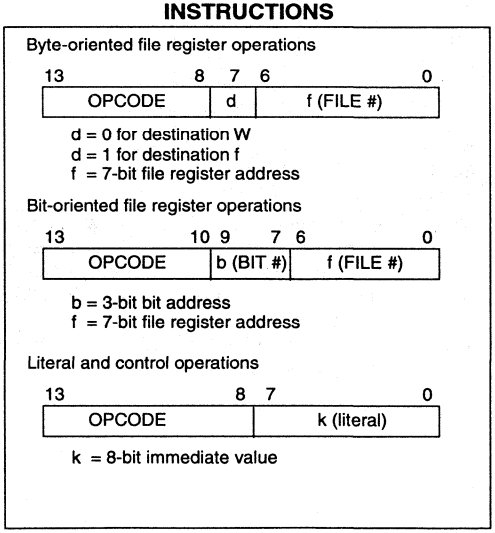


TABLE 14-2: PIC16CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode msb			lsb	Status Affected	Notes
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W and f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W and f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate left through carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W and f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal to W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal to W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear watchdog timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal to W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Excl. OR literal to W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g. MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

Note 2: If this instruction is executed on the TMR0 register (and, where applicable, d=1), the prescaler will be cleared if assigned to the TMR0.

Note 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

© 1995 Microchip Technology Inc.

14.1 Instruction Descriptions

ADDLW **Add Literal to W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → W

Status Affected: C, DC, Z

Encoding:

11	111x	kkkk	kkkk

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

ADDWF **ADD W to f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0111	dfff	ffff

Description: Add the contents of the W register to register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDLW **And Literal and W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → W

Status Affected: Z

Encoding:

11	1001	kkkk	kkkk

Description: The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

00	0101	dfff	ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

© 1995 Microchip Technology Inc.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 0 → f <b>

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example

```
BCF     FLAG_REG,  7
```

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: 1 → f<b>

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example

```
BSF     FLAG_REG,  7
```

Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **BIT Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b ≤ 7

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is skipped.
If bit 'b' is '0' then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address TRUE
if FLAG<1>=1,
PC=address FALSE

© 1995 Microchip Technology Inc.

BTFSS **Bit Test, skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address FALSE
if FLAG<1>=1,
PC=address TRUE

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL   THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → f
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF    FLAG_REG
```

Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W Register**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0XXX	XXXX

Description: W registered is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

© 1995 Microchip Technology Inc.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description: CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

00	1001	dfff	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f)-1 → (dest)

Status Affected: Z

Encoding:

00	0011	dfff	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → d; skip if result = 0

Status Affected: None

Encoding:

00	1011	dfff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE     DECFSZ   CNT, 1
         GOTO     LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

© 1995 Microchip Technology Inc.

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: 0 ≤ k ≤ 2047

Operation: k → PC<10:0>
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example
```
GOTO THERE
```
After Instruction
PC = Address THERE

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
INCF CNT, 1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is decremented. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    INCFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE+1

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
IORLW 0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF

© 1995 Microchip Technology Inc.

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (W)

Status Affected: $\overline{Z}$

Encoding:

00	0100	dfff	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example

```
IORWF          RESULT, 0
```

Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

Description: The contents of register f is moved to destination d. If d=0, destination is W register. If d=1, the destination is file register f itself. d=1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example

```
MOVF     FSR, 0
```

After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

11	00XX	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example

```
MOVLW    0x5A
```

After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

00	0000	1fff	ffff

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example

```
MOVWF       OPTION
```

Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

© 1995 Microchip Technology Inc.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	00 | 0000 | 0XX0 | 0000
Description:	No operation.
Words:	1
Cycles:	1
Example	NOP

OPTION	Load Option Register
Syntax:	[*label*] OPTION
Operands:	None
Operation:	W → OPTION;
Status Affected:	None
Encoding:	00 | 0000 | 0110 | 0010
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.
Words:	1
Cycles:	1
Example	**To maintain upward compatibility with future PIC16CXX products, do not use this instruction.**

RETFIE	Return from Interrupt
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE;
Status Affected:	None
Encoding:	00 | 0000 | 0000 | 1001
Description:	Return from Interrupt. Stack is popped and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting the Global Interrupt Enable (GIE) bit. GIE is the global interrupt enable bit (INTCON<7>). This is a two cycle instruction.
Words:	1
Cycles:	2
Example	RETFIE After Interrupt PC = TOS GIE = 1

RETLW	Return Literal to W
Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W; TOS → PC;
Status Affected:	None
Encoding:	11 | 01XX | kkkk | kkkk
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.
Words:	1
Cycles:	2
Example	CALL TABLE ;W contains table ;offset value • ;W now has table value • • TABLE ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • • RETLW kn ; End of table Before Instruction W = 0x07 After Instruction W = value of k7

2

© 1995 Microchip Technology Inc.

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

00	0000	0000	1000

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETURN
```

After Interrupt
PC = TOS

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

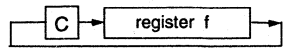
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example

```
RRF             REG1,0
```

Before Instruction
```
REG1    =   11100110
C       =   0
```
After Instruction
```
REG1    =   11100110
W       =   01110011
C       =   1
```

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

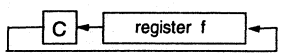
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example

```
RLF             REG1,0
```

Before Instruction
```
REG1    =   11100110
C       =   0
```
After Instruction
```
REG1    =   11100110
W       =   11001100
C       =   1
```

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power down status bit ($\overline{PD}$) is cleared. Time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Section 13.8 for more details.

Words: 1

Cycles: 1

Example:

```
SLEEP
```


© 1995 Microchip Technology Inc.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1:
```
SUBLW   0x02
```
Before Instruction
W = 1
C = ?
After Instruction
W = 1
C = 1; result is positive

Example 2: Before Instruction
W = 2
C = ?
After Instruction
W = 0
C = 1; result is zero

Example 3: Before Instruction
W = 3
C = ?
After Instruction
W = FF
C = 0; result is negative

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: f - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement methodize W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1,1
```
Before Instruction
REG1 = 3
W = 2
C = ?
After Instruction
REG1 = 1
W = 2
C = 1; result is positive

Example 2: Before Instruction
REG1 = 2
W = 2
C = ?
After Instruction
REG1 = 0
W = 2
C = 1; result is zero

Example 3: Before Instruction
REG1 = 1
W = 2
C = ?
After Instruction
REG1 = FF
W = 2
C = 0; result is negative

2

© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label* SWAPF f,d]

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: f<0:3> → d<4:7>,
f<4:7> → d<0:3>

Status Affected: None

Encoding:

00	1110	dfff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example

```
SWAP  REG,  0
F
```

Before Instruction
REG1 = 0xA5

After Instruction
REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: 5 ≤ f ≤ 7

Operation: W → TRIS register f;

Status Affected: None

Encoding:

00	0000	0110	0fff

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

11	1010	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example

```
XORLW  0xAF
```

Before Instruction
W = 0xB5

After Instruction
W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

00	0110	dfff	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example

```
XORWF  REG  1
```

Before Instruction
REG = 0xAF
W = 0xB5

After Instruction
REG = 0x1A
W = 0xB5

© 1995 Microchip Technology Inc.

15.0 DEVELOPMENT SUPPORT

15.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

15.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator provides the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 15-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 and better machines. The development software runs in the Microsoft Windows™ 3.x environment, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Emulator Universal System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host emulation control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x1, two or more PICMASTER emulators can be run simultaneously on the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes currently meet the specifications shown in Table 15-1.

FIGURE 15-1: PICMASTER SYSTEM CONFIGURATION

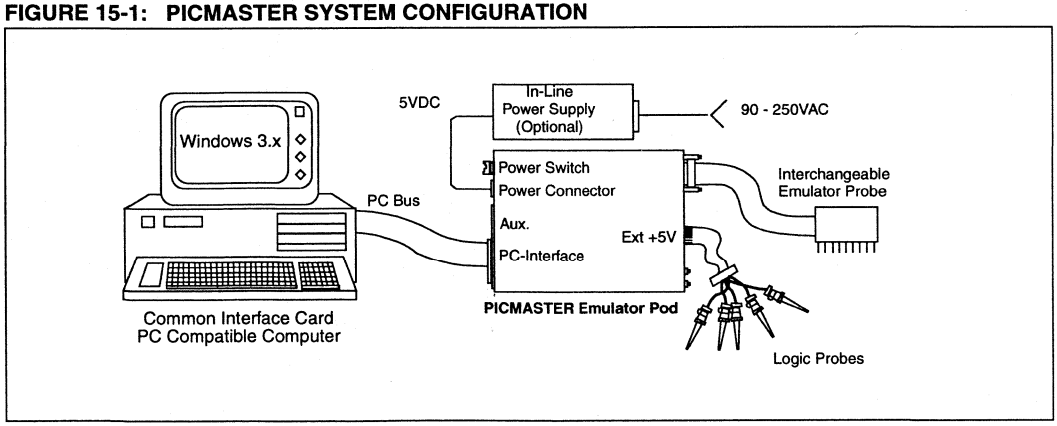


TABLE 15-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

15.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module." Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

15.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

15.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

15.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to and LCD module and a keypad.

© 1995 Microchip Technology Inc.

15.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro Assembly Capability
- Provides Object files, Listing files, Symbol files and special files required for debugging with one of the Microchip Emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** control the allocation of memory and provide a way to refer to data items symbolically, by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and sub-titles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

15.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in single step, execute until break or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

15.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

15.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool comes in two versions: a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design, and a full-featured *fuzzy*TECH-MP Edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

15.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 15-2.

TABLE 15-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator with PRO MATE Programmer, Assembler, Software Simulator, Samples, and your choice of Target Probe,
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

16.0 ELECTRICAL CHARACTERISTICS FOR PIC16C65

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to+ 125°C
Storage Temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{MCLR}$) ..... -0.6V to VDD +0.6V
Voltage on VDD with respect to VSS ..... 0 to +7.5 V
Voltage on $\overline{MCLR}$ with respect to VSS (Note 2) ..... 0 to +14 V
Total power Dissipation (Note 1) ..... 1.0 W
Maximum Current out of VSS pin 300mA
Maximum Current into VDD pin 250mA
Input clamp current, IIK (VI<0 or VI> VDD) ..... ±20mA
Output clamp current, IOK (V0 <0 or V0>VDD) ..... ±20mA
Maximum Output Current sunk by any I/O pin 25mA
Maximum Output Current sourced by any I/O pin 25mA
Maximum Current sunk by PORTA, PORTB, and PORTE (combined) 200mA
Maximum Current sourced by PORTA, PORTB, and PORTE (combined) 200mA
Maximum Current sunk by PORTC and PORTD (combined) 200mA
Maximum Current sourced by PORTC and PORTD (combined) 200mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - Σ IOH} + Σ {(VDD-VOH) x IOH} + Σ(Vol x IOL)

Note 2: Voltage spikes below VSS at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 16-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C65-04	PIC16C65-10	PIC16C65-20	PIC16LC65-04
RC	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21µA Max. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT disabled Freq: 4MHz Max.	VPD: 3.0V to 6.0V IPD: 2.0mA typ. at 3V IPD: 0.9µA typ. at 3V WDT dis Freq: 4MHz typ.
XT	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21µA Max. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT disabled Freq: 4MHz Max.	VPD: 4.5V to 5.5V IPD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT disabled Freq: 4MHz Max.	VPD: 3.0V to 6.0V IPD: 2.0mA typ. at 3V IPD: 0.9µA typ. at 3V WDT dis Freq: 4MHz typ.
HS	VPD: 4.5V to 5.5V IPD: 13.5mA typ. at 5.5V IPD: 1.5µA typ. at 4.5V WDT disabled Freq: 4MHz	VPD: 4.5V to 5.5V IPD: 20mA Max. at 5.5V IPD 1.0µA typ. at 4.5V WDT disabled Freq: 20MHz Max.	VPD: 4.5V to 5.5V IPD: 30mA Max. at 5.5V IPD: 1.5µA typ. at 4.5V WDT disabled Freq: 20MHz Max.	Do not use in HS mode
LP	VPD: 4.0V to 6.0V IPD: 22.5mA typ. at 32 kHz, 4.0V IPD: 0.9µA typ. at 4.0V WDT disabled Freq: 200 kHz typ.	Do not use in LP mode	Do not use in LP mode	VPD: 3.0V to 6.0V IPD: 48µA Max. at 32 KHz, 3.0V IPD: 13.5µA Max. at 3.0V WDT disabled Freq: 200 kHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

16.1 DC CHARACTERISTICS: PIC16C65-04 (Commercial, Industrial, Automotive) PIC16C65-10 (Commercial, Industrial, Automotive) PIC16C65-20 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- - -	2.7 52.5 13.5	5 105 30	mA µA mA	XT, RC, osc configuration (PIC16C65-04) FOSC = 4 MHz, VDD = 5.5V (Note 4) LP osc configuration (PIC16C65-04) FOSC = 32 KHz, VDD = 4.0V, WDT disabled HS osc configuration (PIC16C65-20) FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3, 5)	IPD	- - - - - -	10.5 1.5 1.5 1.5 15* 19*	42 21 24 TBD 32* 40*	µA µA µA µA µA µA	VDD=4.0V, WDT enabled,-40°C to +85°C VDD=4.0V, WDT disabled,-0°C to +70°C VDD=4.0V, WDT disabled,-40°C to +85°C VDD=4.0V, WDT disabled,-40°C to +125°C LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Commercial LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and V_{SS}.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

 © 1995 Microchip Technology Inc.

16.2 DC CHARACTERISTICS: PIC16LC65-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 3.0	- -	6.0 6.0	V V	LP osc configuration XT, RC osc configuration (DC - 4MHz)
RAM Data Retention Voltage (Note 1)	VDR	1.5*	-	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- -	2.0 22.5	3.8 48	mA µA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4) LP OSC CONFIGURATION FOSC = 32 KHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3, 5)	IPD	- - - -	7.5 0.9 0.9 0.9	30 13.5 18 24	µA µA µA µA	VDD=3.0V, WDT enabled,-40°C to +85°C VDD=3.0V, WDT disabled,0°C to +70°C VDD=3.0V, WDT disabled,-40°C to +85°C VDD=3.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

16.3 DC CHARACTERISTICS: PIC16C65-04 (Commercial, Industrial, Automotive) PIC16C65-10 (Commercial, Industrial, Automotive) PIC16C65-20 (Commercial, Industrial, Automotive) PIC16LC65-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec Table 16-1 and Table 16-2

Characteristic	Sym	Min	Typ †	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	-	0.8V	V	
with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
MCLR, RA4/T0CKI,OSC1 (in RC mode)		Vss	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
Input High Voltage						
I/O ports	VIH		-			
with TTL buffer		2.0	-	VDD	V	4.5V ≤ VDD ≤ 5.5V
with Schmitt Trigger buffer		0.85VDD	-	VDD		For entire VDD range
MCLR RA4/T0CKI, RC<7:4>, RD<7.4>, INT		0.85VDD	-	VDD	V	
RE<2:0>, OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	200	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)						
I/O ports	IIL	-	-	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RA4/T0CKI		-	-	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	-	-	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD-4.5V, -40°C to +125°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD-4.5V, -40°C to +125°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +125°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	
SCL, SDA in I^2C mode	Cb			400	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C65 be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

 © 1995 Microchip Technology Inc.

16.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedence)	V	Valid
L	ow	Z	High Impedence
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

16.5 Timing Diagrams and Specifications

FIGURE 16-1: EXTERNAL CLOCK TIMING

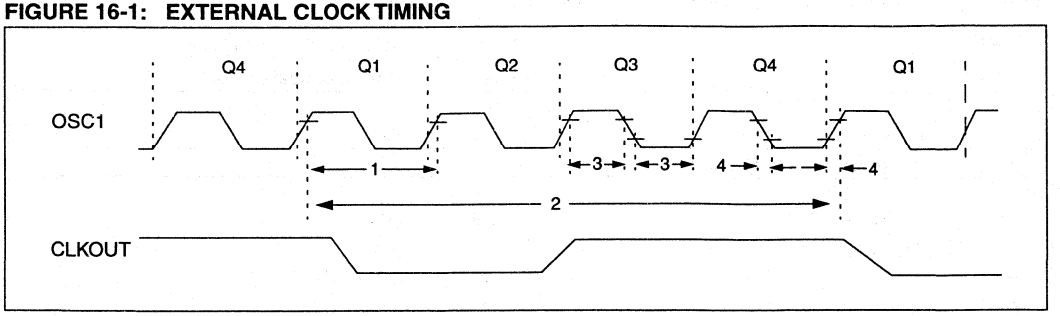


TABLE 16-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C65-04, PIC16LC65-04)
			DC	—	20	MHz	HS osc mode (PIC16C65-20)
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C65-04, PIC16LC65-04)
			4	—	10	MHz	HS osc mode (PIC16C65-10)
			4	—	20	MHz	HS osc mode (PIC16C65-20)
			5	—	200	KHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C65-04, PIC16LC65-04)
			100	—	—	ns	HS osc mode (PIC16C65-10)
			50	—	—	ns	HS osc mode (PIC16C65-20)
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C65-04, PIC16LC65-04)
			100	—	250	ns	HS osc mode (PIC16C65-10)
			50	—	1,000	ns	HS osc mode (PIC16C65-20)
			5	—	—	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	200	—	DC	ns	Tcy = 4/Fosc
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

 © 1995 Microchip Technology Inc.

FIGURE 16-2: LOAD CONDITIONS

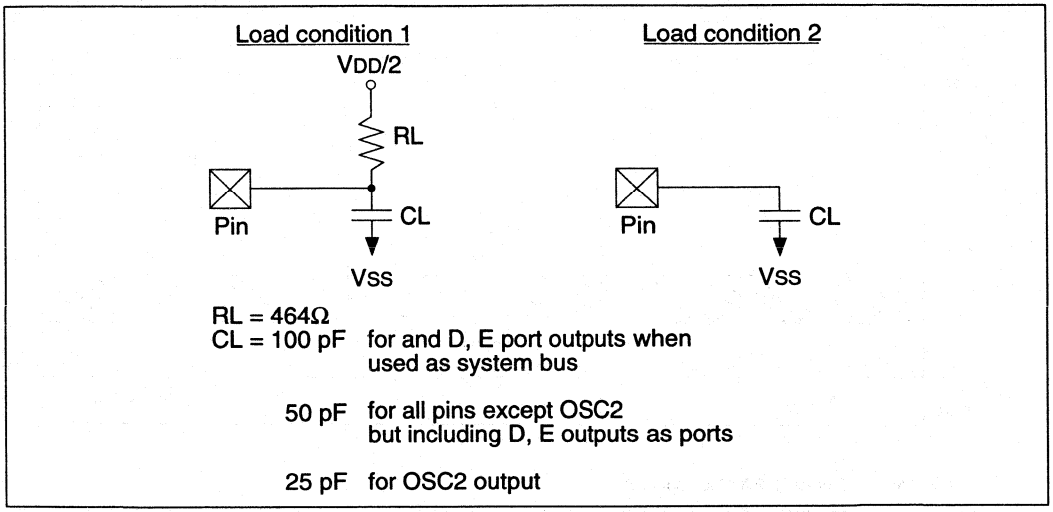
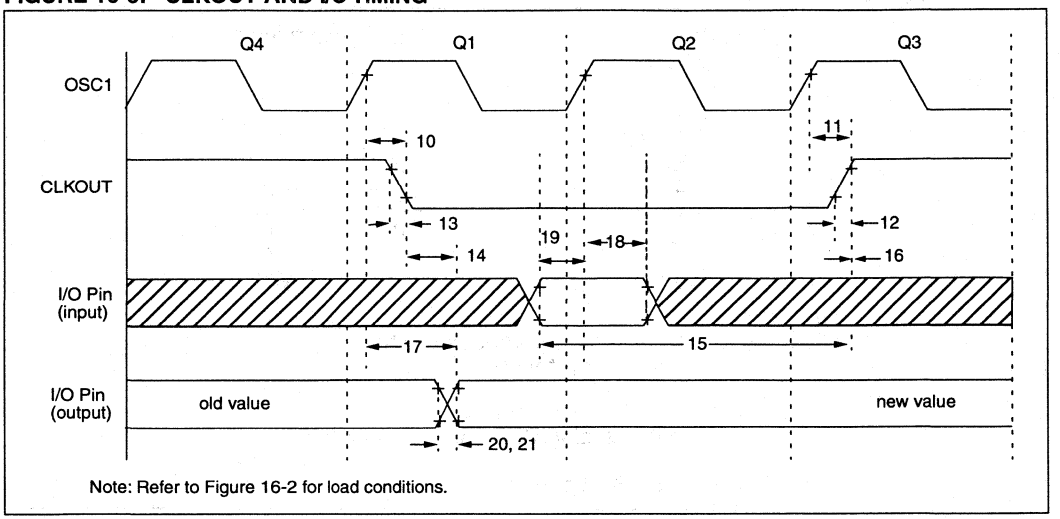


FIGURE 16-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 16-2 for load conditions.

TABLE 16-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	INT pin high or low time	20	—	—	ns	
23††	Trbp	RB<7:4> change INT high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

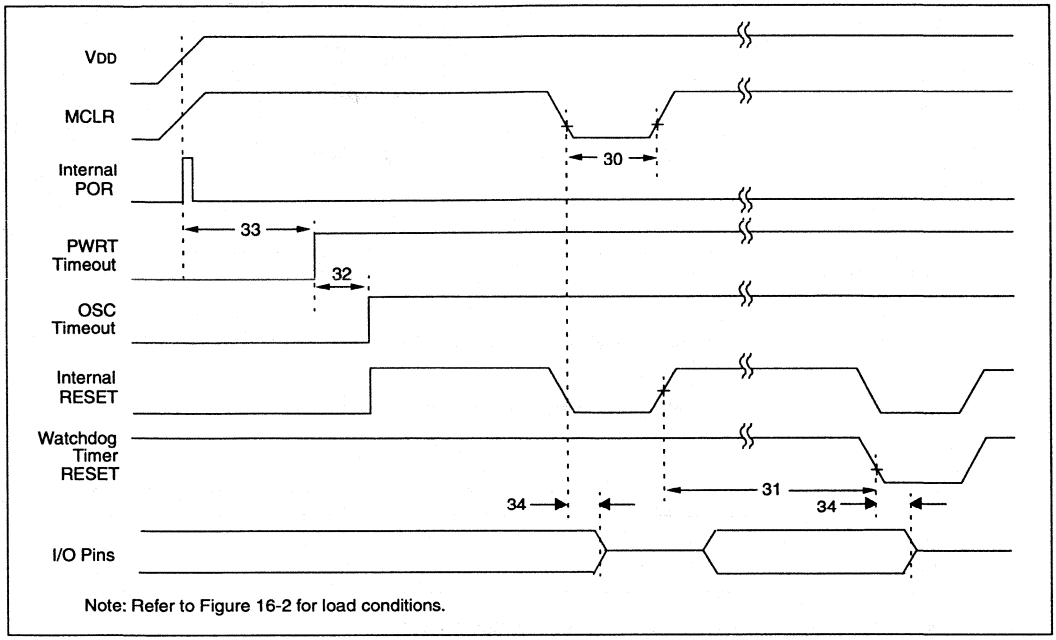
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edge.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC

 © 1995 Microchip Technology Inc.

FIGURE 16-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 16-2 for load conditions.

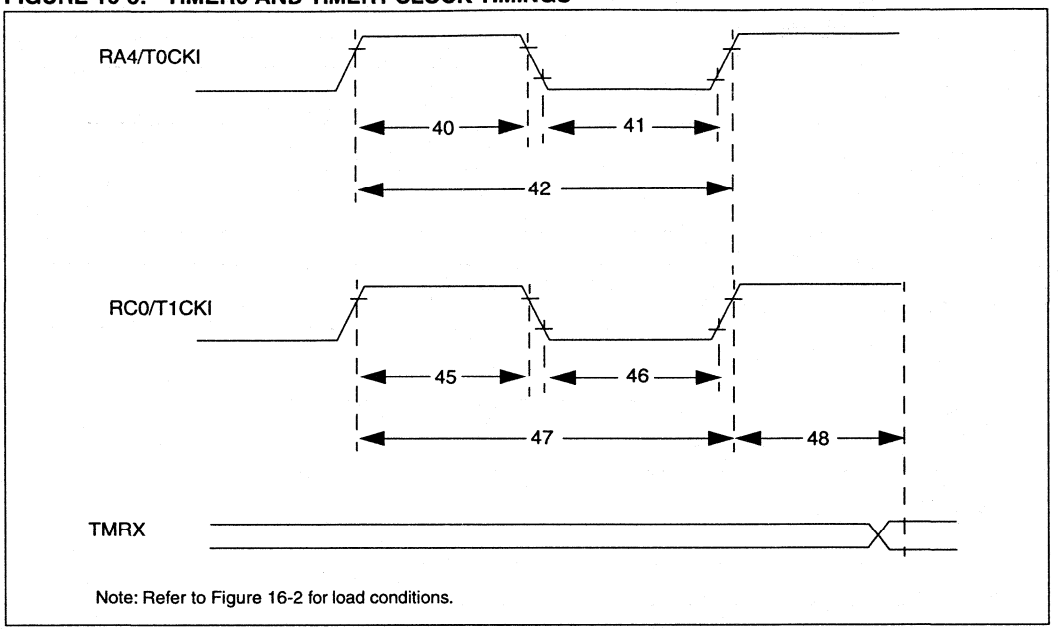
TABLE 16-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	Tioz	I/O High Impedance from $\overline{MCLR}$ Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 16-5: TIMER0 AND TIMER1 CLOCK TIMINGS



Note: Refer to Figure 16-2 for load conditions.

TABLE 16-5: TIMER0 AND TIMER1 CLOCK REQUIREMENTS

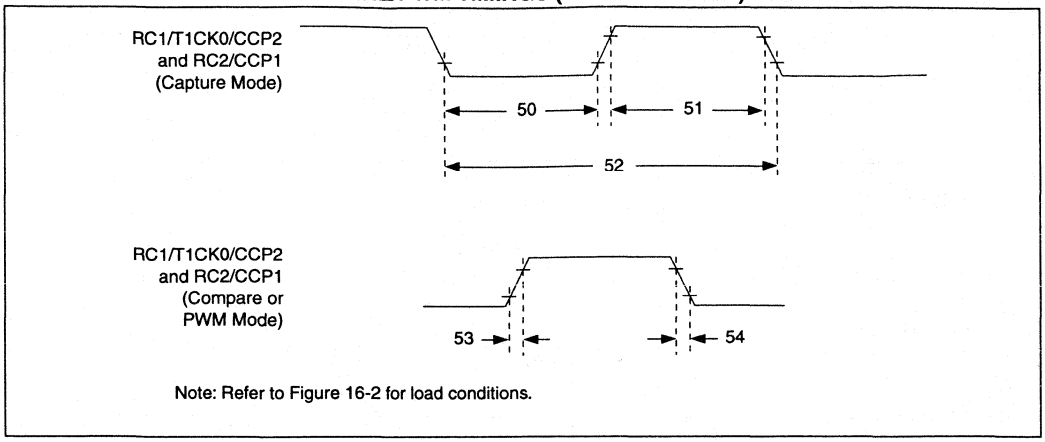
Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, ..., 256)
45	Tt1H	T1CKI High Time	Synchronous, No Prescaler	0.5TCY + 20	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
46	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	0.5TCY + 20*	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
47	Tt1P	T1CKI input period	Synchronous	TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	4 TCY	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting the T1OSCEN bit)		DC	—	200	KHz	
48	TCKEZtmr1	Delay from external clock edge to timer increment		2 TOSC	—	7TOSC	—	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 16-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)



Note: Refer to Figure 16-2 for load conditions.

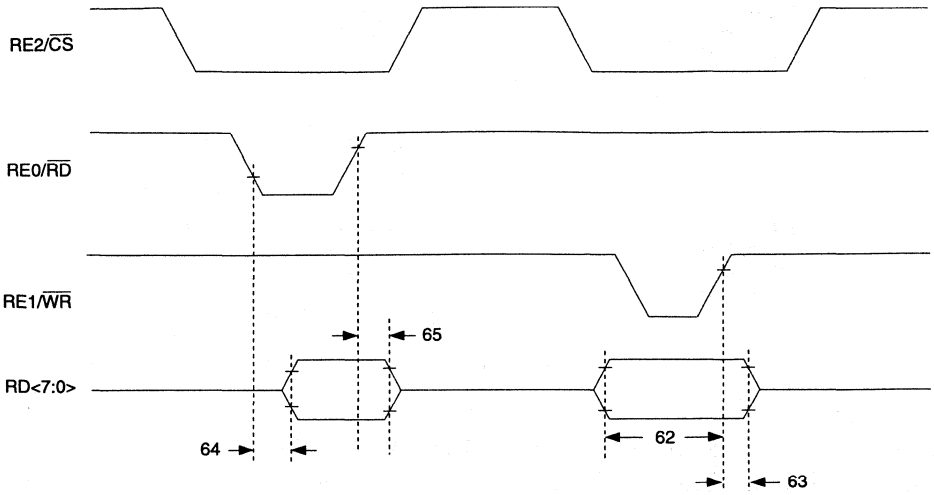
TABLE 16-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
50	TccL	CCP1 and CCP2 input low time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
51	TccH	CCP1 and CCP2 input high time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
52	TccP	CCP1 and CCP2 input period		$\frac{3T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (1,4, or 16)
53	TccR	CCP1 and CCP2 output rise time		—	10	25	ns	
54	TccF	CCP1 and CCP2 output fall time		—	10	25	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 16-7: PARALLEL SLAVE PORT TIMING



Note: Refer to Figure 16-2 for load conditions

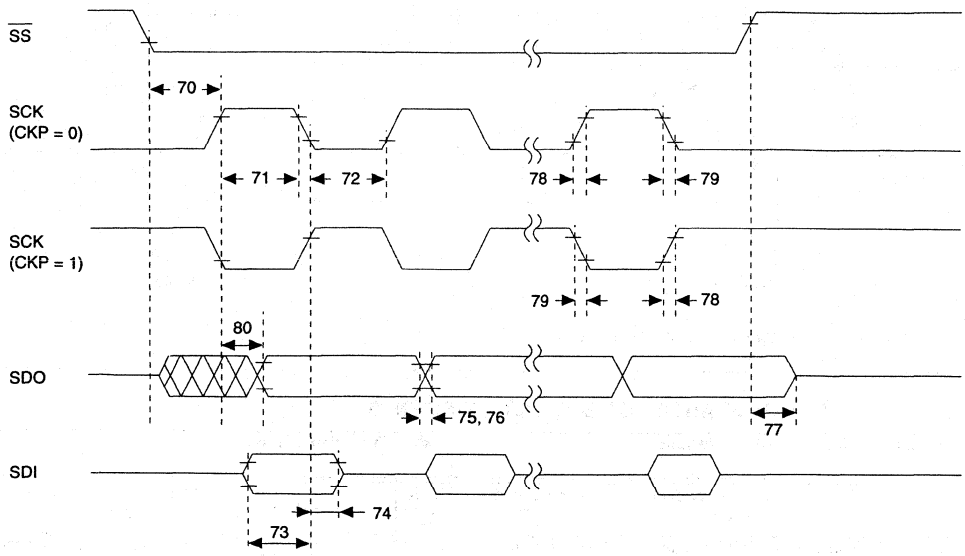
TABLE 16-7: PARALLEL SLAVE PORT REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)	20	—	—	ns	
63	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data–in invalid (hold time)	20	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data–out valid	—	—	40	ns	
65	TrdH2dtI	$\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data–out invalid	10	—	30	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 16-8: SPI MODE TIMING



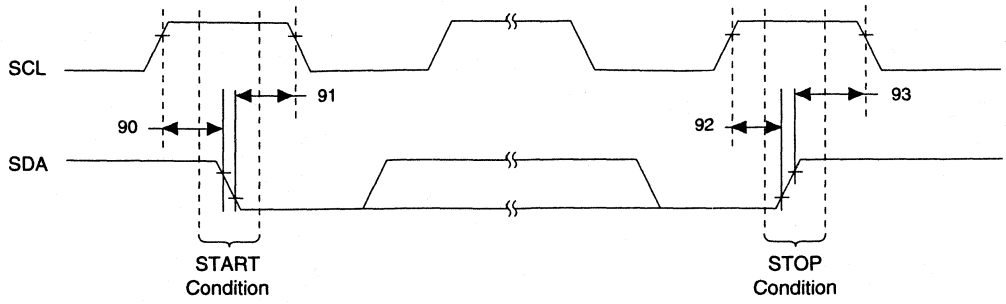
Note: Refer to Figure 16-2 for load conditions

TABLE 16-8: SPI MODE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}$↓ to SCK↓ or SCK↑ input	TCY	—	—	ns	
71	TscH	SCK input high time (slave mode)	TCY + 20	—	—	ns	
72	TscL	SCK input low time (slave mode)	TCY + 20	—	—	ns	
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	TCY	—	—	ns	
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge	0.5 TCY	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
77	TssH2doZ	$\overline{SS}$↓ to SDO output hi-impedence	10	—	50	ns	
78	TscR	SCK output rise time (master mode)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 16-9: I²C BUS START/STOP BITS TIMING



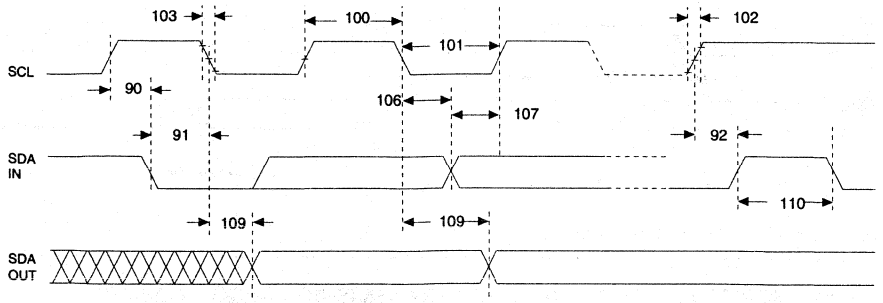
Note: Refer to Figure 16-2 for load conditions

TABLE 16-9: I²C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
90	TSU:STA	START condition Setup time	100 KHZ mode	4700	—	—	ns	Only relevant for repeated START condition
			400 KHz mode	600	—	—		
91	THD:STA	START condition Hold time	100 KHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 KHz mode	600	—	—		
92	TSU:STO	STOP condition Setup time	100 KHz mode	4700	—	—	ns	
			400 KHz mode	600	—	—		
93	THD:STO	STOP condition Hold time	100 KHz mode	4000	—	—	ns	
			400 KHz mode	600	—	—		

 © 1995 Microchip Technology Inc.

FIGURE 16-10: I^2C BUS DATA TIMING



Note: Refer to Figure 16-2 for load conditions

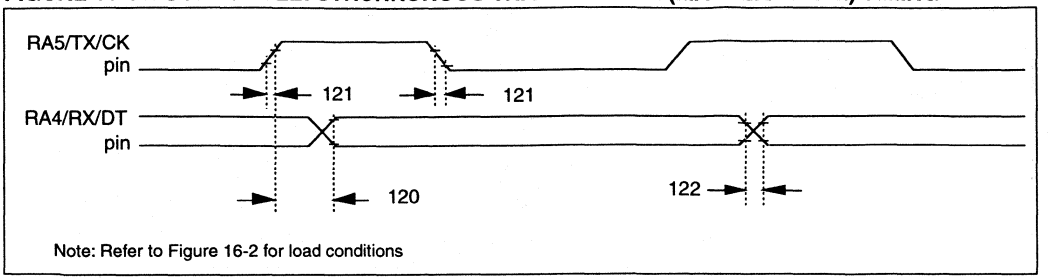
TABLE 16-10: I^2C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock high time	100 KHz mode	4.0	—	µs	PIC16C65 must operate at a minimum of 1.5 MHz
			400 KHz mode	0.6	—	µs	PIC16C65 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
101	TLOW	Clock low time	100 KHz mode	4.7	—	µs	PIC16C65 must operate at a minimum of 1.5 MHz
			400 KHz mode	1.3	—	µs	PIC16C65 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
102	TR	SDA and SCL rise time	100 KHz mode	—	1000	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
103	TF	SDA and SCL fall time	100 KHz mode	—	300	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
90	TSU:STA	START condition setup time	100 KHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 KHz mode	0.6	—	µs	
91	THD:STA	START condition hold time	100 KHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 KHz mode	0.6	—	µs	
106	THD:DAT	Data input hold time	100 KHz mode	0	—	ns	
			400 KHz mode	0	0.9	µs	
107	TSU:DAT	Data input setup time	100 KHz mode	250	—	ns	Note 2
			400 KHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 KHz mode	4.7	—	µs	
			400 KHz mode	0.6	—	µs	
109	TAA	Output valid from clock	100 KHz mode	—	3500	ns	Note 1
			400 KHz mode	—	—	ns	
110	TBUF	Bus free time	100 KHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 KHz mode	1.3	—	µs	
	Cb	Bus capacitive loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode I^2C-bus device can be used in a standard-mode I^2C-bus system, but the requirement tsu;DAT≥250ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT=1000+250=1250ns (according to the standard-mode I^2C bus specification) before the SCL line is released.

FIGURE 16-11: SCI MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



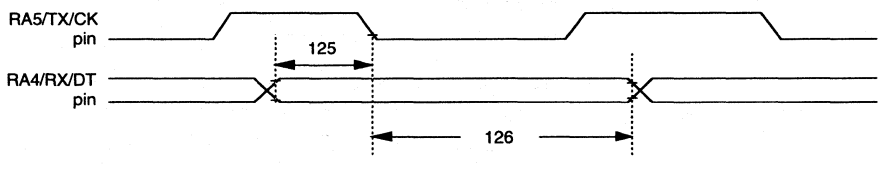
Note: Refer to Figure 16-2 for load conditions

TABLE 16-11: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	tckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	—	—	50	ns	
121	tckrf	Clock out rise time and fall time (Master Mode)	—	—	25	ns	
122	tdtrf	Data out rise time and fall time	—	—	25	ns	

†: Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 16-12: SCI MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 16-2 for load conditions

TABLE 16-12: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	tdtV2ckl	SYNC RCV (MASTER & SLAVE) Data hold before CK ↓ (DT hold time)	15	—	—	ns	
126	tckL2dtl	Data hold after CK ↓ (DT hold time)	15	—	—	ns	

†: Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

17.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES FOR PIC16C65

NOT AVAILABLE AT THIS TIME

© 1995 Microchip Technology Inc.

NOTES:

 © 1995 Microchip Technology Inc.

18.0 ELECTRICAL CHARACTERISTICS FOR PIC16C64

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to+ 125°C
Storage Temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{MCLR}$) ..... -0.6V to VDD +0.6V
Voltage on VDD with respect to VSS ..... 0 to +7.5 V
Voltage on $\overline{MCLR}$ with respect to VSS (Note 2) ..... 0 to +14 V
Total power Dissipation (Note 1) ..... 1.0 W
Maximum Current out of VSS pin 300mA
Maximum Current into VDD pin 250mA
Input clamp current, IIK (VI<0 or VI> VDD) ..... ±20mA
Output clamp current, IOK (V0 <0 or V0>VDD) ..... ±20mA
Maximum Output Current sunk by any I/O pin 25mA
Maximum Output Current sourced by any I/O pin 25mA
Maximum Current sunk by PORTA, PORTB, and PORTE (combined) 200mA
Maximum Current sourced by PORTA, PORTB, and PORTE (combined) 200mA
Maximum Current sunk by PORTC and PORTD (combined) 200mA
Maximum Current sourced by PORTC and PORTD (combined) 200mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - $\sum$ IOH} + $\sum$ {(VDD-VOH) x IOH} + $\sum$(VOl x IOL)

Note 2: Voltage spikes below VSS at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 18-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C64-04	PIC16C64-10	PIC16C64-20	PIC16LC64-04
RC	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21μA Max. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5μA typ. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5μA typ. at 4V WDT disabled Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 2.0mA typ. at 3.0V IPD: 0.9μA typ. at 3V WDT disabled Freq: 4MHz typ.
XT	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21μA Max. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5μA typ. at 4V WDT disabled Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5μA typ. at 4V WDT disabled Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 2.0mA typ. at 3.0V IPD: 0.9μA typ. at 3V.0 WDT disabled Freq: 4MHz typ.
HS	VDD: 4.5V to 5.5V IDD: 13.5mA typ. at 5.5V IPD: 1.5μA typ. at 4.5V WDT disabled Freq: 4MHz	VDD: 4.5V to 5.5V IDD: 30μA Max. at 5.5V IPD: 1.5μA typ. at 4.5V WDT disabled Freq: 20MHz Max.	VDD: 4.5V to 5.5V IDD: 30mA Max. at 5.5V IPD: 1.5μA typ. at 4.5V WDT disabled Freq: 20MHz Max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 22.5μA typ. at 32 KHz, 4.0V IPD: 0.9μA typ. at 4.0V WDT disabled Freq: 200KHz typ.	Do not use in LP mode	Do not use in LP mode	VDD: 3.0V to 6.0V IDD: 48μA Max. at 32 KHz, 3.0V IPD: 13.5μA Max. at 3.0V WDT disabled Freq: 200KHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

18.1 DC CHARACTERISTICS: PIC16C64-04 (Commercial, Industrial, Automotive) PIC16C64-10 (Commercial, Industrial, Automotive) PIC16C64-20 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- - -	2.0 52.5 13.5	3.8 105 30	mA µA mA	XT, RC, osc configuration (PIC16C64-04) FOSC = 4 MHz, VDD = 5.5V (Note 4) LP osc configuration (PIC16C64-04) FOSC = 32 KHz, VDD = 4.0V, WDT disabled HS osc configuration (PIC16C64-20) FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3, 5)	IPD	- - - - - -	10.5 1.5 1.5 1.5 15* 19*	42 21 24 TBD 32* 40*	µA µA µA µA µA µA	VDD=4.0V, WDT enabled,-40°C to +85°C VDD=4.0V, WDT disabled,-0°C to +70°C VDD=4.0V, WDT disabled,-40°C to +85°C VDD=4.0V, WDT disabled,-40°C to +125°C LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Commercial LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and V_{SS}.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

 © 1995 Microchip Technology Inc.

18.2 DC CHARACTERISTICS: PIC16LC64-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 3.0	- -	6.0 6.0	V V	LP osc configuration XT, RC osc configuration (DC - 4MHz)
RAM Data Retention Voltage (Note 1)	VDR	1.5*	-	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- -	2.0 22.5	3.8 48	mA µA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4) LP OSC CONFIGURATION FOSC = 32 KHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3, 5)	IPD	- - - -	7.5 0.9 0.9 0.9	30 13.5 18 24	µA µA µA µA	VDD=3.0V, WDT enabled,-40°C to +85°C VDD=3.0V, WDT disabled,0°C to +70°C VDD=3.0V, WDT disabled,-40°C to +85°C VDD=3.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

18.3 DC CHARACTERISTICS:

PIC16C64-04 (Commercial, Industrial, Automotive)
PIC16C64-10 (Commercial, Industrial, Automotive)
PIC16C64-20 (Commercial, Industrial, Automotive)
PIC16LC64-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ + 125°C for automotive,
-40°C ≤ TA ≤ + 85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Table 18-1 and Table 18-2

Characteristic	Sym	Min	Typ †	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	-	0.8V	V	
with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
MCLR, RA4/T0CKI,OSC1 (in RC mode)		Vss	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
Input High Voltage						
I/O ports	VIH		-			
with TTL buffer		2.0	-	VDD	V	4.5V ≤ VDD ≤ 5.5V
with Schmitt Trigger buffer		0.85VDD	-	VDD		For entire VDD range
MCLR RA4/T0CKI, RC<7:4>, RD<7.4>, INT		0.85VDD	-	VDD	V	
RE<2:0>, OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	200	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)						
I/O ports	IIL	-	-	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RA4/T0CKI		-	-	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	-	-	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD-4.5V, -40°C to +125°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD-4.5V, -40°C to +125°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +125°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	
SCL, SDA in I^2C mode	Cb			400	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C64 be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

 © 1995 Microchip Technology Inc.

18.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

18.5 Timing Diagrams and Specifications

FIGURE 18-1: EXTERNAL CLOCK TIMING

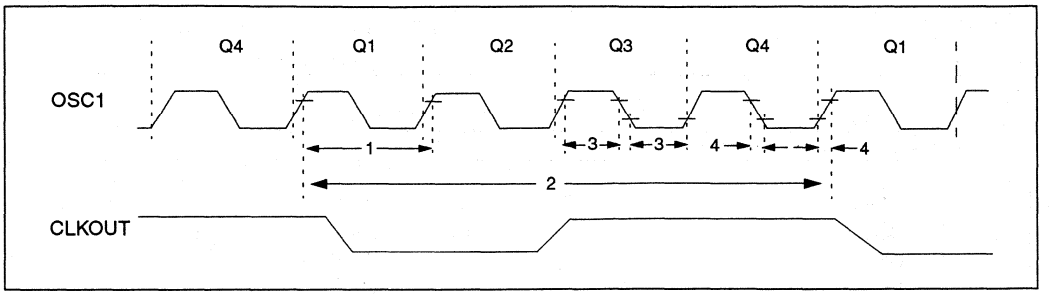


TABLE 18-2: EXTERNAL CLOCK TIMING REQUIREMENTS

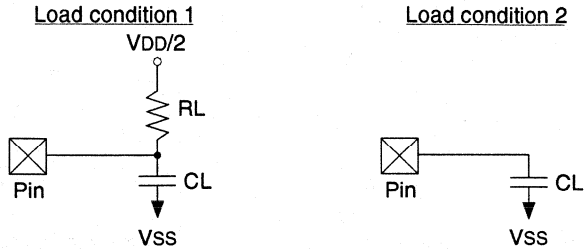
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C64-04, PIC16LC64-04)
			DC	—	20	MHz	HS osc mode (PIC16C64-20)
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C64-04, PIC16LC64-04)
			4	—	10	MHz	HS osc mode (PIC16C64-10)
			4	—	20	MHz	HS osc mode (PIC16C64-20)
			5	—	200	KHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C64-04, PIC16LC64-04)
			100	—	—	ns	HS osc mode (PIC16C64-10)
			50	—	—	ns	HS osc mode (PIC16C64-20)
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C64-04, PIC16LC64-04)
			100	—	250	ns	HS osc mode (PIC16C64-10)
			50	—	1,000	ns	HS osc mode (PIC16C64-20)
			5	—	—	µs	LP osc mode
2	TCY	Instruction Cycle Time (Note 1)	200	—	DC	ns	TCY = 4/FOSC
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

 © 1995 Microchip Technology Inc.

FIGURE 18-2: LOAD CONDITIONS

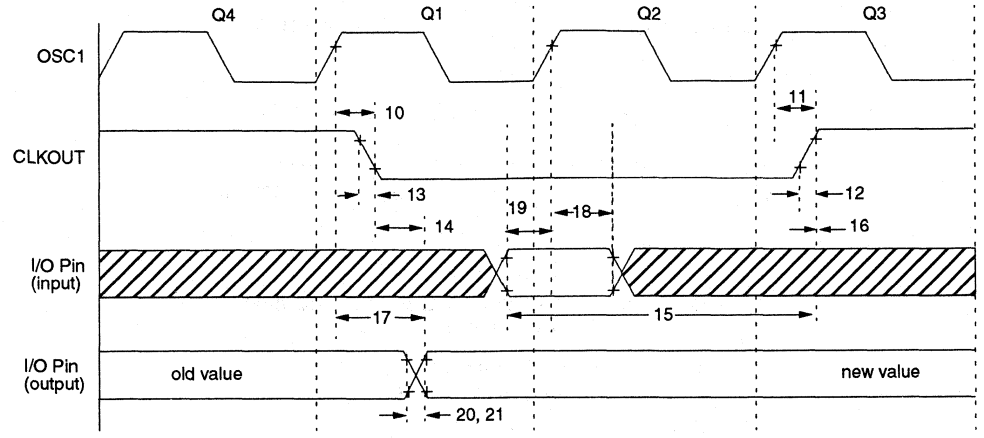


RL = 464Ω
CL = 100 pF for RA<4> and C, D, E port outputs when used as system bus

50 pF for all pins except OSC2 and RA<4> but including C, D, E outputs as ports

25 pF for OSC2 output

FIGURE 18-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 18-2 for load conditions.

TABLE 18-3: CLKOUT AND I/O TIMING REQUIREMENTS

These parameters are asynchronous events not related to any internal clock edge.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	INT pin high or low time	20	—	—	ns	
23††	Trbp	RB<7:4> change INT high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

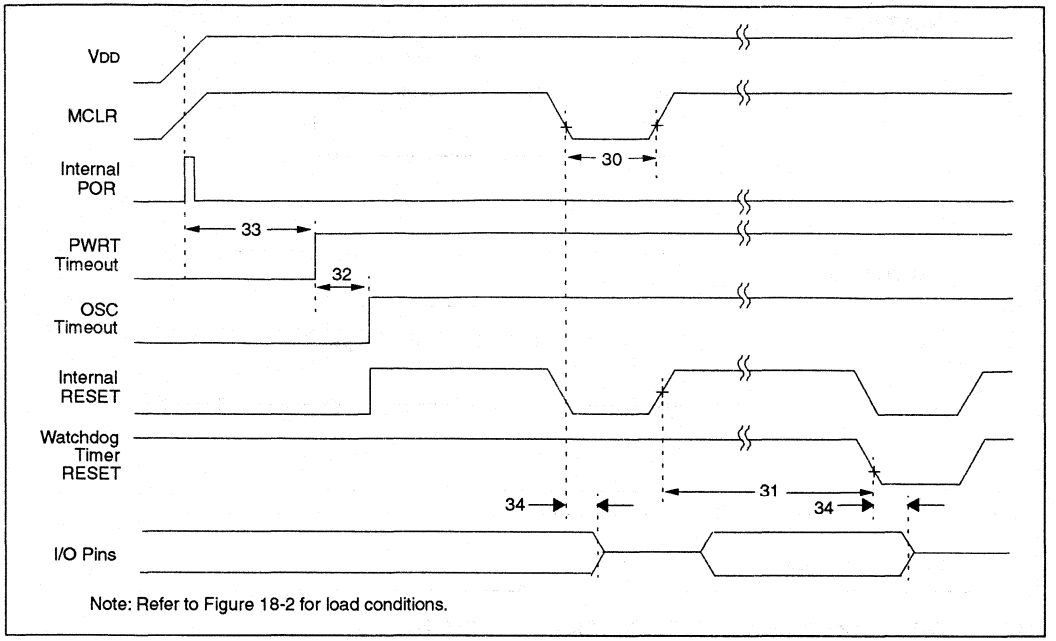
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edge.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC

 © 1995 Microchip Technology Inc.

FIGURE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 18-2 for load conditions.

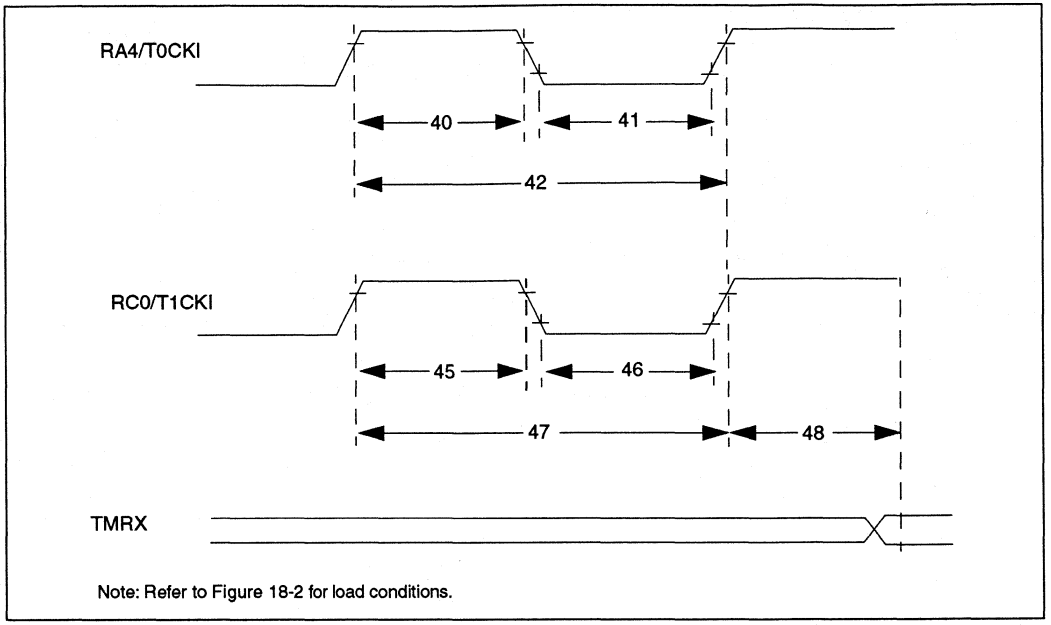
TABLE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O High Impedance from $\overline{MCLR}$ Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 18-5: TIMER0 AND TIMER1 CLOCK TIMINGS



Note: Refer to Figure 18-2 for load conditions.

TABLE 18-5: TIMER0 AND TIMER1 CLOCK REQUIREMENTS

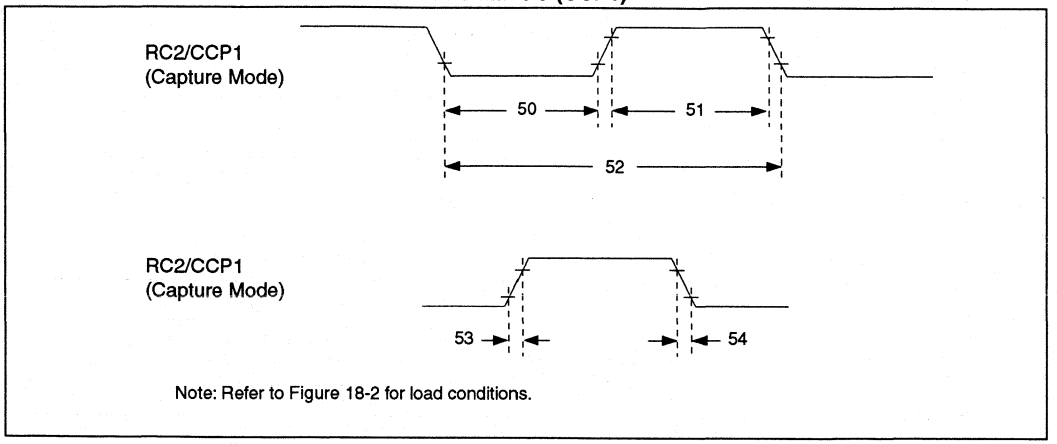
Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, ..., 256)
45	Tt1H	T1CKI High Time	Synchronous, No Prescaler	0.5TCY + 20	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
46	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	0.5TCY + 20*	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
47	Tt1P	T1CKI input period	Synchronous	TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	4 TCY	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting the T1OSCEN bit)		DC	—	200	KHz	
48	TCKEZtmr1	Delay from external clock edge to timer increment		2 Tosc	—	7Tosc	—	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 18-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1)



Note: Refer to Figure 18-2 for load conditions.

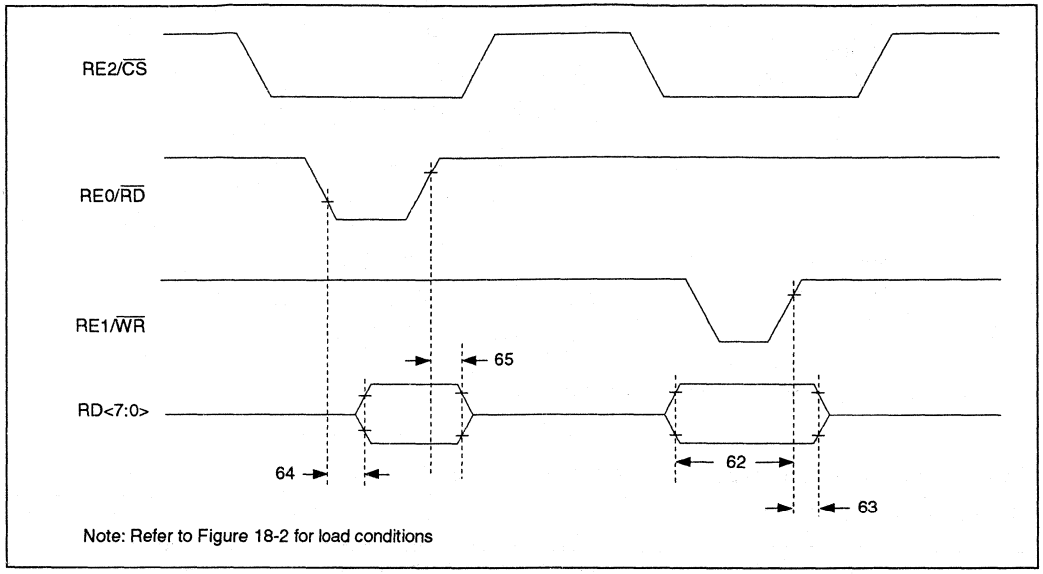
TABLE 18-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
50	TccL	CCP1 input low time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
51	TccH	CCP1 input high time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
52	TccP	CCP1 input period		$\frac{3T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53	TccR	CCP1 output rise time		—	10	25	ns	
54	TccF	CCP1 output fall time		—	10	25	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 18-7: PARALLEL SLAVE PORT TIMING



Note: Refer to Figure 18-2 for load conditions

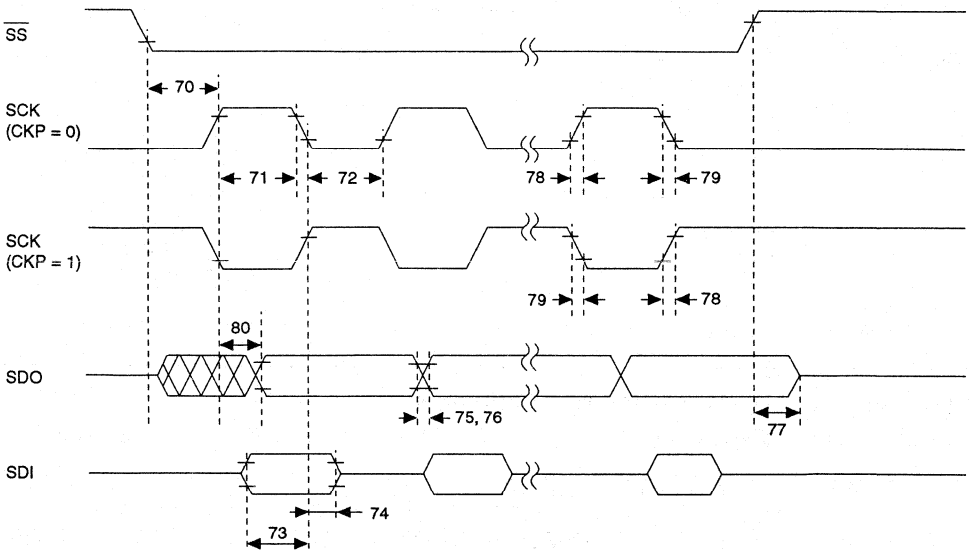
TABLE 18-7: PARALLEL SLAVE PORT REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)	20	—	—	ns	
63	TwrH2dtI	WR↑ or CS↑ to data–in invalid (hold time)	20	—	—	ns	
64	TrdL2dtV	RD↓ and CS↓ to data–out valid	—	—	40	ns	
65	TrdH2dtI	RD↑ or CS↓ to data–out invalid	10	—	30	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 18-8: SPI MODE TIMING



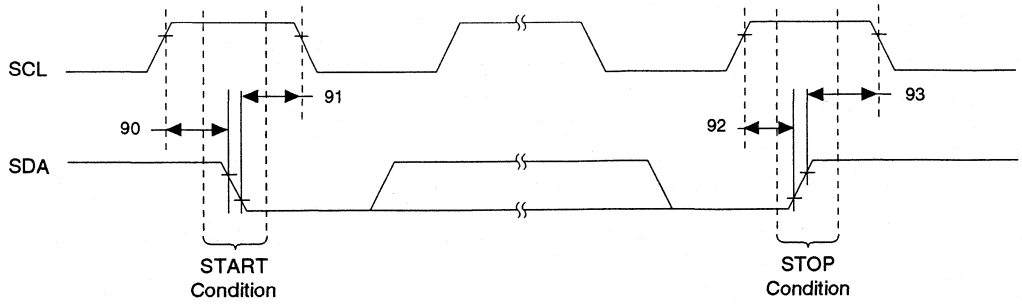
Note: Refer to Figure 18-2 for load conditions

TABLE 18-8: SPI MODE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input	Tcy	—	—	ns	
71	TscH	SCK input high time (slave mode)	Tcy + 20	—	—	ns	
72	TscL	SCK input low time (slave mode)	Tcy + 20	—	—	ns	
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	Tcy	—	—	ns	
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge	0.5 Tcy	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
77	TssH2doZ	$\overline{SS}\downarrow$ to SDO output hi-impedence	10	—	50	ns	
78	TscR	SCK output rise time (master mode)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 18-9: I²C BUS START/STOP BITS TIMING



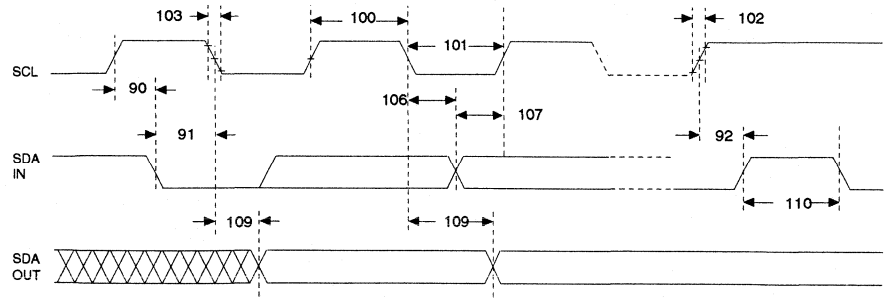
Note: Refer to Figure 18-2 for load conditions

TABLE 18-9: I²C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
90	TSU:STA	START condition Setup time	100 KHZ mode	4700	—	—	ns	Only relevant for repeated START condition
			400 KHz mode	600	—	—		
91	THD:STA	START condition Hold time	100 KHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 KHz mode	600	—	—		
92	TSU:STO	STOP condition Setup time	100 KHZ mode	4700	—	—	ns	
			400 KHz mode	600	—	—		
93	THD:STO	STOP condition Hold time	100 KHz mode	4000	—	—	ns	
			400 KHz mode	600	—	—		

 © 1995 Microchip Technology Inc.

FIGURE 18-10: I²C BUS DATA TIMING



Note: Refer to Figure 18-2 for load conditions

TABLE 18-10: I²C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock high time	100 KHz mode	4.0	—	µs	PIC16C64 must operate at a minimum of 1.5 MHz
			400 KHz mode	0.6	—	µs	PIC16C64 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
101	TLOW	Clock low time	100 KHz mode	4.7	—	µs	PIC16C64 must operate at a minimum of 1.5 MHz
			400 KHz mode	1.3	—	µs	PIC16C64 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
102	TR	SDA and SCL rise time	100 KHz mode	—	1000	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
103	TF	SDA and SCL fall time	100 KHz mode	—	300	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
90	TSU:STA	START condition setup time	100 KHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 KHz mode	0.6	—	µs	
91	THD:STA	START condition hold time	100 KHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 KHz mode	0.6	—	µs	
106	THD:DAT	Data input hold time	100 KHz mode	0	—	ns	
			400 KHz mode	0	0.9	µs	
107	TSU:DAT	Data input setup time	100 KHz mode	250	—	ns	Note 2
			400 KHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 KHz mode	4.7	—	µs	
			400 KHz mode	0.6	—	µs	
109	TAA	Output valid from clock	100 KHz mode	—	3500	ns	Note 1
			400 KHz mode	—	—	ns	
110	TBUF	Bus free time	100 KHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 KHz mode	1.3	—	µs	
	Cb	Bus capacitive loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode I²C-bus device can be used in a standard-mode I²C-bus system, but the requirement tsu;DAT≥250ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT=1000+250=1250ns (according to the standard-mode I²C bus specification) before the SCL line is released.

NOTES:

 © 1995 Microchip Technology Inc.

19.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES FOR PIC16C64

NOT AVAILABLE AT THIS TIME

NOTES:

 © 1995 Microchip Technology Inc.

20.0 ELECTRICAL CHARACTERISTICS FOR PIC16C61

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to+ 125°C
Storage Temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{MCLR}$) ..... -0.6V to VDD +0.6V
Voltage on VDD with respect to VSS ..... 0 to +7.5 V
Voltage on $\overline{MCLR}$ with respect to VSS (Note 2) ..... 0 to +14 V
Total power Dissipation (Note 1) 800mW
Maximum Current out of VSS pin 150mA
Maximum Current into VDD pin 100mA
Input clamp current, IIK (VI<0 or VI> VDD) ..... ±20mA
Output clamp current, IOK (V0 <0 or V0>VDD) ..... ±20mA
Maximum Output Current sunk by any I/O pin 25mA
Maximum Output Current sourced by any I/O pin 20mA
Maximum Current sunk by PORTA 80mA
Maximum Current sourced by PORTA 50mA
Maximum Current sunk by PORTB 150mA
Maximum Current sourced by PORTB 100mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - Σ IOH} + Σ {(VDD-VOH) x IOH} + Σ(VOl x IOL)

Note 2: Voltage spikes below VSS at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 20-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C61-04	16C61-20	16LC61-04
RC	VDD: 4.0V to 6.0V IDD: 3.3mA Max. at 5.5V IPD: 14µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 1.8mA typ. at 5.5V IPD: 1.0µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 1.4mA typ. at 3.0V IPD: 0.6µA typ. at 3V WDT dis Freq: 4MHz typ.
XT	VDD: 4.0V to 6.0V IDD: 3.3mA Max. at 5.5V IPD: 14µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 1.8mA typ. at 5.5V IPD: 1.0µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 1.4mA typ. at 3.0V IPD: 0.6µA typ. at 3V WDT dis Freq: 4MHz typ.
HS	VDD: 4.5V to 5.5V IDD: 13.5mA typ. at 5.5V IPD: 1.0µA typ. at 4.5V WDT dis Freq: 4MHz	VDD: 4.5V to 5.5V IDD: 30mA Max. at 5.5V IPD: 1.0µA typ. at 4.5V WDT dis Freq: 20MHz Max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 15µA typ. at 32 KHz, 4.0V IPD: 0.6µA typ. at 4.0V WDT dis Freq: 200KHz typ.	Do not use in LP mode	VDD: 3.0V to 6.0V IDD: 32µA Max. at 32 KHz, 3.0V IPD: 9µA Max. at 3.0V WDT dis Freq: 200KHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

© 1995 Microchip Technology Inc.

20.1 DC CHARACTERISTICS: PIC16C61-04 (Commercial, Industrial, Automotive) PIC16C61-20 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2)	IDD	- - -	1.8 35 13.5	3.3 70 30	mA µA mA	FOSC = 4 MHz, VDD = 5.5V (Note 4) LP osc configuration (PIC16LC61-04) FOSC = 32 KHz, VDD = 4.0V, WDT disabled HS osc configuration (PIC16C61-20) FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3)	IPD	- - - -	7 1.0 1.0 1.0	28 14 16 20	µA µA µA µA	VDD=4.0V, WDT enabled,-40°C to +85°C VDD=4.0V, WDT disabled,-0°C to +70°C VDD=4.0V, WDT disabled,-40°C to +85°C VDD=4.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and V_{SS}.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

20.2 DC CHARACTERISTICS: PIC16LC61-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0	-	6.0	V	XT, RC, and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2)	IDD	-	1.4	2.5	mA	FOSC = 4 MHz, VDD = 3.0V (Note 4)
		-	15	32	µA	FOSC = 32 KHz, VDD = 3.0V, WDT disabled (Note 5)
Power Down Current (Note 3)	IPD	-	5	20	µA	VDD=3.0V, WDT enabled,-40°C to +85°C
		-	0.6	9	µA	VDD=3.0V, WDT disabled,0°C to +70°C
		-	0.6	12	µA	VDD=3.0V, WDT disabled,-40°C to +85°C
		-	0.6	16	µA	VDD=3.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

4: XT, RC OSC configuration. For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: LP OSC configuration.

© 1995 Microchip Technology Inc.

20.3 DC CHARACTERISTICS: PIC16C61-04 (Commercial, Industrial, Automotive)
PIC16C61-20 (Commercial, Industrial, Automotive)
PIC16LC61-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ + 125°C for automotive,
-40°C ≤ TA ≤ + 85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Table 20-1 and Table 20-2

Characteristic	Sym	Min	Typ †	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	-	0.8V	V	
with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
MCLR, RA4/T0CKI,OSC1 (in RC mode)		Vss	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
Input High Voltage						
I/O ports	VIH		-			
with TTL buffer		0.36VDD	-	VDD	V	4.5V ≤ VDD ≤ 5.5V
		0.45VDD	-	VDD		For entire VDD range
with Schmitt Trigger buffer		0.85 VDD	-	VDD	V	For entire VDD range
MCLR RA4/T0CKI		0.85 VDD	-	VDD	V	
OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	250	† 400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)						
I/O ports	IIL	-	-	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RA4/T0CKI		-	-	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	-	-	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD-4.5V, -40°C to +125°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD-4.5V, -40°C to +125°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +125°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C61 be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

© 1995 Microchip Technology Inc.

20.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. Tcc:st (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

Tcc:st (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

20.5 Timing Diagrams and Specifications

FIGURE 20-1: EXTERNAL CLOCK TIMING

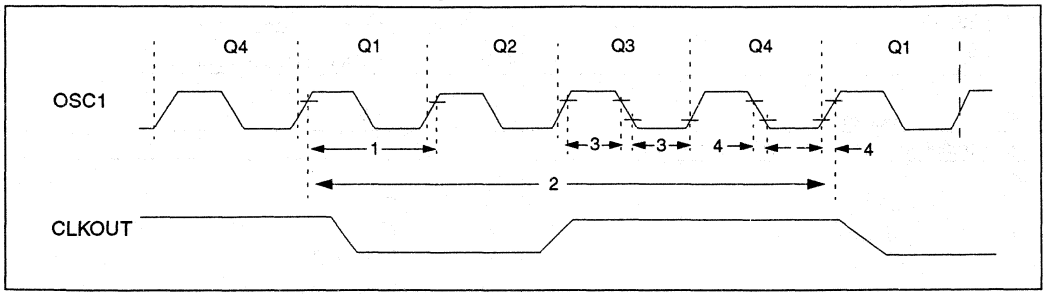


TABLE 20-2: EXTERNAL CLOCK TIMING REQUIREMENTS

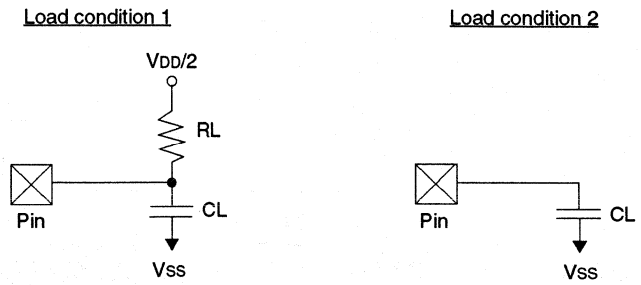
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fos	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C61-04, PIC16C61-04)
			DC	—	20	MHz	HS osc mode (PIC16C61-20)
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			1	—	4	MHz	HS osc mode (PIC16C61-04, PIC16C61-04)
			1	—	20	MHz	HS osc mode (PIC16C61-20)
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C61-04, PIC16C61-04)
			50	—	—	ns	HS osc mode (PIC16C61-20)
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	1,000	ns	HS osc mode (PIC16C61-04, PIC16C61-04)
			50	—	1,000	ns	HS osc mode (PIC16C61-20)
			5	—	—	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	1.0	4/Fosc	DC	µs	
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 1995 Microchip Technology Inc.

FIGURE 20-2: LOAD CONDITIONS

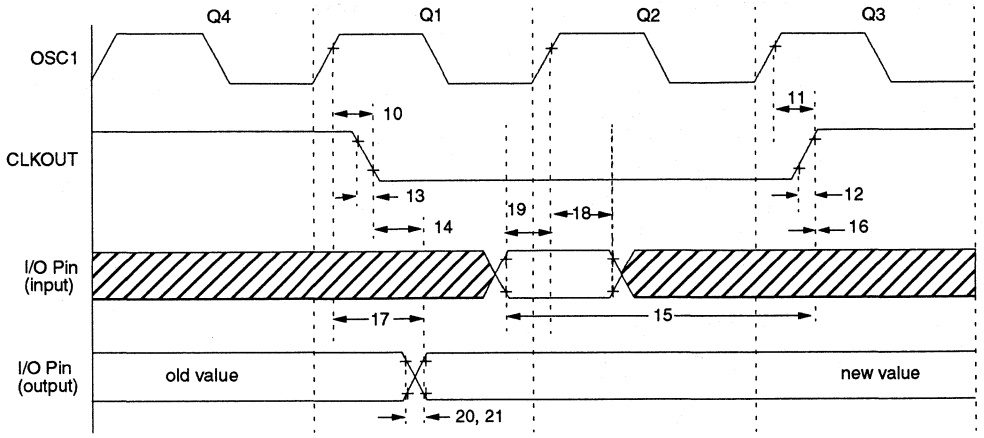


RL = 464 Ω

CL = 50 pF for all pins except OSC2
25 pF for OSC2 output

© 1995 Microchip Technology Inc.

FIGURE 20-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 20-2 for load conditions.

TABLE 20-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	0.25 TCY+30	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	INT pin high or low time	20	—	—	ns	
23††	Trbp	RB<7:4> change INT high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

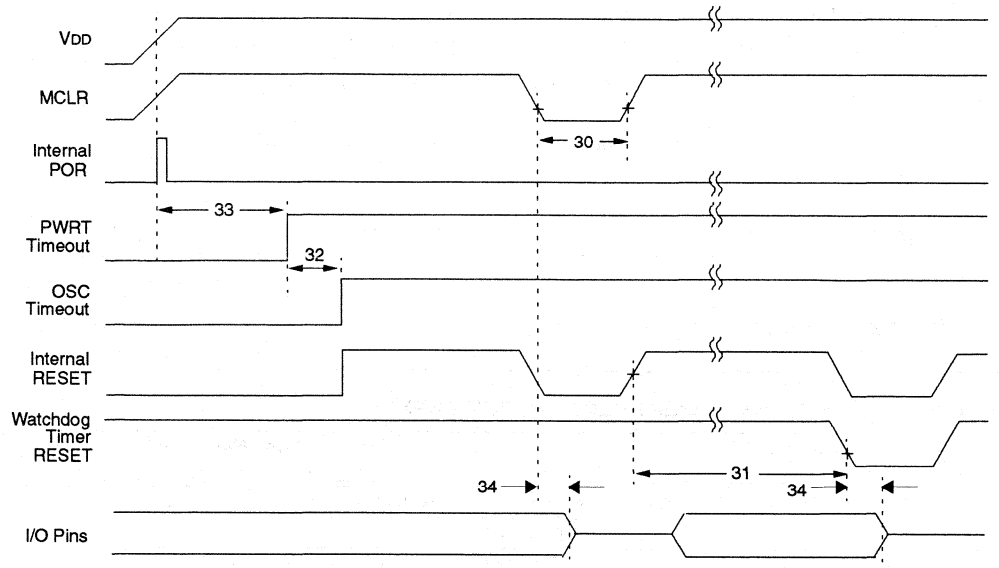
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc

© 1995 Microchip Technology Inc.

FIGURE 20-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 20-2 for load conditions.

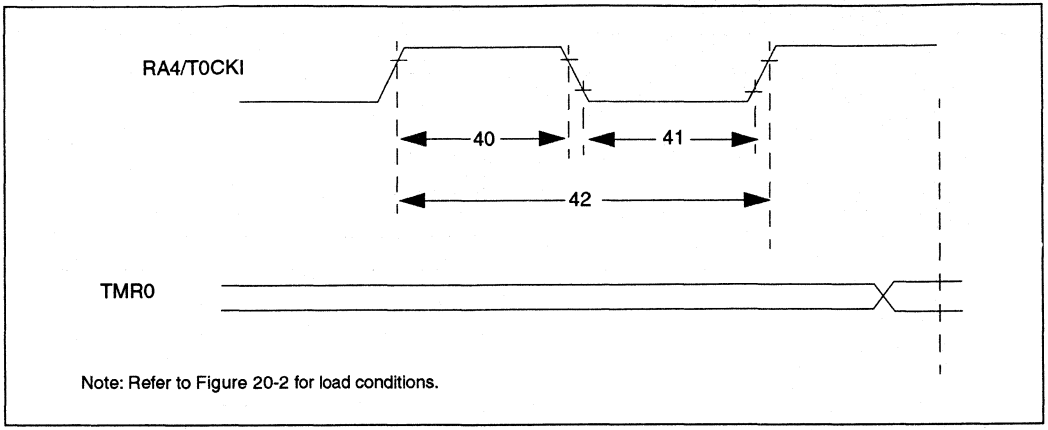
TABLE 20-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	200	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	Tioz	I/O High Impedance from $\overline{MCLR}$ Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 20-5: TIMER0 CLOCK TIMINGS



Note: Refer to Figure 20-2 for load conditions.

TABLE 20-5: TIMER0 AND TIMER1 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

21.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES FOR PIC16C61

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. 'Typical' represents the mean of the distribution while 'max' or 'min' represents (mean + 3s) and (mean - 3s) respectively where s is standard deviation.

FIGURE 21-1: TYPICAL RC OSCILLATOR FREQUENCY VS TEMPERATURE

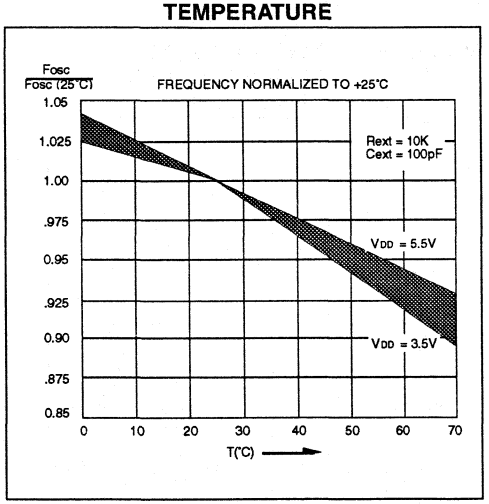


FIGURE 21-2: TYPICAL RC OSCILLATOR FREQUENCY VS VDD

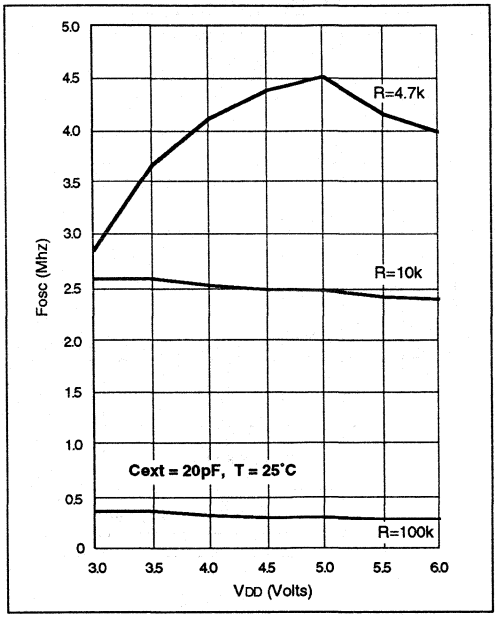


FIGURE 21-3: TYPICAL RC OSCILLATOR FREQUENCY VS VDD

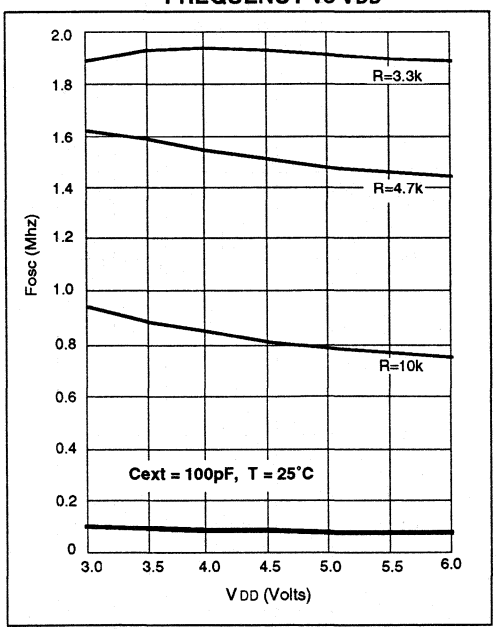


FIGURE 21-4: TYPICAL RC OSCILLATOR FREQUENCY vs VDD

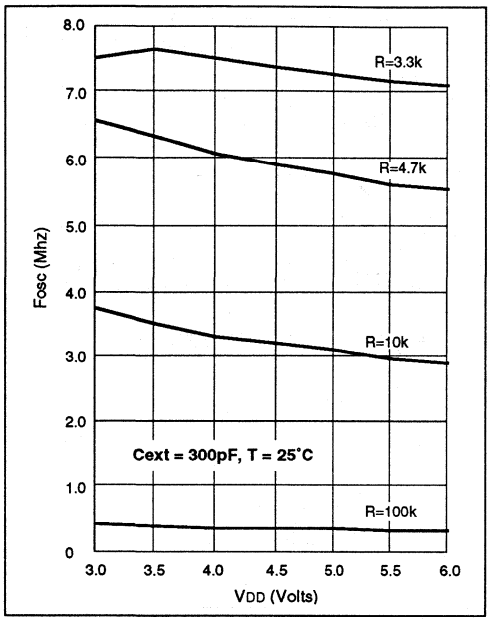


FIGURE 21-5: TYPICAL IPD vs VDD WATCHDOG TIMER DISABLED 25°C

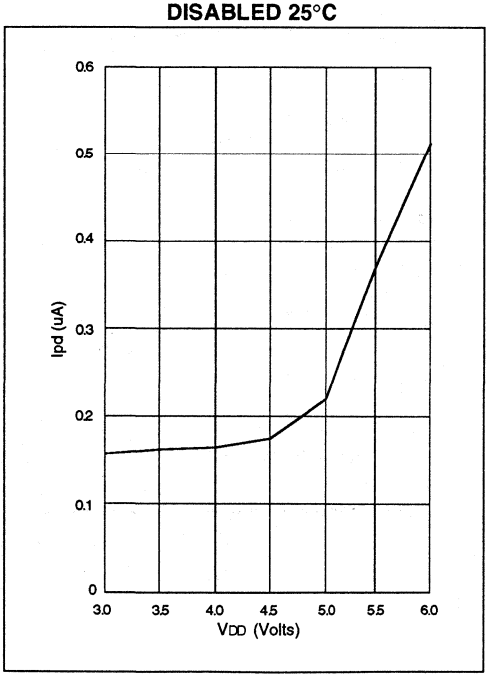
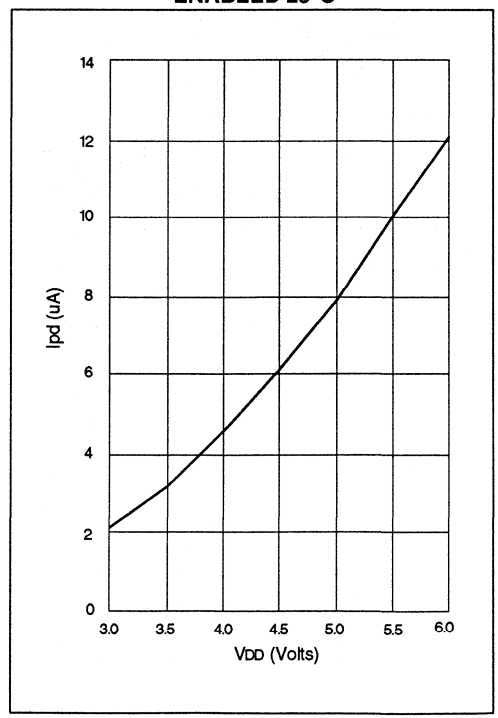


TABLE 21-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20pf	4.7K	4.52 MHz	±17.35%
	10k	2.47 MHz	±10.10%
	100k	290.86 KHz	±11.90%
100pf	3.3k	1.92 MHz	±9.43%
	4.7k	1.48 MHz	±9.83%
	10k	788.77 KHz	±10.92%
	100k	88.11 KHz	±16.03%
300pf	3.3k	726.89 KHz	±10.97%
	4.7k	573.95 KHz	±10.14%
	10k	307.31 KHz	±10.43%
	100k	33.82 KHz	±11.24%

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

FIGURE 21-6: TYPICAL IPD vs VDD WATCHDOG TIMER ENABLED 25°C



© 1995 Microchip Technology Inc.

FIGURE 21-7: MAXIMUM IPD VS VDD WATCHDOG DISABLED

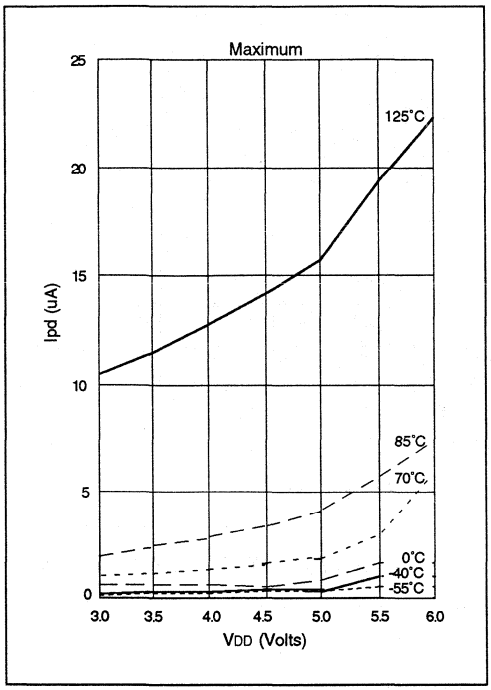
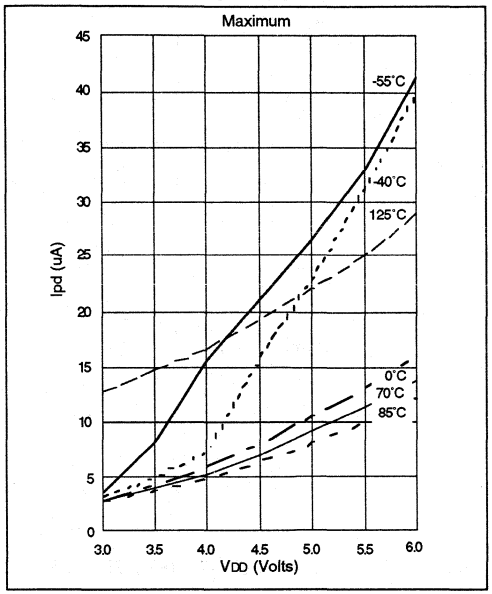


FIGURE 21-8: MAXIMUM IPD VS VDD WATCHDOG ENABLED*



*Ipd, with watchdog timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the watchdog timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 21-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS VS VDD

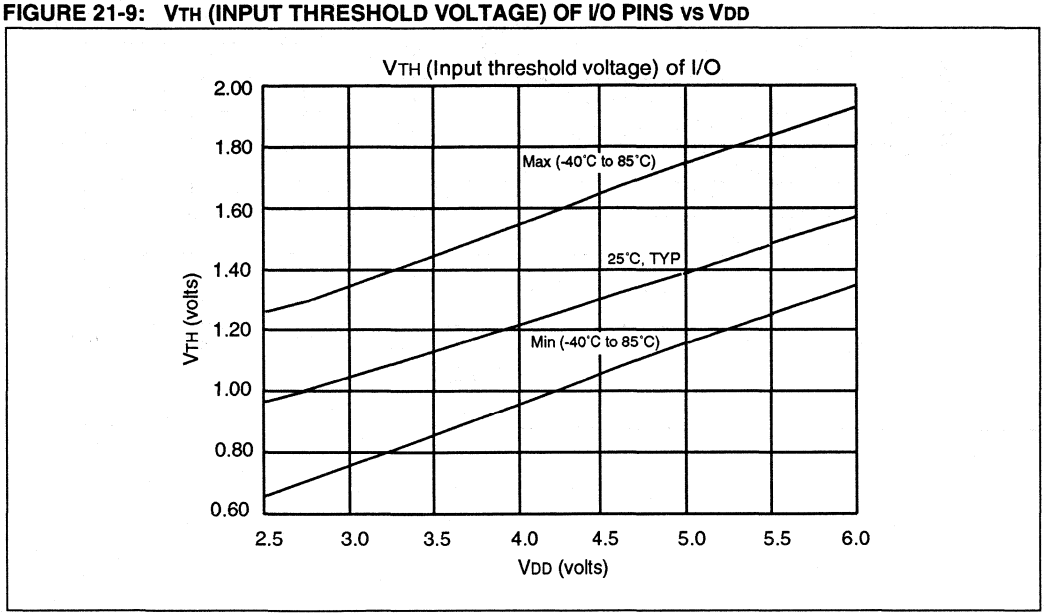
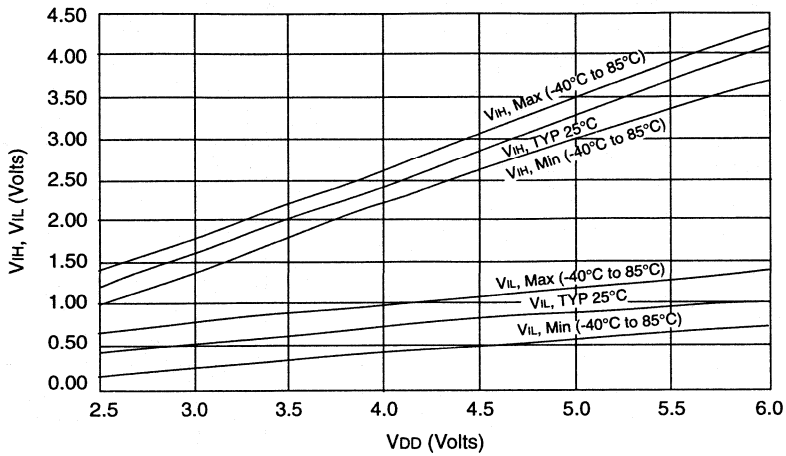
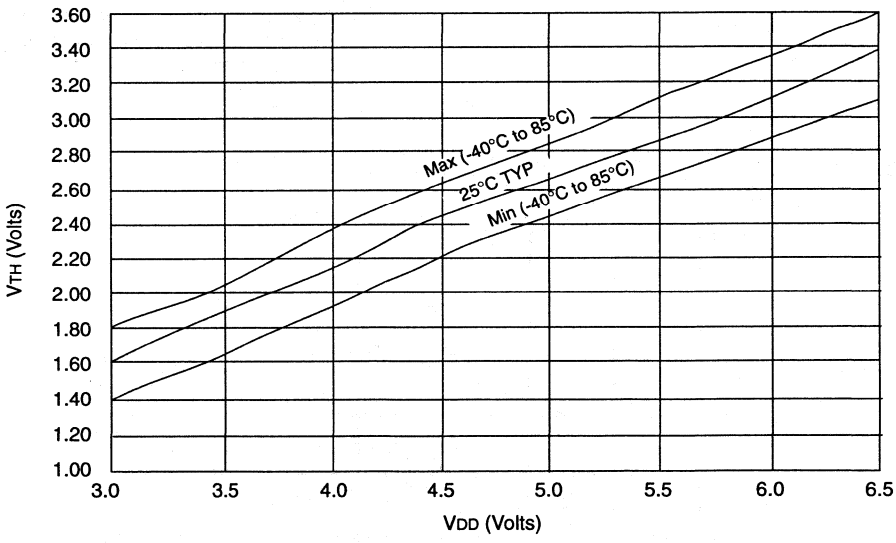


FIGURE 21-10: VIH, VIL OF $\overline{MCLR}$, T0CKI AND OSC1 (IN RC MODE) VS VDD



Note: These input pins have Schmitt trigger input buffer.

FIGURE 21-11: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) VS VDD



© 1995 Microchip Technology Inc.

FIGURE 21-12: TYPICAL IDD VS FREQ (EXT CLOCK, 25°C)

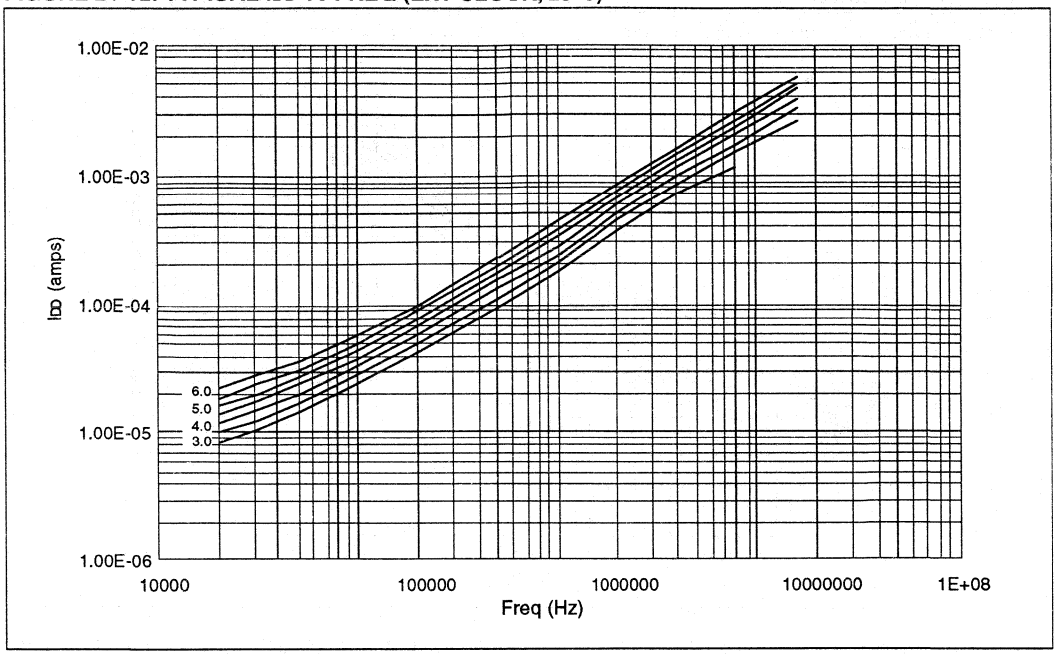


FIGURE 21-13: MAXIMUM, IDD VS FREQ (EXT CLOCK, -40° TO +85°C)

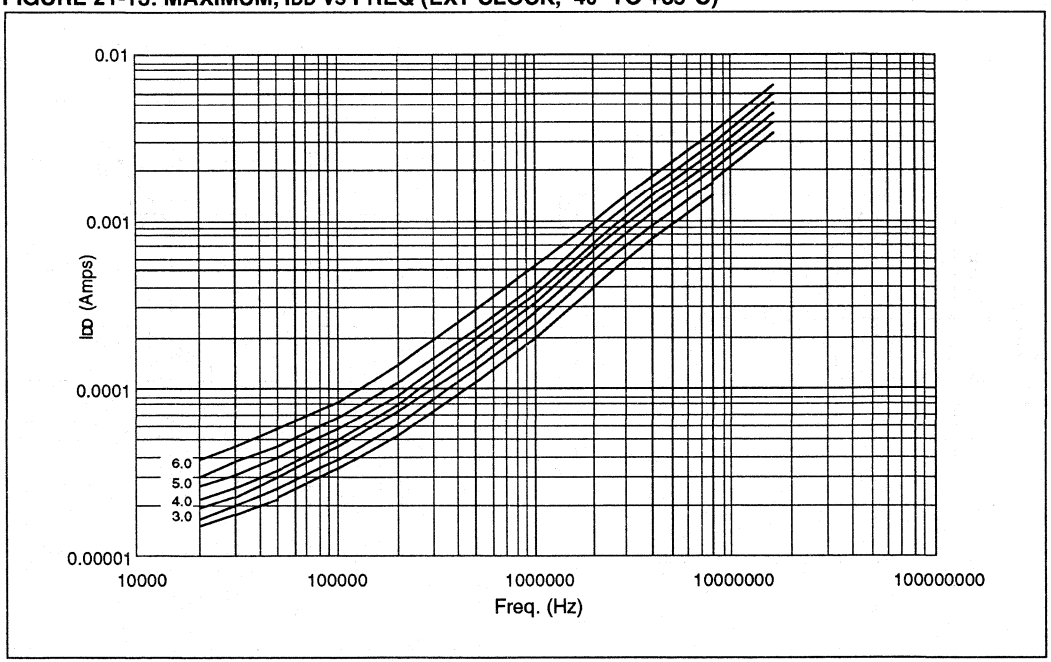


FIGURE 21-14: MAXIMUM IDD VS FREQ WITH A/D OFF (EXT CLOCK, -55° TO +125°C)

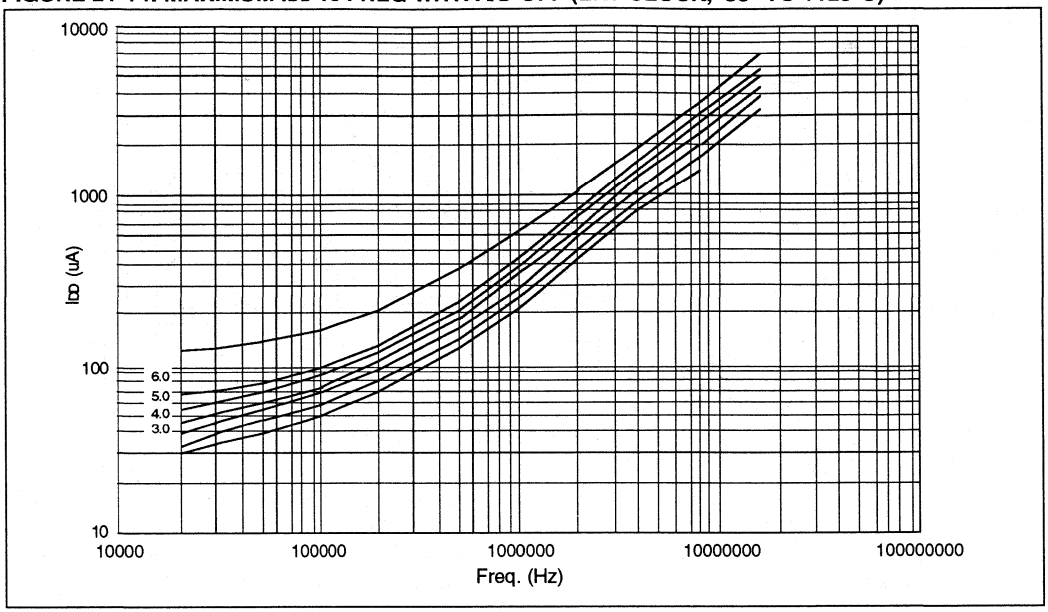


FIGURE 21-15: WDT TIMER TIME-OUT PERIOD VS VDD

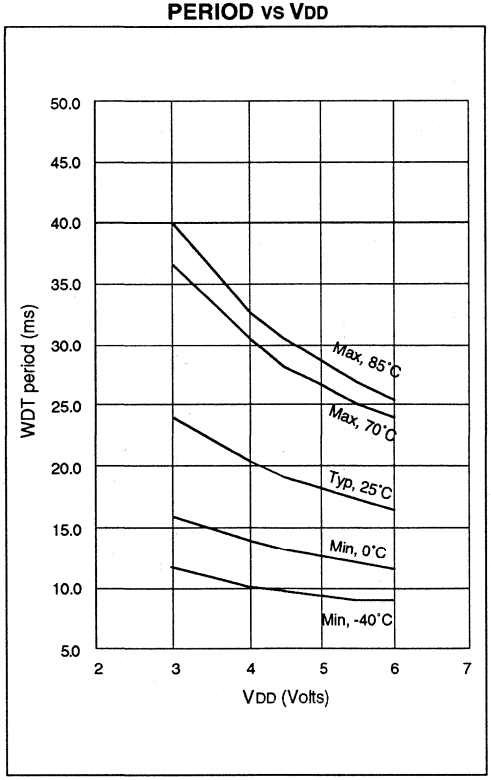
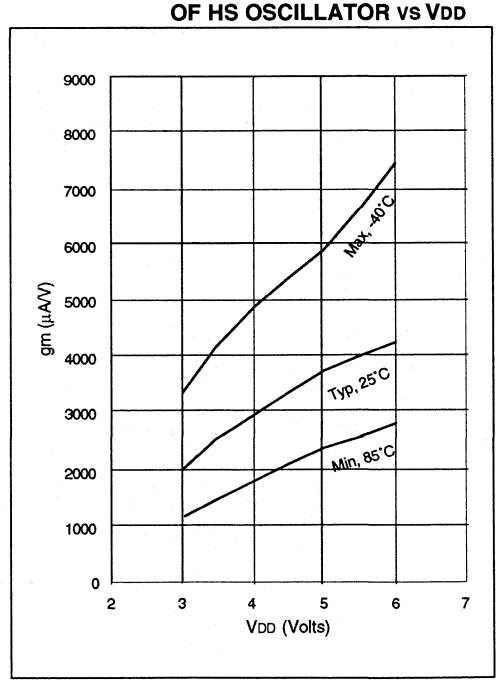


FIGURE 21-16: TRANSCONDUCTANCE (GM) OF HS OSCILLATOR VS VDD



© 1995 Microchip Technology Inc.

FIGURE 21-17: TRANSCONDUCTANCE (GM) OF LP OSCILLATOR vs VDD

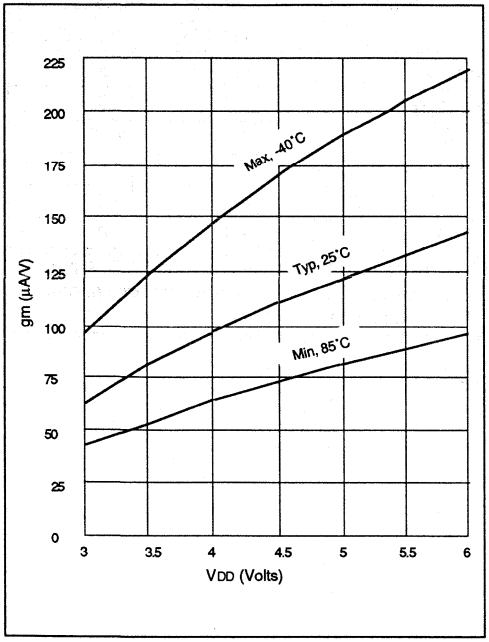


FIGURE 21-18: TRANSCONDUCTANCE (GM) OF XT OSCILLATOR vs VDD

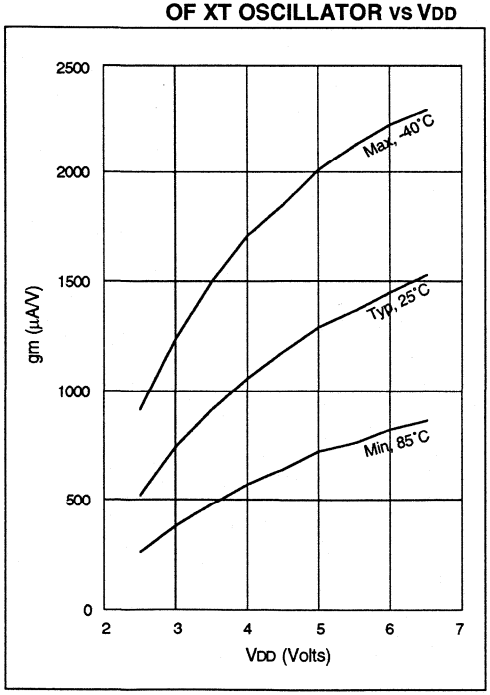


FIGURE 21-19: IOH vs VOH, VDD = 3V

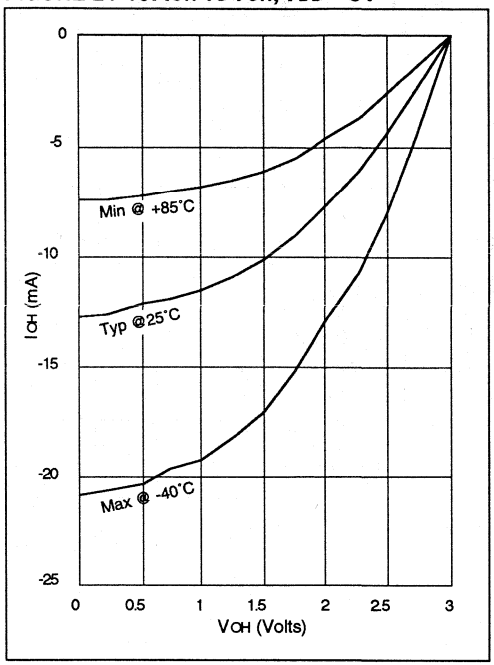
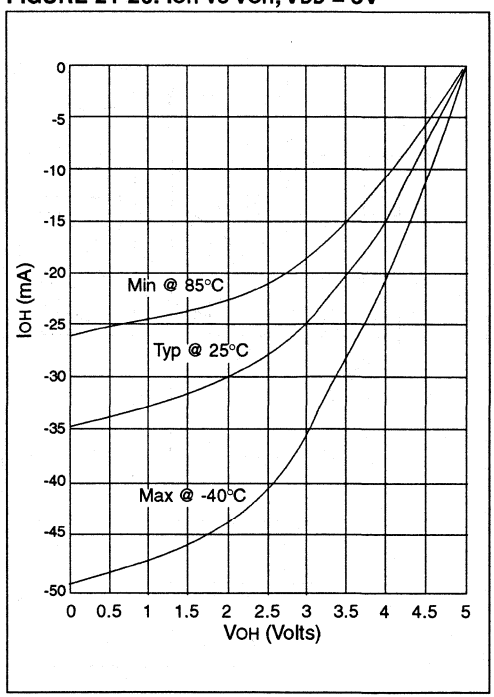


FIGURE 21-20: IOH vs VOH, VDD = 5V



2

FIGURE 21-21: IOL VS VOL, VDD = 3V

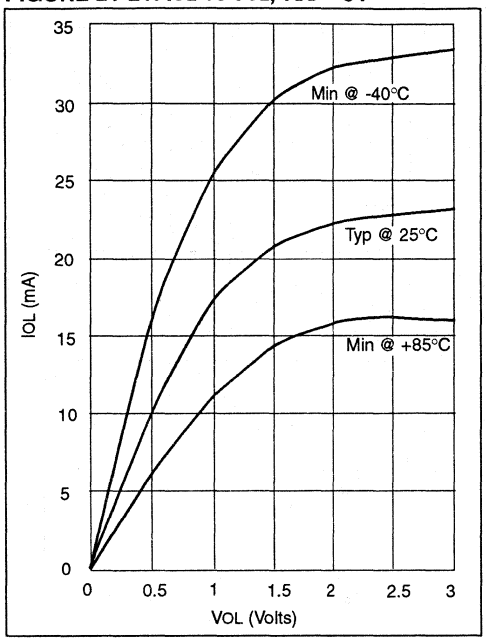


FIGURE 21-22: IOL VS VOL, VDD = 5V

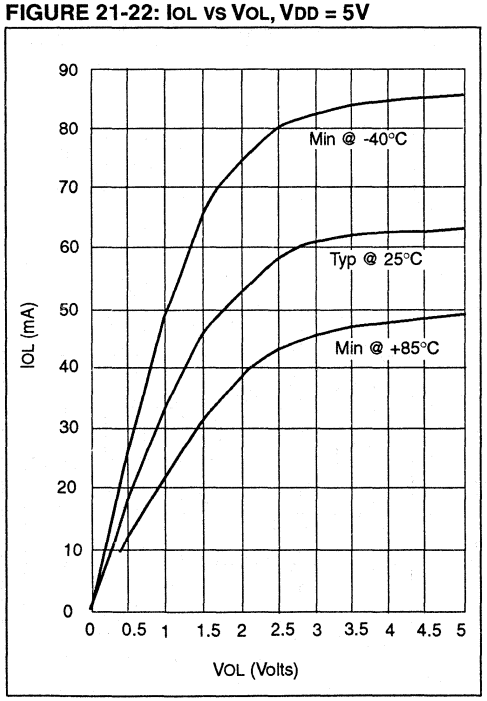


TABLE 21-2: INPUT CAPACITANCE*

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
RA port	5.0	4.3
RB port	5.0	4.3
$\overline{MCLR}$	17.0	17.0
OSC1/CLKIN	4.0	3.5
OSC2/CLKOUT	4.3	3.5
T0CKI	3.2	2.8

*All capacitance values are typical at 25°C. A part to part variation of ±25% (three standard deviations) should be taken into account.

© 1995 Microchip Technology Inc.

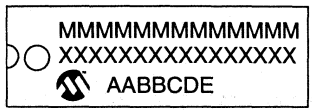
22.0 PACKAGING INFORMATION

For Package Dimensions, please refer to the Packaging Section of the Data Book.

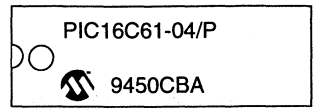
© 1995 Microchip Technology Inc.

22.1 Package Marking Information

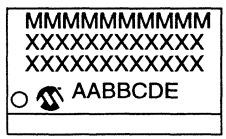
18-Lead PDIP



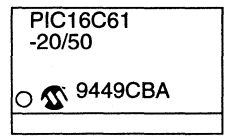
Example



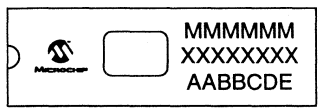
18-Lead SOIC



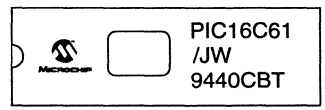
Example



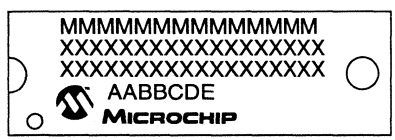
18-Lead CERDIP Windowed



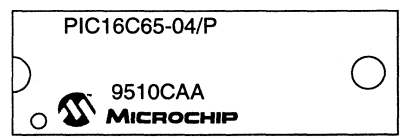
Example



40-Lead PDIP



Example



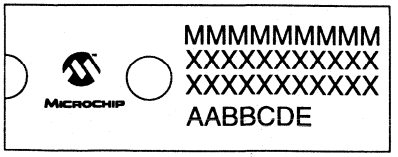
Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	D_2	Mask revision number for EEPROM
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

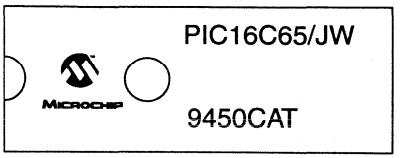
© 1995 Microchip Technology Inc.

Package Marking Information (Cont'd)

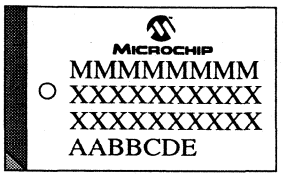
40-Lead CERDIP Windowed



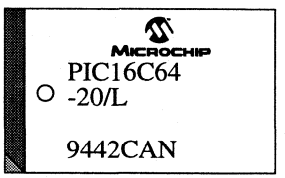
Example



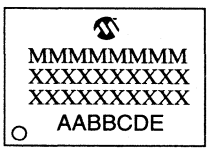
44-Lead PLCC



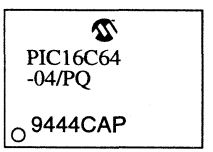
Example



44-Lead MQFP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

APPENDIX A:

The following are the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bit. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (192 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from status register.
3. Data memory paging is redefined slightly. Status register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5x.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers oscillator start-up timer (OST) and power-up timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PortB has weak pull-ups and interrupt on change feature.
13. RTCC pin is also a port pin (RA4/T0CKI) now.
14. FSR is made a full eight bit register.
15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: V_{DD}, V_{SS}, V_{PP}, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on-Reset (POR) status bit.
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S NEW

The format of this data sheet has been changed to be consistent with other product families. This ensures that important topics are covered across all PIC16/17 families. Here is an overview list of new features:

- Data Sheet Structure / Outline

APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This is so that control bits that do the same function, have the same name (regardless of processor family). Care must still be taken, since they may not be in the same special function register. The following shows the register and bit names that have been changed:

REGISTER NAME CHANGES – BIT NAME CHANGES

OLD NAME	NEW NAME
RTE	TOSE
RTS	TOCS
$\overline{\text{T1INSYNC}}$	$\overline{\text{T1SYNC}}$

© 1995 Microchip Technology Inc.

APPENDIX E: PIC16/17 MICROCONTROLLERS

TABLE E-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features			
	Maximum Frequency of Operation (MHz)	EPROM Program Memory (bytes)	RAM Data Memory (bytes)	Timer Module(s)	Capture PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454	TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

*Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchroncously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE E-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Data EEPROM (bytes)	Program Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE E-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72		TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

E.1 Pin Compatibility

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin (20 pin)
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

INDEX

2

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF EXAMPLES

LIST OF FIGURES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF TABLES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks, and SQTP is a service mark of Microchip Technology Incorporated.

ACCESS.bus is a trademark of the ACCESS.bus Industry Group (ABIG).

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16C6X Literature Number: DS30234A

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject? _____

5. What deletions from the data sheet could be made without affecting the overall usefullness? _____

6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

© 1995 Microchip Technology Inc.

PIC16C6X Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)		
Package:	L	=	PLCC
	P	=	PDIP
	SO	=	SOIC (Gull Wing, 300 mil body)
	PQ	=	MQFP (Metric PQFP)
	JW	=	Windowed CERDIP
Temperature Range:	-	=	0°C to +70°C (T for tape/reel)
	I	=	–40°C to +85°C (S for tape/reel)
	E	=	–40°C to +125°C
Frequency Range:	04	=	200 KHz (PIC16C6X-04)
	04	=	4 MHz
	20	=	20 MHz
Device:	PIC16C6X :VDD range 4.0V to 6.0V		
	PIC16C6XT:VDD range 4.0V to 6.0V (Tape and Reel)		
	PIC16LC6X:VDD range 3.0V to 6.0V		
	PIC16LC6XT:VDD range 3.0V to 6.0V(Tape and Reel)		

Examples:

a) PIC16C61 - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301
b) PIC16LC64 - 041/SO = Industrial temp., SOIC package, 4MHz, extended VDD limits
c) PIC16C65 - 10E/P = Automotive temp., PDIP package, 10 MHz, normal VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

PIC16C62

28-Pin EPROM-Based 8-Bit Microcontroller Product Brief

FEATURES

High-Performance RISC-like CPU

- Only 35 single word instructions to learn
- All single cycle instructions (200ns) except for program branches which are two-cycle
- Operating speed:
  - DC - 20 MHz clock input
  - DC - 200ns instruction cycle
- 2048 x 14 on-chip EPROM program memory
- 128 x 8 general purpose registers (SRAM)
- Interrupt capability
- 29 special function hardware registers
- Eight levels deep hardware stack
- Direct, indirect and relative addressing modes

Peripheral Features

- 22 I/O pins with individual direction control
- High current sink/source for direct LED drive
- One pin that can be configured as capture input, PWM output, or compare output
  - Capture is 16-bit, max resolution 12.5ns
  - Compare is 16-bit, max resolution 200ns
  - PWM resolution is 1- to 10-bit. Maximum PWM frequency @: 8-bit resolution = 80 KHz 10-bit resolution = 20 KHz
- TMR1: 16-bit timer/counter (time-base for capture/compare). TMR1 can be incremented during sleep via external crystal/clock (for real-time clock)
- TMR2: 8-bit timer/counter with 8-bit period register (time-base for PWM), prescaler and postscaler
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler
- Synchronous serial port (SSP) with two modes of operation:
  - 3-wire SPI
  - I^2C™/ACCESS.bus compatible

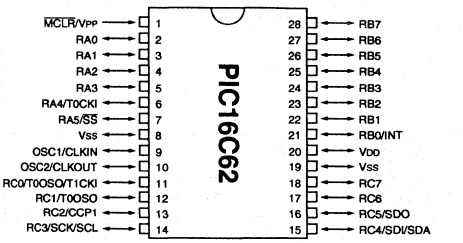
Special Microcontroller Features

- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- EPROM fuse selectable oscillator options
- Serial in-system programming (via two pins)

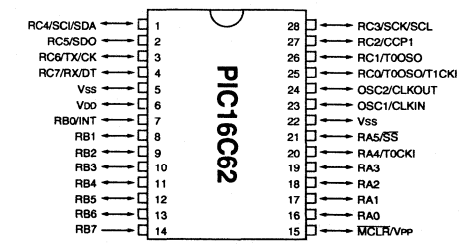
I^2C is a trademark of Philips Corporation.
ACCESS.bus is a trademark of ABIG.

PACKAGE TYPES

PDIP, SOIC, Windowed CERDIP



SSOP



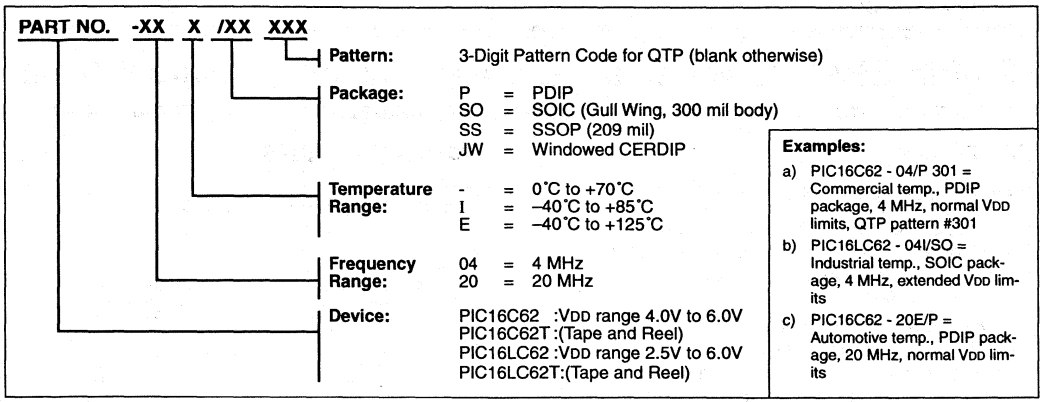
CMOS Technology

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range (2.5V to 6.0V)
- Commercial, Industrial and Automotive Temp. Range
- Low-power consumption
  - < 2mA @ 5V, 4 MHz
  - 15µA typical @ 3V, 32 KHz
  - <1µA typical standby current @ 3V

© 1994 Microchip Technology Inc.

PIC16C62 Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.



Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1994 Microchip Technology Inc.

MICROCHIP

PIC16C63

28-Pin EPROM-Based 8-Bit Microcontroller Product Brief

FEATURES

High-Performance RISC-like CPU

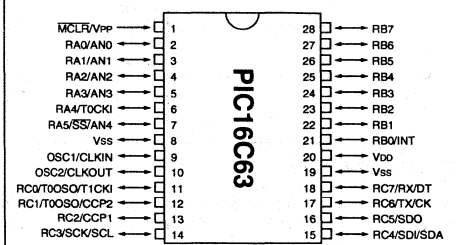
- Only 35 single word instructions to learn
- All single cycle instructions (200ns) except for program branches which are two-cycle
- Operating speed:
  - DC - 20 MHz clock input
  - DC - 200ns instruction cycle
- 4096 x 14 on-chip EPROM program memory
- 192 x 8 general purpose registers (SRAM)
- Interrupt capability
- 42 special function hardware registers
- Eight levels deep hardware stack
- Direct, indirect and relative addressing modes

Peripheral Features

- 22 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Two pins that can be configured as capture input, PWM output, or compare output
  - Capture is 16-bit, max resolution 12.5ns
  - Compare is 16-bit, max resolution 200ns
  - PWM resolution is 1- to 10-bit. Maximum PWM frequency @: 8-bit resolution = 80 KHz 10-bit resolution = 20 KHz
- TMR1: 16-bit timer/counter (time-base for capture/compare). TMR1 can be incremented during sleep via external crystal/clock (for real-time clock)
- TMR2: 8-bit timer/counter with 8-bit period register (time-base for PWM), prescaler and postscaler
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler
- Serial Communications Interface (SCI)
  - Full-duplex Asynchronous Communication or Half Duplex Synchronous Communication
- Synchronous serial port (SSP) with two modes of operation:
  - 3-wire SPI
  - I^2C™/ACCESS.bus compatible

PACKAGE TYPES

PDIP, SOIC, Windowed CERDIP



Special Microcontroller Features

- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- EPROM fuse selectable oscillator options
- Serial in-system programming (via two pins)

CMOS Technology

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range (3.0V to 6.0V)
- Commercial, Industrial and Automotive Temp. Range
- Low-power consumption
  - < 2mA @ 5V, 4 MHz
  - 15μA typical @ 3V, 32 KHz
  - <1μA typical standby current @ 3V

I^2C is a trademark of Philips Corporation.
ACCESS.bus is a trademark of ABIG.

© 1994 Microchip Technology Inc.

PIC16C62 Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)	
Package:	P = PDIP SO = SOIC (Gull Wing, 300 mil body) SS = SSOP (209 mil) JW = Windowed CERDIP	
Temperature Range:	- = 0°C to +70°C I = –40°C to +85°C E = –40°C to +125°C	
Frequency Range:	04 = 4 MHz 20 = 20 MHz	
Device:	PIC16C62 :VDD range 4.0V to 6.0V PIC16C62T :(Tape and Reel) PIC16LC62 :VDD range 2.5V to 6.0V PIC16LC62T:(Tape and Reel)	

Examples:

a) PIC16C62 - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301

b) PIC16LC62 - 04I/SO = Industrial temp., SOIC package, 4 MHz, extended VDD limits

c) PIC16C62 - 20E/P = Automotive temp., PDIP package, 20 MHz, normal VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1994 Microchip Technology Inc.

MICROCHIP

PIC16C7X

8-Bit CMOS Microcontrollers with Analog Converter

Devices included in this Data Sheet:

- PIC16C74
- PIC16C73
- PIC16C71

High-Performance RISC-like CPU

- Only 35 single word instructions to learn
- All single cycle instructions (200ns) except for program branches which are two-cycle
- Operating speed: DC - 20MHz clock input
  DC - 200ns instruction cycle

Device	Program Memory	Data Memory
PIC16C74/73	4K	192
PIC16C71	1K	36

- Interrupt capability
- Eight levels deep hardware stack
- Direct, indirect and relative addressing modes

Peripheral Features of all PIC16C7X

Device	I/O	A/D Channel	TMR0	High Current (sink/source)
PIC16C74	33	8	Yes	25 mA/25 mA
PIC16C73	22	5	Yes	25 mA/25 mA
PIC16C71	13	4	Yes	25 mA/25 mA

Peripheral Features of the PIC16C74/73 only

- Two pins that can be configured as capture input, PWM output, or compare output
  - Capture is 16-bit, max resolution 12.5ns
  - Compare is 16-bit, max resolution 200ns
  - High resolution PWM
- TMR1: 16-bit timer/counter (time-base for capture/compare). TMR1 can be incremented during sleep via external crystal/clock (for real-time clock)
- TMR2: 8-bit timer/counter with 8-bit period register (time-base for PWM), prescaler and postscaler
- Serial Communications Interface (SCI)/USART
- Synchronous serial port (SSP) with SPI and I^2C™/ACCESS.bus™

Peripheral Features of the PIC16C74 only

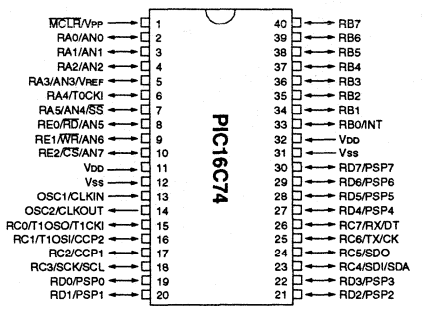
- Parallel Slave Port (PSP): 8-bit wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (microprocessor bus interface)

Special Microcontroller Features

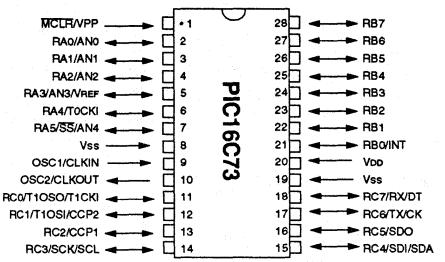
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Fuse selectable oscillator options
- Serial in-system programming (via two pins)

PACKAGE TYPES

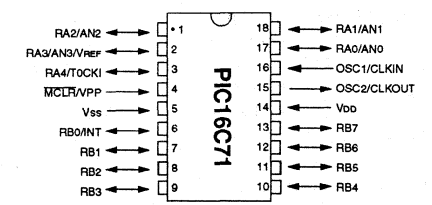
PDIP, Windowed CERDIP



PDIP, SOIC, Windowed CERDIP



PDIP, SOIC, Windowed CERDIP

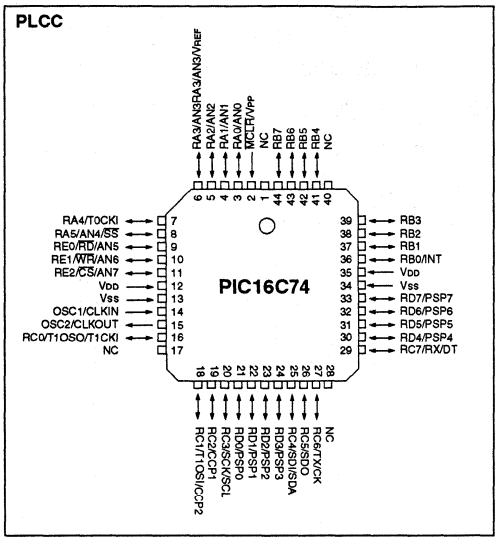


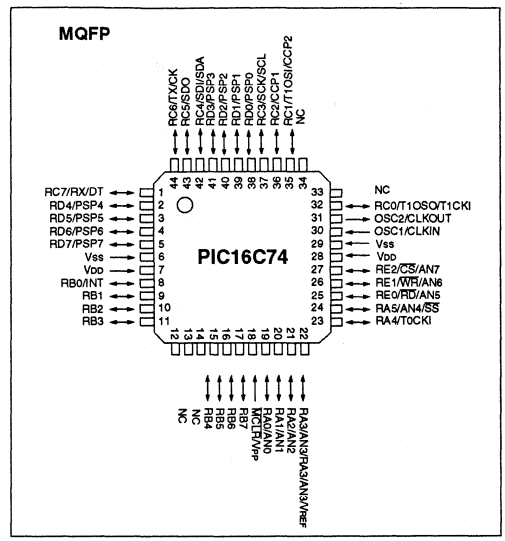
CMOS Technology

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range - (3.0V to 6.0V for PIC16C74/PIC16C73/PIC16C71))
- Commercial, Industrial, and Automotive Temp. Range
- Low-power consumption - < 2mA @ 5V, 4 MHz
  - 15mA typical @ 3V, 32 KHz
  - <1mA typical standby current

© 1995 Microchip Technology Inc.

PACKAGE TYPES





To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

TABLE OF CONTENTS

© 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The PIC16C7X is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC-like architecture. The PIC16CXX has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in its class.

The PIC16C74 device has 192 bytes of RAM and 33 I/O pins. In addition several peripheral features are available including: three timer/counters, two capture/compare/PWM modules and two serial ports. The synchronous serial port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I^2C^{TM}) bus. The Serial Communications Interface (SCI) can be configured as either synchronous or asynchronous (USART). An 8-bit Parallel Slave Port is provided. Also an 8-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The PIC16C73 device has 192 bytes of RAM and 22 I/O pins. In addition several peripheral features are available including: three timer/counters, two capture/compare/PWM modules and two serial ports. The synchronous serial port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit (I^2C) bus. The Serial Communications Interface (SCI) can be configured as either synchronous or asynchronous (USART). Also a 5-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The PIC167C71 device has 36 bytes of RAM and 13 I/O pins. In addition a timer/counter is available. Also a 4-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The PIC16C7X family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake up the chip from SLEEP through several external and internal interrupts and reset.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

Table 1-1 shows the features of the PIC16C7X family.

Table 1-2 shows information on which sections apply to the specific devices.

Simplified block diagrams of the PIC16C74, PIC16C73 and PIC16C71 are shown in Figure 3-1, Figure 3-2, and Figure 3-3 respectively.

The PIC16C7X family fits perfectly in applications ranging from security, remote sensors, appliance control to automotive. The EPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C7X very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, serial communication, capture and compare, PWM functions and co-processor applications).

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to PIC16CXX family of devices (see Appendix B).

1.2 Development Support

The PIC16CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Program Memory (bytes)	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	—	TMR0	—	—	—	—	4 ch	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	5 ch	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	8 ch	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

*Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE 1-2: PIC16C7X FUNCTION SECTIONS

FUNCTION	SECTION		
	PIC16C74	PIC16C73	PIC16C71
I/O Ports	5.0	5.0	5.0
PORTA	5.1	5.1	5.1
PORTB	5.2	5.2	5.2
PORTC	5.3	5.3	–
PORTD	5.4	–	–
PORTE	5.5	–	–
I/O Programming	5.6	5.6	5.6
Parallel Slave Port	5.7	–	–
Timer Module Overview	6.0	6.0	6.0
Timer0	7.0	7.0	7.0
Timer1	8.0	8.0	–
Timer2	9.0	9.0	–
Capture/Compare/PWM	10.0	10.0	–
Synchronous Serial Port (SPI/I^2C)	11.0	11.0	–
Serial Communications Interface (USART)	12.0	12.0	–
A/D Converter	13.0	13.0	13.0
CCP Special Trigger for A/D	13.8	13.8	–

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

2.0 PIC16C7X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements the proper device option can be selected using the information in the PIC16C7X Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC16C7X Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PICSTART® and PRO MATE™ programmers both support programming of the PIC16C7X. Third party programmers also are available, refer to the *Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory using the same bus. Separating program and data busses further allows instructions to be sized differently than 8-bit wide data word. Instruction op-codes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (see Example 3-1). Consequently, all instructions (35) execute in a single cycle (200ns @ 20MHz) except for program branches.

The PIC16C74 and PIC16C73 address 4K x 14 program memory and the PIC16C71 addresses 1K x 14 of program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16CXX device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

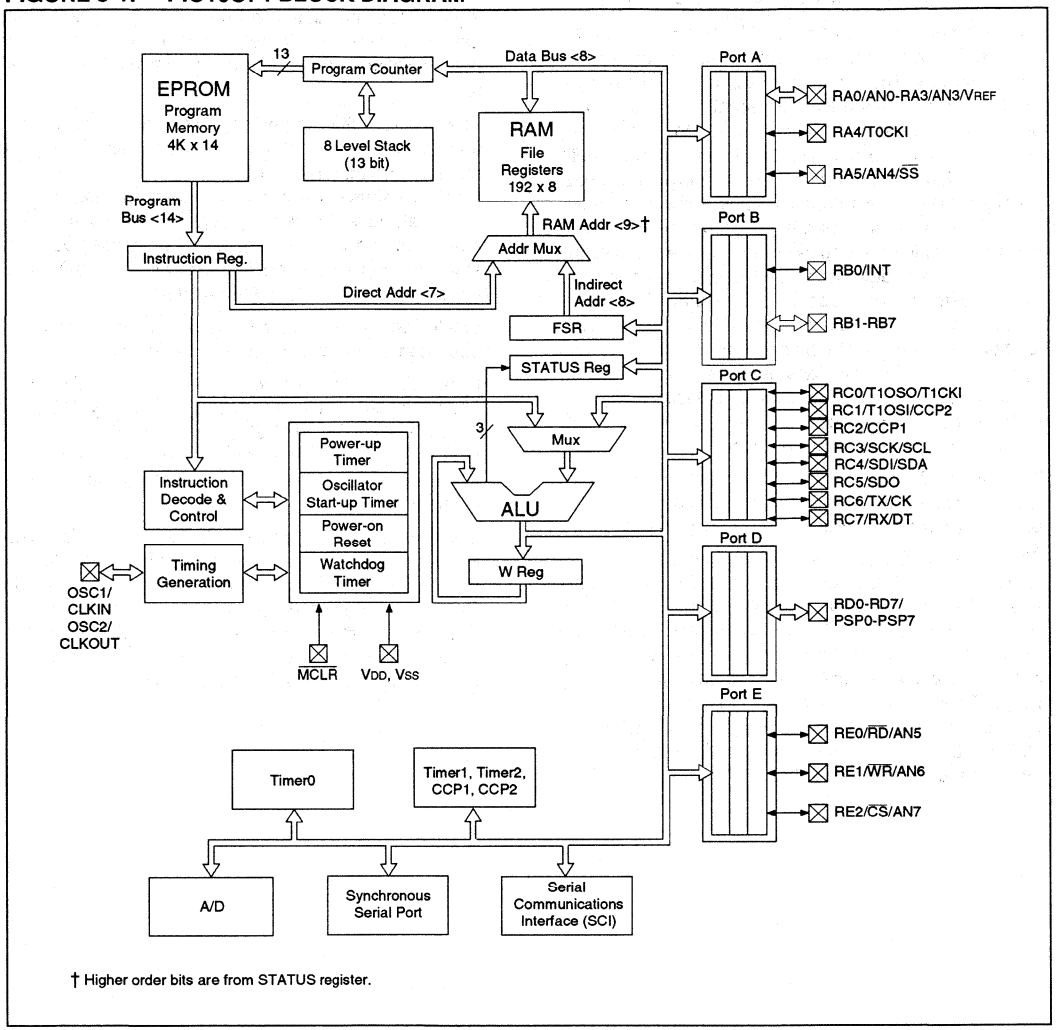
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram for the PIC16C74 is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1, a simplified block diagram for the PIC16C73 is shown in Figure 3-2, its corresponding pin description is shown in Table 3-1, and a simplified block diagram for the PIC16C71 is shown in Figure 3-3, its corresponding pin description is shown in Table 3-1.

© 1995 Microchip Technology Inc.

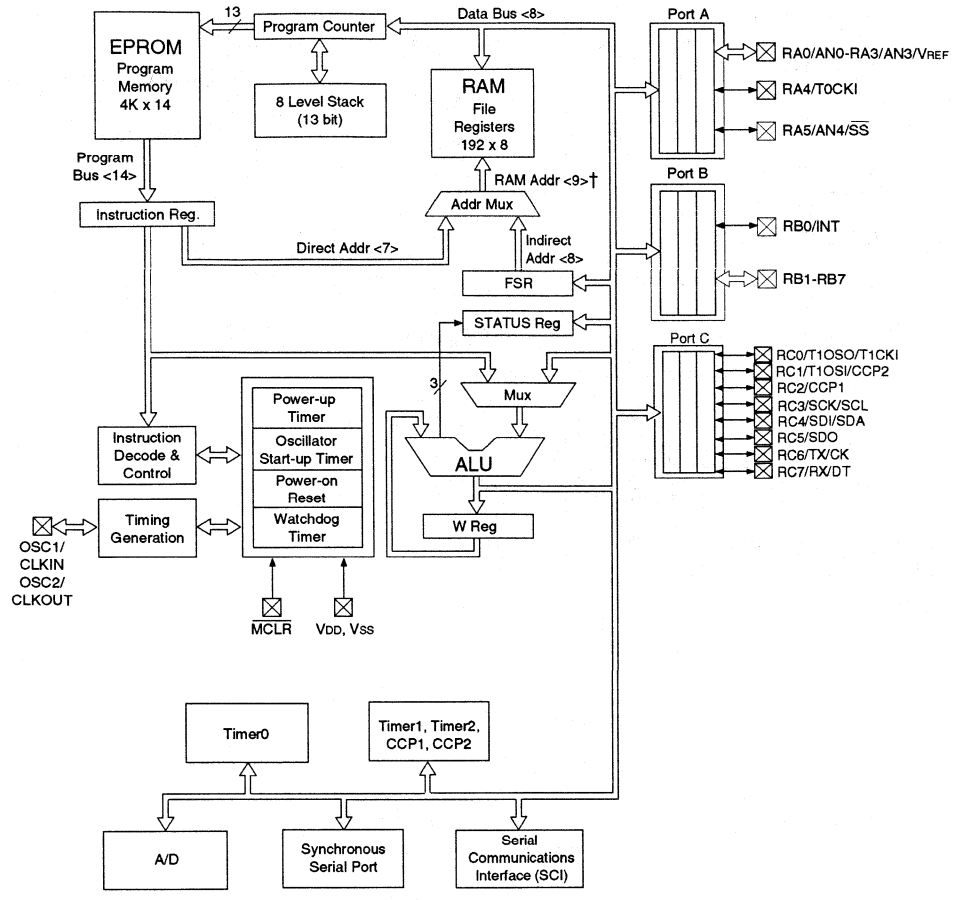
FIGURE 3-1: PIC16C74 BLOCK DIAGRAM



† Higher order bits are from STATUS register.

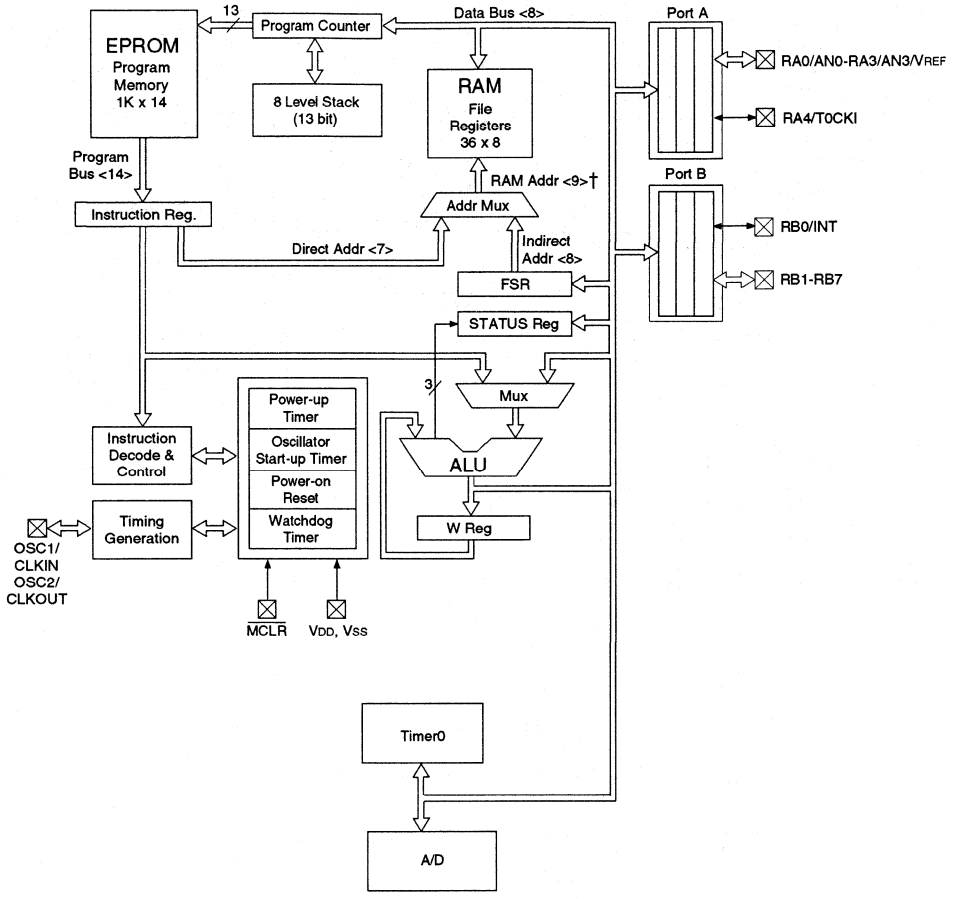
© 1995 Microchip Technology Inc.

FIGURE 3-2: PIC16C73 BLOCK DIAGRAM



† Higher order bits are from STATUS register.

FIGURE 3-3: PIC16C71 BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C74 PINOUT DESCRIPTION

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS[4]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
						PORTA is a bidirectional I/O port.
RA0/AN0	2	3	19	I/O	TTL	Analog input 0
RA1/AN1	3	4	20	I/O	TTL	Analog input 1
RA2/AN2	4	5	21	I/O	TTL	Analog input 2
RA3/AN3/VREF	5	6	22	I/O	TTL	Analog input 3/VREF
RA4/T0CKI	6	7	23	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
RA5/AN4/$\overline{SS}$	7	8	24	I/O	TTL	Analog input 4 can also be the slave select for the synchronous serial port.
						PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	Interrupt on change pin.
RB5	38	42	15	I/O	TTL	Interrupt on change pin.
RB6	39	43	16	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	40	44	17	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
						PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0/T1OSO/T1CKI can also be selected as a Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1/T1OSI/CCP2 can also be selected as a Timer1 oscillator input or Capture 2, input/Compare 2 output/PWM 2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2/CCP1 can also be selected as a capture1 input/compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5/SDO can also be selected as the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6/TX/CK can also be selected as Asynchronous Transmit or SCI Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7/RX/DT can also be selected as the Asynchronous Receive or SCI Synchronous Data.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C74 PINOUT DESCRIPTION (CONT.)

Pin Name	DIP No.	PLCC No.	MQFP No.	I/O/P Type	Buffer Type	Description
						PORTD is a bidirectional I/O port or parallel slave or interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL[3]	
RD1/PSP1	20	22	39	I/O	ST/TTL[3]	
RD2/PSP2	21	23	40	I/O	ST/TTL[3]	
RD3/PSP3	22	24	41	I/O	ST/TTL[3]	
RD4/PSP4	27	30	2	I/O	ST/TTL[3]	
RD5/PSP5	28	31	3	I/O	ST/TTL[3]	
RD6/PSP6	29	32	4	I/O	ST/TTL[3]	
RD7/PSP7	30	33	5	I/O	ST/TTL[3]	
						PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL[3]	RE0/$\overline{RD}$/AN5 read control for parallel slave port, or analog input 5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL[3]	RE1/$\overline{WR}$/AN6 write control for parallel slave port, or analog input 6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL[3]	RE2/$\overline{CS}$/AN7 select control for parallel slave port, or analog input 7.
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

TABLE 3-2: PIC16C73 PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS[3]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	1	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bidirectional I/O port.
RA0/AN0	2	2	I/O	TTL	Analog input 0
RA1/AN1	3	3	I/O	TTL	Analog input 1
RA2/AN2	4	4	I/O	TTL	Analog input 2
RA3/AN3/VREF	5	5	I/O	TTL	Analog input 3/VREF
RA4/TOCKI	6	6	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
RA5/AN4/$\overline{SS}$	7	7	I/O	TTL	Analog input 4 can also be the slave select for the synchronous serial port.
					PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	21	21	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	Interrupt on change pin.
RB5	26	26	I/O	TTL	Interrupt on change pin.
RB6	27	27	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	28	28	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
					PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	11	11	I/O	ST	RC0/T1OSO/T1CKI can also be selected as a Timer0 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	12	12	I/O	ST	RC1/T1OSI/CCP2 can also be selected as a Timer0 oscillator input or Capture 2, input/Compare 2 output/PWM 2 output.
RC2/CCP1	13	13	I/O	ST	RC2/CCP1 can also be selected as a capture1 input/compare1 output/PWM1 output.
RC3/SCK/SCL	14	14	I/O	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock input/output for both SPI and I^2C modes.
RC4/SDI/SDA	15	15	I/O	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	16	16	I/O	ST	RC5/SDO can also be selected as the SPI Data Out (SPI mode).
RC6/TX/CK	17	17	I/O	ST	RC6/TX/CK can also be selected as Asynchronous Transmit or SCI Synchronous Clock.
RC7/RX/DT	18	18	I/O	ST	RC7/RX/DT can also be selected as the Asynchronous Receive or SCI Synchronous Data.
VSS	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

TABLE 3-3: PIC16C71 PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS[3]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bidirectional I/O port.
RA0/AN0	17	17	I/O	TTL	Analog input 0
RA1/AN1	18	18	I/O	TTL	Analog input 1
RA2/AN2	1	1	I/O	TTL	Analog input 2
RA3/AN3/VREF	2	2	I/O	TTL	Analog input 3/VREF
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
					PORTB is a bidirectional I/O port. PortB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL/ST[1]	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
VSS	5	5	P	—	Ground reference for logic and I/O pins.
VDD	14	14	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-4.

3.2 Instruction Flow/Pipelining

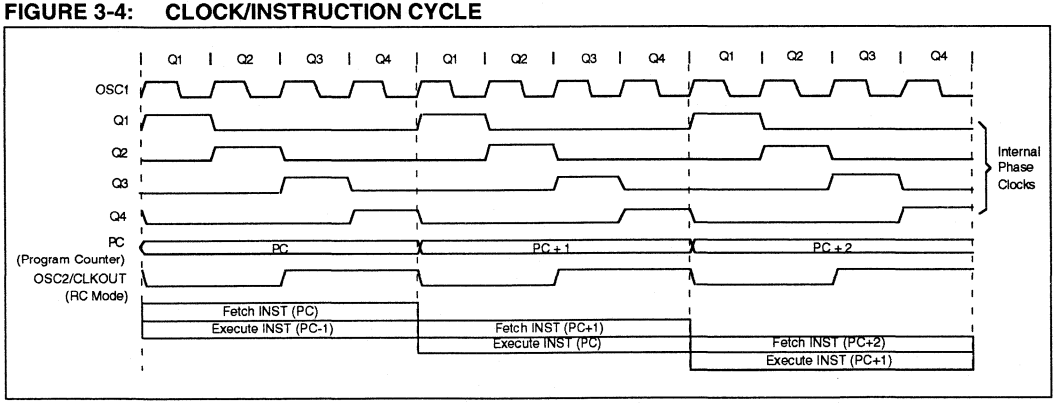
An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

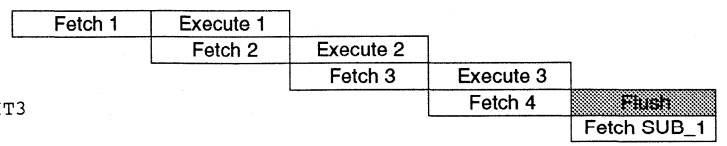
2

FIGURE 3-4: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

```
1. MOVLW   55h
2. MOVWF   PORTB
3. CALL    SUB_1
4. BSF     PORTA, BIT3
```



All instructions are single cycle, except for any program branches. These take two cycles since the fetched instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C7X family has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C74 and PIC16C73 only the first 4K x 14 (0000-0FFFh) are physically implemented, and for the PIC16C71 only the first 1K x 14 (0000-03FFh) is physically implemented. Accessing a location above the physically implemented address will cause a wrap-around. The reset vector is at 0000h and the interrupt vector is at 0004h (see Figure 4-1 and Figure 4-2).

FIGURE 4-1: PIC16C74/73 PROGRAM MEMORY MAP AND STACK

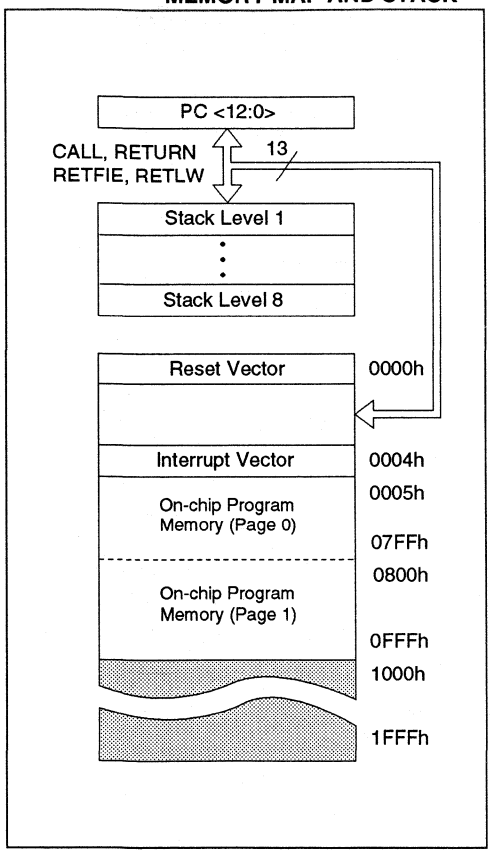
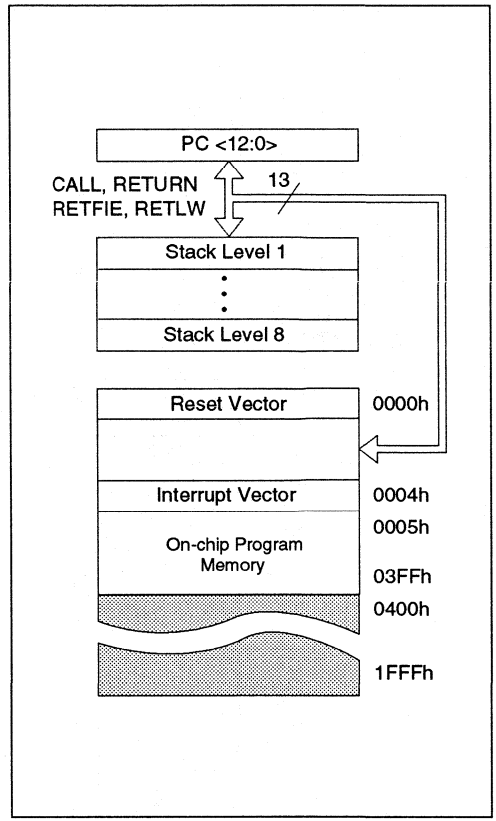


FIGURE 4-2: PIC16C71 PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory (see Figure 4-3 and Figure 4-4) is partitioned into two Banks which contain the general purpose registers and the special function registers. Bank 0 is selected when the RP0 bit in the STATUS register is cleared. Bank 1 is selected when the RP0 bit in the STATUS register is set. Each Bank extends up to 7Fh (128 bytes). The lower locations of each Bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers implemented as static RAM. Some Special Function Registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly, or indirectly through the file select register FSR (see Section 4.4).

The general purpose register locations 8Ch–AFh of Bank 1 on the PIC16C71 are not physically implemented. These locations are mapped into 0Ch–2Fh of Bank 0.

© 1995 Microchip Technology Inc.

FIGURE 4-3: PIC16C74/PIC16C73 REGISTER FILE MAP

File address	Bank 0	Bank 1	
00	INDF[1]	INDF[1]	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07	PORTC	TRISC	87
08	PORTD[2]	TRISD[2]	88
09	PORTE[2]	TRISE[2]	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C	PIR1	PIE1	8C
0D	PIR2	PIE2	8D
0E	TMR1L	PCON	8E
0F	TMR1H		8F
10	T1CON		90
11	TMR2		91
12	T2CON	PR2	92
13	SSPBUF	SSPADD	93
14	SSPCON	SSPSTAT	94
15	CCPR1L		95
16	CCPR1H		96
17	CCP1CON		97
18	RCSTA	TXSTA	98
19	TXREG	SPBRG	99
1A	RCREG		9A
1B	CCPR2L		9B
1C	CCPR2H		9C
1D	CCP2CON		9D
1E	ADRES		9E
1F	ADCON0	ADCON1	9F
20	General Purpose Register	General Purpose Register	A0
7F			FF

▨ Unimplemented data memory locations; read as '0's

Note 1: Not a physical register.
2: These registers are not physically implemented on the PIC16C73 and are read as '0'

FIGURE 4-4: PIC16C71 REGISTER FILE MAP

File Address	Bank 0	Bank 1	
00	INDF[1]	INDF[1]	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07			87
08	ADCON0	ADCON1	88
09	ADRES	ADRES	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C	General Purpose Register	Mapped in page 0	8C
2F			AF
30			B0
7F			FF

▨ Unimplemented data memory locations; reads as '0's

Note 1: Not a physical register.

© 1995 Microchip Technology Inc.

4.2.2 SPECIAL FUNCTION REGISTERS:

The Special Function Registers are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (see Table 4-2 and Table 4-2). These registers are static RAM.

The special registers can be classified into two sets. The special registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C74 AND PIC16C73

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
Bank 0											
00†	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01	TMR0	Timer0								xxxx xxxx	uuuu uuuu
02†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
03†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--xx xxxx	--uu uuuu
06	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08††	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	uuuu uuuu
09††	PORTE	PORTE Data Latch when written: PORTE pins when read								---- -xxx	---- -uuu
0A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC[1]					---0 0000	---0 0000
0B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0C	PIR1	PSPIF[3]	ADIF	RCIF††	TXIF††	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0D	PIR2	—	—	—	—	—	—	—	CCP2IF	---- ---0	---- ---0
0E	TMR1L	Timer1 Least Significant Byte								xxxx xxxx	uuuu uuuu
0F	TMR1H	Timer1 Most Significant Byte								xxxx xxxx	uuuu uuuu
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11	TMR2	Timer2								0000 0000	0000 0000
12	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15	CCPR1L	Capture/Compare/Duty Cycle Register (LSB)								xxxx xxxx	uuuu uuuu
16	CCPR1H	Capture/Compare/Duty Cycle Register (MSB)								xxxx xxxx	uuuu uuuu
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00x
19	TXREG	SCI Transmit Data Register								0000 0000	0000 0000
1A	RCREG	SCI Receive Data Register								0000 0000	0000 0000
1B	CCPR2L	Capture/Compare/Duty Cycle Register 2 (LSB)								xxxx xxxx	uuuu uuuu
1C	CCPR2H	Capture/Compare/Duty Cycle Register 2 (MSB)								xxxx xxxx	uuuu uuuu
1D	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	xx00 0000	xx00 0000
1E	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1F	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

††: These registers or bits are not physically implemented on the PIC16C73 and are read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

3: This bit is reserved for the PIC16C73.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C74 AND PIC16C73 (CONT.)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
Bank 1											
80†	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								–	–
81	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
83†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
86	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
88††	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89††	TRISE	IBF	OBF	IBOV	PSP-MODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111
8A†	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the PC[1]					---0 0000	---0 0000
8B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
8C	PIE1	PSPIE[3]	ADIE	RCIE††	TXIE††	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8D	PIE2	—	—	—	—	—	—	—	CCP2IE	---- ---0	---- ---0
8E	PCON	—	—	—	—	—	—	$\overline{POR}$	—	---- --0-	---- --u-
8F	—	Unimplemented								---- ----	---- ----
90	—	Unimplemented								---- ----	---- ----
91	—	Unimplemented								---- ----	---- ----
92	PR2	Timer2 Period Register								1111 1111	1111 1111
93	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	0000 0000
94	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	--00 0000	--00 0000
95	—	Unimplemented								---- ----	---- ----
96	—	Unimplemented								---- ----	---- ----
97	—	Unimplemented								---- ----	---- ----
98	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8	0000 -010	0000 -010
99	SPBRG	Baud Rate Register								0000 0000	0000 0000
9A	—	Unimplemented								---- ----	---- ----
9B	—	Unimplemented								---- ----	---- ----
9C	—	Unimplemented								---- ----	---- ----
9D	—	Unimplemented								---- ----	---- ----
9E	—	Unimplemented								---- ----	---- ----
9F	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	---- -000	---- -000

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

††: These registers or bits are not physically implemented on the PIC16C73 and are read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

3: This bit is reserved for the PIC16C73.

© 1995 Microchip Technology Inc.

TABLE 4-2: SPECIAL REGISTERS FOR THE PIC16C71

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note 2)
Bank 0											
00†	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01	TMR0	Timer0								xxxx xxxx	uuuu uuuu
02†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
03†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05	PORTA	---	---	---	PORTA Data Latch when written: PORTA pins when read					---x xxxx	---u uuuu
06	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07	---	Unimplemented								---- ----	---- ----
08	ADCON0	ADCS1	ADCS0	---	CHS1	CHS0	GO/$\overline{DONE}$	ADIF	ADON	00-0 0000	00-0 0000
09†	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
0A†	PCLATH	---	---	---	Write Buffer for the upper 5 bits of the PC[1]					---0 0000	---0 0000
0B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80†	INDF (indirect address)	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
81	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82†	PCL	Program Counter's (PC's) Least Significant Byte								0000 0000	0000 0000
83†	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84†	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85	TRISA	---	---	---	PORTA Data Direction Register					---1 1111	---1 1111
86	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
87	–	Unimplemented								---- ----	---- ----
88	ADCON1	–	–	–	–	–	–	PCFG1	PCFG0	---- --00	---- --00
89†	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
8A†	PCLATH	---	---	---	Write Buffer for the upper 5 bits of the PC[1]					---0 0000	---0 0000
8B†	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged, ? = value depends on condition, shaded locations are unimplemented and read as '0'

†: These registers can be addressed from either bank.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ or Watchdog Timer reset.

4.2.2.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

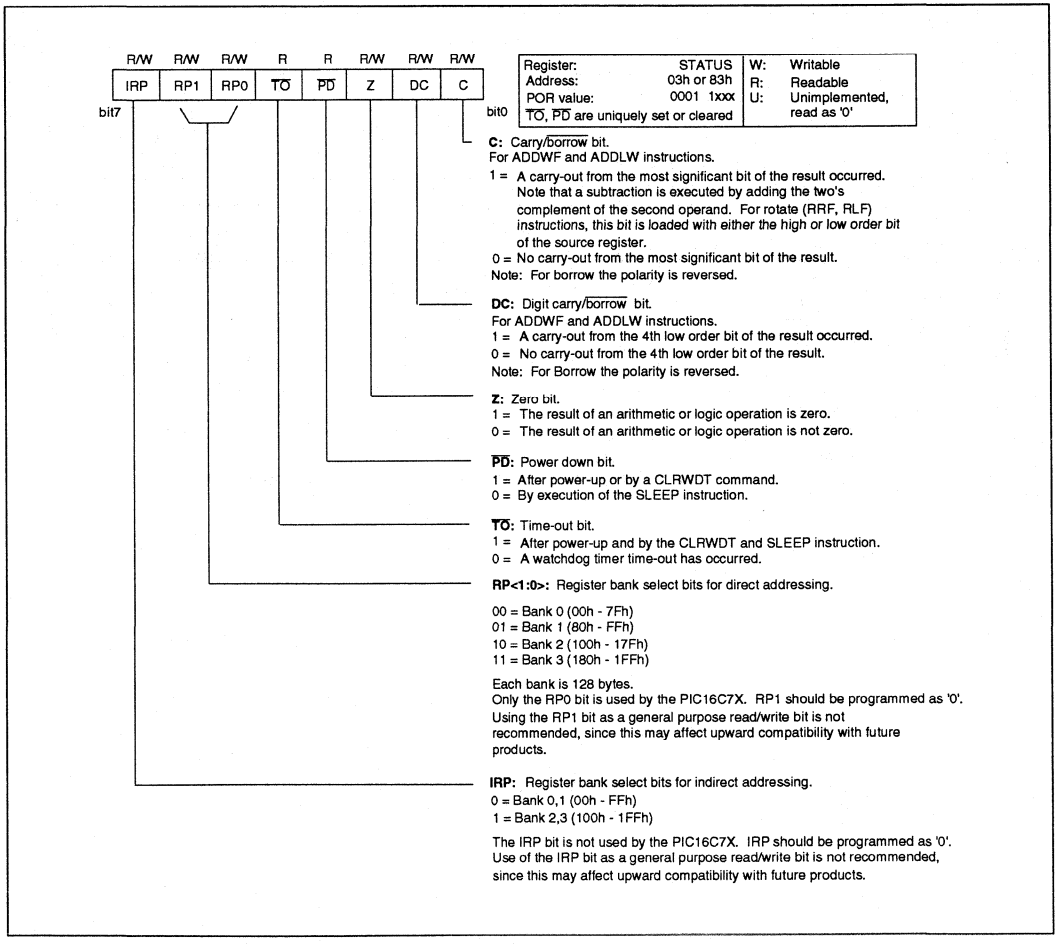
For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the status register as 000UU1UU (where U = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the status registers because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the "Instruction Set Summary

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16C7X and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

FIGURE 4-5: STATUS REGISTER



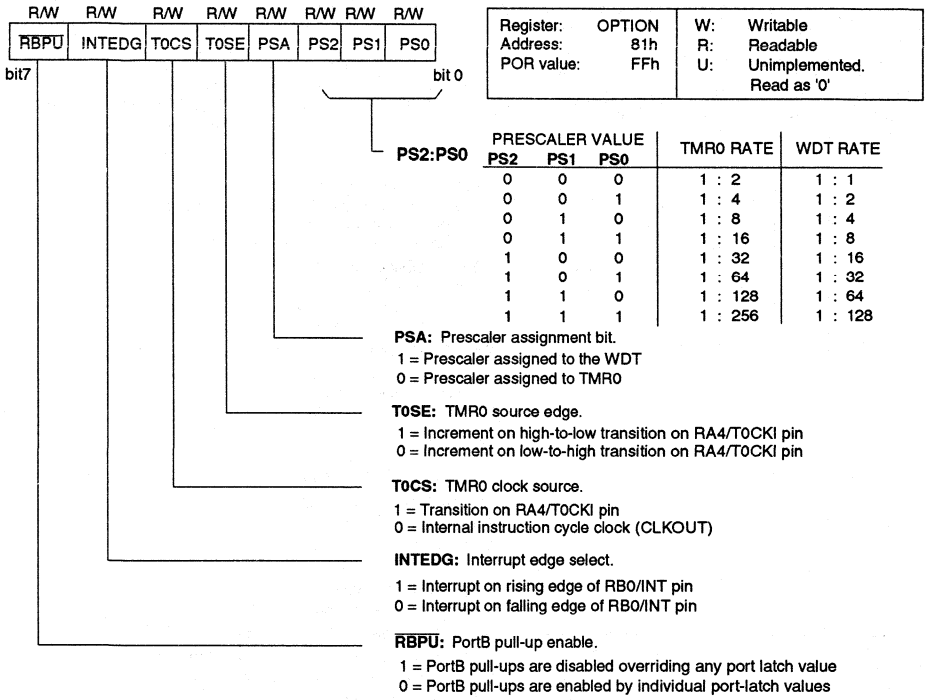
© 1995 Microchip Technology Inc.

4.2.2.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment, assign the prescaler to the WDT (PSA=1)

FIGURE 4-6: OPTION REGISTER



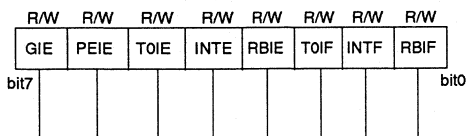
© 1995 Microchip Technology Inc.

4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains the various enable and flag bits for the Timer0 overflow, RB port change and external INT pin interrupts. Figure 4-7 shows the bits for the INTCON register for the PIC16C74 and PIC16C73. Figure 4-8 shows the INTCON register for the PIC16C71.

Note: The T0IF, INIF, or RBIF will be set by the specified condition even if the corresponding Interrupt Enable Bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled).

FIGURE 4-7: INTCON REGISTER FOR PIC16C74 AND PIC16C73 ONLY



Register:	INTCON	W:	Writable
Address:	0Bh or 8Bh	R:	Readable
POR value:	0000 000xb	U:	Unimplemented, read as '0'

RBIF: RB port change interrupt flag.
1 = When at least one of the RB<7:4> inputs change. Must be cleared by software.
0 = None of the RB<7:4> inputs have changed.

INTF: External interrupt flag.
1 = The external INT interrupt has occurred. Must be cleared by software.
0 = The INT interrupt did not occur

T0IF: TMR0 overflow interrupt flag.
1 = The TMR0 has overflowed. Must be cleared by software.
0 = TMR0 did not overflow.

RBIE: RB port change interrupt enable bit.
1 = Enables RBIF interrupt
0 = Disables RBIF interrupt

INTE: INT interrupt enable bit.
1 = Enables INTF interrupt
0 = Disables INTF interrupt

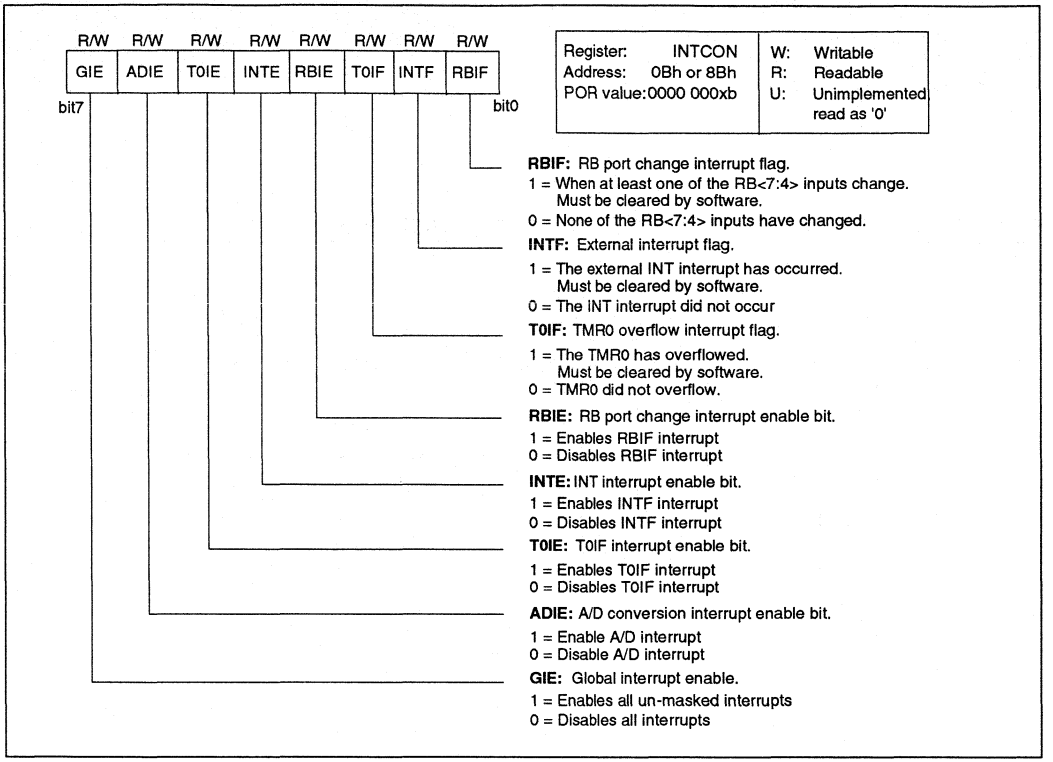
T0IE: T0IF interrupt enable bit.
1 = Enables T0IF interrupt
0 = Disables T0IF interrupt

PEIE: Peripheral interrupt enable bit.
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

GIE: Global interrupt enable.
1 = Enables all un-masked interrupts
0 = Disables all interrupts

© 1995 Microchip Technology Inc.

FIGURE 4-8: INTCON REGISTER FOR PIC16C71 ONLY



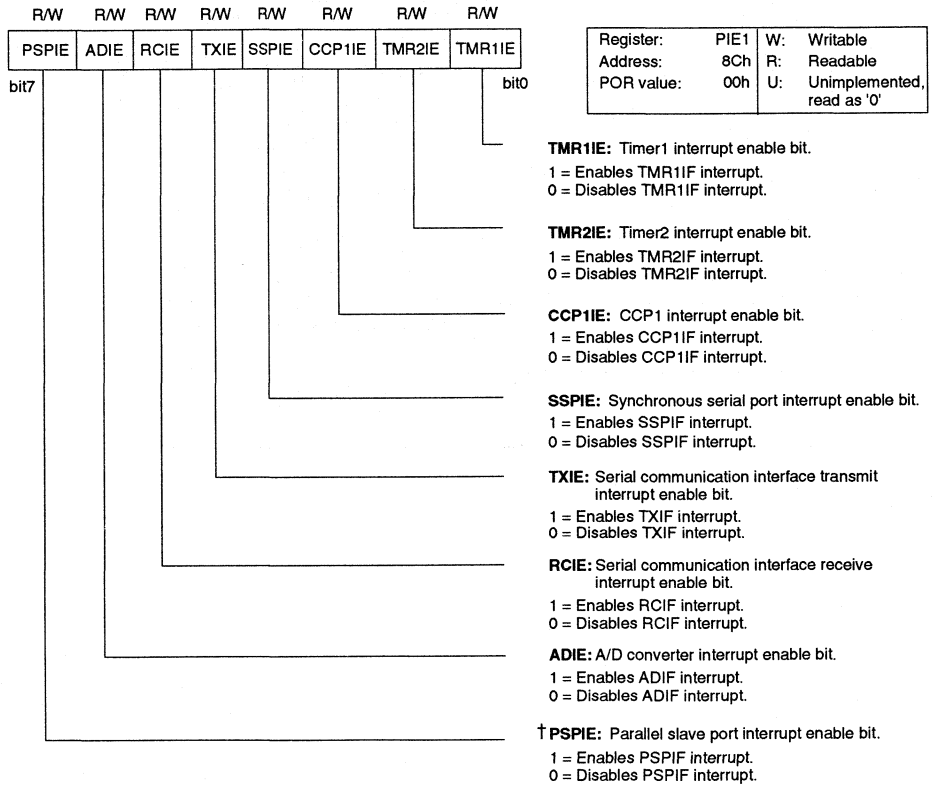
© 1995 Microchip Technology Inc.

4.2.2.4 PIE1 REGISTER

This register, implemented on the PIC16C74 and PIC16C73 only, contains the individual enable bits for the Peripheral interrupts (see Figure 4-9).

Note: INTCON<6> must be enabled to enable any interrupt in PIE1.

FIGURE 4-9: PIE1 REGISTER FOR PIC16C74 AND PIC16C73 ONLY



† This bit is reserved on the PIC16C73

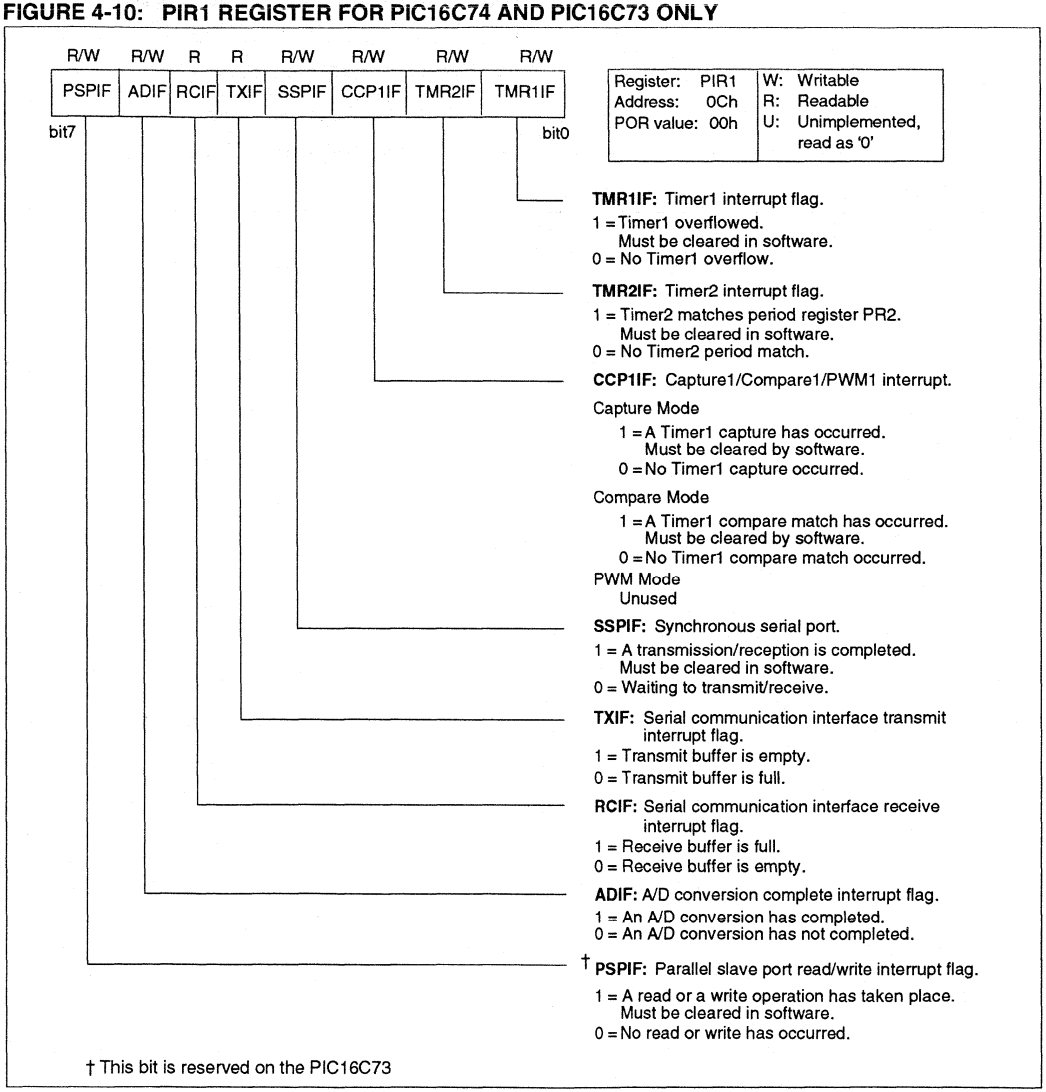
© 1995 Microchip Technology Inc.

4.2.2.5 PIR1 REGISTER

This register, implemented on the PIC16C74 and PIC16C73 only, contains the individual flag bits for the Peripheral interrupts (see Figure 4-10).

Note: These bits will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

FIGURE 4-10: PIR1 REGISTER FOR PIC16C74 AND PIC16C73 ONLY

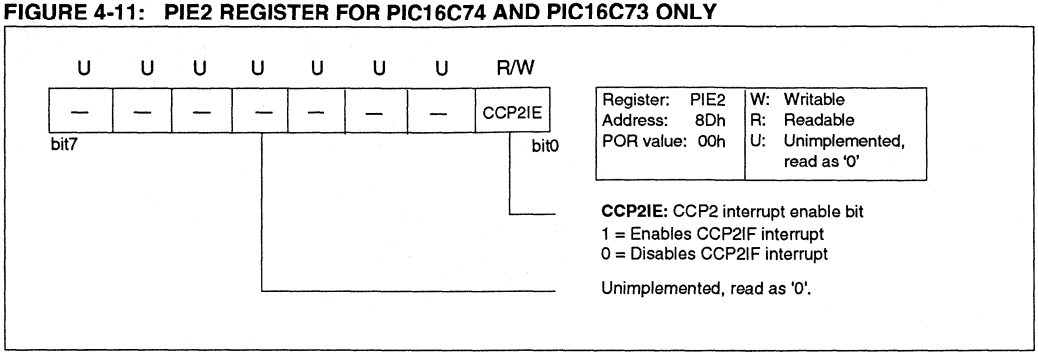


† This bit is reserved on the PIC16C73

4.2.2.6 PIE2 REGISTER

This register, implemented on the PIC16C74 and PIC16C73 only, contains the individual enable bit for the Peripheral Interrupts (see Figure 4-11).

FIGURE 4-11: PIE2 REGISTER FOR PIC16C74 AND PIC16C73 ONLY



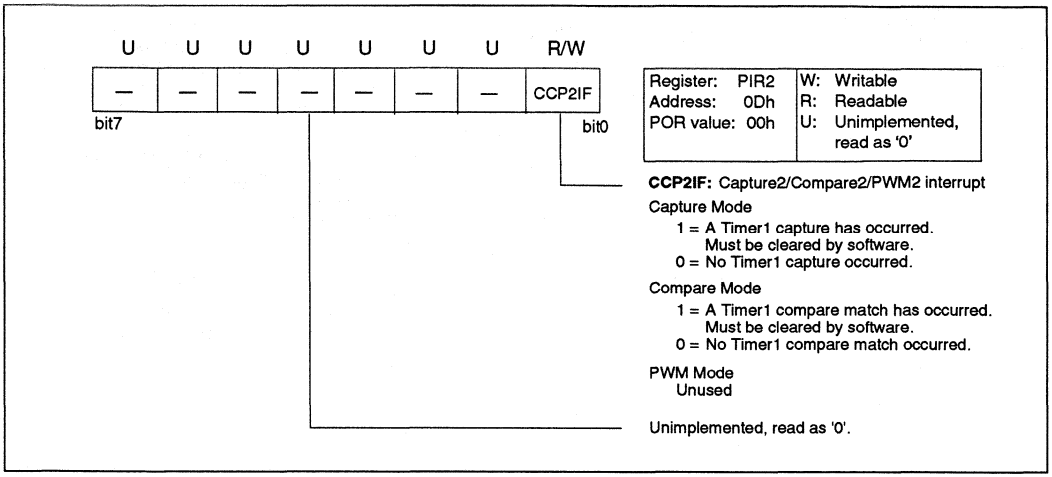
© 1995 Microchip Technology Inc.

4.2.2.7 PIR2 REGISTER

This register, implemented on the PIC16C74 and PIC16C73 only, contains the individual flag bit for the Peripheral interrupts (see Figure 4-12).

Note: This bit will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

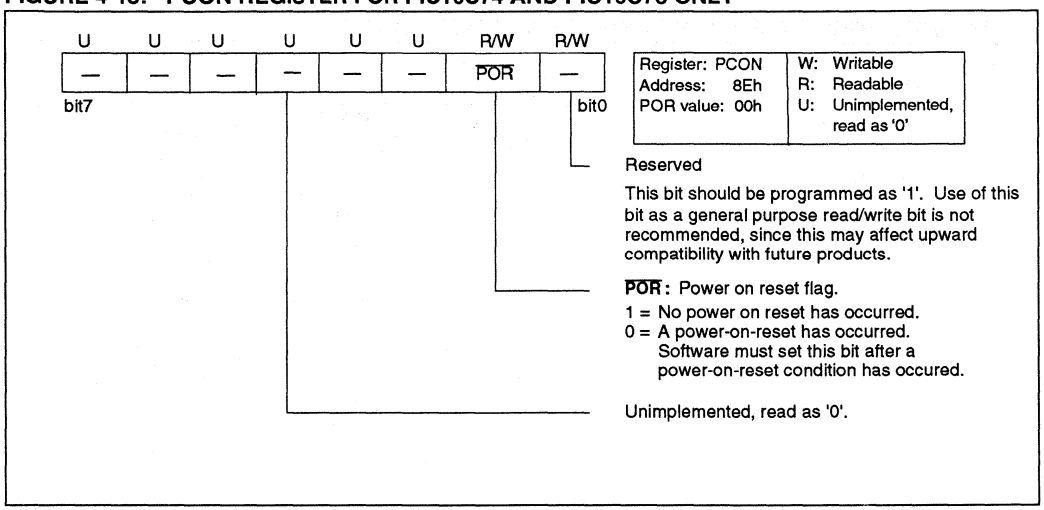
FIGURE 4-12: PIR2 REGISTER FOR PIC16C74 AND PIC16C73 ONLY



4.2.2.8 PCON REGISTER

The Power Control (PCON) register, implemented on the PIC16C74 and PIC16C73 only, contains a flag bit to allow differentiation between a Power-on Reset to an external $\overline{MCLR}$ reset or WDT reset (see Figure 4-13).

FIGURE 4-13: PCON REGISTER FOR PIC16C74 AND PIC16C73 ONLY

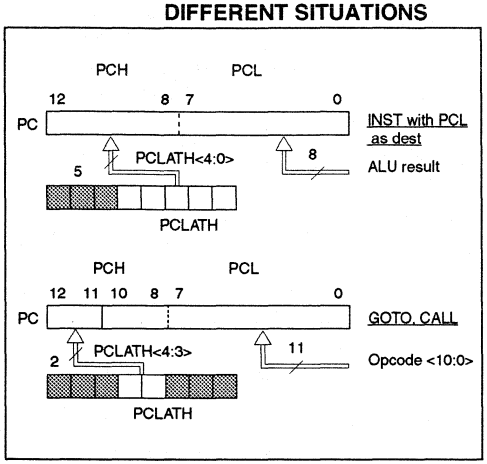


© 1995 Microchip Technology Inc.

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte, PCL, is a readable and writable register. The high byte of the PC (PCH) is not directly readable or writable. PCLATH is a holding register for PC<15:8> where contents are transferred to the upper byte of the program counter. When the PC is loaded with a new value during a CALL, GOTO or a write to PCL, the high bits of PC are loaded from PCLATH as shown in Figure 4-14.

FIGURE 4-14: LOADING OF PC IN DIFFERENT SITUATIONS



Note: On POR, the contents of the PCLATH register are unknown. The PCLATH should be initialized before a CALL, GOTO, or any instruction that modifies the PCL register is executed.

4.3.1 COMPUTED GOTO

When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Table Using the* PIC16CXX*"* (AN556).

4.3.2 STACK

The PIC16CXX has an 8 deep x 13-bit wide hardware stack (see Figure 4-1 and Figure 4-2). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed in the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is POPped in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer. This means that after the stack has been "PUSHed" eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflows or stack underflow conditions.

Note 2: There are no instructions mnemonics called PUSH nor POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address

4.3.3 PROGRAM MEMORY PAGING

The PIC16C74 and PIC16C73 have 4K of program memory, but the CALL and GOTO instructions only have a 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size. To allow CALL and GOTO instructions to address the entire 4K program memory address range, there must be another bit to specify the program memory page. This paging bit comes from the PCLATH<3> bit (see Figure 4-14). When doing a CALL or GOTO instruction, the user must ensure that this page bit (PCLATH<3>) is programmed to the desired program memory page. If a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> is not required for the return instructions (which pops the PC from the stack).

Note: The PIC16C7X ignores the PCLATH<4> bit, which is used for program memory pages 2 and 3 (1000h - 1FFFh). The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.

The PIC16C71 ignores PCLATH<3> bit, which is used for program memory page 1 (0800h-0FFFh). The use of PCLATH<3> as a general purpose read/write bit for the PIC16C71 is not recommended since this may affect upward compatibility with future products.

Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that the PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

© 1995 Microchip Technology Inc.

EXAMPLE 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
ORG 0X500
BSF     PCLATH, 3 ; Select page 1 (800h-FFFh)
CALL    SUB1_P1   ; Call subroutine in
        :         ; page 1 (800h-FFFh)
        :
        :
ORG     0X900
SUB1_P1 :         ; called subroutine
        :         ; page 1 (800h-FFFh)
        :
RETURN            ; return to page 0
                  ; (000h-7FFh)
```

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h.

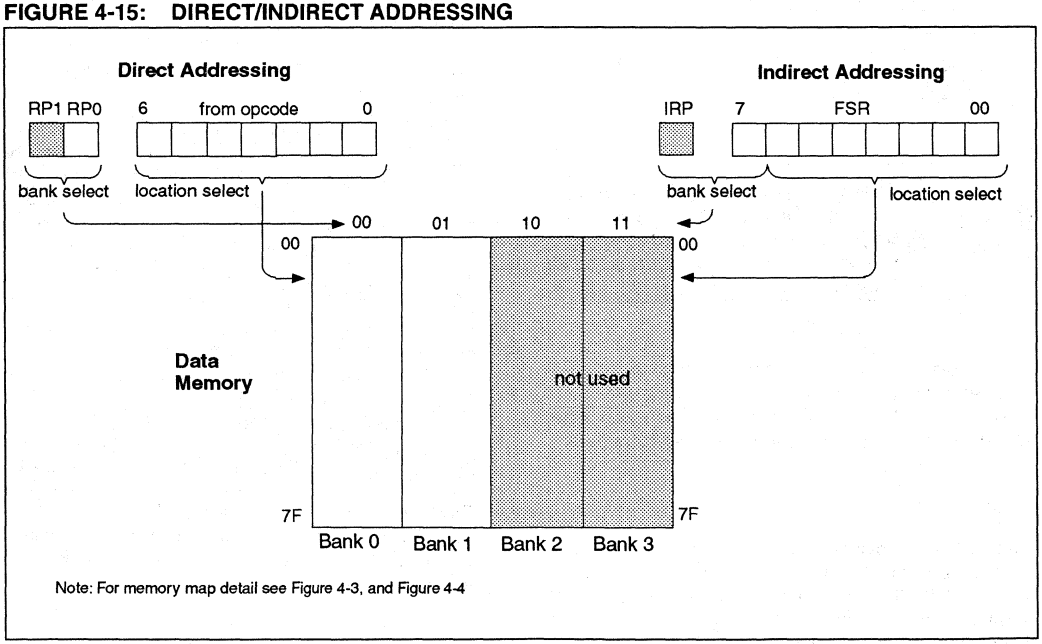
Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-15. However, IRP is not used in the PIC16C7X.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

```
        movlw   0x20    ;initialize pointer
        movf    FSR     ;to RAM
NEXT    clrf    INDF    ;clear INDF register
        inc     FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;no clear next
                        ;yes continue
CONTINUE:
```

FIGURE 4-15: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail see Figure 4-3, and Figure 4-4

5.0 I/O PORTS

The PIC16C74 has five ports, PORTA through PORTE, PIC16C73 has three ports, PORTA through PORTC, and PIC16C71 has two ports, PORTA and PORTB. These ports pins may be multiplexed with an alternate function for the peripheral features on the device.

5.1 PORTA and TRISA Registers

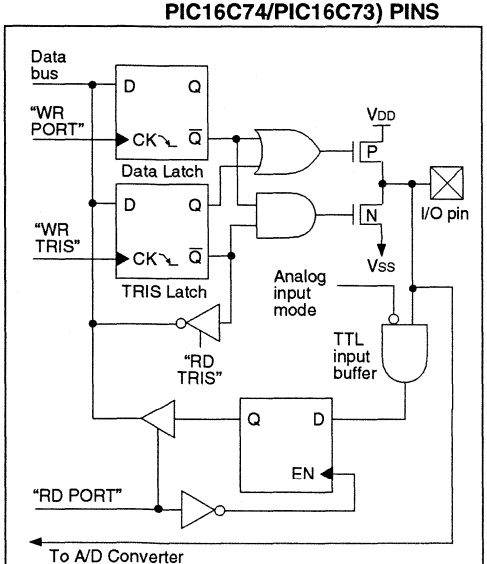
PORTA is a 6-bit wide latch for the PIC16C74 and PIC16C73 and is a 5-bit wide latch for the PIC16C71. RA4 is a Schmitt trigger input and an open collector output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

A '1' in the TRISA register puts the corresponding output driver in a high impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Port RA4 is multiplexed with TMR0 clock input.

FIGURE 5-1: BLOCK DIAGRAM OF RA<3:0> (AND RA<5> ON THE PIC16C74/PIC16C73) PINS



Note: I/O pins have protection diodes to VDD and VSS

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of these pins is selected by control bits in ADCON1 (A/D control register1). When selected as an analog input, these pins will read as '0's.

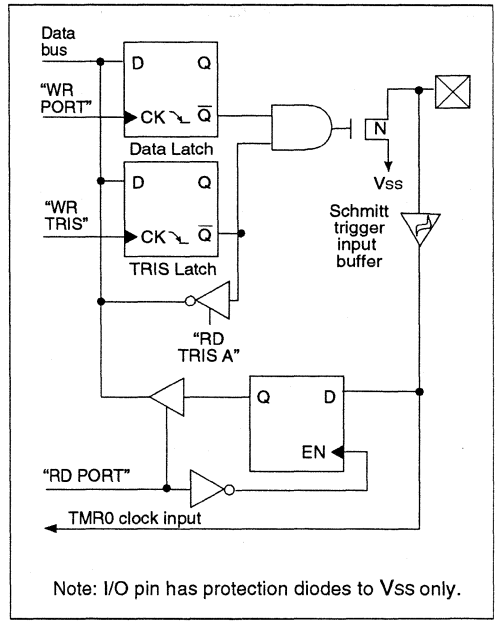
Note: On Power-on Reset, these pins are configured as analog inputs.

TRISA controls the direction of the RA pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF   PORTA        ;Initialize PORTA by setting
                    ; output data latches
BSF    STATUS, RP0  ;Select Bank1
MOVLW  0xCF         ;Value used to initialize
                    ;data direction
MOVWF  TRISA        ;Set RA<3:0> as inputs
                    ;RA<5:4> as outputs
                    ;TRISA<7:6> are always
                    ;read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN



Note: I/O pin has protection diodes to VSS only.

TABLE 5-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0/AN0	bit0	TTL	Input/output or analog input
RA1/AN1	bit1	TTL	Input/output or analog input
RA2/AN2	bit2	TTL	Input/output or analog input
RA3/AN3/VREF	bit3	TTL	Input/output or analog input/VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open collector type.
RA5/AN4/$\overline{SS}$ [1]	bit5	TTL	Input/output, slave select input for synchronous serial port, or analog input.

Legend: TTL = TTL input, ST = Schmitt trigger input

TABLE 5-2: SUMMARY OF PORTA REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTA	PORTA pins when read PORTA data latch when written	05h	--xx xxxx[1]
TRISA	PORTA data direction register 0 = output, 1 = input	85h	--11 1111[1]
ADCON1	PORTA Analog or Digital configuration	9Fh (74/73) 88h (71)	0000 0000

Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 14-7.
Note 1: The PIC16C71 does not have PORTA or TRISA bit 5, read as '0'.

© 1995 Microchip Technology Inc.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bidirectional port (file register address 06h). The corresponding data direction register is TRISB (address 86h). A '1' in the TRISB register puts the corresponding output driver in a high impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

EXAMPLE 5-2: INITIALIZING PORTB

```
CLRF    PORTB       ; Initialize PORTB data
                    ;   latches before setting
                    ;   the data direction
                    ;   register
BSF     STATUS, RPO ; Select Bank1
MOVLW   0xCF        ; Value used to initialize
                    ; data direction
MOVWF   TRISB       ; Set RB<3:0> as inputs
                    ;   RB<5:4> as outputs
                    ;   RBZZ<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on power-on reset.

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7–RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7–RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7–RB4 are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

This interrupt can wake the device up from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

a) Disable the interrupt by clearing RBIE (INTCON<3>) bit.

b) Read PORTB. This will end mismatch condition. Then, clear the RBIF bit.

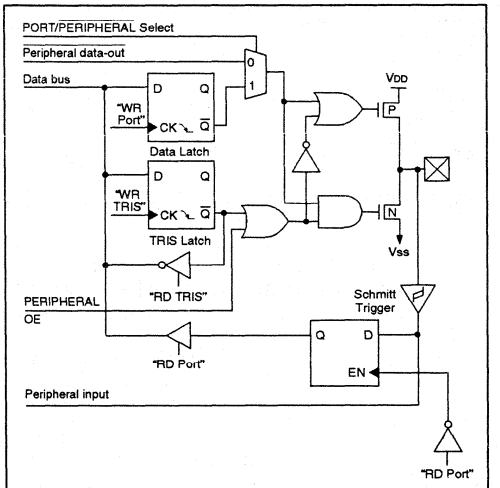
A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the *Embedded Control Handbook*).

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

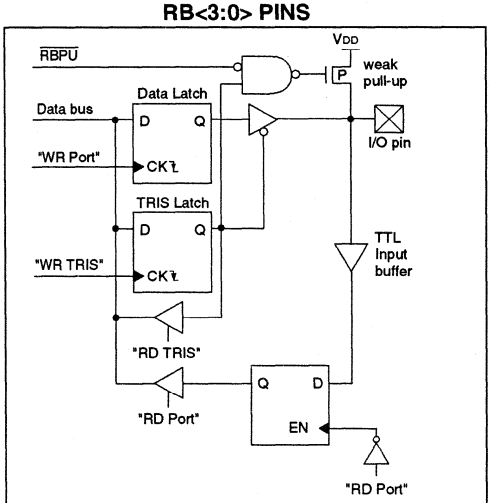
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-3: BLOCK DIAGRAM OF RB<7:4> PINS



Notes: 1. Port/Peripheral select signal selects between port data and peripheral output.
2. Peripheral OE (output enable) is only activated if peripheral select is active.
3. I/O pins have diode protection to VDD and VSS.

FIGURE 5-4: BLOCK DIAGRAM OF RB<3:0> PINS



Notes: 1. TRISB = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = 0 in OPTION register.
2. For RB0/INT pin, the INT input comes through a Schmitt trigger input buffer.
3. I/O pins have diode protection to VDD and VSS.

TABLE 5-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/INT	bit0	TTL/ST†	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger

† This buffer is a Schmitt trigger input when configured as the external interrupt.

‡ This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4: SUMMARY OF PORTB REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTB	PORTB pins when read PORTB data latch when written	06h	xxxx xxxx
TRISB	PORTB data direction register 0 = output, 1 = input	86h	1111 1111
OPTION	Weak pull-up on/off control ($\overline{RBPU}$ bit)	81h	1111 1111

Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 14-7.

© 1995 Microchip Technology Inc.

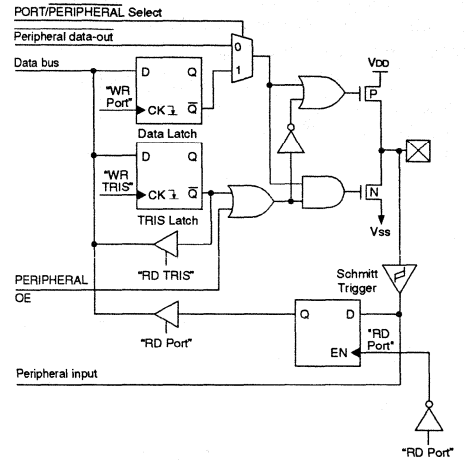
5.3 PORTC and TRISC Registers

PORTC is an 8-bit bidirectional port available in the PIC16C74 and PIC16C73 only. Each pin is individually configurable as input and output through the TRISC register. PORTC is multiplexed with several peripheral functions (see Table 5-5). PORTC pins have Schmitt trigger input buffers.

EXAMPLE 5-3: INITIALIZING PORTC

```
CLRF   PORTC        ; Initialize PORTC data
                    ;    latches before setting
                    ;    the data direction
                    ;    register
BSF    STATUS, RP0  ; Select Bank1
MOVLW  0xCF         ; Value used to initialize
                    ; data direction
MOVWF  TRISC        ; Set RC<3:0> as inputs
                    ;    RC<5:4> as outputs
                    ;    RC<7:6> as inputs
```

FIGURE 5-5: PORTC BLOCK DIAGRAM



Notes: 1. Port/Peripheral select signal selects between port data and peripheral output.
2. Peripheral OE (output enable) is only activated if peripheral select is active.
3. I/O pins have diode protection to VDD and Vss.

TABLE 5-5: PORTC FUNCTIONS

Name	Bit	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin, Timer1 oscillator input, Capture 2 input/Compare 2 output/PWM 2 output
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL	bit3	ST	RC3/SCK/SCL can also be selected as the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4/SDI/SDA can also be selected as the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous serial port data output
RC6/TX/CK	bit6	ST	Input/output port pin, SCI Asynchronous Transmit, or SCI Synchronous Clock
RC7/RX/DT	bit7	ST	Input/output port pin SCI Asynchronous Receive, or SCI Synchronous Data

Legend: ST = Schmitt Trigger Input

TABLE 5-6: SUMMARY OF PORTC REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTC	PORTC pins when read PORTC data latch when written	07h	xxxx xxxx
TRISC	PORTC data direction register 0 = output, 1 = input	87h	1111 1111

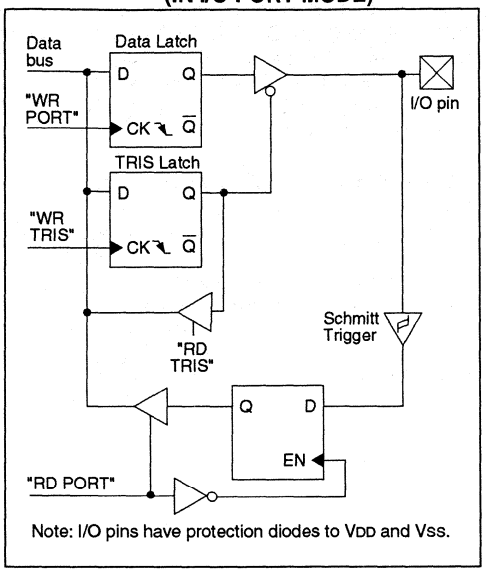
Legend: x = unknown, - = unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 14-7.

5.4 PORTD and TRISD Registers

PORTD is an 8-bit port with Schmitt trigger input buffers available in the PIC16C74 only. Each pin is individually configurable as input or output.

PORTD can be configured as an 8-bit wide microprocessor port (or parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

FIGURE 5-6: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and Vss.

TABLE 5-7: PORTD FUNCTIONS

Name	Bit	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL	Input/output port pin or parallel slave port bit 0
RD1/PSP1	bit1	ST/TTL	Input/output port pin or parallel slave port bit 1
RD2/PSP2	bit2	ST/TTL	Input/output port pin or parallel slave port bit 2
RD3/PSP3	bit3	ST/TTL	Input/output port pin or parallel slave port bit 3
RD4/PSP4	bit4	ST/TTL	Input/output port pin or parallel slave port bit 4
RD5/PSP5	bit5	ST/TTL	Input/output port pin or parallel slave port bit 5
RD6/PSP6	bit6	ST/TTL	Input/output port pin or parallel slave port bit 6
RD7/PSP7	bit7	ST/TTL	Input/output port pin or parallel slave port bit 7

Legend: ST = Schmitt Trigger Input when configured for general purpose I/O, TTL = TTL input when configured for Parallel Slave Port (PSP).

TABLE 5-8: SUMMARY OF PORTD REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTD	PORTD pins when read PORTD data latch when written	08h	xxxx xxxx
TRISD	PORTD data direction register 0 = output, 1 = input	88h	1111 1111

Legend: x = unknown, -= unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 14-7.

© 1995 Microchip Technology Inc.

5.5 PORTE and TRISE Register

PORTE is available on the PIC16C74 and has three pins RE0, RE1 and RE2, which are individually configurable as inputs or outputs. These have Schmitt trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when the PSPMODE bit (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs) and that ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

Figure 5-8 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. The operation of these pins is selected by control bits in ADCON1 (A/D register 1). When selected as an analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

Note: On power-on reset these pins are configured as analog inputs.

FIGURE 5-7: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)

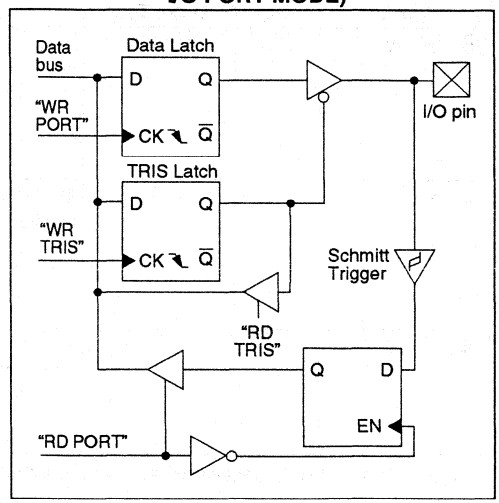


FIGURE 5-8: TRISE REGISTER

R	R	R/W	R/W	U	R/W	R/W	R/W
IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0

bit7 bit0

Register: TRISE	W: Writable
Address: 89h	R: Readable
POR value: 07h	U: Unimplemented, read as '0'

TRISE0: Direction control bit for port pin RE0
1 = Input
0 = Output

TRISE1: Direction control bit for port pin RE1
1 = Input
0 = Output

TRISE2: Direction control bit for port pin RE2
1 = Input
0 = Output

Unimplemented.
Read as '0'

PSPMODE: Selects parallel slave port mode for ports RD and RE.
1 = Parallel slave port mode.
0 = General purpose I/O.

IBOV: Input buffer overflow in microprocessor mode.
1 = A write occurred when a previous input word has not been read.
Must be cleared in software.
0 = No overflow has occurred.

OBF: Output buffer full.
1 = The output buffer still holds a previously written word.
0 = Output buffer has been read.

IBF: Input buffer full.
1 = A word has been received and waiting to be read by the CPU.
0 = No word has been received.

TABLE 5-9: PORTE FUNCTIONS

Name	Bit	Buffer Type	Function
RE0/$\overline{RD}$/AN5	bit0	ST/TTL	Input/output port pin, Read control input in parallel slaveport mode, or analog input $\overline{RD}$ 1 = Not a read operation 0 = Read operation. The system reads the PIC16C74 PORTD register (if chip selected)
RE1/$\overline{WR}$/AN6	bit1	ST/TTL	Input/output port pin, Write control input in parallel slaveport mode, or analog input $\overline{WR}$ 1 = Not a write operation 0 = Write operation. The system writes to the PIC16C74 PORTD register (if chip selected)
RE2/$\overline{CS}$/AN7	bit2	ST/TTL	Input/output port pin, Chip select control input in parallel slave port mode, or analog input $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger Input when configured for general purpose I/O, TTL = TTL input for Parallel Slave Port (PSP).

TABLE 5-10: SUMMARY OF PORTE REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTE	PORTE pins when read PORTE data latch when written	09h	---- -xxx
TRISE	PORTE data direction control bits and PORTD mode control	89h	0000 -111
ADCON1	PORTE Analog or Digital configuration	9Fh	0000 0000

Legend: x = unknown, -= unimplemented, read as a '0'. For reset values of registers in other reset situations refer to Table 14-7.

© 1995 Microchip Technology Inc.

5.6 I/O Programming Considerations

5.6.1 BIDIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit 5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit 5 and PORTB is written to the output latches. If another bit of PORTB is used as a bidirectional I/O pin (e.g., bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit 0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading the PORT register, reads the values of the PORT pins. Writing to the PORT register writes the value to the PORT latch. When using read modify write instructions (ex. BCF, BSF, etc.) on a PORT, the value of the PORT pins is read, the desired operation is done to this value, and this value is then written to the PORT latch.

Example 5-4 shows the effect of two sequential read modify write instructions (ex. BCF, BSF, etc.) on an I/O PORT.

EXAMPLE 5-4: READ MODIFY WRITE INSTRUCTIONS ON AN I/O PORT

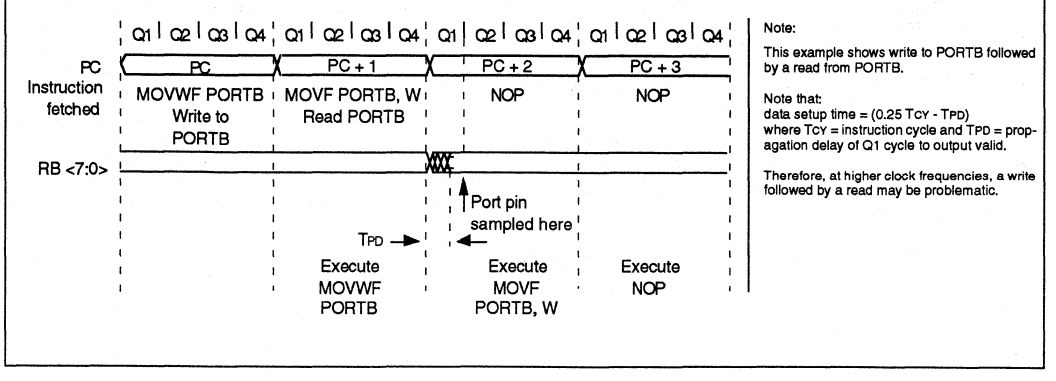
```
; Initial PORT settings:  PORTB<7:4> Inputs
;
;                         PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are not
; connected to other circuitry
;
;                         PORT latch  PORT pins
;                         ----------  ----------

   BCF PORTB, 7         ; 01pp pppp   11pp pppp
   BCF PORTB, 6         ; 10pp pppp   11pp pppp
   BSF STATUS,RP0       ;
   BCF TRISB, 7         ; 10pp pppp   11pp pppp
   BCF TRISB, 6         ; 10pp pppp   10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

5.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (see Figure 5-9). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-9: SUCCESSIVE I/O OPERATION



5.7 Parallel Slave Port

PORTD operates as an 8-bit wide parallel slave port, or microprocessor port when control bit PSPMODE (TRISE<4>) is set. In slave mode it is asynchronously readable and writable by the external world through $\overline{RD}$ control input (RE0/$\overline{RD}$) and $\overline{WR}$ control input (RE1/$\overline{WR}$).

It can directly interface to an 8-bit microprocessor data bus. The microprocessor can read or write the PORTD latch as an 8-bit latch. Setting PSPMODE enables the port pin RE0 to be the $\overline{RD}$ input, RE1 to be the $\overline{WR}$ input and RE2 to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set) and the A/D port configuration bits of the ADCON1 register (PCFG<2:0>) must be set such that RE<2:0> becomes digital I/O.

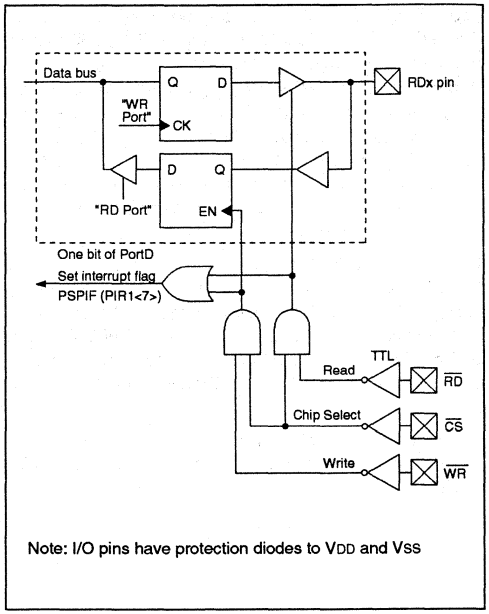
There are actually two 8-bit latches, one for data-out (from the PIC16/17) and one for data input. The user writes 8-bit data to PORTD data latch and reads data from the port pin latch (note that they have the same address). In this mode the TRISD register is ignored, since the microprocessor is controlling the direction of data flow.

Status flag IBF, Input Buffer Full (TRISE<7>), is set if a received word is waiting to be read by the CPU. Once the PORTD input latch is read by the PIC16C74, IBF is cleared. IBF is a read only status bit. Status flag OBF, Output Buffer Full (TRISE<6>), is set if a word written to PORTD latch is waiting to be read by the external bus. Once the PORTD output latch is read by the microprocessor, OBF is cleared. Status flag IBOV, Input Buffer Overflow (TRISE<5>), is set if a second word is written to the microprocessor port when the previous word has not been read by the CPU. It is a read/write bit and must be cleared by the CPU.

When not in PSPMODE, IBF and OBF bits are held clear. However, if the IBOV flag was previously set, it must be cleared in the software.

An interrupt is generated and latched into control bit PSPIF (PIR1<7>) when a read or a write operation is completed. The PSPIF interrupt flag must be cleared by the CPU and the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

FIGURE 5-10: PORTD AND PORTE AS A PARALLEL SLAVE PORT



Note: I/O pins have protection diodes to VDD and VSS

TABLE 5-11: SUMMARY OF PARALLEL SLAVE PORT REGISTERS

Register Name	Function	Address	Power-on Reset Value
PORTD	Parallel slave port Read/Write Data	08h	xxxx xxxx
PORTE	Parallel slave port Read/Write/Chip Select signals	09h	---- -xxx
TRISE	Control bits for PORTE peripheral	89h	0000 -111
PIR1	Interrupt register (PSPIF bit)	0Ch	0000 0000
PIE1	Interrupt Enable register (PSPIE bit)	8Ch	0000 0000

Legend: x = unknown, - = unimplemented, read as a '0'.

Note: For reset values of registers in other reset situations refer to Table 14-7.

© 1995 Microchip Technology Inc.

TABLE 5-12: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
08h	PORTD	PORTD							
09h	PORTE	PORTE							
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE 2	TRISE 1	TRISE 0
0Ch	PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1 IF	TMR2IF	TMR1IF
8Ch	PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1 IE	TMR2IE	TMR1IE
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0

Legend: – = unimplemented locations, read as a '0'.
Note: Shaded boxes are not used by Parallel Slave Port module.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

6.0 OVERVIEW OF TIMER MODULES

The PIC16C74 and PIC16C73 have three timer modules. The PIC16C71 has one timer module. Each module can generate an interrupt to indicate that an event has occurred (i.e. timer overflow). Each of these modules is explained in full detail in the following sections. The timer modules are:

- Timer0 (TMR0) module (see Section 7.0)
- Timer1 (TMR1) module (see Section 8.0)
- Timer2 (TMR2) module (see Section 9.0)

For enhanced time-based functionality, two additional modules can be used with either of the TMR1 or TMR2 modules. There are:

- Capture/Compare/PWM1 (CCP1) module (see Section 10.0)
- Capture/Compare/PWM2 (CCP2) module (see Section 10.0)

6.1 Timer0 (TMR0) Overview

The TMR0 module (previously known as RTCC) is a simple 8-bit overflow counter. The clock source can be either the internal system clock (OSC/4) or an external clock. When the clock source is an external clock, the TMR0 module can be selected to increment on either the rising or falling edge.

The TMR0 module also has a programmable prescaler option. This prescaler can be assigned to either the TMR0 module or the Watchdog timer. The PSA bit (OPTION<3>) assigns the prescaler, and the PS2 -PS0 (OPTION<2:0>) determines the prescaler value. The TMR0 can increment at the following rates: 1:1 (when prescaler assigned to Watchdog timer), 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

6.2 Timer1 (TMR1) Overview

Timer1 (TMR1) is a 16-bit timer/counter. The clock source can be either the internal system clock (OSC/4), an external clock, or an external crystal. TMR1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows TMR1 to operate during sleep, which is useful for applications that require a real time clock as well as the power savings of sleep mode.

TMR1 also has a prescaler option which allows the TMR1 to increment at the following rates: 1:1, 1:2, 1:4, 1:8. TMR1 can be used in conjunction with the Capture / Compare / PWM (CCP1 or CCP2) module. When used with the CCP1or CCP2 module, TMR1 is the time-base for 16-bit capture or the 16-bit compare. When using the TMR1 module with the CCP1 or CCP2 module, TMR1 must be synchronized to the device.

6.3 Timer2 (TMR2) Overview

Timer2 (TMR2) is an 8-bit timer. TMR2 has both a programmable prescaler and postscaler, as well as an 8-bit period register (PR2). TMR2 can be used with the CCP1 module as well as the baud rate generator for the Synchronous Serial Port (SSP). The prescaler option which allows the TMR2 to increment at the following rates: 1:1, 1:4, 1:16.

The postscaler allows TMR2 to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

6.4 CCP1 and CCP2 Overview

The CCP modules can operate in one of these three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM)

Capture mode, captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or the sixteenth rising edge of the CCPx pin.

Compare mode, compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs an interrupt can be generated, and the output pin CCP1 can be forced to given state (High or Low), TMR1 can be reset (CCP1), or TMR1 reset and start A/D conversion (CCP2). This depends on the control bits CCPxM3 - CCPxM0.

PWM mode, compares TMR2 to a 10-bit duty cycle register as well as to an 8-bit period register (PR2). When the TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high. When the TMR2 = Duty Cycle register, the CCPx pin will be forced low.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

7.0 TIMER0 (TMR0) MODULE

The TMR0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 7-1 is a simplified block diagram of the TMR0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two cycles (see Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 module.

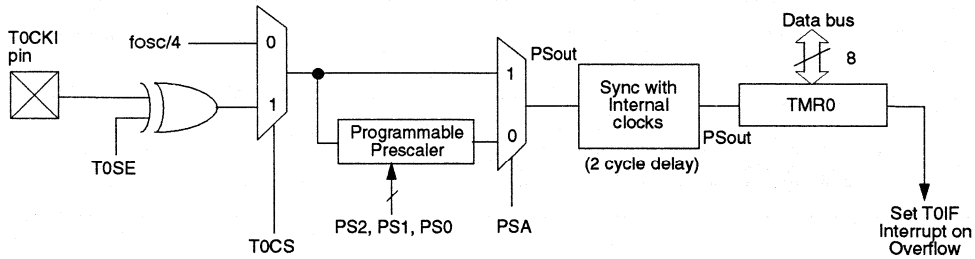
Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the T0 source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

The prescaler is shared between the TMR0 module and the watchdog timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is not readable or writable. When the prescaler is assigned to the TMR0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

7.1 TIMER0 (TMR0) Interrupt

TMR0 interrupt is generated when the TMR0 module timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the TMR0 module interrupt service routine before re-enabling this interrupt. The TMR0 module interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. See Figure 7-4 for TMR0 interrupt timing.

FIGURE 7-1: TIMER0 (TMR0) BLOCK DIAGRAM



Note 1: Bits, T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.
Note 2: The prescaler is shared with Watchdog Timer (refer to Figure 7-6 for detailed diagram)

FIGURE 7-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALE

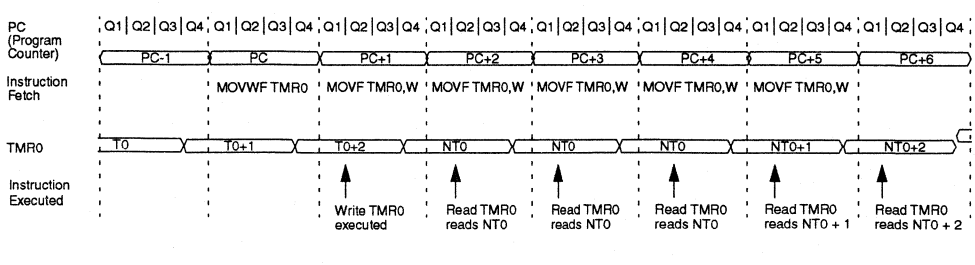


FIGURE 7-3: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/PRESCALE 1:2

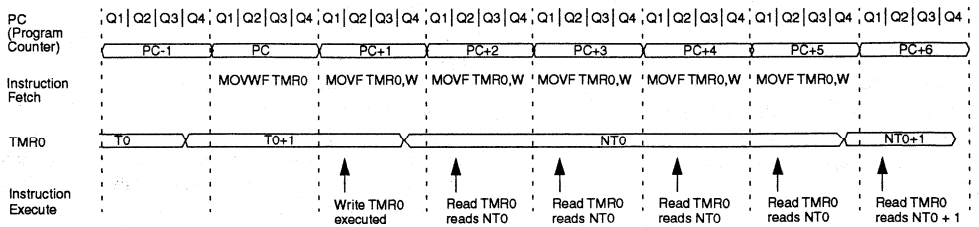
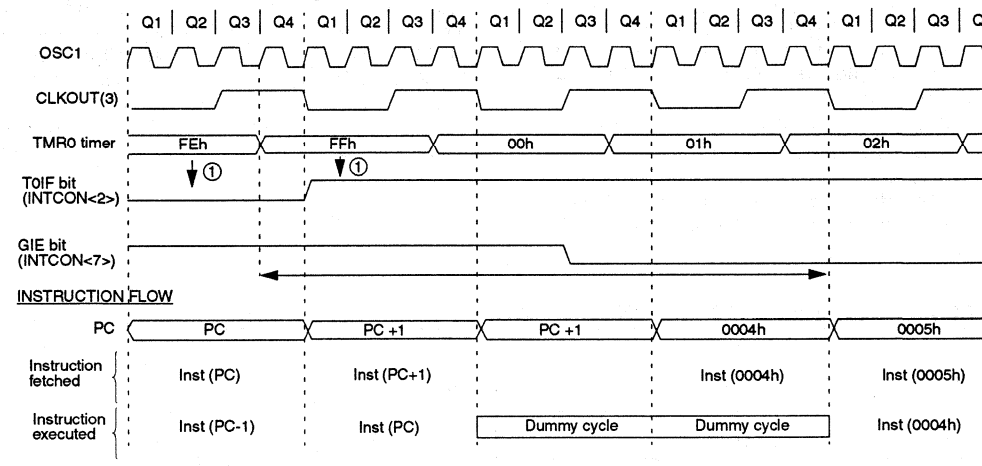


FIGURE 7-4: TIMER0 (TMR0) INTERRUPT TIMING



Note 1: T0IF interrupt flag is sampled here (every Q1).
2: Interrupt latency = 4Tcy where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

© 1995 Microchip Technology Inc.

7.2 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (see Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20ns) and low for at least 2Tosc (and a small RC delay of 20ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 Tosc (and a small RC delay of 40ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

7.3 Prescaler

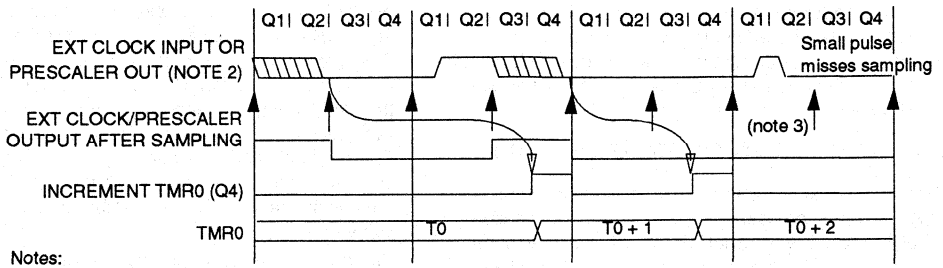
An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer, respectively (see Figure 7-6). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2-PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g. `CLRF 1, MOVWF 1, BSF 1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

© 1995 Microchip Technology Inc.

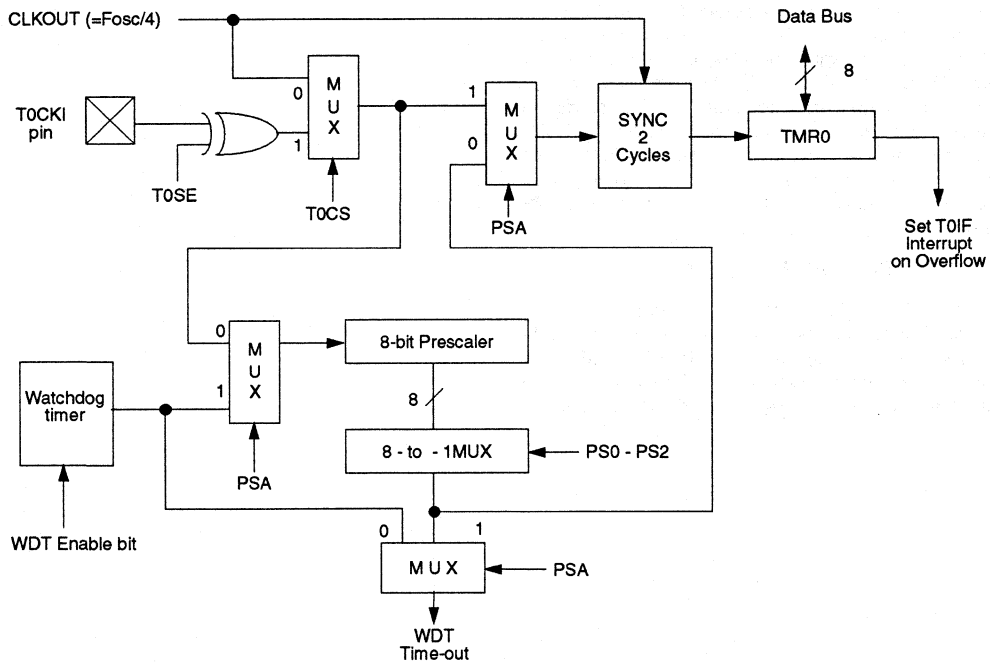
FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Notes:

1. Delay from clock input change to TMR0 increment is 3 tosc to 7 tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 tosc max.
2. External clock if no prescaler selected, Prescaler output otherwise.
3. The arrows indicate the points in time where sampling occurs.

FIGURE 7-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

© 1995 Microchip Technology Inc.

7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from TMR0 to WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TMR0→WDT)

```
BCF     STATUS,RP0    ;Bank 0
CLRF    TMR0          ;Clear TMR0 & Prescaler
BSF     STATUS, RP0   ;Bank 1
CLRWDT                ;Clears WDT
MOVLW   B'xxxx1xxx'   ;Select new prescaler
MOVWF   OPTION        ;value
BCF     STATUS, RP0   ;Bank 0
```

To change prescaler from the WDT to the TMR0 module use the sequence shown in Example 7-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                ;Clear WDT and
                      ;prescaler
BSF     STATUS, RP0
MOVLW   B'xxxx0xxx'   ;Select TMR0, new
                      ;prescale value and
                      ;clock source
MOVWF   OPTION
BCF     STATUS, RP0
```

2

TABLE 7-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. See Figure 7-5.	81h	1111 1111
INTCON	TMR0 overflow interrupt flag and mask bits. See Figure 7-6	0Bh	0000 000x

Legend: x = unknown, - = unimplemented, reads as a '0'.

Note: For reset values of registers in other reset situations refer to Table 14-8.

TABLE 7-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	TMR0	TMR0							
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
85h	TRISA	—	—	TRISA 5	TRISA 4	TRISA 3	TRISA 2	TRISA 1	TRISA 0

Legend: — = Unimplemented locations, Read as '0'

Note: Shaded boxes are not used by TMR0 module

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

8.0 TIMER1 (TMR1) MODULE

TMR1 is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L) which are readable and writable. TMR1 increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled or disabled using the TMR1 interrupt enable bit TMR1IE (PIE1<0>).

TMR1 can operate in one of two modes:

- As a timer
- As a counter

This is determined by the clock select bit, TMR1CS (T1CON<1>).

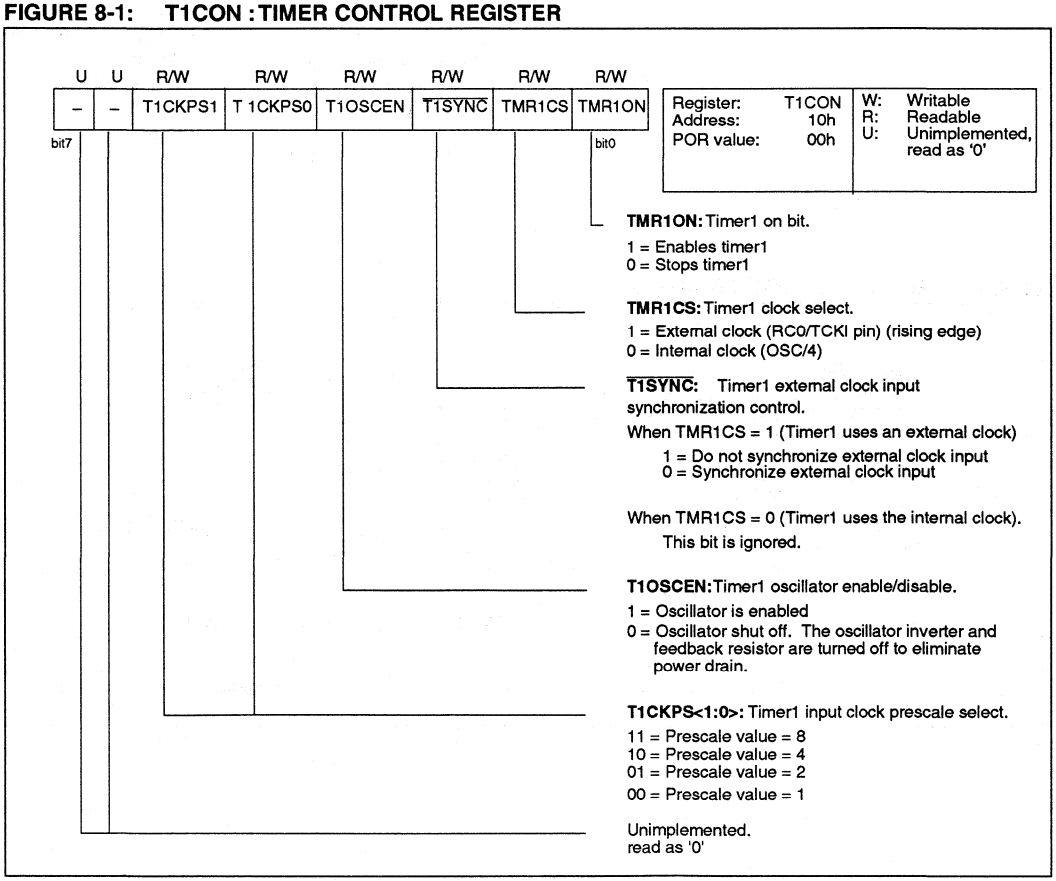
In timer mode, TMR1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input on RC0/T1OSO/T1CKI.

TMR1 can be turned on or off using the control bit TMR1ON (T1CON<0>).

TMR1 also has an internal "reset input". This reset can be generated by CCP1 or CCP2 (Capture/compare/PWM) module. See Section 10.059 for details. Figure 8-1 shows the Timer1 control register.

When the TMR1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 pin becomes an input. That is, the TRISC<1> value is ignored. The RC0/T1OSO/T1CKI pin should normally be configured as an input (for external clock). However, this pin can be configured as an output if self-clocking (through the output pin) is desired.

FIGURE 8-1: T1CON : TIMER CONTROL REGISTER



© 1995 Microchip Technology Inc.

8.1 TMR1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is OSC/4. The synchronize control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) has no effect since the internal clock is always in sync.

8.2 TMR1 Operation in Synchronized Counter Mode

Counter mode is selected by setting the TMR1CS bit. In this mode the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2 (when T1OSCEN is set) or the RC0/T1OSO/T1CKI.

If $\overline{\text{T1SYNC}}$ is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during SLEEP mode, TMR1 will not increment even if external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

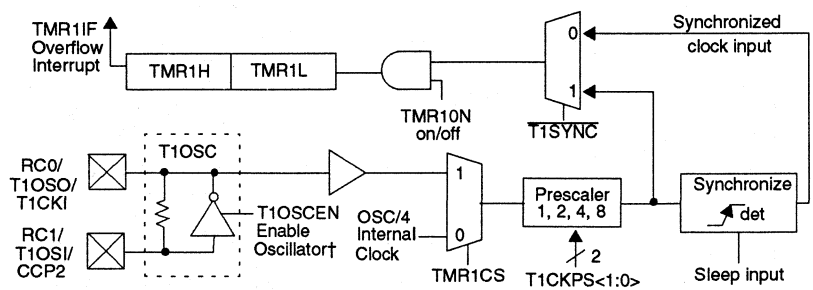
8.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for TMR1 in synchronized counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T1CKI to be high for at least 2Tosc (and a small RC delay of 20ns) and low for at least 2Tosc (and a small RC delay of 20ns). Refer to Figure 17-5, Figure 18-5 and Figure 19-5, parameters 45 and 46.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple counter-type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least 4Tosc (and a small RC delay of 40ns) divided by the prescaler value. The only requirement on T1CKI high and low time is that they do not violate the minimum pulse width requirements of 10ns). Refer to Figure 17-5, parameters 45, 46, and 47, and Figure 19-5, 40, and 42.

FIGURE 8-2: TMR1 BLOCK DIAGRAM



† When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.

© 1995 Microchip Technology Inc.

8.3 TMR1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{T1SYNC}}$ is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and generate an interrupt on overflow which will wake up the processor. However, special precautions in software are needed to read/write the timer (see Section 8.3.2).

In asynchronous counter mode, Timer1 can not be used as timebase for capture or compare operations.

8.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet a certain minimum high time and low time requirements, as specified in timing parameter.

8.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will guarantee a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Example 8-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

EXAMPLE 8-1: READING A 16-BIT FREE-RUNNING TIMER

```
;  All Interrupts are disabled
    MOVF    TMR1H, W        ;Read high byte
    MOVWF   TMPH            ;
    MOVF    TMR1L, W        ;Read low byte
    MOVWF   TMPL            ;
    MOVF    TMR1H, W        ;Read high byte
    SUBWF   TMPH,  W        ;Sub 1st read
                            ;with 2nd read
    BTFSC   STATUS,Z        ;is result = 0
    GOTO    CONTINUE        ;Good 16-bit read
;
; TMR1L may have rolled over between the read
; of the high and low bytes. Reading the high
; and low bytes now will read a good value.
;
    MOVF    TMR1H, W        ;Read high byte
    MOVWF   TMPH            ;
    MOVF    TMR1L, W        ;Read low byte
    MOVWF   TMPL            ;
;   Re-enable Interrupt (if required)
CONTINUE                    ;Continue with
    :                       ;your code
```

8.4 Timer1 Oscillator

A crystal oscillator circuit is built in between T1OSI pin (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200KHz. It will continue to run during SLEEP. It is primarily intended for a 32KHz crystal. Table 8-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must allow software time-out to ensure proper oscillator start-up.

TABLE 8-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Osc Type	Freq	C1	C2
LP	32 KHz§	15 pF	15 pF
	100 KHz	15 pF	15 pF
	200 KHz	0 - 15 pF	0 - 15 pF

Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only.
§For V_{DD} > 4.5V, C1 = C2 ≈ 30pf is recommended.

8.5 Resetting Timer1 using a CCP Trigger Output

If CCP1 or CCP2 module is configured in compare mode to generate a "special event" trigger (CCP1M<3:0> = 1011), this signal will reset timer1.

Timer1 must be configured for timer or synchronized counter mode operation to take advantage of this feature. If the Timer1 is running in asynchronous counter mode, this reset operation may not work.

In the event that a write to Timer1 coincides with a reset trigger from CCP1 or CCP2, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL registers pair effectively becomes the period register for the Timer1.

8.6 Resetting of Timer1 Registers

TMR1H and TMR1L registers are not reset on POR or any other reset except by the CCP1 special reset trigger.

T1CON register is reset to 00h on Power-on Reset. In any other reset, the register is unaffected.

8.7 TMR1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers

TABLE 8-2: REGISTERS ASSOCIATED WITH TMR1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
0C	PIR1	†PSPIF	—	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF
8C	PIE1	†PSPIE	—	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer1 Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1INSYNC	TMR1CS	TMR1ON

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by Timer1 module.
† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

9.0 TIMER2 (TMR2) MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It is especially suitable as PWM time-base (for PWM mode of CCP modules). TMR2 is a readable and writable register, and is cleared on any device reset.

The input clock (OSC/4) has a prescale option of 1, 4 or 16 (selected by control bits T2CKPS1, T2CKPS0, of register T2CON).

Timer2 has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is set to all 1s during a reset.

The overflow (or match) output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling) to generate a Timer2 interrupt (latched in TMR2IF bit, PIR<1>).

TMR2 can be shut off using TMR2ON (T2CON<2>) control bit to minimize power consumption.

Figure 9-2 shows the TMR2 control register.

9.1 TMR2 Prescaler and Postscaler

The prescaler and postscaler counters are cleared when any of the following occurs: a write to the TMR2 register, a write to the T2CON register, or any device reset (Power-on Reset, $\overline{MCLR}$ reset, or Watchdog Timer reset). TMR2 will not clear when T2CON is written, only for a WDT, $\overline{POR}$, and $\overline{MCLR}$ reset.

9.2 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the synchronous serial port module which optionally uses it to generate shift clock.

FIGURE 9-1: TIMER2 BLOCK DIAGRAM

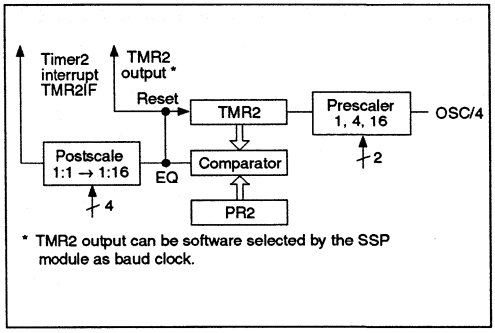


FIGURE 9-2: T2CON: TIMER2 CONTROL REGISTER

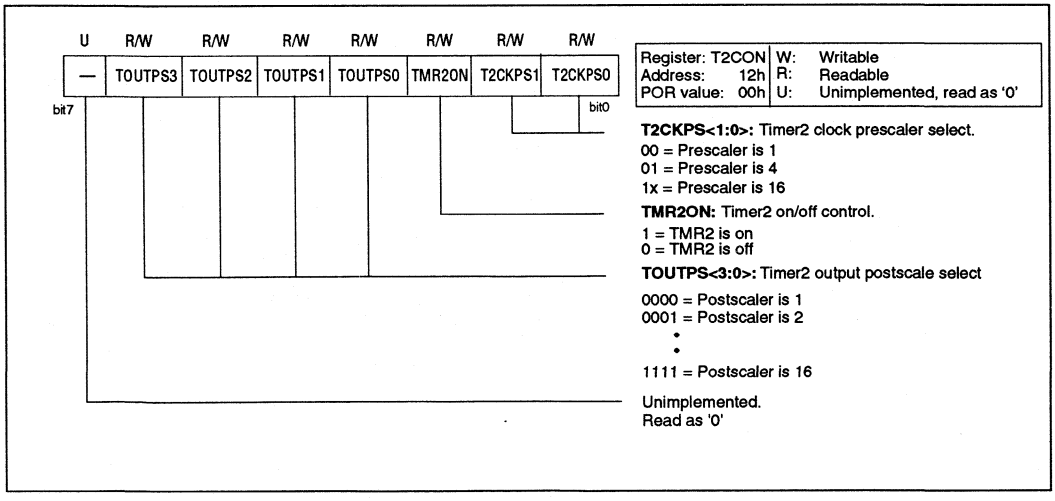


TABLE 9-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR21E	TMR1IE
11h	TMR2	Timer2							
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKS1	T2CKPS0
92h	PR2	Timer2 period Register.							

Legend: — = Unimplemented locations, read as '0'
Note: Shaded boxes are not used by Timer2 module.
† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

10.0 CAPTURE/COMPARE/PWM MODULE

The PIC16C7X has two Capture/Compare/PWM (CCP) modules consisting of a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM output. (Both the CCP1 and CCP2 modules are identical in operation, with the exception of the special trigger.) In the following sections, the operation of a CCP module is described with respect to CCP1. Please note that CCP2 is similar to CCP1, except where noted.

CCP1 module:

Capture/compare/PWM register1 (CCPR1) is made up of two 8-bit sections: low byte, CCPR1L and high byte, CCPR1H. Both are readable and writable.

CCP2 module:

Capture/compare/PWM register2 (CCPR2) is made up of two 8-bit sections: low byte, CCPR2L and high byte, CCPR2H. Both are readable and writable. The interaction of multiple CCP modules is discussed in application note AN594.

10.1 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of TMR1 when an event occurs on pin RC2/CCP1. An event is defined as:

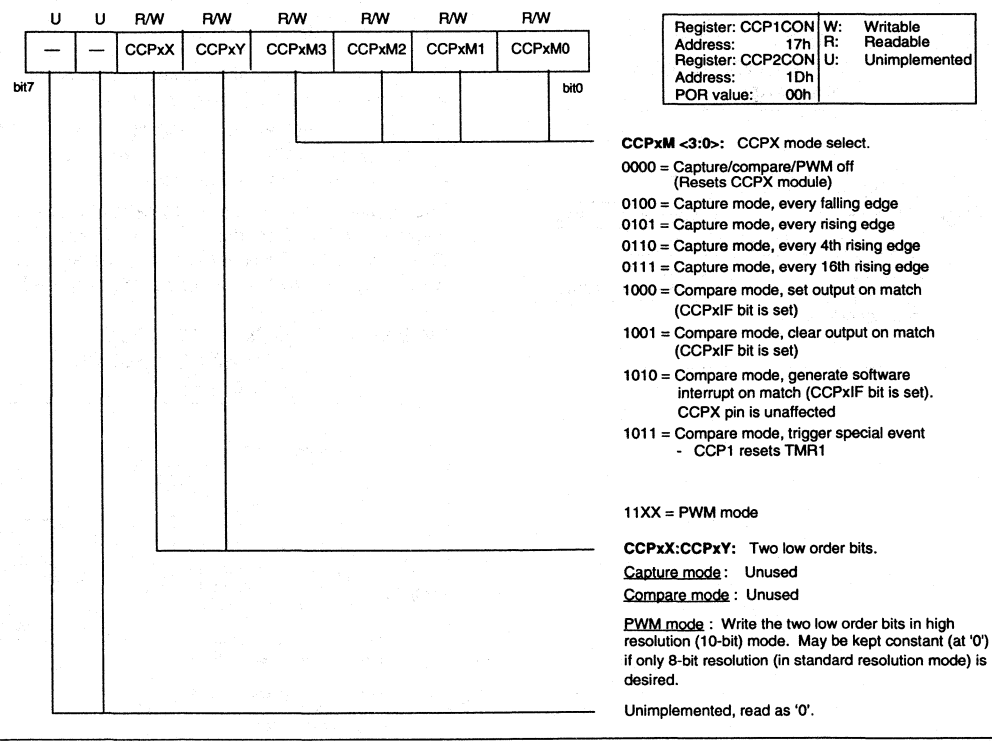
1. A falling edge
2. A rising edge
3. Every 4 rising edges
4. Every 16 rising edges

One of these is selected by the control bits CCP1M <3:0> in register CCP1CON. When a capture is made the interrupt request flag, CCP1IF bit (PIR1<2>) is set. It must be reset in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost. In capture mode, the RC2/CCP1 pin should be configured as an input through its corresponding TRIS bit.

Note: If the RC2/CCP1 is configured as an output, a write to the port can cause a capture condition.

When the capture mode is changed, a false capture interrupt may be generated. The user should keep CCP1IE clear to avoid false interrupts and should clear the CCP1IF bit following any such change in operating mode.

FIGURE 10-1: CCP1CON/CCP2CON REGISTER



10.1.1 PRESCALER

There are four prescaler settings, specified by the CCP1M3-CCP1M0 bits. Whenever the CCP module is turned off, or the CCP module is not in capture mode, the prescaler counter is cleared. This means that any reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, and therefore the first capture may be from a non-zero prescaler. Example 10-1 shows the recommended way to switch between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

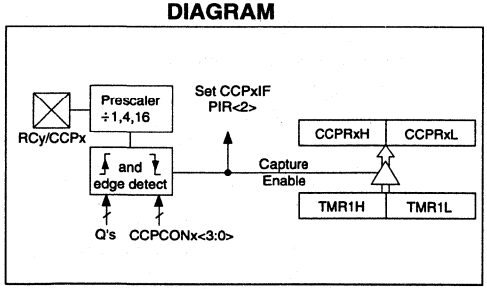
EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON     ; Turn CCP module off
MOVLW   NEW_CAPT_PS ; Load the W reg with
                    ; the new prescaler
                    ; mode value and CCP ON
MOVWF   CCP1CON     ; Load CCP1CON with
                    ; this value
```

10.1.2 TMR1 MODE SELECTION

TMR1 must be running in timer mode or synchronized counter mode for the CCP1 modules to use the capture feature. In asynchronous counter mode the capture operation may not work.

FIGURE 10-2: CAPTURE MODE OPERATION BLOCK DIAGRAM



10.2 Compare Mode

In compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1. When a match occurs, the RC2/CCP1 pin is:

- Driven High
- Driven Low
- Remains Unchanged

The action on the pin is based on the control bits CCP1M <3:0> in register CCP1CON. At the same time, a compare interrupt is also generated. The user must set the RC2/CCP1 pin as an output through the TRISC<2> bit.

Note:	Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the data latch.

10.2.1 TIMER1 MODE SELECTION

Timer1 must be running in timer mode or synchronized counter mode if the CCP1 module is using the compare feature. In asynchronous counter mode, the compare operation may not work.

10.2.2 SOFTWARE INTERRUPT MODE

Another compare mode is software interrupt mode in which the CCP1 pin is not affected. Only CCP1IF interrupt is generated.

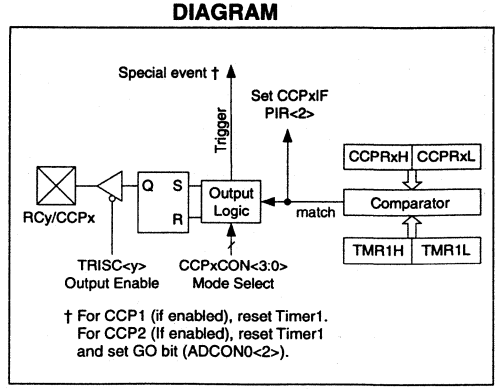
10.2.3 SPECIAL TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The special trigger output of CCP1 resets the TMR1. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special trigger output of CCP2 starts an A/D conversion (if the A/D module is on) and resets Timer1. This allows A/D conversions to be done at a constant (sampling) frequency without software overhead.

FIGURE 10-3: COMPARE MODE OPERATION BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

10.3 PWM Mode

In Pulse Width Modulation (PWM) mode, the RC2/CCP1 produces up to 10-bit resolution PWM output. This pin must be configured as an output through the TRISC<2> bit. The pin is multiplexed with the data latch. In PWM mode, the user writes the 8-bit duty-cycle value to the low byte of the CCPR1 register, namely CCPR1L. The high-byte, CCPR1H is used as the slave buffer to the low byte. The 8-bit data is transferred from the master to the slave when the PWM1 output is set (i.e. at the beginning of the duty cycle). This double buffering is essential for glitchless PWM output. In PWM mode, CCPR1H is readable but not writable. The period of the PWM is determined by the Timer2 period register (PR2).

PWM period is =

[(PR2) + 1] • 4 Tosc • (TMR2 prescale value)

PWM duty cycle =

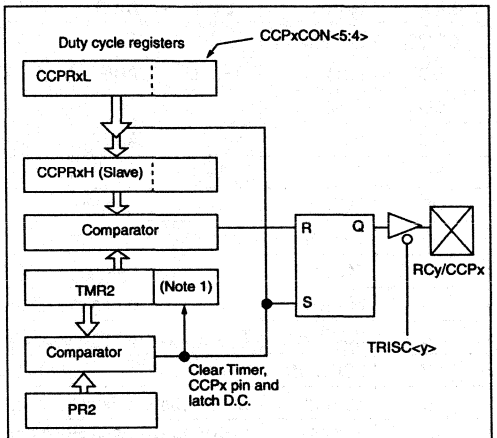
(DC1) • Tosc • (TMR2 prescale value)

where DC1 = 10 bit value from CCPRxL and CCPxCON<5:4> concantenated.

The PWM output resolution is therefore programmable up to a maximum of 10-bit.

Note: Clearing the CCP1CON register will force the RC2/CCP1 PWM output latch to the default low level. This is not the I/O data latch. The TMR2 postscaler is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

FIGURE 10-4: SIMPLIFIED PWM BLOCK DIAGRAM



Note: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

TABLE 10-1: PWM FREQUENCY VS RESOLUTION AT 20 MHZ

Max Resolution (High Resolution Mode)	Frequency		
	TMR2 Prescale=1	TMR2 Prescale=4	TMR2 Prescale=16
10 bit	19.53 kHz	4.88 kHz	1.22 kHz
9 bit	39.06 kHz	9.77 kHz	2.44 kHz
8 bit	78.13 kHz	19.53 kHz	4.88 kHz

TABLE 10-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHZ

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1,4,16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x5F
Resolution (High-resolution mode)	10-bit	10-bit	10-bit	8-bit	7-bit	6.5-bit
Resolution (Standard-resolution mode)†	8-bit	8-bit	8-bit	6-bit	5-bit	4.5-bit

† Standard resolution mode has the CCPIX:CCPIY bit constant (or '0'), and only compares the TMR2 against the PR2. The Q-cycles are not used.

© 1995 Microchip Technology Inc.

TABLE 10-3: REGISTERS ASSOCIATED WITH TIMER1 AND CAPTURE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
15	CCPR1L	Timer1 Capture Register (LSb)							
16	CCPR1H	Timer1 Capture Register (MSb)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L	Timer1 Capture Register (LSb)							
1C	CCPR2H	Timer1 Capture Register (MSb)							
1D	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0' Note: Shaded boxes are not used in this mode.
† This bit reserved on the PIC16C73.

TABLE 10-4: REGISTERS ASSOCIATED WITH TIMER1 AND COMPARE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
0E	TMR1L	Timer1 Least Significant Byte							
0F	TMR1H	Timer Most Significant Byte							
10	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
15	CCPR1L	Timer1 Capture Register (LSb)							
16	CCPR1H	Timer1 Capture Register (MSb)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L	Timer1 Capture Register (LSb)							
1C	CCPR2H	Timer1 Capture Register (MSb)							
1D	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0' Note: Shaded boxes are not used in this mode.
† This bit reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

TABLE 10-5: REGISTERS ASSOCIATED WITH TIMER2 AND PWM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8C	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
11	TMR2	Timer2							
92	PR2	Timer2 period Register							
12	T2CON	—	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	T2CKPS1	T2CKPS0
15	CCPR1L	Timer2 Duty Cycle Register							
16	CCPR1H	Timer2 Duty Cycle Register (Slave)							
17	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
1B	CCPR2L	Timer2 Duty Cycle Register							
1C	CCPR2H	Timer2 Duty Cycle Register (Slave)							
1D	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0

Legend: — = Unimplemented locations, Read as '0' Note: Shaded boxes are not used in this mode.
† This bit reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

NOTES:

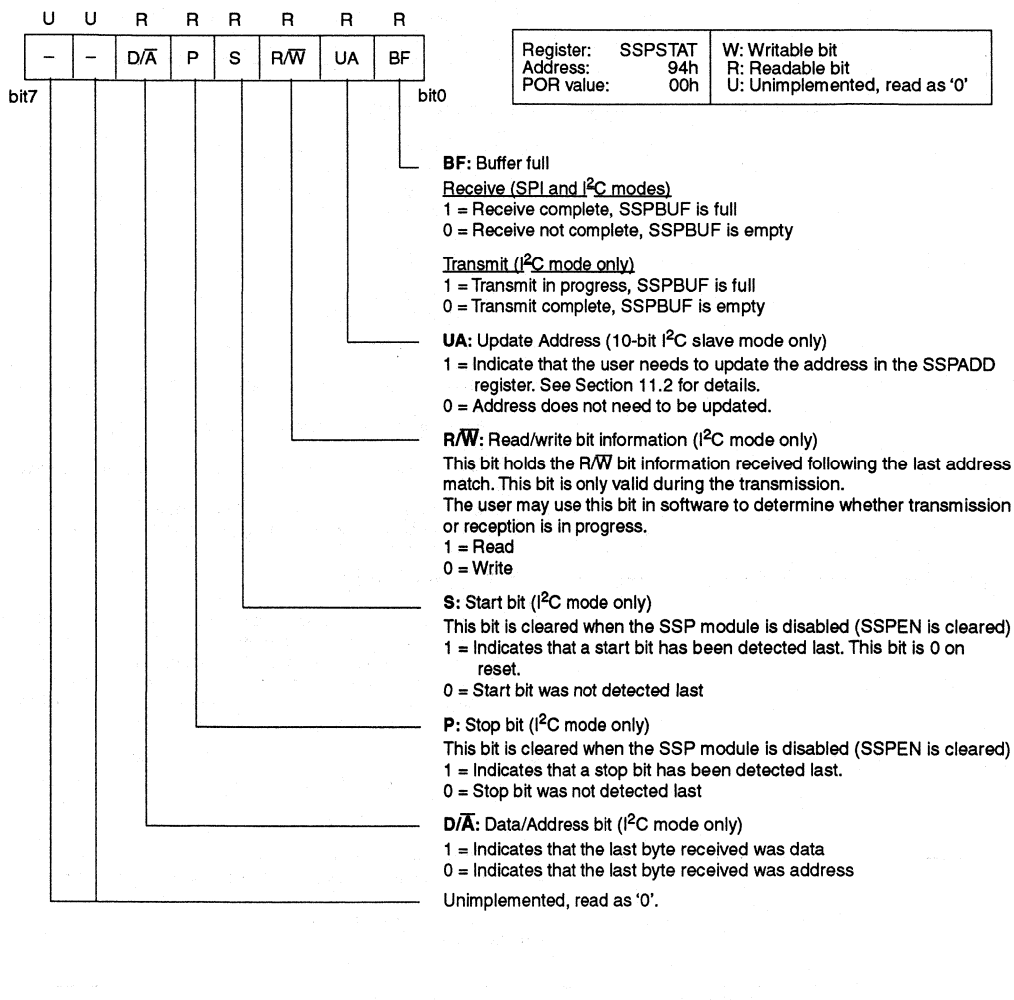
© 1995 Microchip Technology Inc.

11.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

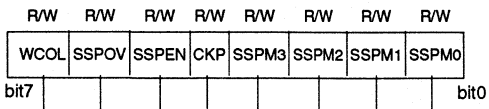
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

FIGURE 11-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER



© 1995 Microchip Technology Inc.

FIGURE 11-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER



Register:	SSPCON	W: Writable bit
Address:	14h	R: Readable bit
POR value:	00h	U: Unimplemented, read as '0'

SSPM<3:0>: Synchronous serial port mode select

0000 = SPI master mode, clock = OSC/4
0001 = SPI master mode, clock = OSC/16
0010 = SPI master mode, clock = OSC/64
0011 = SPI master mode, clock = (TMR2 output/2)
0100 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
0110 = I^2C slave mode, 7-bit address
0111 = I^2C slave mode, 10-bit address
1011 = I^2C master mode support enabled (slave idle)
1110 = I^2C slave mode, 7-bit address with master mode support enabled
1111 = I^2C slave mode, 10-bit address with master mode support enabled

CKP: Clock polarity select.

In SPI modes:
1 = Transmit happens on falling edge, receive on rising edge. Idle state for clock is a High level.
0 = Transmit happens on rising edge, receive on falling edge. Idle state for clock is a Low level.

In I^2C modes:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch)
Note: Used to ensure data setup time

SSPEN: Sync serial port enable

In SPI modes:
1 = Enables serial port and configures SCK, SDO and SDI as serial port pins.
0 = Disables serial port and configures these pins as I/O port pins.

In I^2C modes:
1 = Enables the serial port and configures SDA and SCL pins as serial port pins.
0 = Disables serial port and configures these pins as I/O port pins.
In both modes, when enabled, these pins must be properly configured as input or output.

SSPOV: Receive overflow flag.

In SPI modes:
1 = A new byte is received while SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode overflow bit is not set since each new reception (and transmission) is initiated by writing to SSPBUF.
0 = No overflow

In I^2C modes:
1 = A byte is received while the SSPBUF is still holding the previous byte. SSPOV is a don't care in transmit mode. SSPOV must be cleared in software in either mode.

WCOL: Write collision detect.

1 = the SSPBUF register is written while it is still transmitting the previous word.
Must be cleared in software.
0 = No collision

© 1995 Microchip Technology Inc.

11.1 SPI Mode

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI
- Serial Clock (SCK) RC3/SCK

Additionally a fourth pin may be used when in a slave mode of operation:

- Slave Select ($\overline{SS}$) RA5/$\overline{SS}$

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bit in the SSPCON register (SSPCON<5:0>). These control bits allow the following to be specified:

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

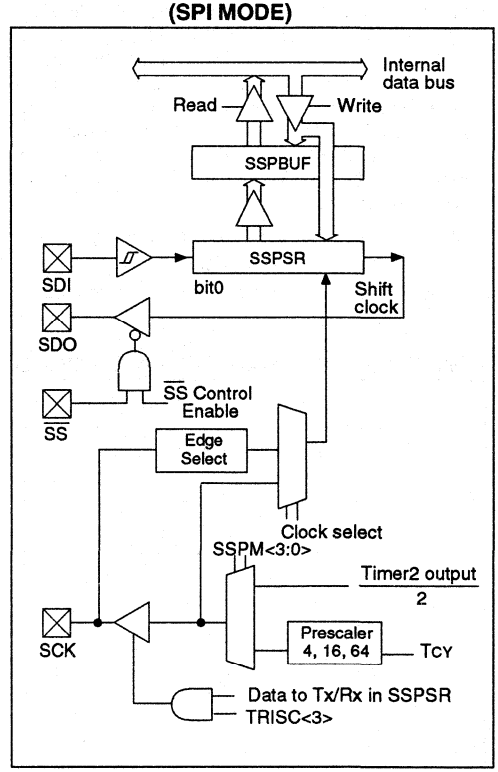
The SSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSB first, while the SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that information is moved to the SSPBUF register, the Buffer Full (BF) bit (SSPSTAT <0>) and the SSPIF bit are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect (WCOL) bit (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF completed successfully. When the application software is expecting to receive valid data, the SPPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full (BF) bit (SSPSTAT<0>) indicates when the SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF can then be read (if data is meaningful) and/or the SSPBUF (SSPSR) can be written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-1 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOPBSF   STATUS, RP0  ;Specify Bank 1
  BTFSS   SSPSTAT, BF  ;Has data been
                       ;received
                       ;(transmit
                       ;complete)?
  GOTO    LOOP         ;No
  BCF     STATUS, RP0  ;Specify Bank 0
  MOVF    SSPBUF, W    ;W reg = contents
                       ;of SSPBUF
  MOVWF   RXDATA       ;Save in user RAM
  MOVF    TXDATA, W    ;W reg = contents of
                       ; TXDATA
  MOVWF   SSPBUF       ;New data to xmit.
```

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)



To enable the serial port, the SSP enable bit (SSPEN) must be set. To reset or reconfigure SPI mode, clear SPEN, re-initialize SSPCON, and then set SSPEN. This configures the SDI, SDO, SCK, and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and $\overline{SS}$ could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-4 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2) wishes to broadcast data by the software protocol.

In master mode the data is transmitted/received as soon as the SSPBUF is written to. If the SPI is only going to receive, the SCK output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor" mode.

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag (SSPIF) is set (PIR1<3>).

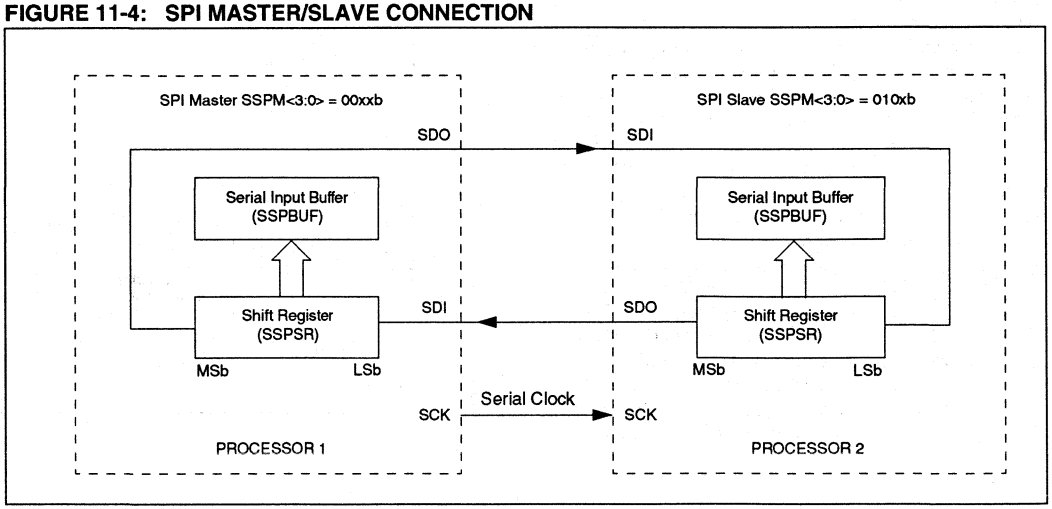
The clock polarity is selected by appropriately programming the CKP bit (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-5 and Figure 11-6 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- OSC / 4 (or TCY)
- OSC / 16 (or 4 • TCY)
- OSC / 64 (or 16 • TCY)
- Timer2 output / 2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times (see Table 17-7).

In sleep mode, the slave can transmit and receive data and wake-up the device from sleep.

FIGURE 11-4: SPI MASTER/SLAVE CONNECTION



© 1995 Microchip Technology Inc.

The $\overline{SS}$ pin allows a synchronous slave mode. The SPI must be in slave mode (SSPCON<3:0> = 04h) and the TRISA<5> bit must be set the for the synchronous slave mode to be enabled. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up / pull-down resisters may be desirable, depending on the application.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 11-5: SPI MODE TIMING (MASTER MODE OR SLAVE MODE W/O $\overline{SS}$ CONTROL)

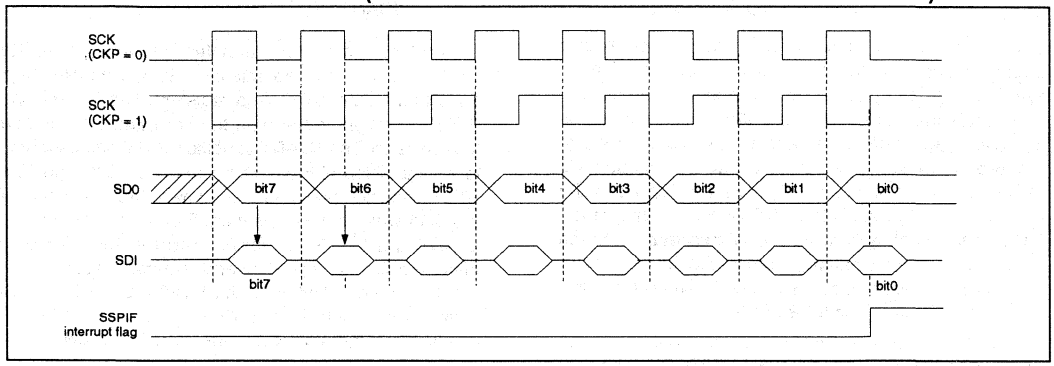


FIGURE 11-6: SPI MODE TIMING (SLAVE MODE WITH $\overline{SS}$ CONTROL)

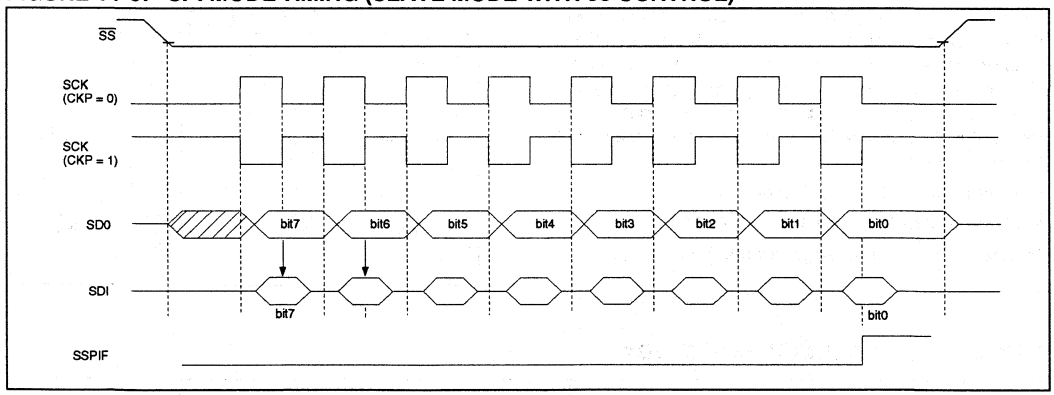


TABLE 11-1: REGISTERS ASSOCIATED WITH SPI OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	†PSPIF	—			SSPIF	CCP1IF	TMR2IF	TMR1IF
8C	PIE1	†PSPIE	—			SSPIE	CCP1IE	TMR2IE	TMR1IE
13	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
14	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
94	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF

Legend: — = Unimplemented locations, Read as '0'
Note: Shaded boxes are not used by SSP module in SPI mode.
† This bit is reserved on the PIC16C73.

11.2 I²C™ Overview

This section provides an overview of the Inter-IC (I^2C) bus, with Section 11.3 discussing the operation of the SSP module in I^2C mode. The I^2C bus is a two-wire serial interface developed by the Philips Corporation. The original specification, or standard mode, was for data transfers of up to 100 Kbps. An enhanced specification, or fast mode, supports data transmission up to 400 Kbps. Both standard mode and fast mode devices will inter-operate if attached to the same bus.

The I^2C interface employs a comprehensive protocol to ensure reliable transmission and reception of data. When transmitting data, one device is the "master" (generates the clock), while the other device(s) acts as the "slave." All portions of the slave protocol are implemented in the SSP module's hardware, while portions of the master protocol need to be addressed in the PIC16CXX software. Table 11-2 defines some of the I^2C-bus terminology. For additional information on the I^2C interface specification, refer to the Philips' document "*The I^2C-bus and how to use it.*" The order number for this document is 98-8080-575.

In the I^2C interface protocol each device has an address. When a master wishes to initiate a data transfer, it first transmits the address of the device that it wishes to "talk" to. All devices "listen" to see if this is their address. Within this address, a bit specifies if the master wishes to read / write from / to the slave device. The master and slave are always in opposite modes (transmitter/receiver) of operation during a data transfer. That is they can be thought of operating in either of these two relations:

- Master-transmitter and Slave-receiver
- Slave-transmitter and Master-receiver

In both cases the master generates the clock signal.

The output stages of the clock (SCL) and data (SDA) lines must have an open-drain or open-collector in order to perform the wired-AND function of the bus. External pull-up resistors are used to ensure a high level, when no device is pulling the line down. The number of devices that may be attached to the I^2C bus is limited only by the maximum bus loading specification of 400 pF.

11.2.1 INITIATING AND TERMINATING DATA TRANSFER

During times of no data transfer (idle time), both the clock line (SCL) and the data line (SDA) are pulled high through the external pull-up resistors. The START and STOP conditions determine the start and stop of data transmission. The START condition is defined as a high to low transition of the SDA when the SCL is high. The STOP condition is defined as a low to high transition of the SDA when the SCL is high. Figure 11-7 shows the START and STOP conditions. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP conditions, when data is being transmitted the SDA line can only change state when the SCL line is low.

FIGURE 11-7: START AND STOP CONDITIONS

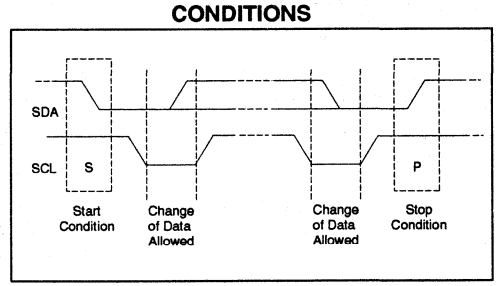


TABLE 11-2: I^2C-BUS TERMINOLOGY

Term	Description
Transmitter	The device that sends the data to the bus
Receiver	The device that receives the data from the bus
Master	The device which initiates the transfer, generates the clock and terminates the transfer
Slave	The device addressed by a master
Multi-master	More than one master device in a system. These masters can attempt to control the bus at the same time without corrupting the message.
Arbitration	Procedure that ensures that only one of the master devices will control the bus. This ensure that the transfer data does not get corrupted.
Synchronization	Procedure where the clock signals of two or more devices are synchronized.

© 1995 Microchip Technology Inc.

11.2.2 ADDRESSING I²C DEVICES

There are two address formats. The simplest is the 7-bit address format with a R/$\overline{W}$ bit (see Figure 11-8). The more complex is the 10-bit address with a R/$\overline{W}$ bit (see Figure 11-9). For 10-bit address format, two bytes must be transmitted with the first five bits specifying this to be a 10-bit address.

FIGURE 11-8: 7-BIT ADDRESS FORMAT

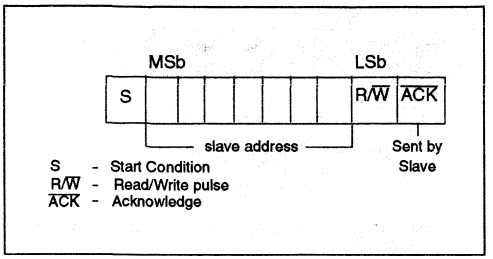
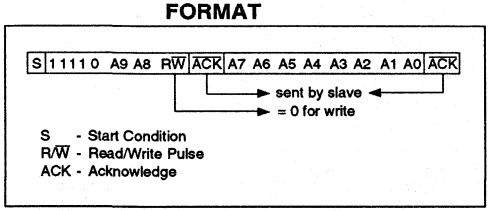


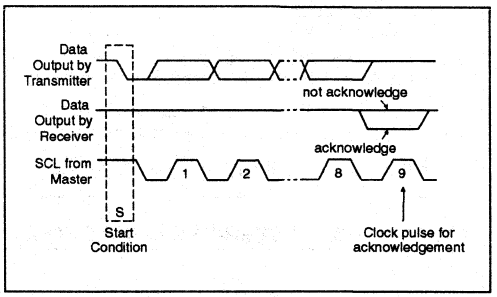
FIGURE 11-9: I²C 10-BIT ADDRESS FORMAT



11.2.3 TRANSFER ACKNOWLEDGE

All data must be transmitted per byte, with no limit to the number of bytes transmitted per data transfer. After each byte, the slave-receiver generates an acknowledge bit ($\overline{ACK}$). This is shown in Figure 11-10. When a slave-receiver doesn't acknowledge the slave address or received data, the master must abort the transfer. The slave must leave SDA high so that the master can generate the STOP condition (see Figure 11-7).

FIGURE 11-10: SLAVE-RECEIVER ACKNOWLEDGE



If the master is receiving the data (master-receiver), it generates an acknowledge signal for each received byte of data, except for the last byte. To signal the end of data to the slave-transmitter, the master does not generate an acknowledge (not acknowledge). The slave then releases the SDA line so the master can generate the STOP condition. The master can also generate the STOP condition during the acknowledge pulse for valid termination of data transfer.

If the slave needs to delay the transmission of the next byte, holding the SCL line low will force the master into a wait state. Data transfer continues when the slave releases the SCL line. This allows the slave to move the received data or fetch the data it needs to transfer before allowing the clock to start. This wait state technique can also be implemented at the bit level. Figure 11-11 shows a data transfer waveform.

FIGURE 11-11: A DATA TRANSFER

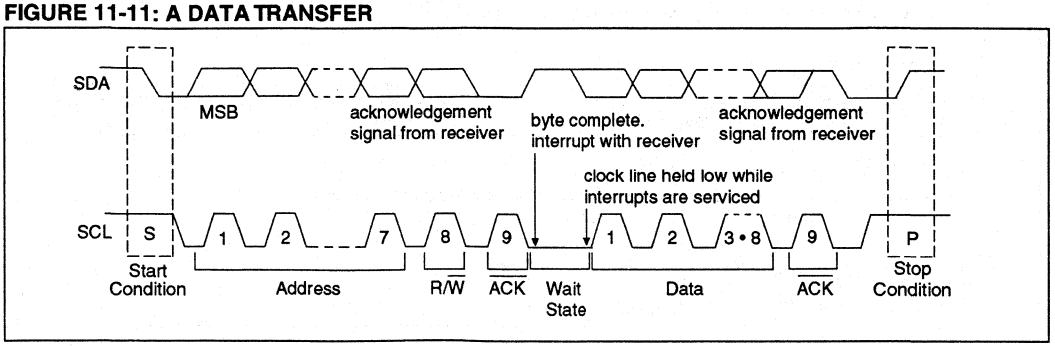


Figure 11-12 and Figure 11-13 show Master-transmitter and Master-receiver data transfer sequences.

When a master does not wish to relinquish the bus (by generating a STOP condition), a repeated START condition (Sr) must be generated. This condition is identical to the start condition (SDA goes high-to-low while SCL is high), but occurs after a data transfer acknowledge pulse (not the bus-free state). This allows a master to send "commands" to the slave and then receive the requested information or to address a different slave device. This sequence is shown in Figure 11-14.

FIGURE 11-12: MASTER-TRANSMITTER SEQUENCE

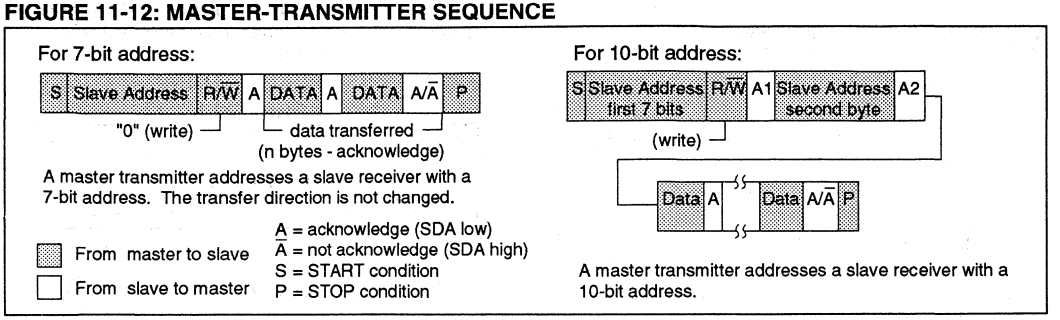


FIGURE 11-13: MASTER-RECEIVER SEQUENCE

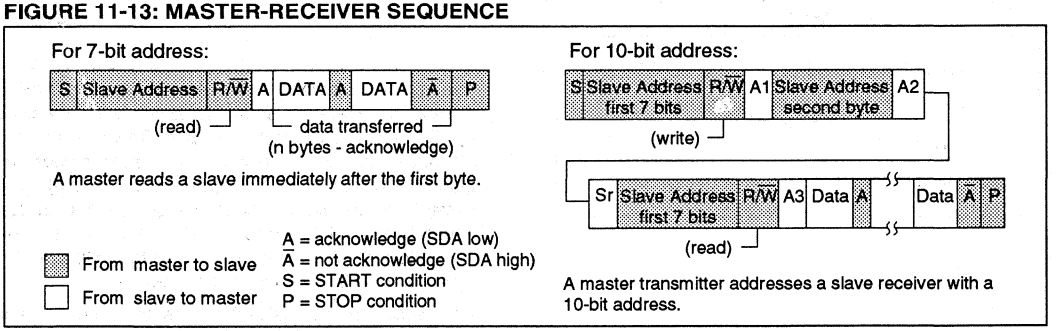
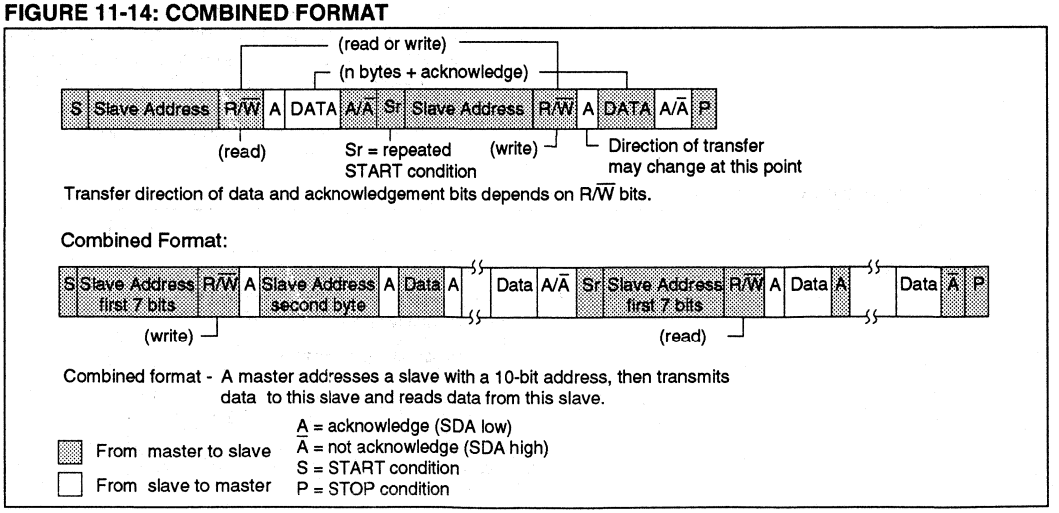


FIGURE 11-14: COMBINED FORMAT



© 1995 Microchip Technology Inc.

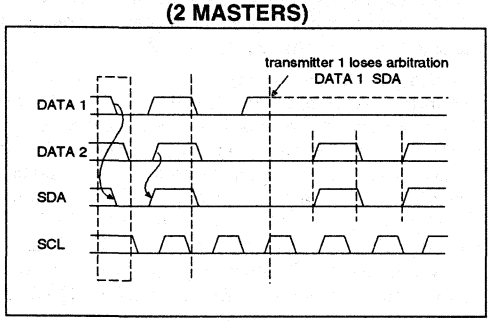
11.2.4 MULTI-MASTER

The I^2C protocol allows a system to have more than one master. This is called multi-master. When two or more masters try to transfer data at the same time, arbitration and synchronization occur.

11.2.4.1 ARBITRATION

Arbitration takes place on the SDA line, while the SCL line is high. The master which transmits a high when the other master transmits a low loses arbitration (see Figure 11-15), and turns off its data output stage. A master which lost arbitration can generate clock pulses until the end of the data byte where it lost arbitration. When the master devices are addressing the same device, arbitration continues into the data.

FIGURE 11-15: MULTI-MASTER ARBITRATION (2 MASTERS)



Masters that also incorporate the slave function, and have lost arbitration must immediately switch over to slave-receiver mode. This is because the winning master-transmitter may be addressing it.

Arbitration is not allowed between:

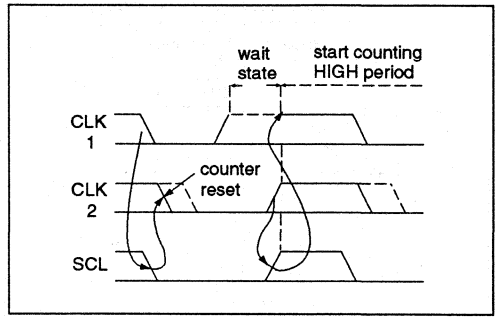
- A repeated START condition
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Care needs to be taken to ensure that these conditions do not occur.

11.2.4.2 Clock Synchronization

Clock synchronization occurs after the devices have started arbitration. This is performed using a wired-AND connection to the SCL line. A high to low transition on the SCL line causes the concerned devices to start counting off their low period. Once a device clock has gone low, it will hold the SCL line low until its SCL high state is reached. The low to high transition of this clock may not change the state of the SCL line, if another device clock is still within its low period. The SCL line is held low by the device with the longest low period. Devices with shorter low periods enter a high wait-state, until the SCL line comes high. When the SCL line comes high, all devices start counting off their high periods. The first device to complete its high period will pull the SCL line low. The SCL line high time is determined by the device with the shortest high period. This is shown in Figure 11-16.

FIGURE 11-16: CLOCK SYNCHRONIZATION

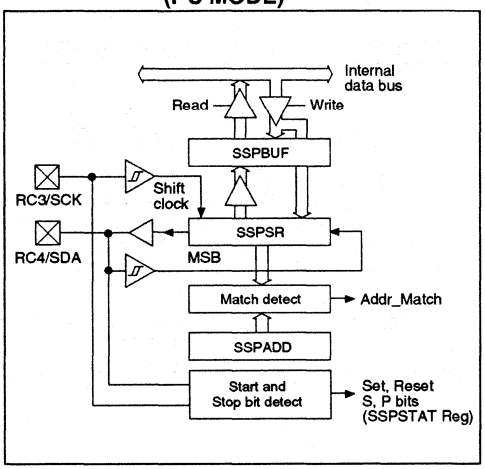


2

11.3 SSP I²C Operation

The SSP module in I²C mode fully implements all slave functions, and provides support in hardware to facilitate software implementations of the master functions. The SSP module implements the standard and fast mode specifications as well as 7-bit and 10-bit addressing. Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits. A block diagram of the SSP module in I²C mode is shown in Figure 11-17. The SSP module functions are enabled by setting the SSP Enable (SSPEN) bit (SSPCON<5>).

FIGURE 11-17: SSP BLOCK DIAGRAM (I²C MODE)



The SSP module has five registers for I²C operation. These are the:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive / Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- Address Register(SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allows one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with master-mode support enabled
- I²C Slave mode (10-bit address), with master-mode support enabled
- I²C Master mode, support enabled slave is idle

Selection of any I²C mode and with the SSPEN bit set, forces the SCL and SDA pins to be open collector, provided these pins are set to inputs through the TRISC bits.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer. The SSPSTAT register is read only.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF and the SSPIF is set. If another complete byte is received before the SSPBUF is read, a receiver overflow has occurred and the SSPOV bit (SSPCON<6> is set.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (1 1 1 1 0 A9 A8 0). Following the high byte address match, the low byte of the address needs to be loaded (A7 - A0).

© 1995 Microchip Technology Inc.

11.3.1 SLAVE MODE

In slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched or the data transfer from an address match is received, the hardware automatically will generate the acknowledge ($\overline{ACK}$) pulse, and then load the SSPBUF with the received value in the SSPSR.

There are certain conditions that will cause the SSP module not to give this $\overline{ACK}$ pulse. These are if either (or both):

- The Buffer Full (BF) bit was set before the transfer was received.
- The Overflow (SSPOV) bit was set before the transfer was received.

In this case, the SSPSR value is not loaded into the SSPBUF, but the SSPIF bit is set. Table 11-3 shows what happens when a data transfer byte is received, given the status of the BF and SSPOV bits. The shaded boxes shows the condition where user software did not properly clear the overflow condition. The BF flag is cleared by reading the SSPBUF register while the SSPOV bit is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I^2C specification as well as the requirement of the SSP module is shown in timing parameter #100 and parameter #101.

11.3.1.1 ADDRESSING

Once the SSP module has been enabled, The SSP waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR. All incoming bits are sampled with the rising edge of the clock (SCL) line. The SSPSR<7:1> is compared to the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following things happen:

- SSPSR loaded into SSPBUF
- Buffer Full (BF) bit is set
- $\overline{ACK}$ pulse is generated
- SSP Interrupt Flag (SSPIF) is set (interrupt is generated if enabled) - on falling edge of ninth SCL pulse

In 10-bit address mode, two address bytes need to be received by the slave (see Figure 11-9). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. The R/$\overline{W}$ bit (bit 0) must specify a write, so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1 1 1 1 0 A9 A8 0', where A9 and A8 are the two MSbs of the address. The sequence of events for 10-bit address are as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (SSPIF, BF and UA are set)
2. Update SSPADD with second (low) byte of Address (clears UA and releases SCL line)
3. Read SSPBUF (clears BF) and clear SSPIF
4. Receive second (low) byte of Address (SSPIF, BF and UA are set)
5. Update SSPADD with first (high) byte of Address (clears UA, if match releases SCL line)
6. Read SSPBUF (clears BF) and clear SSPIF
7. Receive Repeated START condition
8. Receive first (high) byte of Address (SSPIF and BF are set)
9. Read SSPBUF (clears BF) and clear SSPIF

TABLE 11-3: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received				
BF	**SSPOV**	**SSPSR → SPBUF**	**Generate $\overline{ACK}$ Pulse**	**Set SSPIF bit (SSP Interrupt if Enabled)**
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

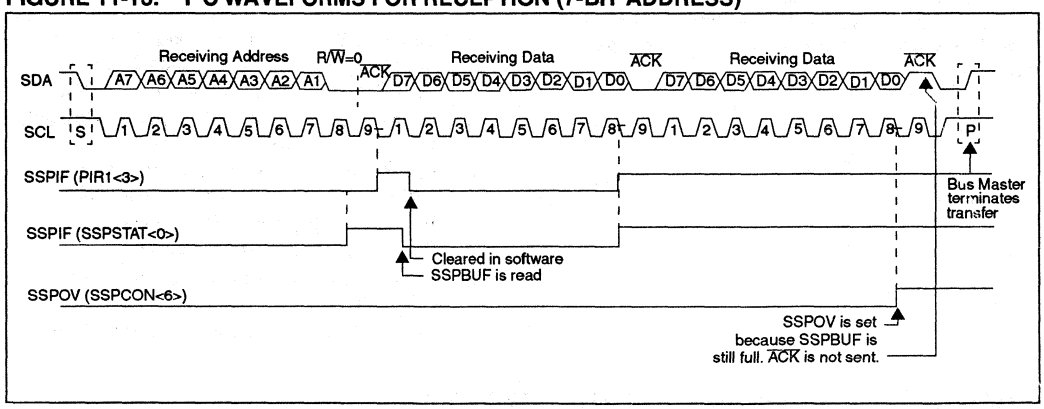
11.3.1.2 RECEPTION

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF.

When the address byte overflow condition exists then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either the BF bit (SSPSTAT<0>) is set or the SSPOV bit (SSPCON<6>) is set.

A SSPIF interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software, and the SSPSTAT register is used to determine the status of the byte.

FIGURE 11-18: I^2C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)



© 1995 Microchip Technology Inc.

11.3.1.3 TRANSMISSION

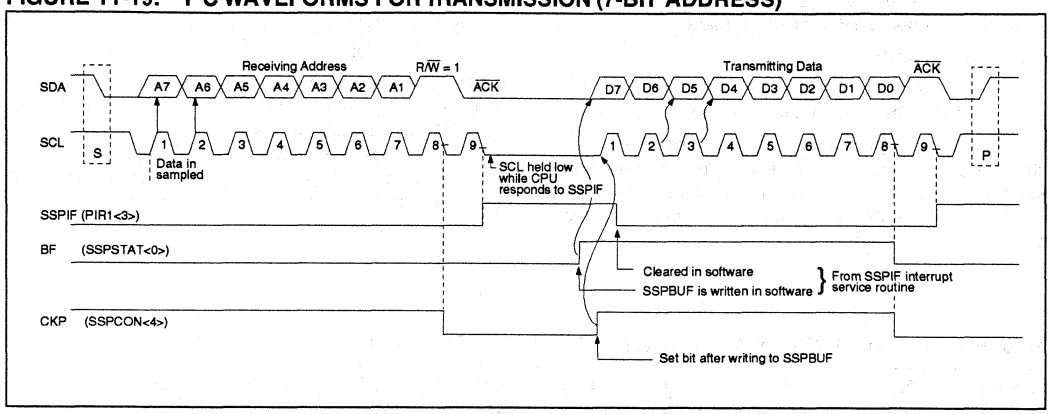
When the R/$\overline{W}$ bit of the address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF. The $\overline{ACK}$ pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit (SSPCON<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (see Figure 11-19).

A SSPIF interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. The slave then monitors for another occurrence of the START bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then the SCL pin should be enabled by setting the CKP bit (SSPCON<4>).

2

FIGURE 11-19: I^2C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



11.3.2 MASTER MODE

Master mode of operation is supported by interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are cleared

In master mode the SCL and SDA lines are manipulated by changing the corresponding TRISC<4:3> bit(s) to an output (cleared). The output level is always low, irrespective of the value(s) in PORTB<4:3>. So when transmitting data, a "1" data bit must have the TRISC<4> bit set (input) and a "0" data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit.

The following events will cause the SSP Interrupt Flag (SSPIF) to be set (SSP Interrupt if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received

Master mode of operation can be done with either the slave mode idle (SSPM3 - SSPM0 = 1011) or with the slave active. When both master and slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

11.3.3 MULTI-MASTER MODE

In multi-master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are cleared. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the stop condition occurs.

In multi-master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, the device may be being addressed. If addressed an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

TABLE 11-4: REGISTERS ASSOCIATED WITH I^2C OPERATION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0Ch	PIR1	†PSPIF		RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
8Ch	PIE1	†PSPIE		RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register							
93h	DSSPADD	Synchronous Serial Port (I^2C mode) Address Register							
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
94h	SSPSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF

Legend: — = Unimplemented locations, Read as '0'.

Note 1: Shaded boxes are not used by the SSP module in I^2C mode.

† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

FIGURE 11-20: OPERATION OF THE I²C MODULE IN IDLE_MODE, RCV_MODE OR XMIT_MODE

```
IDLE_MODE (7-bit):
if (Addr_match)                 {   Set interrupt;
                                    if (R/W = 1)    {   Send ACK = 0;
                                                        set XMIT_MODE;
                                                    }
                                    else if (R/W = 0) set RCV_MODE;
                                }
RCV_MODE:
if ((SSPBUF=Full) OR (SSPOV = 1))
        {   Set SSPOV;
            Do not acknowledge;
        }
else    {   transfer SSPSR → SSPBUF;
            send ACK = 0;
        }
Receive 8-bits in SSPSR;
Set interrupt;
XMIT_MODE:
While ((SSPBUF = Empty) AND (CKP=0)) Hold SCL Low;
Send byte;
Set interrupt;
if ( ACK Received = 1)          {   End of transmission;
                                    Go back to IDLE_MODE;
                                }
else if ( ACK Received = 0)  Go back to XMIT_MODE;
IDLE_MODE (10-Bit):
If (High_byte_addr_match AND (R/W = 0))
        {   PRIOR_ADDR_MATCH = FALSE;
            Set interrupt;
            if ((SSPBUF = Full) OR ((SSPOV = 1))
                {   Set SSPOV;
                    Do not acknowledge;
                }
            else    {   Set UA = 1;
                        Send ACK = 0;
                        While (SSPADD not updated) Hold SCL low;
                        Clear UA = 0;
                        Receive Low_addr_byte;
                        Set interrupt;
                        Set UA = 1;
                        If (Low_byte_addr_match)
                            {   PRIOR_ADDR_MATCH = TRUE;
                                Send ACK = 0;
                                while (SSPADD not updated) Hold SCL low;
                                Clear UA = 0;
                                Set RCV_MODE;
                            }
                    }
        }
else if (High_byte_addr_match AND (R/W = 1)
        {   if (PRIOR_ADDR_MATCH)
                {   send ACK = 0;
                    set XMIT_MODE;
                }
            else PRIOR_ADDR_MATCH = FALSE;
        }
```

NOTES:

© 1995 Microchip Technology Inc.

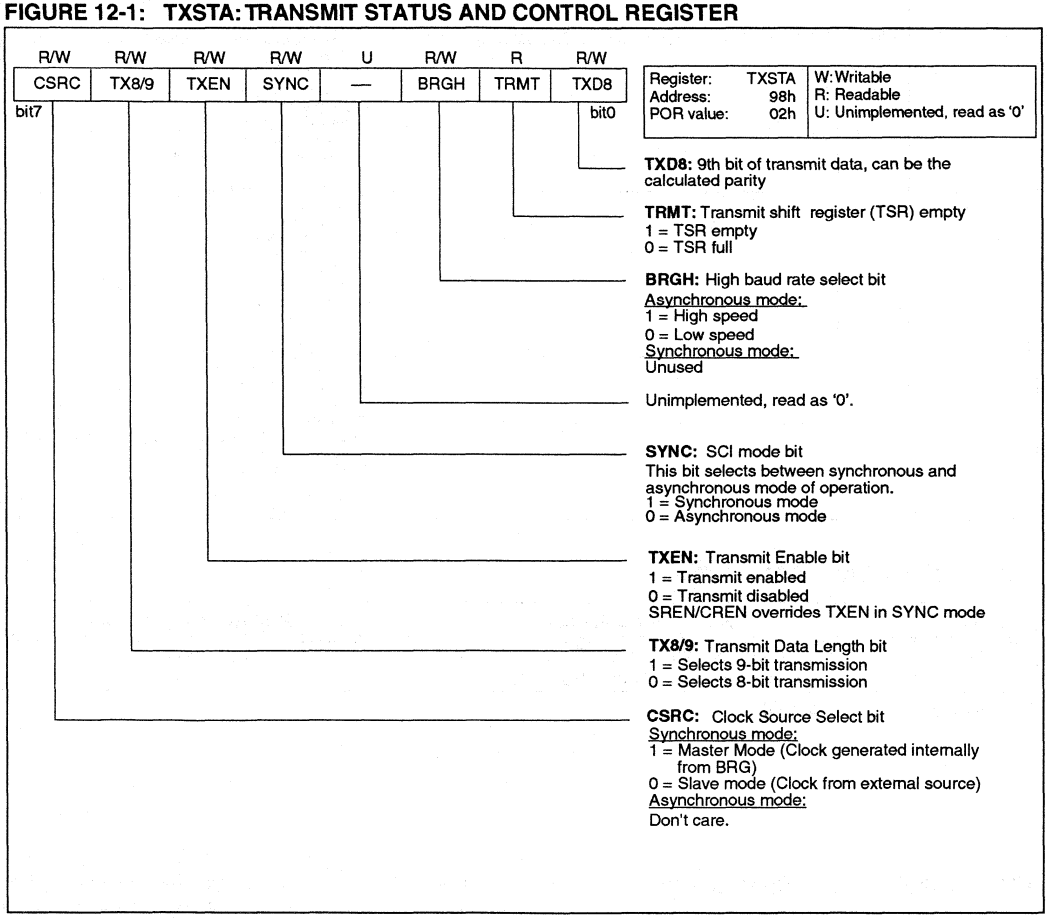
12.0 SERIAL COMMUNICATION INTERFACE (SCI) MODULE

The Serial Communication Interface (SCI) module, available in the PIC16C74 and PIC16C73 only, is one of the two serial I/O modules. The SCI can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc. The SCI can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

The SPEN (RCSTA<7>) bit has to be set in order to configure RC6 and RC7 as the Serial Communication Interface.

FIGURE 12-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER



© 1995 Microchip Technology Inc.

FIGURE 12-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

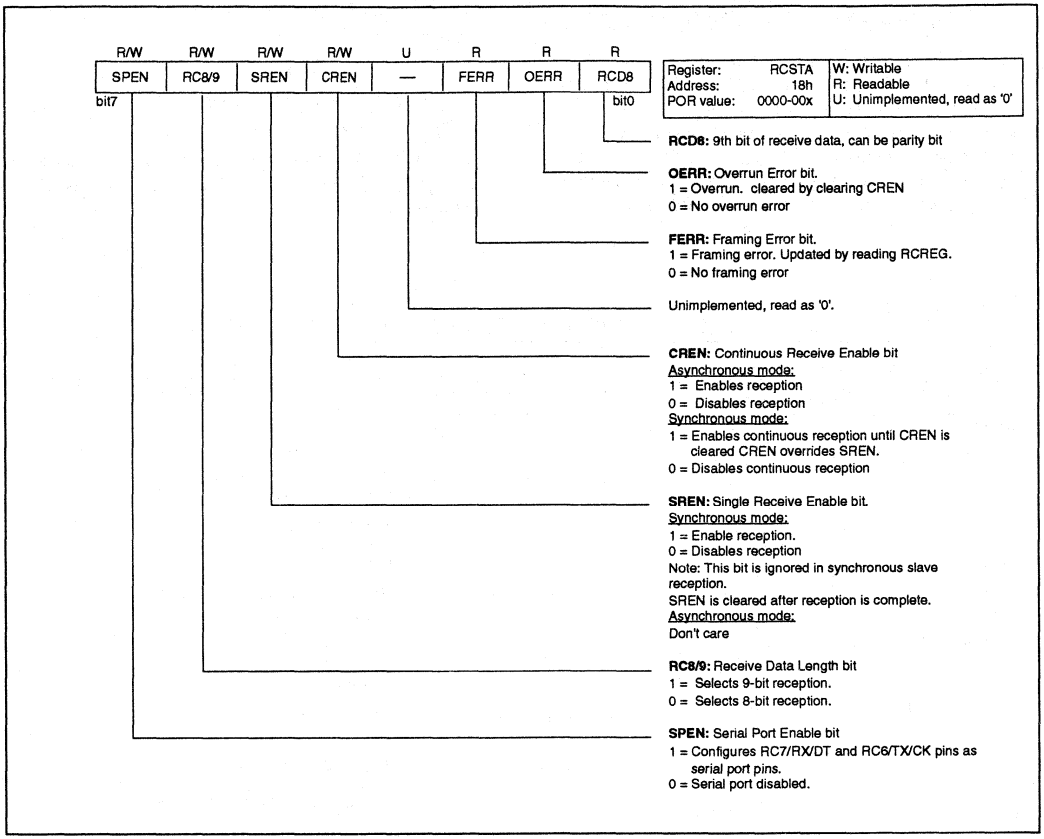


TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH=0 (Low Speed)	BRGH=1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

X = value in SPBRG (0 to 255)

12.1 SCI Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the SCI. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode the BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different SCI modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for SPBRG can be calculated using the formula in Table 12-1. From this, the error in baud rate can be determined.

© 1995 Microchip Technology Inc.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz
Desired Baud Rate = 9600
BRGH = 0
SYNC = 0

EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

Desired Baud rate=Fosc / (64 (X + 1))

9600 = 16000000 /(64 (X + 1))

X = ⌊25.042⌋ = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

Error = $\frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}}$

= (9615 - 9600) / 9600

= 0.16%

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the Fosc/(16(x + 1)) equation can reduce the baud rate error in some cases.

Writing a new value to the SPREG, causes the BRG timer to be reset (or cleared), this guarantees that the BRG does not wait for a timer overflow before outputting the new baud rate.

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0X98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	-	BRGH	TRMT	TXD8
0X99h	SPBRG	Baud Rate Register							

TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 20MHZ			16MHZ			10MHZ			7.15909MHZ		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.766	+1.73	255	9.622	+0.23	185
19.2	19.53	+1.73	255	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92
76.8	76.92	+0.16	64	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22
96	96.15	+0.16	51	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18
300	294.1	-1.96	16	307.69	+2.56	12	312.5	+4.17	7	298.3	-0.57	5
500	500	0	9	500	0	7	500	0	4	NA	-	-
HIGH	5000	-	0	4000	-	0	2500	-	0	1789.8	-	0
LOW	19.53	-	255	15.625	-	255	9.766	-	255	6.991	-	255

BAUD RATE (K)	Fosc = 5.0688MHZ			3.579545MHZ			1MHZ			32.768KHZ		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	0.303	+1.14	26
1.2	NA	-	-	NA	-	-	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	-	-	NA	-	-	2.404	+0.16	103	NA	-	-
9.6	9.6	0	131	9.622	+0.23	92	9.615	+0.16	25	NA	-	-
19.2	19.2	0	65	19.04	-0.83	46	19.24	+0.16	12	NA	-	-
76.8	79.2	+3.13	15	74.57	-2.90	11	83.34	+8.51	2	NA	-	-
96	97.48	+1.54	12	99.43	+3.57	8	NA	-	-	NA	-	-
300	316.8	+5.60	3	298.3	-0.57	2	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	1267	-	0	894.9	-	0	250	-	0	8.192	-	0
LOW	4.950	-	255	3.496	-	255	0.9766	-	255	0.032	-	255

© 1995 Microchip Technology Inc.

TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH=0)

BAUD RATE (K)	Fosc = 20MHZ KBAUD	%ERROR	SPBRG value (decimal)	16MHZ KBAUD	%ERROR	SPBRG value (decimal)	10MHZ KBAUD	%ERROR	SPBRG value (decimal)	7.15909MHZ KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.221	+1.73	255	1.202	+0.16	207	1.202	+0.16	129	1.203	+0.23	92
2.4	2.404	+0.16	129	2.404	+0.16	103	2.404	+0.16	64	2.380	-0.83	46
9.6	9.469	-1.36	32	9.615	+0.16	25	9.766	+1.73	15	9.322	-2.90	11
19.2	19.53	+1.73	15	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5
76.8	78.13	+1.73	3	83.33	+8.51	2	78.13	+1.73	1	NA	-	-
96	104.2	+8.51	2	NA	-	-	NA	-	-	NA	-	-
300	312.5	+4.17	0	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	312.5	-	0	250	-	0	156.3	-	0	111.9	-	0
LOW	1.221	-	255	0.977	-	255	0.6104	-	255	0.437	-	255

BAUD RATE (K)	Fosc = 5.0688MHZ KBAUD	%ERROR	SPBRG value (decimal)	3.579545MHZ KBAUD	%ERROR	SPBRG value (decimal)	1MHZ KBAUD	%ERROR	SPBRG value (decimal)	32.768KHZ KBAUD	%ERROR	SPBRG value (decimal)
0.3	0.31	+3.13	255	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.2	0	65	1.190	-0.83	46	1.202	+0.16	12	NA	-	-
2.4	2.4	0	32	2.432	+1.32	22	2.232	-6.99	6	NA	-	-
9.6	9.9	+3.13	7	9.322	-2.90	5	NA	-	-	NA	-	-
19.2	19.8	+3.13	3	18.64	-2.90	2	NA	-	-	NA	-	-
76.8	79.2	+3.13	0	NA	-	-	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	79.2	-	0	55.93	-	0	15.63	-	0	0.512	-	0
LOW	0.3094	-	255	0.2185	-	255	0.0610	-	255	0.0020	-	255

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH=1)

BAUD RATE (K)	Fosc = 20MHZ KBAUD	%ERROR	SPBRG value (decimal)	16MHZ KBAUD	%ERROR	SPBRG value (decimal)	10MHZ KBAUD	%ERROR	SPBRG value (decimal)	7.16MHZ KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.615	+0.16	129	9.615	+0.16	103	9.615	+0.16	64	9.520	-0.83	46
19.2	19.230	+0.16	64	19.230	+0.16	51	18.939	-1.36	32	19.454	+1.32	22
38.4	37.878	-1.36	32	38.461	+0.16	25	39.062	+1.7	15	37.286	-2.90	11
57.6	56.818	-1.36	21	58.823	+2.12	16	56.818	-1.36	10	55.930	-2.90	7
115.2	113.636	-1.36	10	111.111	-3.55	8	125	+8.51	4	111.860	-2.90	3
250	250	0	4	250	0	3	NA	-	-	NA	-	-
625	625	0	1	NA	-	-	625	0	0	NA	-	-
1250	1250	0	0	NA	-	-	NA	-	-	NA	-	-

BAUD RATE (K)	Fosc = 5.068MHZ KBAUD	%ERROR	SPBRG value (decimal)	3.579MHZ KBAUD	%ERROR	SPBRG value (decimal)	1MHZ KBAUD	%ERROR	SPBRG value (decimal)	32.768KHZ KBAUD	%ERROR	SPBRG value (decimal)
9.6	9.6	0	32	9.727	+1.32	22	8.928	-6.99	6	NA	-	-
19.2	18.645	-2.94	16	18.643	-2.90	11	20.833	+8.51	2	NA	-	-
38.4	39.6	+3.12	7	37.286	-2.90	5	31.25	-18.61	1	NA	-	-
57.6	52.8	-8.33	5	55.930	-2.90	3	62.5	+8.51	0	NA	-	-
115.2	105.6	-8.33	2	111.860	-2.90	1	NA	-	-	NA	-	-
250	NA	-	-	223.721	-10.51	0	NA	-	-	NA	-	-
625	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1250	NA	-	-	NA	-	-	NA	-	-	NA	-	-

© 1995 Microchip Technology Inc.

12.1.1 SAMPLING

The data on the RX pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. If BRGH is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a 16x clock (see Figure 12-3). If BRGH is set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a 4x clock (see Figure 12-4 and Figure 12-5).

FIGURE 12-3: RX PIN SAMPLING SCHEME (BRGH=0)

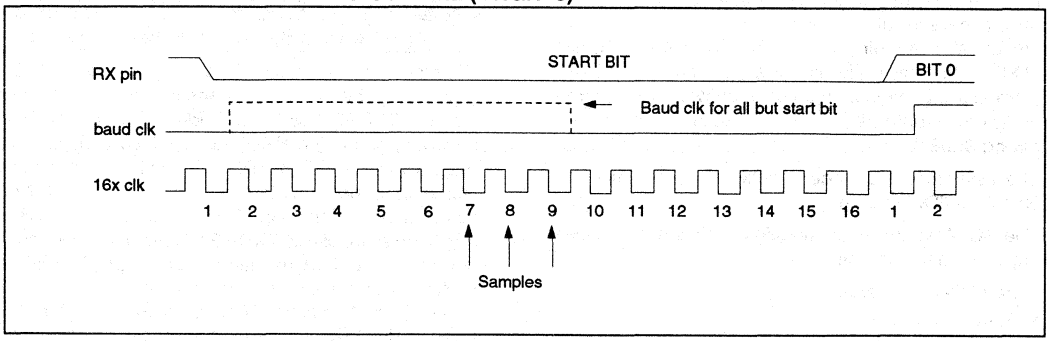


FIGURE 12-4: RX PIN SAMPLING SCHEME (BRGH=1)

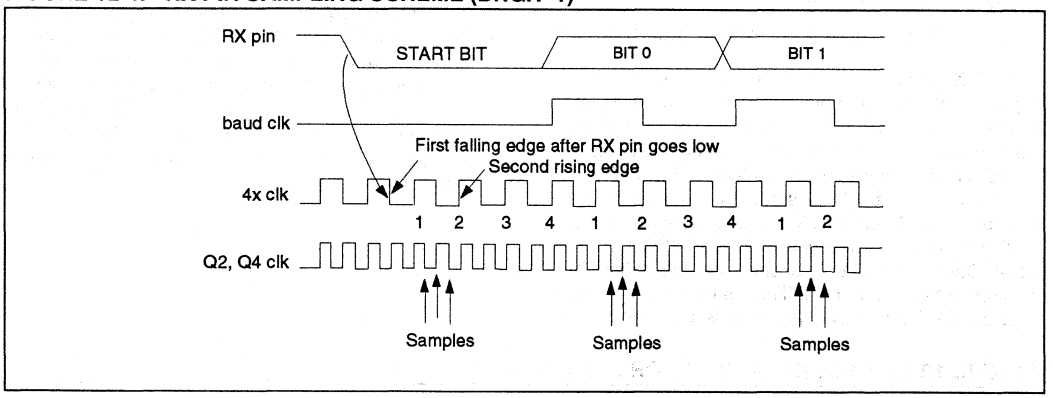
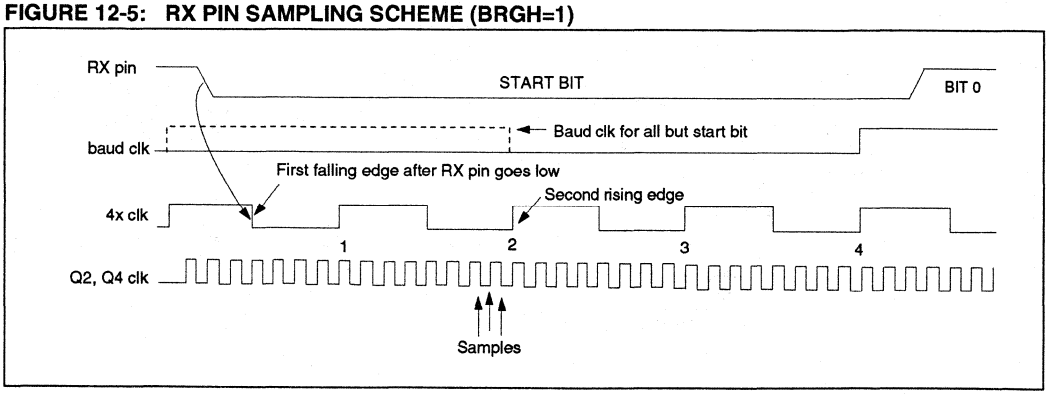


FIGURE 12-5: RX PIN SAMPLING SCHEME (BRGH=1)



12.2 SCI Asynchronous Mode

In this mode, the SCI used standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The SCI transmits and receives the LSB bit first. The SCI's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock either 16x or 64x of the bit shift rate, depending on the BRGH (TXSTA<2>) bit. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The asynchronous mode is selected by resetting the SYNC bit (TXSTA<4>).

The SCI Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

12.2.1 SCI ASYNCHRONOUS TRANSMITTER

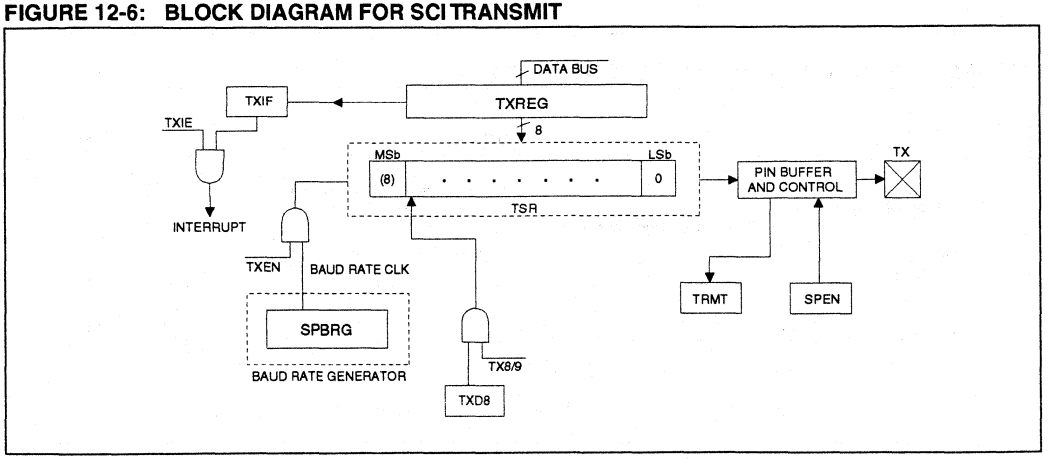
The SCI transmitter block diagram is shown in Figure 12-6. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). The TXREG is loaded with data in software. The TSR is not loaded until the stop bit has been transmitted from the previous load. As soon as the stop bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG transfers the data to the TSR (occurs in one TCY), the TXREG is empty and a interrupt bit, TXIF (PIR1<4>) is set. This interrupt can be enabled or disabled by the TXIE bit (PIE1<4>). TXIF will be set regardless of TXIE and cannot be reset in software. It will reset only when new data is loaded into the TXREG. While TXIF indicated the status of the TXREG, another bit TRMT(TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

Note: The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until the TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 12-6). The transmission can also be started by first loading the TXREG and then setting TXEN. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 12-8). Clearing TXEN during a transmission, will cause the transmission to be aborted and will reset the transmitter as a result the TX pin will revert to hi-impedance.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to the TXD8(TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). In such a case, an incorrect ninth data bit maybe loaded in the TSR.

FIGURE 12-6: BLOCK DIAGRAM FOR SCI TRANSMIT



© 1995 Microchip Technology Inc.

Steps to follow when setting up a Asynchronous Transmission:

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set BRGH to 1. See Section 12.1600 for details.
- Enable the asynchronous serial port by configuring the bits SYNC = 0 and SPEN = 1.
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9 bit should be set.
- Enable the transmission by setting TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Load data to the TXREG (starts transmission).

FIGURE 12-7: ASYNCHRONOUS MASTER TRANSMISSION

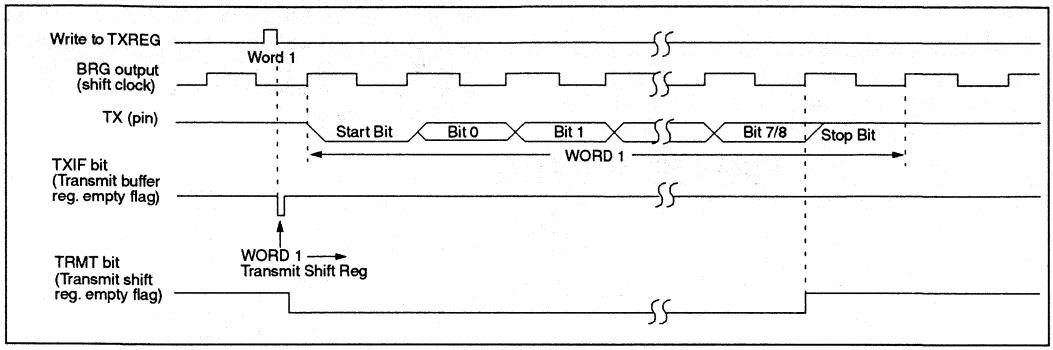
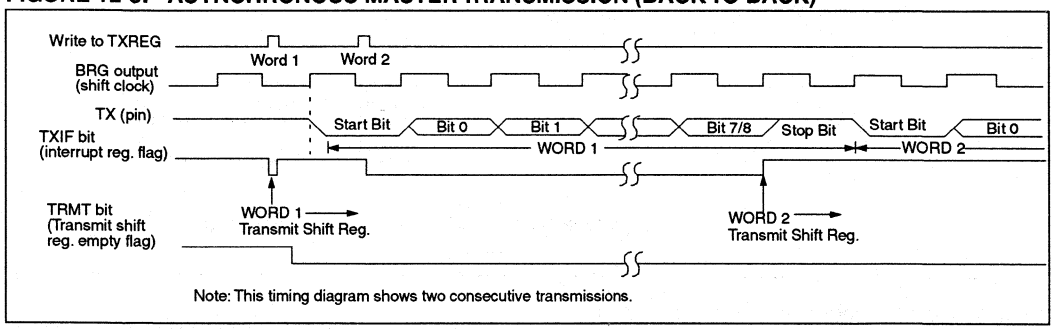


FIGURE 12-8: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD6
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

12.2.2 SCI ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 12-9. The data comes in the RX pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once the asynchronous mode is selected, reception is enabled by setting CREN(RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the stop bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled or disabled by the RCIE(PIE1<5>) bit. The RCIF is a read only bit which is reset by the hardware. It is cleared when the RCREG has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On the detection of the stop bit of the third byte, if the RCREG is still full then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software and this is done by resetting the receive logic (CREN is cleared and then set). If the OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR if it is set. The framing error bit FERR (RCSTA<2>) is set if a stop bit is detected as a 0. The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG, will load the RCD8 and the FERR with new values, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RCD8 information.

FIGURE 12-9: BLOCK DIAGRAM FOR SCI RECEIVE

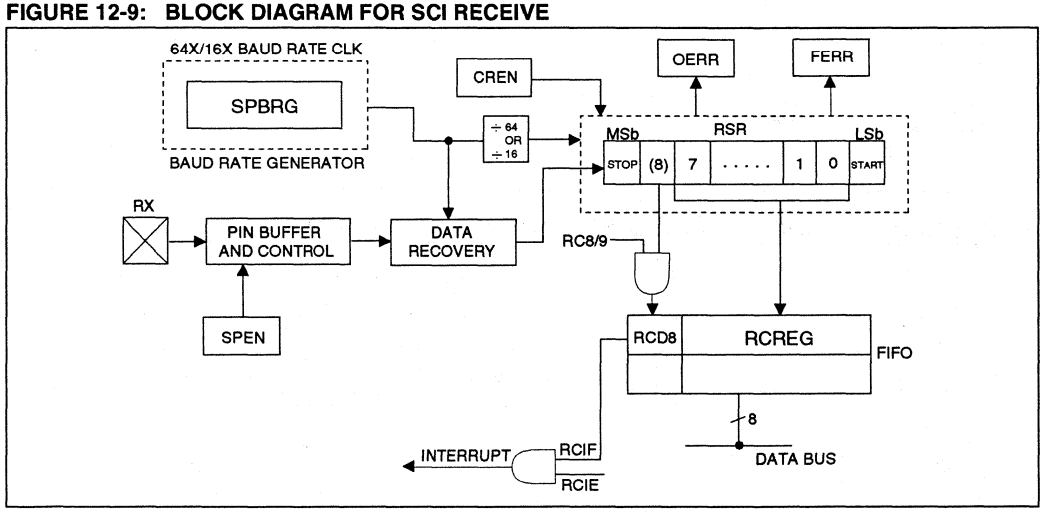
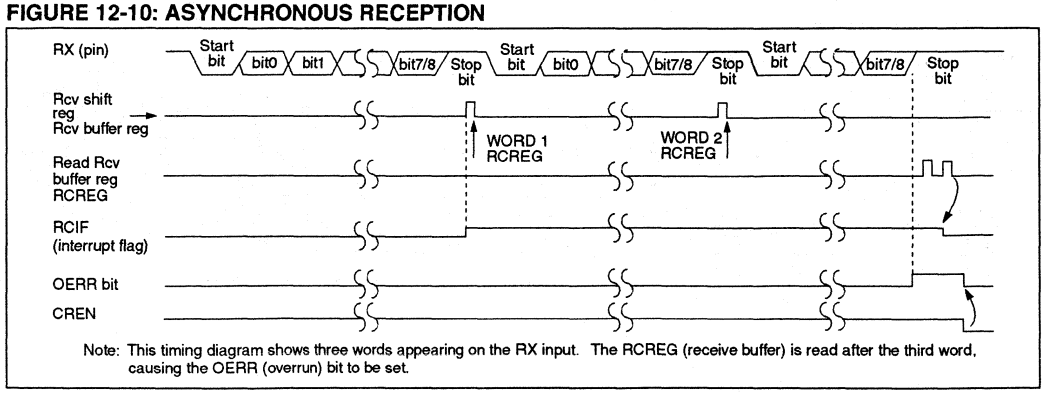


FIGURE 12-10: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

© 1995 Microchip Technology Inc.

Steps to follow when setting up a Asynchronous Reception:

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set BRGH. See Section 12.1600 for details.
- Enable the asynchronous serial port by configuring the SYNC = 0 and SPEN = 1.
- If interrupts are desired, then the RCIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- Enable the reception by setting CREN.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
0x18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
0x1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
0x8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
0x98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
0x99h	SPBRG	Baud Rate Register							

† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

12.3 SCI Synchronous Master Mode

In Master Synchronous mode the data is transmitted in a half-duplex manner i.e. transmission and reception do not occur at the same time, when transmitting data, the reception is inhibited and vice versa. The synchronous mode is entered by setting the SYNC (TXSTA<4>) bit. In addition the SPEN (RCSTA<7>) bit is set in order to configure the RC6 and RC7 I/O ports to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC (TXSTA<7>) bit.

12.3.1 SCI SYNCHRONOUS MASTER TRANSMISSION

The SCI transmitter block diagram is shown in Figure 12-6. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer TXREG. The TXREG is loaded with data in software. The TSR is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG transfers the data to the TSR (occurs in one Tcycle), the TXREG is empty and a interrupt bit, TXIF (PIR1<4>) is set. This interrupt can be enabled or disabled by the TXIE bit (PIE1<4>). TXIF will be set regardless of TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG. While TXIF indicates the status of the TXREG, another bit TRMT(TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until the TXREG has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on CK. Data out is stable around the falling edge of the synchronous clock (Figure 12-11). The transmission can also be started by first loading the TXREG and then setting TXEN. This is advantageous when slow baud rates are selected, since the BRG is kept in reset when TXEN=CREN=SREN = 0. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing TXEN during a transmission, will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either CREN or SREN are set, during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if CSRC=1(internal clock). The transmitter logic however is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear TXEN. If the SREN bit is set (to interrupt an on going transmission and receive a single word), then after the single word is received, the SREN will = 0 and the serial port will revert back to transmitting since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, TXEN should be cleared.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to TXD8(TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the "new" TXD8, the "present" value of TXD8 is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

- Initialize the SPBRG register for the appropriate baud rate (see Section 12.1600 for details)
- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9-bit should be set.
- Enable the transmission by setting TXEN to 1.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Start transmission by loading data to the TXREG.

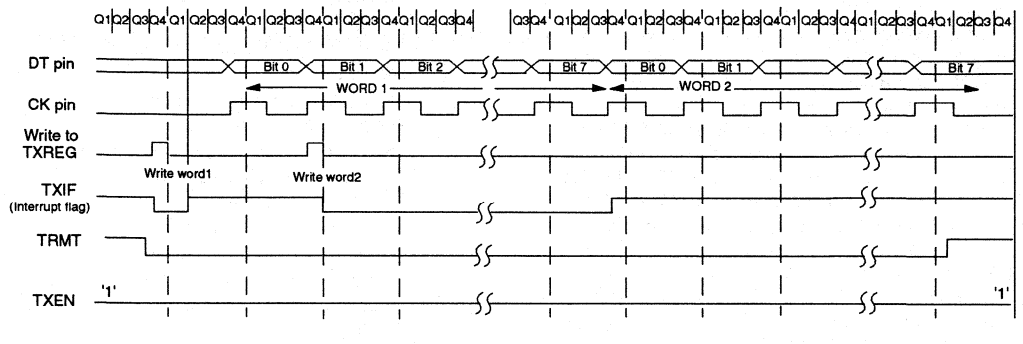
© 1995 Microchip Technology Inc.

TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

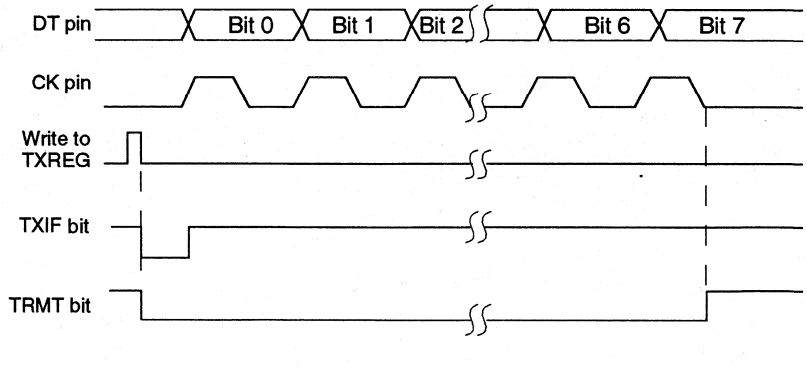
† This bit is reserved on the PIC16C73.

FIGURE 12-11: SYNCHRONOUS TRANSMISSION



Note: Sync master mode; SPBRG = 0. Continuous transmission of two 8 bit words

FIGURE 12-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



12.3.2 SCI SYNCHRONOUS MASTER RECEPTION

Once the synchronous mode is selected, reception is enabled by setting either the SREN(RCSTA<5>) bit or the CREN(RCSTA<4>). Data is sampled on the DT pin on the falling edge of the clock. If SREN is set, then only a single word is received, if CREN is set, the reception is continuous until CREN is reset. If both the bit are set then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register(RSR) is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled or disabled by the RCIE(PIE1<5>) bit. The RCIF is a read only bit which is reset by the hardware. In this case it is reset when the RCREG has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR. On the clocking of the last bit of the third byte, if the RCREG is still full then the overrun error bit, OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software and this is done by clearing CREN. If the OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear the OERR if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG, will load the RCD8 with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RCD8 information.

Steps to follow when setting up a Synchronous Master Reception:

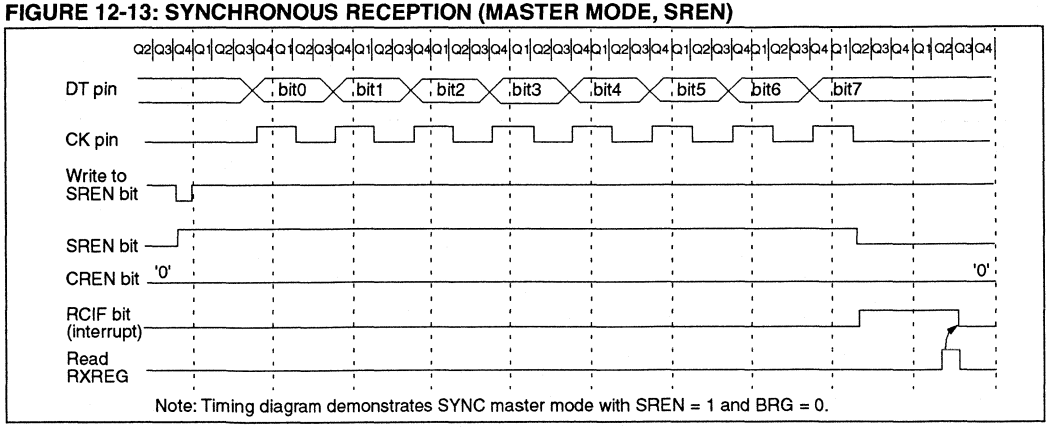
- Initialize the SPBRG register for the appropriate baud rate. See section 12.1 for details.
- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
- CREN= SREN = 0
- If interrupts are desired, then the RXIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- If a single reception is required, set SREN. For continuous reception set CREN.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD6
19h	PCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† This bit is reserved on the PIC16C73.

FIGURE 12-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates SYNC master mode with SREN = 1 and BRG = 0.

© 1995 Microchip Technology Inc.

12.4 SCI Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC(TXSTA<7>) bit.

12.4.1 SCI SYNCHRONOUS SLAVE TRANSMIT

The operation of the sync master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction executed, the following will occur. The first word will immediately transfer to the TSR and transmit. The second word will remain in TXREG. The TXIF will not be set. When the first word has been shifted out of TSR, the TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If the TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
- Make CREN = SREN = 0
- If interrupts are desired, then the TXIE bit should be set.
- If 9-bit transmission is desired, then TX8/9-bit should be set.
- Enable the transmission by setting TXEN to 1.
- If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
- Start transmission by loading data to the TXREG.

12.4.2 SCI SYNCHRONOUS SLAVE RECEPTION

The operation of the synchronous master and slave modes is identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to the RCREG and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

- Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
- If interrupts are desired, then the RCIE bit should be set.
- If 9-bit reception is desired, then RX8/9-bit should be set.
- To enable reception, set CREN = 1.
- The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
- Read the RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
- Read the 8-bit received data by reading the RCREG.
- If any error occurred, clear the error by clearing CREN.

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
19h	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† This bit is reserved on the PIC16C73.

TABLE 12-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0Ch	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
18h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0
8Ch	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
98h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	BRGH	TRMT	TXD8
99h	SPBRG	Baud Rate Register							

† This bit is reserved on the PIC16C73.

© 1995 Microchip Technology Inc.

13.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 8-bit digital number (refer to Application Note AN546 for use of A/D Converter). The A/D module has eight analog inputs for the PIC16C74, five for the PIC16C73 and four for the PIC16C71, which are multiplexed into one sample and hold. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (VDD) or the voltage level on the RA3/AN3/VREF pin. The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode.

The A/D module has three registers. These registers are:

- A/D result register (ADRES)
- A/D control register 0 (ADCON0)
- A/D control register 1 (ADCON1)

The ADCON0 register, shown in Figure 13-1 and Figure 13-2, controls the operation of the A/D module. The ADCON1 register, shown in Figure 13-3 and Figure 13-4, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or as digital I/O.

2

FIGURE 13-1: ADCON0 REGISTER FOR PIC16C74 AND PIC16C73 ONLY

R/W	R/W	R/W	R/W	R/W	R/W	U	R/W
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	–	ADON
bit7							bit0

Register:	ADCON0	W: Writable bit
Address:	1Fh	R: Readable bit
POR value:	00h	U: Unimplemented, read as '0'

ADON: A/D on bit
1 = A/D converter module is operating.
0 = A/D converter module is shut off and consumes no operating current.

Reserved.

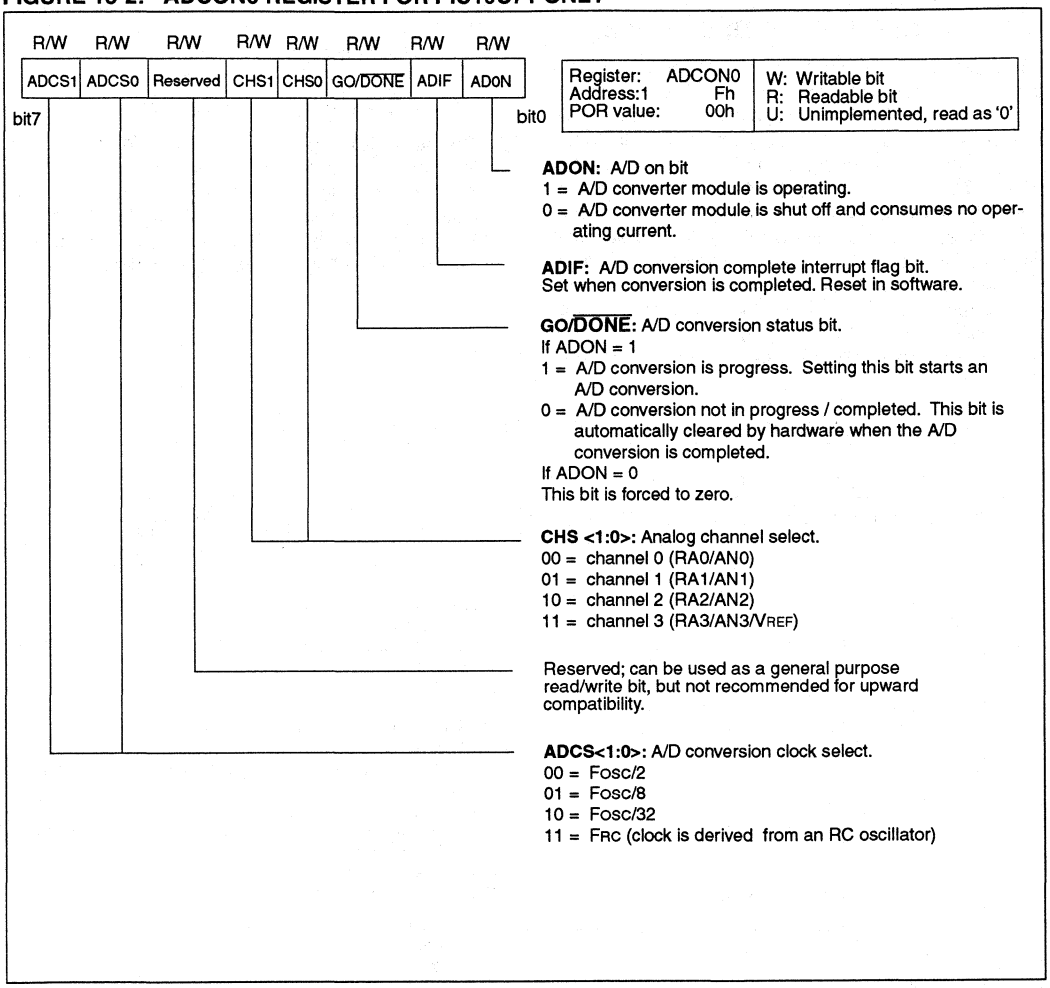
GO/$\overline{\text{DONE}}$: A/D conversion status bit.
If ADON = 1
1 = A/D conversion is progress. Setting this bit starts an A/D conversion.
0 = A/D conversion not in progress / completed. This bit is automatically cleared by hardware when the A/D conversion is completed.
If ADON = 0
This bit is forced to zero.

CHS <2:0>: Analog channel select.
000 = channel 0 (RA0/AN0)
001 = channel 1 (RA1/AN1)
010 = channel 2 (RA2/AN2)
011 = channel 3 (RA3/AN3/VREF)
100 = channel 4 (RA5/AN4)
101 = channel 5 (RE0/AN5)†
110 = channel 6 (RE1/AN6 †
111 = channel 7 (RE2/AN7) †

ADCS<1:0>: A/D conversion clock select.
00 = Fosc/2
01 = Fosc/8
10 = Fosc/32
11 = FRC (clock is derived from an RC oscillator)

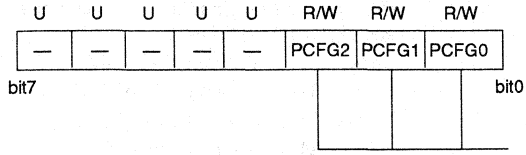
† Not available in PIC16C73

FIGURE 13-2: ADCON0 REGISTER FOR PIC16C71 ONLY



© 1995 Microchip Technology Inc.

FIGURE 13-3: ADCON1 REGISTER FOR PIC16C74 AND PIC16C73 ONLY



Register:	ADCON1	W:	Writable
Address:	9Fh	R:	Readable
POR value:	00h	U:	Unimplemented, read as '0'

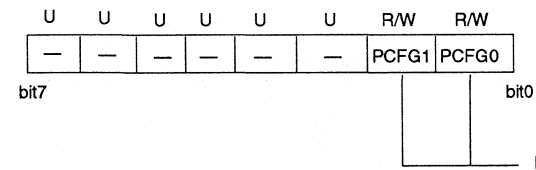
PCFG<2:0>: A/D port configuration bits. These bits configure the analog port pins to the various modes of operation.

PCFG<2:0>	RA0	RA1	RA2	RA5	RA3	RE0	RE1	RE2	Ref
000	A	A	A	A	A	A	A	A	VDD
001	A	A	A	A	VREF	A	A	A	RA3
010	A	A	A	A	A	D	D	D	VDD
011	A	A	A	A	VREF	D	D	D	RA3
100	A	A	D	D	A	D	D	D	VDD
101	A	A	D	D	VREF	D	D	D	RA3
11X	D	D	D	D	D	D	D	D	–

A = Analog input

D = Digital input/output depending on corresponding TRIS bit

FIGURE 13-4: ADCON1 REGISTER FOR PIC16C71 ONLY



Register:	ADCON1	W:	Writable
Address:	08h	R:	Readable
POR value:	00h	U:	Unimplemented, read as '0'

PCFG<1:0>: A/D port configuration bits. These bits configure the analog port pins to the various modes of operation.

PCFG<1:0>	RA0	RA1	RA2	RA3	Ref
00	A	A	A	A	VDD
01	A	A	A	VREF	RA3
10	A	A	D	D	VDD
11	D	D	D	D	VDD

A = Analog input

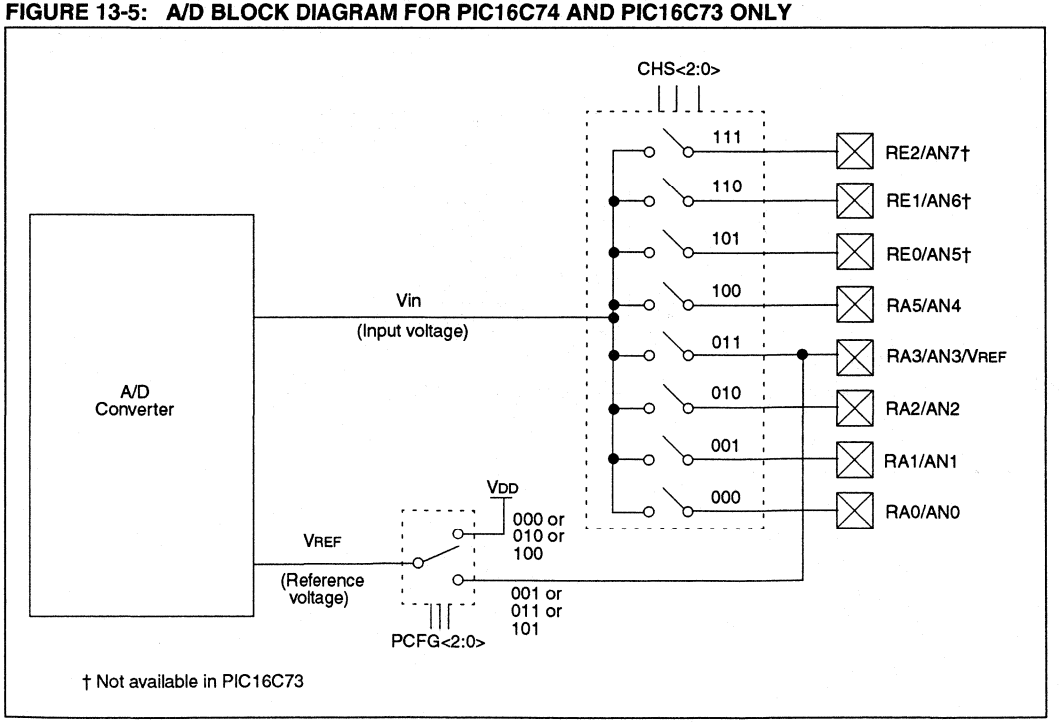
D = Digital input/output depending on corresponding TRIS bit

The ADRES register contains the result of the A/D conversion. When the A/D conversion is completed, the result is loaded into the ADRES, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and the A/D interrupt flag (ADIF) is set. The block diagrams of the A/D module are shown in Figure 13-5 and Figure 13-6.

After the A/D module has been configured as desired, the selected channel must be sampled before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. The A/D bit sample time is defined as TAD. To determine sample time see Section 13.1618. After this sample time has elapsed the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

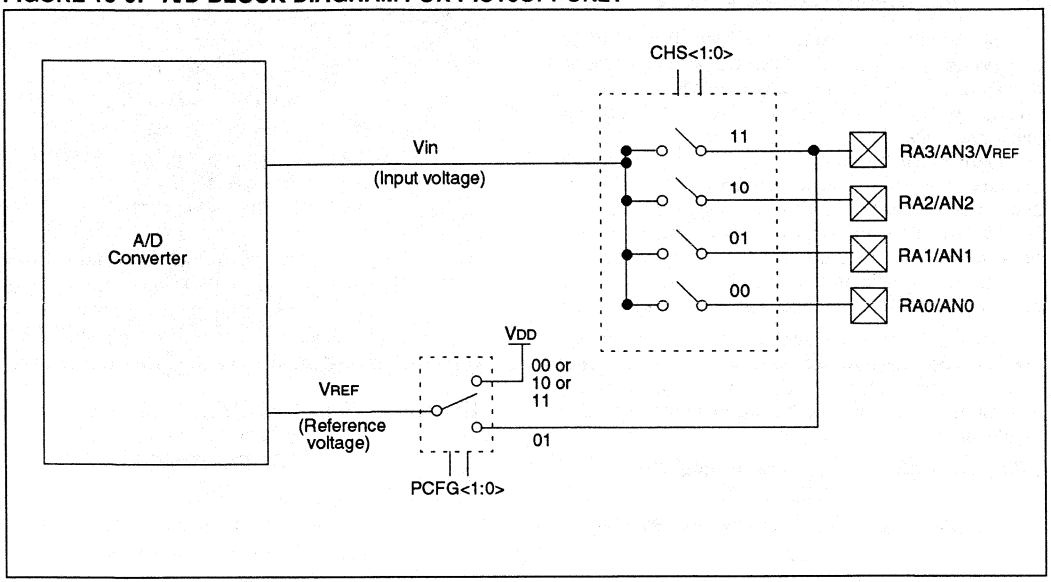
1. Configure the A/D module
   - Configure analog pins / voltage reference / and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if required)
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required sampling time
4. Start conversion
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared
   - Waiting for the A/D interrupt
6. Read A/D Result register (ADRES), clear ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. A minimum wait of 2TAD is required before next sampling starts.

FIGURE 13-5: A/D BLOCK DIAGRAM FOR PIC16C74 AND PIC16C73 ONLY



© 1995 Microchip Technology Inc.

FIGURE 13-6: A/D BLOCK DIAGRAM FOR PIC16C71 ONLY



© 1995 Microchip Technology Inc.

13.1 A/D Sampling Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 13-7. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 13-7. **The maximum recommended impedance for analog sources is 10-KΩ.** After the analog input channel is selected (changed) this sampling must be done before the conversion can be started.

To calculate the minimum sampling time, Equation 13-1 may be used. This equation assumes that 1/2 LSb error is used (512 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

EQUATION 13-1: A/D SAMPLING TIME EQUATION

$$V_{REF} = (V_{REF} - V_{REF}/512) \bullet (1 - e^{(-t/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

or

$$t = -51.2\ pF\ (1K\Omega + R_{SS} + R_S)\ \ln(1/511)$$

Example 13-1 shows the calculation of the minimum required sample time. This calculation is based on the following system assumptions:

RS = 10-KΩ

1/2 LSb error

VDD = 5V → RSS = 7-KΩ

Temp (system Max.) = 50°C

Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (Chold) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10-KΩ. This is required to meet the pin leakage specification.

4: After a conversion has completed, 2 TAD time must be waited before sampling can begin again. During this time the holding capacitor is not connected to the selected A/D input channel.

EXAMPLE 13-1: CALCULATING THE MINIMUM REQUIRED SAMPLE TIME

Sampling Time = Amplifier Settling Time
+ Holding Capacitor Charging Time
+Temperature Coefficient †

= 5µs + t + [(Temp - 25•C) (0.05µs/°C)]†

t = -CHOLD (RIC + RSS + RS) ln (1/511)
-51.2 pF (8-KΩ + 10-KΩ) ln (0.0020)
-51.2 pF (18-KΩ) ln (-0.0020)
-0.921µs (-6.2364)
5.724µs

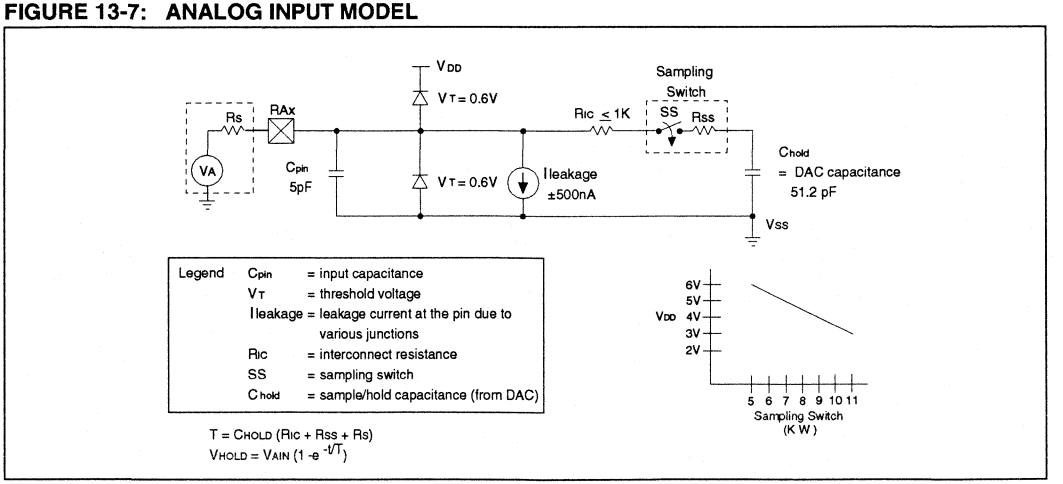
Sampling Time = 5µs + 5.724µs + [(50°C - 25°C) (0.05µs / °C)]

= 10.724µs + 1.25µs

= 11.974µs

† The temperature coefficient is only required for temperatures > 25°C.

FIGURE 13-7: ANALOG INPUT MODEL



© 1995 Microchip Technology Inc.

13.2 Selecting the A/D Conversion Clock

The A/D conversion requires 10 TAD. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2 TOSC
- 8 TOSC
- 32 TOSC
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6µs for the PIC16C74 and PIC16C73, and 2.0 µs for the PIC16C71. Table 13-1 and Table 13-2 show the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

13.3 Configuring Analog Port Pins

The use of the ADCON1, TRISA, and TRISE registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

Note 1: When reading the PORT register, all pins configured as analog input channel will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

2: Analog levels on any pin that is defined as a digital input (including the AN7-0 pins), may cause the input buffer to consume current that is out of the devices specification.

2

TABLE 13-1: TAD VS. DEVICE OPERATING FREQUENCIES FOR PIC16C74/PIC16C73 ONLY

AD Clock Source (TAD)		Device Frequency			
Operation	**ADCS1:ADCS0**	**20 MHz**	**5 MHz**	**1.25 MHz**	**333.33 KHz**
2 TOSC	00	100 ns§	400 ns§	1.6 µs	6 µs
8 TOSC	01	400 ns§	1.6 µs	6.4 µs	24 µs†
32 TOSC	10	1.6 µs	6.4 µs	25.6 µs†	96 µs†
RC	11	2 - 6 µs*	2 - 6 µs*	2 - 6 µs*	2 - 6 µs*

* The RC source has a typical TAD time of 4 µs.
§ These values violate the minimum required TAD time.
† For faster conversion times, the selection of another clock source is recommended.

TABLE 13-2: TAD VS. DEVICE OPERATING FREQUENCIES FOR PIC16C71 ONLY

AD Clock Source (TAD)		Device Frequency				
Operation	**ADCS1:ADCS0**	**20 MHz**	**16 MHz**	**4 MHz**	**1 MHz**	**333.33 KHz**
2 TOSC	00	100 ns§	125 ns§	500 ns§	2.0 µs	6 µs
8 TOSC	01	400 ns§	500 ns§	2.0 µs	8.0 µs	24 µs†
32 TOSC	10	1.6 µs§	2.0 µs	8.0 µs	32.0 µs†	96 µs†
RC	11	2 - 6 µs*	2-6 µs	2 - 6 µs*	2 - 6 µs*	2 - 6 µs*

* The RC source has a typical TAD time of 4 µs.
§ These values violate the minimum required TAD time.
† For faster conversion times, the selection of another clock source is recommended.

© 1995 Microchip Technology Inc.

13.4 A/D Conversions

Example 13-2 and Example 13-3 show how to perform an A/D conversion. The RA pins are configured as analog inputs. The analog reference (VREF) is the device VDD. The A/D interrupt is enabled, and the A/D conversion clock is TRC. The conversion is performed on the RA0 channel.

EXAMPLE 13-2: DOING AN A/D CONVERSION (PIC16C74/73)

```
   BSF   STATUS, RP0  ;Select Page 1
   CLRF  ADCON1       ;Configure A/D Inputs
   BSF   PIE1, ADIE   ;Enable A/D interrupt
   BCF   STATUS, RP0  ;Select Page 0
   MOVLW 0xC1         ;RC clock, A/D is on,
                      ;Ch 0 is selected
   MOVWF ADCON0       ;
   BCF   PIR1, ADIF   ;Clear A/D Int Flag
   BSF   INTCON, PEIE ;Enable Peripheral
   BSF   INTCON, GIE  ;Enable All Interrupts
;
; Ensure that the required sampling time for the
; selected input channel has lapsed. Then the
; conversion may be started.
   BSF   ADCON0, GO   ;Start A/D Conversion
      :               The ADIF bit will be
                      set and the GO/DONE bit
      :               cleared upon comple-
                      tion of the A/D conversion.
```

EXAMPLE 13-3: DOING AN A/D CONVERSION (PIC16C71)

```
   BSF   STATUS, RP0  ;Select Page 1
   CLRF  ADCON1       ;Configure A/D Inputs
   BCF   STATUS, RP0  ;Select Page 0
   MOVLW 0xC1         ;RC clock, A/D is on,
                      ;Ch 0 is selected
   MOVWF ADCON0       ;
   BSF   INTCON,ADIE  ;Enable A/D Interrupt
                      ;Interrupts
   BSF   INTCON, GIE  ;Enable All Interrupts
;
; Ensure that the required sampling time for the
; selected input channel has lapsed. Then the
; conversion may be started.
   BSF   ADCON0, GO   ;Start A/D Conversion
      :               The ADIF bit will be
                      set and the GO/DONE bit
      :               cleared upon comple-
                      tion of the A/D conversion.
```

Note: The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The ADRES will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, the 2TAD wait is required before the next sampling is started. After this 2TAD wait, sampling is automatically started on the selected channel.

13.4.1 FASTER CONVERSION - LOWER RESOLUTION TRADE-OFF

Not all applications require a result with 8-bits of resolution, but may instead require a faster conversion time. The A/D module allows users to make the trade off of conversion speed to resolution. Regardless of the resolution required, the sampling time is the same. To speed up the conversion, the clock source of the A/D module may be switched so that the TAD time violates the minimum specified time (see parameter #130). Once the TAD time violates the minimum specified time, all the following A/D result bits are not valid (see A/D Conversion Timing in the Electrical Specifications section.) The clock sources may only be switched between the three oscillator versions (cannot be switched from/to RC). The equation to determine the time before the oscillator can be switched is as follows:

Conversion time = $2T_{AD} + N \cdot T_{AD} + (8 - N)(2\ T_{OSC})$
Where: N = number of bits of resolution required

Since the TAD is based from the device oscillator, the user must use some method (a timer, software loop, etc) to determine when the A/D oscillator may be changed. Example 13-4 shows a comparison of time required for a conversion with 4-bits of resolution, versus the 8-bit resolution conversion. The example is for devices operating at 20 MHz and 16 MHz (The A/D clock is programmed for 32 TOSC), and assumes that immediately after 6 Tad, the A/D clock is programmed for 2TOSC.

The 2 TOSC violates the minimum TAD time since the last 4-bits will not be converted to correct values.

EXAMPLE 13-4: 4-BIT VS. 8-BIT CONVERSION TIMES

	Freq. (MHz)*	Resolution	
		4-bit	8-bit
TAD	20	1.6 µs	1.6 µs
	16	2.0 µs	2.0 µs
TOSC	20	50 ns	50 ns
	16	62.5 ns	62.5 ns
$2T_{AD}+N \cdot T_{AD}+(8-N)(2T_{OSC})$	20	10 µs	16 µs
	16	12.5 µs	20 µs

*The PIC16C74/73 A/D has a minimum TAD time of 1.6 µs, the PIC16C71 has a minimum TAD time of 2.0 µs.

© 1995 Microchip Technology Inc.

13.5 A/D Operation During Sleep

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS<1:0> = 11b). When the RC clock source is selected the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed the GO/$\overline{\text{DONE}}$ bit will be cleared, and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

Note: For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS<1:0>=11b).

13.6 A/D Accuracy/Error

The overall accuracy of the A/D is less than ± 1 LSb for V_{DD} = 5 V ± 10% and the analog V_{REF} = V_{DD}. This overall accuracy includes offset error, full scale error, and integral error. The A/D converter is guaranteed to be monotonic. The resolution and accuracy may be less when either the analog reference (V_{DD}) is less then 5.0V or when the analog reference (V_{REF}) is less then V_{DD}.

The maximum pin leakage current is ± 5 µA.

In systems where the device frequency is low, use of the A/D RC clock derived from the device oscillator is preferred. At moderate to high frequencies, T_{AD} should be derived from the device oscillator. T_{AD} must not violate the minimum and should be ≤8µs for preferred operation. This is because T_{AD}, when derived from T_{OSC} is kept away from on-chip phase clock transitions. This reduces, to a large extent, the effects of digital switching noise. This is not possible with the RC derived clock. The loss of accuracy due to digital switching noise can be significant if many I/O pins are active.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives the highest accuracy.

© 1995 Microchip Technology Inc.

13.7 Effects of a RESET

A device reset forces all register to their reset state. This forces the A/D module to be turned off, and any conversion is aborted. The value that is in the ADRES register is not modified for a power-on reset. The ADRES register will contain unknown data after a power-up reset.

13.8 Use of the CCP2 Trigger for PIC16C74 and PIC16C73 only

An A/D conversion can be started by the "special event trigger" of the CCP2 module. This requires that the CCP2M<3:0> bits (CCP2CON<3:0>) be programmed as 1011b and that the A/D module is enabled (ADON is set). When the trigger occurs, the GO/$\overline{\text{DONE}}$ bit will be set to start the A/D conversion and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D sampling period, with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum sampling done before the "special event trigger" sets the GO/$\overline{\text{DONE}}$ bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the "special event trigger" will be ignored by the A/D module, but will still reset the Timer1 counter.

© 1995 Microchip Technology Inc.

13.9 Connection Considerations

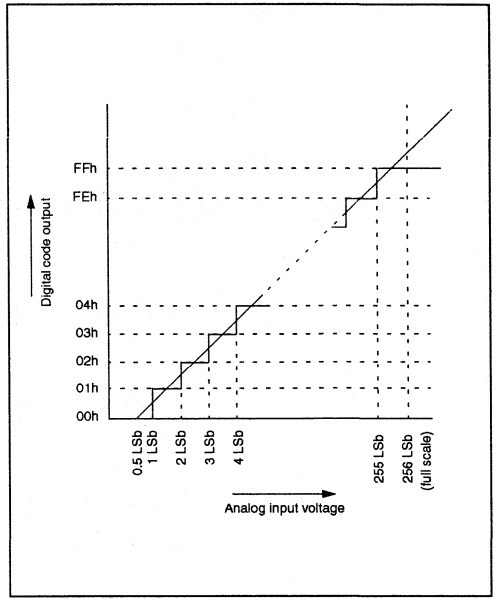
Since the analog inputs employ ESD protection, they have reversed biased diodes to VDD and VSS. This requires that the analog input must be between VDD and VSS. If the input voltage exceeds this range by greater than 0.6V (either direction), one of the diodes becomes forward biased and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 KΩ recommended specification. Any external components connected (via high impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

13.10 Transfer Function

The ideal transfer function of the A/D converter is as follows: The first transition occurs when the analog input voltage (VAIN) is 1 LSb (or Analog VREF / 256). This is shown in Figure 13-8.

FIGURE 13-8: A/D TRANSFER FUNCTION



© 1995 Microchip Technology Inc.

FIGURE 13-9: FLOWCHART OF A/D OPERATION

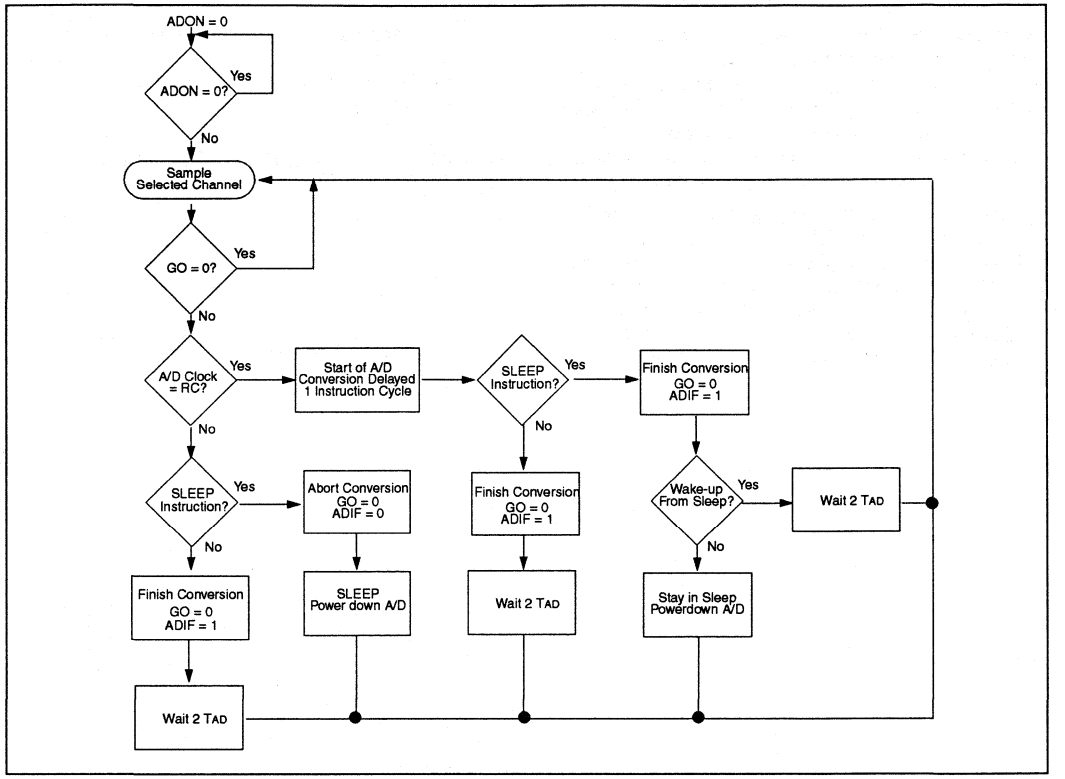


TABLE 13-3: SUMMARY OF A/D REGISTERS FOR PIC16C74 AND PIC16C73 ONLY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
0C	PIR1	†PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
8C	PIE1	†PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
0D	PIR2	—	—	—	—	—	—	—	CCP2IF
8D	PIE2	—	—	—	—	—	—	—	CCP2IE
1E	ADRES	A/D Result Register							
1F	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
9F	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0
05	PORTA	PortA Data Latch when written to, PortA pin when read							
85	TRISA	PortA Data Direction Latch							
09	PORTE	PortE Data Latch when written to, PortA pin when read							
89	TRISE	PortE Data direction latch							

† This bit is reserved on the PIC16C73.

TABLE 13-4: SUMMARY OF A/D REGISTERS FOR PIC16C71 ONLY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0B/8B	INTCON	GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
09	ADRES	A/D Result Register							
08	ADCON0	ADCS1	ADCS0	—	CHS1	CHS0	GO/DONE	ADIF	ADON
88	ADCON1	—	—	—	—	—	—	PCFG1	PCFG0
05	PORTA	PortA Data Latch when written to, PortA pin when read							
85	TRISA	PortA Data Direction Latch							

© 1995 Microchip Technology Inc.

14.0 SPECIAL FEATURES OF THE CPU

What sets apart a microcontroller from other processors are special circuits to deal with the needs of real time applications. The PIC16CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

1. OSC selection
2. Reset
   - Power-On Reset (POR)
   - Power-Up Timer (PWRT)
   - Oscillator Start-Up Timer (OST)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID locations
8. In-circuit serial programming

The PIC16CXX has a watchdog timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the oscillator start-up timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the power-up timer (PWRT), which provides a fixed delay of 72ms (nominal) on power up only, designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

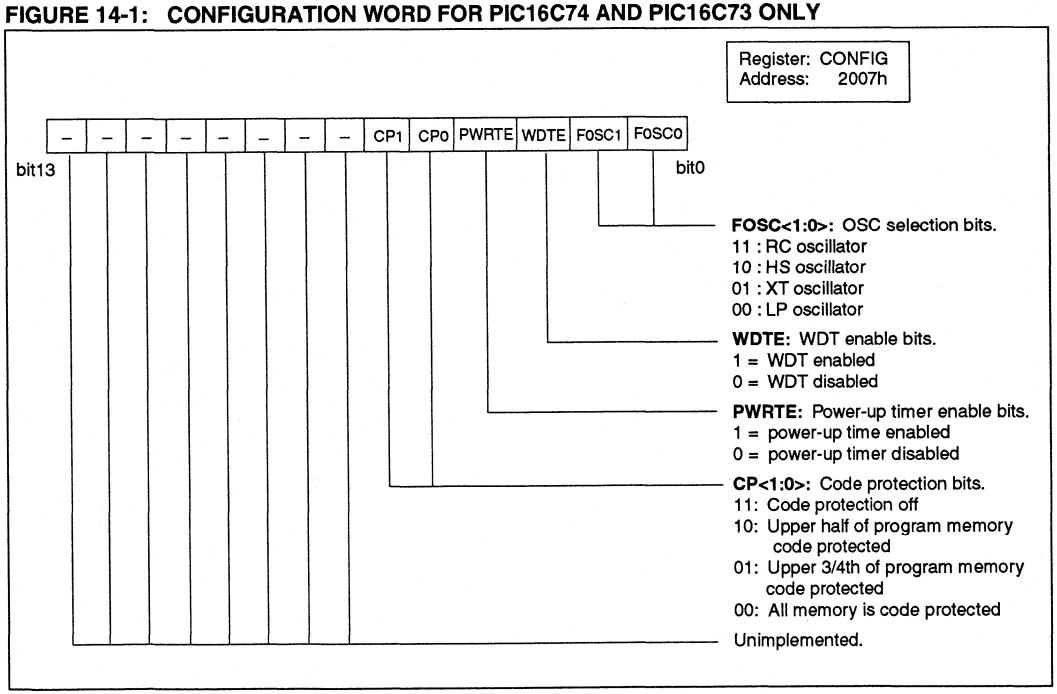
The SLEEP mode is designed to offer a very low current power-down mode. The user can wake up from SLEEP through external reset, watchdog timer time-out or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

14.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

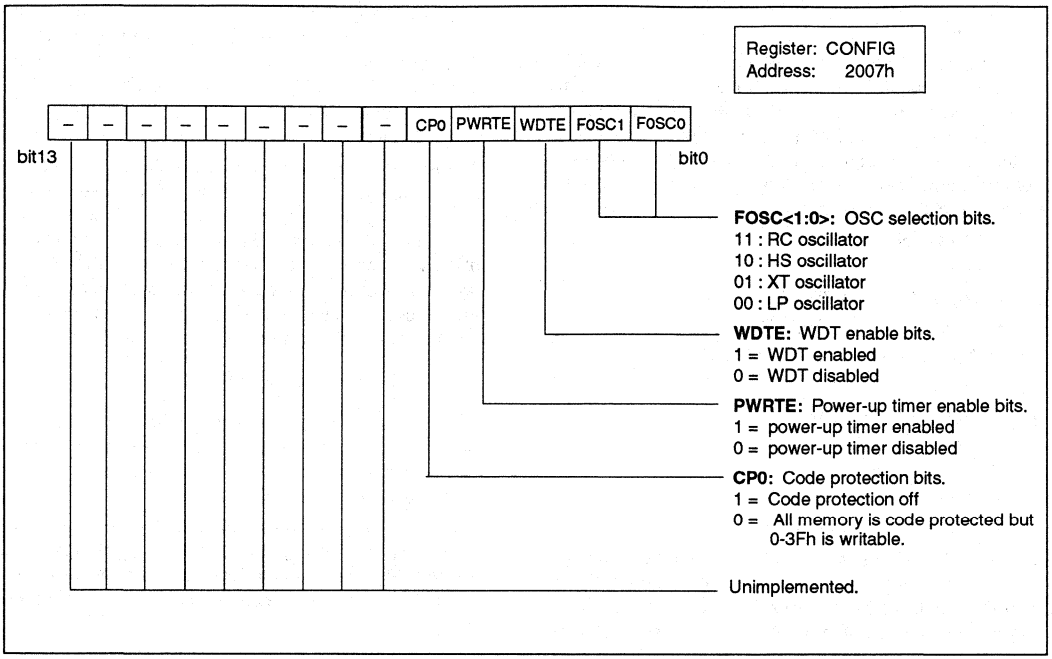
The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

FIGURE 14-1: CONFIGURATION WORD FOR PIC16C74 AND PIC16C73 ONLY



© 1995 Microchip Technology Inc.

FIGURE 14-2: CONFIGURATION WORD FOR PIC16C71 ONLY



© 1995 Microchip Technology Inc.

14.2 Oscillator Configurations

14.2.1 OSCILLATOR TYPES

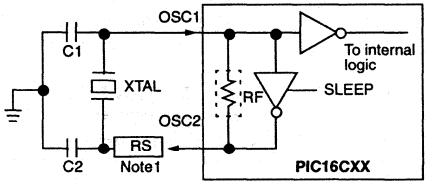
The PIC16CXX can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

14.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 14-3). The PIC16CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin. This is shown in Figure 14-4.

FIGURE 14-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



See Table 14-1, Table 14-2, Table 14-3 and Table 14-4 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

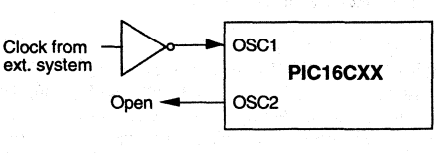


TABLE 14-1: CERAMIC RESONATORS PIC16C74/PIC16C73

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455kHZ	68 - 100 pF	68 - 100 pF
	2.0MHz	15 - 68 pF	15 - 68 pF
	4.0MHz	15 - 68 pF	15 - 68 pF
HS	8.0MHz	10 - 68 pF	10 - 68 pF
	16.0MHz	10 - 22 pF	10 - 22 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.

Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:		
455kHz	Panasonic EFO-A455K04B	+/-.3%
2.0MHz	Murata Erie CSA2.00MG	+/-.5%
4.0MHz	Murata Erie CSA4.00MG	+/-.5%
8.0MHz	Murata Erie CSA8.00MT	+/-.5%
16.0MHz	Murata Erie CSA16.00MX	+/-.5%

All resonators used did not have built-in capacitors.

TABLE 14-2: CERAMIC RESONATORS PIC16C71

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455kHZ	68 - 100 pF	68 - 100 pF
	2.0MHz	15 - 68 pF	15 - 68 pF
	4.0MHz	15 - 68 pF	15 - 68 pF
HS	8.0MHz	10 - 68 pF	10 - 68 pF
	16.0MHz	10 - 22 pF	10 - 22 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.

Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:		
455KHz	Panasonic EFO-A455K04B	+/-.3%
2.0MHz	Murata Erie CSA2.00MG	+/-.5%
4.0MHz	Murata Erie CSA4.00MG	+/-.5%
8.0MHz	Murata Erie CSA8.00MT	+/-.5%
16.0MHz	Murata Erie CSA16.00MX	+/-.5%

All resonators used did not have built-in capacitors.

© 1995 Microchip Technology Inc.

TABLE 14-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C74 AND PIC16C73

Mode	Freq	OSC1	OSC2
LP	32kHz§	33 - 68 pF	33 - 68 pF
	200kHz	15 - 47 pF	15 - 47 pF
XT	100kHz	47 - 100 pF	47 - 100 pF
	500 kHz	20 - 68 pF	20 - 68 pF
	1MHz	15 - 68 pF	15 - 68 pF
	2MHz	15 - 47 pF	15 - 47 pF
	4MHz	15 - 33 pF	15 - 33 pF
HS	8MHz	15 - 47 pF	15 - 47 pF
	20MHz	15 - 47 pF	15 - 47 pF

Note: Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

§ For VDD > 4.5V, C1 = C2 ≈ 30pf is recommended.

TABLE 14-4: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR FOR PIC16C71 ONLY

Mode	Freq	OSC1	OSC2
LP	32kHz§	33 - 68 pF	33 - 68 pF
	200kHz	15 - 47 pF	15 - 47 pF
XT	100kHz	47 - 100 pF	47 - 100 pF
	500kHz	20 - 68 pF	20 - 68 pF
	1MHz	15 - 68 pF	15 - 68 pF
	2MHz	15 - 47 pF	15 - 47 pF
	4MHz	15 - 33 pF	15 - 33 pF
HS	8MHz	15 - 47 pF	15 - 47 pF
	20MHz	15 - 47 pF	15 - 47 pF

Note: Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

§ For VDD > 4.5V, C1 = C2 ≈ 30pf is recommended.

14.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

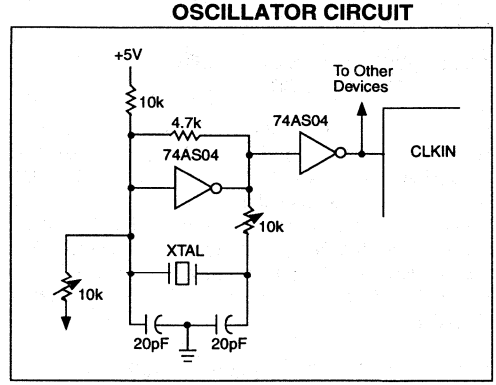
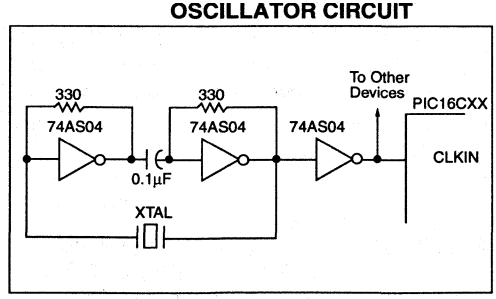


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



14.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low

© 1995 Microchip Technology Inc.

Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-7 shows how the R/C combination is connected to the PIC16CXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

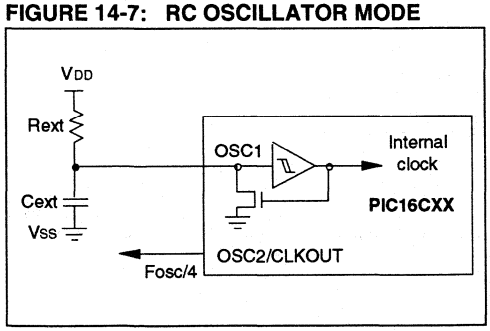
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See characterization data for desired device for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characterization data for desired device for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-4 for waveform).

FIGURE 14-7: RC OSCILLATOR MODE



14.3 Reset

The PIC16CXX differentiates between various kinds of reset:

a) Power-on reset ($\overline{POR}$)
b) $\overline{MCLR}$ reset during normal operation
c) $\overline{MCLR}$ reset during SLEEP
d) WDT time-out reset during normal operation
e) WDT time-out reset during SLEEP

Some registers are not affected in any reset condition; their status is unknown on $\overline{POR}$ and unchanged in any other reset. Most other registers are reset to a "reset state" on power-on reset ($\overline{POR}$), on $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 14-6. These bits are used in software to determine the nature of reset. See Table 14-8 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 14-8.

© 1995 Microchip Technology Inc.

14.4 Power-On Reset (POR), Power-Up-Timer (PWRT) and Oscillator Start-up Timer (OST)

14.4.1 POWER-ON RESET (POR)

A Power-On Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.6V - 1.8V). To take advantage of the POR, just tie MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-On Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

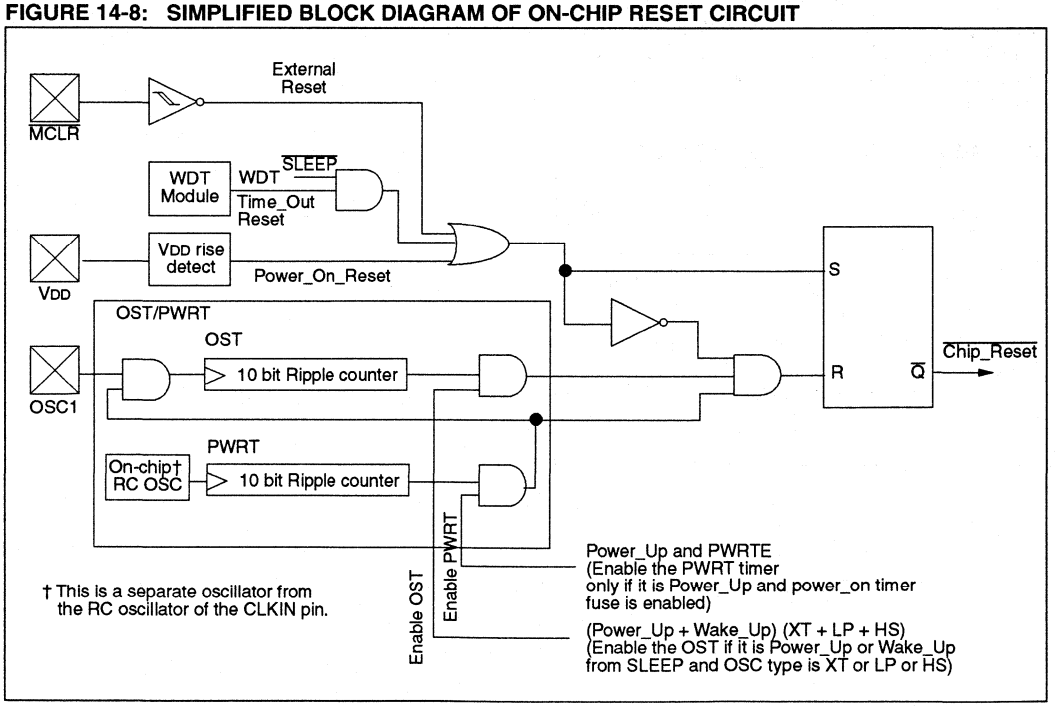
The POR circuit does not produce internal reset when VDD declines.

14.4.2 POWER-UP TIMER (PWRT)

The Power-Up Timer provides a fixed 72ms nominal time-out on power-up only, from POR. The power-up timer operates on an internal RC oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, PWRTE can enable (if set) or disable (if cleared or programmed) the power-up timer.

The Power-Up Time delay will vary from chip to chip and due to VDD, temperature, and process variation. See DC parameters for details.

FIGURE 14-8: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



© 1995 Microchip Technology Inc.

14.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This guarantees that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on power-on reset or wake-up from SLEEP.

14.4.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after $\overline{POR}$ has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and PWRTE bit status. For example, in RC mode with PWRTE bit cleared (PWRT disabled), there will be no time-out at all. Figure 14-9, Figure 14-10, and Figure 14-11 depict time-out sequences on power up.

TABLE 14-5: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake up from SLEEP
	PWRTE=1	PWRTE=0	
XT, HS, LP	72 ms + 1024Tosc	1024Tosc	1024 Tosc
RC	72 ms	—	—

Since the time-outs occur from the $\overline{POR}$ pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately (see Figure 14-10). This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 14-7 shows the reset conditions for some special registers, while Table 14-8 shows the reset conditions for all the registers.

14.4.5 POWER CONTROL/STATUS REGISTER (PCON)

The power control/status register, PCON (address 8Eh) has only one bit.

Bit1 is $\overline{POR}$ (Power-on-reset). It is cleared on power-on-reset and unaffected otherwise. The user must set this bit following power-on-reset. On a subsequent reset if $\overline{POR}$ is cleared, it will indicate that a Power-On Reset must have occurred VDD may have gone too low).

TABLE 14-6: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{TO}$	$\overline{PD}$	
0	1	1	Power-on reset
0	0	X	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	X	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
1	0	1	WDT reset during normal operation
1	0	0	WDT timeout wakeup from SLEEP
1	1	1	$\overline{MCLR}$ reset during normal operation
1	1	0	$\overline{MCLR}$ reset during SLEEP or interrupt wake-up from SLEEP

TABLE 14-7: RESET CONDITION FOR SPECIAL REGISTERS

	PCL Addr: 02h	STATUS Addr: 03h	PCON on PIC16C74 and PIC16C73 only Addr: 8Eh
Power-On Reset	000h	0001 1xxx	---- --0-
$\overline{MCLR}$ reset during normal operation	000h	0001 1uuu	---- --u-
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu	---- --u-
WDT reset during normal operation	000h	0000 1uuu	---- --u-
WDT during SLEEP	PC + 1	uuu0 0uuu	---- --u-
Interrupt wake-up from SLEEP	PC + 1 (Note1)	uuu1 0uuu	---- --u-

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0'
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Device	Power-on Reset	MCLR Reset during: – normal operation – SLEEP WDT timeout during normal operation	Wake up from SLEEP through interrupt Wake up from SLEEP through WDT timeout
W	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	7X	-	-	-
TMR0	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	7X	0000h	0000h	PC + 1 (2)
STATUS	7X	0001 1xxx	000? ?uuu (3)	uuu? ?uuu (3)
FSR	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	71	---u uuuu	---u uuuu	---u uuuu
	Other 7X	--xx xxxx	--uu uuuu	--uu uuuu
PORTB	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	74	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	74	---- -xxx	---- -uuu	---- -uuu
PCLATH	7X	---0 0000	---0 0000	---u uuuu
INTCON	7X	0000 000x	0000 000u	uuuu uuuu(1)
PIR1	7X	0000 0000	0000 0000	uuuu uuuu(1)
PIR2	7X	---- ---0	---- ---0	---- ---u(2)
TMR1L	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	7X	--00 0000	--uu uuuu	--uu uuuu
TMR2	7X	0000 0000	0000 0000	uuuu uuuu
T2CON	7X	-000 0000	-000 0000	-uuu uuuu
SSPBUF	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	7X	0000 0000	0000 0000	uuuu uuuu
CCPR1L	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	7X	--00 0000	--00 0000	--uu uuuu
RCSTA	7X	0000 -00x	0000 -00x	uuuu uuuu
TXREG	7X	0000 0000	0000 0000	uuuu uuuu
RCREG	7X	0000 0000	0000 0000	uuuu uuuu
CCPR2L	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2H	7X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	7X	0000 0000	0000 0000	uuuu uuuu
ADRES	7X	xxxx xxxx	uuuu uuuu	uuu uuuu
ADCON0	71	00-0 0000	00-0 0000	uu-u uuuu
	Other 7X	0000 00-0	0000 00-0	uuuu uu-u
OPTION	7X	1111 1111	1111 1111	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', ? = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 14-7 for reset value for specific condition.

© 1995 Microchip Technology Inc.

TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONT.)

Register	Device	Power-on Reset	MCLR Reset during: – normal operation – SLEEP WDT timeout during normal operation	Wake up from SLEEP through interrupt Wake up from SLEEP through WDT timeout
TRISA	71	---1 1111	---1 1111	---u uuuu
	Other 7X	--11 1111	--11 1111	--uu uuuu
TRISB	7X	1111 1111	1111 1111	uuuu uuuu
TRISC	7X	1111 1111	1111 1111	uuuu uuuu
TRISD	74	1111 1111	1111 1111	uuuu uuuu
TRISE	74	0000 -111	0000 -111	uuuu -uuu
PIE1	7X	0000 0000	0000 0000	uuuu uuuu
PIE2	7X	---- ---0	---- ---0	---- ---u
PCON	7X	---- --0-	---- --u-	---- --u-
PR2	7X	1111 1111	1111 1111	1111 1111
SSPADD	7X	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	7X	--00 0000	--00 0000	--uu uuuu
TXSTA	7X	0000 -010	0000 -010	uuuu -uuu
SPBRG	7X	0000 0000	0000 0000	uuuu uuuu
ADCON1	71	---- --00	---- --00	---- --uu
	Other 7X	---- -000	---- -000	---- -uuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', ? = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 14-7 for reset value for specific condition.

© 1995 Microchip Technology Inc.

FIGURE 14-9: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

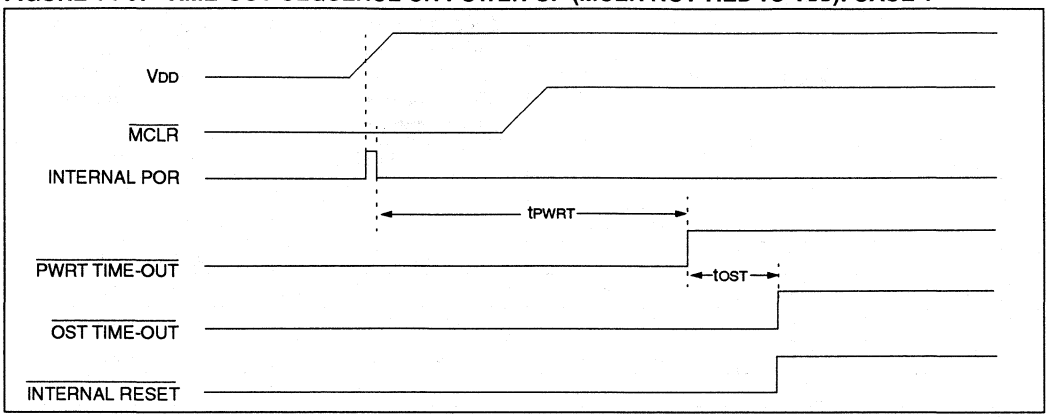


FIGURE 14-10: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

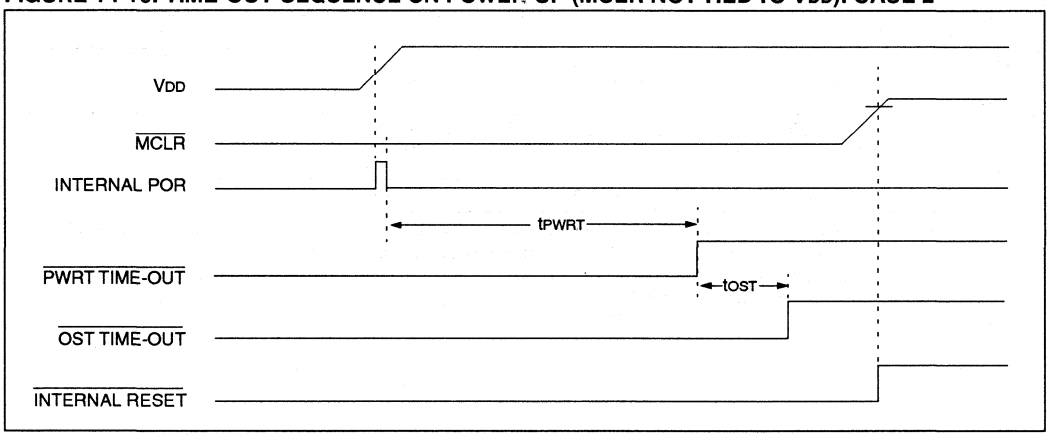
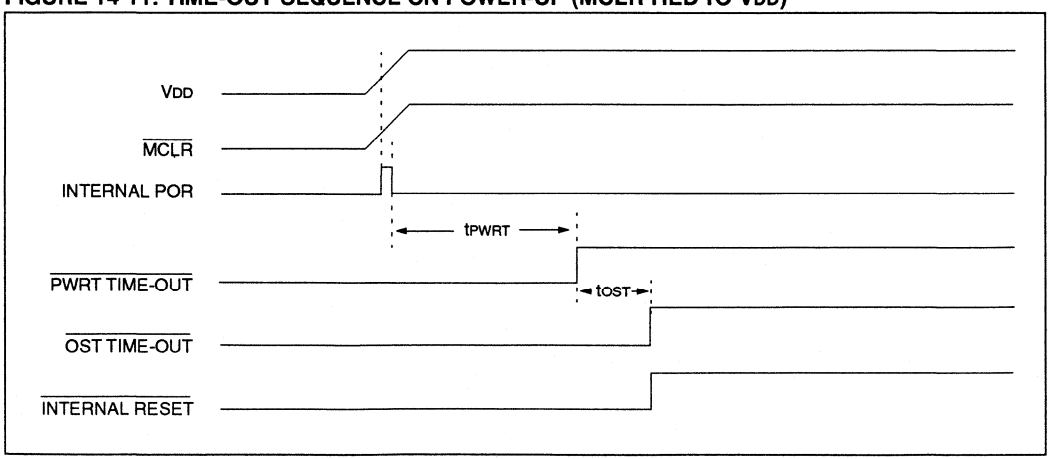
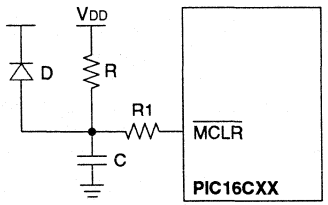


FIGURE 14-11: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)



© 1995 Microchip Technology Inc.

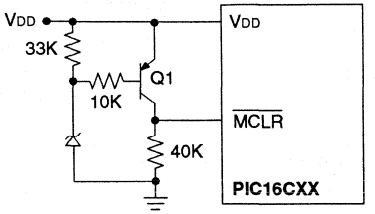
FIGURE 14-12: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note:

1. External power-on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40KΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5μA). A larger voltage drop will degrade VIH level on $\overline{MCLR}$ pin.
3. R1 = 100Ω to 1KΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of $\overline{MCLR}$ pin breakdown due to ESD or EOS.

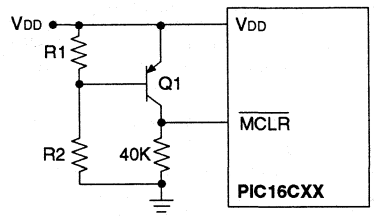
FIGURE 14-13: BROWN-OUT PROTECTION CIRCUIT 1



Note:

1. This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 14-14: BROWN-OUT PROTECTION CIRCUIT 2



Note:

1. This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V.$$

14.5 Interrupts

The PIC16C7X family has up to 12 sources of interrupt:

Interrupt sources:	Device:		
	PIC16C74	PIC16C73	PIC16C71
External interrupt RB0/INT	X	X	X
TMR0 overflow interrupt	X	X	X
PORTB change interrupts (pins RB<7:4>)	X	X	X
A/D interrupt	X	X	X
TMR1 overflow interrupt	X	X	
TMR2 matches period interrupt	X	X	
CCP1 interrupt	X	X	
CCP2 interrupt	X	X	
SCI asynchronous transmit and receive	X	X	
Synchronous serial port interrupt	X	X	
Parallel slave port read/write interrupt	X		

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The "return from interrupt" instruction, RETFIE, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special register PIR1. The corresponding interrupt enable bits are contained in special register PIE1 and the peripheral interrupt enable bit is contained in special register INTCON.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (see Figure 14-17). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged
2. The program branches to the interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GIE bit was cleared by the instruction, as shown in the following code:

```
LOOP    BCF INTCON,GIE      ; Disable Global
                            ; Interrupts
        BTFSC INTCON,GIE    ; Global Interrupts
                            ; Disabled?
        GOTO    LOOP;       ; NO, try again
        :                   ; Yes, continue
                            ; with program
                            ; flow
```


© 1995 Microchip Technology Inc.

FIGURE 14-15: INTERRUPT LOGIC FOR PIC16C74 AND PIC16C73

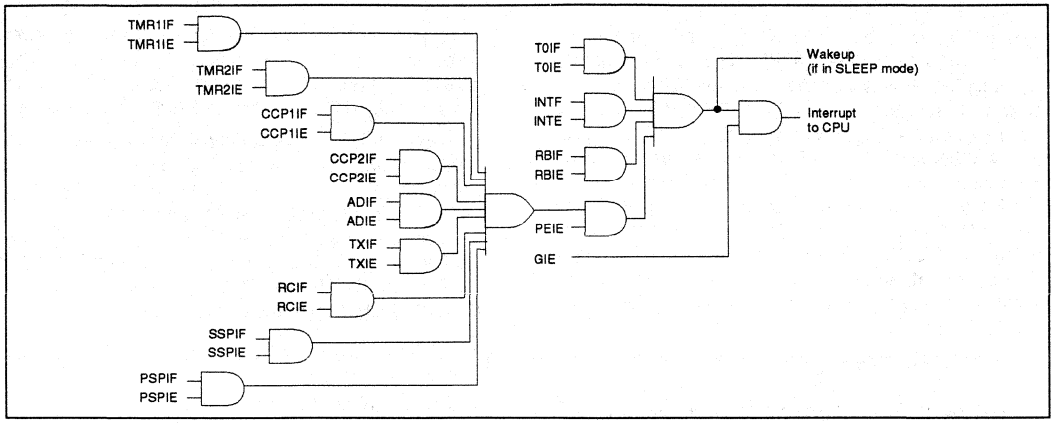


FIGURE 14-16: INTERRUPT LOGIC FOR PIC16C71

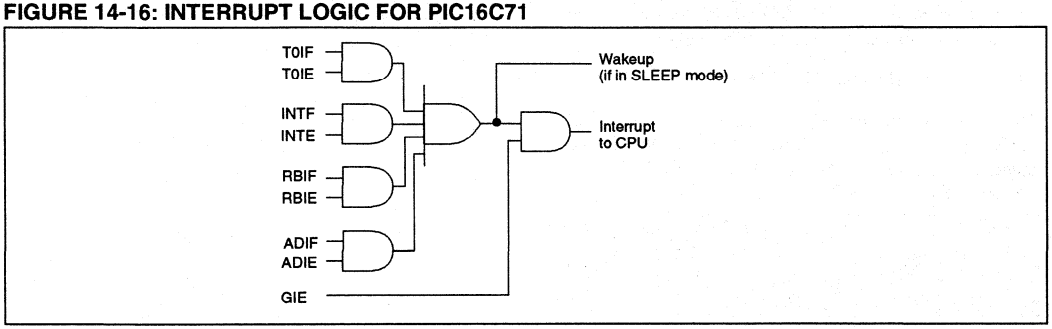
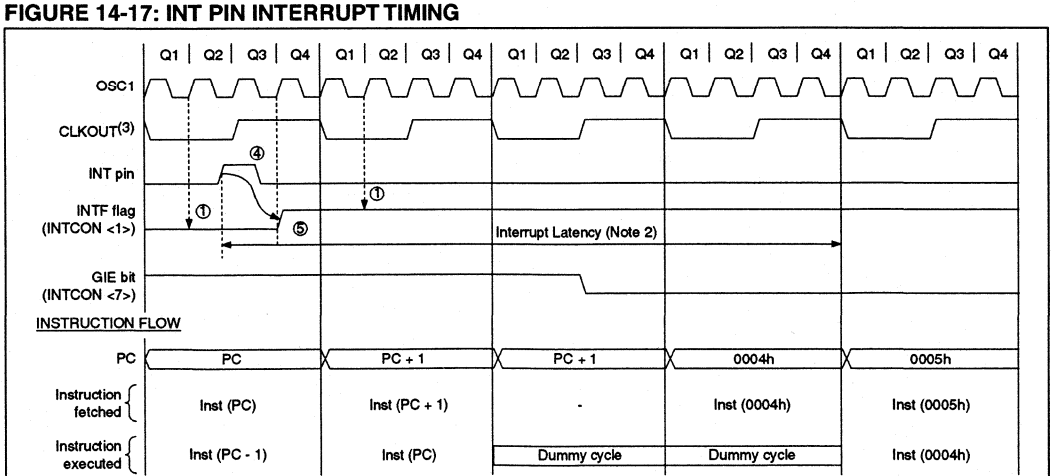


FIGURE 14-17: INT PIN INTERRUPT TIMING



Notes:
1. INTF flag is sampled here (every Q1)
2. Interrupt latency = 3-4 Tcy where Tcy = instruction cycle time.
   Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3. CLKOUT is available only in RC oscillator mode.
4. For minimum width spec of INT pulse, refer to AC specs.
5. INTF is enabled to be set anytime during the Q4-Q1 cycles.

14.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the INT pin, the INTF bit (INTCON<1>) is set . This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 14.8 for details on SLEEP mode.

14.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the TMR0 module, see Section 7.0.

14.5.3 PORTB INTCON CHANGE

An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB, see Section 5.2.

Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

14.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt e.g. W register and STATUS register. This will have to be implemented in software.

Example 14-1 and Example 14-2 store and restore the STATUS and W registers. For PIC16C74 and PIC16C73, the register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1). For PIC16C74 and PIC16C73, the user register, STATUS_TEMP, must be defined in bank 0.

The example:

- Stores W register
- Stores STATUS register in bank 0
- Executes ISR code
- Restores STATUS (and bank select bit) register
- Restores W register

© 1995 Microchip Technology Inc.

EXAMPLE 14-1: SAVING STATUS AND W REGISTERS IN RAM (PIC16C74 AND PIC16C73)

```
MOVWF     W_TEMP          ;Copy W to TEMP register, could be bank one or zero
SWAPF     STATUS,W        ;Swap status to be saved into W
BCF       STATUS,RP0      ;Change to bank zero, regardless of current bank
MOVWF     STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
:         (ISR)
:
SWAPF     STATUS_TEMP,W   ;Swap STATUS_TEMP register into W (sets bank to
                          ;original state)
MOVWF     STATUS          ;Move W into STATUS register
SWAPF     W_TEMP,F        ;Swap W_TEMP
SWAPF     W_TEMP,W        ;Swap W_TEMP into W
```

EXAMPLE 14-2: SAVING STATUS AND W REGISTERS IN RAM (PIC16C71)

```
MOVWF     W_TEMP          ;Copy W to TEMP register
SWAPF     STATUS,W        ;Swap status to be saved into W
MOVWF     STATUS_TEMP     ;Save status to STATUS_TEMP register
:
:         (ISR)
:
SWAPF     STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
MOVWF     STATUS          ;Move W into STATUS register
SWAPF     W_TEMP,F        ;Swap W_TEMP
SWAPF     W_TEMP,W        ;Swap W_TEMP into W
```


© 1995 Microchip Technology Inc.

14.7 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT timeout causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration fuse WDTE as a '0' (Section 14.1).

14.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

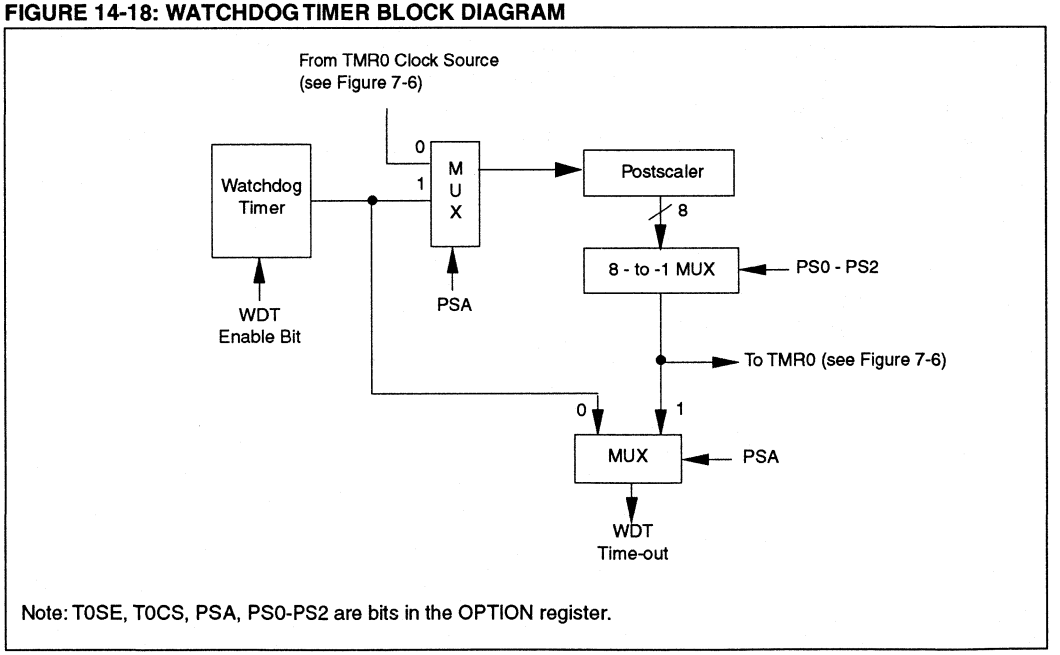
The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a watchdog timer time-out.

14.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

FIGURE 14-18: WATCHDOG TIMER BLOCK DIAGRAM



Note: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

FIGURE 14-19: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	–	–	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

© 1995 Microchip Technology Inc.

14.8 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP command was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin, power down the A/D, disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a RESET generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

14.8.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin
2. Watchdog timer time-out reset (if WDT was enabled)
3. Interrupt from INT pin, RB port change, or some Peripheral Interrupts.

The following peripheral interrupts can wake-up from SLEEP:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. SSP (Start/Stop) bit detect interrupt.
3. CCP capture mode interrupt.
4. Parallel Slave port read or write.

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

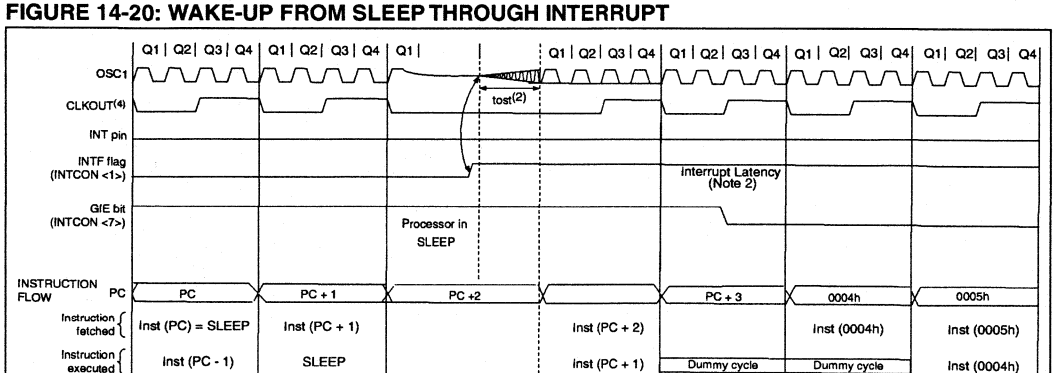
The first event will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. $\overline{PD}$ bit, which is set on power-up is cleared when SLEEP is invoked. $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

Note: If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wakeup from sleep. The sleep instruction is completely executed.

The WDT is cleared when the device wakes-up from sleep, regardless of the source of wake-up.

FIGURE 14-20: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Notes:
1. XT, HS or LP oscillator mode assumed
2. tost = 1024 tosc (drawing not to scale). This delay will not be there for RC osc mode.
3. GIE = 1 assumed. In this case after wake up processor jumps to interrupt routine. If GIE = 0, execution will continue in line.
4. CLKOUT is not available in these osc modes, but shown here for timing reference.

14.9 Code Protection

The code in the program memory can be protected by programming the code protect bits.

The PIC16C74 and PIC16C73 each have two code protect bits (CP<1:0>) and the PIC16C71 has one code protect bit (CP<0>).

The PIC16C74 and PIC16C73 code protection scheme allows the user to selectively protect portions of the program memory. Refer to Figure 14-1 for code protection bit assignments for the PIC16C74 and PIC16C73. Once a segment has been code protected, those memory locations cannot be further programmed. Unprotected segments can be read and reprogrammed.

The PIC16C71 has one code protection bit, CP0 (refer to Figure 14-2). When code protection is enabled, all locations 40h and above cannot be reprogrammed. The first 64 locations, 00h–3Fh, can be reprogrammed.

The configuration word and ID locations are not code protected for all devices.

14.10 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. Only the 4 least significant bits of ID location are usable.

14.11 In-Circuit Serial Programming

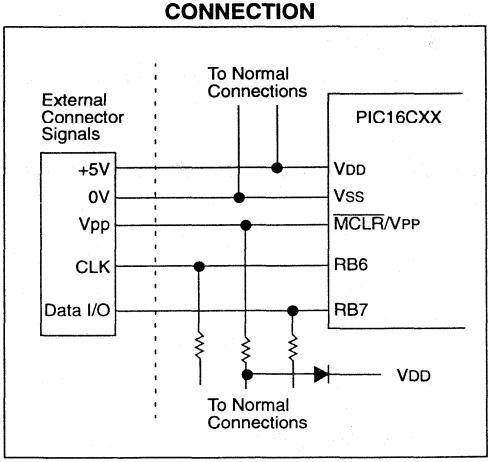
The PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

A typical in-system serial programming connection is shown in Figure 14-21.

FIGURE 14-21: TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION



© 1995 Microchip Technology Inc.

15.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 15-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 15-1 shows the opcode field descriptions.

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be utilized by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 15-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
w	Working register (accumulator)
b	Bit address within an 8 bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable Bit
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-out Bit
$\overline{PD}$	Power-down Bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte oriented operations
- Bit oriented operations
- Literal and control operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µsec.

Table 15-2 lists the instructions recognized by the MPASM assembler.

Figure 15-1 shows the three general formats that the instructions can have.

Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS

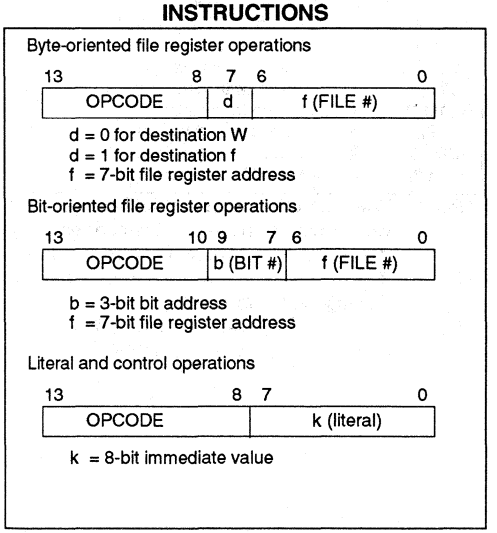


TABLE 15-2: PIC16CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode msb			lsb	Status Affected	Notes
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W and f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W and f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate left through carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W and f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal to W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal to W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear watchdog timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal to W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Excl. OR literal to W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g. MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

Note 2: If this instruction is executed on the TMR0 register (and, where applicable, d=1), the prescaler will be cleared if assigned to the TMR0.

Note 3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

© 1995 Microchip Technology Inc.

15.1 Instruction Descriptions

ADDLW **Add Literal to W**

Syntax: [*label*] ADDLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) + k → W

Status Affected: C, DC, Z

Encoding:

11	111x	kkkk	kkkk

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

ANDLW **And Literal and W**

Syntax: [*label*] ANDLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .AND. (k) → W

Status Affected: Z

Encoding:

11	1001	kkkk	kkkk

Description: The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ADDWF **ADD W to f**

Syntax: [*label*] ADDWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0111	dfff	ffff

Description: Add the contents of the W register to register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

00	0101	dfff	ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

© 1995 Microchip Technology Inc.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow f<b>$

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example
```
BCF     FLAG_REG,  7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow f<b>$

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example
```
BSF     FLAG_REG,  7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **BIT Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is skipped.
If bit 'b' is '0' then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address TRUE
if FLAG<1>=1,
PC=address FALSE

© 1995 Microchip Technology Inc.

BTFSS **Bit Test, skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '1' then the next instruction is skipped.
If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address FALSE
if FLAG<1>=1,
PC=address TRUE

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL   THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE + 1

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → f
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF    FLAG_REG
```

Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W Register**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0XXX	XXXX

Description: W registered is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

© 1995 Microchip Technology Inc.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description: CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

00	1001	dfff	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f)-1 → (dest)

Status Affected: Z

Encoding:

00	0011	dfff	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → d; skip if result = 0

Status Affected: None

Encoding:

00	1011	dfff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

© 1995 Microchip Technology Inc.

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: 0 ≤ k ≤ 2047

Operation: k → PC<10:0>
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example
```
GOTO THERE
```
After Instruction
PC = Address THERE

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is decremented. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    INCFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT + 1
if CNT= 0,
PC = address CONTINUE
if CNT≠ 0,
PC = address HERE+1

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
INCF    CNT, 1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

IORLW **Inclusive OR Literal with W**

Syntax: [*label*] IORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk

Description: The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
IORLW   0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF

© 1995 Microchip Technology Inc.

IORWF **Inclusive OR W with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (W)

Status Affected: Z

Encoding:

00	0100	dfff	ffff

Description: Inclusive OR the W register with register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
IORWF          RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

Description: The contents of register f is moved to destination d. If d=0, destination is W register. If d=1, the destination is file register f itself. d=1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example
```
MOVF     FSR, 0
```
After Instruction
W = value in FSR register

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

11	00XX	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example
```
MOVLW    0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

00	0000	1fff	ffff

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example
```
MOVWF       OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

© 1995 Microchip Technology Inc.

NOP — **No Operation**

Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	00 | 0000 | 0XX0 | 0000
Description:	No operation.
Words:	1
Cycles:	1
Example	NOP

OPTION — **Load Option Register**

Syntax:	[*label*] OPTION
Operands:	None
Operation:	W → OPTION;
Status Affected:	None
Encoding:	00 | 0000 | 0110 | 0010
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.
Words:	1
Cycles:	1
Example	**To maintain upward compatibility with future PIC16CXX products, do not use this instruction.**

RETFIE — **Return from Interrupt**

Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE;
Status Affected:	None
Encoding:	00 | 0000 | 0000 | 1001
Description:	Return from Interrupt. Stack is popped and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting the Global Interrupt Enable (GIE) bit. GIE is the global interrupt enable bit (INTCON<7>). This is a two cycle instruction.
Words:	1
Cycles:	2
Example	RETFIE After Interrupt PC = TOS GIE = 1

RETLW — **Return Literal to W**

Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W; TOS → PC;
Status Affected:	None
Encoding:	11 | 01XX | kkkk | kkkk
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.
Words:	1
Cycles:	2

Example

```
        CALL TABLE  ;W contains table
                    ;offset value
        •           ;W now has table value
        •
        •
TABLE   ADDWF PC    ;W = offset
        RETLW k1    ;Begin table
        RETLW k2    ;
        •
        •
        •
        RETLW kn    ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k7

© 1995 Microchip Technology Inc.

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;

Status Affected: None

Encoding:

00	0000	0000	1000

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETURN
```

After Interrupt
PC = TOS

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

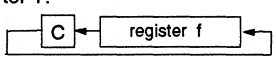
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example

```
RLF     REG1,0
```

Before Instruction
```
REG1    =   11100110
C       =   0
```
After Instruction
```
REG1    =   11100110
W       =   11001100
C       =   1
```

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

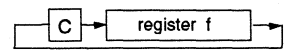
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example

```
RRF     REG1,0
```

Before Instruction
```
REG1    =   11100110
C       =   0
```
After Instruction
```
REG1    =   11100110
W       =   01110011
C       =   1
```

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power down status bit ($\overline{PD}$) is cleared. Time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Section 14.8 for more details.

Words: 1

Cycles: 1

Example:

```
SLEEP
```


© 1995 Microchip Technology Inc.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1:
```
SUBLW   0x02
```
Before Instruction
W = 1
C = ?
After Instruction
W = 1
C = 1; result is positive

Example 2: Before Instruction
W = 2
C = ?
After Instruction
W = 0
C = 1; result is zero

Example 3: Before Instruction
W = 3
C = ?
After Instruction
W = FF
C = 0; result is negative

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: f - (W) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement methodize W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1,1
```
Before Instruction
REG1 = 3
W = 2
C = ?
After Instruction
REG1 = 1
W = 2
C = 1; result is positive

Example 2: Before Instruction
REG1 = 2
W = 2
C = ?
After Instruction
REG1 = 0
W = 2
C = 1; result is zero

Example 3: Before Instruction
REG1 = 1
W = 2
C = ?
After Instruction
REG1 = FF
W = 2
C = 0; result is negative

© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label* SWAPF f,d]

Operands: $0 \le f \le 127$, $d \in [0,1]$

Operation: f<0:3> → d<4:7>, f<4:7> → d<0:3>

Status Affected: None

Encoding:

00	1110	dfff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example

```
SWAP  REG,  0
F
```

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: $5 \le f \le 7$

Operation: W → TRIS register f;

Status Affected: None

Encoding:

00	0000	0110	0fff

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

11	1010	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example

```
XORLW  0xAF
```

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

Operands: $0 \le f \le 127$, $d \in [0,1]$

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

00	0110	dfff	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example

```
XORWF  REG  1
```

Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

© 1995 Microchip Technology Inc.

16.0 DEVELOPMENT SUPPORT

16.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®-MP)

16.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator provides the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 16-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 and better machines. The development software runs in the Microsoft Windows™ 3.x environment, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Emulator Universal System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host emulation control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x1, two or more PICMASTER emulators can be run simultaneously on the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes currently meet the specifications shown in Table 16-1.

FIGURE 16-1: PICMASTER SYSTEM CONFIGURATION

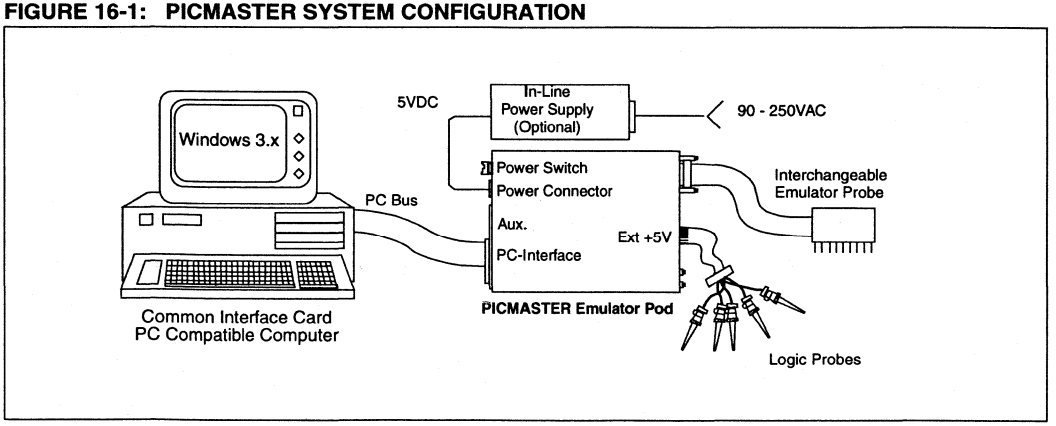


TABLE 16-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

16.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module." Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

16.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

16.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

16.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to and LCD module and a keypad.

© 1995 Microchip Technology Inc.

16.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro Assembly Capability
- Provides Object files, Listing files, Symbol files and special files required for debugging with one of the Microchip Emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** control the allocation of memory and provide a way to refer to data items symbolically, by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and sub-titles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

16.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in single step, execute until break or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

16.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

16.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool comes in two versions: a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design, and a full-featured *fuzzy*TECH-MP Edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

16.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 16-2.

TABLE 16-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator with PRO MATE Programmer, Assembler, Software Simulator, Samples, and your choice of Target Probe,
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

17.0 ELECTRICAL CHARACTERISTICS FOR PIC16C74 AND PIC16C73

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to+ 125°C

Storage Temperature ..... -65°C to +150°C

Voltage on any pin with respect to Vss (except VDD and $\overline{MCLR}$) ..... -0.6V to VDD +0.6V

Voltage on VDD with respect to Vss ..... 0 to +7.5 V

Voltage on $\overline{MCLR}$ with respect to Vss (Note 2) ..... 0 to +14 V

Total power Dissipation (Note 1) ..... 1.0 W

Maximum Current out of Vss pin 300mA

Maximum Current into VDD pin 250mA

Input clamp current, IIK (VI<0 or VI> VDD) ..... ±20mA

Output clamp current, IOK (V0 <0 or V0>VDD) ..... ±20mA

Maximum Output Current sunk by any I/O pin 25mA

Maximum Output Current sourced by any I/O pin 25mA

Maximum Current sunk by PORTA, PORTB, and PORTE (combined)†† 200mA

Maximum Current sourced by PORTA, PORTB, and PORTE (combined)†† 200mA

Maximum Current sunk by PORTC and PORTD (combined)†† 200mA

Maximum Current sourced by PORTC and PORTD (combined)†† 200mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - Σ IOH} + Σ {(VDD-VOH) x IOH} + Σ(VOl x IOL)

Note 2: Voltage spikes below Vss at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to Vss.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

††PORTD and PORTE are not implemented on the PIC16C73.

TABLE 17-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C74-04 PIC16C73-04	PIC16C74-10 PIC16C73-10	PIC16C74-20 PIC16C73-20	PIC16LC74-04 PIC16LC73-04
RC	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 2.0mA typ. at 3.0V IPD: 0.9µA typ. at 3V WDT dis Freq: 4MHz typ.
XT	VDD: 4.0V to 6.0V IDD: 5mA Max. at 5.5V IPD: 21µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 2.7mA typ. at 5.5V IPD: 1.5µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 2.0mA typ. at 3.0V IPD: 0.9µA typ. at 3V WDT dis Freq: 4MHz typ.
HS	VDD: 4.5V to 5.5V IDD: 13.5mA typ. at 5.5V IPD: 1.5µA typ. at 4.5V WDT dis Freq: 4MHz	VDD: 4.5V to 5.5V IDD: 30mA Max. at 5.5V IPD: 1.5µA typ. at 4.5V WDT dis Freq: 20MHz Max.	VDD: 4.5V to 5.5V IDD: 30mA Max. at 5.5V IPD: 1.5µA typ. at 4.5V WDT dis Freq: 20MHz Max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 52.5µA typ. at 32 KHz, 4.0V IPD: 0.9µA typ. at 4.0V WDT dis Freq: 200KHz typ.	Do not use in LP mode	Do not use in LP mode	VDD: 3.0V to 6.0V IDD: 48µA Max. at 32 KHz, 3.0V IPD: 13.5µA Max. at 3.0V WDT dis Freq: 200KHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

17.1 DC Characteristics:

PIC16C74-04 (Commercial, Industrial, Automotive)
PIC16C74-10 (Commercial, Industrial, Automotive)
PIC16C74-20 (Commercial, Industrial, Automotive)
PIC16C73-04 (Commercial, Industrial, Automotive)
PIC16C73-10 (Commercial, Industrial, Automotive)
PIC16C73-20 (Commercial, Industrial, Automotive

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- - - -	2.7 52.5 13.5 10.5	5 105 30 42	mA µA mA µA	XT, RC, osc configuration (PIC16C74-04) FOSC = 4 MHz, VDD = 5.5V (Note 4) LP osc configuration (PIC16C74-04) FOSC = 32 KHz, VDD = 4.0V, WDT disabled HS osc configuration (PIC16C74-20) FOSC = 20 MHz, VDD = 5.5V VDD=4.0V, WDT enabled,-40°C to +85°C
Power Down Current (Note 3, 5)	IPD	- - - - - -	1.5 1.5 1.5 15* 19*	21 24 TBD 32* 40*	µA µA µA µA µA	VDD=4.0V, WDT disabled,-0°C to +70°C VDD=4.0V, WDT disabled,-40°C to +85°C VDD=4.0V, WDT disabled,-40°C to +125°C LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Commercial LP osc, VDD=2.5V, Sleep Mode, TMR1 External Clock=32KHz, Industrial

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD.
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and V_{SS}.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

 © 1995 Microchip Technology Inc.

17.2 DC Characteristics: PIC16LC74-04 (Commercial, Industrial, Automotive) PIC16LC73-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0 3.0	- -	6.0 6.0	V V	LP osc configuration XT, RC osc configuration (DC - 4MHz)
RAM Data Retention Voltage (Note 1)	VDR	1.5*	-	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2, 5)	IDD	- -	2.0 22.5	3.8 48	mA µA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4) LP OSC configuration FOSC = 32 KHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3, 5)	IPD	- - - -	7.5 0.9 0.9 0.9	30 13.5 18 24	µA µA µA µA	VDD=3.0V, WDT enabled,-40°C to +85°C VDD=3.0V, WDT disabled,0°C to +70°C VDD=3.0V, WDT disabled,-40°C to +85°C VDD=3.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD.
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

17.3 DC Characteristics:

PIC16C74-04 (Commercial, Industrial, Automotive)
PIC16C74-10 (Commercial, Industrial, Automotive)
PIC16C74-20 (Commercial, Industrial, Automotive)
PIC16LC74-04 (Commercial, Industrial, Automotive)
PIC16C73-04 (Commercial, Industrial, Automotive)
PIC16C73-10 (Commercial, Industrial, Automotive)
PIC16C73-20 (Commercial, Industrial, Automotive)
PIC16LC73-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ + 125°C for automotive,
-40°C ≤ TA ≤ + 85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Table 17-1 and Table 17-2

Characteristic	Sym	Min	Typ†	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	-	0.8V	V	
with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
$\overline{MCLR}$, RA4/T0CKI,OSC1 (in RC mode)		Vss	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		Vss	-	0.3VDD	V	Note1
Input High Voltage						
I/O ports	VIH		-			
with TTL buffer		2.0	-	VDD	V	4.5 ≤ VDD ≤5.5V
		0.85 VDD	-	VDD	V	For entire VDD range
with Schmitt Trigger buffer		0.85 VDD	-	VDD	V	For entire VDD range
$\overline{MCLR}$ RA4/T0CKI, RC<7:4>, RD<7.4>INT		0.85 VDD	-	VDD	V	
RE<2:0>, OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	200	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2,3)						
I/O ports	IIL	-	-	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
$\overline{MCLR}$, RA4/T0CKI		-	-	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	-	-	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD-4.5V, -40°C to +125°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD-4.5V, -40°C to +125°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +125°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	
SCL, SDA in I^2C mode	Cb			400	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C74 be driven with external clock in RC mode.
2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
3: Negative current is defined as coming out of the pin.
4: The user may use better of the two specs.

© 1995 Microchip Technology Inc.

17.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

17.5 Timing Diagrams and Specifications

FIGURE 17-1: EXTERNAL CLOCK TIMING

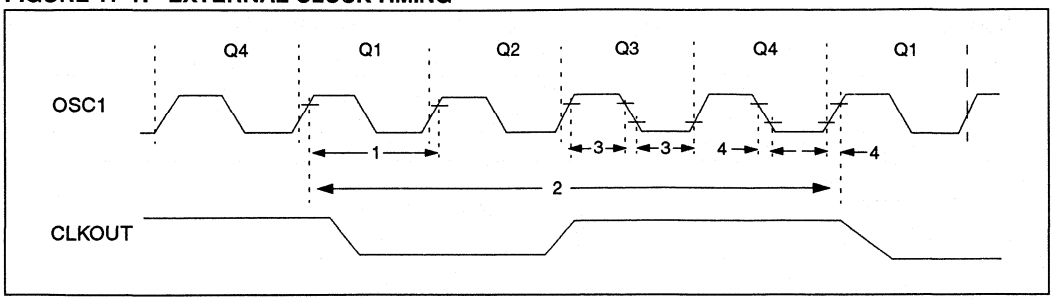


TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

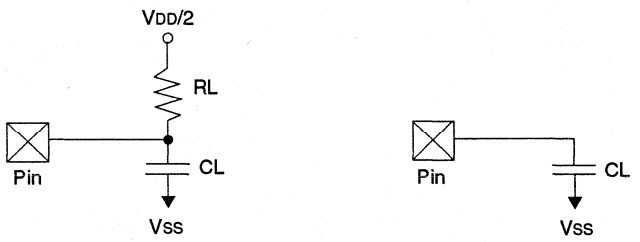
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C74-04, PIC16LC74-04)
			DC	—	20	MHz	HS osc mode (PIC16C74-20)
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	4	MHz	HS osc mode (PIC16C74-04, PIC16LC74-04)
			4	—	10	MHz	HS osc mode (PIC16C74-10)
			4	—	20	MHz	HS osc mode (PIC16C74-20)
			5	—	200	KHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C74-04, PIC16LC74-04)
			100	—	—	ns	HS osc mode (PIC16C74-10)
			50	—	—	ns	HS osc mode (PIC16C74-20)
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (PIC16C74-04, PIC16LC74-04)
			100	—	250	ns	HS osc mode (PIC16C74-10)
			50	—	250	ns	HS osc mode (PIC16C74-20)
			5	—	200	µs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	200	—	DC	ns	Tcy = 4/Fosc
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

 © 1995 Microchip Technology Inc.

FIGURE 17-2: LOAD CONDITIONS



RL = 464 Ω

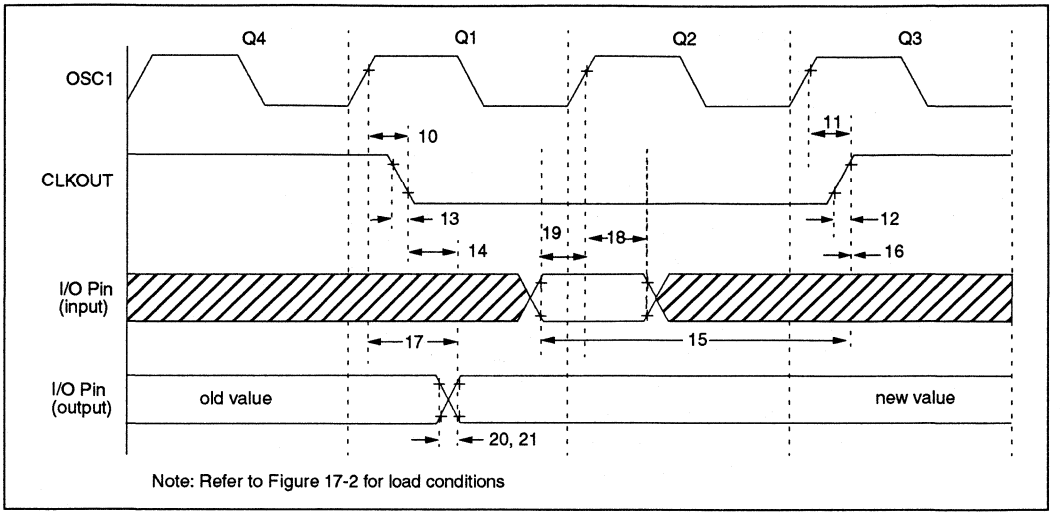
CL = 100 pF for D and E port outputs when used as system bus

50 pF for all pins except OSC2 but including D and E outputs as ports

25 pF for OSC2 output

Note: PORTD and PORTE are not implemented on the PIC16C73.

FIGURE 17-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 17-2 for load conditions

TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	0.25 TCY+25	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	INT pin high or low time	20	—	—	ns	
23††	Trbp	RB<7:4> change INT high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc

 © 1995 Microchip Technology Inc.

FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

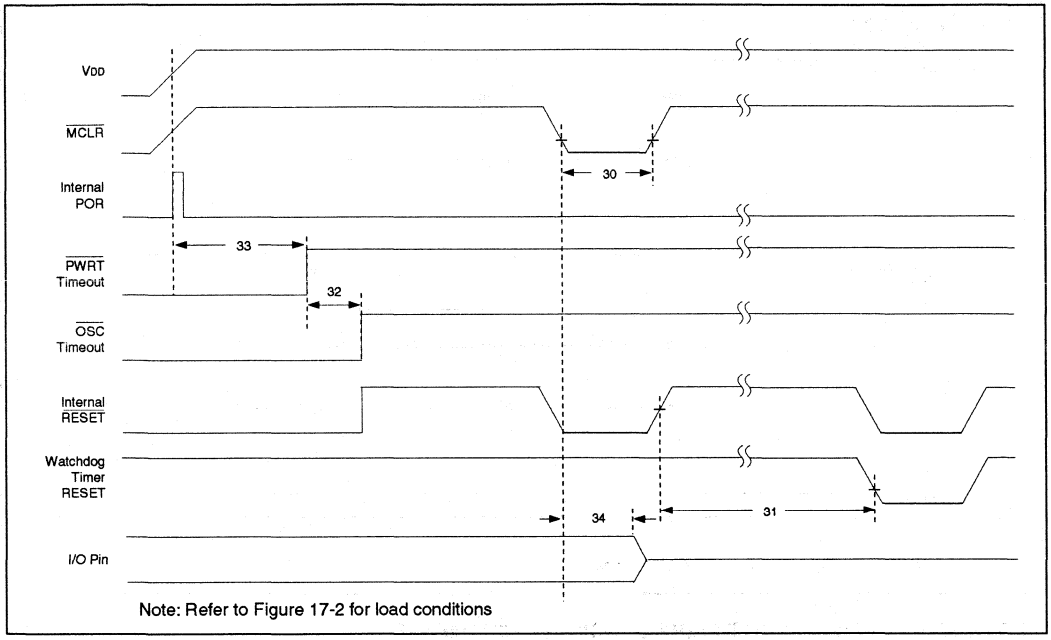


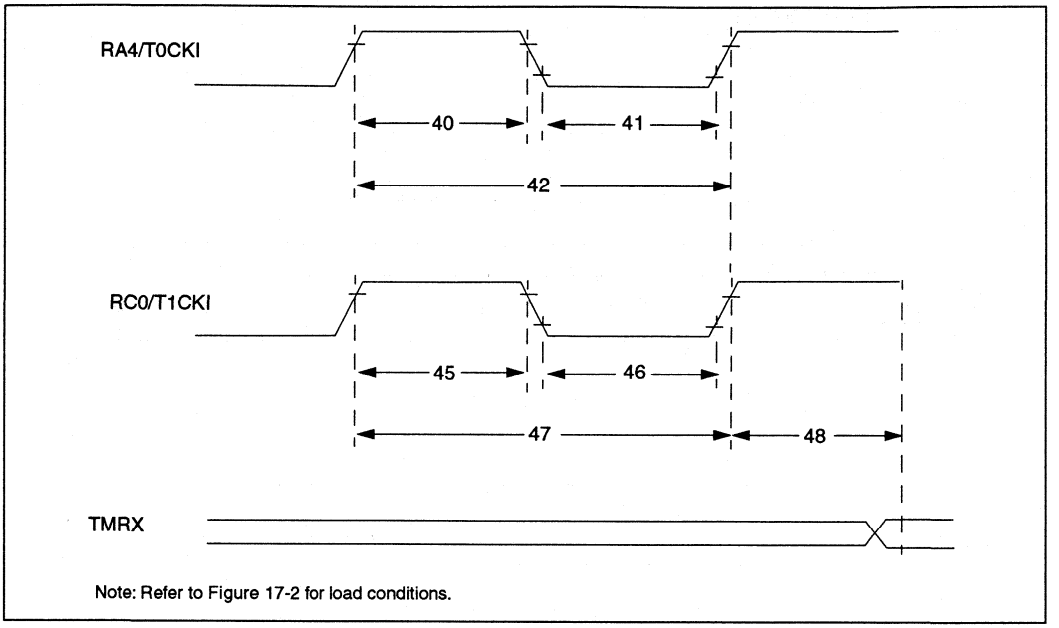
TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O High Impedance from MCLR Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-5: TIMER0 AND TIMER1 CLOCK TIMINGS



Note: Refer to Figure 17-2 for load conditions.

TABLE 17-5: TIMER0 AND TIMER1 CLOCK REQUIREMENTS

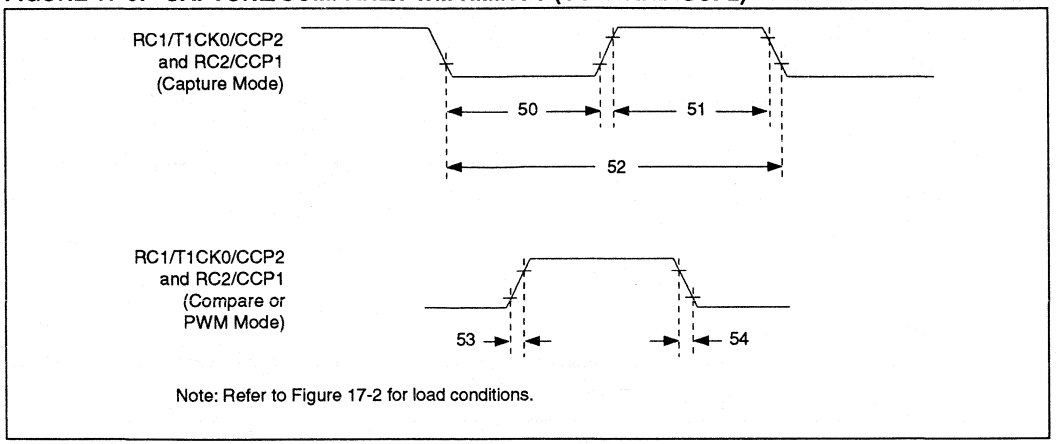
Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, ..., 256)
45	Tt1H	T1CKI High Time	Synchronous, No Prescaler	0.5TCY + 20	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
46	Tt1L	T1CKI Low Time	Synchronous, No Prescaler	0.5TCY + 20*	—	—	ns	
			Synchronous, With Prescaler	10*	—	—	ns	
			Asynchronous	2 TCY	—	—	ns	
47	Tt1P	T1CKI input period	Synchronous	TCY + 40* / N	—	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	4 TCY	—	—	ns	
	Ft1	Timer1 oscillator input frequency range (oscillator enabled by setting the T1OSCEN bit)		DC	—	200	KHz	
48	TCKE2tmrI	Delay from external clock edge to timer increment		2 TOSC	—	7TOSC	—	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 17-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)



Note: Refer to Figure 17-2 for load conditions.

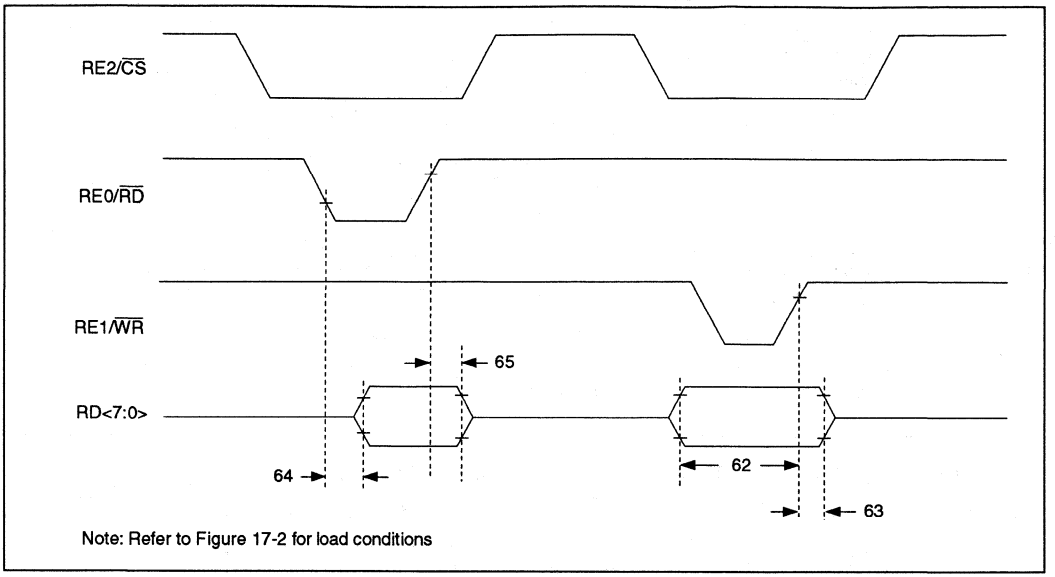
TABLE 17-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
50	TccL	CCP1 and CCP2 input low time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
51	TccH	CCP1 and CCP2 input high time	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
52	TccP	CCP1 and CCP2 input period		3TCY + 40* / N	—	—	ns	N = prescale value (1,4 or 16)
53	TccR	CCP1 and CCP2 output rise time		—	10	25	ns	
54	TccF	CCP1 and CCP2 output fall time		—	10	25	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-7: PARALLEL SLAVE PORT TIMING FOR THE PIC16C74 ONLY



Note: Refer to Figure 17-2 for load conditions

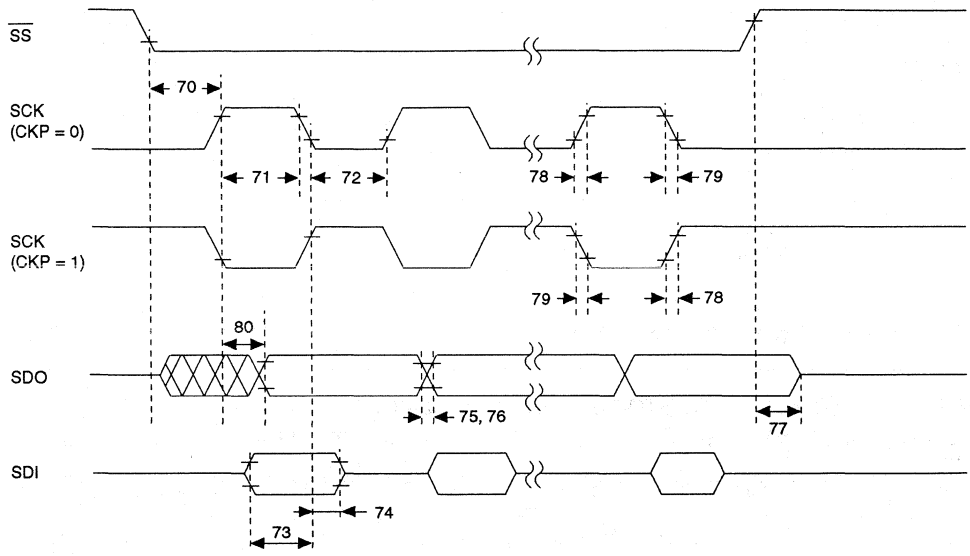
TABLE 17-7: PARALLEL SLAVE PORT REQUIREMENTS FOR THE PIC16C74 ONLY

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before WR↑ or CS↑ (setup time)	20	—	—	ns	
63	TwrH2dtI	WR↑ or CS↑ to data–in invalid (hold time)	20	—	—	ns	
64	TrdL2dtV	RD↓ and CS↓ to data–out valid	—	—	40	ns	
65	TrdH2dtI	RD↑ or CS↓ to data–out invalid	10	—	30	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

FIGURE 17-8: SPI MODE TIMING



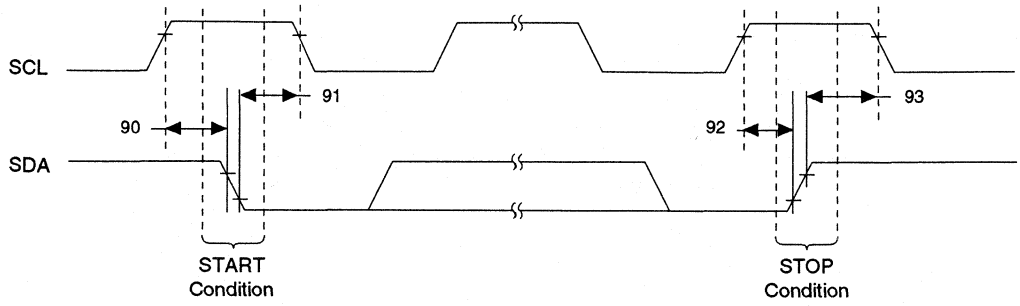
Note: Refer to Figure 17-2 for load conditions

TABLE 17-8: SPI MODE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}$↓ to SCK↓ or SCK↑ input	Tcy	—	—	ns	
71	TscH	SCK input high time (slave mode)	Tcy + 20	—	—	ns	
72	TscL	SCK input low time (slave mode)	Tcy + 20	—	—	ns	
73	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	Tcy	—	—	ns	
74	TscH2diL, TscL2diL	Hold time of SDI data input to SCK edge	0.5 Tcy	—	—	ns	
75	TdoR	SDO data output rise time	—	10	25	ns	
76	TdoF	SDO data output fall time	—	10	25	ns	
77	TssH2doZ	$\overline{SS}$↓ to SDO output hi-impedance	10	—	50	ns	
78	TscR	SCK output rise time (master mode)	—	10	25	ns	
79	TscF	SCK output fall time (master mode)	—	10	25	ns	
80	TscH2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-9: I²C BUS START/STOP BITS TIMING



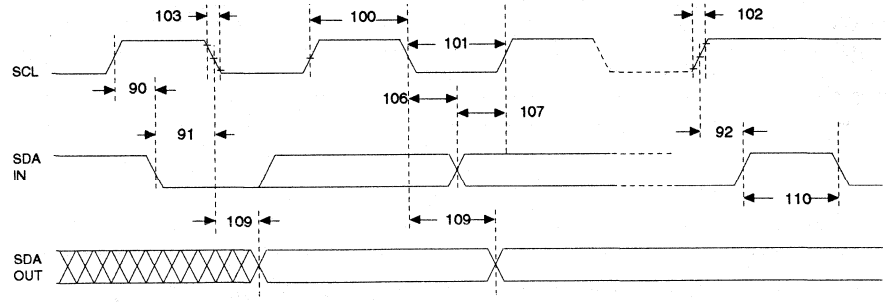
Note: Refer to Figure 17-2 for load conditions

TABLE 17-9: I²C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
90	TSU:STA	START condition Setup time	100 KHZ mode	4700	—	—	ns	Only relevant for repeated START condition
			400 KHz mode	600	—	—		
91	THD:STA	START condition Hold time	100 KHz mode	4000	—	—	ns	After this period the first clock pulse is generated
			400 KHz mode	600	—	—		
92	TSU:STO	STOP condition Setup time	100 KHZ mode	4700	—	—	ns	
			400 KHz mode	600	—	—		
93	THD:STO	STOP condition Hold time	100 KHz mode	4000	—	—	ns	
			400 KHz mode	600	—	—		

 © 1995 Microchip Technology Inc.

FIGURE 17-10: I²C BUS DATA TIMING



Note: Refer to Figure 17-2 for load conditions

TABLE 17-10: I²C BUS DATA REQUIREMENTS

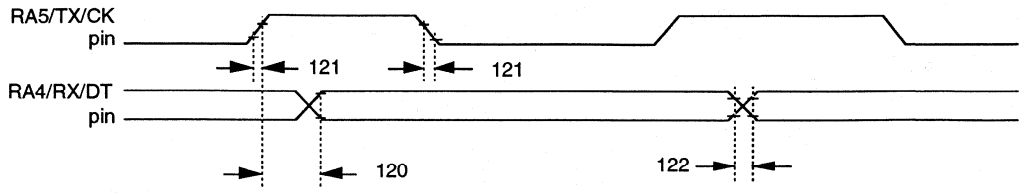
Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock high time	100 KHz mode	4.0	—	µs	PIC16C74 must operate at a minimum of 1.5 MHz
			400 KHz mode	0.6	—	µs	PIC16C74 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
101	TLOW	Clock low time	100 KHz mode	4.7	—	µs	PIC16C74 must operate at a minimum of 1.5 MHz
			400 KHz mode	1.3	—	µs	PIC16C74 must operate at a minimum of 10 MHz
			SSP Module	1.5 TCY	—		
102	TR	SDA and SCL rise time	100 KHz mode	—	1000	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
103	TF	SDA and SCL fall time	100 KHz mode	—	300	ns	
			400 KHz mode	20+0.1 Cb	300	ns	Cb is specified to be from 10-400 pF
90	TSU:STA	START condition setup time	100 KHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 KHz mode	0.6	—	µs	
91	THD:STA	START condition hold time	100 KHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 KHz mode	0.6	—	µs	
106	THD:DAT	Data input hold time	100 KHz mode	0	—	ns	
			400 KHz mode	0	0.9	µs	
107	TSU:DAT	Data input setup time	100 KHz mode	250	—	ns	Note 2
			400 KHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 KHz mode	4.7	—	µs	
			400 KHz mode	0.6	—	µs	
109	TAA	Output valid from clock	100 KHz mode	—	3500	ns	Note 1
			400 KHz mode	—	—	ns	
110	TBUF	Bus free time	100 KHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 KHz mode	1.3	—	µs	
	Cb	Bus capacitive loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode I²C-bus device can be used in a standard-mode I²C-bus system, but the requirement tsu;DAT≥250ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT=1000+250=1250ns (according to the standard-mode I²C bus specification) before the SCL line is released.

© 1995 Microchip Technology Inc.

FIGURE 17-11: SCI MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



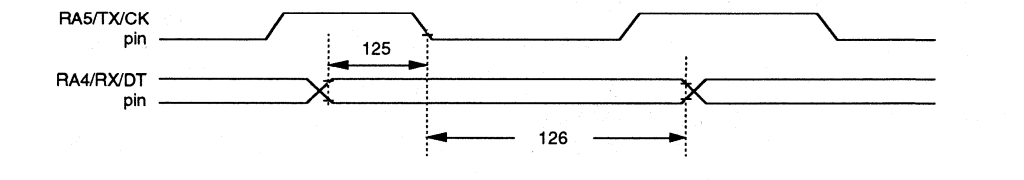
Note: Refer to Figure 17-2 for load conditions

TABLE 17-11: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	tckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	—	—	50	ns	
121	tckrf	Clock out rise time and fall time (Master Mode)	—	—	25	ns	
122	tdtrf	Data out rise time and fall time	—	—	25	ns	

†: Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-12: SCI MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 17-2 for load conditions

TABLE 17-12: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	tdtV2ckl	SYNC RCV (MASTER & SLAVE) Data hold before CK ↓ (DT hold time)	15	—	—	ns	
126	tckL2dtl	Data hold after CK ↓ (DT hold time)	15	—	—	ns	

†: Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

 © 1995 Microchip Technology Inc.

TABLE 17-13: A/D CONVERTER CHARACTERISTICS:
PIC16C74-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)
PIC16C74-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)
PIC16C74-20 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)
PIC16C73-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)
PIC16C73-10 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)
PIC16C73-20 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	NR	Resolution	—	—	8 bits	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NINT	Integral error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NDIF	Differential error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NFS	Full scale error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NOFF	Offset error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	—	Monotonicity	—	guaranteed	—	—	VSS ≤ AIN ≤ VREF
	VREF	Reference voltage	3.0V	—	VDD + 0.3	V	
	VAIN	Analog input voltage	Vss-0.3	—	VREF + 0.3	V	
	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	KΩ	
	IAD	A/D conversion current (VDD)	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
	IREF	VREF input current (Note 2)	—	—	1 10	mA μA	During sampling All other times

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

TABLE 17-14: A/D CONVERTER CHARACTERISTICS: PIC16LC74-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C73-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	NR	Resolution	—	—	8 bits	—	VREF = VDD = 3.0V (Note1)
	NINT	Integral error	—	—	less than ±1 LSb	—	VREF = VDD = 3.0V (Note1)
	NDIF	Differential error	—	—	less than ±1 LSb	—	VREF = VDD = 3.0V (Note1)
	NFS	Full scale error	—	—	less than ±1 LSb	—	VREF = VDD = 3.0V (Note1)
	NOFF	Offset error	—	—	less than ±1 LSb	—	VREF = VDD = 3.0V (Note1)
	—	Monotonicity	—	guaranteed	—	—	VSS ≤ AIN ≤ VREF
	VREF	Reference voltage	3.0V	—	VDD + 0.3	V	
	VAIN	Analog input voltage	Vss-0.3	—	VREF + 0.3	V	
	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	KΩ	
	IAD	A/D conversion current (VDD)	—	90	—	μA	Average current consumption when A/D is on. (Note2)
	IREF	VREF input current (Note3)	—	—	1 10	mA μA	During sampling All other times

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

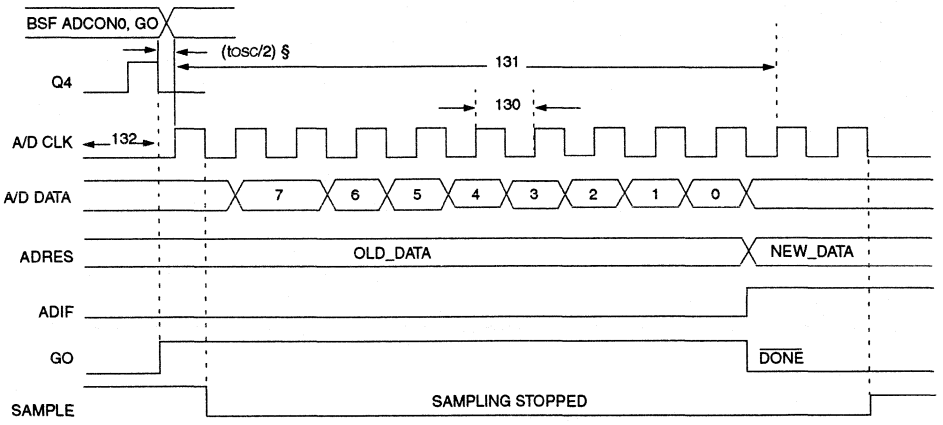
Note 1: These specifications apply if VREF = 3.0V and if VDD ≥ 3.0V. Vin must be between VSS and VREF

2: When A/D is off, it will not consume any current other than minor leakage current. The power down current spec includes any such leakage from the A/D module.

3: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

 © 1995 Microchip Technology Inc.

FIGURE 17-13: A/D CONVERSION TIMING



§: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the sleep instruction to be executed.

TABLE 17-15: A/D CONVERSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	1.6		—	µs	VREF≥3.0V
			2.0		—	µs	VREF full range
130	TAD	A/D Internal RC Oscillator source					ADCS1,0 = 11 (RC oscillator source)
			3.0	6.0	9.0	µs	PIC16LC74
			2.0	4.0	6.0	µs	PIC16C74
131	TCNV	Conversion time (not including S/H time)(Note 1)	—	9.5TAD	—	—	
132	TSMP	Sampling time	5	—	—	µs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1LSb (i.e. 20mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: ADRES register may be read on the following TCY cycle.

NOTES:

 © 1995 Microchip Technology Inc.

18.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES FOR PIC16C74 AND PIC16C73

Not Available at this time.

© 1995 Microchip Technology Inc.

NOTES:

 © 1995 Microchip Technology Inc.

19.0 ELECTRICAL CHARACTERISTICS FOR PIC16C71

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to+ 125°C
Storage Temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{MCLR}$) ..... -0.6V to VDD +0.6V
Voltage on VDD with respect to VSS ..... 0 to +7.5 V
Voltage on $\overline{MCLR}$ with respect to VSS (Note 2) ..... 0 to +14 V
Total power Dissipation (Note 1) 800mW
Maximum Current out of VSS pin 150mA
Maximum Current into VDD pin 100mA
Input clamp current, IIK (VI<0 or VI> VDD) ..... ±20mA
Output clamp current, IOK (V0 <0 or V0>VDD) ..... ±20mA
Maximum Output Current sunk by any I/O pin 25mA
Maximum Output Current sourced by any I/O pin 20mA
Maximum Current sunk by PORTA 80mA
Maximum Current sourced by PORTA 50mA
Maximum Current sunk by PORTB 150mA
Maximum Current sourced by PORTB 100mA

Note 1: Power dissipation is calculated as follows: Pdis = VDD x {IDD - $\sum$ IOH} + $\sum$ {(VDD-VOH) x IOH} + $\sum$(VOl x IOL)

Note 2: Voltage spikes below VSS at the $\overline{MCLR}$ pin, inducing currents greater than 80mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to VSS.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 19-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C71-04	16C71-20	16LC71-04
RC	VDD: 4.0V to 6.0V IDD: 3.3mA Max. at 5.5V IPD: 14µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 1.8mA typ. at 5.5V IPD: 1.0µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 1.4mA typ. at 3.0V IPD: 0.6µA typ. at 3V WDT dis Freq: 4MHz typ.
XT	VDD: 4.0V to 6.0V IDD: 3.3mA Max. at 5.5V IPD: 14µA Max. at 4V WDT dis Freq: 4MHz Max.	VDD: 4.5V to 5.5V IDD: 1.8mA typ. at 5.5V IPD: 1.0µA typ. at 4V WDT dis Freq: 4MHz Max.	VDD: 3.0V to 6.0V IDD: 1.4mA typ. at 3.0V IPD: 0.6µA typ. at 3V WDT dis Freq: 4MHz typ.
HS	VDD: 4.5V to 5.5V IDD: 13.5mA typ. at 5.5V IPD: 1.0µA typ. at 4.5V WDT dis Freq: 4MHz	VDD: 4.5V to 5.5V Idd: 30mA Max. at 5.5V IPD: 1.0µA typ. at 4.5V WDT dis Freq: 20MHz Max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 15µA typ. at 32 KHz, 4.0V IPD: 0.6µA typ. at 4.0V WDT dis Freq: 200KHz typ.	Do not use in LP mode	VDD: 3.0V to 6.0V IDD: 32µA Max. at 32 KHz, 3.0V IPD: 9µA Max. at 3.0V WDT dis Freq: 200KHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

© 1995 Microchip Technology Inc.

19.1 DC CHARACTERISTICS: PIC16C71-04 (Commercial, Industrial, Automotive) PIC16C71-20 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0 4.5	- -	6.0 5.5	V V	XT, RC and LP osc configuration HS osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2)	IDD	- - -	1.8 35 13.5	3.3 70 30	mA µA mA	FOSC = 4 MHz, VDD = 5.5V (Note 4) LP osc configuration (PIC16LC71-04) FOSC = 32 KHz, VDD = 4.0V, WDT disabled HS osc configuration (PIC16C71-20) FOSC = 20 MHz, VDD = 5.5V
Power Down Current (Note 3)	IPD	- - - -	7 1.0 1.0 1.0	28 14 16 20	µA µA µA µA	VDD=4.0V, WDT enabled,-40°C to +85°C VDD=4.0V, WDT disabled,-0°C to +70°C VDD=4.0V, WDT disabled,-40°C to +85°C VDD=4.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and V_{SS}.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

19.2 DC CHARACTERISTICS: PIC16LC71-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ + 125°C for automotive, -40°C ≤ TA ≤ + 85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 3.0V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	3.0	-	6.0	V	XT, RC, and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	-	VSS	-	V	See section on power-on reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	-	-	V/ms	See section on power-on reset for details
Supply Current (Note 2)	IDD	-	1.4	2.5	mA	FOSC = 4 MHz, VDD = 3.0V (Note 4)
		-	15	32	µA	FOSC = 32 KHz, VDD = 3.0V, WDT disabled
Power Down Current (Note 3)	IPD	-	5	20	µA	VDD=3.0V, WDT enabled,-40°C to +85°C
		-	0.6	9	µA	VDD=3.0V, WDT disabled,0°C to +70°C
		-	0.6	12	µA	VDD=3.0V, WDT disabled,-40°C to +85°C
		-	0.6	16	µA	VDD=3.0V, WDT disabled,-40°C to +125°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedence state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

19.3 DC CHARACTERISTICS: PIC16C71-04 (Commercial, Industrial, Automotive) PIC16C71-20 (Commercial, Industrial, Automotive) PIC16LC71-04 (Commercial, Industrial, Automotive)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ + 125°C for automotive,
-40°C ≤ TA ≤ + 85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Table 19-1 and Table 19-2

Characteristic	Sym	Min	Typ †	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	-	0.8V	V	
with Schmitt Trigger buffer		VSS	-	0.2VDD	V	
MCLR, RA4/T0CKI,OSC1 (in RC mode)		VSS	-	0.2VDD	V	
OSC1 (in XT, HS and LP)		VSS	-	0.3VDD	V	Note1
Input High Voltage						
I/O ports (Note 5)	VIH		-			
with TTL buffer		0.36VDD	-	VDD	V	4.5 ≤ VDD ≤5.5V
		0.45VDD	-	VDD		Otherwise for entire VDD range
with Schmitt Trigger buffer		0.85VDD	-	VDD		For entire VDD range
MCLR RA4/T0CKI		0.85 VDD	-	VDD	V	
OSC1 (XT, HS and LP)		0.7 VDD	-	VDD	V	Note1
PortB weak pull-up current	IPURB	50	250	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)						
I/O ports	IIL	-	-	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RA4/T0CKI		-	-	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1		-	-	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	-	-	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
		-	-	0.6	V	IOL = 7.0mA, VDD-4.5V, -40°C to +125°C
OSC2/CLKOUT		-	-	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
(RC osc configuration)		-	-	0.6	V	IOL = 1.2mA, VDD-4.5V, -40°C to +125°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD-0.7	-	-	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
		VDD-0.7	-	-	V	IOH = -2.5mA, VDD=4.5V, -40°C to +125°C
OSC2/CLKOUT		VDD-0.7	-	-	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
(RC osc configuration)		VDD-0.7	-	-	V	IOH = -1.0mA, VDD=4.5V, -40°C to +125°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO			50	pF	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC16C71 be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: The user may use better of the two specs.

5: PIC16C71 Rev. A INT pin has a TTL input buffer. PIC16C71 Rev. B INT pin has a Schmitt trigger input buffer.

© 1995 Microchip Technology Inc.

19.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I^2C specifications only)
4. Ts (I^2C specifications only)

T			
F	Frequency	T	Time

Lowercase subscripts (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O port	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance
I^2C only			
AA	output access	High	High
BUF	Bus free	Low	Low

TCC:ST (I^2C specifications only)

CC			
HD	Hold	SU	Setup
ST			
DAT	DATA input hold	STO	STOP condition
STA	START condition		

© 1995 Microchip Technology Inc.

19.5 Timing Diagrams and Specifications

FIGURE 19-1: EXTERNAL CLOCK TIMING

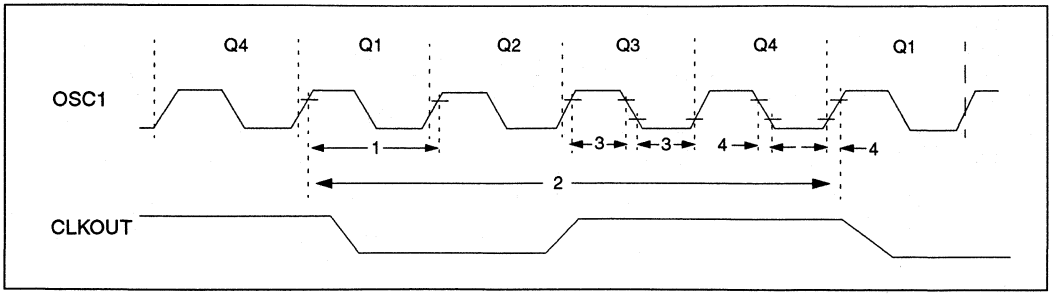


TABLE 19-2: EXTERNAL CLOCK TIMING REQUIREMENTS

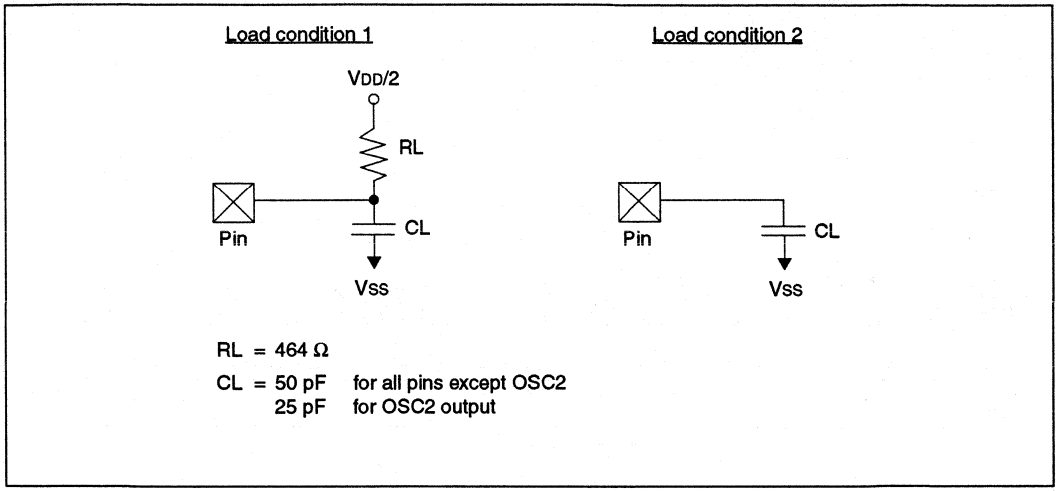
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fos	**External CLKIN Frequency (Note 1)**	DC	—	4	MHz	XT and RC osc mode
			DC	—	4	MHz	HS osc mode (PIC16C71-04, PIC16C71-04)
			DC	—	20	MHz	HS osc mode (PIC16C71-20)
			DC	—	200	KHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			1	—	4	MHz	HS osc mode (PIC16C71-04, PIC16C71-04)
			1	—	20	MHz	HS osc mode (PIC16C71-20)
1	Tosc	**External CLKIN Period (Note 1)**	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C71-04, PIC16C71-04)
			50	—	—	ns	HS osc mode (PIC16C71-20)
			50	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	1,000	ns	HS osc mode (PIC16C71-04, PIC16C71-04)
			50	—	1,000	ns	HS osc mode (PIC16C71-20)
			5	—	—	µs	LP osc mode
2	Tcy	**Instruction Cycle Time (Note 1)**	1.0	4/Fosc	DC	µs	
3	TosL, TosH	Clock in (OSC1) High or Low Time	50	—	—	ns	XT oscillator
			2.5	—	—	µs	LP oscillator
			10	—	—	ns	HS oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	25	—	—	ns	XT oscillator
			50	—	—	ns	LP oscillator
			15	—	—	ns	HS oscillator

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

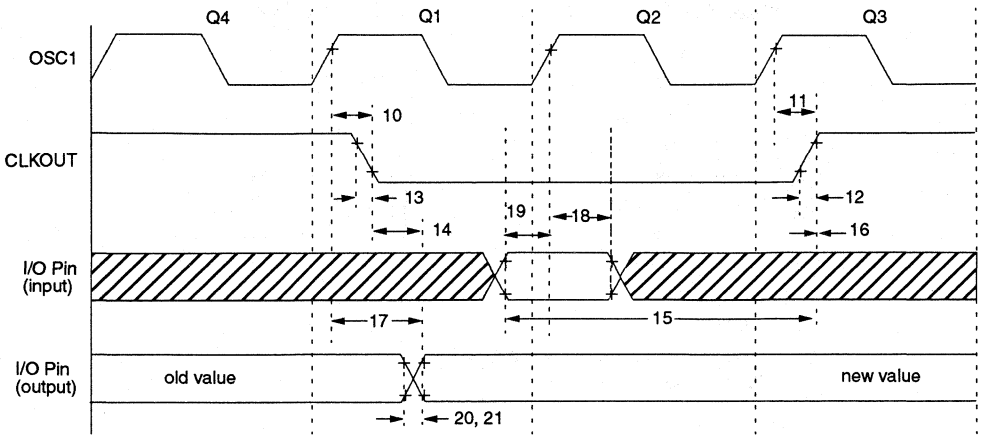
© 1995 Microchip Technology Inc.

FIGURE 19-2: LOAD CONDITIONS



© 1995 Microchip Technology Inc.

FIGURE 19-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 19-2 for load conditions.

TABLE 19-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT ↑	0.25 TCY+30	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT ↑	0	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	80 - 100	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	INT pin high or low time	20	—	—	ns	
23††	Trbp	RB<7:4> change INT high or low time	20	—	—	ns	

* These parameters are characterized but not tested.

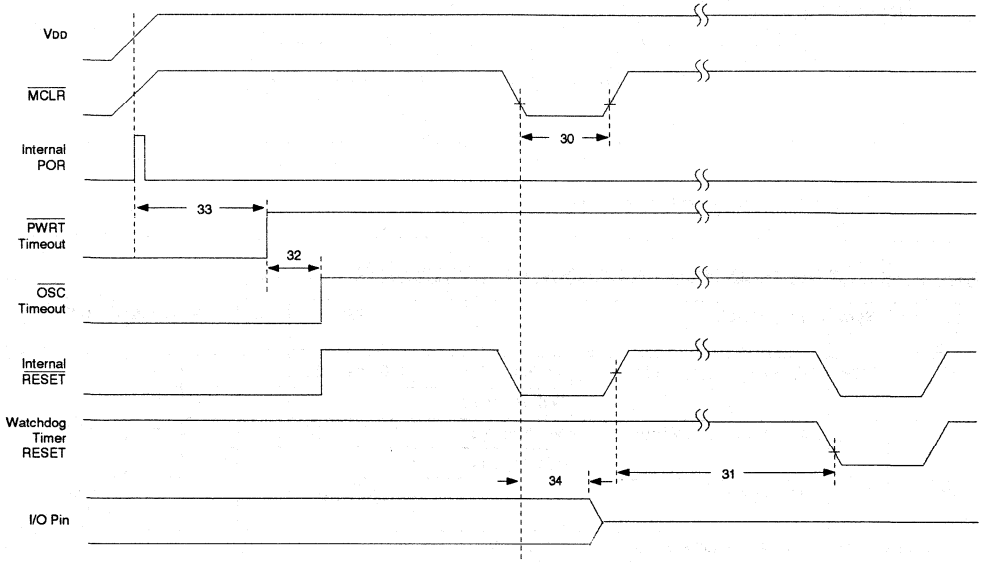
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x Tosc

© 1995 Microchip Technology Inc.

FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 19-2 for load conditions.

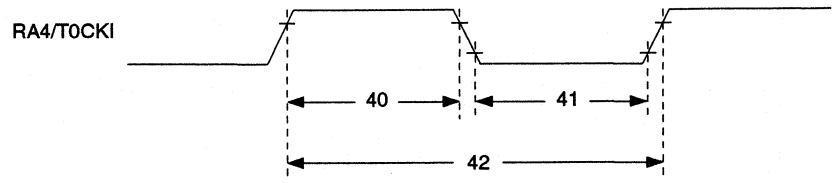
TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	200	—	—	ns	VDD = 5V, -40°C to +125°C
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period		1024 Tosc		ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	Tioz	I/O High Impedance from MCLR Low			100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-5: TIMER0 CLOCK TIMINGS



Note: Refer to Figure 19-2 for load conditions.

TABLE 19-5: TIMER0 AND TIMER1 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40^*}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

TABLE 19-6: A/D CONVERTER CHARACTERISTICS: PIC16C71-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE) PIC16C71-20 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	NR	Resolution	—	—	8 bits	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NINT	Integral error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NDIF	Differential error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NFS	Full scale error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	NOFF	Offset error	—	—	less than ±1 LSb	—	VREF = VDD = 5.12V, VSS ≤ AIN ≤ VREF
	—	Monotonicity	—	guaranteed	—	—	VSS ≤ AIN ≤ VREF
	VREF	Reference voltage	3.0V	—	VDD + 0.3	V	
	VAIN	Analog input voltage	Vss-0.3	—	VREF	V	
	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	KΩ	
	IAD	A/D conversion current (VDD)	—	180	—	μA	Average current consumption when A/D is on. (Note 1)
	IREF	VREF input current (Note 2)	—	—	1 40	mA μA	During sampling All other times

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

TABLE 19-7: A/D CONVERTER CHARACTERISTICS: PIC16LC71-04 (COMMERCIAL, INDUSTRIAL, AUTOMOTIVE)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	NR	Resolution	—	—	8 bits	—	VREF = VDD = 3.0V (Note1)
	NINT	Integral error	—	—	less than ±2 LSb	—	VREF = VDD = 3.0V (Note1)
	NDIF	Differential error	—	—	less than ±2 LSb	—	VREF = VDD = 3.0V (Note1)
	NFS	Full scale error	—	—	less than ±2 LSb	—	VREF = VDD = 3.0V (Note1)
	NOFF	Offset error	—	—	less than ±2 LSb	—	VREF = VDD = 3.0V (Note1)
	—	Monotonicity	—	guaranteed	—	—	VSS ≤ AIN ≤ VREF
	VREF	Reference voltage	3.0V	—	VDD + 0.3	V	
	VAIN	Analog input voltage	Vss-0.3	—	VREF	V	
	ZAIN	Recommended impedance of analog voltage source	—	—	10.0	KΩ	
	IAD	A/D conversion current (VDD)	—	90	—	μA	Average current consumption when A/D is on. (Note2)
	IREF	VREF input current (Note3)	—	—	1 10	mA μA	During sampling All other times

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

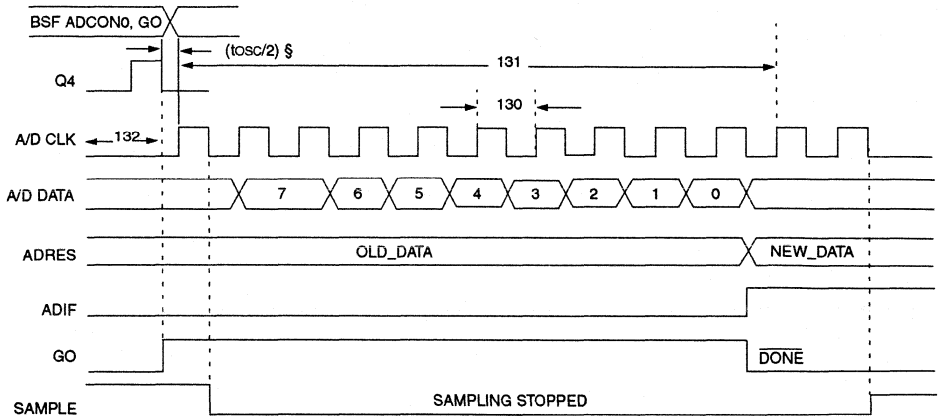
Note 1: These specifications apply if VREF = 3.0V and if VDD ≥ 3.0V. Vin must be between Vss and VREF

2: When A/D is off, it will not consume any current other than minor leakage current. The power down current spec includes any such leakage from the A/D module.

3: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

© 1995 Microchip Technology Inc.

FIGURE 19-6: A/D CONVERSION TIMING



§: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the sleep instruction to be executed.

TABLE 19-8: A/D CONVERSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	2.0		—	µs	
130	TAD	A/D Internal RC Oscillator source					ADCS1,0 = 11 (RC oscillator source)
			3.0	6.0	9.0	µs	PIC16LC71
			2.0	4.0	6.0	µs	PIC16C71
131	TCNV	Conversion time (not including S/H time)(Note 1)	—	10 TAD	—	—	
132	TSMP	Sampling time	5	—	—	µs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1LSb (i.e. 20mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: ADRES register may be read on the following TCY cycle.

NOTES:

© 1995 Microchip Technology Inc.

20.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES FOR PIC16C71 (ONLY)

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. 'Typical' represents the mean of the distribution while 'max' or 'min' represents (mean + 3s) and (mean - 3s) respectively where s is standard deviation.

FIGURE 20-1: TYPICAL RC OSCILLATOR FREQUENCY vs TEMPERATURE

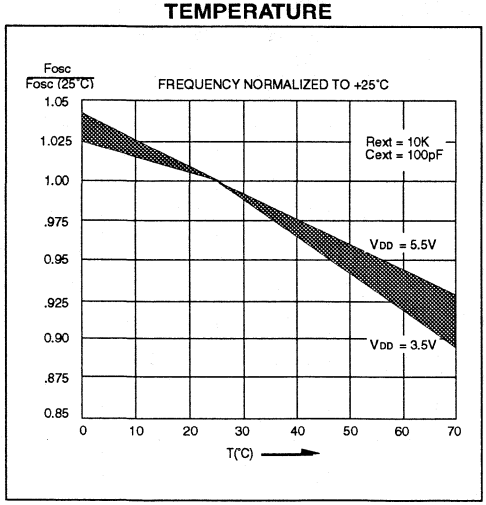


FIGURE 20-2: TYPICAL RC OSCILLATOR FREQUENCY vs VDD

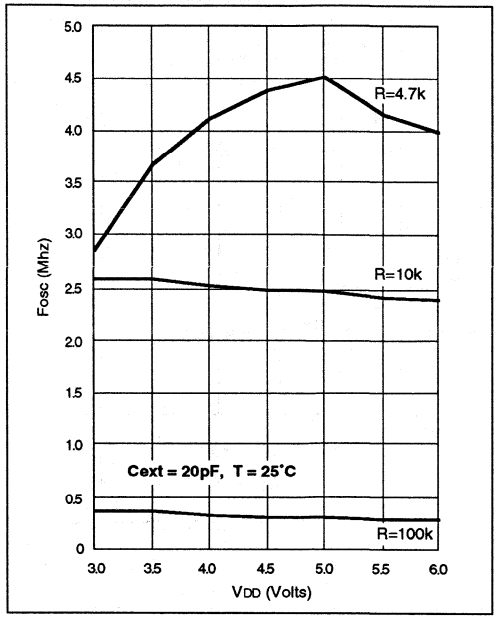
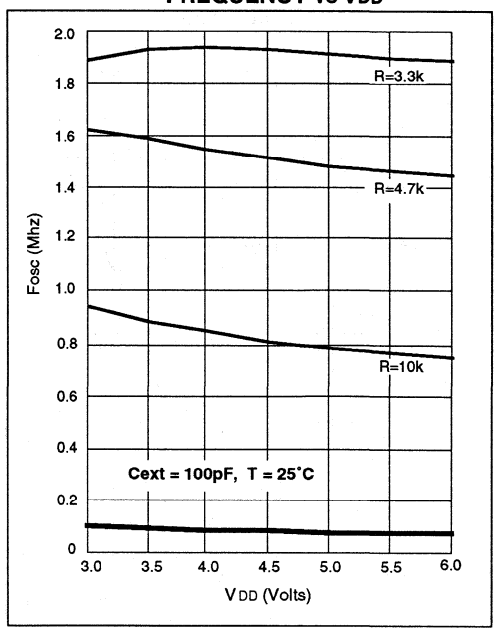


FIGURE 20-3: TYPICAL RC OSCILLATOR FREQUENCY vs VDD



© 1995 Microchip Technology Inc.

FIGURE 20-4: TYPICAL RC OSCILLATOR FREQUENCY vs VDD

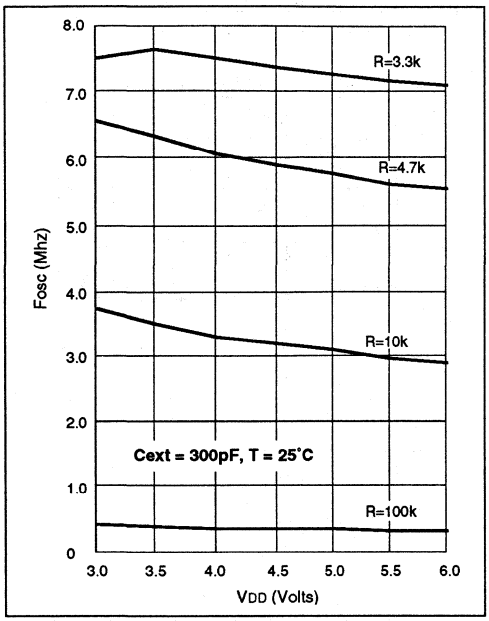


FIGURE 20-5: TYPICAL IPD vs VDD WATCHDOG TIMER DISABLED 25°C

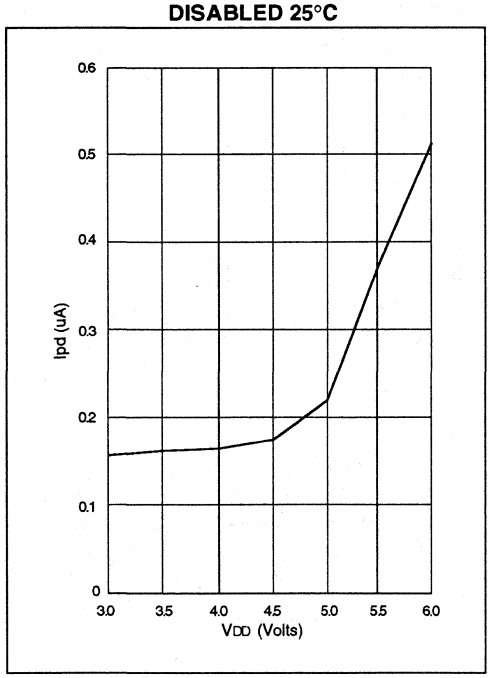
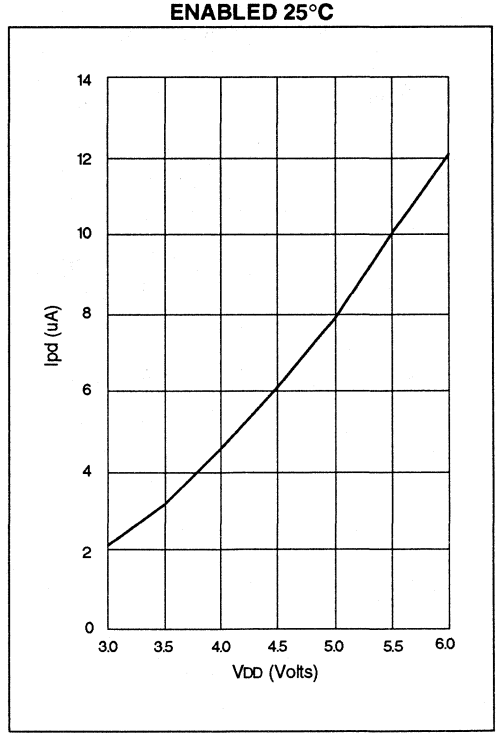


TABLE 20-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
20pf	4.7K	4.52 MHz	±17.35%
	10k	2.47 MHz	±10.10%
	100k	290.86 KHz	±11.90%
100pf	3.3k	1.92 MHz	±9.43%
	4.7k	1.48 MHz	±9.83%
	10k	788.77 KHz	±10.92%
	100k	88.11 KHz	±16.03%
300pf	3.3k	726.89 KHz	±10.97%
	4.7k	573.95 KHz	±10.14%
	10k	307.31 KHz	±10.43%
	100k	33.82 KHz	±11.24%

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for VDD = 5V.

FIGURE 20-6: TYPICAL IPD vs VDD WATCHDOG TIMER ENABLED 25°C



© 1995 Microchip Technology Inc.

FIGURE 20-7: MAXIMUM IPD VS VDD WATCHDOG DISABLED

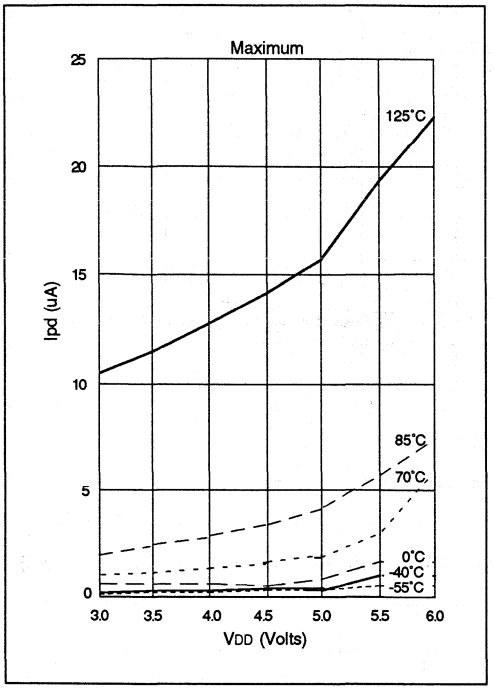
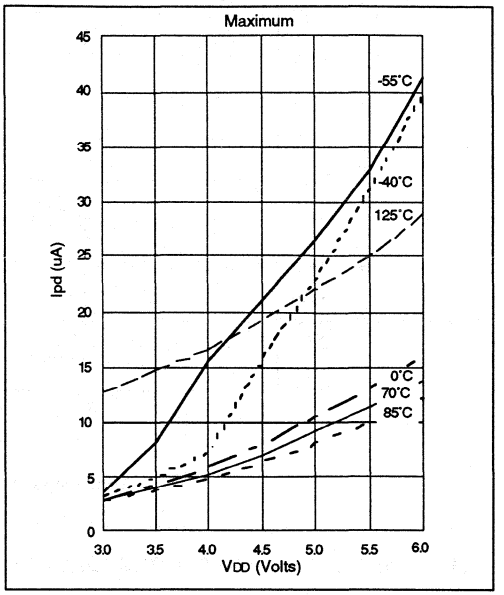


FIGURE 20-8: MAXIMUM IPD VS VDD WATCHDOG ENABLED*



*Ipd, with watchdog timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the watchdog timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 20-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS VS VDD

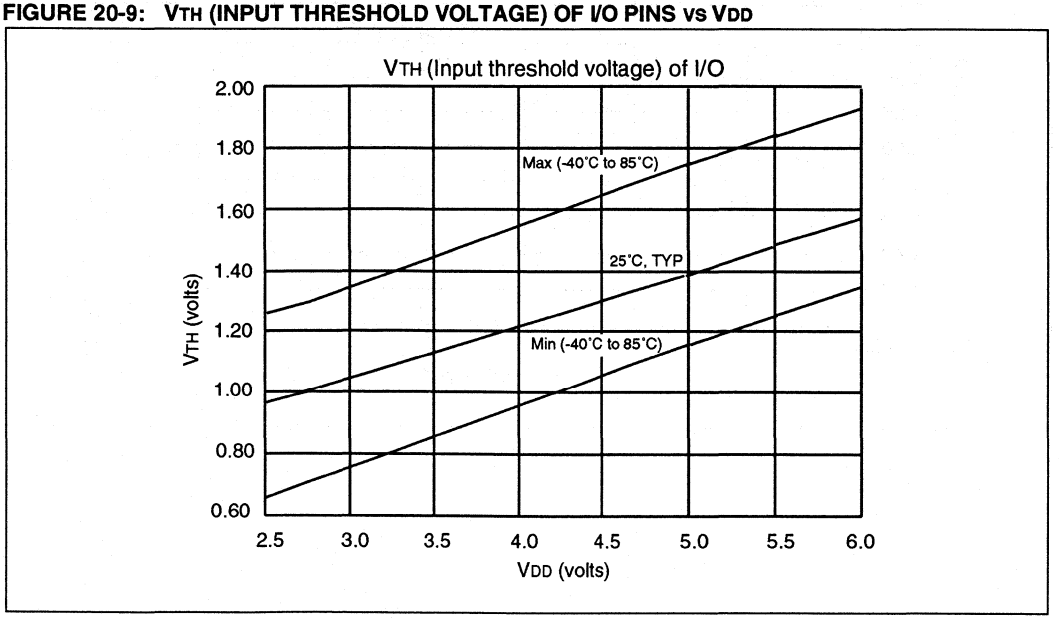
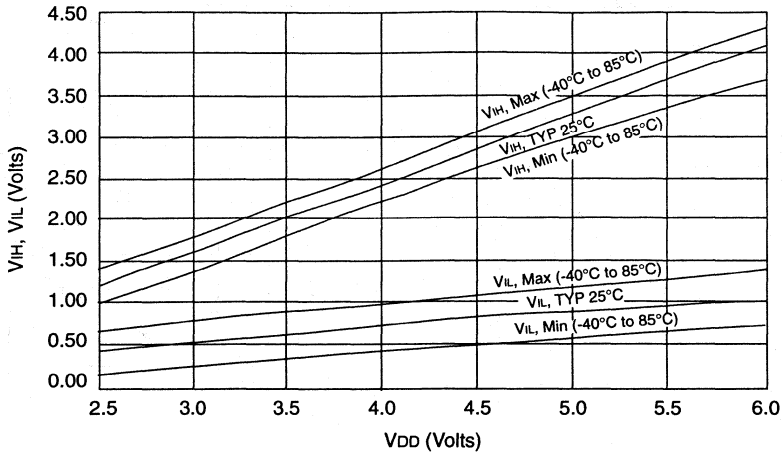
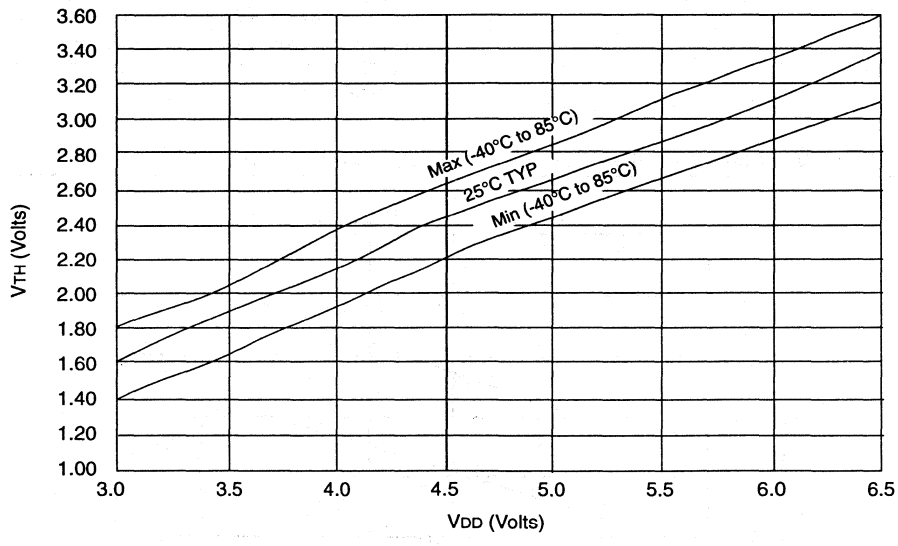


FIGURE 20-10: VIH, VIL OF $\overline{MCLR}$, T0CKI AND OSC1 (IN RC MODE) VS VDD



Note: These input pins have Schmitt trigger input buffer.

FIGURE 20-11: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) VS VDD



© 1995 Microchip Technology Inc.

FIGURE 20-12: TYPICAL IDD VS FREQ (EXT CLOCK, 25°C)

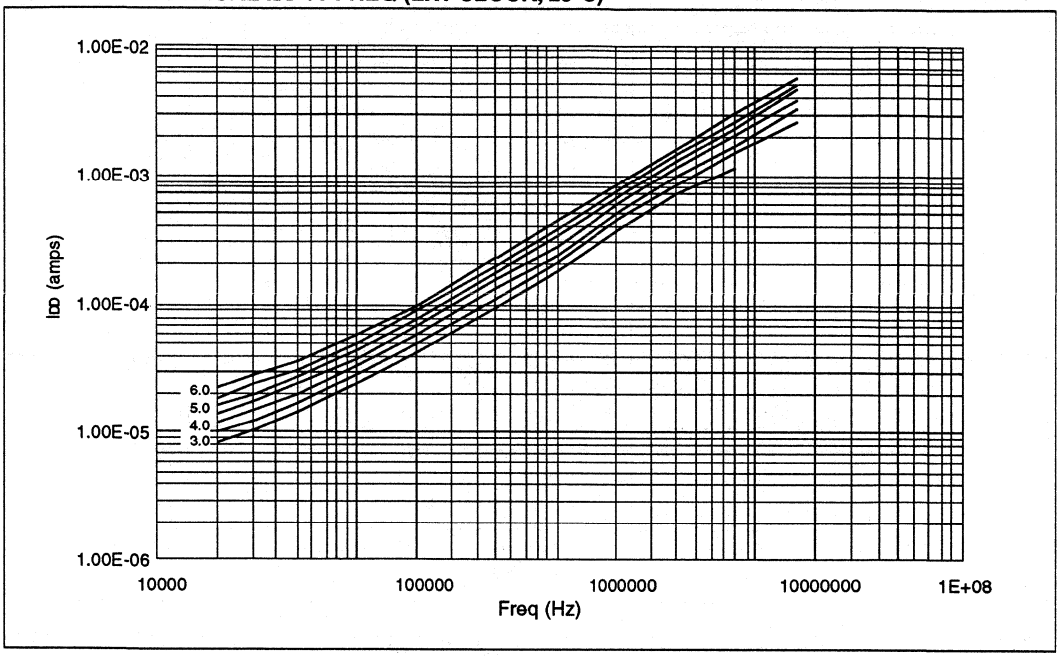


FIGURE 20-13: MAXIMUM, IDD VS FREQ (EXT CLOCK, -40° TO +85°C)

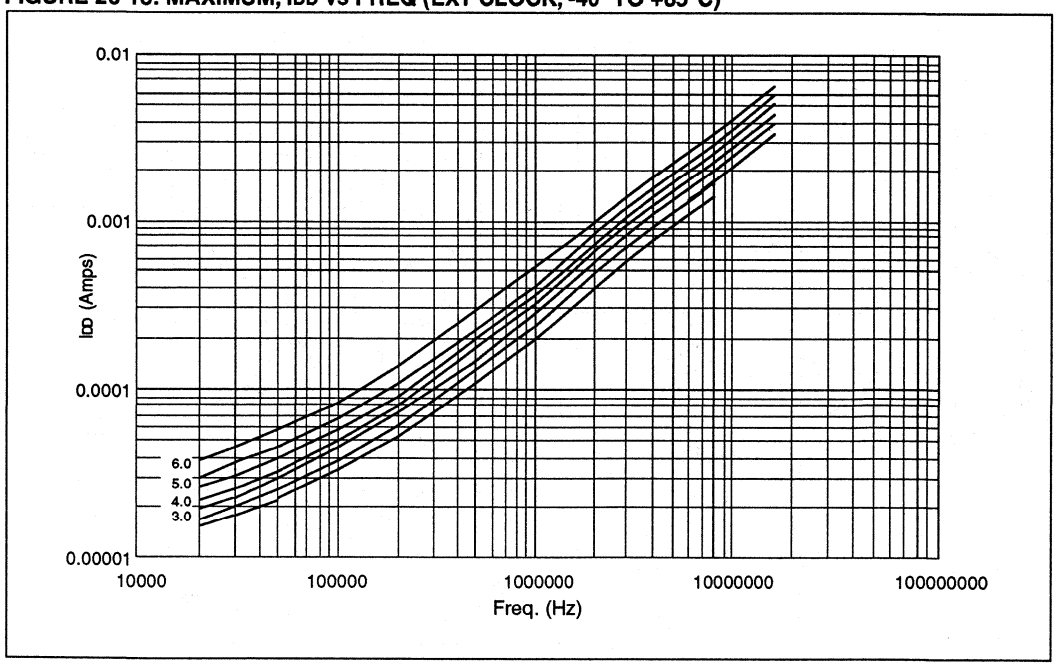


FIGURE 20-14: MAXIMUM IDD vs FREQ WITH A/D OFF (EXT CLOCK, -55° TO +125°C)

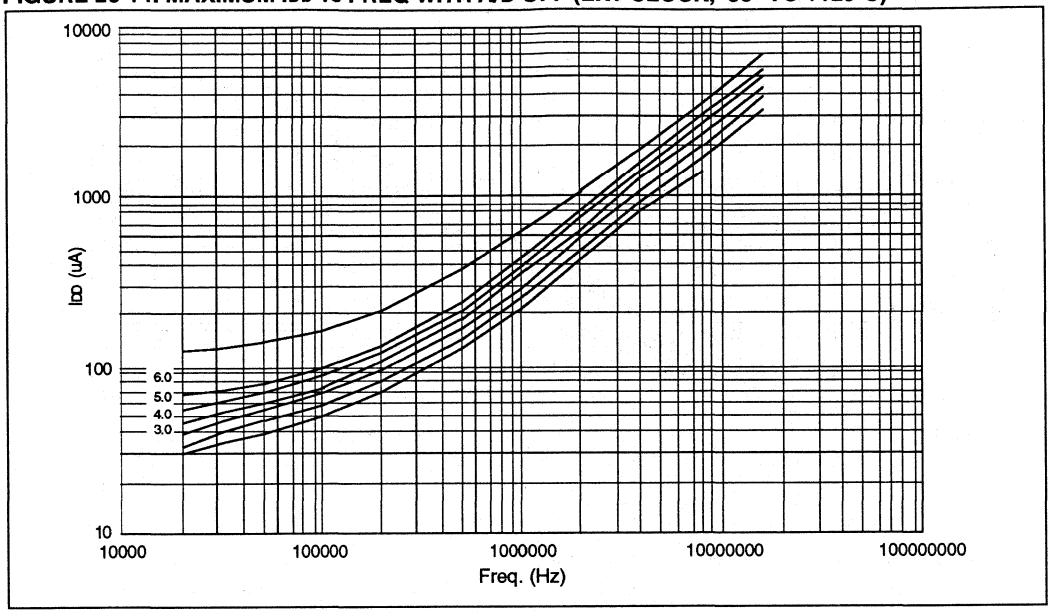


FIGURE 20-15: WDT TIMER TIME-OUT PERIOD vs VDD

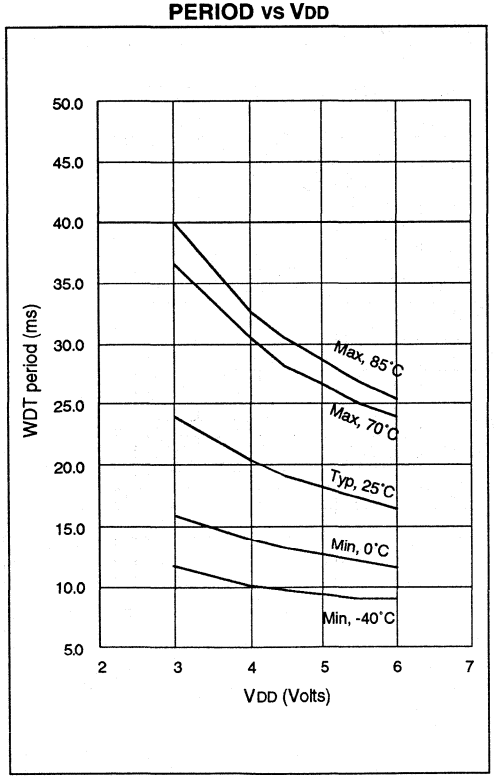
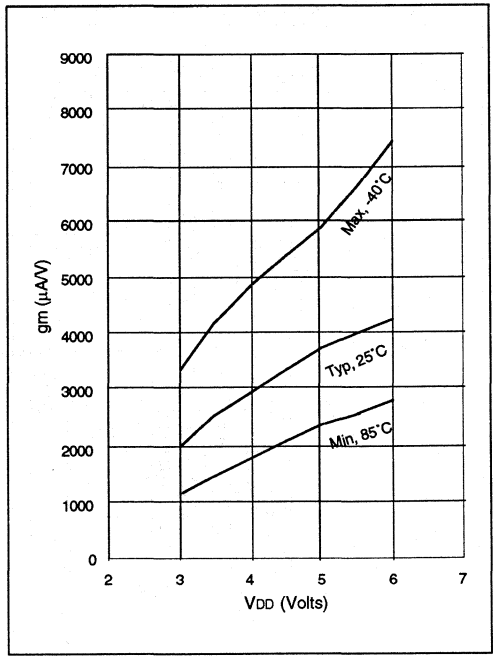


FIGURE 20-16: TRANSCONDUCTANCE (GM) OF HS OSCILLATOR vs VDD



© 1995 Microchip Technology Inc.

FIGURE 20-17: TRANSCONDUCTANCE (GM) OF LP OSCILLATOR VS VDD

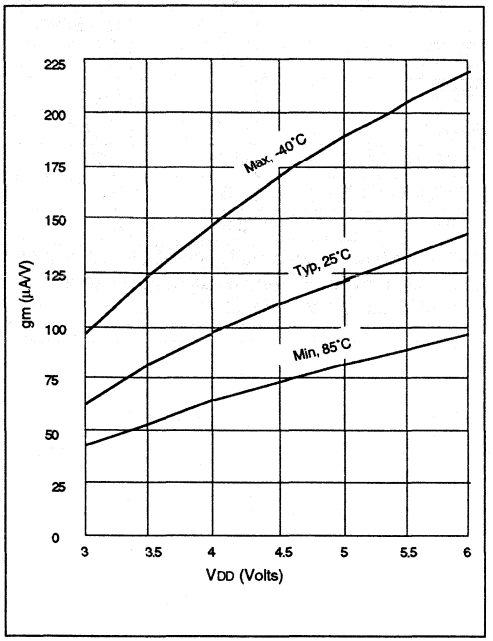


FIGURE 20-18: TRANSCONDUCTANCE (GM) OF XT OSCILLATOR VS VDD

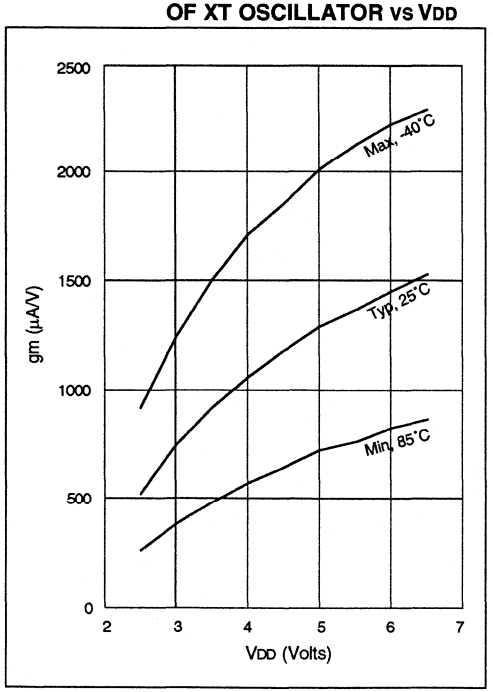


FIGURE 20-19: IOH VS VOH, VDD = 3V

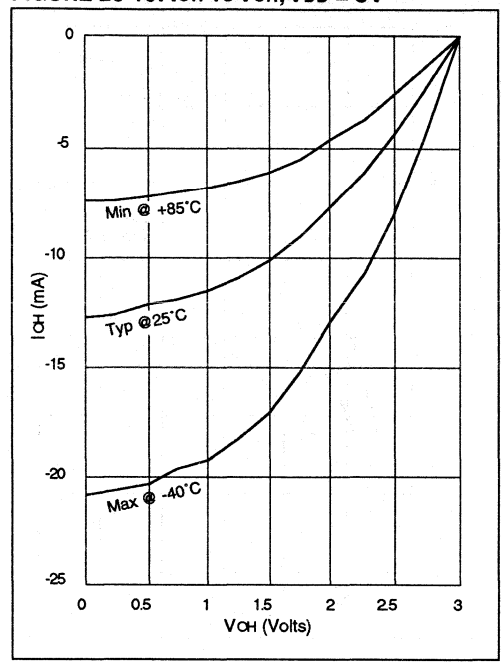
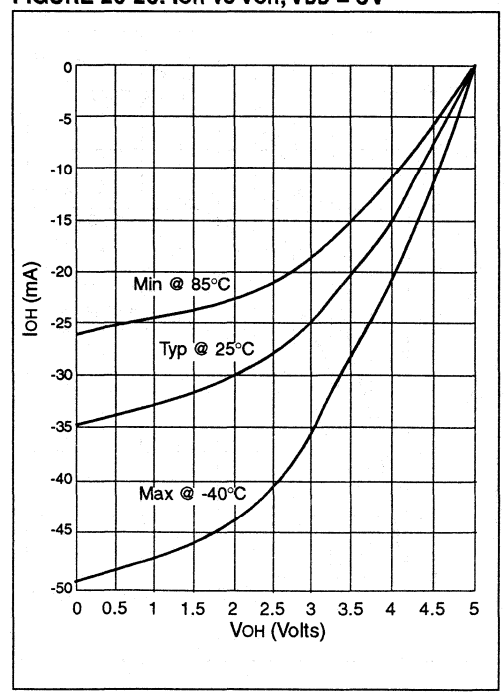


FIGURE 20-20: IOH VS VOH, VDD = 5V



© 1995 Microchip Technology Inc.

FIGURE 20-21: IOL VS VOL, VDD = 3V

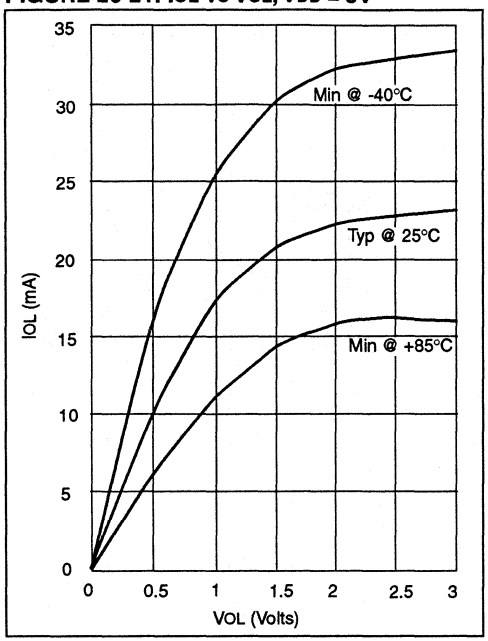


FIGURE 20-22: IOL VS VOL, VDD = 5V

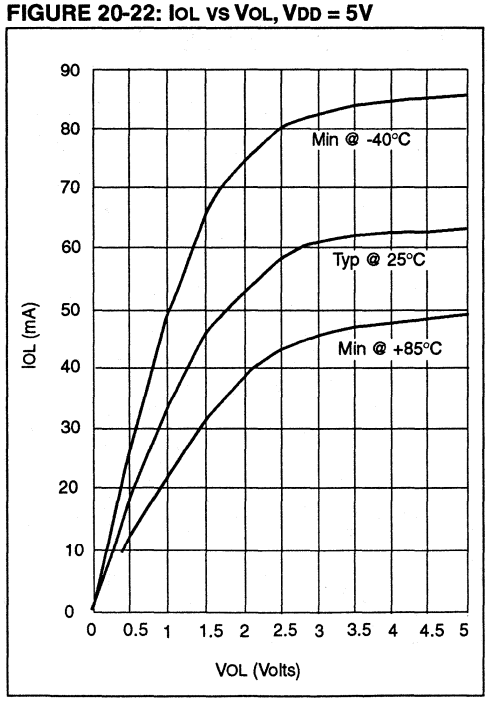


TABLE 20-2: INPUT CAPACITANCE*

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
RA port	5.0	4.3
RB port	5.0	4.3
MCLR	17.0	17.0
OSC1/CLKN	4.0	3.5
OSC2/CLKOUT	4.3	3.5
TMR0	3.2	2.8

*All capacitance values are typical at 25°C. A part to part variation of ±25% (three standard deviations) should be taken into account.

© 1995 Microchip Technology Inc.

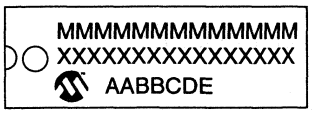
21.0 PACKAGING INFORMATION

For Package Dimensions please refer to the Packaging Section of the Data Book.

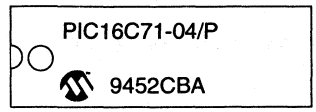
© 1995 Microchip Technology Inc.

21.1 Package Marking Information

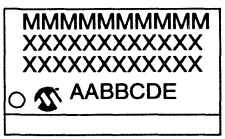
18-Lead PDIP



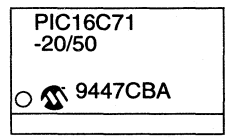
Example



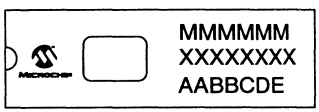
18-Lead SOIC



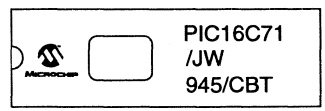
Example



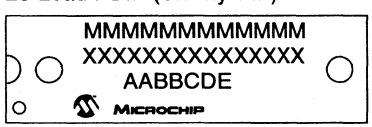
18-Lead CERDIP Windowed



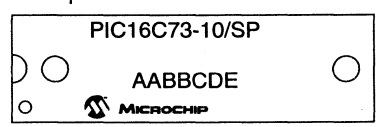
Example



28-Lead PDIP (Skinny DIP)



Example



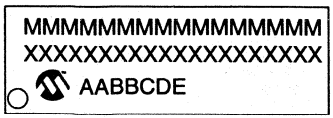
Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

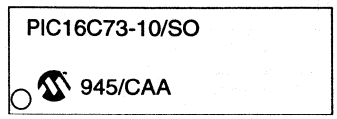
© 1995 Microchip Technology Inc.

Package Marking Information (Cont'd)

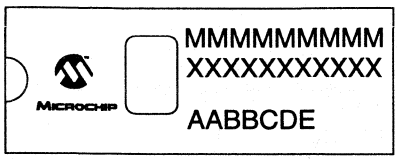
28-Lead SOIC



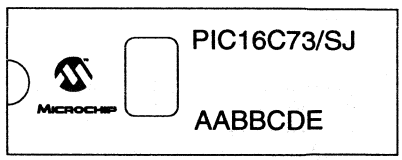
Example



28-Lead CERDIP Windowed

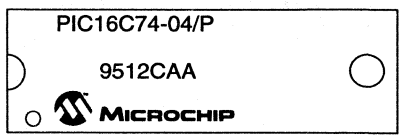


Example

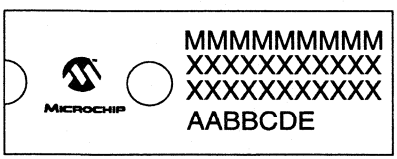


40-Lead PDIP

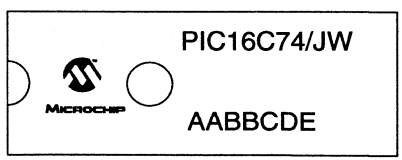
Example



40-Lead CERDIP Windowed



Example

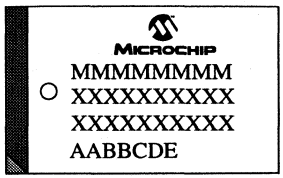


Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

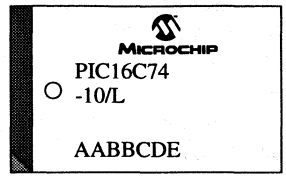
* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

Package Marking Information (Cont'd)

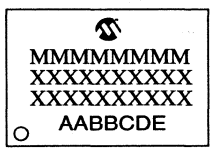
44-Lead PLCC



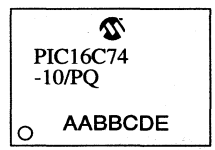
Example



44-Lead MQFP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calender year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.
	D_1	Mask revision number for microcontroller
	E	Assembly code of the plant or country of origin in which part was assembled.
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

APPENDIX A:

The following are the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bit. This allows larger page sizes both in program memory (4K now as opposed to 512 before) and register file (192 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. PA2, PA1, PA0 bits are removed from status register.
3. Data memory paging is redefined slightly. Status register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions TRIS and OPTION are being phased out although they are kept for compatibility with PIC16C5x.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers oscillator start-up timer (OST) and power-up timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PortB has weak pull-ups and interrupt on change feature.
13. RTCC pin is also a port pin (RA4) now.
14. FSR is made a full eight bit register.
15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, /VPP, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on-Reset (POR) status bit.
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S NEW

The format of this data sheet has been changed to be consistent with other product families. This ensures that important topics are covered across all PIC16/17 families. Here is an overview list of new features:

- Data Sheet Structure / Outline
- Section on Table Read and Table Writes

APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This is so that control bits that do the same function, have the same name (regardless of processor family). Care must still be taken, since they may not be in the same special function register. The following shows the register and bit names that have been changed:

TABLE 21-1: REGISTER NAME CHANGESBIT NAME CHANGES

OLD NAME	NEW NAME
RTE	TOSE
RTS	TOCS
$\overline{\text{T1INSYNC}}$	$\overline{\text{T1SYNC}}$

© 1995 Microchip Technology Inc.

APPENDIX E: PIC16/17 MICROCONTROLLERS

TABLE E-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory			Peripherals					Features				
	Maximum Frequency of Operation (MHz)	Program Memory (Bytes)	EPROM	RAM Data Memory (Bytes)	Timer Module(s)	Capture	PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0, TMR1, TMR2, TMR3	2	2	Yes	1	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0, TMR1, TMR2, TMR3	2	2	Yes	1	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0, TMR1, TMR2, TMR3	2	2	Yes	1	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller mode.

3: PORTB has software-configurable weak pull-ups.

TABLE E-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	Program Memory EPROM	Program Memory EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE E-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72		TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72		TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73		TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73		TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

E.1 Pin Compatibility

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin (20 pin)
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

INDEX

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF EXAMPLES

LIST OF FIGURES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF TABLES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER.↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER.↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER.↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER.↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks, and SQTP is a service mark of Microchip Technology Incorporated.

ACCESS.bus is a trademark of the ACCESS.bus Industry Group (ABIG).

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16C7X Literature Number: DS30390A

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefullness?

6. Is there any incorrect or misleading information (what and where)?

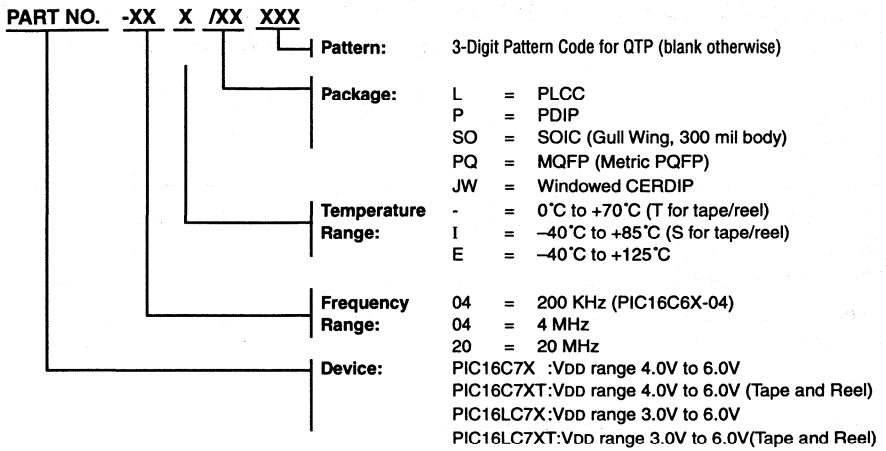
7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

© 1995 Microchip Technology Inc.

PIC16C7X Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.



Examples:

a) PIC16C71 - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301
b) PIC16LC73 - 041/SO = Industrial temp., SOIC package, 4MHz, extended VDD limits
c) PIC16C74 - 10E/P = Automotive temp., PDIP package, 10 MHz, normal VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

MICROCHIP

PIC16C84

8-Bit CMOS EEPROM Microcontroller

High Performance RISC-like CPU Features

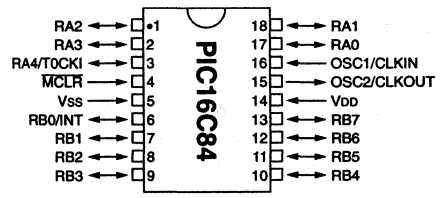
- Only 35 single word instructions to learn
- All instructions single cycle (400 ns) except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle
- 14-bit wide instructions
- 8-bit wide data path
- 1K x 14 On-chip EEPROM program memory
- 36 x 8 general purpose registers (SRAM)
- 15 special function hardware registers
- 64 x 8 EEPROM data memory
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on change
  - Data EEPROM write complete
- 1,000,000 ERASE/WRITE cycles - Typical (Data Memory)
- Data Retention >40 years

Peripheral Features

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Configuration

PDIP, SOIC



Special Microcontroller Features

- Power-On Reset (POR)
- Power-Up Timer (PWRT)
- Oscillator Start-Up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options:
  - RC: Low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High-speed crystal/resonator
  - LP: Power saving, low frequency crystal
- Serial In-System Programming (via two pins)

CMOS Technology

- Low-power, high-speed CMOS EEPROM technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 6.0V
  - Industrial: 2.0V to 6.0V
- Low power consumption:
  - < 2 mA @ 5V, 4 MHz
  - 60 µA typical @ 2V, 32 kHz
  - 26 µA typical standby current @ 2V

© 1995 Microchip Technology Inc.

TABLE OF CONTENTS

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error from the previous version of the PIC16C84 Data Sheet (Literature Number DS30081D), please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

To assist you in the use of this document, Appendix C contains a list of new information in this data sheet, while Appendix D contains information that has changed

© 1995 Microchip Technology Inc.

1.0 GENERAL DESCRIPTION

The PIC16C84 is a low-cost, high-performance, CMOS, fully-static, 8-bit microcontroller. All PIC16/17 microcontrollers employ an advanced RISC-like architecture. PIC16C84 devices have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with a separate 8-bit wide data bus. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C84 microcontrollers typically achieve a 2:1 code compression and a 2:1 speed improvement over other 8-bit microcontrollers in its class.

The PIC16C84 has 36 bytes of RAM, 64 bytes of Data EEPROM memory, and 13 I/O pins. A timer/counter is also available.

The PIC16CXX family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power saving. The user can wake the chip from SLEEP through several external and internal interrupts and resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

The EEPROM program memory allows the same device package to be used for prototyping and production. In-circuit reprogrammability allows the code to be updated without the device being removed from the end application. This is useful in the development of many applications where the device may not be easily accessible, but the prototypes may require code updates. This is also useful for remote applications, where the code may need to be updated (such as rate information).

Table 1-1 lists the features of the PIC16C84.

Figure 3-1 is a simplified block diagram of the PIC16C84.

The PIC16C84 fits perfectly in applications ranging from high speed automotive and appliance motor control to low-power remote sensors, electronic locks, security devices, and smart cards. The EEPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16C84 very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, serial communication, capture and compare, PWM functions and co-processor applications).

The in-system programming feature (via two pins) offers flexibility of customizing the product after complete assemble and test. This feature can be used to serialize a product, store calibration data, or program the device with the current firmware before shipping.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for PIC16C5X can be easily ported to the PIC16C84 (Appendix B).

1.2 Development Support

The PIC16CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A "C" compiler and fuzzy logic support tools are also available.

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	EPROM Program Memory	EEPROM Program Memory	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

2.0 PIC16C84 DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements the proper device option can be selected using the information in this section. When placing orders, please use the "PIC16C84 Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 Electrically Erasable Devices

All PIC16C84 versions are electrically erasable. These devices are offered in the lower cost plastic package, even though the device can be erased and reprogrammed. This allows the same device to be used for prototype development and pilot programs as well as production.

A further advantage of the electrically erasable version can be erased and reprogrammed in-circuit, or by Microchip's PICSTART™ or PRO MATE™ programmers.

2.2 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices have all EEPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your Microchip's Technology sales office for more details.

2.3 Serialized Quick-Turnaround-Production (SQTP^{SM}) Devices

Microchip offers the unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C84 can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C84 uses a Harvard architecture, in which, program and data are accessed from separate memories. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (see Example 3-1). Consequently, all instructions execute in a single cycle (400 ns @ 10 MHz) except for program branches.

The PIC16C84 addresses 1K x 14 program memory. All program memory is internal.

The PIC16C84 can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. An orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16C84 simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16C84 devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register), and the other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

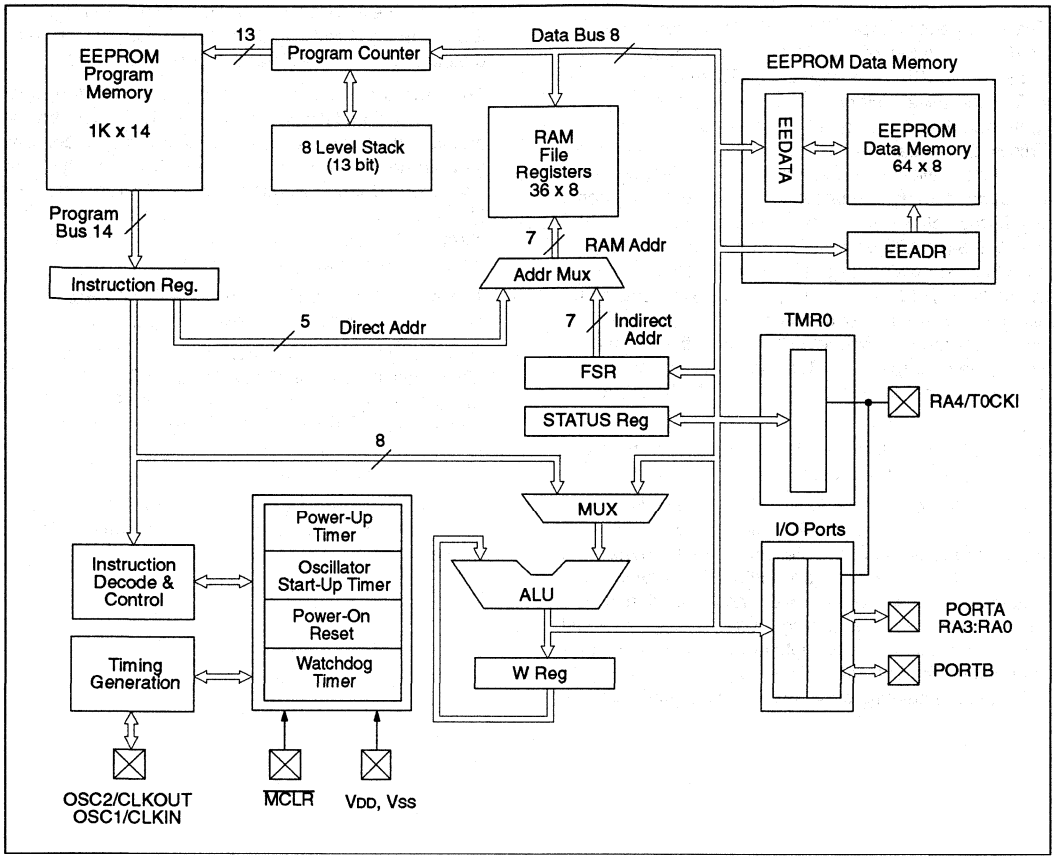
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram for the PIC16C84 is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

© 1995 Microchip Technology Inc.

FIGURE 3-1: PIC16C84 BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

TABLE 3-1: PIC16C84 PINOUT DESCRIPTION

Pin Name	DIP No.	SOIC No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	I	ST/CMOS[3]	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0	17	17	I/O	TTL	
RA1	18	18	I/O	TTL	
RA2	1	1	I/O	TTL	
RA3	2	2	I/O	TTL	
RA4/T0CKI	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open collector type.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	I/O	TTL	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	I/O	TTL	
RB2	8	8	I/O	TTL	
RB3	9	9	I/O	TTL	
RB4	10	10	I/O	TTL	Interrupt on change pin.
RB5	11	11	I/O	TTL	Interrupt on change pin.
RB6	12	12	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming clock.
RB7	13	13	I/O	TTL/ST[2]	Interrupt on change pin. Serial programming data.
VSS	5	5	P	—	Ground reference for logic and I/O pins.
VDD	14	14	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = output I/O = Input/Output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

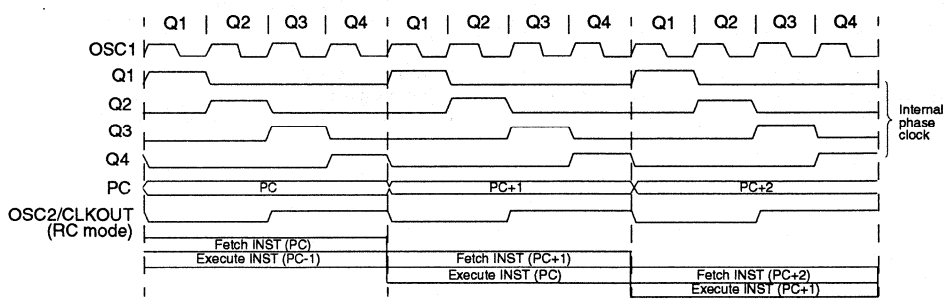
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-1).

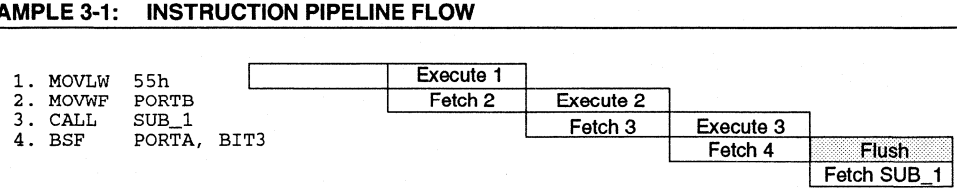
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the "Instruction Register" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

4.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16C84. These are the program memory and the data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

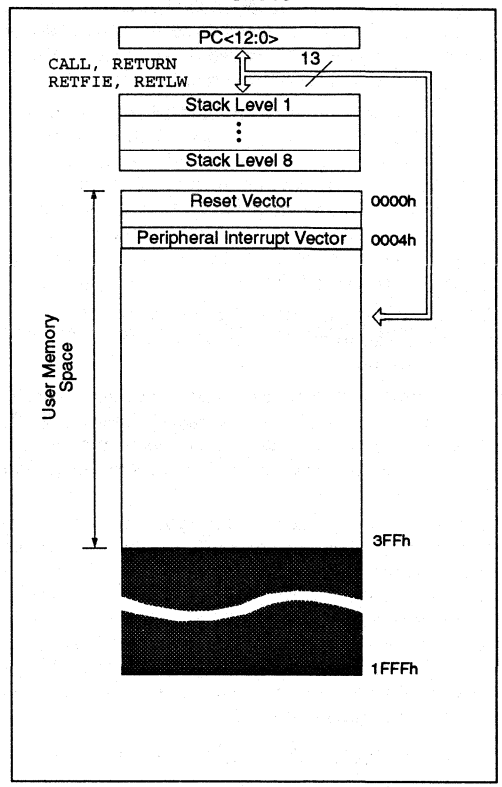
The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory, but is indirectly mapped. That is an indirect address pointer specifies the address of the data EEPROM memory to read/write. The 64 bytes of data EEPROM memory have the address range 0 - 3Fh. More details on the EEPROM memory can be found in Section 7.0.

4.1 Program Memory Organization

The PIC16C84 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C84 only the first 1K x 14 (0000-03FFh) are physically implemented. Accessing a location above the physically implemented address will cause a wrap-around. For example, locations 20h, 420h, 820h, C20h, 1020h, 1420h, 1820h, and 1C20h will be the same instruction.

The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK



© 1995 Microchip Technology Inc.

4.2 Data Memory Organization

The data memory is partitioned into two areas. The first is the Special Function Registers (SFR) area, while the second is the General Purpose Registers (GPR) area. The SFRs control the operation of the device.

Portions of data memory are banked. This is for both the SFR area and the GPR area. The GPR area is banked to allow greater than 96 bytes of general purpose RAM. The banked areas of the SFR are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the STATUS Register. Figure 4-2 shows the data memory map organization.

Instructions `MOVWF` and `MOVF` can move values from the W register to any location in the register file ("F"), and vice-versa.

The entire data memory can be accessed either directly or indirectly through the File Select Register (FSR) (Section 4.4). Indirect addressing uses the present value of the RP1:RP0 bits for access into the banked areas of data memory.

Data memory is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected by clearing the RP0 bit (STATUS<5>). Setting the RP0 bit selects Bank 1. Each Bank extends up to 7Fh (128 bytes). The lower locations of each Bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers implemented as static RAM. (Figure 4-2)

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly, or indirectly through the FSR (Section 4.4).

All devices have some amount of GPR area. The GPR is 8-bits wide. When the GPR area is greater then 96, banking must be performed to access the additional memory space.

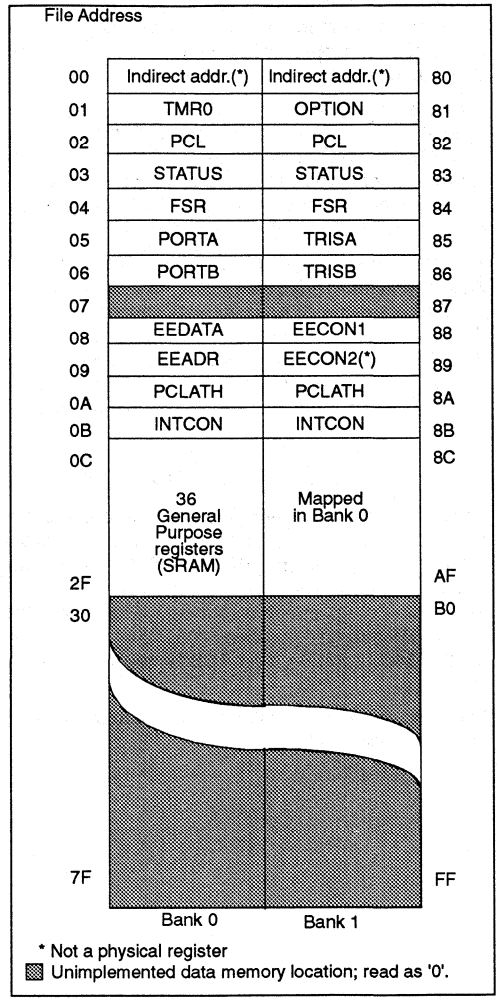
PIC16C84 devices do not have banked memory in the GPR area. Any access to Bank 1 will cause the access to occur in Bank 0. That is, the MSb of the direct address will be ignored.

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are used by the CPU and Peripheral functions to control the device operation. (Figure 4-2 and Table 4-1). These registers are static RAM.

The special registers can be classified into two sets. Those associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for that specific feature.

FIGURE 4-2: PIC16C84 REGISTER FILE MAP



© 1995 Microchip Technology Inc.

TABLE 4-1: REGISTER FILE SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note3)
Bank 0											
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of PC								0000 0000	0000 0000
03h	STATUS [2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h										---- ----	---- ----
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC [1]					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 0000
Bank 1											
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
81h	OPTION	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Low order 8 bits of PC								0000 0000	0000 0000
83h	STATUS [2]	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000? ?uuu
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	PORTA data direction register					---1 1111	---1 1111
86h	TRISB	PORTB data direction register								1111 1111	1111 1111
87h										---- ----	---- ----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 ?000
89h	EECON2	EEPROM control register 2 (not a physical register)								---- ----	---- ----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC [1]					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', ? = Value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> is never transferred to PCLATH.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in STATUS are not affected by a $\overline{MCLR}$ reset.

3: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.

© 1995 Microchip Technology Inc.

4.2.2.1 STATUS REGISTER

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bit for data memory.

As with any register, the STATUS register can be the destination for any instruction. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

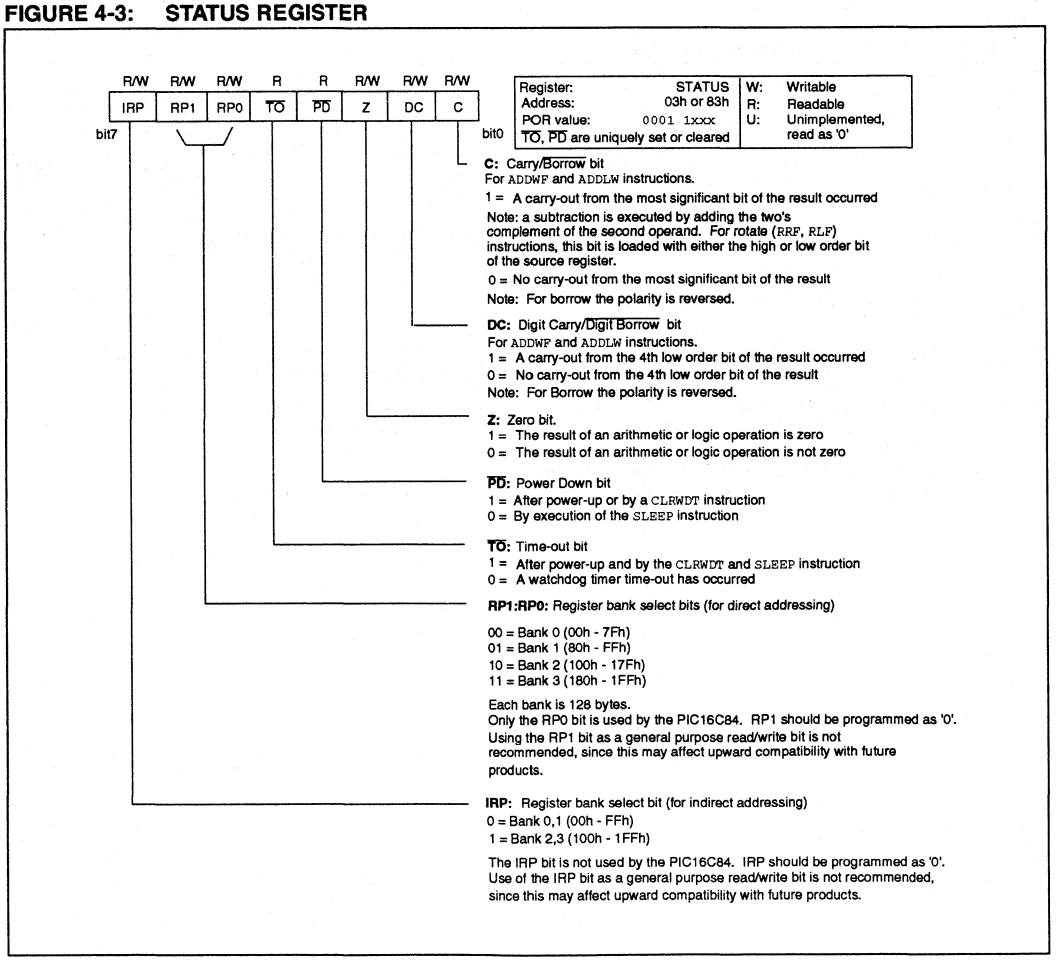
For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where `u` = unchanged).

Only the `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions should be used to alter the STATUS register because these instructions do not affect any status bit. (Table 9-2 Instruction Set Summary)

Note 1: The IRP and RP1 bits (STATUS<7,6>) are not used by the PIC16C84 and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

FIGURE 4-3: STATUS REGISTER



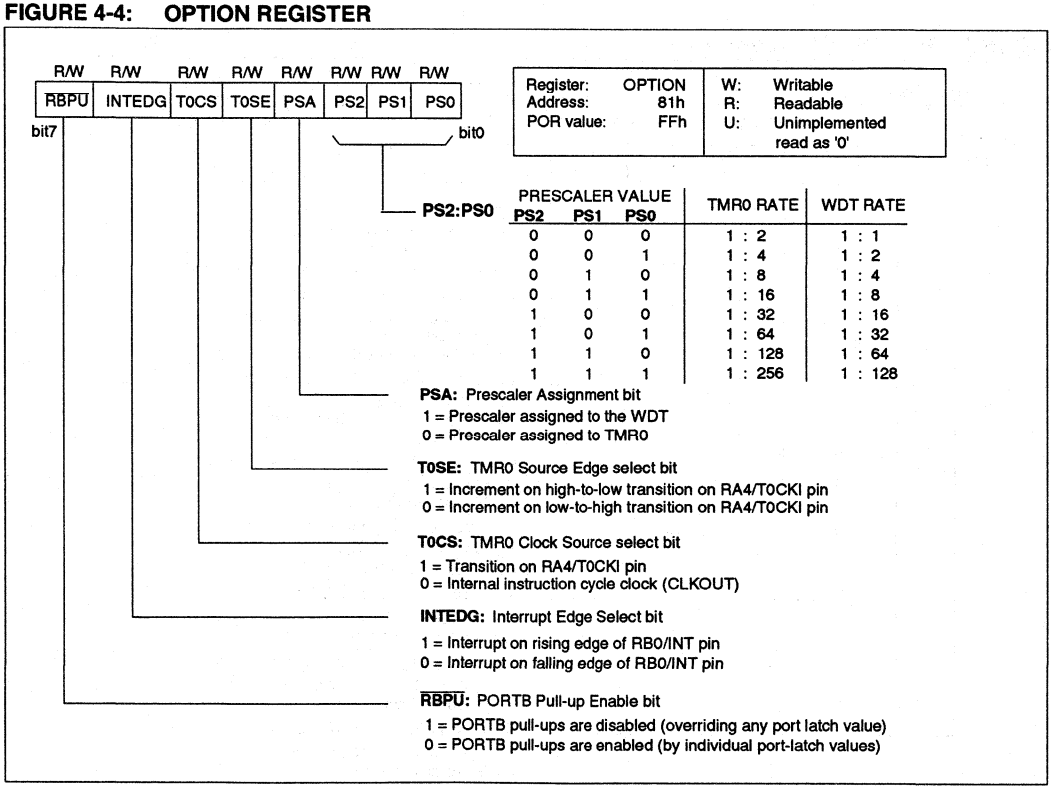
© 1995 Microchip Technology Inc.

4.2.2.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT interrupt, TMR0, and the weak pull-ups on PORTB.

Note: When the prescaler is assigned to the WDT (PSA = '1'), TMR0 has a 1:1 prescaler assignment.

FIGURE 4-4: OPTION REGISTER



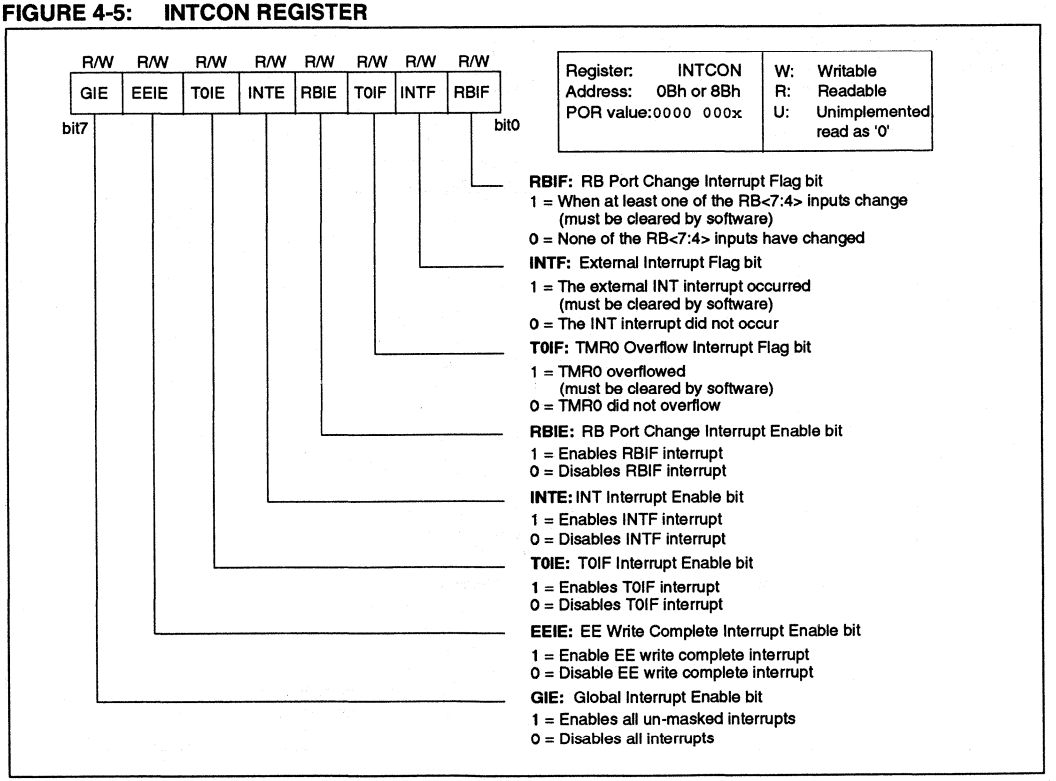
© 1995 Microchip Technology Inc.

4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains the various enable bits for all interrupt sources.

Note: T0IF, INTF, or RBIF will be set by the specified condition even if the corresponding interrupt enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled).

FIGURE 4-5: INTCON REGISTER

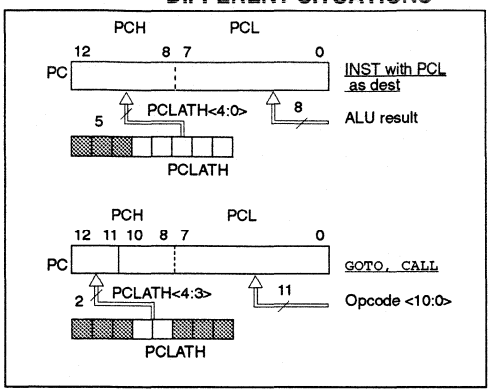


© 1995 Microchip Technology Inc.

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte, PCL, is a readable and writable register. The high byte of the PC (PCH) is not directly readable nor writable. PCLATH (PC latch high) is a holding register for PC<12:8> where contents are transferred to the upper byte of the program counter. When the PC is loaded with a new value during a CALL, GOTO or a write to PCL, the high bits of PC are loaded from PCLATH as shown in Figure 4-6.

FIGURE 4-6: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note "Table Read Using the PIC16CXX" (AN556).

4.3.2 STACK

The PIC16C84 has an 8 deep x 13-bit wide hardware stack (Figure 4-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The entire 13-bit PC is PUSH'ed onto the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is POP'ed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or a POP operation.

The stack operates as a circular buffer. That is, after the stack has been PUSH'ed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

If the stack is effectively POP'ed nine times, the PC value is the same as the value from the first POP.

Note 1: There are no status bits to indicate stack overflow or stack underflow conditions.

Note 2: There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address

4.3.3 PROGRAM MEMORY PAGING

The PIC16C84 has 1K of program memory. The CALL and GOTO instructions have an 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size.

For future PIC16C8X program memory expansion, there must be another two bits to specify the program memory page. These paging bits come from the PCLATH<4:3> bits (Figure 4-6). When doing a CALL or a GOTO instruction, the user must ensure that these page bits (PCLATH<4:3>) are programmed to the desired program memory page. If a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<4:3> is not required for the return instructions (which POPs the PC from the stack).

Note: The PIC16C84 ignores the PCLATH<4:3> bits, which are used for program memory pages 1, 2 and 3 (0800h - 1FFFh). The use of PCLATH<4:3> as general purpose R/W bits is not recommended since this may affect upward compatibility with future products.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register and is used in conjunction with the FSR register to perform indirect addressing.

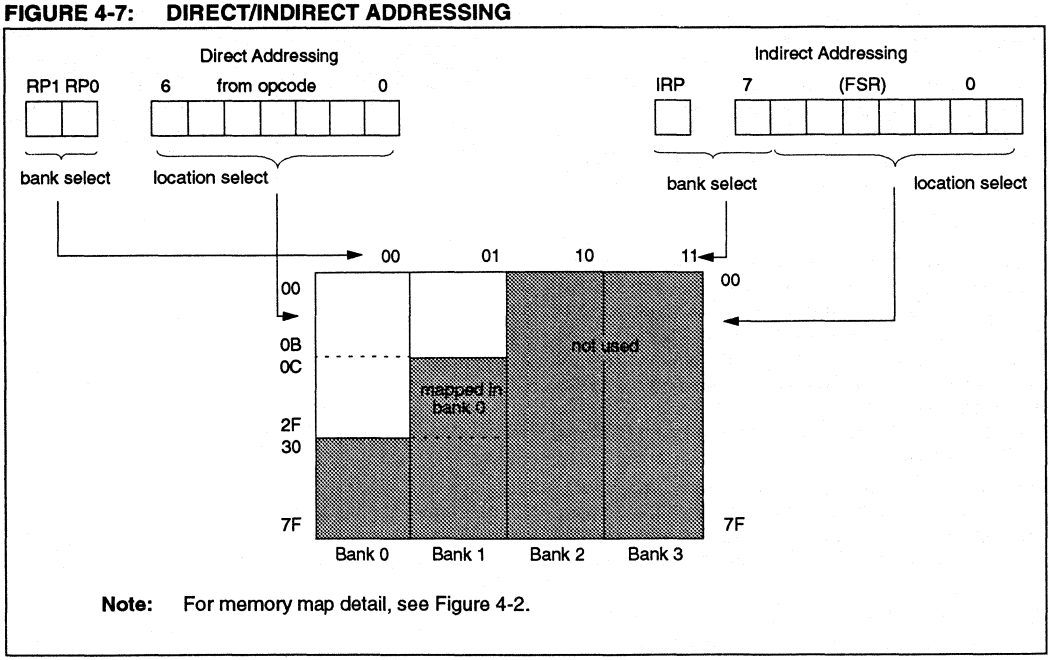
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16C84.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
        movlw   0x20    ; initialize pointer
        movf    FSR     ;   to RAM
NEXT    clrf    INDF    ; clear INDF register
        incf    FSR     ; inc pointer
        btfss   FSR,4   ; all done?
        goto    NEXT    ; NO, clear next
CONTINUE :              ; YES, continue
```

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail, see Figure 4-2.

© 1995 Microchip Technology Inc.

5.0 I/O PORTS

The PIC16C84 device has two ports, PORTA and PORTB. Some port pins are multiplexed with an alternate function for other features on the device.

5.1 PORTA and TRISA Registers

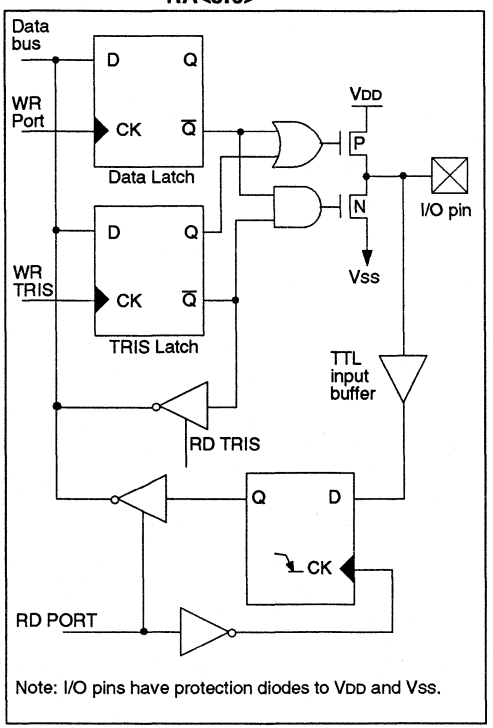
PORTA is a 5-bit wide latch. RA4 is a Schmitt trigger input and an open collector output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

A '1' on any bit in the TRISA register puts the corresponding output driver in a hi-impedance mode. A '0' on any bit in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The RA4 pin is multiplexed with the TMR0 clock input.

FIGURE 5-1: BLOCK DIAGRAM OF RA<3:0>

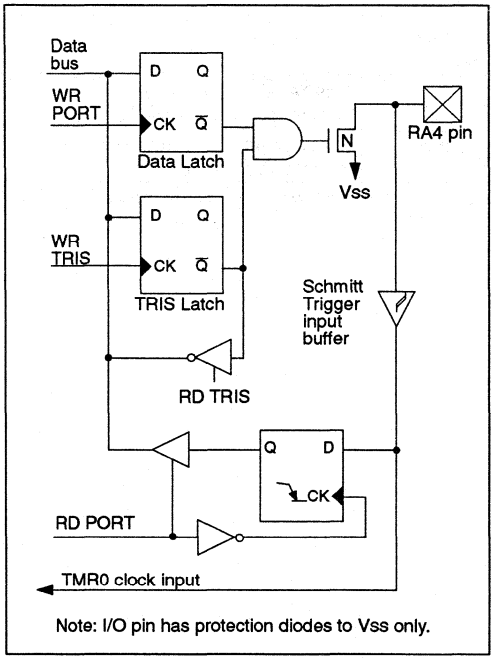


Note: I/O pins have protection diodes to VDD and Vss.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF  PORTA       ;Initialize PORTA by setting
                  ; output data latches
BSF   STATUS, RP0 ;Select Bank1
MOVLW 0xCF        ;Value used to initialize
                  ;data direction
MOVWF TRISA       ;Set RA<3:0> as inputs
                  ;RA<5:4> as outputs
                  ;TRISA<7:6> are always
                  ;read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN



Note: I/O pin has protection diodes to Vss only.

Note: For crystal oscillator configurations operating below 500 kHz, the device may generate a spurious internal Q-clock when PORTA<0> switches state. This does not occur with an external clock in RC mode. To avoid this, the RA0 pin should be kept static, i.e. in input/output mode, RA0 should not be toggled.

TABLE 5-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open collector type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Register Name	Function	Address	Power-on Reset Value
PORTA	PORTA pins when read PORTA data latch when written	05h	---x xxxx
TRISA	PORTA data direction register 0 = output, 1 = input	85h	---1 1111

Legend: x = unknown, - = unimplemented, read as '0'.

© 1995 Microchip Technology Inc.

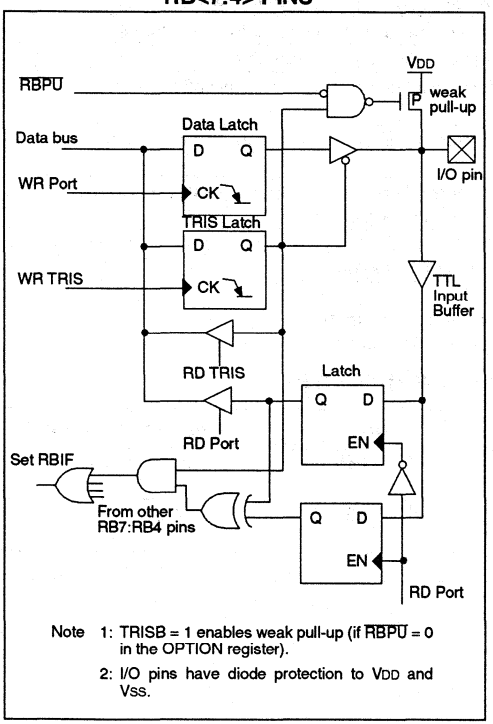
5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. A '1' on any bit in the TRISB register puts the corresponding output driver in a high impedance mode. A '0' on any bit in the TRISB register puts the contents of the output latch on the selected pin(s).

Each of the PORTB pins have a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on POR.

Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The pins value in input mode are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of the pins are OR'ed together to generate the RBIF interrupt (INTCON<0>).

FIGURE 5-3: BLOCK DIAGRAM OF RB<7:4> PINS



Note 1: TRISB = 1 enables weak pull-up (if $\overline{\text{RBPU}}$ = 0 in the OPTION register).
2: I/O pins have diode protection to VDD and VSS.

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

1. Disable the interrupt by clearing the RBIE (INTCON<3>) bit.
2. Read PORTB, then clear the RBIF bit.

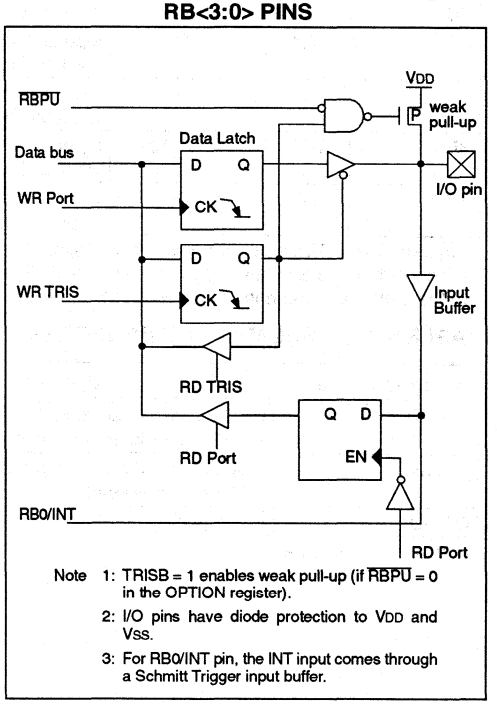
A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the *Embedded Control Handbook*).

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), the RBIF interrupt flag may not be set.

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-4: BLOCK DIAGRAM OF RB<3:0> PINS



Note 1: TRISB = 1 enables weak pull-up (if $\overline{\text{RBPU}}$ = 0 in the OPTION register).
2: I/O pins have diode protection to VDD and VSS.
3: For RB0/INT pin, the INT input comes through a Schmitt Trigger input buffer.

EXAMPLE 5-2: INITIALIZING PORTB

```
CLRF  PORTB       ;Initialize PORTB by setting
                  ; output data latches
BSF   STATUS, RP0 ;Select Bank1
MOVLW 0xCF        ;Value used to initialize
                  ;data direction
MOVWF TRISB       ;Set RB<3:0> as inputs
                  ;RB<5:4> as outputs
                  ;TRISB<7:6> are always
                  ;read as '0'.
```

TABLE 5-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	I/O Consistancy Function
RB0/INT	bit0	TTL	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST‡	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger.
† This buffer is a Schmitt Trigger input when configured as the external interrupt.
‡ This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Register Name	Function	Address	Power-on Reset Value
PORTB	PORTB pins when read PORTB data latch when written	06h	xxxx xxxx
TRISB	PORTB data direction register 0 = output, 1 = input	86h	1111 1111
OPTION	Weak pull-up on/off control ($\overline{RBPU}$ bit)	81h	1111 1111

Legend: x = unknown.

© 1995 Microchip Technology Inc.

5.3 I/O Programming Considerations

5.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and writes the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (i.e., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch is unknown.

Reading the PORT[A,B] register reads the values of the PORT[A,B] pins. Writing to the PORT[A,B] register writes the value to the PORT[A,B] latch. When using read modify write instructions (i.e. BCF, BSF, etc.) on a port, the value of the PORT[A,B] pins is read, the desired operation is done to this value, and this value is then written to the PORT[A,B] latch.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

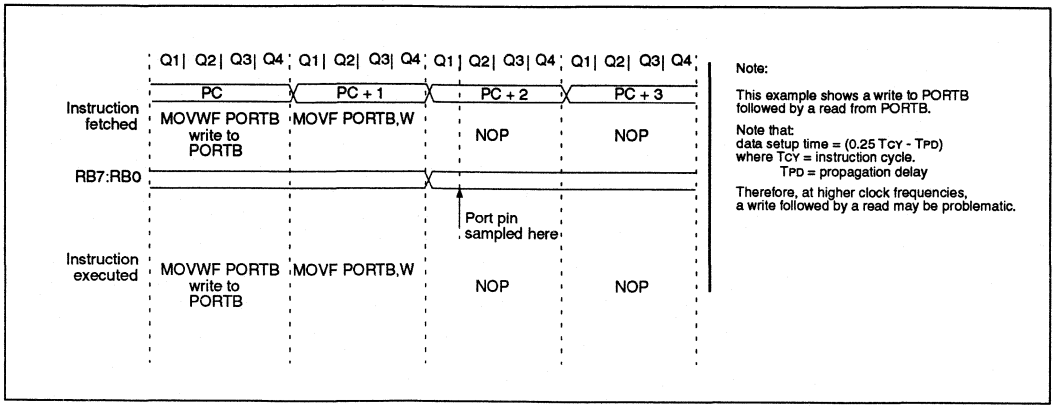
The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-5). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such that the pin voltage stabilizes (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

Example 5-3 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O port.

EXAMPLE 5-3: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
; Initial PORT settings: PORTB<7:4> Inputs
;                        PORTB<3:0> Outputs
; PORTB<7:6> have external pull-ups and are not
; connected to other circuitry
;
;                        PORT latch   PORT pins
;                        ----------   ----------

    BCF PORTB, 7        ; 01pp pppp   11pp pppp
    BCF PORTB, 6        ; 10pp pppp   11pp pppp
    BCF STATUS, RP0     ;
    BCF TRISB, 7        ; 10pp pppp   11pp pppp
    BCF TRISB, 6        ; 10pp pppp   10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

FIGURE 5-5: SUCCESSIVE I/O OPERATION



NOTES:

© 1995 Microchip Technology Inc.

6.0 TIMER0 (TMR0) MODULE

The TMR0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the TMR0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 module will increment every instruction cycle (without prescaler). If TMR0 is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 module.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode TMR0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the T0 source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit (OPTION<4>) selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

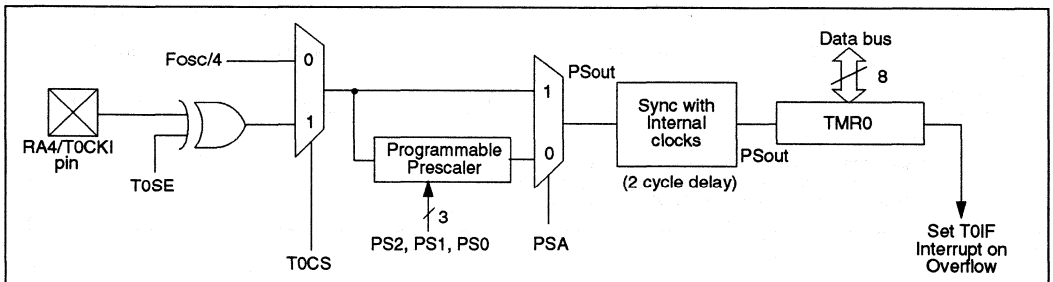
The prescaler is shared between the TMR0 module and the Watchdog Timer. The prescaler assignment is controlled, in software, by control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to TMR0. The prescaler is not readable nor writable. When the prescaler is assigned to the TMR0 module, the prescale value (1:2, 1:4, ..., 1:256) is software selectable. Section 6.3 details the operation of the prescaler.

6.1 TIMER0 (TMR0) Interrupt

TMR0 interrupt is generated when the TMR0 module timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit (INTCON<2>). The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit must be cleared in software by the TMR0 module interrupt service routine before re-enabling this interrupt. The TMR0 module interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. Figure 6-4 shows the TMR0 interrupt timing.

2

FIGURE 6-1: TMR0 BLOCK DIAGRAM



Note 1: Bits T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.
2: The prescaler is shared with the Watchdog Timer (Figure 6-6)

FIGURE 6-2: TMR0 TIMING: INTERNAL CLOCK/NO PRESCALE

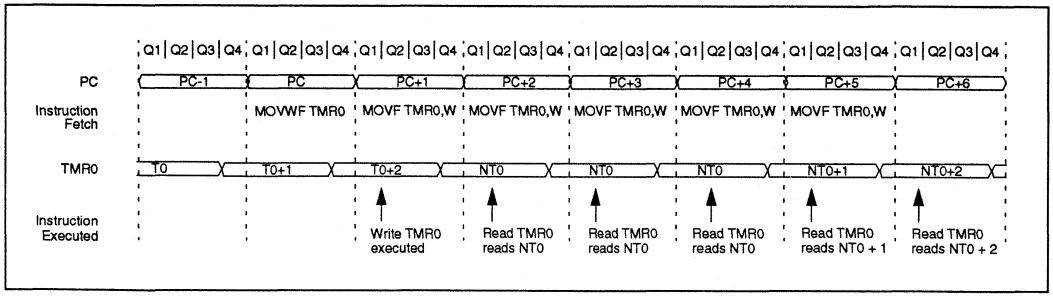


FIGURE 6-3: TMR0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

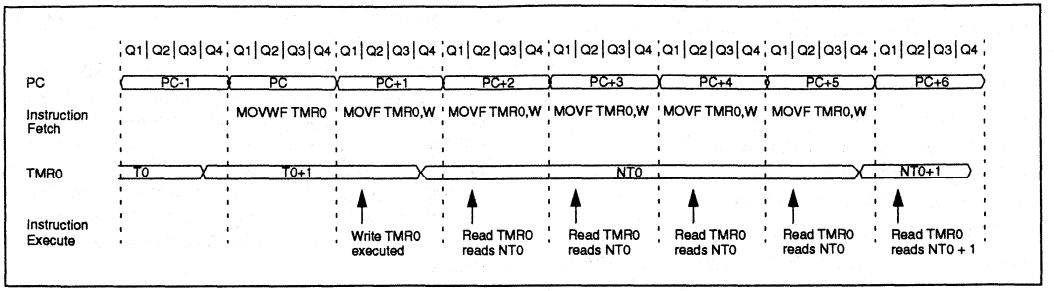
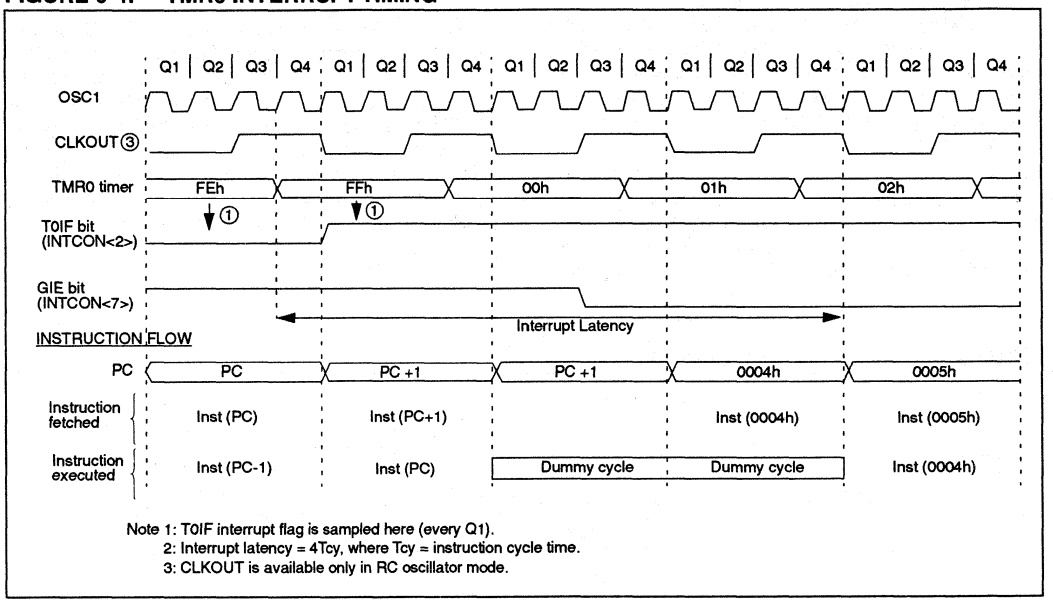


FIGURE 6-4: TMR0 INTERRUPT TIMING



Note 1: T0IF interrupt flag is sampled here (every Q1).
2: Interrupt latency = 4Tcy, where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

© 1995 Microchip Technology Inc.

6.2 Using TMR0 with External Clock

When an external clock input is used for TMR0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of TMR0 after synchronization.

6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2 Tosc (plus a small RC delay) and low for at least 2 Tosc (plus a small RC delay). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by an asynchronous ripple counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4 Tosc (plus a small RC delay) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the AC Electrical Specifications of the desired device.

6.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

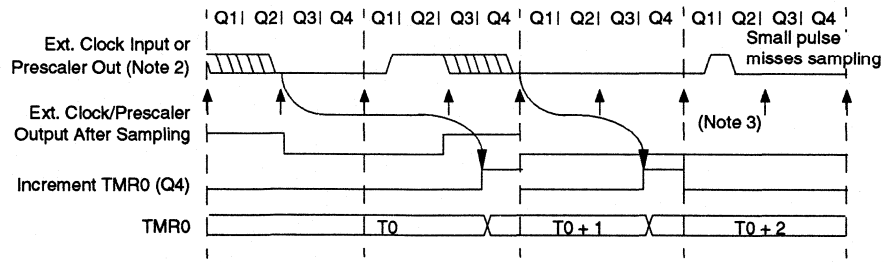
6.3 Prescaler

An 8-bit counter is available as a prescaler for the TMR0 module, or as a post-scaler for the Watchdog Timer (Figure 6-6). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the TMR0 module and the Watchdog Timer. Thus, a prescaler assignment for the TMR0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the TMR0 module, all instructions writing to the TMR0 module (e.g. `CLRF 1`, `MOVWF 1`, `BSF 1,x` ....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable nor writable.

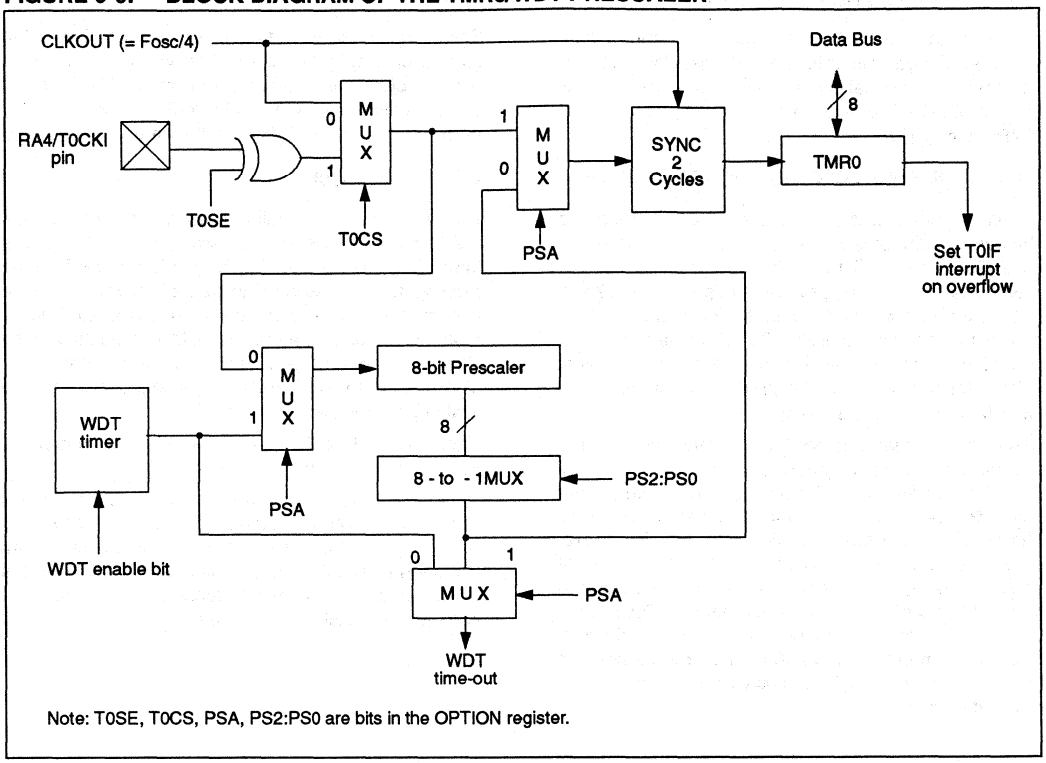
FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: Delay from clock input change to TMR0 increment is 3 Tosc to 7 Tosc. (Duration of Q = tosc). Therefore, the error in measuring the interval between two edges on TMR0 input = ± 4 Tosc max.
2: External clock if no prescaler selected, Prescaler output otherwise.
3: The arrows ↑ indicate where sampling occurs.

© 1995 Microchip Technology Inc.

FIGURE 6-6: BLOCK DIAGRAM OF THE TMR0/WDT PRESCALER



Note: T0SE, T0CS, PSA, PS2:PS0 are bits in the OPTION register.

© 1995 Microchip Technology Inc.

6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution. To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from TMR0 to the WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TMR0→WDT)

```
BCF     STATUS, RP0   ; Bank 0
CLRF    TMR0          ; Clear TMR0 & Prescaler
BSF     STATUS, RP0   ; Bank 1
CLRWDT                ; Clears WDT
MOVLW   'xxxx1xxx'b   ; Select new prescaler
MOVWF   OPTION        ; value
BCF     STATUS, RP0   ; Bank 0
```

To change prescaler from the WDT to the TMR0 module use the sequence shown in Example 6-2. This sequence must be taken even if the WDT is disabled.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TMR0)

```
CLRWDT                ; Clear WDT and
                      ;   prescaler
BSF     STATUS, RP0   ;
MOVLW   'xxxx0xxx'b   ; Select TMR0, new
                      ;   prescale value
                      ;   and clock source
MOVWF   OPTION        ;
BCF     STATUS, RP0   ;
```

2

TABLE 6-1: SUMMARY OF TMR0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0. (Figure 6-5)	81h	1111 1111
INTCON	TMR0 overflow interrupt flag and mask bits. (Figure 6-6)	0Bh	0000 000x

Legend: x = unknown.
Note: For reset values of registers in other reset situations refer to the Special Features of the CPU section.

TABLE 6-2: REGISTERS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	TMR0	Timer0							
0Bh/8Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
85h	TRISA	—	—	—	TRISA 4	TRISA 3	TRISA 2	TRISA 1	TRISA 0

Legend: — = Unimplemented locations, read as '0'.
Note 1: Shaded cells are not used by TMR0 module.
2: The PIC16C84 device does not have an RA5 pin.

© 1995 Microchip Technology Inc.

Notes:

© 1995 Microchip Technology Inc.

7.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (full VDD range). This memory is not directly mapped in the register file space. Instead it is indirectly addressed through the Special Function Registers. There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write, and EEADR holds the address of the EEPROM location being accessed. PIC16C84 devices have 64 bytes of data EEPROM with an address range from 0h to 3Fh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is nominally 10 ms, and is controlled by an on-chip timer. The actual write-time will vary with voltage and temperature as well as from chip to chip. Please refer to AC specifications for exact limits.

When the device is code protected, only the CPU may complete reads or writes of the data memory. That is, the device programmer can no longer access this memory (external reads/writes are disabled).

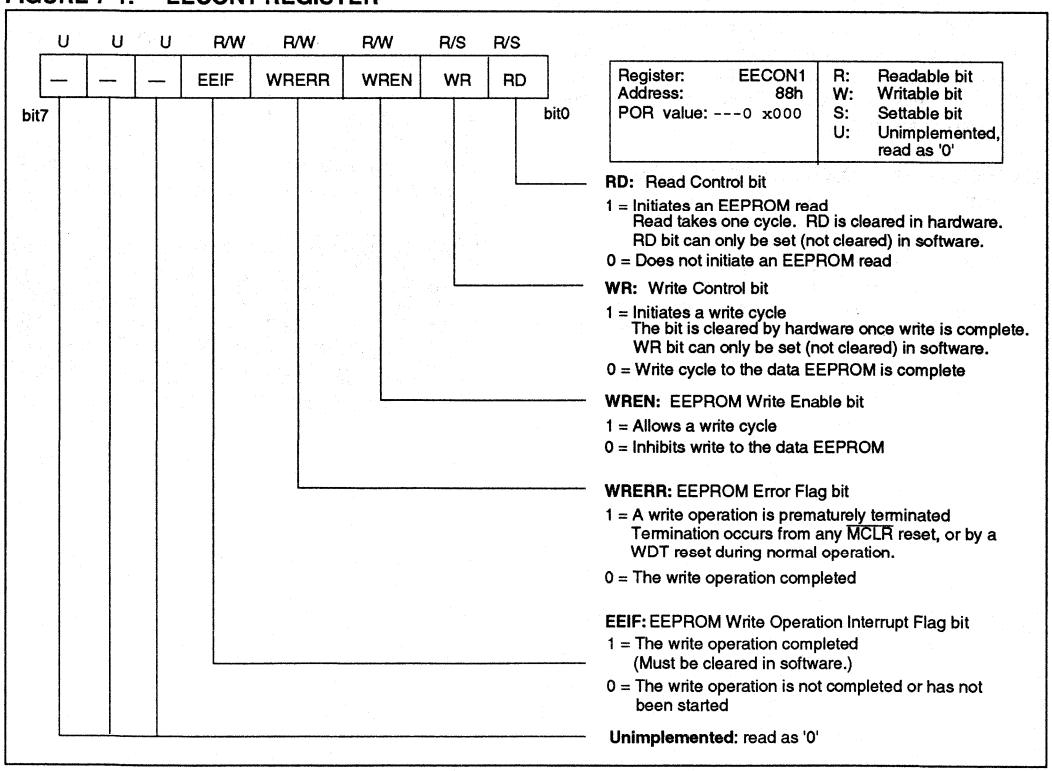
7.1 EEADR

The EEADR register can address up to a maximum of 256 bytes of data EEPROM. Only the first 64 bytes of data EEPROM are implemented and only six of the eight bits in the register (EEADR<5:0>) are required.

The upper two bits are not address decoded. This allows four addresses to map into the 64 byte memory space. We recommend using absolute address (0 - 3Fh) to ensure future upward compatibility.

2

FIGURE 7-1: EECON1 REGISTER



© 1995 Microchip Technology Inc.

7.2 EECON1 and EECON2 Registers

EECON1 is the control register with five low order bits physically implemented. The upper-three bits are non-existent and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of read or write operation. Inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up WREN is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

The EEIF interrupt flag bit is set when write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's.

7.3 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 7-1: DATA EEPROM READ

```
BCF     STATUS, RP0  ; Bank 0
MOVLW   CONFIG_ADDR  ;
MOVWF   EEADR        ; Address to read
BSF     STATUS, RP0  ; Bank 1
BSF     EECON, RD    ; EE Read
BCF     STATUS, RP0  ; Bank 0
MOVF    EEDATA, W    ; W = EEDATA
```

7.4 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate write.

EXAMPLE 7-2: DATA EEPROM WRITE

```
BSF     STATUS, RP0  ; Bank 1
BCF     INTCON, GIE  ; Disable INTs.
MOVLW   55h          ;
MOVWF   EECON2       ; Write 55h
MOVLW   AAh          ;
MOVWF   EECON2       ; Write AAh
BSF     EECON1,WR    ; Set WR bit
                     ;   begin write
BSF     INTCON, GIE  ; Enable INTs.
```

Write will not initiate if this sequence (write 55h to EECON2, write AAh to EECON2, then set WR bit) is not followed with exact timing. We strongly recommend that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set until the WREN bit has been set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

Note: The data EEPROM memory E/W cycle time may occasionally exceed the 10 ms specification (typical). To ensure that the write cycle is complete, use the EE interrupt or poll the WR bit (EECON1<1>). Both these events signify the completion of the write cycle.

© 1995 Microchip Technology Inc.

7.5 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the power-up timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

7.6 Power Consumption Considerations

Note: It is recommended that the EEADR<7:6> bits be cleared. When either of these bits is set, the maximum IDD for the device is higher than when both are cleared. The specification is 400 µA. With EEADR<7:6> cleared, the maximum is approximately 150 µA.

TABLE 7-1: REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 ?000
89h	EECON2	EEPROM control register 2								---- ----	---- ----

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0', ? = Value depends upon condition.

© 1995 Microchip Technology Inc.

Notes:

© 1995 Microchip Technology Inc.

8.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC16C84 has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features are:

- OSC selection
- Reset
  - Power-On Reset (POR)
  - Power-Up Timer (PWRT)
  - Oscillator Start-Up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations
- In-circuit serial programming

The PIC16C84 has a Watchdog Timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-Up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the Power-Up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. This design keeps the device in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

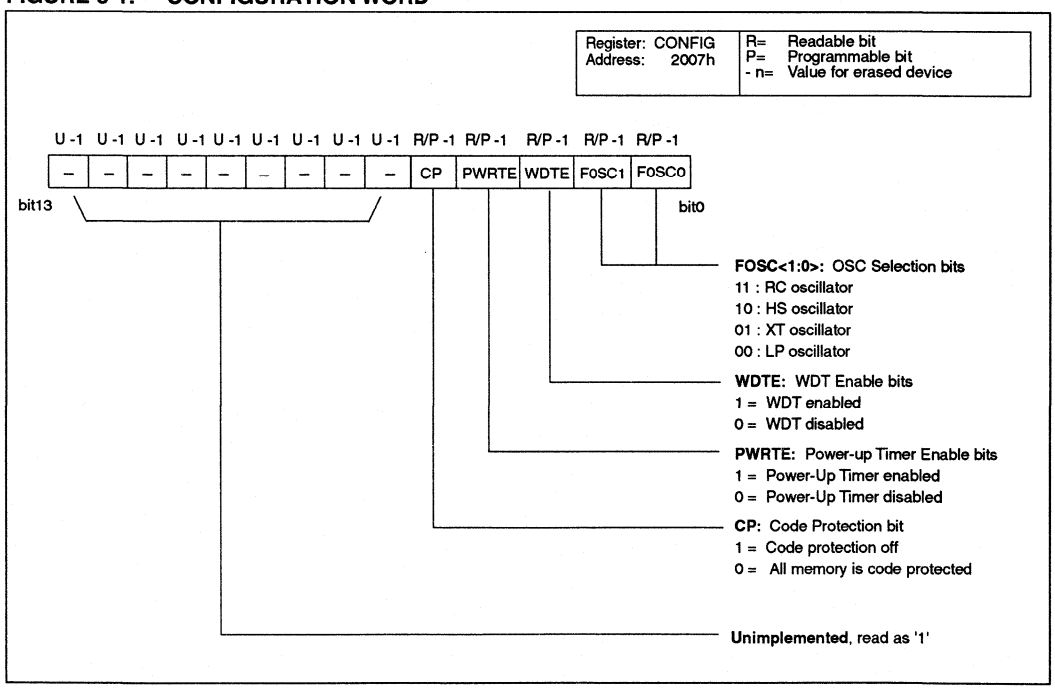
SLEEP mode offers a very low current power-down mode. The user can wake up from SLEEP through external reset, Watchdog Timer time-out or through an interrupt. Several oscillator options are provided to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select the various options.

8.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

Note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

FIGURE 8-1: CONFIGURATION WORD



8.2 Oscillator Configurations

8.2.1 OSCILLATOR TYPES

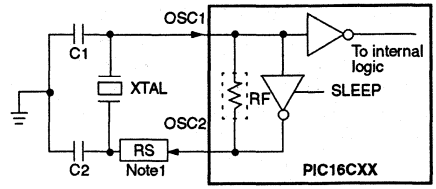
The PIC16C84 can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

8.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 8-2). The PIC16C84 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin. This is shown in Figure 8-3.

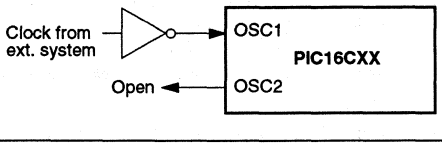
FIGURE 8-2: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



See Table 8-1 and Table 8-2 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

FIGURE 8-3: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)



© 1995 Microchip Technology Inc.

TABLE 8-1: PIC16C84 CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq**	**OSC1/C1**	**OSC2/C2**
XT	455 kHz 2.0 MHz 4.0 MHz	47 - 100 pF 15 - 68 pF 15 - 68 pF	47 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 10.0 MHz	15 - 68 pF 10 - 47 pF	15 - 68 pF 10 - 47 pF

Note: Recommended values of C1 and C2 are identical to the ranges tested table.

Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for the appropriate values of external components.

Resonators Tested:		
455 kHz	Panasonic EFO-A455K04B	± 0.3%
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%
10.0 MHz	Murata Erie CSA10.00MTZ	± 0.5%

None of the resonators had built-in capacitors.

TABLE 8-2: PIC16C84 CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Mode	Freq	OSC1/C1	OSC2/C2
LP	32 kHz 200 kHz	68 - 100 pF 15 - 30 pF	68 - 100 pF 15 - 30 pF
XT	100 kHz 2 MHz 4 MHz	68 - 150 pF 15 - 33 pF 15 - 33 pF	150 - 200 pF 15 - 33 pF 15 - 33 pF
HS	4 MHz 10 MHz	15 - 33 pF 15 - 47 pF	15 - 33 pF 15 - 47 pF

Note: Higher capacitance increases the stability of oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

For $V_{DD} > 4.5V$, C1 = C2 ≈ 30 pF is recommended.

Crystals Tested:		
32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
100 kHz	Epson C-2 100.00 KC-P	± 20 PPM
200 kHz	STD XTL 200.000 KHz	± 20 PPM
1.0 MHz	ECS ECS-10-13-2	± 50 PPM
2.0 MHz	ECS ECS-20-S-2	± 50 PPM
4.0 MHz	ECS ECS-40-S-4	± 50 PPM
10.0 MHz	ECS ECS-100-S-4	± 50 PPM

© 1995 Microchip Technology Inc.

8.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits are available; one with series resonance, or one with parallel resonance.

Figure 8-4 shows a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 8-4: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

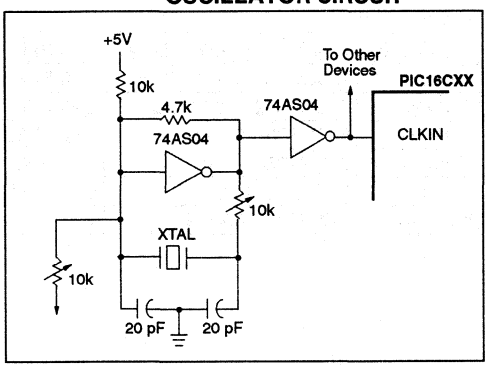
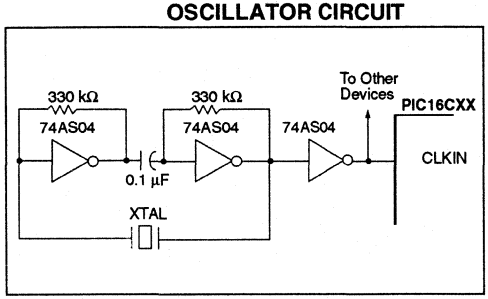


Figure 8-5 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 8-5: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



8.2.4 RC OSCILLATOR

For timing insensitive applications the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) values, capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types also affects the oscillation frequency, especially for low Cext values. The user needs to take into account variation due to tolerance of the external R and C components. Figure 8-6 shows how an R/C combination is connected to the PIC16C84. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping Rext between 3 kΩ and 100 kΩ.

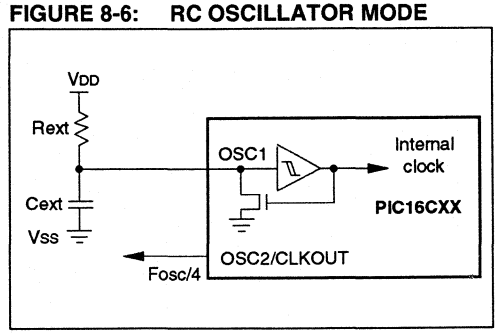
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See the electrical specification section for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance has a greater affect on RC frequency).

See the electrical specification section for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-2 for waveform).

FIGURE 8-6: RC OSCILLATOR MODE



© 1995 Microchip Technology Inc.

8.3 Reset

The PIC16C84 differentiates between various kinds of reset:

- Power-On Reset (POR)
- $\overline{MCLR}$ reset during normal operation
- $\overline{MCLR}$ reset during SLEEP
- WDT time-out reset during normal operation
- WDT time-out reset during SLEEP

Some registers are not affected in any reset condition; their status is unknown on POR reset and unchanged in any other reset. Most other registers are reset to a "reset state" on POR, $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 8-4. These bits are used in software to determine the nature of the reset. Table 8-6 gives a full description of reset states for all registers.

Figure 8-7 shows a simplified block diagram of the on-chip reset circuit.

8.4 Power-On Reset (POR), Power-Up-Timer (PWRT) and Oscillator Start-Up Timer (OST)

8.4.1 POWER-ON RESET (POR)

A Power-On Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.6V - 1.8V). To take advantage of the POR, just tie the $\overline{MCLR}$ pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-On Reset. A maximum rise time for VDD is required for this to operate properly. See Electrical Specifications for details.

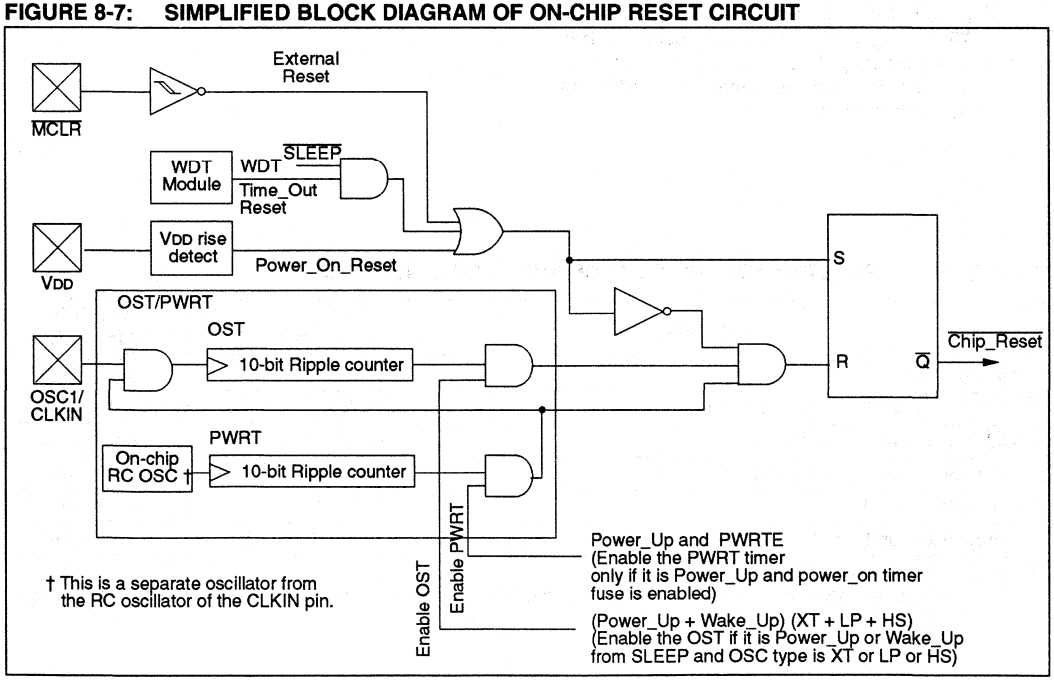
The POR circuit does not produce an internal reset when VDD declines.

8.4.2 POWER-UP TIMER (PWRT)

The Power-Up Timer provides a fixed 72 ms nominal time-out on power-up only, from POR. The power-up timer operates on an internal RC oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration fuse, PWRTE, can enable (if set) or disable (if cleared or programmed) the power-up timer.

The Power-Up Time delay will vary from chip to chip due to VDD, temperature, and process variation. See DC parameters for details.

FIGURE 8-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



8.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle delay (from OSC1 input) after the PWRT delay ends. This ensures the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on power-on reset or wake-up from SLEEP.

8.4.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after a POR has expired. Then the OST is activated. The total time-out will vary based on oscillator configuration and PWRTE fuse status. For example, in RC mode with the PWRTE fuse cleared (PWRT disabled), there will be no time-out at all. Figure 8-8, Figure 8-9, and Figure 8-10 depict time-out sequences on power-up.

TABLE 8-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Wake up from SLEEP
	PWRTE = 1	PWRTE = 0	
XT, HS, LP	72 ms + 1024 Tosc	1024 Tosc	1024 tosc
RC	72 ms	—	—

Since the time-outs occur from the POR reset pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin immediately (see Figure 8-9). This is useful for testing purposes or to synchronize more than one PIC16CXX device when operating in parallel.

Table 8-4 shows the significance of the $\overline{TO}$ and $\overline{PD}$ bits. Table 8-5 lists the reset conditions for some special registers, while Table 8-6 lists the reset conditions for all the registers.

TABLE 8-4: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{TO}$	$\overline{PD}$	Condition
1	1	Power-On Reset
0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
0	1	WDT reset during normal operation
0	0	WDT timeout wakeup from SLEEP
1	1	$\overline{MCLR}$ reset during normal operation
1	0	$\overline{MCLR}$ reset during SLEEP or interrupt wake-up from SLEEP

TABLE 8-5: RESET CONDITION FOR PCL AND STATUS REGISTERS

Condition	PCL Addr: 02h	STATUS Addr: 03h/83h
Power-On Reset	000h	0001 1xxx
$\overline{MCLR}$ reset during normal operation	000h	0001 1uuu
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu
WDT reset during normal operation	000h	0000 1uuu
WDT during SLEEP	PC + 1	uuu0 0uuu
Interrupt wake-up from SLEEP	PC + 1 (Note1)	uuu1 0uuu

Legend: u = unchanged, x = unknown.

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

© 1995 Microchip Technology Inc.

TABLE 8-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Address	Power-On Reset	MCLR Reset during: – normal operation – SLEEP WDT timeout during normal operation	Wake up from SLEEP: – through interrupt – through WDT timeout
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1[2]
STATUS	03h	0001 1xxx	000? ?uuu[3]	uuu? ?uuu[3]
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---u uuuu	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu[1]
INDF	80h	---- ----	---- ----	---- ----
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC + 1
STATUS	83h	0001 1xxx	000? ?uuu[3]	uuu? ?uuu[3]
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 0000	---0 ?000	---0 ?uuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu[1]

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', ? = value depends on condition

Note 1: One or more bits in INTCON will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: Table 8-5 lists the reset value for each specific condition.

FIGURE 8-8: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

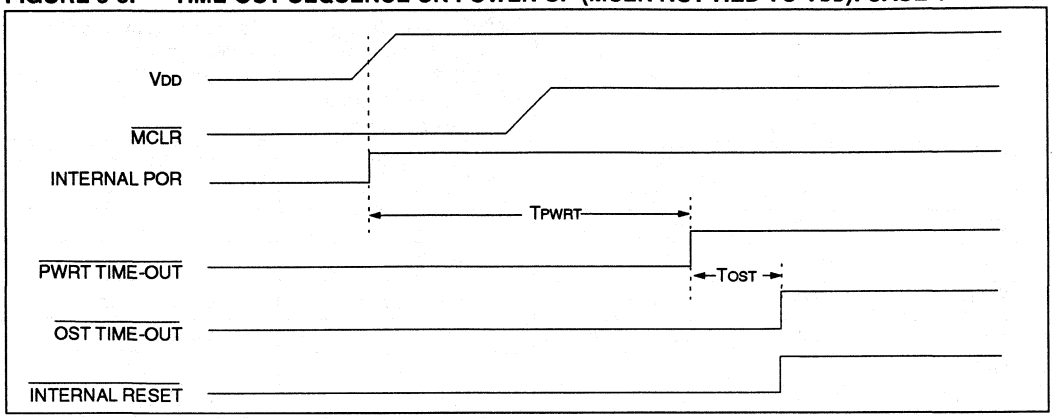


FIGURE 8-9: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

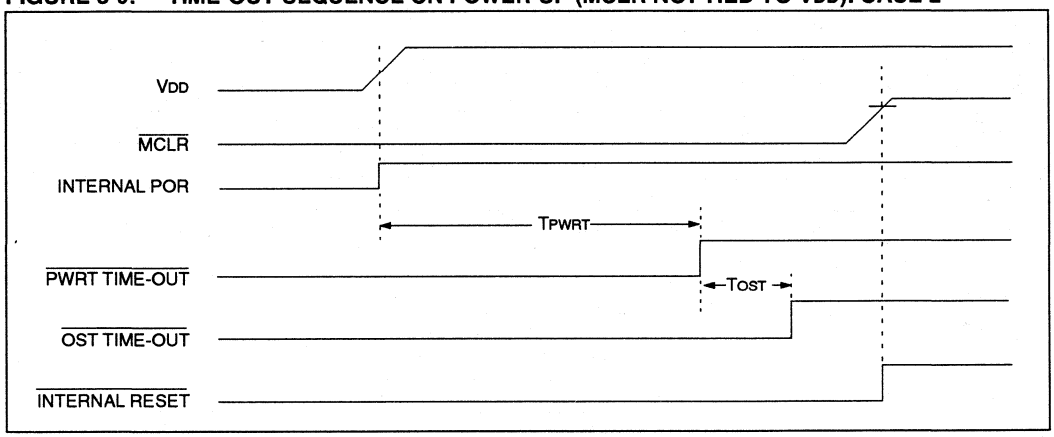
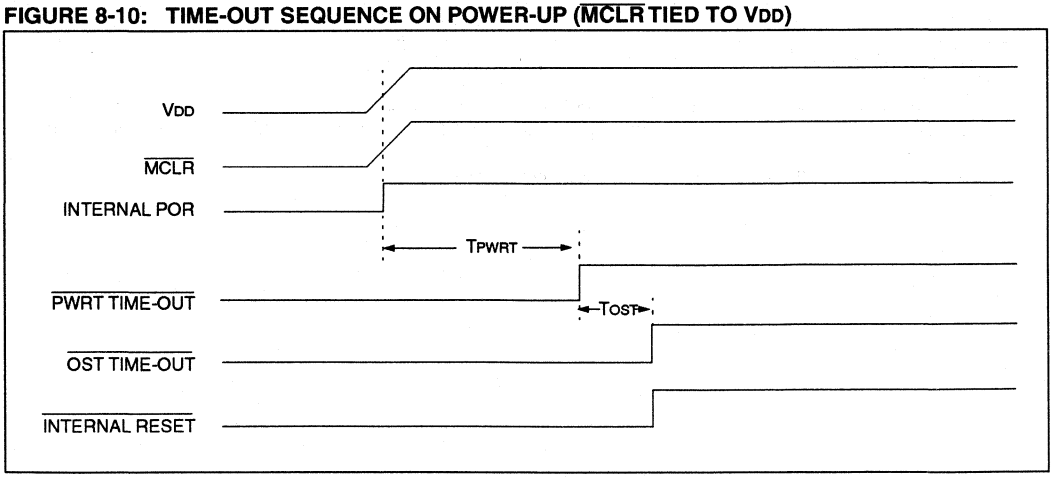
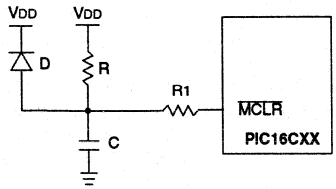


FIGURE 8-10: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)



© 1995 Microchip Technology Inc.

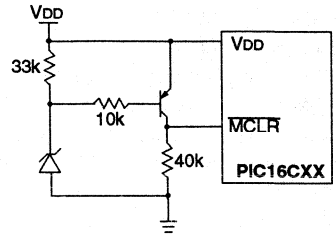
FIGURE 8-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note:

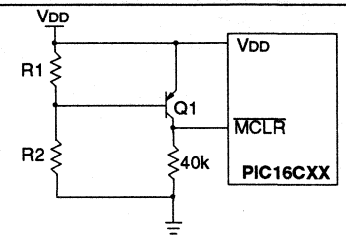
1. External Power-On Reset circuit is required only if VDD power-up rate is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
2. R < 40 kΩ is recommended to make sure that voltage drop across R does not exceed 0.2V (max leakage current spec on $\overline{MCLR}$ pin is 5 μA). A larger voltage drop will degrade VIH level on the $\overline{MCLR}$ pin.
3. R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C in the event of an $\overline{MCLR}$ pin breakdown due to ESD or EOS.

FIGURE 8-12: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 8-13: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V$$

8.5 Interrupts

The PIC16C84 family has up to 4 sources of interrupt:

- External interrupt RB0/INT pin
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB7:RB4)
- EEPROM write complete interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also contains the individual and global interrupt enable bits.

The global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The "return from interrupt" instruction, `RETFIE`, exits interrupt routine as well as sets the GIE bit, which re-enable interrupts.

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

When an interrupt is responded to; the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. For external interrupt events, such as the RB0/INT pin or PORTB change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 8-15). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

Note 2: If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the `RETFIE` instruction). The events that would cause this to occur are:

1. An instruction clears the GIE bit while an interrupt is acknowledged
2. The program branches to the interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the `RETFIE` instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GIE bit is cleared by the instruction, as shown in the following code:

```
LOOP    BCF   INTCON,GIE  ; Disable Global
                          ;   Interrupts
        BTFSC INTCON,GIE  ; Global Interrupts
                          ;   Disabled?
        GOTO  LOOP        ; NO, try again
                          ;   Yes, continue
                          ;   with program
                          ;   flow
```


© 1995 Microchip Technology Inc.

FIGURE 8-14: INTERRUPT LOGIC

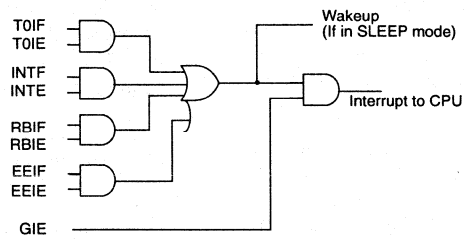
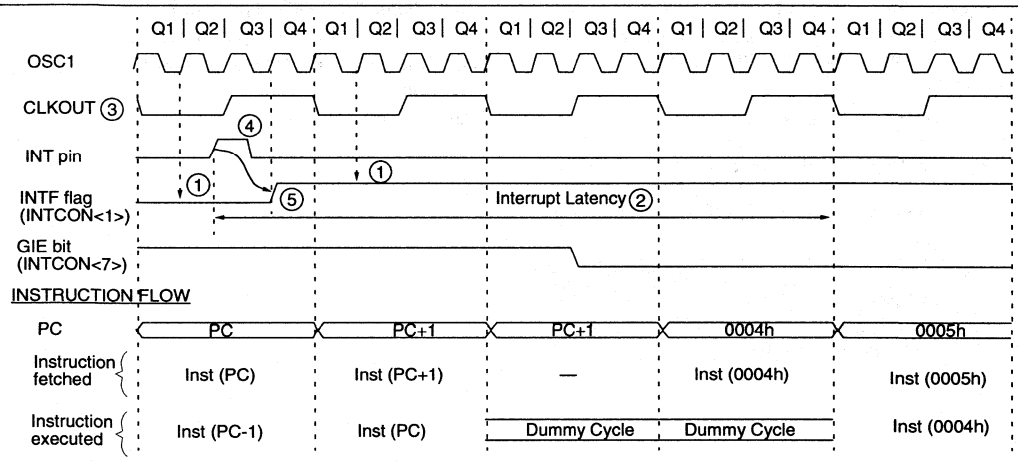


FIGURE 8-15: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4 Tcy where Tcy = instruction cycle time.
Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in RC oscillator mode.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

© 1995 Microchip Technology Inc.

8.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set . This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software via the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake the processor from SLEEP only if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether the processor branches to the interrupt vector following wake-up. Section 8.8 details SLEEP mode.

8.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in TMR0 will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit (Section 6.0).

8.5.3 PORT RB INTERRUPT

An input change on PORTB<7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<3>) bit (Section 5.2).

Note: If a change on an I/O pin should occur when a read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not be set.

8.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users wish to save key register values during an interrupt (e.g. W register and STATUS register). This is implemented in software.

Example 8-1 stores and restores the STATUS and W register's values. The register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1). User register, STATUS_TEMP, must be defined in bank 0.

Example 8-1 does the following:

a) Stores the W register.
b) Stores the STATUS register in bank 0.
c) Executes the Interrupt Service Routine code.
d) Restores the STATUS (and bank select bit) register.
e) Restores the W register.

EXAMPLE 8-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP          ; Copy W to TEMP register, could be bank one or zero
SWAPF   STATUS, W       ; Swap status to be saved into W
BCF     STATUS, RP0     ; Change to bank zero, regardless of current bank
MOVWF   STATUS_TEMP     ; Save status to bank zero STATUS_TEMP register
:                       :
:                       ; Interrupt Service Routine
:                       ;
SWAPF   STATUS_TEMP, W  ; Swap STATUS_TEMP register into W
                        ;   (sets bank to    original state)
MOVWF   STATUS          ; Move W into STATUS register
SWAPF   W_TEMP, F       ; Swap W_TEMP
SWAPF   W_TEMP, W       ; Swap W_TEMP into W
```


© 1995 Microchip Technology Inc.

8.7 Watchdog Timer (WDT)

The watchdog timer is realized as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming configuration fuse WDTE as a '0' (Section 8.1).

8.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

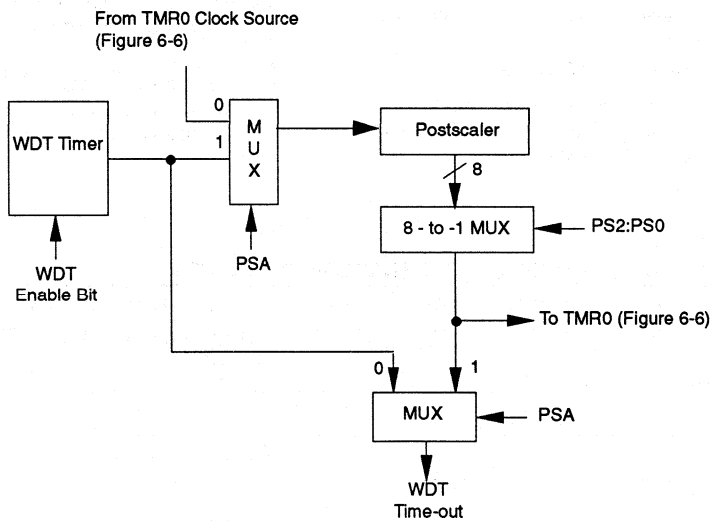
The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a WDT time-out.

8.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

FIGURE 8-16: WATCHDOG TIMER BLOCK DIAGRAM



Note: PSA and PS2:PS0 are bits in the OPTION register.

TABLE 8-7: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	–	–	–	CP	PWRTE	WDTE	FOSC1	FOSC0
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Note 1: The shaded cells are not used by the Watchdog Timer.

© 1995 Microchip Technology Inc.

8.8 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing the SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption, in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin, and disable external clocks. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS. The contribution from on chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

It should be noted that a RESET generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

8.8.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin.
2. WDT time-out reset (if WDT was enabled).
3. Interrupt from RB0/INT pin, RB port change, or data EEPROM write complete.

Peripherals cannot generate interrupts during SLEEP, since no on-chip Q clocks are present.

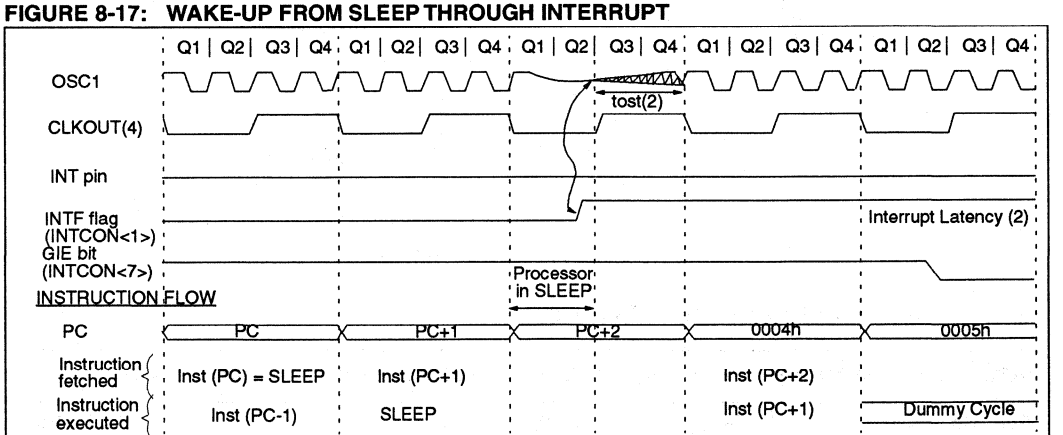
The first event ($\overline{MCLR}$ reset) will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

While the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

Note: If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake from sleep. The SLEEP instruction is completely executed.

The WDT is cleared when the device wakes-up from sleep, regardless of the source of wake-up.

FIGURE 8-17: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT or LP oscillator mode assumed.
2: Tost = 1024 Tosc (drawing not to scale). This delay will not be there for RC osc mode.
3: When GIE is set, processor jumps to interrupt routine after wake-up. If GIE is clear, execution will continue in line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

© 1995 Microchip Technology Inc.

8.9 Code Protection

The code in the program memory and data EEPROM memory can be protected by programming the code protect bit.

Refer to Figure 8-1 for the code protection bit assignment for the PIC16C84.

8.10 ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations to store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable only during program/verify. Only the 4 least significant bits of ID location are usable.

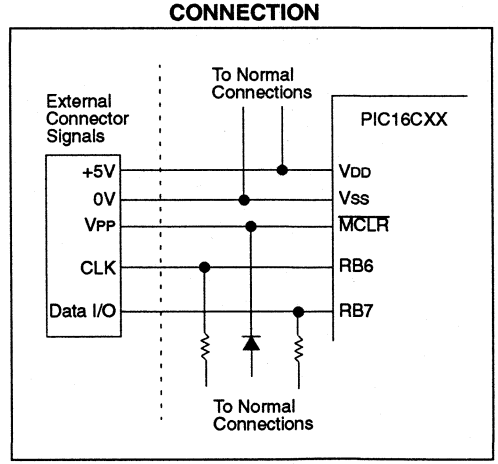
8.11 In-Circuit Serial Programming

PIC16C84 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. Customers can manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product, allowing the most recent firmware or custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low, while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) points to location 00h. A 6-bit command is then supplied to the device, 14-bits of program data is then supplied to or from the device, using load or a read-type instructions. For complete details of serial programming, please refer to the PIC16CXX Programming Specifications (Literature #DS30189).

FIGURE 8-18: TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION



NOTES:

© 1995 Microchip Technology Inc.

9.0 INSTRUCTION SET SUMMARY

Each PIC16C84 instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C84 instruction set summary in Table 9-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 9-1 shows the opcode field descriptions.

Byte-oriented instructions: 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed in the file register specified by the instruction.

Bit-oriented instructions: 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

Literal and control operations: 'k' represents an eight or eleven bit constant or literal value.

TABLE 9-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination (Either the W register or the specified register file location)
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- Byte-oriented
- Bit-oriented
- Literal and control

All instructions are executed within a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. The execution takes two instruction cycles with the second cycle executed as a NOP. Each cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µsec. The instruction execution time is 2 µsec for program branches.

Table 9-2 lists the instructions recognized by Microchip's assembler (MPASM).

Figure 9-1 shows the three general formats of instructions.

Note: To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 9-1: GENERAL FORMAT FOR INSTRUCTIONS

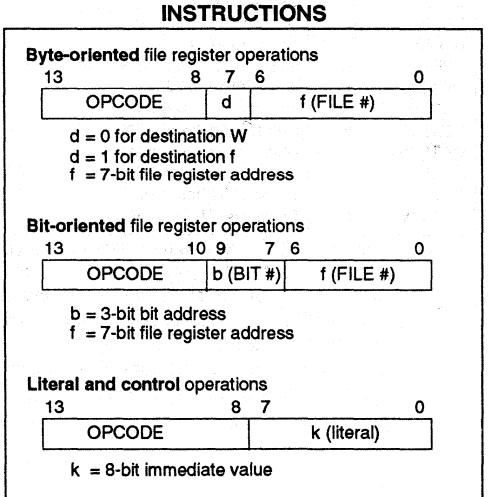


TABLE 9-2: INSTRUCTION SET SUMMARY

Mnemonic, Operands		Description	Cycles	14-Bit Opcode msb			lsb	Status Affected	Notes
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W and f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	None	1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	None	1,2,3
IORWF	**f, d**	Inclusive OR W and f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff	None	
NOP	**-**	No Operation	1	00	0000	0xx0	0000	None	
RLF	**f, d**	Rotate left f through carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff	None	1,2
XORWF	**f, d**	Exclusive OR W and f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff	None	1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff	None	1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff	None	3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff	None	3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal to W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal to W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear watchdog timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk	None	
IORLW	**k**	Inclusive OR literal to W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk	None	
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001	None	
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk	None	
RETURN	**-**	Return from subroutine	2	00	0000	0000	1000	None	
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Exclusive OR literal to W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (i.e., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d=1), the prescaler will be cleared if assigned to the TMR0.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

© 1995 Microchip Technology Inc.

9.1 Instruction Descriptions

ADDLW **Add Literal to W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → (W)

Status Affected: C, DC, Z

Encoding:

11	111x	kkkk	kkkk

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed back in the W register.

Words: 1

Cycles: 1

Example
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

ANDLW **And Literal to W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

11	1001	kkkk	kkkk

Description: The contents of W register is AND'ed with the eight bit literal 'k'. The result is placed back in the W register.

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

ADDWF **Add W and f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

00	0111	dfff	ffff

Description: Add the contents of the W register to register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

ANDWF **AND W to f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

00	0101	dfff	ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

© 1995 Microchip Technology Inc.

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

01	00bb	bfff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example
```
BCF     FLAG_REG,  7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

01	01bb	bfff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example
```
BSF     FLAG_REG,  7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BTFSC **Bit Test f, Skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

01	10bb	bfff	ffff

Description: If bit 'b' in register 'f' is 0 then the next instruction is skipped.
If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```
Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address TRUE
if FLAG<1>=1,
PC=address FALSE

© 1995 Microchip Technology Inc.

BTFSS **Bit Test f, skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is 1 then the next instruction is skipped.
If bit 'b' is 1, then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE    BTFSC   FLAG,1
FALSE   GOTO    PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1>=0,
PC=address FALSE
if FLAG<1>=1,
PC=address TRUE

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 2047

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Subroutine call. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
HERE    CALL    THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: 0 ≤ f ≤ 127

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

00	0001	1fff	ffff

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example

```
CLRF    FLAG_REG
```

Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W Register**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

00	0001	0xxx	xxxx

Description: W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example

```
CLRW
```

Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

© 1995 Microchip Technology Inc.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0100

Description: The CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescale = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

00	1001	dfff	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) – 1 → (dest)

Status Affected: Z

Encoding:

00	0011	dfff	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, Skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) – 1 → (dest); skip if result = 0

Status Affected: None

Encoding:

00	1011	dfff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

© 1995 Microchip Technology Inc.

GOTO **Go to address**

Syntax: [*label*] GOTO k

Operands: 0 ≤ k ≤ 2047

Operation: k → PC<10:0>
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Example
```
GOTO THERE
```
After Instruction
PC = Address THERE

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
INCF    CNT, 1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.
If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example
```
HERE     INCFSZ    CNT, 1
         GOTO      LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT + 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE +1

IORLW **Inclusive OR Literal to W**

Syntax: [*label*] IORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk

Description: The contents of the W register are OR'ed to the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
IORLW   0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF

© 1995 Microchip Technology Inc.

IORWF **Inclusive OR W to f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (W)

Status Affected: Z

Encoding:

00	0100	dfff	ffff

Description: Inclusive OR the W register to register 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example
```
IORWF          RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93

MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

00	1000	dfff	ffff

Description: The contents of register f is moved to destination d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example
```
MOVF     FSR, 0
```
After Instruction
W = value in FSR register

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

11	00XX	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example
```
MOVLW    0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

00	0000	1fff	ffff

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example
```
MOVWF       OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

© 1995 Microchip Technology Inc.

NOP — **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

00	0000	0xx0	0000

Description: No operation.

Words: 1

Cycles: 1

Example

```
NOP
```

RETFIE — **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

00	0000	0000	1001

Description: The Stack is popped and Top of Stack (TOS) is loaded into the PC. Interrupts are enabled by setting the Global Interrupt Enable This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

OPTION — **Load Option Register**

Syntax: [*label*] OPTION

Operands: None

Operation: W → OPTION;

Status Affected: None

Encoding:

00	0000	0110	0010

Description: The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.

Words: 1

Cycles: 1

Example

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

RETLW — **Return Literal to W**

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W;
TOS → (PC)

Status Affected: None

Encoding:

11	01xx	kkkk	kkkk

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.

Words: 1

Cycles: 2

Example

```
      CALL TABLE  ;W contains table
                  ;offset value
      •           ;W now has table value
      •
      •
TABLE ADDWF PC    ;W = offset
      RETLW k1    ;Begin table
      RETLW k2    ;
      •
      •
      •
      RETLW kn    ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k7

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → (PC)

Status Affected: None

Encoding:

00	0000	0000	1000

Description: Return from subroutine. The stack is popped and the Top of Stack (TOS) is loaded into the program counter. This is a two cycle instruction.

Words: 1

Cycles: 2

Example
```
RETURN
```
After Interrupt
PC = TOS

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

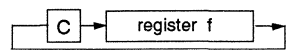
Operation: See description below

Status Affected: C

Encoding:

00	1100	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example
```
RRF     REG1,0
```
Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 0111 0011
C = 1

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

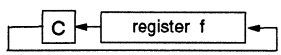
Operation: See description below

Status Affected: C

Encoding:

00	1101	dfff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example
```
RLF     REG1,0
```
Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

SLEEP **Go into Standby Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

00	0000	0110	0011

Description: The power down status bit ($\overline{PD}$) is cleared. Time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped.

Words: 1

Cycles: 1

Example:
```
SLEEP
```


© 1995 Microchip Technology Inc.

SUBLW **Subtract W from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: $k - (W) \rightarrow (W)$

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example 1:
```
SUBLW   0x02
```
Before Instruction
W = 1
C = ?
After Instruction
W = 1
C = 1; result is positive

Example 2: Before Instruction
W = 2
C = ?
After Instruction
W = 0
C = 1; result is zero

Example 3: Before Instruction
W = 3
C = ?
After Instruction
W = FF
C = 0; result is negative

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: $(f) - (W) \rightarrow (dest)$

Status Affected: C, DC, Z

Encoding:

00	0010	dfff	ffff

Description: Subtract (2's complement methodize W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1,1
```
Before Instruction
REG1 = 3
W = 2
C = ?
After Instruction
REG1 = 1
W = 2
C = 1; result is positive

Example 2: Before Instruction
REG1 = 2
W = 2
C = ?
After Instruction
REG1 = 0
W = 2
C = 1; result is zero

Example 3: Before Instruction
REG1 = 1
W = 2
C = ?
After Instruction
REG1 = FF
W = 2
C = 0; result is negative

© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label* SWAPF f,d]

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: f<3:0> → d<7:4>,
f<7:4> → d<3:0>

Status Affected: None

Encoding:

00	1110	dfff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example

```
SWAP F REG,  0
```

Before Instruction
REG1 = 0xA5

After Instruction
REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: 5 ≤ f ≤ 7

Operation: W → TRIS register f;

Status Affected: None

Encoding:

00	0000	0110	0fff

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

Note: To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

XORLW **Exclusive OR Literal to W**

Syntax: [*label*] XORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

11	1010	kkkk	kkkk

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example

```
XORLW  0xAF
```

Before Instruction
W = 0xB5

After Instruction
W = 0x1A

XORWF **Exclusive OR W to f**

Syntax: [*label*] XORWF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

00	0110	dfff	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example

```
XORWF   REG 1
```

Before Instruction
REG = 0xAF
W = 0xB5

After Instruction
REG = 0x1A
W = 0xB5

© 1995 Microchip Technology Inc.

10.0 DEVELOPMENT SUPPORT

10.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

10.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 10-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 (and better) machines in the Microsoft Windows™ 3.x environment. Thus, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC-Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 10-1.

FIGURE 10-1: PICMASTER SYSTEM CONFIGURATION

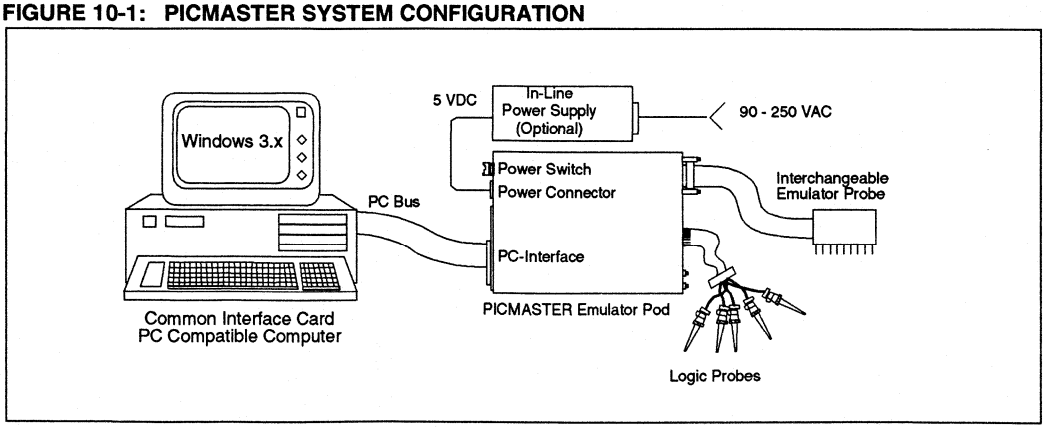


TABLE 10-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

10.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

10.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

10.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84 and PIC17C42. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

10.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

© 1995 Microchip Technology Inc.

10.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically, i.e., by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and subtitles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

10.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

10.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

10.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP Edition, for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

10.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 10-2.

TABLE 10-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

11.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) ..... 0 to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)
Total power dissipation (Note 1) ..... 800 mW
Maximum current out of V_{SS} pin ..... 150 mA
Maximum current into V_{DD} pin ..... 100 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 20 mA
Maximum current sunk by PORTA ..... 80 mA
Maximum current sourced by PORTA ..... 50 mA
Maximum current sunk by PORTB ..... 150 mA
Maximum current sourced by PORTB ..... 100 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 11-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	16C84-04	16C84-10	16LC84-04
RC	V_{DD}: 4.0V to 6.0V I_{DD}: 4.5 mA max. at 5.5V I_{PD}: 100 µA max. at 4V WDT dis Freq: 4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4V WDT dis Freq: 4 MHz max.	V_{DD}: 2.0V to 6.0V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 3V WDT dis Freq: 4 MHz max.
XT	V_{DD}: 4.0V to 6.0V I_{DD}: 4.5 mA max. at 5.5V I_{PD}: 100 µA max. at 4V WDT dis Freq: 4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4V WDT dis Freq: 4 MHz max.	V_{DD}: 2.0V to 6.0V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 3V WDT dis Freq: 4 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 4.5 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4.5V WDT dis Freq: 4 MHz	V_{DD}: 4.5V to 5.5V I_{DD}: 10 mA max. at 5.5V typ. I_{PD}: 1.0 µA typ. at 4.5V WDT dis Freq: 10 MHz max.	Do not use in HS mode
LP	V_{DD}: 2.0V to 6.0V I_{DD}: 60 µA typ. at 32 kHz, 2.0V I_{PD}: 26 µA typ. at 2.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	V_{DD}: 2.0V to 6.0V I_{DD}: 60 µA max. at 32 kHz, 2.0V I_{PD}: 100 µA max. at 4.0V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

© 1995 Microchip Technology Inc.

11.1 DC CHARACTERISTICS: PIC16C84-04 (Commercial, Industrial) PIC16C84-10 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	4.0	—	6.0	V	XT, RC and LP osc configuration (16C84-04)
		4.5	—	5.5	V	HS osc configuration (16C84-10)
RAM Data Retention Voltage (Note 1)	VDR	1.5*	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					RC and XT osc configuration
		—	7.3	10	mA	FOSC = 4 MHz, VDD = 5.5V
						During EEPROM programming
		—	1.8	4.5	mA	FOSC = 4 MHz, VDD = 5.5V (Note 4)
		—	35	400	µA	LP osc configuration (PIC16C84-04)
						FOSC = 32 kHz, VDD = 4.0V, WDT disabled
		—	5	10	mA	HS osc configuration (PIC16C84-10)
						FOSC = 10 MHz, VDD = 5.5V
Power Down Current (Note 3)	IPD	—	40	100	µA	VDD = 4.0V, WDT enabled, -40°C to +85°C
		—	38	100	µA	VDD = 4.0V, WDT disabled, -0°C to +70°C
		—	38	100	µA	VDD = 4.0V, WDT disabled, -40°C to +85°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

© 1995 Microchip Technology Inc.

11.2 DC CHARACTERISTICS: PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.0	—	6.0	V	XT, RC, and LP osc configuration
RAM Data Retention Voltage (Note 1)	VDR	1.5 *	—	—	V	Device in SLEEP mode
VDD start voltage to guarantee Power-On Reset	VPOR	—	VSS	—	V	See section on Power-On Reset for details
VDD rise rate to guarantee Power-On Reset	SVDD	0.05*	—	—	V/ms	See section on Power-On Reset for details
Supply Current (Note 2)	IDD					RC and XT osc configuration
		—	7.3	10	mA	FOSC = 4 MHz, VDD = 5.5V
						During EEPROM programming
		—	1.8	4.5	mA	FOSC = 4 MHz, VDD = 5.5V (Note 4)
						LP osc configuration
		—	60	400	µA	FOSC = 32 kHz, VDD = 2.0V, WDT disabled
Power Down Current (Note 3)	IPD	—	26	100	µA	VDD = 2.0V, WDT enabled, -40°C to +85°C
		—	26	100	µA	VDD = 2.0V, WDT disabled, 0°C to +70°C
		—	26	100	µA	VDD = 2.0V, WDT disabled, -40°C to +85°C

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

11.3 DC CHARACTERISTICS: PIC16C84-04 (Commercial, Industrial) PIC16C84-10 (Commercial, Industrial) PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature
-40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Table 11-1 and Table 11-2.

Characteristic	Sym	Min	Typ†	Max	Units	Conditions
Input Low Voltage						
I/O ports	VIL					
with TTL buffer		VSS	—	0.8	V	
with Schmitt Trigger buffer		VSS	—	0.2 VDD	V	
MCLR, RA4/T0CKI,OSC1 (in RC mode)		Vss	—	0.2 VDD	V	
OSC1 (in XT, HS and LP)		Vss	—	0.3 VDD	V	Note1
Input High Voltage						
I/O ports	VIH					
with TTL buffer		0.36 VDD	—	VDD	V	VDD ≤ 5.5V (Note 4)
with Schmitt Trigger buffer		0.45 VDD	—	VDD		VDD ≤ 6.0V (Note 4)
MCLR, RA4/T0CKI, OSC1 (RC mode)		0.85 VDD	—	VDD	V	
OSC1 (XT, HS and LP)		0.7 VDD	—	VDD	V	Note1
PORTB weak pull-up current	IPURB	50*	250*	400*	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 2, 3)						
I/O ports	IIL	—	—	±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
MCLR, RA4/T0CKI		—	—	±5	µA	Vss ≤ VPIN ≤ VDD
OSC1/CLKIN		—	—	±5	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
Output Low Voltage						
I/O ports	VOL	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C
OSC2/CLKOUT (RC osc configuration)		—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C
Output High Voltage						
I/O ports (Note 3)	VOH	VDD - 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
OSC2/CLKOUT (RC osc configuration)		VDD - 0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
Capacitive Loading Specs on Output Pins						
OSC2/CLKOUT pin	COSC2	—	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
All I/O pins and OSC2 (in RC mode)	CIO	—	—	50	pF	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C84 be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: The user may use better of the two specs.

© 1995 Microchip Technology Inc.

11.4 DC CHARACTERISTICS: PIC16C84-04 (Commercial, Industrial) PIC16C84-10 (Commercial, Industrial) PIC16LC84-04 (Commercial, Industrial)

DC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in DC spec Table 11-1 and Table 11-2					
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Data EEPROM Memory						
Endurance	ED	100,000	1,000,000	—	E/W	
VDD for read/write	VDRW	VMIN	—	0.2 VDD	V	VMIN = Minimum operating voltage
Erase/Write cycle time	TDEW	—	10	—	ms	Note1
Program EEPROM Memory						
Endurance	EP	100	—	—	E/W	
VDD for read	VPR	VMIN	—	VDD	V	VMIN = Minimum operating voltage
VDD for erase/write	VPEW	4.5	—	5.5	V	
Erase/Write cycle time	TPEW	—	10	-	ms	Note1

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The user should use interrupts or pull the EEIF or WR bits to ensure the write cycle has completed.

11.5 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

T			
F	Frequency	T	Time

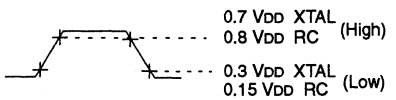
Lowercase symbols (pp) and their meanings:

pp			
ck	CLKOUT	osc	OSC1
io	I/O port	t0	T0CKI
mc	$\overline{\text{MCLR}}$		

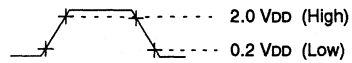
Uppercase symbols and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-impedance

FIGURE 11-1: PARAMETER MEASUREMENT INFORMATION



OSC1 Measurement Points



I/O Port Measurement Points

All timings are measured between high and low measurement points as indicated in the figure.

© 1995 Microchip Technology Inc.

11.6 Timing Diagrams and Specifications

FIGURE 11-2: EXTERNAL CLOCK TIMING

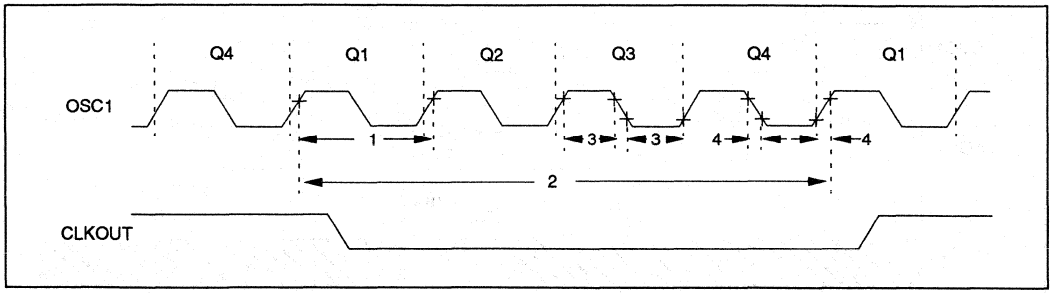


TABLE 11-2: EXTERNAL CLOCK TIMING REQUIREMENTS

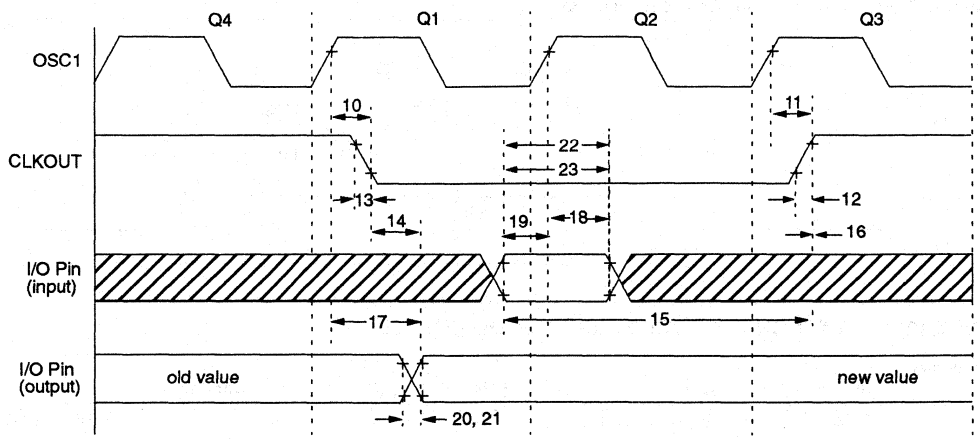
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency (Note 1)**	DC	—	2	MHz	XT, RC osc mode, 2V ≤VDD ≤6V
			DC	—	4	MHz	XT, RC osc mode, 3V ≤VDD ≤6V
			DC	—	10	MHz	HS osc mode (PIC16C84-10)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	2	MHz	RC osc mode, 2V ≤ VDD ≤ 6V
			DC		4	MHz	RC osc mode, 3V ≤ VDD ≤ 6V
			0.1	—	2	MHz	XT osc mode, 2V ≤ VDD ≤ 6V
			0.1	—	4	MHz	XT osc mode, 3V ≤ VDD ≤ 6V
			1		10	MHz	HS osc mode (PIC16C84-04)
			DC	—	200	kHz	LP osc mode (PIC16LC84-04)
1	Tosc	**External CLKIN Period (Note 1)**	250	—	—	ns	XT and RC osc mode
			250	—	—	ns	HS osc mode (PIC16C84-04)
			100	—	—	ns	HS osc mode (PIC16C84-10)
			5	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	1,000	ns	HS osc mode
			100	—	1,000	ns	HS osc mode (PIC16C84-10)
			5	—	—	µs	LP osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	0.4	4/Fosc	DC	µs	
3	TosL,TosH	Clock in (OSC1) Low or High Time	60 *	—	—	ns	XT oscillator, 2.0V ≤ VDD ≤ 3.0V
			50 *	—	—	ns	XT oscillator, 3.0V ≤ VDD ≤ 6.0V
			2 *	—	—	µs	LP oscillator
			50 *	—	—	ns	HS oscillator
4	TosR,TosF	Clock in (OSC1) Rise or Fall Time	25 *	—	—	ns	XT oscillator
			50 *	—	—	ns	LP oscillator
			25 *	—	—	ns	HS oscillator

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 11-3: CLKOUT AND I/O TIMING



Note: All tests must be done with specified capacitive loads (see datasheet) 50 pF on I/O pins and CLKOUT.

TABLE 11-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15	30	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15	30	ns	Note 1
12	TckR	CLKOUT rise time	—	5	15	ns	Note 1
13	TckF	CLKOUT fall time	—	5	15	ns	Note 1
14	TckL2ioV	CLKOUT↓ to Port out valid	—	—	0.5TCY+20	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.30TCY+30 *	—	—	ns	Note 1
16	TckH2iol	Port in hold after CLKOUT↑	0 *	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	100	ns	
18	TosH2iol	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22	Tinp	INT pin high or low time	20 *	—	—	ns	
23	Trbp	RB<7:4> change INT high or low time	20 *	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

© 1995 Microchip Technology Inc.

FIGURE 11-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

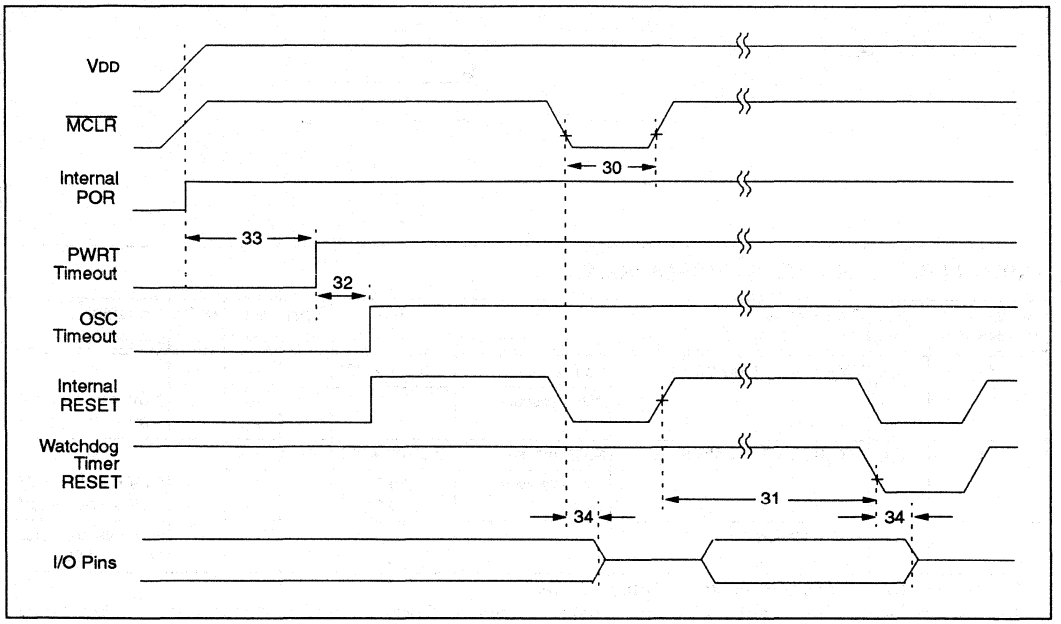


TABLE 11-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	350 * 150 *	— —	— —	ns ns	2.0V ≤ VDD ≤ 3.0V 3.0V ≤ VDD ≤ 6.0V
31	Twdt	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillation Start-up Timer Period	—	1024 Tosc	—	ms	Tosc = OSC1 period
33	Tpwrt	Power up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O Hi-impedance from $\overline{MCLR}$ Low or reset	—	—	100	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 11-5: TIMER0 CLOCK TIMINGS

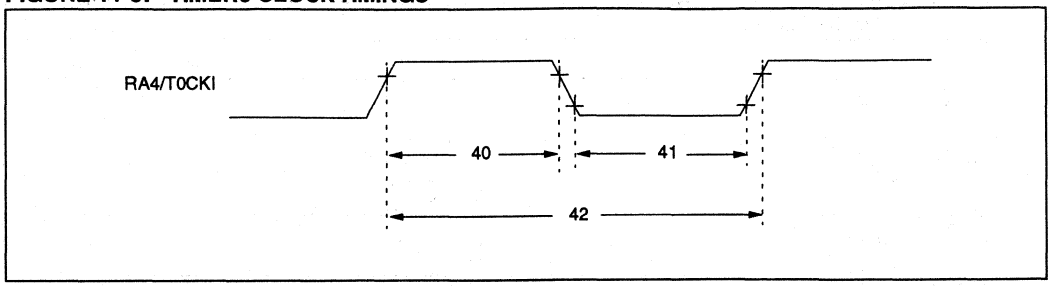


TABLE 11-5: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	50 *	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				30 *	—	—	ns	3.0V ≤ VDD ≤ 6.0V
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	50 *	—	—	ns	2.0V ≤ VDD ≤ 3.0V
				20 *	—	—	ns	3.0V ≤ VDD ≤ 6.0V
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

© 1995 Microchip Technology Inc.

12.0 DC & AC CHARACTERISTICS GRAPHS/TABLES

The data graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g., outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while 'max' or 'min' represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

FIGURE 12-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

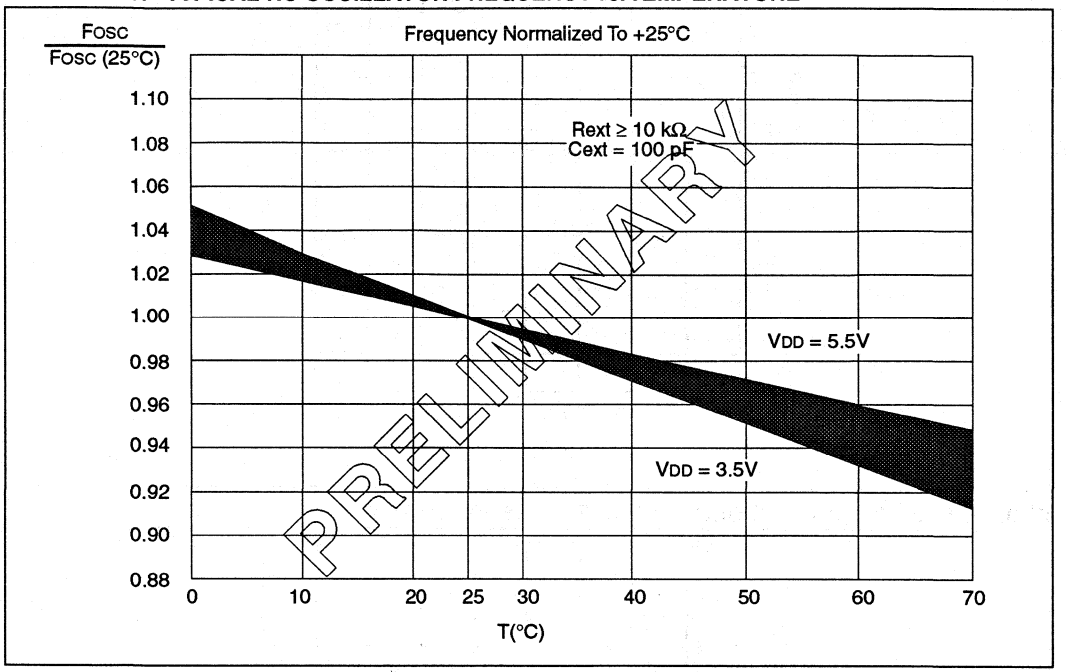


TABLE 12-1: RC OSCILLATOR FREQUENCIES *

Cext	Rext	Average Fosc @ 5V, 25°C	
20 pF	3.3k	4.68 MHz	± 27%
	5.1k	3.94 MHz	± 25%
	10k	2.34 MHz	± 29%
	100k	250.16 kHz	± 33%
100 pF	3.3k	1.49 MHz	± 25%
	5.1k	1.12 MHz	± 25%
	10k	620.31 kHz	± 30%
	100k	90.25 kHz	± 26%
300 pF	3.3k	524.24 kHz	± 28%
	5.1k	415.52 kHz	± 30%
	10k	270.33 kHz	± 26%
	100k	25.37 kHz	± 25%

*Measured in PDIP Packages.The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value.

FIGURE 12-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD*

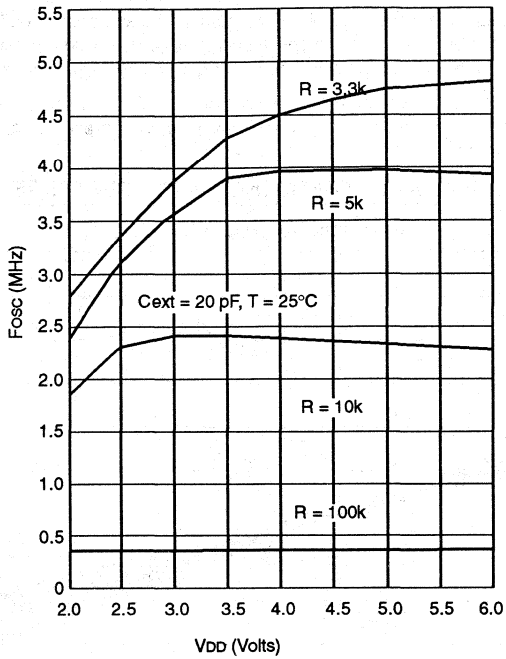
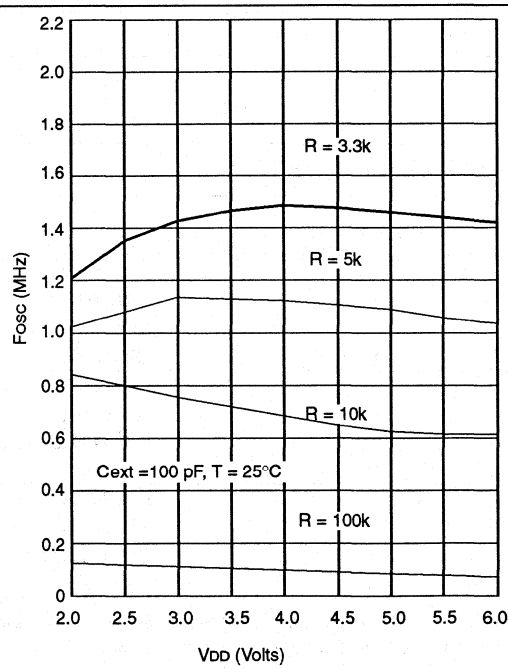


FIGURE 12-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD*



© 1995 Microchip Technology Inc.

FIGURE 12-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD*

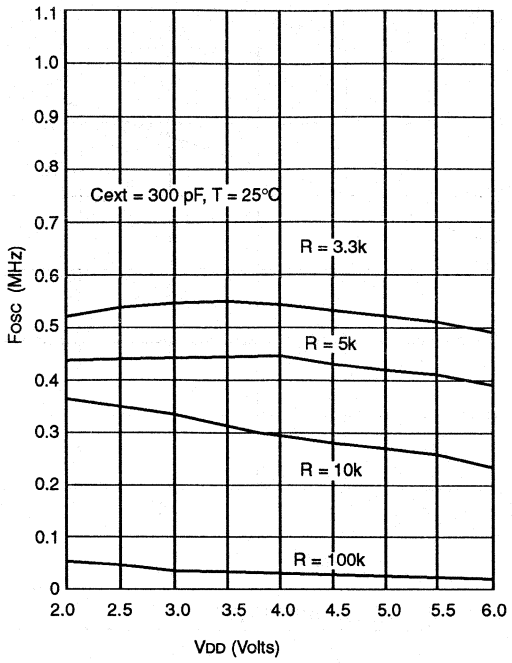


FIGURE 12-5: TYPICAL IPD vs. VDD WATCHDOG DISABLED (25°C)

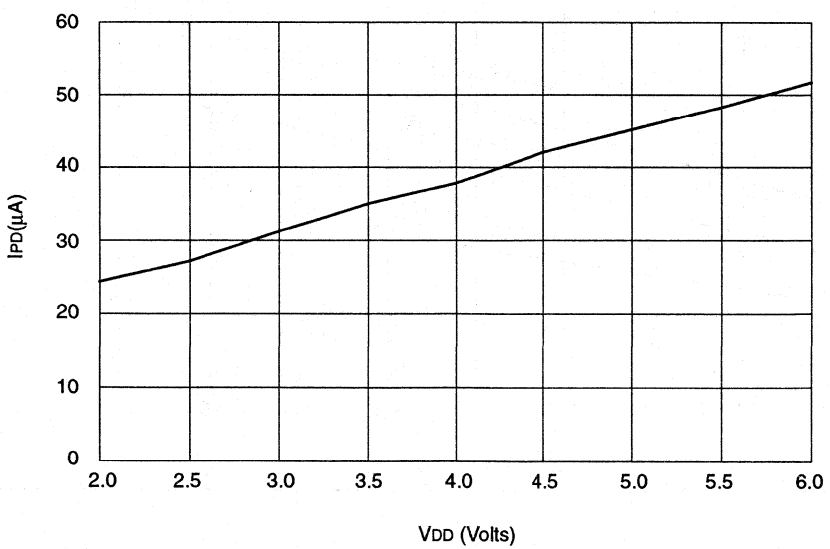


FIGURE 12-6: TYPICAL IPD vs. VDD WATCHDOG ENABLED (25°C)

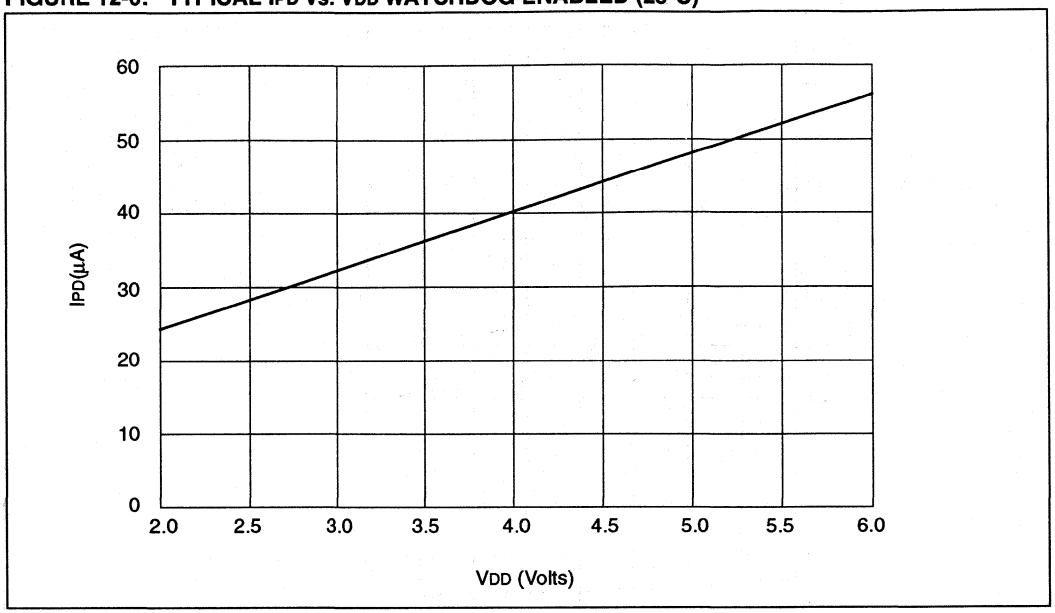
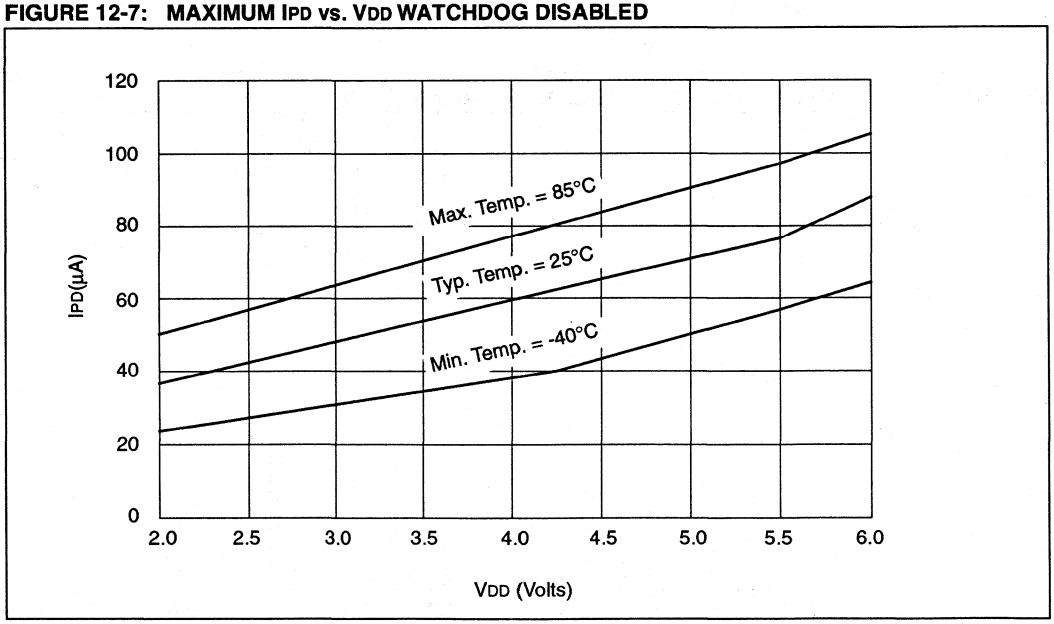
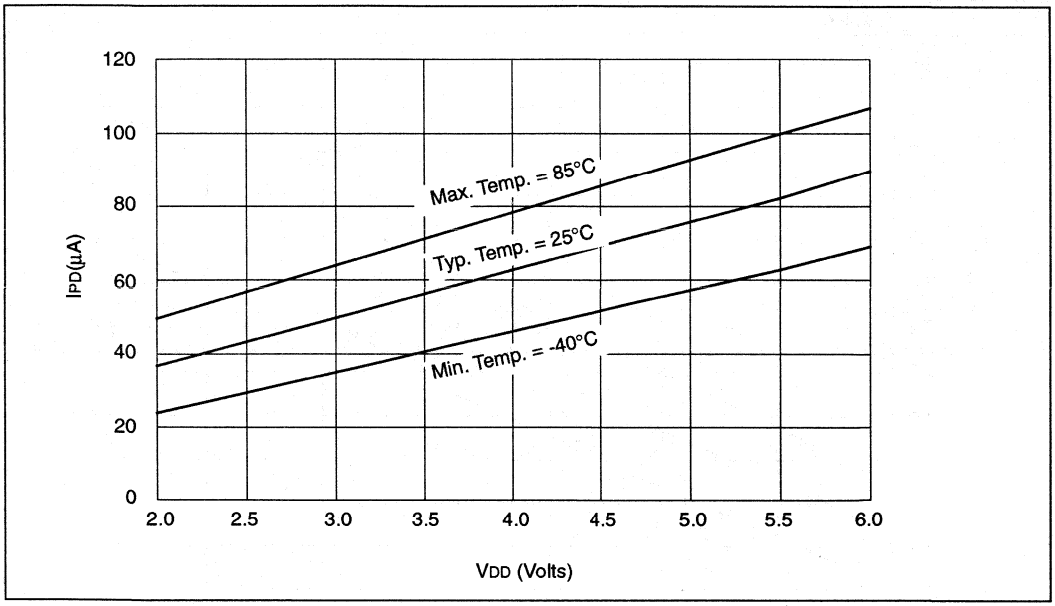


FIGURE 12-7: MAXIMUM IPD vs. VDD WATCHDOG DISABLED



© 1995 Microchip Technology Inc.

FIGURE 12-8: MAXIMUM IPD vs. VDD WATCHDOG ENABLED*



* IPD, with watchdog timer enabled, has two components: The leakage current which increases with higher temperature and the operating current of the watchdog timer logic which increases with lower temperature. At -40°C, the latter dominates explaining the apparently anomalous behavior.

FIGURE 12-9: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS vs. VDD

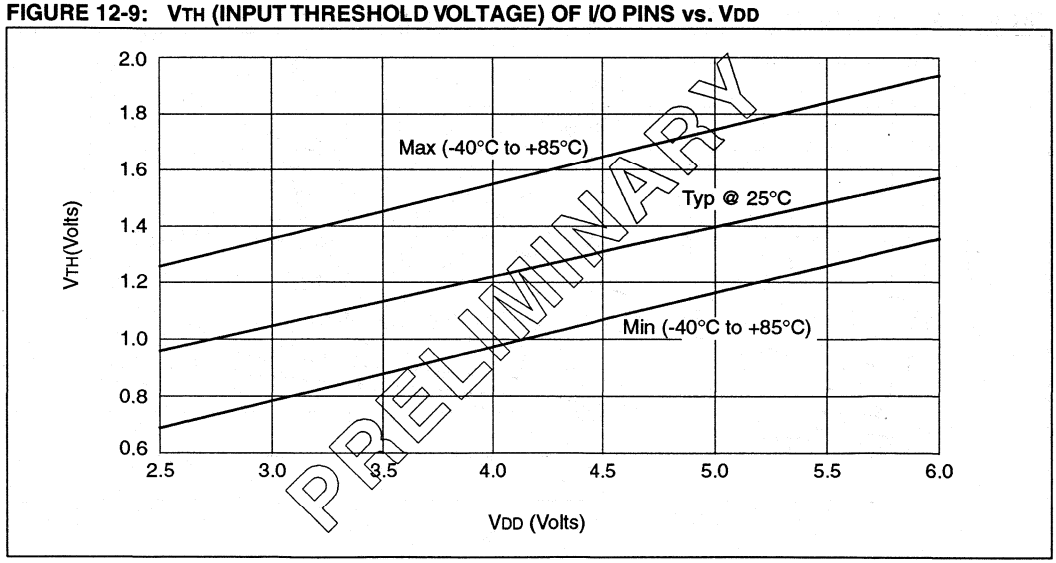


FIGURE 12-10: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs. VDD

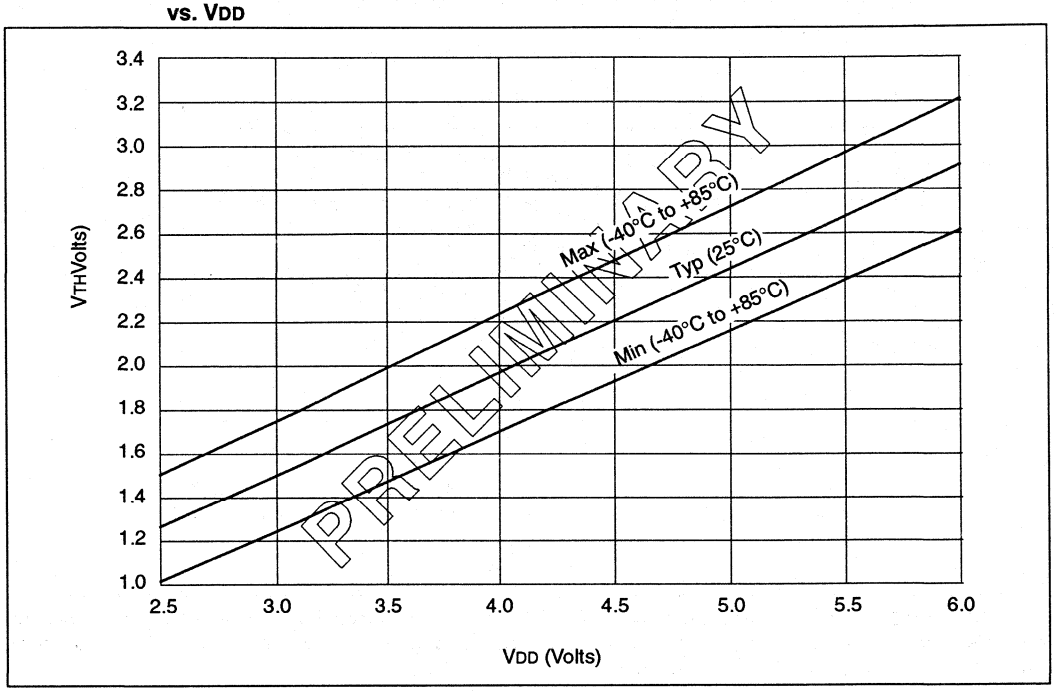
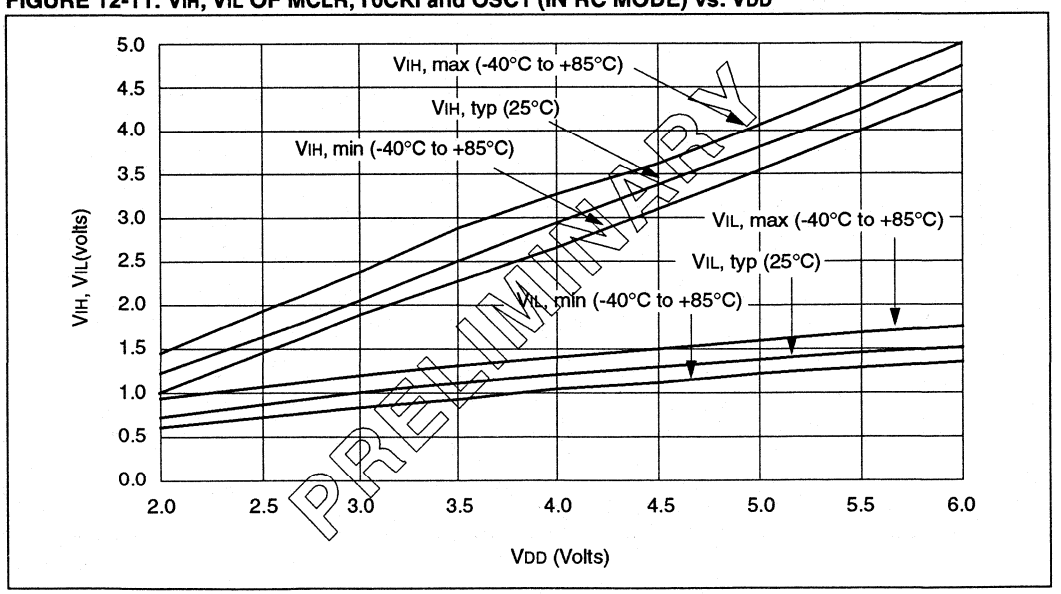


FIGURE 12-11: VIH, VIL OF MCLR, T0CKI and OSC1 (IN RC MODE) vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 12-12: TYPICAL IDD vs. FREQ (EXT CLOCK, 25°C)

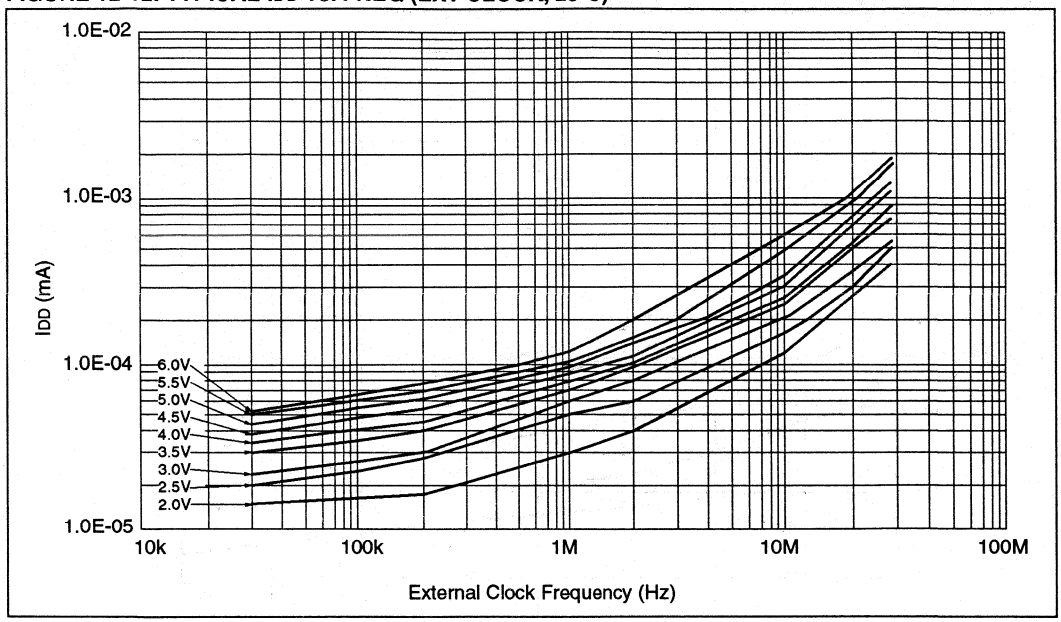


FIGURE 12-13: MAXIMUM IDD vs. FREQ (EXT CLOCK, -40° TO +85°C)

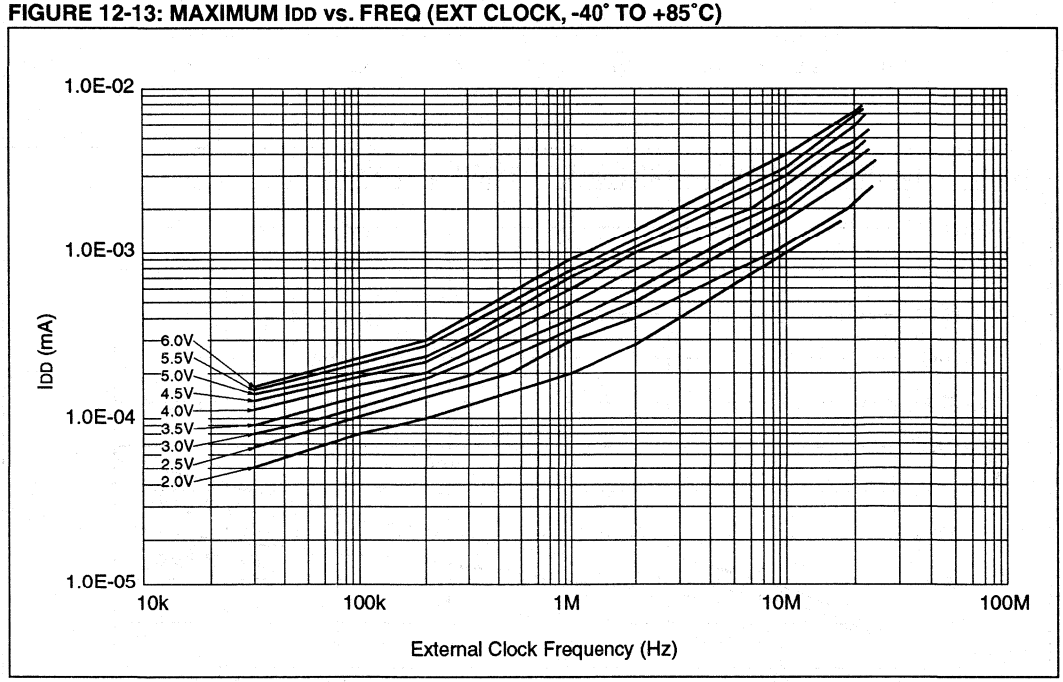


FIGURE 12-14: WDT TIMER TIME-OUT PERIOD vs. VDD

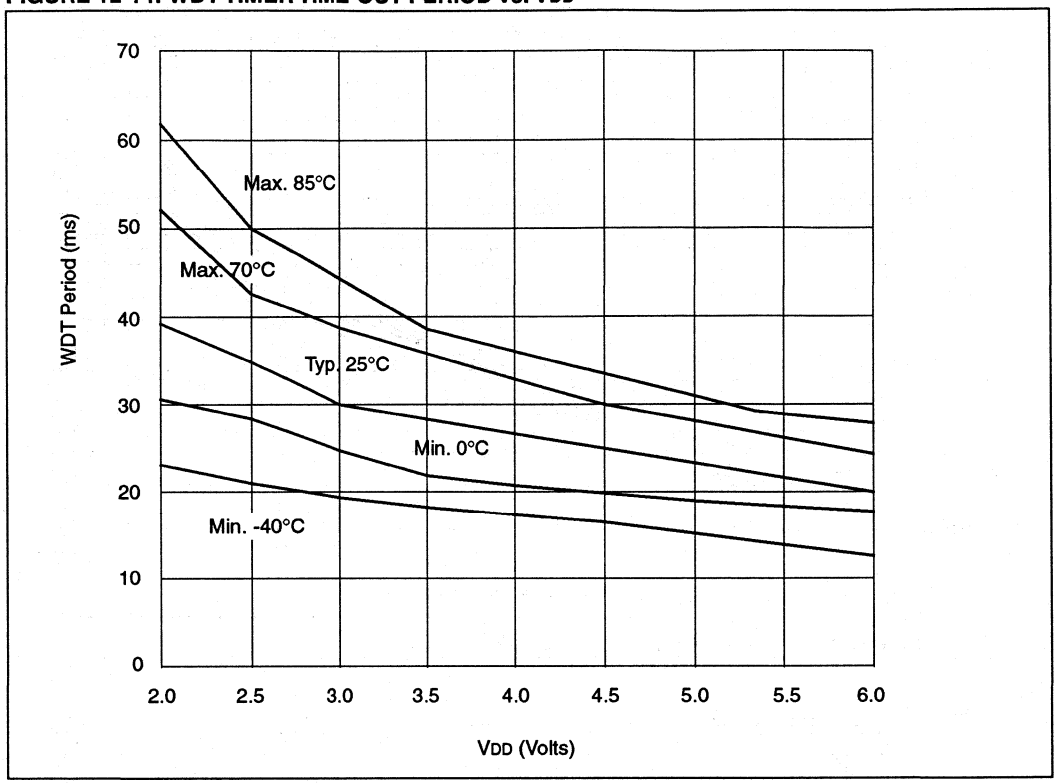
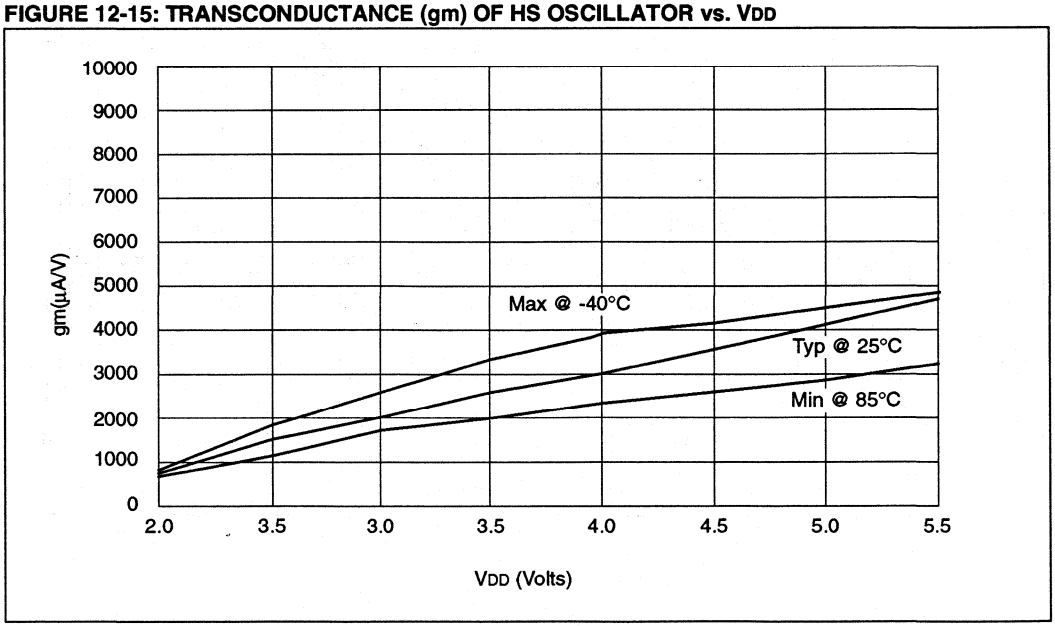


FIGURE 12-15: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 12-16: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR vs. VDD

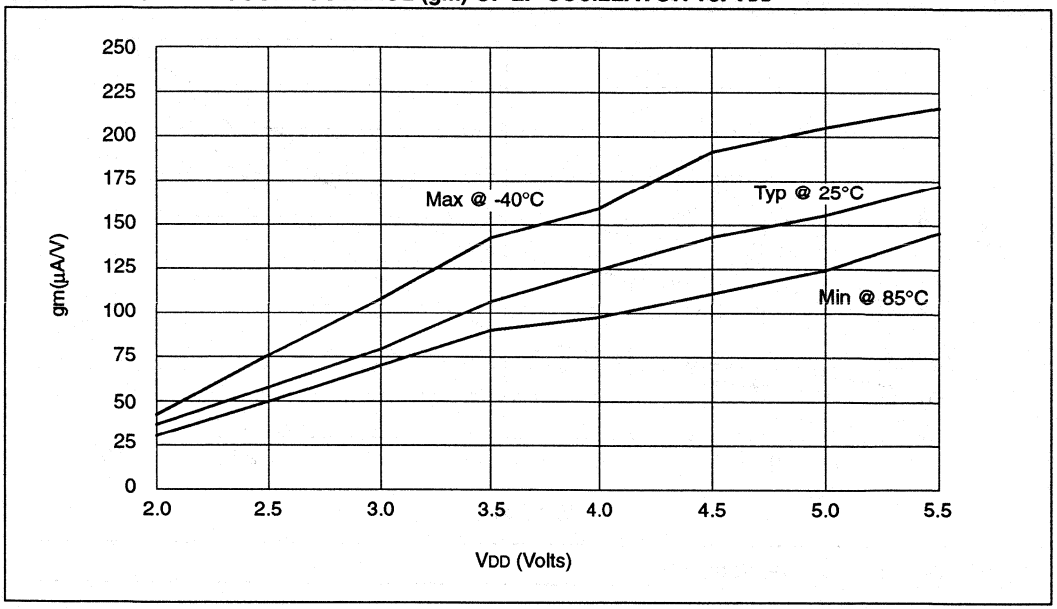


FIGURE 12-17: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD

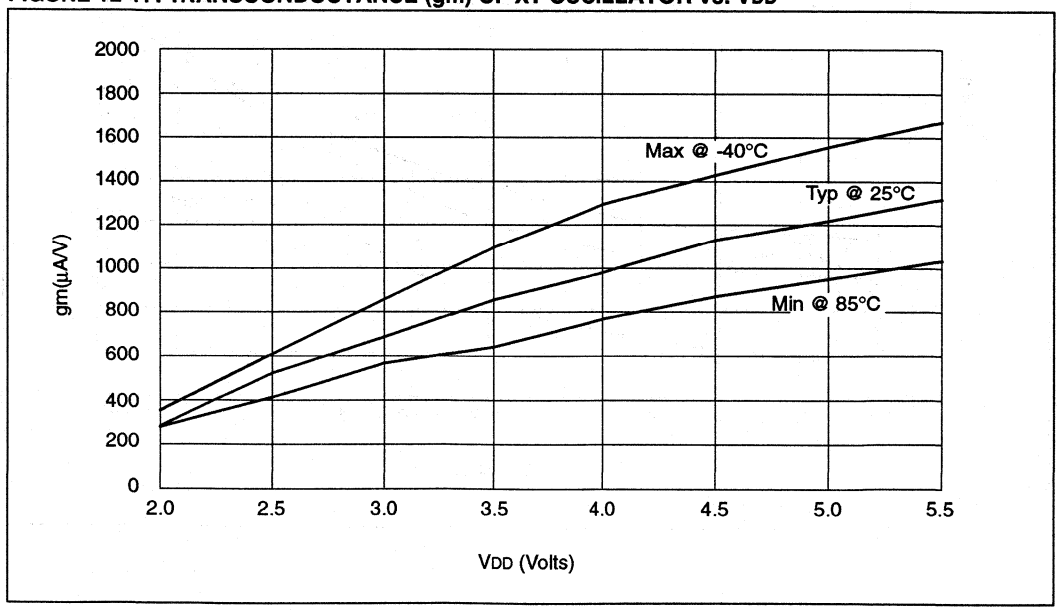


FIGURE 12-18: IOH vs. VOH, VDD = 3V

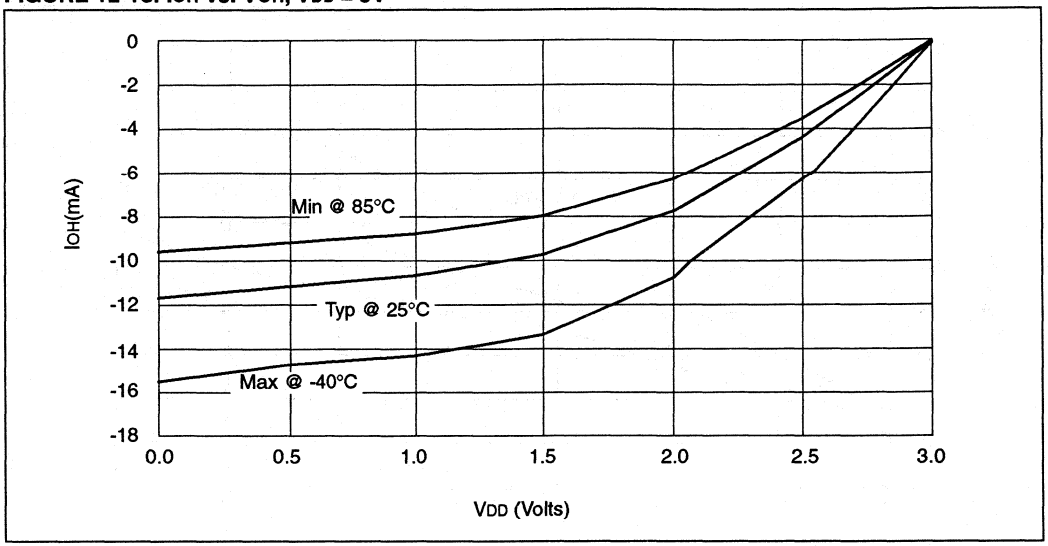
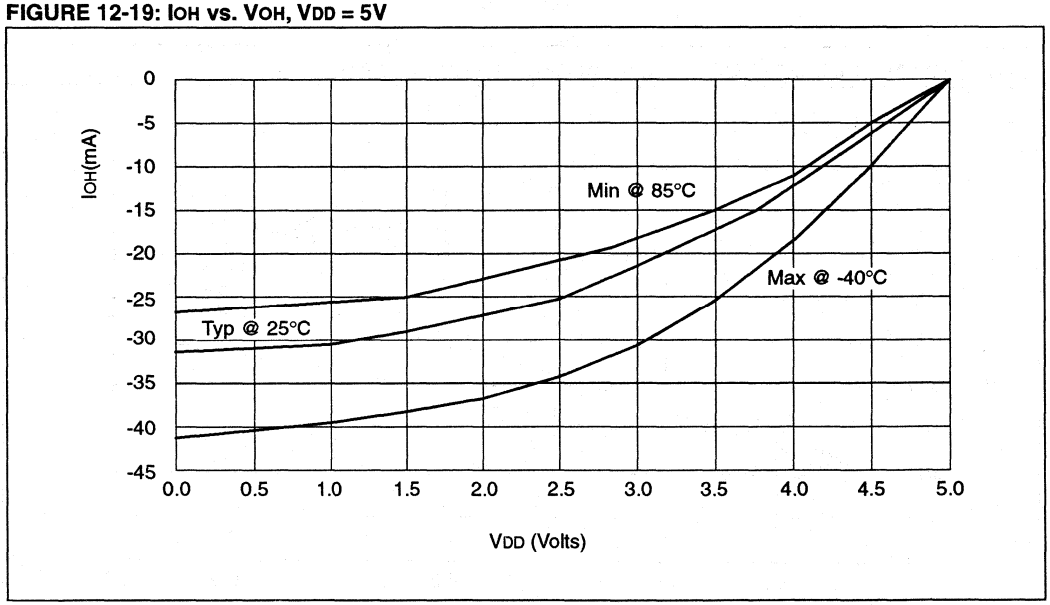


FIGURE 12-19: IOH vs. VOH, VDD = 5V



© 1995 Microchip Technology Inc.

FIGURE 12-20: IOL vs. VOL, VDD = 3V

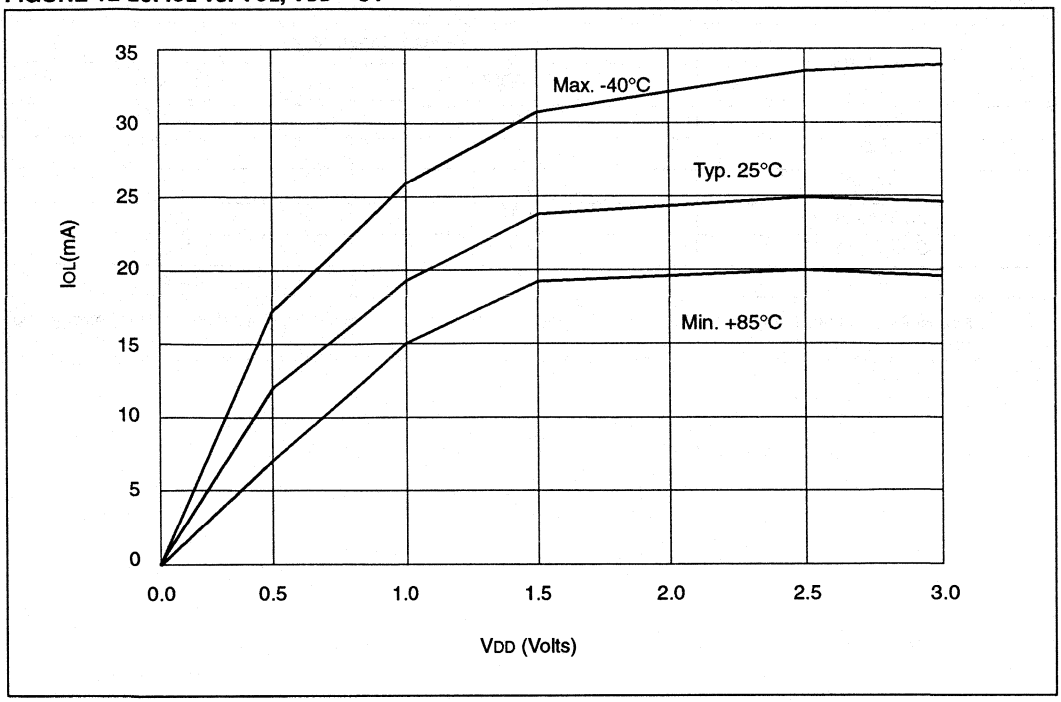


FIGURE 12-21: IOL vs. VOL, VDD = 5V

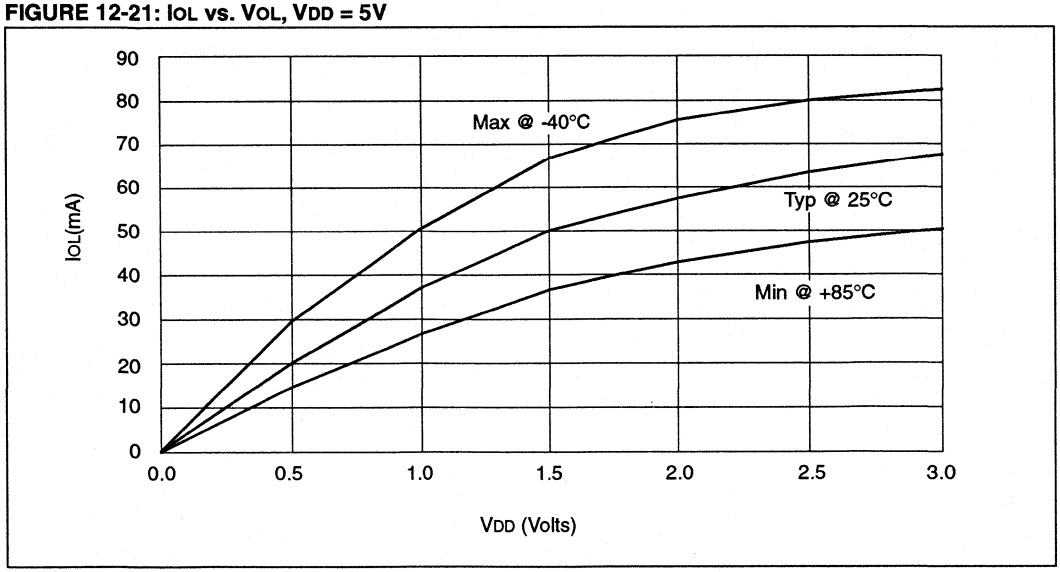


TABLE 12-2: INPUT CAPACITANCE *

Pin Name	Typical Capacitance (pF)	
	18L PDIP	18L SOIC
PORTA	5.0	4.3
PORTB	5.0	4.3
$\overline{MCLR}$	17.0	17.0
OSC1/CLKIN	4.0	3.5
OSC2/CLKOUT	4.3	3.5
T0CKI	3.2	2.8

* All capacitance values are typical at 25°C. A part to part variation of ±25% (three standard deviations) should be taken into account.

© 1995 Microchip Technology Inc.

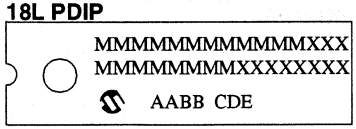
13.0 PACKAGING INFORMATION

For Package Dimension,
please refer to the Packaging section of the Data Book

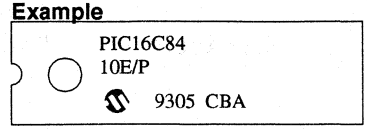
© 1995 Microchip Technology Inc.

13.1 Package Marking Information

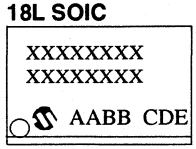
18L PDIP



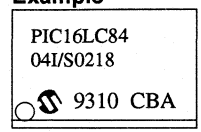
Example



18L SOIC



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A., S = Tempe, Arizona, U.S.A.
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

APPENDIX A: CHANGES

The following is the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14 bits. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and the register file (128 bytes now versus 32 bytes before).
2. A PC latch register (PCLATH) is added to handle program memory paging. PA2, PA1 and PA0 bits are removed from the status register and placed in the option register.
3. Data memory paging is redefined slightly. The status register is modified.
4. Four new instructions have been added: RETURN, RETFIE, ADDLW, and SUBLW. Two instructions, TRIS and OPTION, are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers, the Oscillator Start-Up Timer (OST) and Power-Up Timer (PWRT), are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt on change features.
13. T0CKI pin is also a port pin (RA4/T0CKI).
14. FSR is a full 8-bit register.
15. "In system programming" is made possible. The user can program PIC16CXX devices using only five pins: VDD, VSS, VPP, RB6 (clock) and RB7 (data in/out).

APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16C84, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for CALL, GOTO.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables for reallocation.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S NEW

The conversion of this Data Sheet into the desktop publishing software package, The structure of the document has been made consistent with other data sheet. This ensures that important topics are covered across all PIC16/17 families. Here is an overview list of new features:

- Data Sheet Structure / Outline

APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, some of the registers and control bits have been changed. Now control bits that do the same function, have the same name (regardless of processor family). Care must still be taken, since they may not be in the same special function register. The following lists the register and bit names that have changed:

TABLE 13-1: BIT NAME CHANGES

OLD NAME	NEW NAME
RTS	T0CS
RTE	T0SE

© 1995 Microchip Technology Inc.

APPENDIX E: PIC16/17 MICROCONTROLLERS

TABLE E-1: PIC17CXX FAMILY OF DEVICES

	Clock		Memory			Peripherals			Features			
	Maximum Frequency of Operation (MHz)	EPROM	RAM Data Memory (bytes)	Program Memory	Timer Module(s)	Captures PWMs	Serial Port(s) (SCI)	External Interrupts	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller modes.

3: PORTB has software-configurable weak pull-ups.

TABLE E-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	EPROM	EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
		Program Memory															
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	—	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE E-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals	Features		
	Maximum Frequency of Operation (MHz)	Program Memory (words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

E.1 Pin Compatibility

Devices that have the same package type; and VDD, VSS, and $\overline{MCLR}$ pin locations, are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

INDEX

© 1995 Microchip Technology Inc.

LIST OF EXAMPLES

LIST OF FIGURES

© 1995 Microchip Technology Inc.

LIST OF TABLES

© 1995 Microchip Technology Inc.

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks of Microchip Technology Incorporated.

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC16C84 Literature Number: DS30081E

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject? _____

5. What deletions from the data sheet could be made without affecting the overall usefullness? _____

6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

© 1995 Microchip Technology Inc.

PIC16C84 Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern:	3-Digit Pattern Code for QTP (blank otherwise)	
Package:	P = PDIP SO = SOIC (Gull Wing, 300 mil body)	
Temperature Range:	- = 0°C to +70°C I = –40°C to +85°C	
Frequency Range:	04 = 4 MHz -RC, -XT / 200 kHz -LP 10 = 10 MHz	
Device:	PIC16C84 :VDD range 4.0V to 6.0V PIC16C84T :(Tape and Reel) PIC16LC84 :VDD range 2.0V to 6.0V PIC16LC84T :(Tape and Reel)	

Examples:

a) PIC16C84 - 04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301

b) PIC16LC84 - 04I/SO = Industrial temp., SOIC package, 200 kHz, extended VDD limits

c) PIC16C84 - 10I/P = Industrial temp., PDIP package, 10 MHz, normal VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

MICROCHIP

PIC17C4X

High-Performance 8-Bit CMOS EPROM Microcontroller

Devices Included In This Data Sheet

- PIC17C42
- PIC17C43
- PIC17C44

High-Performance RISC-like CPU Features

- Only 58 single word instructions to learn
- All single cycle instructions (160 ns) except for program branches which are two-cycle and table reads/writes
- Operating speed:
  - DC - 25 MHz clock input
  - DC - 160 ns instruction cycle

Device	Program Memory	Data Memory
PIC17C44	8K	454
PIC17C43	4K	454
PIC17C42	2K	232

- Hardware Multiplier (PIC17C43/C44 Devices only)
- Interrupt capability
- 16 levels deep hardware stack
- Direct, indirect and relative addressing modes
- Internal/External program memory execution
- 64K x 16 addressable program memory space

Peripheral Features

- 33 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - RA2 and RA3 are open collector, high voltage (12V), high current (60 mA), I/O
- Two capture inputs and two PWM outputs
  - Captures are 16-bit, max resolution 160 ns
  - PWM resolution is 1- to 10-bit
- TMR0: 16-bit timer/counter with 8-bit programmable prescaler
- TMR1: 8-bit timer/counter
- TMR2: 8-bit timer/counter
- TMR3: 16-bit timer/counter
- Serial Communications Interface (SCI/USART)

Special microcontroller features

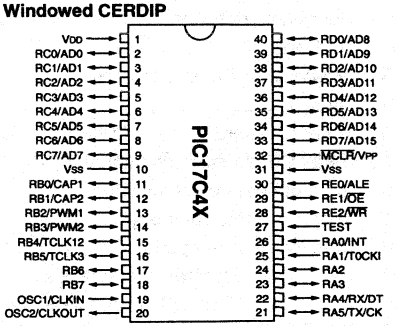
- Power-On Reset (POR), Power-Up Timer (PWRT) and Oscillator Start-Up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Technology

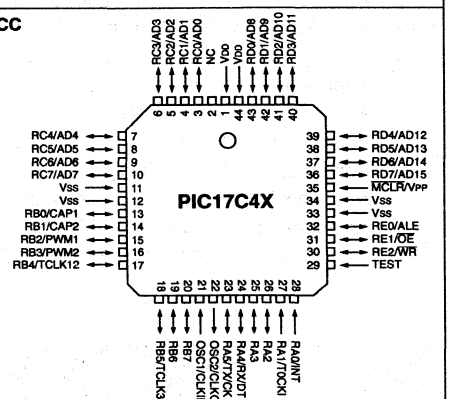
- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide operating voltage range (2.5V to 6.0V)
- Commercial and Industrial Temperature Range
- Low-power consumption
  - < 5 mA @ 5V, 4 MHz
  - 100 µA typical @ 4.5V, 32 kHz
  - < 1 µA typical standby current @ 5V

PACKAGE TYPES

PDIP, Windowed CERDIP



PLCC



QFP

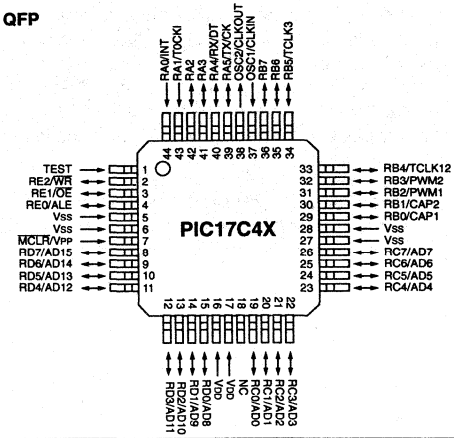


TABLE OF CONTENTS

To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. To this end, we recently converted to a new publishing software package which we believe will enhance our entire documentation process and product. As in any conversion process, information may have accidently been altered or deleted. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error from the previous version of the PIC17C42 Data Sheet (Literature Number DS30073D), please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

To assist you in the use of this document, Appendix C contains a list of new information in this data sheet, while Appendix D contains information that has changed

© 1995 Microchip Technology Inc.

1.0 OVERVIEW

This data sheet covers the PIC17C4X group of the PIC17CXX family of microcontrollers. The following devices are discussed in this data sheet:

- PIC17C42
- PIC17C43
- PIC17C44

The PIC17C43 and PIC17C44 devices include architectural enhancements over the PIC17C42. These enhancements will be discussed throughout this data sheet.

The PIC17C4X devices are 40-Pin, EPROM-based members of the versatile PIC17CXX family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers.

All PIC16/17 microcontrollers employ an advanced RISC-like architecture. The PIC17CXX has enhanced core features, sixteen-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 16-bit wide instruction word with a separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 55 instructions (reduced instruction set) are available in the PIC17C42 and 58 instructions in the PIC17C43 and PIC17C44 devices. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance. For mathematical intensive applications the PIC17C43 and PIC17C44 devices have a single cycle 8 x 8 Hardware Multiplier.

PIC17CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC17C4X devices have up to 454 bytes of RAM and 33 I/O pins. In addition, the PIC17C4X adds several peripheral features useful in many high performance applications including:

- Four timer/counters
- Two capture inputs
- Two PWM outputs
- A Serial Communications Interface (SCI)

The SCI can be configured for either synchronous or asynchronous communications (USART).

These special features reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LF oscillator is for low frequency crystals and minimizes power consumption, XT is a standard crystal, and the EC is for external clock input. The SLEEP (power-down) mode offers additional power saving. The user can wake up the chip from SLEEP through several external and internal interrupts and device resets.

There are four configuration options for the device operational modes:

- Microprocessor
- Microcontroller
- Extended microcontroller
- Protected microcontroller

The microprocessor and extended microcontroller modes allow up to 64K-words of external program memory.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software malfunction.

Table 1-1 lists the features of the PIC17C4X devices.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

A simplified block diagram of the PIC17C42 is shown in Figure 3-1 and the block diagram for the PIC17C43 and PIC17C44 devices is shown in Figure 3-2.

The PIC17C4X fits perfectly in applications ranging from precise motor control and industrial process control to automotive, instrumentation, and telecom applications. Other applications that require extremely fast execution of complex software programs or the flexibility of programming the software code as one of the last steps of the manufacturing process would also be well suited. The EPROM technology makes customization of application programs (with unique security codes, combinations, model numbers, parameter storage, etc.) fast and convenient. Small footprint package options make the PIC17C4X ideal for applications with space limitations that require high performance. High speed execution, powerful peripheral features, flexible I/O, and low power consumption all at low cost make the PIC17C4X ideal for a wide range of embedded control applications.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5x and PIC16Cxx families of microcontrollers will see the architectural enhancements that have been implemented. These enhancements allow the device to be more efficient in software and hardware requirements. Please refer to Appendix A for a detailed list of enhancements and modifications. Code written for PIC16C5X or PIC16CXX can be easily ported to PIC17CXX family of devices (see Appendix B).

1.2 Development Support

The PIC17CXX family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a universal programmer, a "C" compiler, and fuzzy logic support tools.

© 1995 Microchip Technology Inc.

TABLE 1-1: PIC17CXX FAMILY OF DEVICES

		PIC17C42	PIC17C43	PIC17C44
Maximum Frequency of Operation		25 MHz	25 MHz	25 MHz
Operating Voltage Range		4.5 - 5.5V	2.5 - 6.0V	2.5 - 6.0V
On-chip Program Memory (16-bits wide)		2K	4K	8K
Data Memory (bytes)		232	454	454
Hardware Multiplier (8 x 8)		No	Yes	Yes
Timer0 (16-bit + 8-bit postscaler)		Yes	Yes	Yes
Timer1 (8-bit)		Yes	Yes	Yes
Timer2 (8-bit)		Yes	Yes	Yes
Timer3 (16-bit)		Yes	Yes	Yes
Capture inputs (16-bit)		2	2	2
PWM outputs (up to 10-bit)		2	2	2
Serial Communications Interface (SCI/USART)		Yes	Yes	Yes
Power-On Reset		Yes	Yes	Yes
Watchdog Timer		Yes	Yes	Yes
External Interrupts		Yes	Yes	Yes
Interrupt Sources		11	11	11
Program Memory Code Protect		Yes	Yes [1]	Yes [1]
I/O		33	33	33
I/O High Current Capability	Source	25 mA	25 mA	25 mA
	Sink	25 mA[2]	25 mA[2]	25 mA[2]
Package Types		40-Pin DIP, 44-pin PLCC 44-pin MQFP	40-Pin DIP, 44-pin PLCC 44-pin TQFP	40-Pin DIP, 44-pin PLCC 44-pin TQFP

Note 1: The Code Protect Feature is different from the PIC17C42.
2: RA2 and RA3 can sink up to 60 mA.

© 1995 Microchip Technology Inc.

2.0 PIC17C4X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC17C4X Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC17C4X Product Identification System" on the back page of this data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PRO MATE™ programmer supports programming of the PIC17C4X. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP^{SM}) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC17C4X can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC17C4X uses a modified Harvard architecture. This architecture has the program and data accessed from separate memories. So the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture, where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. PIC17C4X opcodes are 16-bits wide, enabling single word instructions. The full 16-bit wide program memory bus fetches a 16-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions execute in a single cycle (160 ns @ 25 MHz), except for program branches and two special instructions that transfer data between program and data memory.

The PIC17C4X can address up to 64K x 16 of program memory space. The PIC17C42 integrates 2K x 16 EPROM program memory on-chip, while the PIC17C43 integrates 4K x 16 and the PIC17C44 integrates 8K x 16 EPROM program memory. Program execution can be internal only (microcontroller or protected mircocontroller mode), external only (microprocessor mode) or both (extended microcontroller mode).

The PIC17CXX can directly or indirectly address its register files or data memory. All special function registers, including the Program Counter (PC) and Working Register (WREG), are mapped in the data memory. The PIC17CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC17CXX simple yet efficient. In addition, the learning curve is reduced significantly.

One of the PIC17CXX family architectural enhancements from the PIC16CXX family allows two file registers to be used in some two operand instructions. This allows data to be moved directly between two registers without going through the WREG register. This increases performance and decreases program memory usage.

The PIC17CXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature.

The WREG register is an 8-bit working register used for ALU operations.

The PIC17C43 and PIC17C44 devices also have an 8 x 8 hardware multiplier. This multiplier generates a 16-bit result in a single cycle.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

Although the ALU does not perform signed arithmetic, the Overflow bit (OV) can be used to implement signed math. Signed arithmetic is comprised of a magnitude and a sign bit. The overflow bit indicates if the magnitude overflows and causes the sign bit to change state. Signed math can have greater than 7-bit values (magnitude), if more than one byte is used. The use of the overflow bit only operates on bit6 (MSb of magnitude) and bit7 (sign bit) of the value in the ALU. That is, the overflow bit is not useful if trying to implement signed math where the magnitude, for example, is 11-bits. If the signed math values are greater than 7-bits (15-, 24- or 31-bit), the algorithm must ensure that the low order bytes ignore the overflow status bit.

Care should be taken when adding and subtracting signed numbers to ensure that the correct operation is executed. Example 3-1 shows an item that must be taken into account when doing signed arithmetic on an ALU which operates as an unsigned machine.

EXAMPLE 3-1: SIGNED MATH

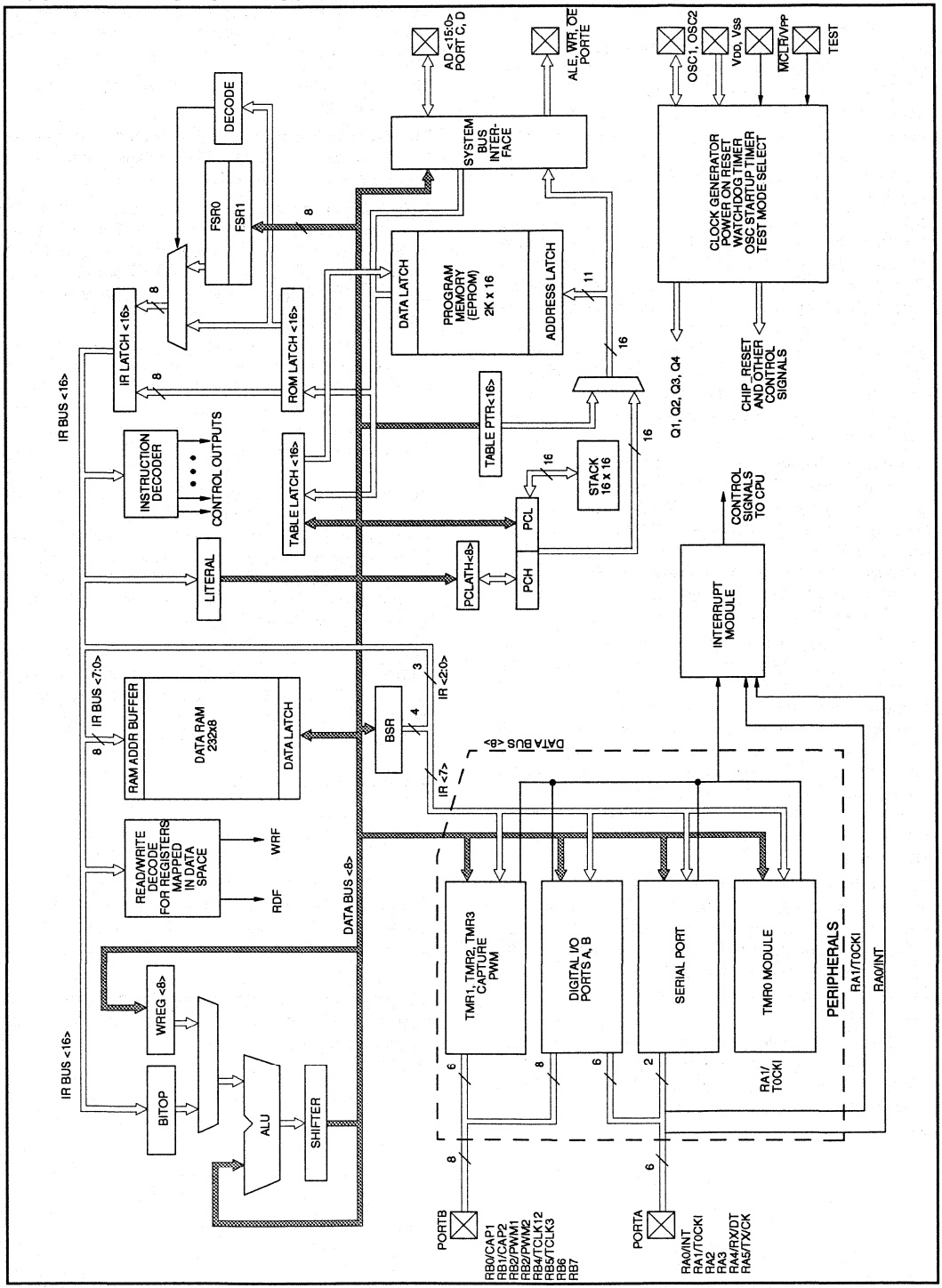
```
Signed Math          Unsigned Math
  -127   (FFh)          255   (FFh)
+    1   (01h)        +   1   (01h)
= -126   (FEh)        =   0   (00h) ; C = 1
```

Signed math requires the result in REG to be FEh (-126). This would be accomplished by subtracting one as opposed to adding one.

Simplified block diagrams are shown in Figure 3-1 and Figure 3-2. The descriptions of the device pins are listed in Table 3-1.

© 1995 Microchip Technology Inc.

FIGURE 3-1: PIC17C42 BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

FIGURE 3-2: PIC17C43 AND PIC17C44 BLOCK DIAGRAM

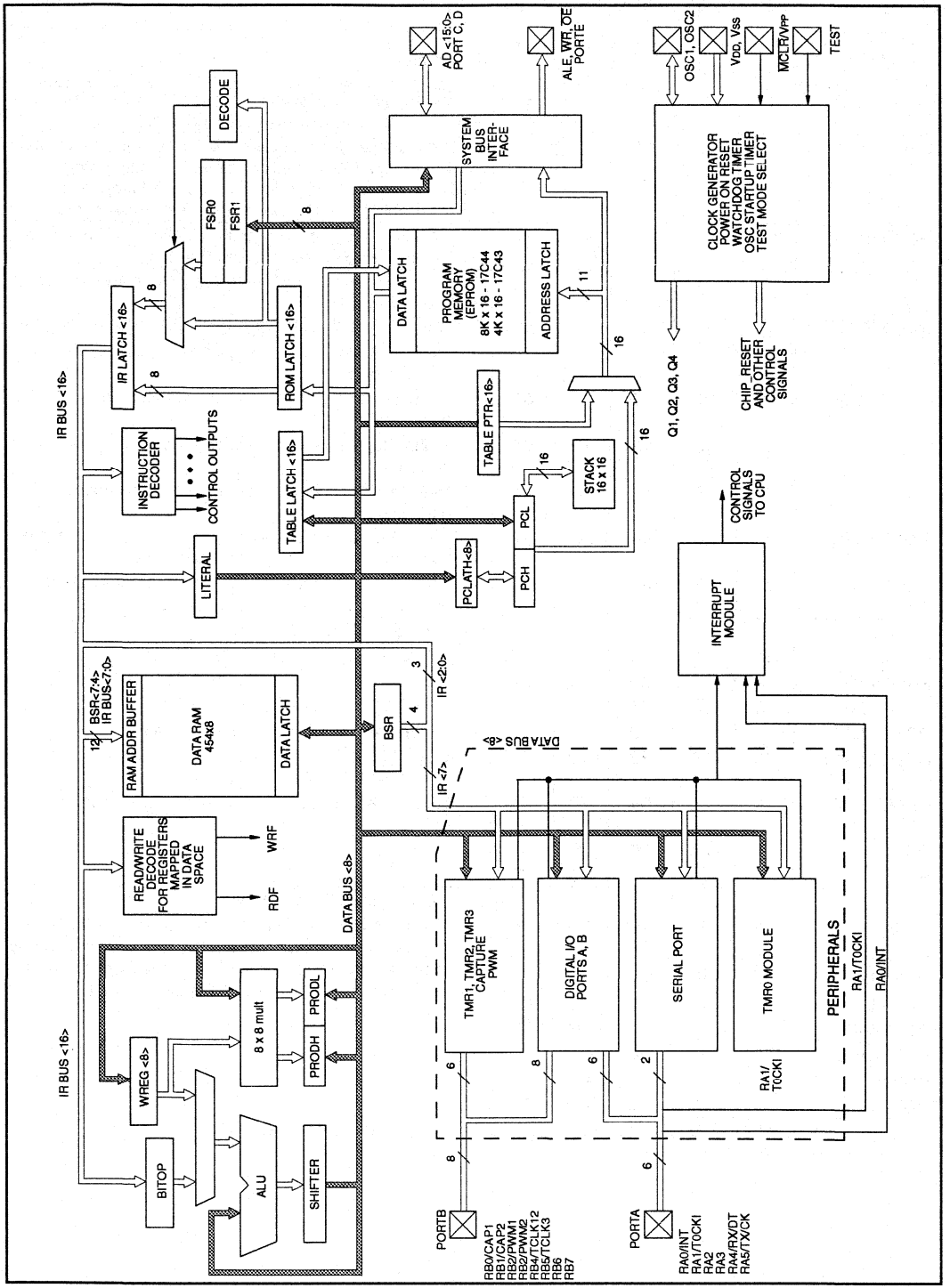


TABLE 3-1: PIC17C4X PINOUT DESCRIPTIONS

Name	DIP No.	PLCC No.	QFP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	19	21	37	I	ST	Oscillator input in crystal/resonator or RC oscillator mode. External clock input in external clock mode.
OSC2/CLKOUT	20	22	38	O	—	Oscillator output. Connects to crystal or resonator in crystal oscillator mode. In RC oscillator or external clock modes OSC2 pin outputs CLKOUT which has one fourth the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	32	35	7	I/P	ST	Master clear (reset) input/Programming Voltage (VPP) input. This is the active low reset input to the chip.
						PORTA is a bidirectional I/O Port except for RA0 and RA1 which are input only.
RA0/INT	26	28	44	I	ST	RA0/INT can also be selected as an external interrupt input. Interrupt can be configured to be on positive or negative edge.
RA1/T0CKI	25	27	43	I	ST	Can also be selected to be the clock input to the TMR0 timer/counter.
RA2	24	26	42	I/O	ST	High voltage, high current open collector input/output port pins.
RA3	23	25	41	I/O	ST	High voltage, high current open collector input/output port pins.
RA4/RX/DT	22	24	40	I/O	ST	RA4/RX/DT can also be selected as the SCI Asynchronous Receive or SCI Synchronous Data.
RA5/TX/CK	21	23	39	I/O	ST	RA5/TX/CK can also be selected as the SCI Asynchronous Transmit or SCI Synchronous Clock.
						PORTB is a bidirectional I/O Port.
RB0/CAP1	11	13	29	I/O	ST	RB0/CAP1 can also be the CAP1 input pin.
RB1/CAP2	12	14	30	I/O	ST	RB1/CAP2 can also be the CAP2 input pin.
RB2/PWM1	13	15	31	I/O	ST	RB2/PWM1 can also be the PWM1 output pin.
RB3/PWM2	14	16	32	I/O	ST	RB3/PWM2 can also be the PWM2 output pin.
RB4/TCLK12	15	17	33	I/O	ST	RB4/TCLK12 can also be the external clock input to Timer1 and Timer2.
RB5/TCLK3	16	18	34	I/O	ST	RB5/TCLK3 can also be the external clock input to Timer3.
RB6	17	19	35	I/O	ST	
RB7	18	20	36	I/O	ST	
						PORTC is a bidirectional I/O Port.
RC0/AD0	2	3	19	I/O	TTL	This is also the lower half of the 16 bit wide system bus in microprocessor mode or extended microcontroller mode. In multiplexed system bus configuration, these pins are address output as well as data input or output.
RC1/AD1	3	4	20	I/O	TTL	
RC2/AD2	4	5	21	I/O	TTL	
RC3/AD3	5	6	22	I/O	TTL	
RC4/AD4	6	7	23	I/O	TTL	
RC5/AD5	7	8	24	I/O	TTL	
RC6/AD6	8	9	25	I/O	TTL	
RC7/AD7	9	10	26	I/O	TTL	

Legend: I = Input only; O = Output only; I/O = Input/Output; P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input.

© 1995 Microchip Technology Inc.

TABLE 3-1: PIC17C4X PINOUT DESCRIPTIONS (CONT.)

Name	DIP No.	PLCC No.	QFP No.	I/O/P Type	Buffer Type	Description
						PORTD is a bidirectional I/O Port.
RD0/AD8	40	43	15	I/O	TTL	This is also the upper byte of the 16-bit system bus in microprocessor mode or extended microprocessor mode or extended microcontroller mode. In multiplexed system bus configuration these pins are address output as well as data input or output.
RD1/AD9	39	42	14	I/O	TTL	
RD2/AD10	38	41	13	I/O	TTL	
RD3/AD11	37	40	12	I/O	TTL	
RD4/AD12	36	39	11	I/O	TTL	
RD5/AD13	35	38	10	I/O	TTL	
RD6/AD14	34	37	9	I/O	TTL	
RD7/AD15	33	36	8	I/O	TTL	
						PORTE is a bidirectional I/O Port.
RE0/ALE	30	32	4	I/O	TTL	In microprocessor mode or extended microcontroller mode, it is the Address Latch Enable (ALE) output. Address should be latched on the falling edge of ALE output.
RE1/$\overline{OE}$	29	31	3	I/O	TTL	In microprocessor or extended microcontroller mode, it is the Output Enable ($\overline{OE}$) control output (active low).
RE2/$\overline{WR}$	28	30	2	I/O	TTL	In microprocessor or extended microcontroller mode, it is the Write Enable ($\overline{WR}$) control output (active low).
TEST	27	29	1	I	ST	Test mode selection control input. Always tie to Vss for normal operation.
Vss	10, 31	11, 12, 33, 34	5, 6, 27, 28	P		Ground reference for logic and I/O pins.
VDD	1	1, 44	16, 17	P		Positive supply for logic and I/O pins.

Legend: I = Input only; O = Output only; I/O = Input/Output; P = Power; — = Not Used; TTL = TTL input; ST = Schmitt Trigger input.

© 1995 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, and the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 3-3.

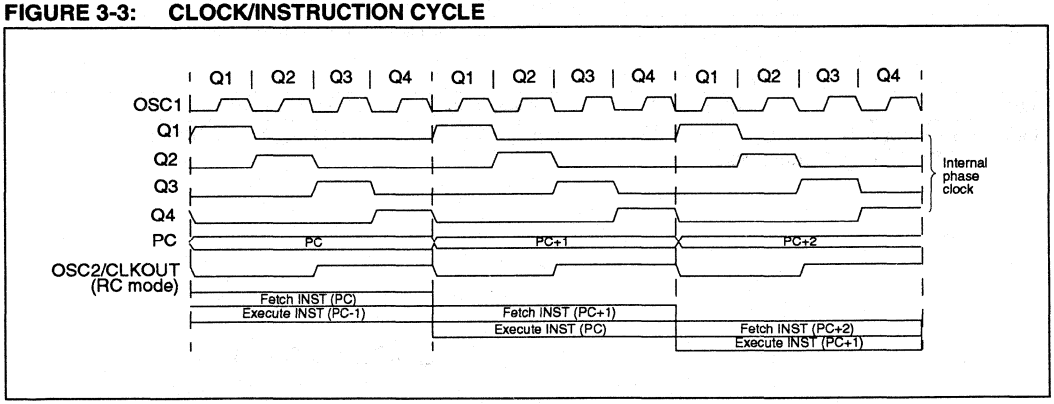
3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (see Example 3-2).

A fetch cycle begins with the program counter incrementing in Q1.

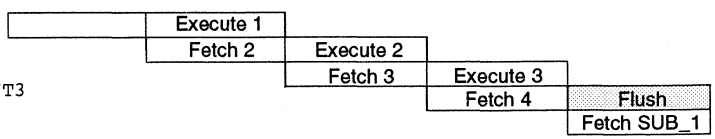
In the execution cycle, the fetched instruction is latched into the "Instruction Register (IR)" in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-3: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-2: INSTRUCTION PIPELINE FLOW

```
1. MOVLW  55h
2. MOVWF  PORTB
3. CALL   SUB_1
4. BSF    PORTA, BIT3
```



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

© 1995 Microchip Technology Inc.

4.0 RESET

The PIC17CXX differentiates between various kinds of reset:

- Power-On Reset (POR)
- $\overline{MCLR}$ reset during normal operation
- WDT time-out reset during normal operation

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-On Reset (POR), on $\overline{MCLR}$ or WDT reset during normal operation and on $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT reset during SLEEP, since this reset is viewed as the resumption of normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 4-3. These bits are used in software to determine the nature of reset. See Table 4-4 for a full description of reset states of all registers.

Note: While the device is in a reset state, the internal phase clock is held in the Q1 state. Any processor mode that allows external execution will force the RE0/ALE pin as a low output and the RE1/$\overline{OE}$ and RE2/$\overline{WR}$ pins as high outputs.

A simplified block diagram of the on-chip reset circuit is shown in Figure 4-1.

4.1 Power-On Reset (POR), Power-Up-Timer (PWRT) and Oscillator Start-Up Timer (OST)

4.1.1 POWER-ON RESET (POR)

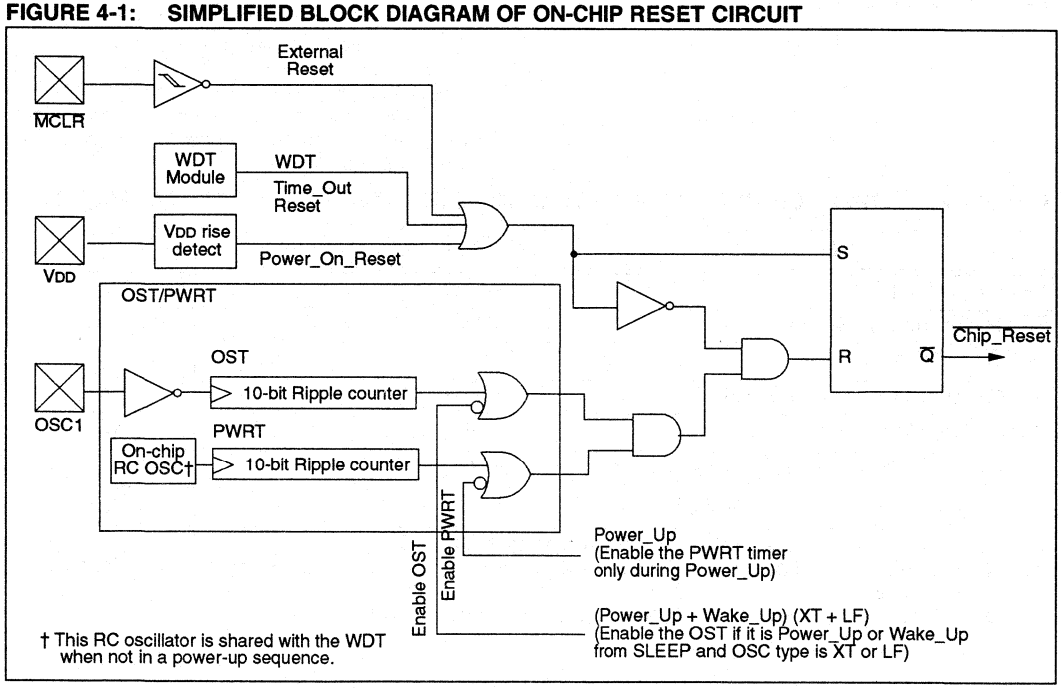
The Power-On Reset circuit holds the device in reset until VDD is above the trip point (in the range of 1.4V - 2.3V). PIC17C43 and PIC17C44 will produce an internal reset for both rising and falling VDD. The PIC17C42 does not produce an internal reset when VDD declines. To take advantage of the POR, just tie the $\overline{MCLR}$/VPP pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create Power-On Reset. A minimum rise time for VDD is required. See Electrical Specifications for details.

4.1.2 POWER-UP TIMER (PWRT)

The Power-Up Timer provides a fixed 96 ms time-out (nominal) on power-up. This occurs from rising edge of the POR signal and after the first rising edge of $\overline{MCLR}$ (detected high). The power-up timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. In most cases the PWRT delay allows the VDD to rise to an acceptable level.

The power-up time delay will vary from chip to chip and to VDD and temperature. See DC parameters for details.

FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



4.1.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (1024 TOSC) delay after $\overline{MCLR}$ is detected high or a wake-up from SLEEP event occurs.

The OST time-out is invoked only for XT and LF oscillator modes on a Power-On Reset or a Wake-Up from SLEEP.

The OST counts the oscillator pulses on the OSC1/CLKIN pin. The counter only starts incrementing after the amplitude of the signal reaches the oscillator input thresholds. This delay allows the crystal oscillator or resonator to stabilized before the device exits reset. The length of time-out is a function of the crystal/resonator frequency.

4.1.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First the internal POR signal goes high when the POR trip point is reached. If $\overline{MCLR}$ is high, then both the OST and PWRT timers start. In general the PWRT time-out is longer, except with low frequency crystals/resonators. The total time-out also varies based on oscillator configuration. Table 4-1 shows the times that are associated with the oscillator configuration. Figure 4-2 and Figure 4-3 display these time-out sequences.

If the device voltage is not within electrical specification at the end of a time-out, the $\overline{MCLR}$/VPP pin must be held low until the voltage is within the device specification. The use of an external RC delay is sufficient for many of these applications.

TABLE 4-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up	Wake up from SLEEP	$\overline{MCLR}$ Reset
XT, LF	Greater of (96 ms, 1024 TOSC)	1024 TOSC	—
EC, RC	Greater of (96 ms, 1024 TOSC)	—	—

The time-out sequence begins from the first rising edge of $\overline{MCLR}$.

Table 4-3 shows the reset conditions for some special registers, while Table 4-4 shows the initialization conditions for all the registers. The shaded registers (in Table 4-4) only exist for PIC17C43 and PIC17C44 devices. In the PIC17C42, the PRODH and PRODL registers are general purpose RAM.

TABLE 4-2: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{TO}$	$\overline{PD}$	Event
1	1	Power-On Reset, $\overline{MCLR}$ reset during normal operation, or CLRWDT instruction executed
1	0	$\overline{MCLR}$ reset during SLEEP or interrupt wake-up from SLEEP
0	1	WDT reset during normal operation
0	0	WDT time-out wake-up from SLEEP

In Figure 4-2, Figure 4-3 and Figure 4-4, TPWRT > TOST, as would be the case in higher frequency crystals. For lower frequency crystals, (i.e., 32 kHz) TOST would be greater.

TABLE 4-3: RESET CONDITION FOR THE PROGRAM COUNTER AND THE CPUSTA REGISTER

Event		PCH:PCL	CPUSTA	OST Active
Power-On Reset		0000h	--11 11--	Yes
$\overline{MCLR}$ reset during normal operation		0000h	--11 11--	No
$\overline{MCLR}$ reset during SLEEP		0000h	--11 10--	Yes [2]
WDT reset during normal operation		0000h	--11 01--	No
WDT during SLEEP		0000h	--11 00--	Yes [2]
Interrupt wake-up from SLEEP	GLINTD is set	PC + 1	--11 10--	Yes [2]
	GLINTD is clear	PC + 1 [1]	--10 10--	Yes [2]

Legend: u = unchanged, x = unknown, - = unimplemented, reads as '0'.

Note 1: On wake-up, this instruction is executed. The instruction at the appropriate interrupt vector is fetched and then executed.

2: The OST is only active when the Oscillator is configured for XT or LF modes.

© 1995 Microchip Technology Inc.

FIGURE 4-2: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

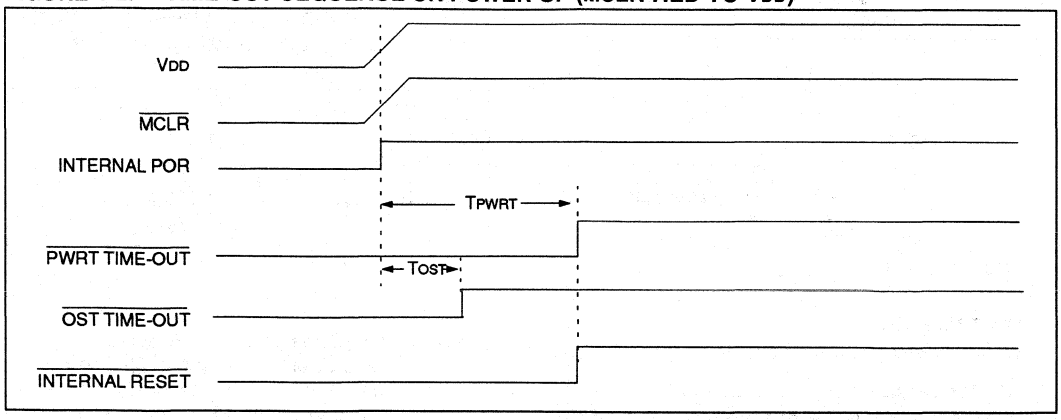


FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD)

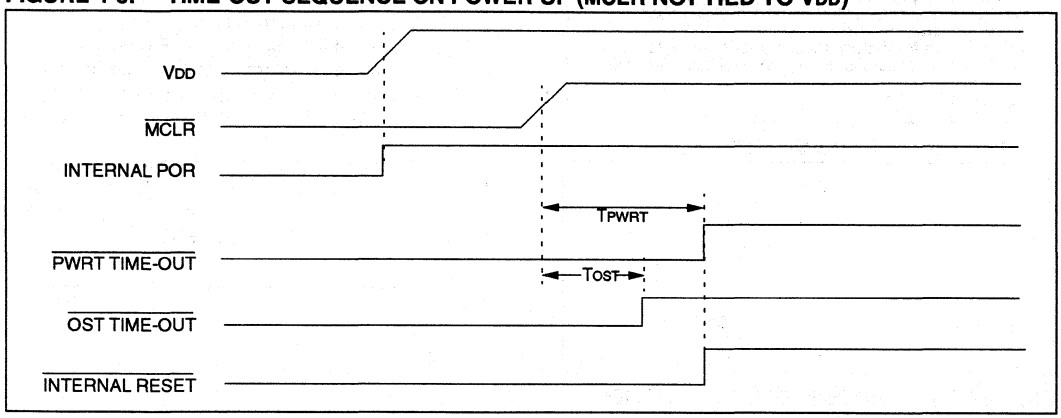
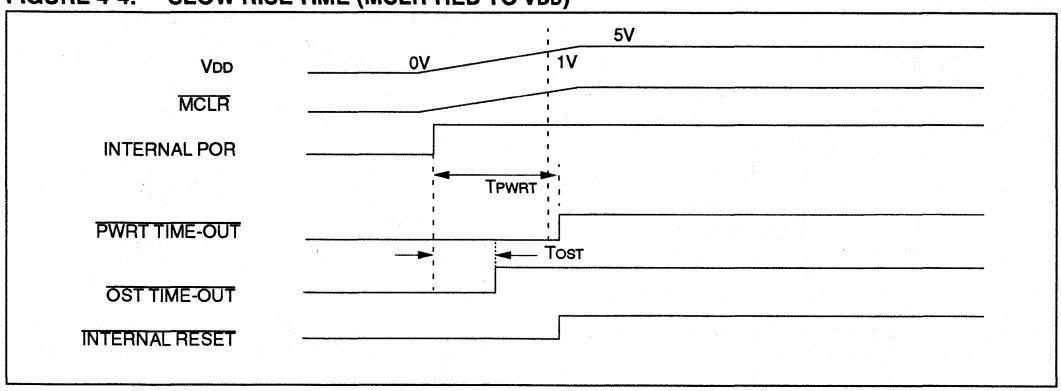
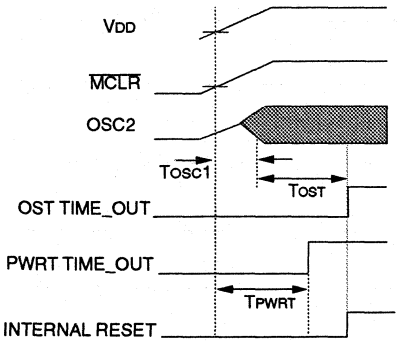


FIGURE 4-4: SLOW RISE TIME (MCLR TIED TO VDD)



© 1995 Microchip Technology Inc.

FIGURE 4-5: OSCILLATOR START-UP TIME



This figure shows in greater detail the timings involved with the oscillator start-up timer. In this example the low frequency crystal start-up time is larger than power-up time (TPWRT).
Tosc1 = time for the crystal oscillator to react to an oscillation level detectable by the Oscillator Start-Up Timer (ost)
TOST = 1024 Tosc

FIGURE 4-6: USING ON-CHIP POR

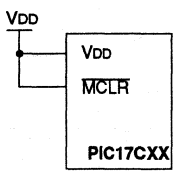
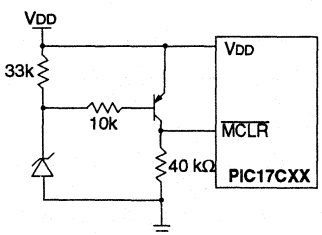
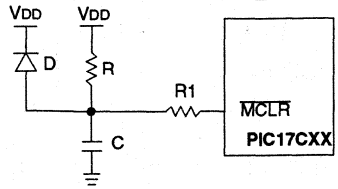


FIGURE 4-7: BROWN-OUT PROTECTION CIRCUIT 1



This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

FIGURE 4-8: PIC17C42 EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

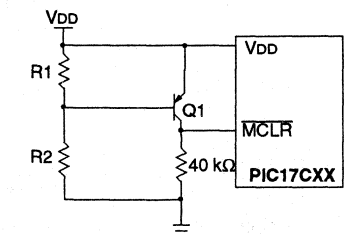


Note 1: An external Power-On Reset circuit is required only if VDD power-up time is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to ensure that the voltage drop across R does not exceed 0.2V (max. leakage current spec. on the MCLR/VPP pin is 5 µA). A larger voltage drop will degrade VIH level on the MCLR/VPP pin.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to ESD or EOS.

FIGURE 4-9: BROWN-OUT PROTECTION CIRCUIT 2



This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \cdot \frac{R1}{R1 + R2} = 0.7V.$$

© 1995 Microchip Technology Inc.

TABLE 4-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS

Register	Address	Power-On Reset	MCLR Reset WDT Reset	Wake up from SLEEP through interrupt
UNBANKED				
INDF0	00h	0000 0000	0000 0000	0000 0000
FSR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1[2]
PCLATH	03h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ALUSTA	04h	1111 xxxx	1111 uuuu	1111 uuuu
T0STA	05h	0000 000-	0000 000-	0000 000-
CPUSTA[3]	06h	--11 11--	--11 ??--	--uu ??--
INTSTA	07h	0000 0000	0000 0000	uuuu uuuu[1]
INDF1	08h	0000 0000	0000 0000	uuuu uuuu
FSR1	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	0Ah	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0L	0Bh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR0H	0Ch	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLPTRL	0Dh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLPTRH	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLPTRL[4]	0Dh	0000 0000	0000 0000	uuuu uuuu
TBLPTRH[4]	0Eh	0000 0000	0000 0000	uuuu uuuu
BSR	0Fh	0000 0000	0000 0000	uuuu uuuu
BANK 0				
PORTA	10h	0-xx xxxx	0-uu uuuu	uuuu uuuu
DDRB	11h	1111 1111	1111 1111	uuuu uuuu
PORTB	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
RCSTA	13h	0000 -00x	0000 -00u	uuuu -uuu
RCREG	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXSTA	15h	0000 --1x	0000 --1u	uuuu --uu
TXREG	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRG	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
BANK 1				
DDRC	10h	1111 1111	1111 1111	uuuu uuuu
PORTC	11h	xxxx xxxx	uuuu uuuu	uuuu uuuu
DDRD	12h	1111 1111	1111 1111	uuuu uuuu
PORTD	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
DDRE	14h	---- -111	---- -111	---- -uuu
PORTE	15h	---- -xxx	---- -uuu	---- -uuu
PIR	16h	0000 0010	0000 0010	uuuu uuuu[1]
PIE	17h	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented, reads as '0', ? = value depends on condition

Note 1: One or more bits in INTSTA, PIR will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.

3: See Table 4-3 for reset value of specific condition.

4: These values are for the PIC17C43 and PIC17C44 only.

© 1995 Microchip Technology Inc.

TABLE 4-4: INITIALIZATION CONDITIONS FOR SPECIAL FUNCTION REGISTERS (CONT.)

Register	Address	Power-On Reset	$\overline{\text{MCLR}}$ Reset WDT Reset	Wake up from SLEEP through interrupt
BANK 2				
TMR1	10h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR2	11h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3H	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR1	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR2	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR3/CA1L	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PR3/CA1H	17h	xxxx xxxx	uuuu uuuu	uuuu uuuu
BANK 3				
PW1DCL	10h	xx-- ----	uu-- ----	uu-- ----
PW2DCL	11h	xx-- ----	uu-- ----	uu-- ----
PW1DCH	12h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PW2DCH	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CA2L	14h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CA2H	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TCON1	16h	0000 0000	0000 0000	uuuu uuuu
TCON2	17h	0000 0000	0000 0000	uuuu uuuu
UNBANKED				
PRODL[4]	18h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODH[4]	19h	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented, reads as '0', ? = value depends on condition

Note 1: One or more bits in INTSTA, PIR will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GLINTD bit is cleared, the PC is loaded with the interrupt vector.

3: See Table 4-3 for reset value of specific condition.

4: These values are for the PIC17C43 and PIC17C44 only.

© 1995 Microchip Technology Inc.

5.0 INTERRUPTS

The PIC17C4X devices have 11 sources of interrupt:

- External interrupt from the RA0/INT pin
- Change on RB<7:0> pins
- Timer0 Overflow
- Timer1 Overflow
- Timer2 Overflow
- Timer3 Overflow
- SCI Transmit buffer empty
- SCI Receive buffer full
- Capture1
- Capture2
- T0CKI edge occurred

There are four registers used in the control and status of interrupts. These are:

- CPUSTA
- INTSTA
- PIE
- PIR

The CPUSTA register contains the GLINTD bit. This is the Global Interrupt Disable bit. When this bit is set, all interrupts are disabled. This bit is part of the controller core functionality and is described in the Memory Organization section.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with the interrupt vector address. There are four interrupt vectors. Each vector address is for a specific interrupt source (except the peripheral interrupts which have the same vector address). These sources are:

- External interrupt from the RA0/INT pin
- Timer0 Overflow
- T0CKI edge occurred
- Any peripheral interrupt

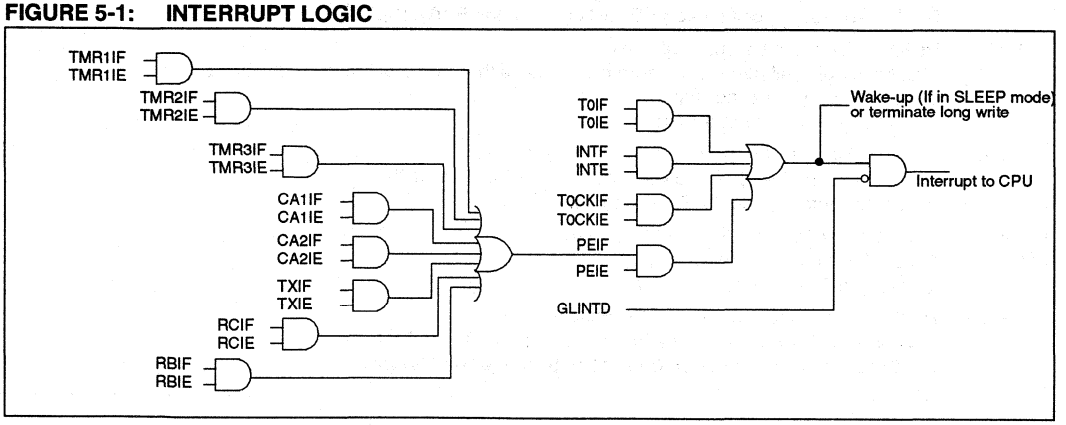
When program execution vectors to one of these interrupt vector addresses (except for the peripheral interrupt address), the interrupt flag bit is automatically cleared. Vectoring to the peripheral interrupt vector address does not automatically clear the source of the interrupt. In the peripheral interrupt service routine, the source(s) of the interrupt can be determined by testing the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

All of the individual interrupt flag bits will be set regardless of the status of their corresponding mask bit or the GLINTD bit.

For external interrupt events, there will be an interrupt latency. For two cycle instructions, the latency could be one instruction cycle longer.

The "return from interrupt" instruction, `RETFIE`, can be used to mark the end of the interrupt service routine. When this instruction is executed, the stack is "POPed", and the GLINTD bit is cleared (to re-enable interrupts).

FIGURE 5-1: INTERRUPT LOGIC



5.1 Interrupt Status Register (INTSTA)

The Interrupt Status/Control register (INTSTA) records the individual interrupt requests in flag bits, and contains the individual enable bits (not for the peripherals).

The PEIF bit is a read only, bit wise OR of all the peripheral flag bits in the PIR register (Figure 5-4).

Note: T0IF, INTF, T0CKIF, or PEIF will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled) or the GLINTD bit is set (all interrupts disabled).

FIGURE 5-2: INTSTA REGISTER (ADDRESS: 07H, UNBANKED)

R - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE
bit7							bit0

R = Readable bit
W = Writable bit
- n = Value at POR reset

bit 7: **PEIF**: Peripheral Interrupt Flag bit
This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits.
1 = A peripheral interrupt is pending
0 = No peripheral interrupt is pending

bit 6: **T0CKIF**: External Interrupt on T0CKI Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (18h).
1 = The software specified edge occurred on the RA1/T0CKI pin
0 = The software specified edge did not occur on the RA1/T0CKI pin

bit 5: **T0IF**: Timer0 Overflow Interrupt Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (10h).
1 = Timer0 overflowed
0 = Timer0 did not overflow

bit 4: **INTF**: External Interrupt on INT Pin Flag bit
This bit is cleared by hardware, when the interrupt logic forces program execution to vector (08h).
1 = The software specified edge occurred on the RA0/INT pin
0 = The software specified edge did not occur on the RA0/INT pin

bit 3: **PEIE**: Peripheral Interrupt Enable bit
This bit enables all peripheral interrupts that have their corresponding enable bits set.
1 = Enable peripheral interrupts
0 = Disable peripheral interrupts

bit 2: **T0CKIE**: External Interrupt on T0CKI Pin Enable bit
1 = Enable software specified edge interrupt on the RA1/T0CKI pin
0 = Disable interrupt on the RA1/T0CKI pin

bit 1: **T0IE**: Timer0 Overflow Interrupt Enable bit
1 = Enable Timer0 overflow interrupt
0 = Disable Timer0 overflow interrupt

bit 0: **INTE**: External Interrupt On INT Pin Enable bit
1 = Enable software specified edge interrupt on the RA0/INT pin
0 = Disable software specified edge interrupt on the RA0/INT pin

© 1995 Microchip Technology Inc.

5.2 Peripheral Interrupt Enable Register (PIE)

This register contains the individual flag bits for the Peripheral interrupts.

FIGURE 5-3: PIE REGISTER (ADDRESS: 17H, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **RBIE**: PORTB Interrupt on Change Enable bit
1 = Enable PORTB interrupt on change
0 = Disable PORTB interrupt on change

bit 6: **TMR3IE**: Timer3 Interrupt Enable bit
1 = Enable Timer3 interrupt
0 = Disable Timer3 interrupt

bit 5: **TMR2IE**: Timer2 Interrupt Enable bit
1 = Enable Timer2 interrupt
0 = Disable Timer2 interrupt

bit 4: **TMR1IE**: Timer1 Interrupt Enable bit
1 = Enable Timer1 interrupt
0 = Disable Timer1 interrupt

bit 3: **CA2IE**: Capture2 Interrupt Enable bit
1 = Enable Capture interrupt on RB1/CAP2 pin
0 = Disable Capture interrupt on RB1/CAP2 pin

bit 2: **CA1IE**: Capture1 Interrupt Enable bit
1 = Enable Capture interrupt on RB2/CAP1 pin
0 = Disable Capture interrupt on RB2/CAP1 pin

bit 1: **TXIE**: SCI Transmit Interrupt Enable bit
1 = Enable Transmit buffer empty interrupt
0 = Disable Transmit buffer empty interrupt

bit 0: **RCIE**: SCI Receive Interrupt Enable bit
1 = Enable Receive buffer full interrupt
0 = Disable Receive buffer full interrupt

© 1995 Microchip Technology Inc.

5.3 Peripheral Interrupt Request Register (PIR)

This register contains the individual flag bits for the peripheral interrupts.

Note: These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral interrupt service routine.

FIGURE 5-4: PIR REGISTER (ADDRESS: 16H, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R - 1	R - 0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **RBIF**: PORTB Interrupt on Change Flag bit
1 = One of the PORTB inputs changed (Software must end the mismatch condition)
0 = None of the PORTB inputs have changed

bit 6: **TMR3IF**: Timer3 Interrupt Flag bit
If Capture1 is enabled (CA1/$\overline{PR3}$ = 1)
1 = Timer3 overflowed
0 = Timer3 did not overflow

If Capture1 is disabled (CA1/$\overline{PR3}$ = 0)
1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value
0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value

bit 5: **TMR2IF**: Timer2 Interrupt Flag bit
1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value
0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value

bit 4: **TMR1IF**: Timer1 Interrupt Flag bit
If Timer1 is in 8-bit mode (T16 = 0)
1 = Timer1 value has rolled over to 0000h from equalling the period register (PR) value
0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value

If Timer1 is in 16-bit mode (T16 = 1)
1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value
0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value

bit 3: **CA2IF**: Capture2 Interrupt Flag bit
1 = Capture event occurred on RB1/CAP2 pin
0 = Capture event did not occur on RB1/CAP2 pin

bit 2: **CA1IF**: Capture1 Interrupt Flag bit
1 = Capture event occurred on RB0/CAP1 pin
0 = Capture event did not occur on RB0/CAP1 pin

bit 1: **TXIF**: SCI Transmit Interrupt Flag bit
1 = Transmit buffer is empty
0 = Transmit buffer is full

bit 0: **RCIF**: SCI Receive Interrupt Flag bit
1 = Receive buffer is full
0 = Receive buffer is empty

© 1995 Microchip Technology Inc.

5.4 Interrupt Operation

Global Interrupt Disable bit, GLINTD (CPUSTA<4>), enables all unmasked interrupts (if clear) or disables all interrupts (if set). Individual interrupts can be disabled through their corresponding enable bits in the INTSTA register. Peripheral interrupts need either the global peripheral enable PEIE bit disabled, or the specific peripheral enable bit disabled. Disabling the peripherals via the global peripheral enable bit, disables all peripheral interrupts. GLINTD is set on reset (interrupts disabled).

The RETFIE instruction allows returning from interrupt and re-enable interrupts at the same time.

When an interrupt is responded to, the GLINTD bit is automatically set to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with interrupt vector. There are four interrupt vectors to reduce interrupt latency.

The peripheral interrupt vector has multiple interrupt sources. Once in the peripheral interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The peripheral interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid continuous interrupts.

The PIC17C4X devices have 4 interrupt vectors. These vectors and their hardware priority are shown in Table 5-1. If two enabled interrupts occur "at the same time", the interrupt of the highest priority will be serviced first. This means that the vector address of that interrupt will be loaded into the program counter (PC).

TABLE 5-1: INTERRUPT VECTORS/ PRIORITIES

Address	Vector	Priority
0008h	External Interrupt on RA0/ INT pin (INTF)	1 (Highest)
0010h	TMR0 overflow interrupt (T0IF)	2
0018h	External Interrupt on T0CKI (T0CKIF)	3
0020h	Peripherals (PEIF)	4 (Lowest)

Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GLINTD bit.

Note 2: For the PIC17C42 only:
If an interrupt occurs while the Global Interrupt Disable (GLINTD) bit is being set, the GLINTD bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

1. An interrupt occurs simultaneously with an instruction that sets the GLINTD bit.
2. The program branches to the interrupt vector and executes the Interrupt Service Routine.
3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GLINTD bit to be cleared (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.

The method to ensure that interrupts are globally disabled is:

1. Ensure that the GLINTD bit was set by the instruction, as shown in the following code:

```
LOOP   BCF    CPUSTA, GLINTD ; Disable Global
                              ; Interrupts
       BTFSC  CPUSTA, GLINTD ; Global Interrupts
                              ; Disabled?
       GOTO   LOOP           ; NO, try again
                              ; YES, continue
                              ; with program
                              ; low
```

5.5 INT Interrupt

The external interrupt on the RA0/INT pin is edge triggered. Either the rising edge, if INTEDG bit (T0STA<7>) is set, or the falling edge, if INTEDG bit is clear. When a valid edge appears on the RA0/INT pin, the INTF bit (INTSTA<4>) is set. This interrupt can be disabled by clearing the INTE control bit (INTSTA<0>). The INT interrupt can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

5.6 TMR0 Interrupt

An overflow (FFFFh → 0000h) in TMR0 will set the T0IF (INTSTA<5>) bit. The interrupt can be enabled/disabled by setting/clearing the T0IE control bit (INTSTA<1>). For operation of the TMR0 module, see Section 11.0.

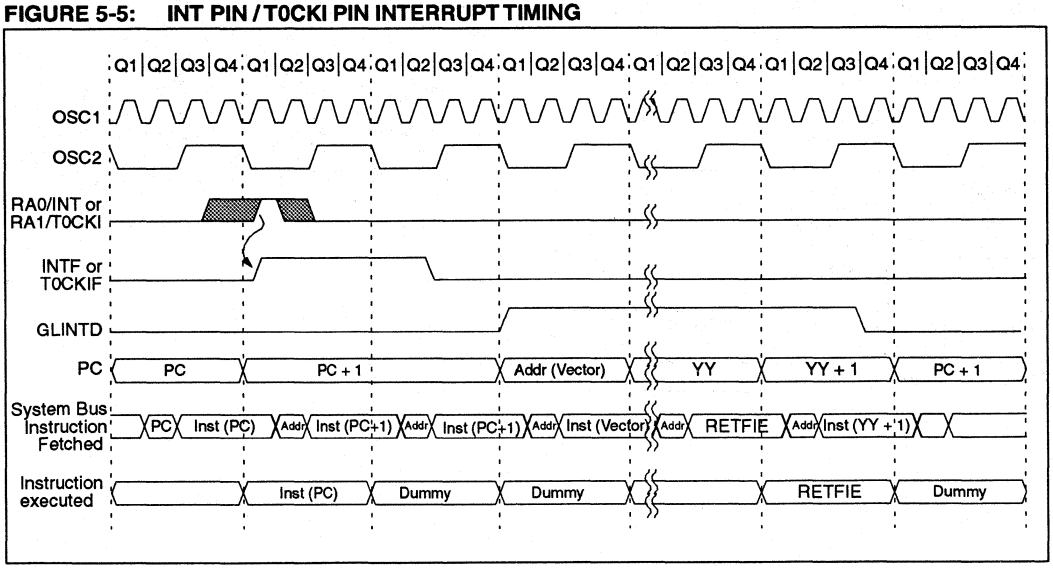
5.7 T0CKI Interrupt

The external interrupt on the RA1/T0CKI pin is edge triggered. Either the rising edge, if the T0SE bit (T0STA<6>) is set, or the falling edge, if the T0SE bit is clear. When a valid edge appears on the RA1/T0CKI pin, the T0CKIF bit (INTSTA<6>) is set. This interrupt can be disabled by clearing the T0CKIE control bit (INTSTA<2>). The T0CKI interrupt can wake up the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

5.8 Peripheral Interrupt

The peripheral interrupt flag indicates that at least one of the peripheral interrupts occurred (PEIF is set). The PEIF bit is a read only bit, and is a bit wise OR of all the flag bits in the PIR register AND'ed with the corresponding enable bits in the PIE register. Some of the peripheral interrupts can wake the processor from SLEEP. See Section 14.4 for details on SLEEP operation.

FIGURE 5-5: INT PIN / T0CKI PIN INTERRUPT TIMING



© 1995 Microchip Technology Inc.

5.9 Context Saving During Interrupts

During an interrupt, only the returned PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt; e.g. WREG, ALUSTA and the BSR registers. This requires implementation in software.

Example 5-1 shows the saving and restoring of information for an interrupt service routine. The PUSH and POP routines could either be in each interrupt service routine or could be subroutines that were called. Depending on the application, other registers may also need to be saved, such as PCLATH.

EXAMPLE 5-1: SAVING STATUS AND WREG IN RAM

```
;
; The addresses that are used to store the CPUSTA and WREG values
; must be in the data memory address range of 18h - 1Fh. Up to
; 8 locations can be saved and restored using
; the MOVFP instruction. This instruction neither affects the status
; bits, nor corrupts the WREG register.
;
;
PUSH    MOVFP   WREG, TEMP_W          ; Save WREG
        MOVFP   ALUSTA, TEMP_STATUS   ; Save ALUSTA
        MOVFP   BSR, TEMP_BSR         ; Save BSR

ISR     :                             ; This is the interrupt service routine
        :
POP     MOVFP   TEMP_W, WREG          ; Restore WREG
        MOVFP   TEMP_STATUS, ALUSTA   ; Restore ALUSTA
        MOVFP   TEMP_BSR, BSR         ; Restore BSR
        RETFIE                        ; Return from Interrupts enabled
```

© 1995 Microchip Technology Inc.

Notes:

© 1995 Microchip Technology Inc.

6.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC17C4X; program memory and data memory. Each block has its own bus, so that access to each block can occur during the same oscillator cycle.

The data memory can further be broken down into General Purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the "core" are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

6.1 Program Memory Organization

PIC17C4X devices have a 16-bit program counter capable of addressing a 64K x 16 program memory space. The reset vector is at 0000h and the interrupt vectors are at 0008h, 0010h, 0018h, and 0020h (Figure 6-1).

6.1.1 PROGRAM MEMORY OPERATION

The PIC17C4X can operate in one of four possible program memory configurations. The configuration is selected by two configuration bits. The possible modes are:

- Microprocessor
- Microcontroller
- Extended Microcontroller
- Protected Microcontroller

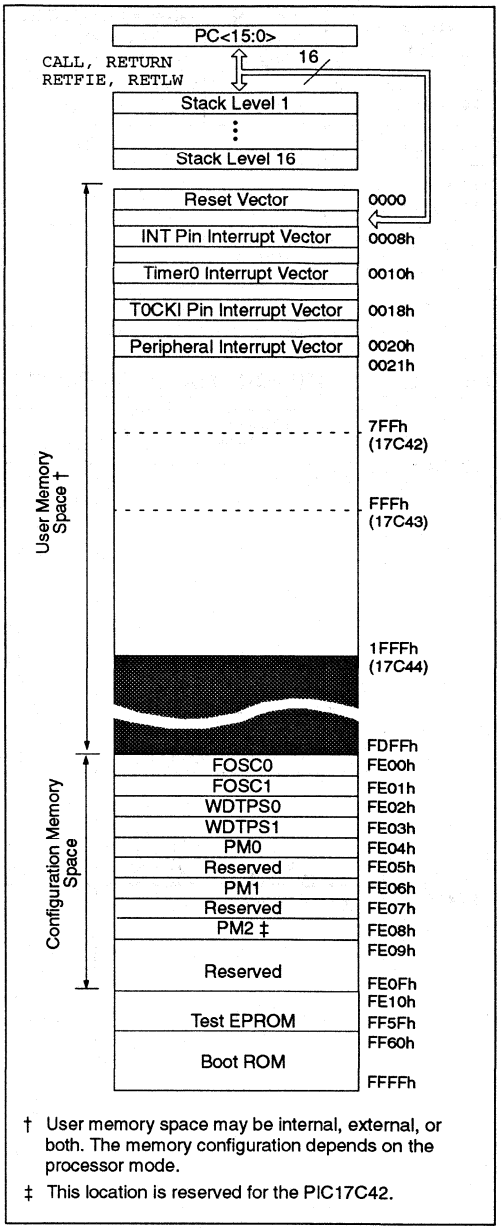
The microcontroller and protected microcontroller modes only allow internal execution. Any access beyond the program memory reads unknown data. The protected microcontroller mode also enables the code protection feature.

The extended microcontroller mode accesses both the internal program memory as well as external program memory. Execution automatically switches between internal and external memory. The 16-bits of address allow a program memory range of 64K-words.

The microprocessor mode only accesses the external program memory. The on-chip program memory is ignored. The 16-bits of address allow a program memory range of 64K-words. Microprocessor mode is the default mode of an unprogrammed device.

The different modes allow different access to the configuration bits, test memory, and boot ROM. Table 6-1 lists which modes can access which areas in memory. Test Memory and Boot Memory are not required for normal operation of the device. Care should be taken to ensure that no unintended branches occur to these areas.

FIGURE 6-1: PROGRAM MEMORY MAP AND STACK



† User memory space may be internal, external, or both. The memory configuration depends on the processor mode.

‡ This location is reserved for the PIC17C42.

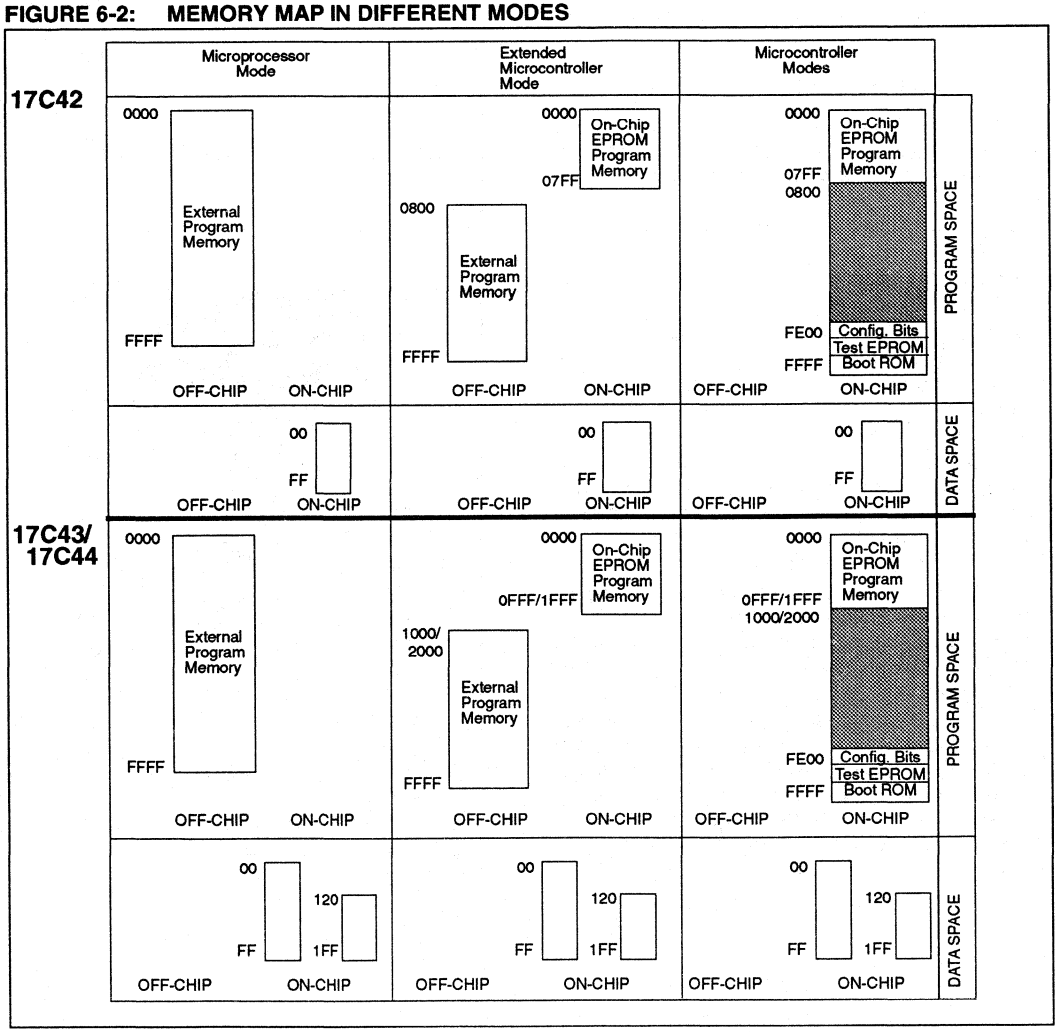
TABLE 6-1: MODE MEMORY ACCESS

Operating Mode	Internal Program Memory	Configuration Bits, Test Memory, Boot ROM
Microprocessor	No Access	No Access
Microcontroller	Access	Access
Extended Microcontroller	Access	No Access
Protected Microcontroller	Access	Access

The PIC17C4X can operate in modes where the program memory is off-chip. They are the microprocessor and extended microcontroller modes. The microprocessor mode is the default for an unprogrammed device.

Regardless of the processor mode, data memory is always on-chip.

FIGURE 6-2: MEMORY MAP IN DIFFERENT MODES

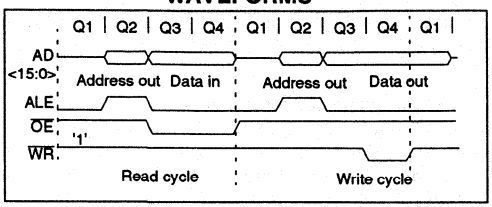


© 1995 Microchip Technology Inc.

6.1.2 EXTERNAL MEMORY INTERFACE

When either microprocessor or extended microcontroller mode is selected, PORTC, PORTD and PORTE are configured as the system bus. PORTC and PORTD are the multiplexed address/data bus and PORTE is for the control signals. External components are needed to demultiplex the address and data. This can be done as shown in Figure 6-4. The waveforms of address and data are shown in Figure 6-3. For complete timings, please refer to the electrical specification section.

FIGURE 6-3: EXTERNAL PROGRAM MEMORY ACCESS WAVEFORMS



As the speed of the processor increases, external EPROM memory with faster access time must be used. Table 6-2 lists external memory speed requirements for a given PIC17C4X device frequency.

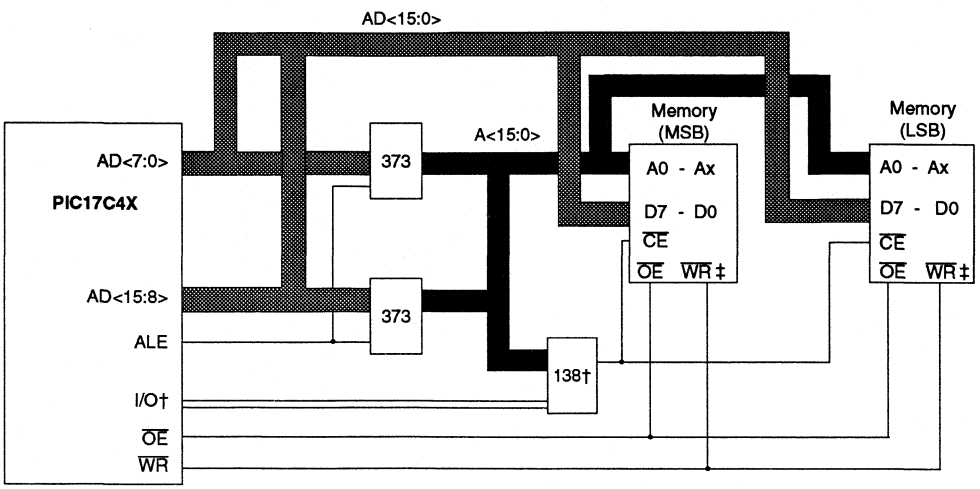
In extended microcontroller mode, when the device is executing out of internal memory, the control signals will continue to be active. That is, they indicate the action that is occurring in the internal memory. The external memory access is ignored.

TABLE 6-2: EPROM MEMORY ACCESS TIME ORDERING SUFFIX †

PIC17C4X Oscillator Frequency	Instruction Cycle Time (Tcy)	EPROM Suffix	
		PIC17C42	PIC17C43 PIC17C44
8 MHz	500 ns	-25	-25
16 MHz	250 ns	-12	-15
20 MHz	200 ns	-90	-10
25 MHz	160 ns	N.A.	-70

† This selection is for use with Microchip EPROMs. For interfacing to other manufacturers memory, please refer to the electrical specifications of the desired PIC17C4X device, as well as the desired memory device to ensure compatibility.

FIGURE 6-4: TYPICAL EXTERNAL PROGRAM MEMORY CONNECTION DIAGRAM



† Use of I/O pins is only required for paged memory.
‡ This signal is unused for ROM and EPROM devices.

6.2 Data Memory Organization

Data memory is partitioned into two areas. The first is the General Purpose Registers (GPR) area, while the second is the Special Function Registers (SFR) area. The SFRs control the operation of the device.

Portions of data memory are banked, this is for both areas. The GPR area is banked to allow greater than 232 bytes of general purpose RAM. SFRs are for the registers that control the peripheral functions. Banking requires the use of control bits for bank selection. These control bits are located in the Bank Select Register (BSR). If an access is made to a location outside this banked region, the BSR bits are ignored. Figure 6-5 shows the data memory map organization for the PIC17C42 and Figure 6-6 for the PIC17C43 and PIC17C44 devices.

Instructions `MOVPF` and `MOVFP` provide the means to move values from the peripheral area ("P") to any location in the register file ("F"), and vice-versa. The definition of the "P" range is from 0h to 1Fh, while the "F" range is 0h to FFh. The "P" range has 8 more locations than peripheral registers (6 locations for the PIC17C43 and PIC17C44 devices)which can be used as General Purpose Registers. This can be useful in some applications where variables need to be copied to other locations in the general purpose RAM (such as saving status information during an interrupt).

The entire data memory can be accessed either directly or indirectly through file select registers FSR0 and FSR1 (see Section 6.4). Indirect addressing uses the appropriate control bits of the BSR for accesses into the banked areas of data memory. The BSR is explained in greater detail in Section 6.8.

6.2.1 GENERAL PURPOSE REGISTER (GPR)

All devices have some amount of GPR area. The GPRs are 8-bits wide. When the GPR area is greater than 232, it must be banked to allow access to the additional memory space.

Only the PIC17C43 and PIC17C44 devices have banked memory in the GPR area. To facilitate switching between these banks, the `MOVLR bank` instruction has been added to the instruction set. GPRs are not initialized by a POR reset and are unchanged on all other resets.

6.2.2 SPECIAL FUNCTION REGISTERS (SFR)

The SFRs are used by the CPU and peripheral functions to control the operation of the device (see Figure 6-5 and Figure 6-6). These registers are static RAM.

The SFRs can be classified into two sets, those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described here, while those related to a peripheral feature are described in the section for each peripheral feature.

The peripheral registers are in the banked portion of memory, while the core registers are in the unbanked region. To facilitate switching between the peripheral banks, the `MOVLB bank` instruction has been provided.

© 1995 Microchip Technology Inc.

FIGURE 6-5: PIC17C42 REGISTER FILE MAP

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	**Bank 1** [1]	**Bank 2** [1]	**Bank 3** [1]
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h 1Fh	General Purpose RAM			
20h FFh				

Note 1: SFR file locations 10h - 17h are banked. All other SFRs ignore the Bank Select Register (BSR) bits.

FIGURE 6-6: PIC17C43 AND PIC17C44 REGISTER FILE MAP

Addr	Unbanked			
00h	INDF0			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	T0STA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	**Bank 1** [1]	**Bank 2** [1]	**Bank 3** [1]
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h	PRODL			
19h	PRODH			
1Ah 1Fh	General Purpose RAM [2]			
20h FFh		General Purpose RAM [2]		

Note 1: SFR file locations 10h - 17h are banked. All other SFRs ignore the Bank Select Register (BSR) bits.

2: General Purpose Registers (GPR) locations 20h - FFh and 120h - 1FFh are banked. All other GPRs ignore the Bank Select Register (BSR) bits.

© 1995 Microchip Technology Inc.

TABLE 6-3: SPECIAL FUNCTION REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets [3]
Unbanked											
00h	INDF0	Uses contents of FSR0 to address data memory (not a physical register)								---- ----	---- ----
01h	FSR0	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8-bits of PC								0000 0000	0000 0000
03h	PCLATH [1]	Holding register for upper 8-bits of PC								xxxx xxxx	uuuu uuuu
04h	ALUSTA	FS3	FS2	FS1	FS0	OV	Z	DC	C	1111 xxxx	1111 uuuu
05h	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—	0000 000-	0000 000-
06h	CPUSTA [2]	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 ??--
07h	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
08h	INDF1	Uses contents of FSR1 to address data memory (not a physical register)								---- ----	---- ----
09h	FSR1	Indirect data memory address pointer 1								xxxx xxxx	uuuu uuuu
0Ah	WREG	Working register								xxxx xxxx	uuuu uuuu
0Bh	TMR0L	Timer0 low byte								xxxx xxxx	uuuu uuuu
0Ch	TMR0H	Timer0 high byte								xxxx xxxx	uuuu uuuu
0Dh	TBLPTRL	Low byte of program memory table pointer								Note 4	Note 4
0Eh	TBLPTRH	High byte of program memory table pointer								Note 4	Note 4
0Fh	BSR	Bank select register								0000 0000	0000 0000
Bank 0											
10h	PORTA	$\overline{RBPU}$	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
11h	DDRB	Data direction register for PORTB								1111 1111	1111 1111
12h	PORTB	PORTB data latch								xxxx xxxx	uuuu uuuu
13h	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
14h	RCREG	Serial port receive register								xxxx xxxx	uuuu uuuu
15h	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
16h	TXREG	Serial port transmit register								xxxx xxxx	uuuu uuuu
17h	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu
Bank 1											
10h	DDRC	Data direction register for PORTC								1111 1111	1111 1111
11h	PORTC	RC7/ AD7	RC6/ AD6	RC5/ AD5	RC4/ AD4	RC3/ AD3	RC2/ AD2	RC1/ AD1	RC0/ AD0	xxxx xxxx	uuuu uuuu
12h	DDRD	Data direction register for PORTD								1111 1111	1111 1111
13h	PORTD	RD7/ AD15	RD6/ AD14	RD5/ AD13	RD4/ AD12	RD3/ AD11	RD2/ AD10	RD1/ AD9	RD0/ AD8	xxxx xxxx	uuuu uuuu
14h	DDRE	Data direction register for PORTE								---- -111	---- -111
15h	PORTE	—	—	—	—	—	RE2/$\overline{WR}$	RE1/$\overline{OE}$	RE0/ALE	---- -xxx	---- -uuu
16h	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000

Legend: x = Unknown, u = Unchanged, - = Unimplemented, reads as '0', ? - Value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated from or transferred to the upper byte of the program counter.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in CPUSTA are not affected by a $\overline{MCLR}$ reset.

3: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.

4: The following values are for both TBLPTRL and TBLPTRH:
PIC17C42 (Power-On Reset xxxx xxxx) and (All other resets uuuu uuuu)
PIC17C43/PIC17C44 (Power-On Reset 0000 0000) and (All other resets 0000 0000)

© 1995 Microchip Technology Inc.

TABLE 6-3: SPECIAL FUNCTION REGISTERS (CONT.)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets [3]
Bank 2											
10h	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h	TMR2	Timer2								xxxx xxxx	uuuu uuuu
12h	TMR3L	Timer3 low byte								xxxx xxxx	uuuu uuuu
13h	TMR3H	Timer3 high byte								xxxx xxxx	uuuu uuuu
14h	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
16h	PR3L/CA1L	Timer3 period register, low byte/capture1 register, low byte								xxxx xxxx	uuuu uuuu
17h	PR3H/CA1H	Timer3 period register, high byte/capture1 register, high byte								xxxx xxxx	uuuu uuuu
Bank 3											
10h	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu
16h	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
Unbanked											
18h [4]	PRODL	Low Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu
19h [4]	PRODH	High Byte of 16-bit Product (8 x 8 Hardware Multiply)								xxxx xxxx	uuuu uuuu

Legend: x = Unknown, u = Unchanged, - = Unimplemented, reads as '0', ? - Value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<15:8> whose contents are updated from or transferred to the upper byte of the program counter.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in CPUSTA are not affected by a $\overline{MCLR}$ reset.

3: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.

4: These values are for the PIC17C43 and the PIC17C44 only.

6.2.2.1 ALU STATUS REGISTER (ALUSTA)

The ALUSTA register contains the status bits of the Arithmetic and Logic Unit and the mode control bits for the indirect addressing register.

As with all the other registers, the ALUSTA register can be the destination for any instruction. If the ALUSTA register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the ALUSTA register as destination may be different than intended.

For example, `CLRF ALUSTA` will clear the upper four bits and set the Z bit. This leaves the status register as `0000u1uu` (where `u` = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions be used to alter the status registers because these instructions do not affect any status bit. To see how other instructions affect the status bits, see the "Instruction Set Summary".

Note: The C and DC bits operate as a borrow out bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

Arithmetic and Logic Unit (ALU) is capable of carrying out arithmetic or logical operations on two operands or a single operand. All single operand instructions operate either on the WREG register or a file register. For two operand instructions, one of the operands is the WREG register and the other one is either a file register or an 8-bit immediate constant.

FIGURE 6-7: ALUSTA REGISTER (ADDRESS: 04H, UNBANKED)

R/W - 1	R/W - 1	R/W - 1	R/W - 1	R/W - x	R/W - x	R/W - x	R/W - x
FS3	FS2	FS1	FS0	OV	Z	DC	C
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset
(x = unknown)

bit 7-6: **FS3:FS2**: FSR1 Mode Select bits
00 = Post auto-decrement FSR1 value
01 = Post auto-increment FSR1 value
1x = FSR1 value does not change

bit 5-4: **FS1:FS0**: FSR0 Mode Select bits
00 = Post auto-decrement FSR0 value
01 = Post auto-increment FSR0 value
1x = FSR0 value does not change

bit 3: **OV**: Overflow bit
This bit is used for signed arithmetic. It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit7) to change state.
1 = Overflow occurred for signed arithmetic, (in this arithmetic operation)
0 = No overflow occurred

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The results of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{\text{borrow}}$ bit
For `ADDWF` and `ADDLW` instructions.
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result
Note: For borrow the polarity is reversed.

bit 0: **C**: carry/$\overline{\text{borrow}}$ bit
For `ADDWF` and `ADDLW` instructions.
1 = A carry-out from the most significant bit of the result occurred
Note that a subtraction is executed by adding the two's complement of the second operand. For rotate (`RRCF`, `RLCF`) instructions, this bit is loaded with either the high or low order bit of the source register.
0 = No carry-out from the most significant bit of the result
Note: For borrow the polarity is reversed.

© 1995 Microchip Technology Inc.

6.2.2.2 CPU STATUS REGISTER (CPUSTA)

The CPUSTA register contains the status and control bits for the CPU. This register is used to globally enable/disable interrupts. If only a specific interrupt is desired to be enabled/disabled, please refer to the INTerrupt STAtus (INTSTA) register and the Peripheral Interrupt Enable (PIE) register. This register also indicates if the stack is available and contains the Power Down ($\overline{PD}$) and Time-Out ($\overline{TO}$) bits. The $\overline{TO}$, $\overline{PD}$, and STKAV bits are not writable. These bits are set or cleared according to device logic. Therefore, the result of an instruction with the CPUSTA register as destination may be different than intended.

FIGURE 6-8: CPUSTA REGISTER (ADDRESS: 06H, UNBANKED)

U - 0	U - 0	R - 1	R/W - 1	R - 1	R - 1	U - 0	U - 0
—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, Read as '0'
- n = Value at POR reset

bit 7-6: **Unimplemented**, reads as '0'

bit 5: **STKAV**: Stack Available bit
This bit indicates that the 4-bit stack pointer value is Fh, or has rolled over from Fh→ 0h (stack overflow).
1 = Stack is available
0 = Stack is full, or a stack-overflow may have occurred (Once this bit has been cleared by a stack overflow, only a device reset will set this bit)

bit 4: **GLINTD**: Global Interrupt Disable bit
This bit disables all interrupts. When enabling interrupts, only the sources with their enable bits set can cause an interrupt.
1 = Disable all interrupts
0 = Enables all un-masked interrupts

bit 3: $\overline{TO}$: WDT Time-Out status bit
1 = After power-up or by a `CLRWDT` instruction
0 = A Watchdog Timer time-out occurred

bit 2: $\overline{PD}$: Power Down status bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 1-0: **Unimplemented**, reads as '0'

© 1995 Microchip Technology Inc.

6.2.2.3 TMR0 STATUS/CONTROL REGISTER (T0STA)

This register contains various control bits. One bit is used to control the edge upon which a signal on the RA0/INT pin will set the INT interrupt flag. The other bits (shaded) configure the Timer0 prescaler and clock source. These shaded bits are described in Figure 11-1.

FIGURE 6-9: T0STA REGISTER (ADDRESS: 05H, UNBANKED)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0
INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented, reads as '0'
-n = Value at POR reset

bit 7: **INTEDG**: INT Pin Interrupt Edge Select bit
This bit selects the edge upon which the interrupt is detected.
1 = Rising edge of RA0/INT pin generates interrupt
0 = Falling edge of RA0/INT pin generates interrupt

bit 6: **T0SE**: Timer0 Clock Input Edge Select bit
This bit selects the edge upon which TMR0 will increment.
When T0CS = 0
1 = Rising edge of RA1/T0CKI pin increments Timer0 and/or generates a T0CKIF interrupt
0 = Falling edge of RA1/T0CKI pin increments Timer0 and/or generates a T0CKIF interrupt
When T0CS = 1
Don't care

bit 5: **T0CS**: Timer0 Clock Source Select bit
This bit selects the clock source for TMR0.
1 = Internal instruction clock cycle (TCY)
0 = T0CKI pin

bit 4-1: **PS3:PS0**: Timer0 Prescale Selection bits
These bits select the prescale value for TMR0.

PS3:PS0	Prescale Value
0000	1:1
0001	1:2
0010	1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

bit 0: **Unimplemented**: reads as '0'

© 1995 Microchip Technology Inc.

6.3 Stack Operation

The PIC17C4X devices have a 16 x 16-bit wide hardware stack (see Figure 6-1). The stack is not part of either the program or data memory space, and the stack pointer is neither readable nor writable. The PC is "PUSHed" onto the stack when a CALL instruction is executed or an interrupt is acknowledged. The stack is "POPed" in the event of a RETURN, RETLW, or a RETFIE instruction execution. PCLATH is not affected by a "PUSH" or a "POP" operation.

The stack operates as a circular buffer, with the stack pointer initialized to '0' after all resets. There is a stack available bit (STKAV) to allow software to ensure that the stack has not overflowed. The STKAV bit is set after a device reset. When the stack pointer equals Fh, STKAV is cleared. When the stack pointer rolls over from Fh to 0h, the STKAV bit will be held clear until a device reset.

Note 1: There is not a status bit for stack underflow. The STKAV bit can be used to detect the underflow which results in the stack pointer being at the top of stack.

Note 2: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt vector.

Note 3: After a reset, if a "POP" operation occurs before "PUSH" operation, the STKAV bit will be cleared. This will appear as if the stack is full (underflow has occured). If a "PUSH" operation occurs next, the STKAV bit will be locked clear. Only a device reset will cause this bit to set.

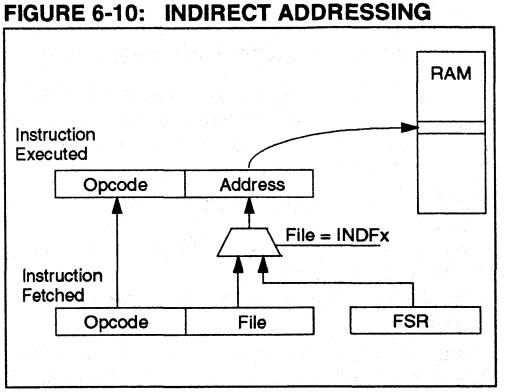
After the device is "PUSHed" sixteen times (without a "POP"), the seventeenth push overwrites the value from the first push. The eighteenth push overwrites the second push (and so on).

6.4 Indirect Addressing

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. That is, the register that is to be read or written can be modified by the program. This can be useful for data tables in the data memory. Figure 6-10 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Example 6-1 shows the use of indirect addressing to clear RAM in a minimum number of instructions. A similar concept could be used to move a defined number of bytes (block) of data to the SCI transmit register (TXREG). The starting address of the block of data to be transmitted could easily be modified by the program.

FIGURE 6-10: INDIRECT ADDRESSING



6.4.1 INDIRECT ADDRESSING REGISTERS

The PIC17C4X has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a NOP, and the status bits are not affected.

6.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVPF and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR.

A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

EXAMPLE 6-1: INDIRECT ADDRESSING

```
    MOVLW   0x20          ;
    MOVWF   FSR0          ; FSR0 = 20h
    BCF     ALUSTA, FS1   ; Increment FSR
    BSF     ALUSTA, FS0   ; after access
    BCF     ALUSTA, C     ; C = 0
    MOVLW   END_RAM + 1   ;
LP  CLRF    INDF0         ; Addr(FSR) = 0
    CPFSEQ  FSR0          ; FSR0 = END_RAM+1?
    GOTO    LP            ; NO, clear next
    :                     ; YES, All RAM is
    :                     ; cleared
```

6.5 Table Pointer (TBLPTRL and TBLPTRH)

File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

6.6 Table Latch (TBLATH, TBLATL)

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see descriptions of instructions TABLRD, TABLWT, TLRD and TLWT). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

© 1995 Microchip Technology Inc.

6.7 Program Counter Module

The Program Counter (PC) is a 16-bit register. PCL, the low byte of the PC, is mapped in the data memory. PCL is readable and writable just as is any other register. PCH is the high byte of the PC and is not directly addressable. Since PCH is not mapped in data or program memory, an 8-bit register PCLATH (PC high latch) is used as a holding latch for the high byte of the PC. PCLATH is mapped into data memory. The user can read or write PCH through PCLATH.

The 16-bit wide PC is incremented after each instruction fetch during Q1 unless:

- Modified by GOTO, CALL, LCALL, RETURN, RETLW, or RETFIE instruction
- Modified by an interrupt response
- Due to destination write to PCL by an instruction

"Skips" are equivalent to a forced NOP cycle at the skipped address.

Note: On POR, the contents of the PCLATH register are unknown. The PCLATH should be initialized before a CALL, GOTO, or any instruction that modifies the PCL register is executed.

The operations of the PC and PCLATH for different instructions are as follows:

a) LCALL instruction:
   An 8-bit destination address is provided in the instruction (opcode). PCLATH is unchanged.
   PCLATH → PCH
   Opcode<7:0> → PCL

b) CALL, GOTO instructions:
   A 13-bit destination address is provided in the instruction (opcode).
   Opcode<12:0> → PC <12:0>
   PC<15:13> → PCLATH<7:5>
   Opcode<12:8> → PCLATH <4:0>

c) Read instructions on PCL:
   Any instruction that reads PCL.
   PCL → data bus → ALU or destination
   PCH → PCLATH

d) Write instructions on PCL:
   Any instruction that writes to PCL.
   8-bit data → data bus → PCL
   PCLATH → PCH

e) Read-Modify-Write instructions on PCL:
   Any instruction that does a read-write-modify operation on PCL, such as ADDWF PCL.
   Read: PCL → data bus → ALU
   Write: 8-bit result → data bus → PCL
   PCLATH → PCH

f) RETURN instruction:
   PCH → PCLATH

The read-modify-write only affects the PCL with the result. PCH is loaded with the value in the PCLATH. For example, ADDWF PCL will result in a jump within the current page. If PC = 03F0h, WREG = 30h and PCLATH = 03h before instruction, PC = 0320h after the instruction. To accomplish a true 16-bit computed jump, the user needs to compute the 16-bit destination address, write the high byte to PCLATH and then write the low value to PCL.

The following PC related operations do not change PCLATH:

a) LCALL, RETLW, and RETFIE instructions.
b) Interrupt vector is forced onto the PC.
c) Read-modify-write instructions on PCL (e.g. BSF PCL).

© 1995 Microchip Technology Inc.

6.8 Bank Select Register (BSR)

The BSR is used to switch between banks in the data memory area (see Figure 6-11). In the PIC17C42, only the lower nibble is implemented. While in the PIC17C43 and PIC17C44 devices, the entire byte is implemented. The lower nibble is used to select the peripheral register bank. The upper nibble is used to select the general purpose memory bank.

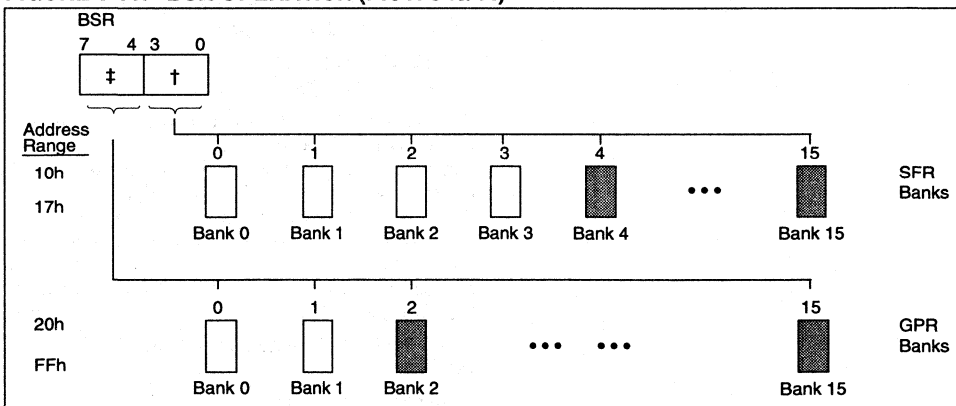
All the Special Function Registers (SFRs) are mapped into the data memory space. In order to accommodate the large number of registers, a banking scheme has been used. A segment of the SFRs, from address 10h to address 17h, is banked. The lower nibble of the bank select register (BSR) selects the currently active "peripheral bank". Effort has been made to group the peripheral registers of related functionality in one bank. However, it will still be necessary to switch from bank to bank in order to address all peripherals related to a single task. To assist this, a `MOVLB bank` instruction is in the instruction set.

For the PIC17C43 and PIC17C44 devices, the need for a large general purpose memory space dictated a general purpose RAM banking scheme. The upper nibble of the BSR selects the currently active general purpose RAM bank. To assist this, a `MOVLR bank` instruction has been provided in the instruction set.

If the currently selected bank is not implemented (such as Bank 13), any read will read all '0's. Any write is completed to the bit bucket and the ALU status bits will be set/cleared as appropriate.

Note: Registers in Bank 15 in the Special Function Register area, are reserved for Microchip use. Reading of registers in this bank may cause random values to be read.

FIGURE 6-11: BSR OPERATION (PIC17C43/44)



† Only Banks 0 thru Bank 3 are implemented. Selection of an unimplemented bank is not recommended. **Bank 15 is reserved for Microchip use, reading of registers in this bank may cause random values to be read.**

‡ Only Banks 0 and Bank 1 are implemented. Selection of an unimplemented bank is not recommended.

© 1995 Microchip Technology Inc.

7.0 TABLE READS AND TABLE WRITES

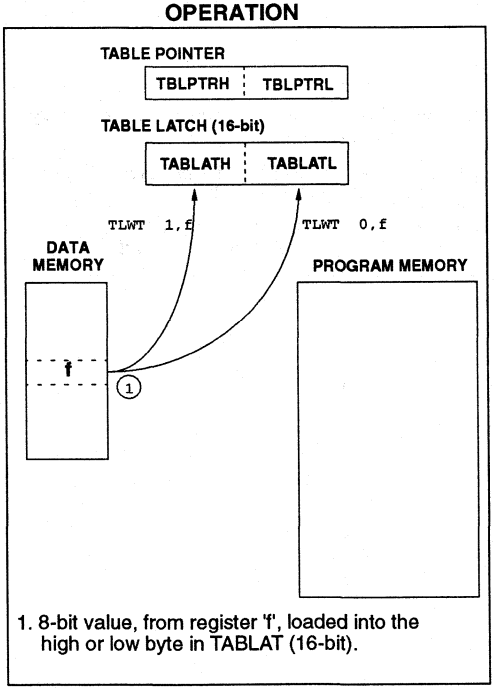
The PIC17C4X has four instructions that allow the processor to move data from the data memory space to the program memory space, and vice versa. Since the program memory space is 16-bits wide and the data memory space is 8-bits wide, two operations are required to move 16-bit values to/from the data memory.

The `TLWT t,f` and `TABLWT t,i,f` instructions are used to write data from the data memory space to the program memory space. The `TLRD t,f` and `TABLRD t,i,f` instructions are used to write data from the program memory space to the data memory space.

The program memory can be internal or external. For the program memory access to be external, the device needs to be operating in extended microcontroller or microprocessor mode.

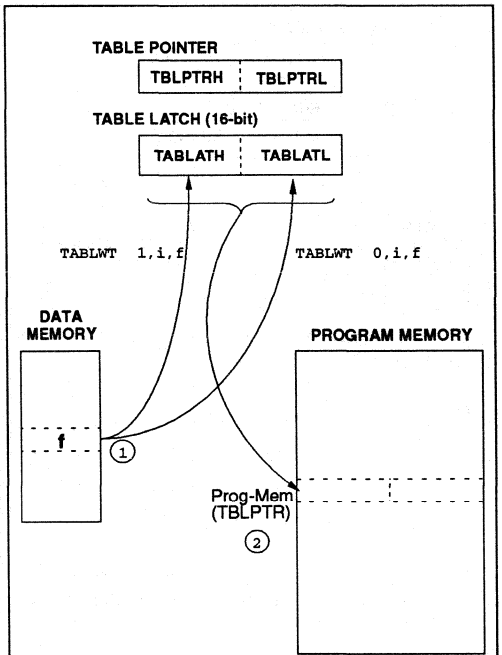
Figure 7-1 through Figure 7-4 show the operation of these four instructions.

FIGURE 7-1: TLWT INSTRUCTION OPERATION



1. 8-bit value, from register 'f', loaded into the high or low byte in TABLAT (16-bit).

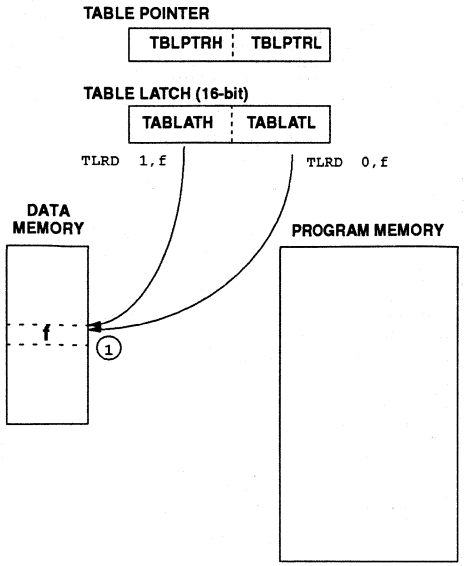
FIGURE 7-2: TABLWT INSTRUCTION OPERATION



1. 8-bit value, from register 'f', loaded into the high or low byte in TABLAT (16-bit).
2. 16-bit TABLAT value written to address Program Memory (TBLPTR).
3. If "i" = 1, then TBLPTR = TBLPTR + 1. If "i" = 0, then TBLPTR is unchanged.

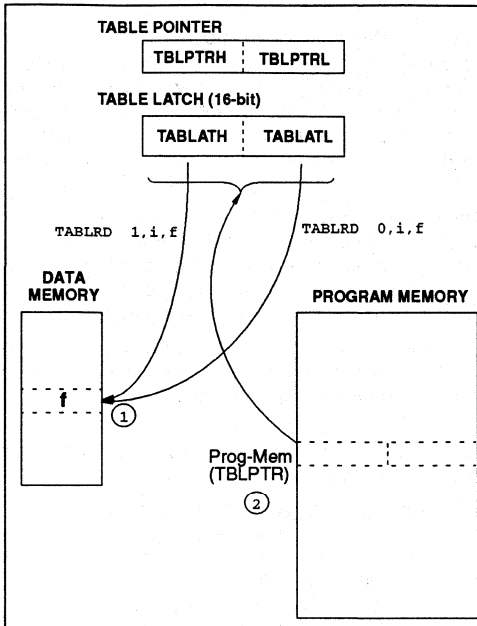
2

FIGURE 7-3: TLRD INSTRUCTION OPERATION



1. 8-bit value, from TABLAT (16-bit) high or low byte, loaded into register 'f'.

FIGURE 7-4: TABLRD INSTRUCTION OPERATION



1. 8-bit value, from TABLAT (16-bit) high or low byte, loaded into register 'f'.
2. 16-bit value at Program Memory (TBLPTR) loaded into TABLAT register.
3. If "i" = 1, then TBLPTR = TBLPTR + 1.
   If "i" = 0, then TBLPTR is unchanged.

© 1995 Microchip Technology Inc.

7.1 Table Writes to Internal Memory

A table write operation to internal memory causes a long write operation. The long write is necessary for programming the internal EPROM. Instruction execution is halted while in a long write cycle. The long write will be terminated by any enabled interrupt. To ensure that the EPROM location has been well programmed, a minimum programming time is required (see specification #D114). Having only one interrupt enabled to terminate the long write ensures that no unintentional interrupts will prematurely terminate the long write.

The sequence of events for programming an internal program memory location should be:

1. Disable all interrupt sources, except the source to terminate EPROM program write.
2. Raise $\overline{MCLR}/V_{PP}$ pin to the programming voltage.
3. Clear the WDT.
4. Do the table write. The interrupt will terminate the long write.
5. Verify the memory location (table read).

Note: Programming timing requirements must be met. See timing specification in electrical specifications for the desired device.

7.1.1 TERMINATING LONG WRITES

An interrupt source or reset are the only events that terminate a long write operation. Terminating the long write from an interrupt source requires that the interrupt enable and flag bits are set. The GLINTD bit only enables the vectoring to the interrupt address.

If the T0CKI, INT, or TMR0 interrupt source is used to terminate the long write; the interrupt flag, of the highest priority enabled interrupt, will terminate the long write and automatically be cleared.

Note 1: If an interrupt is pending, the TABLWT is aborted (an NOP is executed). The highest priority pending interrupt, from the T0CKI, INT, or TMR0 sources that is enabled, has its flag cleared.

Note 2: If the interrupt is not being used for the program write timing, the interrupt should be disabled. This will ensure that the interrupt is not lost, nor will it terminate the long write prematurely.

If a peripheral interrupt source is used to terminate the long write, the interrupt enable and flag bits must be set. The interrupt flag will not be automatically cleared upon the vectoring to the interrupt vector address.

If the GLINTD bit is cleared prior to the long write, when the long write is terminated, the program will branch to the interrupt vector.

If the GLINTD bit is set prior to the long write, when the long write is terminated, the program will not vector to the interrupt address.

TABLE 7-1: INTERRUPT - TABLE WRITE INTERACTION

Interrupt Source	GLINTD	Enable Bit	Flag Bit	Action
INT, TMR0, T0CKI	0	1	1	Terminate table write, branch to interrupt vector (branch clears flag bit).
	0	1	0	None
	1	0	x	None
	1	1	1	Terminate table write, do not branch to interrupt vector (flag is automatically cleared).
Peripheral	0	1	1	Terminate table write, branch to interrupt vector.
	0	1	0	None
	1	0	x	None
	1	1	1	Terminate table write, do not branch to interrupt vector (flag is set).

7.2 Table Writes to External Memory

Table writes to external memory are always two cycle instructions. The second cycle writes the data to the external memory location. The sequence of events for an external memory write are the same as an internal write.

Note: If an interrupt is pending or occurs during the TABLWT, the two cycle table write completes. The INT, TMR0, or TOCKI interrupt flag is automatically cleared or the pending peripheral interrupt is acknowledged.

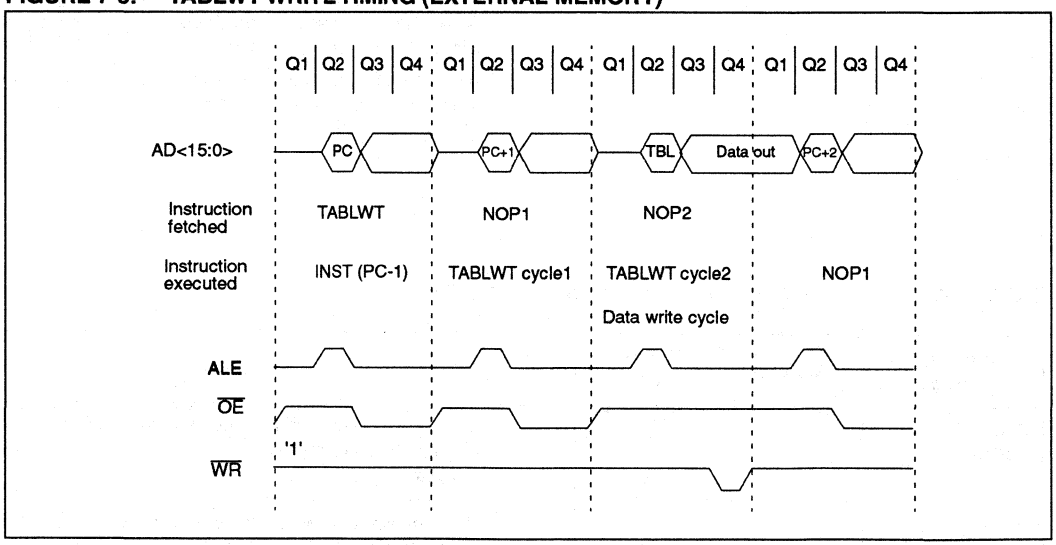
7.2.2 TABLE WRITE CODE

The "i" operand of the TABLWT instruction can specify that the value in the 16-bit TBLPTR register is automatically incremented for the next write. In Example 7-1, the TBLPTR register is not automatically incremented.

EXAMPLE 7-1: TABLE WRITE

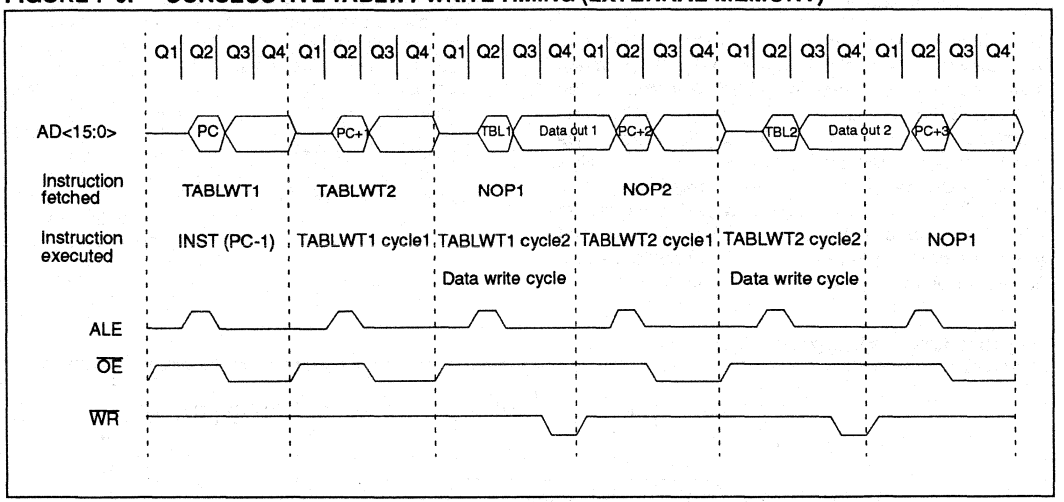
```
CLRWDT                  ; Clear WDT
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH         ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL         ;
MOVLW   HIGH (DATA)     ; Load HI byte
TLWT    1, WREG         ;   in TABLATCH
MOVLW   LOW (DATA)      ; Load LO byte
TABLWT  0,0,WREG        ;   in TABLATCH
                        ;   and write to
                        ;   program memory
                        ;   (Ext. SRAM)
```

FIGURE 7-5: TABLWT WRITE TIMING (EXTERNAL MEMORY)



© 1995 Microchip Technology Inc.

FIGURE 7-6: CONSECUTIVE TABLWT WRITE TIMING (EXTERNAL MEMORY)



© 1995 Microchip Technology Inc.

7.3 Table Reads

The table read allows the program memory to be read. This allows constant data to be stored in the program memory space, and retrieved into data memory when needed. Example 7-2 reads the 16-bit value at program memory address TBLPTR. After the dummy byte has been read from the TABLATH, the TABLATH is loaded with the 16-bit data from program memory address TBLPTR + 1. The first read loads the data into the latch, and can be considered a dummy read (unknown data loaded into 'f'). INDF0 should be configured for either auto-increment or auto-decrement.

EXAMPLE 7-2: TABLE READ

```
MOVLW   HIGH (TBL_ADDR) ; Load the Table
MOVWF   TBLPTRH         ;   address
MOVLW   LOW (TBL_ADDR)  ;
MOVWF   TBLPTRL         ;
TABLRD  0,0,DUMMY       ; Dummy read,
                        ;  Updates TABLATCH
TLRD    1, INDF0        ; Read HI byte
                        ;   of TABLATCH
TABLRD  0,1,INDF0       ; Read LO byte
                        ;   of TABLATCH and
                        ;   Update TABLATCH
```

FIGURE 7-7: TABLRD TIMING

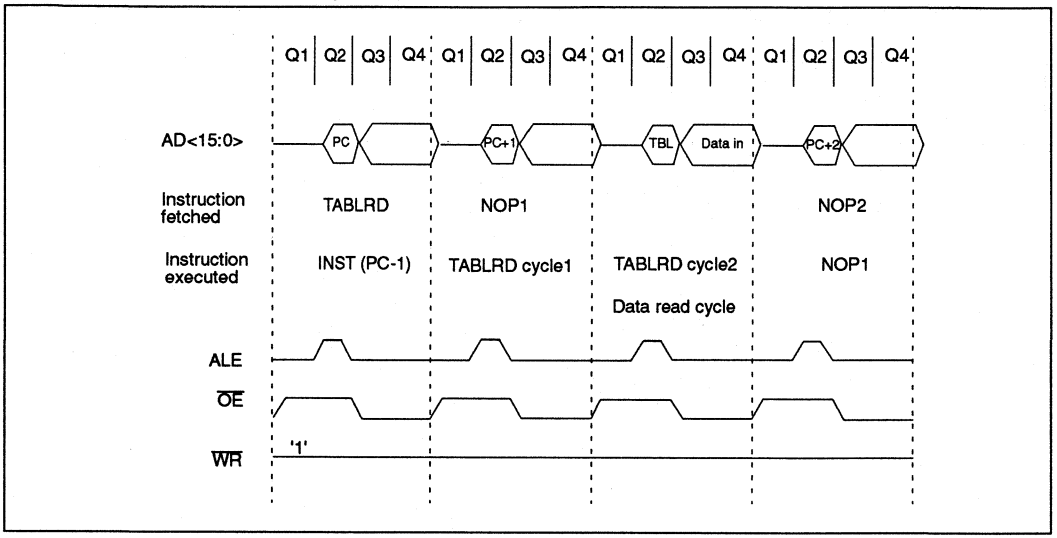
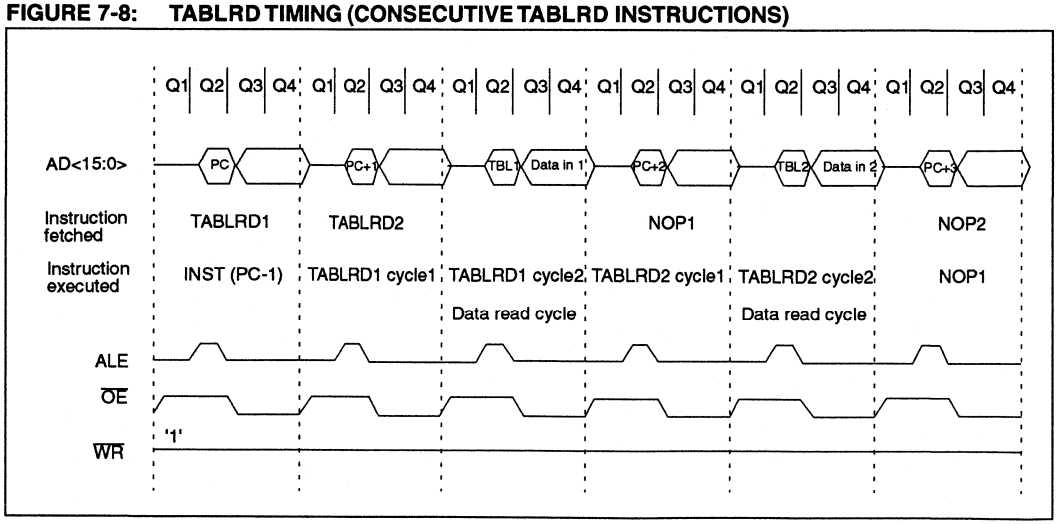


FIGURE 7-8: TABLRD TIMING (CONSECUTIVE TABLRD INSTRUCTIONS)



© 1995 Microchip Technology Inc.

8.0 HARDWARE MULTIPLIER

The PIC17C43 and PIC17C44 devices have an 8 x 8 hardware multiplier included in the ALU of the device. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored into the 16-bit PRODuct register (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the PIC17C43 and PIC17C44 devices to be used in applications previously reserved for Digital Signal Processors.

Example 8-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiply. To account for the sign bits of the arguments, each argument's most significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 X 8 MULTIPLY ROUTINE

```
MOVFP   ARG1, WREG
MULWF   ARG2          ; ARG1 * ARG2 ->
                      ;   PRODH:PRODL
```

EXAMPLE 8-2: 8 X 8 SIGNED MULTIPLY ROUTINE

```
MOVFP   ARG1, WREG
MULWF   ARG2          ; ARG1 * ARG2 ->
                      ;   PRODH:PRODL
BTFSC   ARG2, SB      ; Test Sign Bit
SUBWF   PRODH, F      ; PRODH = PRODH
                      ;         - ARG1
MOVFP   ARG2, WREG
BTFSC   ARG1, SB      ; Test Sign Bit
SUBWF   PRODH, F      ; PRODH = PRODH
                      ;         - ARG2
```

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in 4 registers RES3:RES0.

EQUATION 8-1: 16 X 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L * ARG2H:ARG2L \\
&= (ARG1H * ARG2H * 2^{16}) + \\
&\quad (ARG1H * ARG2L * 2^{8}) + \\
&\quad (ARG1L * ARG2H * 2^{8}) + \\
&\quad (ARG1L * ARG2L)
\end{aligned}
$$

EXAMPLE 8-3: 16 X 16 MULTIPLY ROUTINE

```
    MOVFP   ARG1L, WREG
    MULWF   ARG2L         ; ARG1L * ARG2L ->
                          ;   PRODH:PRODL
    MOVPF   PRODH, RES1   ;
    MOVPF   PRODL, RES0   ;
;
    MOVFP   ARG1H, WREG
    MULWF   ARG2H         ; ARG1H * ARG2H ->
                          ;   PRODH:PRODL
    MOVPF   PRODH, RES3   ;
    MOVPF   PRODL, RES2   ;
;
    MOVFP   ARG1L, WREG
    MULWF   ARG2H         ; ARG1L * ARG2H ->
                          ;   PRODH:PRODL
    MOVFP   PRODL, WREG   ;
    ADDWF   RES1, F       ; Add cross
    MOVFP   PRODH, WREG   ;    products
    ADDWFC  RES2, F       ;
    CLRF    WREG, F       ;
    ADDWFC  RES3, F       ;
;
    MOVFP   ARG1H, WREG   ;
    MULWF   ARG2L         ; ARG1H * ARG2L ->
                          ;   PRODH:PRODL

    MOVFP   PRODL, WREG   ;
    ADDWF   RES1, F       ; Add cross
    MOVFP   PRODH, WREG   ;    products
    ADDWFC  RES2, F       ;
    CLRF    WREG, F       ;
    ADDWFC  RES3, F       ;
```

2

© 1995 Microchip Technology Inc.

Example 8-4 shows the sequence to do an 16 x 16 signed multiply. Equation 8-2 shows the algorithm that used. The 32-bit result is stored in 4 registers RES3:RES0. To account for the sign bits of the arguments, each argument pairs most significant bit (MSb) is tested and the appropriate subtractions are done.

EQUATION 8-2: 16 X 16 SIGNED MULTIPLICATION ALGORITHM

RES3:RES0

= ARG1H:ARG1L * ARG2H:ARG2L

= (ARG1H * ARG2H * 2^{16}) +
(ARG1H * ARG2L * 2^{8}) +
(ARG1L * ARG2H * 2^{8}) +
(ARG1L * ARG2L) +
(-1 * ARG2H<7> * ARG1H:ARG1L * 2^{16}) +
(-1 * ARG1H<7> * ARG2H:ARG2L * 2^{16})

EXAMPLE 8-4: 16 X 16 SIGNED MULTIPLY ROUTINE

```
        MOVFP   ARG1L, WREG
        MULWF   ARG2L           ; ARG1L * ARG2L ->
                                ;   PRODH:PRODL
        MOVPF   PRODH, RES1 ;
        MOVPF   PRODL, RES0 ;
;
        MOVFP   ARG1H, WREG
        MULWF   ARG2H           ; ARG1H * ARG2H ->
                                ;   PRODH:PRODL
        MOVPF   PRODH, RES3 ;
        MOVPF   PRODL, RES2 ;
;
        MOVFP   ARG1L, WREG
        MULWF   ARG2H           ; ARG1L * ARG2H ->
                                ;   PRODH:PRODL
        MOVFP   PRODL, WREG ;
        ADDWF   RES1, F         ; Add cross
        MOVFP   PRODH, WREG ;     products
        ADDWFC  RES2, F         ;
        CLRF    WREG, F         ;
        ADDWFC  RES3, F         ;
;
        MOVFP   ARG1H, WREG ;
        MULWF   ARG2L           ; ARG1H * ARG2L ->
                                ;   PRODH:PRODL

        MOVFP   PRODL, WREG ;
        ADDWF   RES1, F         ; Add cross
        MOVFP   PRODH, WREG ;     products
        ADDWFC  RES2, F         ;
        CLRF    WREG, F         ;
        ADDWFC  RES3, F         ;
;
        BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
        GOTO    SIGN_ARG1       ; no, check ARG1
        MOVFP   ARG1L, WREG ;
        SUBWF   RES2            ;
        MOVFP   ARG1H, WREG ;
        SUBWFB  RES3
;
SIGN_ARG1
        BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
        GOTO    CONT_CODE       ; no, done
        MOVFP   ARG2L, WREG ;
        SUBWF   RES2            ;
        MOVFP   ARG2H, WREG ;
        SUBWFB  RES3
;
CONT_CODE
        :
```


© 1995 Microchip Technology Inc.

TABLE 8-1: PERFORMANCE COMPARISON

Routine	Device	Program Memory (Words)	Cycles (Max)	Time (@ 25 MHz)
8 x 8 unsigned	17C42	13	69	11.04 µs
	17C43 & 17C44	1	1	160 ns
8 x 8 signed	17C42	—	—	—
	17C43 & 17C44	6	6	960 ns
16 x 16 unsigned	17C42	21	242	38.72 µs
	17C43 & 17C44	24	24	3.84 µs
16 x 16 signed	17C42	52	254	40.64 µs
	17C43 & 17C44	36	36	5.76 µs

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

9.0 I/O PORTS

The PIC17C4X devices have five I/O ports, PORTA through PORTE. PORTB through PORTE have a corresponding Data Direction Register (DDR), which is used to configure the port pins as inputs or outputs. These five ports are made up of 33 I/O pins. Some of these ports pins are multiplexed with alternate functions.

PORTC, PORTD and PORTE are multiplexed with the system bus. These pins are configured as the system bus when the device's fuses are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, these pins are general purpose I/O.

PORTA and PORTB are multiplexed with the peripheral features of the device. These peripheral features are:

- Timer modules
- Capture module
- PWM module
- SCI module (USART)
- External Interrupt pin

When some of these peripheral modules are turned on, the port pin will automatically configure to the alternate function. The modules that do this are:

- PWM module
- SCI module

When a pin is automatically configured as an output by a peripheral module, its data direction bit may be left in an unknown state. After disabling the peripheral module, the user should re-initialize the DDR bit to the desired configuration.

The other peripheral modules (which require an input) must have their data direction bit configured appropriately.

Note:	A pin that is a peripheral input, can be configured as an output (DDRx<y> is cleared). The peripheral events will be determined by the action output on the port pin.

9.1 PORTA Register

PORTA is a 6-bit wide latch. PORTA does not have a corresponding Data Direction Register (DDR).

Reading PORTA reads the status of the pins.

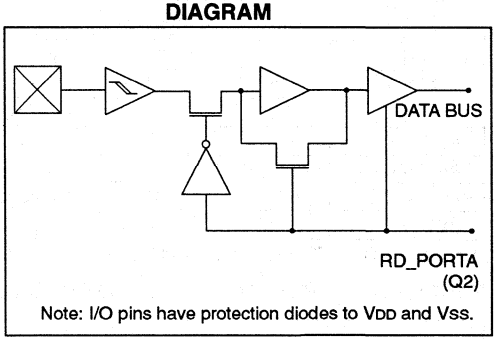
The RA1 pin is multiplexed with TMR0 clock input, and RA4 and RA5 are multiplexed with the SCI functions. The control of RA4 and RA5 as outputs is automatically configured by the SCI module.

9.1.1 USING RA2, RA3 AS OUTPUTS

The RA2 and RA3 pins are open collector outputs. To use the RA2 or the RA3 pin(s) as output(s), simply write to the PORTA register the desired value. A '0' will cause the pin to drive low, while a '1' will cause the pin to float (hi-impedance). An external pull-up resistor should be used to pull the pin high. Writes to PORTA will not affect the other pins.

Note:	When using the RA2 or RA3 pin(s) as output(s), read-modify-write instructions (such as BCF, BSF, BTG) on PORTA are not recommended. Such operations read the port pins, do the desired operation, and then write this value to the data latch. This may inadvertently cause the RA2 or RA3 pins to switch from input to output (or vice-versa).

FIGURE 9-1: RA0 AND RA1 BLOCK DIAGRAM



Note: I/O pins have protection diodes to VDD and Vss.

FIGURE 9-2: RA2 AND RA3 BLOCK DIAGRAM

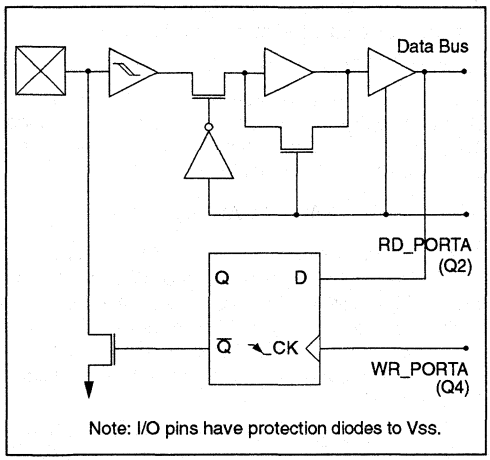


FIGURE 9-3: RA4 AND RA5 BLOCK DIAGRAM

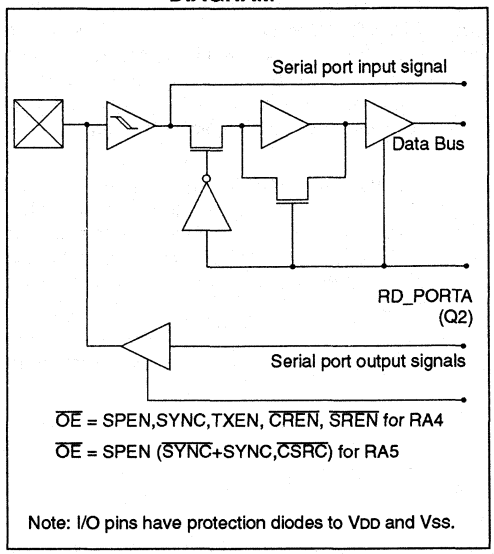


TABLE 9-1: PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0/INT	bit0	ST	Input or external interrupt input.
RA1/T0CKI	bit1	ST	Input or clock input to the TMR0 timer/counter.
RA2	bit2	ST	Input/Output. Output is open collector type.
RA3	bit3	ST	Input/Output. Output is open collector type.
RA4/RX/DT	bit4	ST	Input or SCI Asynchronous Receive or SCI Synchronous Data.
RA5/TX/CK	bit5	ST	Input or SCI Asynchronous Transmit or SCI Synchronous Clock.
RBPU	bit7	—	Control bit for PORTB weak pull-ups.

Legend: TTL = TTL input, ST = Schmitt Trigger input.

TABLE 9-2: REGISTERS/BITS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
10h, Bank 0	PORTA	RBPU	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
05h, Unbanked	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—	0000 000-	0000 000-
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u

Legend: x = unknown, u = unchanged, - = unimplemented, reads as '0'.
Note 1: Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer time-out reset.
2: Shaded cells are not used by PORTA.

© 1995 Microchip Technology Inc.

9.2 PORTB and DDRB Registers

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is DDRB. A '1' in DDRB configures the corresponding port pin as an input. A '0' in the DDRB register configures the corresponding port pin as an output. Reading PORTB reads the status of the pins, whereas writing to it will write to the port latch.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (PORTA<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are enabled on any reset.

PORTB also has an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB<7:0> pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB<7:0>) are compared with the value in the PORTB data latch. The "mismatch" outputs of RB<7:0> are OR'ed together to generate the RBIF interrupt (flag latched in PIR<7>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

a) Disable the interrupt by clearing the RBIE (PIE<7>) bit.

b) Read-Write PORTB (`MOVPF PORTB, PORTB`). This will end mismatch condition. Then, clear the RBIF bit.

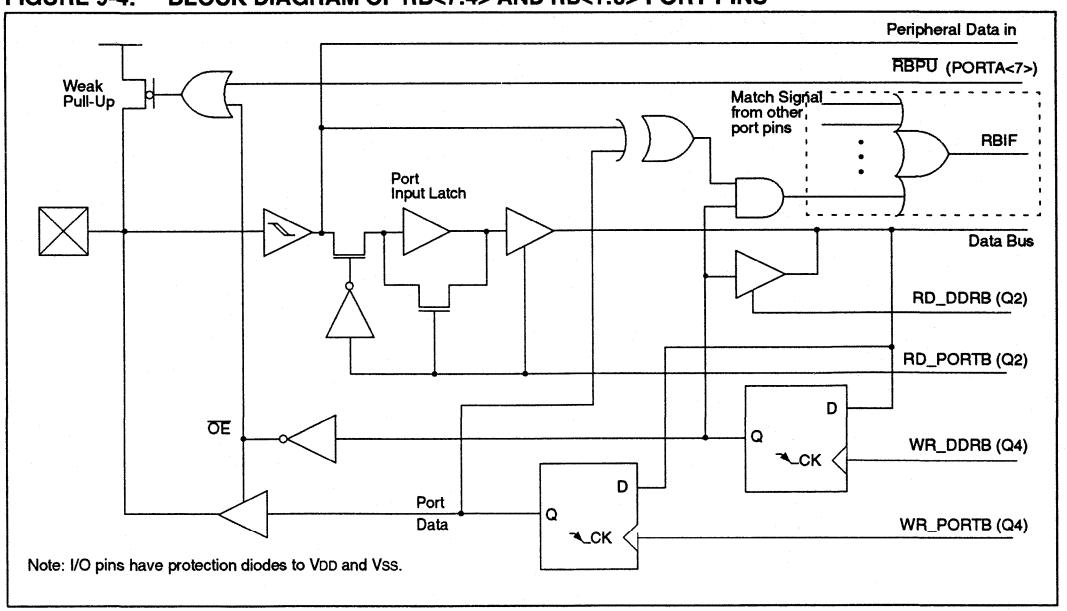
A mismatch condition will continue to set the RBIF bit. Reading then writing PORTB will end the mismatch condition, and allow the RBIF bit to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on this port, allows easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552 in the *Embedded Control Handbook*).

The interrupt on change feature is recommended for wake-up on operations where PORTB is only used for the interrupt on change feature and key depression operation.

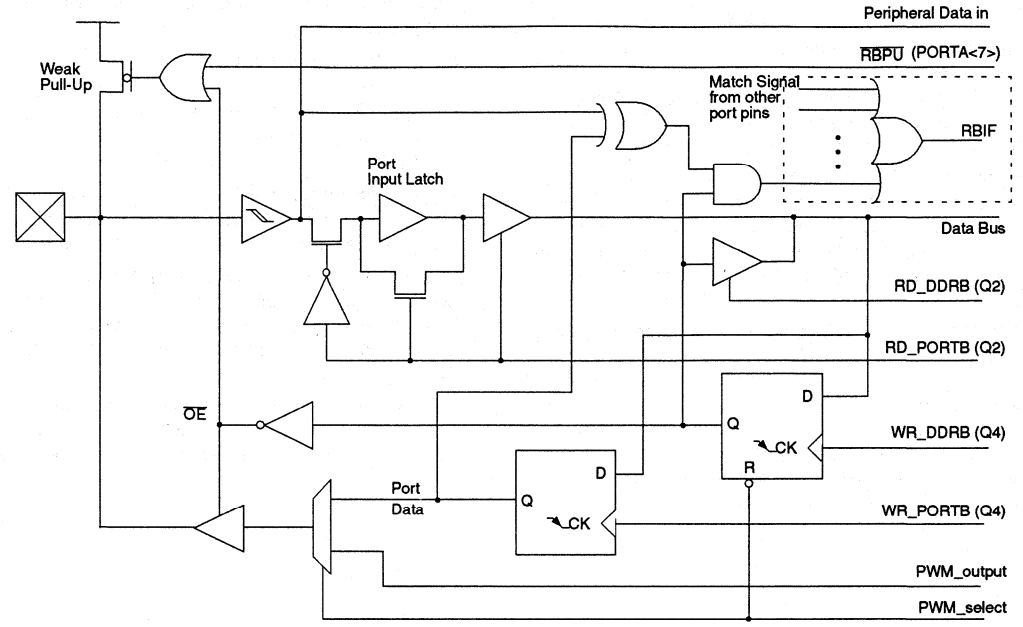
2

FIGURE 9-4: BLOCK DIAGRAM OF RB<7:4> AND RB<1:0> PORT PINS



Note: I/O pins have protection diodes to VDD and VSS.

FIGURE 9-5: BLOCK DIAGRAM OF RB<3:2> PORT PINS



Note: I/O pins have protection diodes to VDD and Vss.

© 1995 Microchip Technology Inc.

Example 9-1 shows the instruction sequence to initialize PORTB. The Bank Select Register (BSR) must be selected to Bank 0 for the port to be initialized.

EXAMPLE 9-1: INITIALIZING PORTB

```
MOVLB  0          ; Select Bank 0
CLRF   PORTB      ;Initialize PORTB by clearing
                  ; output data latches
MOVLW  0xCF       ;Value used to initialize
                  ;data direction
MOVWF  DDRB       ;Set RB<3:0> as inputs
                  ;   RB<5:4> as outputs
                  ;   RB<7:6> as inputs
```


TABLE 9-3: PORTB FUNCTIONS

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/Output or the RB0/CAP1 input pin. Software programmable weak pull-up and interrupt on change features.
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pull-up and interrupt on change features.
RB2/PWM1	bit2	ST	Input/Output or the RB2/PWM1 output pin. Software programmable weak pull-up and interrupt on change features.
RB3/PWM2	bit3	ST	Input/Output or the RB3/PWM2 output pin. Software programmable weak pull-up and interrupt on change features.
RB4/TCLK12	bit4	ST	Input/Output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt on change features.
RB5/TCLK3	bit5	ST	Input/Output or the external clock input to Timer3. Software programmable weak pull-up and interrupt on change features.
RB6	bit6	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.
RB7	bit7	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.

Legend: TTL = TTL input, ST = Schmitt Trigger input.

TABLE 9-4: REGISTERS/BITS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
12h, Bank 0	PORTB	PORTB data latch								xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data direction register for PORTB								1111 1111	1111 1111
10h, Bank 0	PORTA	$\overline{RBPU}$	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 ??--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented, reads as '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.
2: Shaded cells are not used by PORTB.

© 1995 Microchip Technology Inc.

9.3 PORTC and DDRC Registers

PORTC is an 8-bit bi-directional port. The corresponding data direction register is DDRC. A '1' in DDRC configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTC reads the status of the pins, whereas writing to it will write to the port latch. PORTC is multiplexed with the system bus. When operating as the system bus, PORTC is the low order byte of the address/data bus (AD<7:0>). The timing for the system bus is shown in the Electrical Characteristics section.

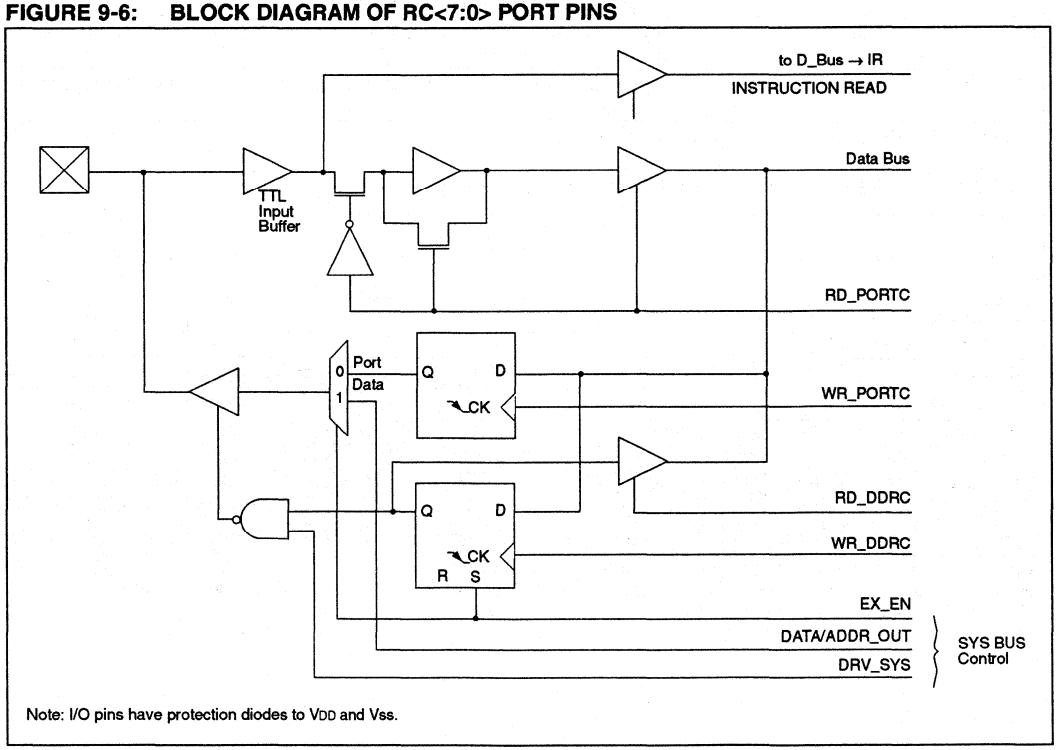
Note: This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-2 shows the instruction sequence to initialize PORTC. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

EXAMPLE 9-2: INITIALIZING PORTC

```
MOVLB  1          ;  Select Bank 1
CLRF   PORTC      ;  Initialize PORTC data
                  ;   latches before setting
                  ;   the data direction
                  ;   register
MOVLW  0xCF       ;  Value used to initialize
                  ;   data direction
MOVWF  DDRC       ;  Set RC<3:0> as inputs
                  ;   RC<5:4> as outputs
                  ;   RC<7:6> as inputs
```

FIGURE 9-6: BLOCK DIAGRAM OF RC<7:0> PORT PINS



Note: I/O pins have protection diodes to VDD and Vss.

© 1995 Microchip Technology Inc.

TABLE 9-5: PORTC FUNCTIONS

Name	Bit	Buffer Type	Function
RC0/AD0	bit0	TTL	Input/Output or system bus address/data pin.
RC1/AD1	bit1	TTL	Input/Output or system bus address/data pin.
RC2/AD2	bit2	TTL	Input/Output or system bus address/data pin.
RC3/AD3	bit3	TTL	Input/Output or system bus address/data pin.
RC4/AD4	bit4	TTL	Input/Output or system bus address/data pin.
RC5/AD5	bit5	TTL	Input/Output or system bus address/data pin.
RC6/AD6	bit6	TTL	Input/Output or system bus address/data pin.
RC7/AD7	bit7	TTL	Input/Output or system bus address/data pin.

Legend: TTL = TTL input.

TABLE 9-6: REGISTERS/BITS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
11h, Bank 1	PORTC	RC7/ AD7	RC6/ AD6	RC5/ AD5	RC4/ AD4	RC3/ AD3	RC2/ AD2	RC1/ AD1	RC0/ AD0	xxxx xxxx	uuuu uuuu
10h, Bank 1	DDRC	Data direction register for PORTC								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and the Watchdog Timer time-out reset.

© 1995 Microchip Technology Inc.

9.4 PORTD and DDRD Registers

PORTD is an 8-bit bi-directional port. The corresponding data direction register is DDRD. A '1' in DDRD configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTD reads the status of the pins, whereas writing to it will write to the port latch. PORTD is multiplexed with the system bus. When operating as the system bus, PORTD is the high order byte of the address/data bus (AD<15:8>). The timing for the system bus is shown in the Electrical Characteristics section.

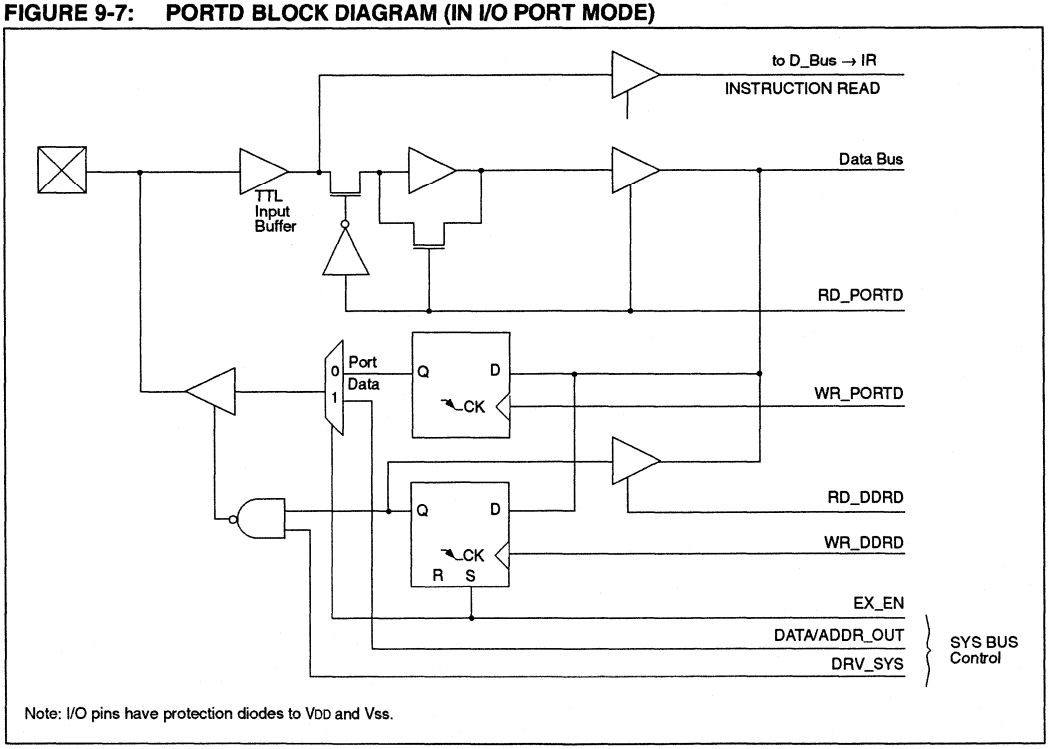
Note: This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-3 shows the instruction sequence to initialize PORTD. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

EXAMPLE 9-3: INITIALIZING PORTD

```
MOVLB  1          ; Select Bank 1
CLRF   PORTD      ; Initialize PORTD data
                  ;  latches before setting
                  ;  the data direction
                  ;  register
MOVLW  0xCF       ; Value used to initialize
                  ;  data direction
MOVWF  DDRD       ; Set RD<3:0> as inputs
                  ;  RD<5:4> as outputs
                  ;  RD<7:6> as inputs
```

FIGURE 9-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and Vss.

© 1995 Microchip Technology Inc.

TABLE 9-7: PORTD FUNCTIONS

Name	Bit	Buffer Type	Function
RD0/AD8	bit0	TTL	Input/Output or system bus address/data pin.
RD1/AD9	bit1	TTL	Input/Output or system bus address/data pin.
RD2/AD10	bit2	TTL	Input/Output or system bus address/data pin.
RD3/AD11	bit3	TTL	Input/Output or system bus address/data pin.
RD4/AD12	bit4	TTL	Input/Output or system bus address/data pin.
RD5/AD13	bit5	TTL	Input/Output or system bus address/data pin.
RD6/AD14	bit6	TTL	Input/Output or system bus address/data pin.
RD7/AD15	bit7	TTL	Input/Output or system bus address/data pin.

Legend: TTL = TTL input.

TABLE 9-8: REGISTERS/BITS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
13h, Bank 1	PORTD	RD7/ AD15	RD6/ AD14	RD5/ AD13	RD4/ AD12	RD3/ AD11	RD2/ AD10	RD1/ AD9	RD0/ AD8	xxxx xxxx	uuuu uuuu
12h, Bank 1	DDRD	Data direction register for PORTD								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and the Watchdog Timer time-out reset.

© 1995 Microchip Technology Inc.

9.4.1 PORTE AND DDRE REGISTER

PORTE is a 3-bit bi-directional port. The corresponding data direction register is DDRE. A '1' in DDRE configures the corresponding port pin as an input. A '0' in the DDRE register configures the corresponding port pin as an output. Reading PORTE reads the status of the pins, whereas writing to it will write to the port latch. PORTE is multiplexed with the system bus. When operating as the system bus, PORTE contains the control signals for the address/data bus (AD<15:0>). These control signals are Address Latch Enable (ALE), Output Enable ($\overline{OE}$), and Write ($\overline{WR}$). The control signals $\overline{OE}$ and $\overline{WR}$ are active low signals. The timing for the system bus is shown in the Electrical Characteristics section.

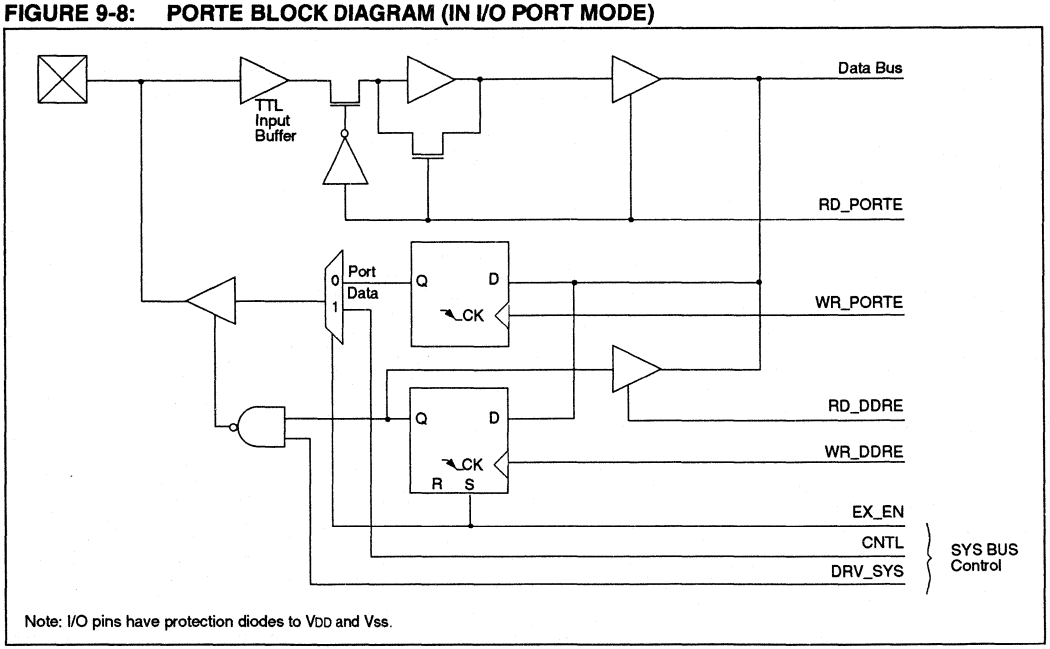
Note: This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

Example 9-4 shows the instruction sequence to initialize PORTE. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

EXAMPLE 9-4: INITIALIZING PORTE

```
MOVLB  1          ;  Select Bank 1
CLRF   PORTE      ;  Initialize PORTE data
                  ;   latches before setting
                  ;   the data direction
                  ;   register
MOVLW  0x03       ;  Value used to initialize
                  ;   data direction
MOVWF  DDRE       ;  Set RE<1:0> as inputs
                  ;   RE<2> as outputs
                  ;   RE<7:3> are always
                     read as '0'
```

FIGURE 9-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and Vss.

© 1995 Microchip Technology Inc.

TABLE 9-9: PORTE FUNCTIONS

Name	Bit	Buffer Type	Function
RE0/ALE	bit0	TTL	Input/Output or system bus Address Latch Enable (ALE) control pin.
RE1/$\overline{OE}$	bit1	TTL	Input/Output or system bus Output Enable ($\overline{OE}$) control pin.
RE2/$\overline{WR}$	bit2	TTL	Input/Output or system bus Write ($\overline{WR}$) control pin.

Legend: TTL = TTL input.

TABLE 9-10: REGISTERS/BITS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
15h, Bank 1	PORTE	—	—	—	—	—	RE2/$\overline{WR}$	RE1/$\overline{OE}$	RE0/ALE	---- -xxx	---- -uuu
14h, Bank 1	DDRE	Data direction register for PORTE								---- -111	---- -111

Legend: x = unknown, u = unchanged, -- = unimplemented, reads as '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.
2: Shaded cells are not used by PORTE.

© 1995 Microchip Technology Inc.

9.5 I/O Programming Considerations

9.5.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. For example, the BCF and BSF instructions read the register into the CPU, execute the bit operation, and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading a port reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (BCF, BSF, BTG, etc.) on a port, the value of the port pins is read, the desired operation is performed with this value, and the value is then written to the port latch.

Example 9-5 shows the effect of two sequential read-modify-write instructions on an I/O port.

EXAMPLE 9-5: READ MODIFY WRITE INSTRUCTIONS ON AN I/O PORT

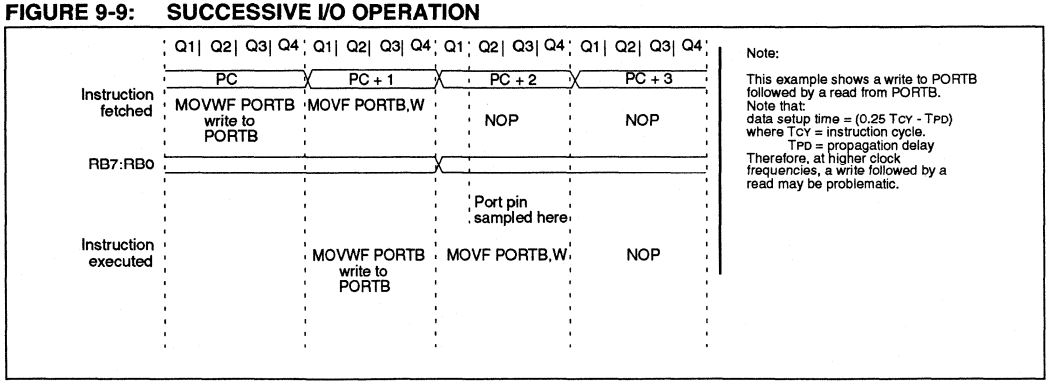
```
; Initial PORT settings: PORTB<7:4> Inputs
;                        PORTB<3:0> Outputs
; PORTB<7:6> have pull-ups and are
; not connected to other circuitry
;
;                        PORT latch  PORT pins
;                        ----------  ---------
;
   BCF   PORTB, 7        01pp pppp   11pp pppp
   BCF   PORTB, 6        10pp pppp   11pp pppp
;
   BCF   DDRB, 7         10pp pppp   11pp pppp
   BCF   DDRB, 6         10pp pppp   10pp pppp
;
; Note that the user may have expected the
; pin values to be 00pp pppp. The 2nd BCF
; caused RB7 to be latched as the pin value
; (High).
```

Note: A pin actively outputting a Low or High should not be driven from external devices in order to change the level on this pin (i.e. "wired-or", "wired-and"). The resulting high output currents may damage the device.

9.5.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (see Figure 9-9). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before executing the instruction that reads the values on that I/O port. Otherwise, the previous state of that pin may be read into the CPU rather than the "new" state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 9-9: SUCCESSIVE I/O OPERATION



© 1995 Microchip Technology Inc.

10.0 OVERVIEW OF TIMER RESOURCES

The PIC17C4X has four timer modules. Each module can generate an interrupt to indicate that an event has occurred. These timers are called:

- TMR0 - Timer0 (16-bit timer with programmable 8-bit prescaler)
- TMR1 - Timer1 (8-bit timer)
- TMR2 - Timer2 (8-bit timer)
- TMR3 - Timer3 (16-bit timer)

For enhanced time-base functionality, two input Captures and two Pulse Width Modulation (PWM) outputs are possible. The PWMs use the TMR1 and TMR2 resources and the input Captures use the TMR3 resource.

10.1 TMR0 Overview

The TMR0 module is a simple 16-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock.

The TMR0 module also has a programmable prescaler option. The PS3:PS0 bits (T0STA<4:1>) determine the prescaler value. TMR0 can increment at the following rates: 1:1, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, 1:256.

When TMR0's clock source is an external clock, the TMR0 module can be selected to increment on either the rising or falling edge.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

10.2 TMR1 Overview

The TMR1 module is an 8-bit timer/counter with an 8-bit period register (PR1). When the TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the TMR2 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR1 register is the LSB and TMR2 is the MSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

10.3 TMR2 Overview

The TMR2 module is an 8-bit timer/counter with an 8-bit period register (PR2). When the TMR2 value rolls over from the period match value to 0h, the TMR2IF flag is set, and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB4/TCLK12 pin, which can also be selected to be the clock for the TMR1 module.

TMR1 can be concatenated to TMR2 to form a 16-bit timer. The TMR2 register is the MSB and TMR1 is the LSB. When in the 16-bit timer mode, there is a corresponding 16-bit period register (PR2:PR1). When the TMR2:TMR1 value rolls over from the period match value to 0h, the TMR1IF flag is set, and an interrupt will be generated when enabled.

10.4 TMR3 Overview

The TMR3 module is a 16-bit timer/counter with a 16-bit period register. When the TMR3H:TMR3L value rolls over to 0h, the TMR3IF bit is set and an interrupt will be generated when enabled. In counter mode, the clock comes from the RB5/TCLK3 pin.

When operating in the dual capture mode, the period registers become the second 16-bit capture register.

10.5 Role of the Timer/Counters

The timer modules are general purpose, but have dedicated resources associated with them. TMR1 and TMR2 are the time-bases for the two Pulse Width Modulation (PWM) outputs, while TMR3 is the time-base for the two input captures.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

11.0 TIMER0

The Timer0 (TMR0) module consists of a 16-bit timer/counter, TMR0. The high byte is TMR0H and the low byte is TMR0L. A software programmable 8-bit prescaler makes an effective 24-bit overflow timer. The clock source is also software programmable as either the internal instruction clock or the RA1/T0CKI pin. The control bits for this module are in register T0STA (Figure 11-1).

FIGURE 11-1: T0STA REGISTER (ADDRESS: 05H, UNBANKED)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0
INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented, Read as '0'
-n = Value at POR reset

bit 7: **INTEDG**: INT Pin Interrupt Edge Select bit
This bit selects the edge upon which the interrupt is detected
1 = Rising edge of RA0/INT pin generates interrupt
0 = Falling edge of RA0/INT pin generates interrupt

bit 6: **T0SE**: Timer0 Clock Input Edge Select bit
This bit selects the edge upon which TMR0 will increment
When T0CS = 0
1 = Rising edge of RA1/T0CKI pin increments Timer0 and/or generates a T0CKIF interrupt
0 = Falling edge of RA1/T0CKI pin increments Timer0 and/or generates a T0CKIF interrupt
When T0CS = 1
Don't care

bit 5: **T0CS**: Timer0 Clock Source Select bit
This bit selects the clock source for TMR0.
1 = Internal instruction clock cycle (TCY)
0 = T0CKI pin

bit 4-1: **PS3:PS0**: Timer0 Prescale Selection bits
These bits select the prescale value for TMR0.

PS3:PS0	Prescale Value
0000	1:1
0001	1:2
0010	1:4
0011	1:8
0100	1:16
0101	1:32
0110	1:64
0111	1:128
1xxx	1:256

bit 0: **Unimplemented**: read as '0'

© 1995 Microchip Technology Inc.

11.1 TMR0 Operation

When T0CS is set, TMR0 increments on the internal clock. When T0CS is clear, TMR0 increments on the external clock (RA1/T0CKI pin). The external clock edge can be configured in software. When T0SE is set, the timer will increment on the rising edge of the RA1/T0CKI pin. When T0SE is clear, the timer will increment on the falling edge of the RA1/T0CKI pin. The prescaler can be programmed to introduce a prescale of 1:1 to 1:256. The timer increments from 0000h to FFFFh and rolls over to 0000h. On overflow, the TMR0 Interrupt Flag bit (T0IF) is set. The TMR0 interrupt can be masked off by clearing the corresponding TMR0 Interrupt Enable bit (T0IE). The TMR0 Interrupt Flag bit (T0IF) is automatically cleared when vectoring to the TMR0 interrupt vector.

11.2 Using TMR0 with External Clock

When the external clock input is used for TMR0, it is synchronized with the internal phase clocks. Figure 11-3 shows the synchronization of the external clock. This synchronization is done after the prescaler. The output of the prescaler (PSOUT) is sampled twice in every instruction cycle to detect a rising or a falling edge. The timing requirements for the external clock are detailed in the electrical specification section for the desired device.

11.2.1 DELAY FROM EXTERNAL CLOCK EDGE

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time TMR0 is actually incremented. Figure 11-3 shows that this delay is between 3 Tosc and 7 Tosc. Thus, for example, measuring the interval between two edges (e.g. period) will be accurate within ±4 Tosc (±160 ns @ 25 MHz).

FIGURE 11-2: TMR0 MODULE BLOCK DIAGRAM

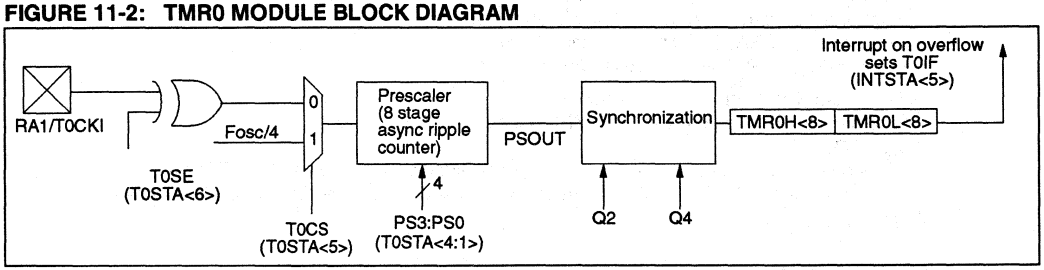
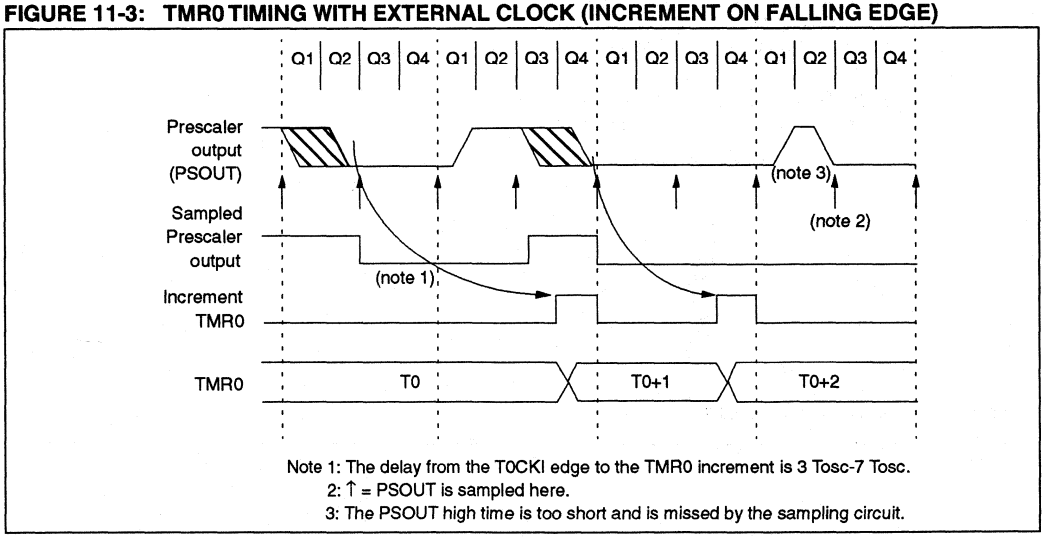


FIGURE 11-3: TMR0 TIMING WITH EXTERNAL CLOCK (INCREMENT ON FALLING EDGE)



Note 1: The delay from the T0CKI edge to the TMR0 increment is 3 Tosc-7 Tosc.
2: ↑ = PSOUT is sampled here.
3: The PSOUT high time is too short and is missed by the sampling circuit.

© 1995 Microchip Technology Inc.

11.3 Read/Write Consideration for TMR0

Although TMR0 is a 16-bit timer/counter, only 8-bits at a time can be read or written during a single instruction cycle. Care must be taken during any read or write.

11.3.1 READING 16-BIT VALUE

The problem in reading the entire 16-bit value is that after reading the low (or high) byte, its value may change from FFh to 00h.

EXAMPLE 11-1: 16-BIT READ

```
MOVPF   TMR0L, TMPLO    ;read low tmr0
MOVPF   TMR0H, TMPHI    ;read high tmr0
MOVFP   TMPLO, WREG     ;tmplo -> wreg
CPFSLT  TMR0L, WREG     ;tmr0l < wreg?
RETFIE                  ;no then return
MOVPF   TMR0L, TMPLO    ;read low tmr0
MOVPF   TMR0H, TMPHI    ;read high tmr0
RETFIE                  ;return
```

Interrupts must be disabled during this subroutine.

11.3.2 WRITING A 16-BIT VALUE TO TMR0

Since writing to either TMR0L or TMR0H will effectively inhibit increment of that half of the TMR0 in the next cycle (following write), but not inhibit increment of the other half, the user must write to TMR0L first and TMR0H next in two consecutive instructions, as shown in Example 11-2. The interrupt must be disabled. Any write to either TMR0L or TMR0H clears the prescaler.

EXAMPLE 11-2: 16-BIT WRITE

```
BSF   CPUSTA, GLINTD ; Disable interrupt
MOVFP RAM_L, TMR0L   ;
MOVFP RAM_H, TMR0H   ;
BCF   CPUSTA, GLINTD ; Done, enable interrupt
```

11.4 Prescaler Assignments

Timer0 has an 8-bit prescaler. The prescaler assignment is fully under software control; i.e., it can be changed "on the fly" during program execution. When changing the prescaler assignment, clearing the prescaler is recommended before changing assignment. The value of the prescaler is "unknown", and assigning a value that is less then the present value makes it difficult to take this unknown time into account.

FIGURE 11-4: TMR0 TIMING: WRITE HIGH OR LOW BYTE

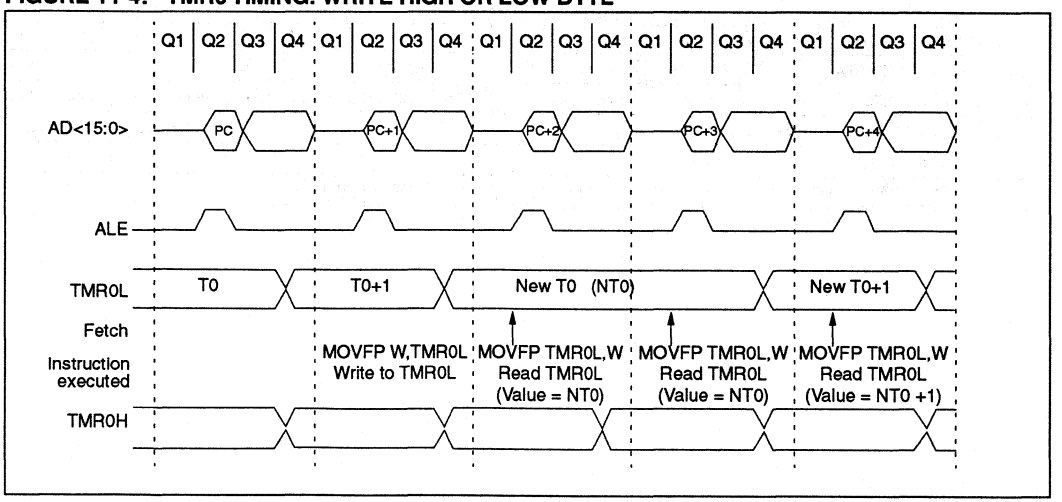
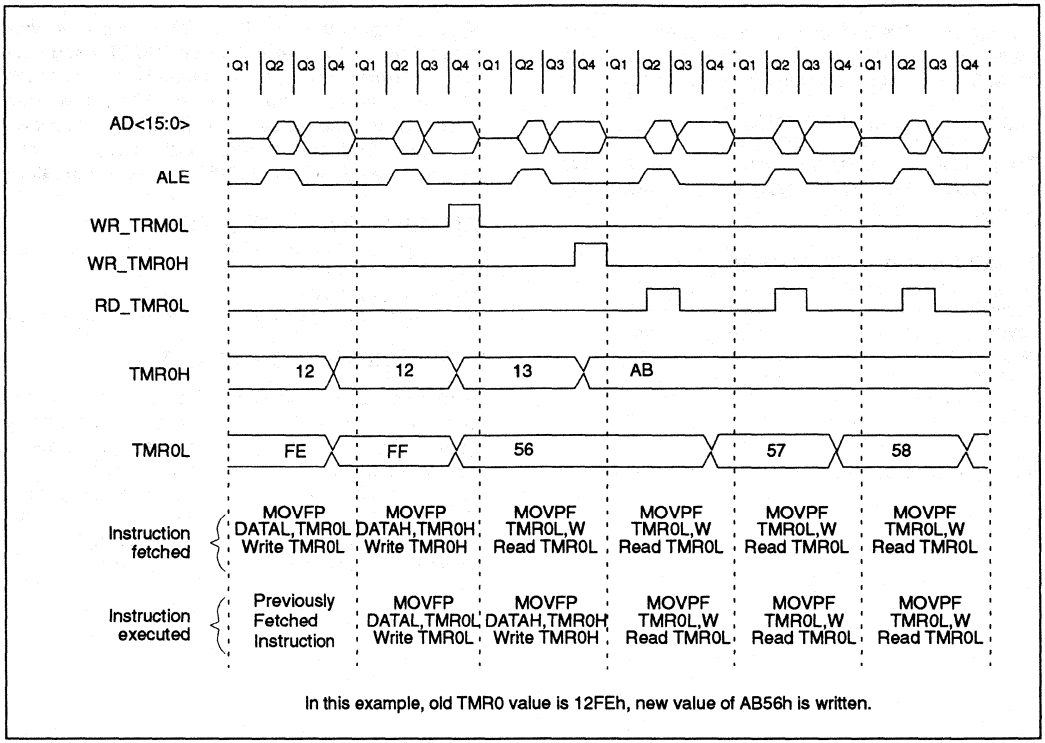


FIGURE 11-5: TMR0 READ/WRITE IN TIMER MODE



In this example, old TMR0 value is 12FEh, new value of AB56h is written.

TABLE 11-1: REGISTERS/BITS ASSOCIATED WITH TMR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
05h, Unbanked	T0STA	INTEDG	T0SE	T0CS	PS3	PS2	PS1	PS0	—	0000 000-	0000 000-
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 ??--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
0Bh, Unbanked	TMR0L	Timer0 low byte								xxxx xxxx	uuuu uuuu
0Ch, Unbanked	TMR0H	Timer0 high byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0', ? - Value depends on condition.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer time-out reset.
2: Shaded cells are not used by TMR0.

© 1995 Microchip Technology Inc.

12.0 TIMER1, TIMER2, TIMER3, PWMS AND CAPTURES

The PIC17C4X has a wealth of timers and time-based functions to ease the implementation of control applications. These time-base functions include two PWM outputs and two Capture inputs.

Timer1 (TMR1) and Timer2 (TMR2) are two 8-bit incrementing timers, each with a period register (PR1 and PR2 respectively) and separate overflow interrupt flags. TMR1 and TMR2 can operate either as timers (increment on internal OSC/4 clock) or as counters (increment on falling edge of external clock on pin RB4/TCLK12). They are also software configurable to operate as a single 16-bit timer. These timers are also used as the time-base for the PWM (pulse width modulation) module.

TMR3 is a 16-bit timer/counter consisting of the TMR3H and TMR3L registers. This timer has four other associated registers. Two registers are used as a 16-bit period register or a 16-bit Capture1 register (PR3H/CA1H:PR3L/CA1L). The other two registers are strictly the Capture2 registers (CA2H:CA2L). Timer3 is the time-base for the two 16-bit captures.

Timer3 can be software configured to increment from the internal system clock or from an external signal on the RB5/TCLK3 pin.

Figure 12-1 and Figure 12-2 are the control registers for the operation of Timer1, Timer2, and Timer3, as well as PWM1, PWM2, Capture1, and Capture2.

2

FIGURE 12-1: TCON1 REGISTER (ADDRESS: 16H, BANK 3)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7-6: **CA2ED1:CA2ED0**: Capture2 Mode Select bits
00 = Capture on every falling edge
01 = Capture on every rising edge
10 = Capture on every 4th rising edge
11 = Capture on every 16th rising edge

bit 5-4: **CA1ED1:CA1ED0**: Capture1 Mode Select bits
00 = Capture on every falling edge
01 = Capture on every rising edge
10 = Capture on every 4th rising edge
11 = Capture on every 16th rising edge

bit 3: **T16**: Timer1:Timer2 Mode Select bit
1 = Timer1 and Timer2 form a 16-bit timer
0 = Timer1 and Timer2 are two 8-bit timers

bit 2: **TMR3CS**: Timer3 Clock Source Select bit
1 = Timer3 increments off the falling edge of the RB5/TCLK3 pin
0 = Timer3 increments off the internal clock

bit 1: **TMR2CS**: Timer2 Clock Source Select bit
1 = Timer2 increments off the falling edge of the RB4/TCLK12 pin
0 = Timer2 increments off the internal clock

bit 0: **TMR1CS**: Timer1 Clock Source Select bit
1 = Timer1 increments off the falling edge of the RB4/TCLK12 pin
0 = Timer1 increments off the internal clock

© 1995 Microchip Technology Inc.

FIGURE 12-2: TCON2 REGISTER (ADDRESS: 17H, BANK 3)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0
CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset

bit 7: **CA2OVF**: Capture2 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (CA2H:CA2L) before the next capture event occured. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the Timer3 value until the capture register has been read (both bytes) and this bit is cleared in software.
1 = Overflow occurred on Capture2 register
0 = No overflow occurred on Capture2 register

bit 6: **CA1OVF**: Capture1 Overflow Status bit
This bit indicates that the capture value had not been read from the capture register pair (PR3H/CA2H:PR3L/CA2L) before the next capture event occured. The capture register retains the oldest unread capture value (last capture before overflow). Subsequent capture events will not update the capture register with the Timer3 value until the capture register has been read (both bytes) and this bit is cleared in software.
1 = Overflow occurred on Capture1 register
0 = No overflow occurred on Capture1 register

bit 5: **PWM2ON**: PWM2 On bit
1 = PWM2 is enabled (The RB3/PWM2 pin ignores the state of the DDRB<3> bit)
0 = PWM2 is disabled (The RB3/PWM2 pin uses the state of the DDRB<3> bit for data direction)

bit 4: **PWM1ON**: PWM1 On bit
1 = PWM1 is enabled (The RB2/PWM1 pin ignores the state of the DDRB<2> bit)
0 = PWM1 is disabled (The RB2/PWM1 pin uses the state of the DDRB<2> bit for data direction)

bit 3: **CA1/$\overline{PR3}$**: CA1/$\overline{PR3}$ Register Mode Select bit
1 = Enables Capture1 (PR3H/CA1H:PR3L/CA1L is the Capture1 register. Timer3 runs without a period register)
0 = Enables the Period register (PR3H/CA1H:PR3L/CA1L is the Period register for Timer3)

bit 2: **TMR3ON**: Timer3 On bit
1 = Starts Timer3
0 = Stops Timer3

bit 1: **TMR2ON**: Timer2 On bit
This bit controls the incrementing of the Timer2 register. When Timer2:Timer1 form the 16-bit timer (T16 is set), TMR2ON must be set. This allows the MSB of the timer to increment.
1 = Starts Timer2 (Must be enabled if the T16 bit (TCON1<3>) is set)
0 = Stops Timer2

bit 0: **TMR1ON**: Timer1 On bit
When T16 is set (in 16-bit Timer Mode)
1 = Starts 16-bit Timer2:Timer1
0 = Stops 16-bit Timer2:Timer1

When T16 is clear (in 8-bit Timer Mode)
1 = Starts 8-bit Timer1
0 = Stops 8-bit Timer1

© 1995 Microchip Technology Inc.

12.1 Timer1 and Timer2

12.1.1 TIMER1, TIMER2 IN 8-BIT MODE

Both Timer1 and Timer2 will operate in 8-bit mode when the T16 bit is clear. These two timers can be independently configured to increment from the internal instruction cycle clock or from an external clock source on the RB4/TCLK12 pin. The timer clock source is configured by the TMRxCS bit (x = 1 for Timer1 or = 2 for Timer2). When TMRxCS is clear, the clock source is internal and increments once every instruction cycle (OSC/4). When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge of the RB4/TCLK12 pin.

The timer increments from 00h until it equals the Period register (PRx). It then resets to 00h at the next increment cycle. The timer interrupt flag is set when the timer is reset. Timer1 and Timer2 have individual interrupt flag bits. The Timer1 interrupt flag bit is latched into TMR1IF, and the Timer2 interrupt flag bit is latched into TMR2IF.

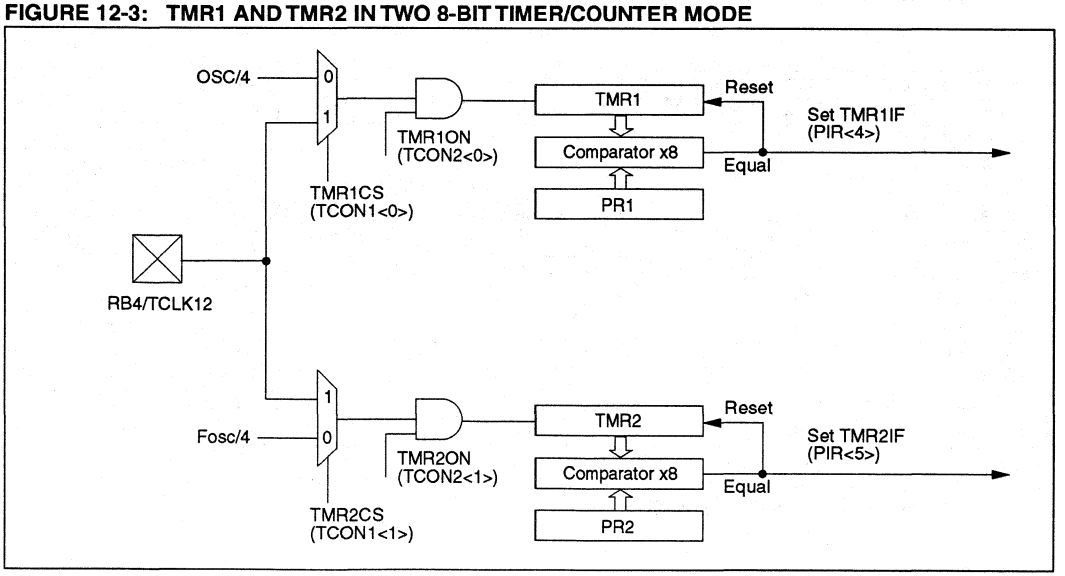
Each timer also has a corresponding interrupt enable bit (TMRxIE). The timer interrupt can be enabled by setting this bit and disabled by clearing this bit. For peripheral interrupts to be enabled, the Peripheral Interrupt Enable bit must be enabled (PEIE is set) and global interrupts must be enabled (GLINTD is cleared).

The timers can be turned on and off under software control. When the Timerx on control bit (TMRxON) is set, the timer increments from the clock source. When TMRxON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

12.1.1.1 EXTERNAL CLOCK INPUT FOR TMR1 OR TMR2

When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge on the RB4/TCLK12 pin. The TCLK12 input is synchronized with internal phase clocks. This causes a delay from the time a falling edge appears on TCLK12 to the time TMR1 or TMR2 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

FIGURE 12-3: TMR1 AND TMR2 IN TWO 8-BIT TIMER/COUNTER MODE



12.1.2 TIMER1 & TIMER2 IN 16-BIT MODE

To select 16-bit mode, the T16 bit must be set. In this mode TMR1 and TMR2 are concatenated to form a 16-bit timer (TMR2:TMR1). The 16-bit timer increments until it matches the 16-bit period register (PR2:PR1). On the following timer clock, the timer value is reset to 0h, and the TMR1IF bit is set.

When selecting the clock source for the16-bit timer, the TMR1CS bit controls the entire 16-bit timer and TMR2CS is a "don't care". When TMR1CS is clear, the timer increments once every instruction cycle (OSC/4). When TMR1CS is set, the timer increments on every falling edge of the RB4/TCLK12 pin. For the 16-bit timer to increment, both TMR1ON and TMR2ON bits must be set (see Table 12-1).

12.1.2.1 EXTERNAL CLOCK INPUT FOR TMR1:TMR2

When TMR1CS is set, the 16-bit TMR2:TMR1 increments on the falling edge of clock input TCLK12. The input on the RB4/TCLK12 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on RB4/TCLK12 to the time TMR2:TMR1 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

TABLE 12-1: TURNING ON 16-BIT TIMER

TMR2ON	TMR1ON	Result
1	1	16-bit timer (TMR2:TMR1) ON
0	1	Only TMR1 increments
x	0	16-bit timer OFF

FIGURE 12-4: TMR1 AND TMR2 IN 16-BIT TIMER/COUNTER MODE

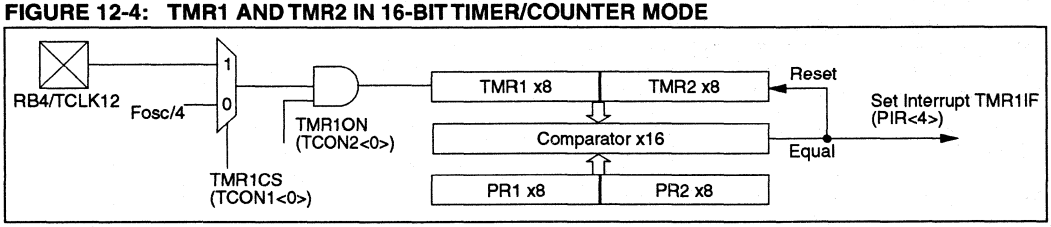


TABLE 12-2: SUMMARY OF TIMER1 AND TIMER2 REGISTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA2IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	---	---	STKAV	GLINTD	TO	PD	---	---	--11 11--	--11 ??--
14h, Bank 2	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	---	---	---	---	---	---	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	---	---	---	---	---	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0', ? - Value depends on condition.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and WDT Timer time-out reset.
2: Shaded cells are not used by TMR1 and TMR2.

© 1995 Microchip Technology Inc.

12.1.3 USING PULSE WIDTH MODULATION (PWM) OUTPUTS WITH TMR1 AND TMR2

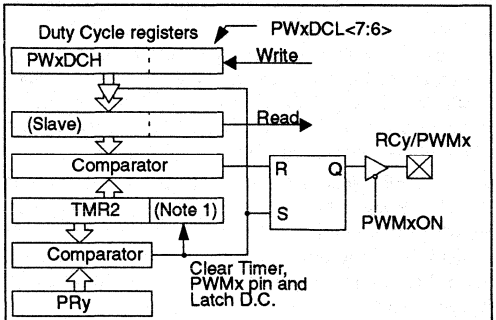
Two high speed pulse width modulation (PWM) outputs are provided. The PWM1 output uses Timer1 as its time-base, while PWM2 may be software configured to use either Timer1 or Timer 2 as the time-base. The PWM outputs are on the RB2/PWM1 and RB3/PWM2 pins.

Each PWM output has a maximum resolution of 10-bits. At 10-bit resolution, the PWM output frequency is 24.4 kHz (@ 25 MHz clock) and at 8-bit resolution the PWM output frequency is 97.7 kHz. The duty cycle of the output can vary from 0% to 100%.

Figure 12-5 shows a simplified block diagram of the PWM module. The duty cycle register is double buffered for glitch free operation. Figure 12-6 shows how a glitch could occur if the duty cycle registers were not double buffered.

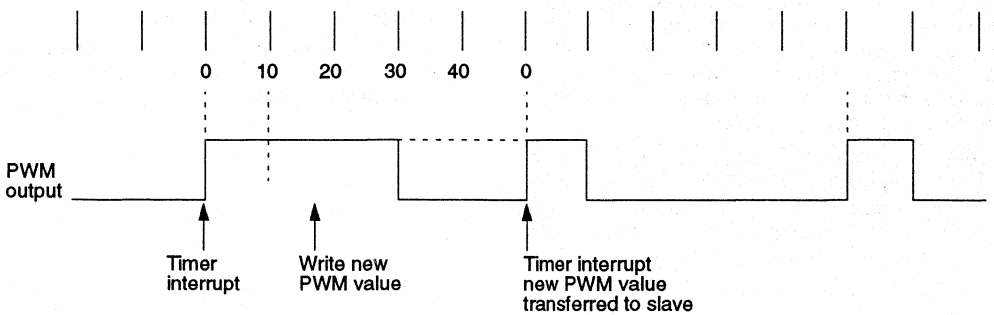
The user needs to set the PWM1ON bit (TCON2<4>) to enable the PWM1 output. When the PWM1ON bit is set, the RB2/PWM1 pin is configured as PWM1 output and forced as an output irrespective of the data direction bit (DDRB<2>). When the PWM1ON bit is clear, the pin behaves as a port pin and its direction is controlled by its data direction bit (DDRB<2>). Similarly, the PWM2ON (TCON2<5>) bit controls the configuration of the RB3/PWM2 pin.

FIGURE 12-5: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time-base.

FIGURE 12-6: PWM OUTPUT



Note 1: The dotted line shows PWM output if duty cycle registers were not double buffered. If the new duty cycle is written after the timer has passed that value, then the PWM does not reset at all during the current cycle causing a "glitch".

2: In this example, PWM period = 50. Old duty cycle is 30. New duty cycle value is 10.

12.1.3.1 PWM PERIODS

The period of the PWM1 output is determined by Timer1 (TMR1) and its period register (PR1). The period of the PWM2 output can be software configured to use either Timer1 or Timer2 as the time-base. When TM2PW2 bit (PW2DCL<5>) is clear, the time-base is determined by TMR1 and PR1. When TM2PW2 is set, the time-base is determined by TMR2 and PR2.

Running two different PWM outputs on two different timers allows different PWM periods. Running both PWMs from Timer1 allows the best use of resources by freeing Timer2 to operate as an 8-bit timer. Timer1 and Timer2 can not be used as a 16-bit timer if either PWM is being used.

The PWM periods can be calculated as follows:

period of PWM1 =[(PR1) + 1] x 4 TOSC

period of PWM2 =[(PR1) + 1] x 4 TOSC or
[(PR2) + 1] x 4 TOSC

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and the lower 2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>). Table 12-3 shows the maximum PWM frequency given the value in the period register. The PWMx duty cycle is as follows:

PWMx Duty Cycle = (DCx) x TOSC

where DCx represents the 10-bit value from PWxDCH:PWxDCL.

If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four Q-clock (depending on the state of the PWxDCL<7:6> bits). For a Duty Cycle to be 100%, the PWxDCH value must be greater then the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR1 (or TMR2) overflows and a new PWM period begins, the master latch values are transferred to the slave latches and the PWMx pin is forced high.

Note: For PW1DCH, PW1DCL, PW2DCH and PW2DCL registers, a write operation writes to the "master latches" while a read operation reads the "slave latches". As a result, the user may not read back what was just written to the duty cycle registers.

The user should also avoid any "read-modify-write" operations on the duty cycle registers, such as: `ADDWF PW1DCH`. This may cause duty cycle outputs that are unpredictable.

TABLE 12-3: PWM FREQUENCY vs. RESOLUTION AT 25 MHz

PWM Frequency	Frequency (kHz)				
	24.4	48.8	65.104	97.66	390.6
PRx Value	0xFF	0x7F	0x5F	0x3F	0x0F
High Resolution	10-bit	9-bit	8.5-bit	8-bit	6-bit
Standard Resolution	8-bit	7-bit	6.5-bit	6-bit	4-bit

12.1.3.2 PWM INTERRUPTS

The PWM module makes use of Timer1 or Timer2 interrupts. A timer interrupt is generated when TMR1 or TMR2 equals its period register and is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer roll-over. The Timer1 interrupt is latched into the TMR1IF bit and the Timer2 interrupt is latched into the TMR2IF bit . These flags must be cleared in software.

12.1.3.3 EXTERNAL CLOCK SOURCE

The PWMs will operate regardless of the clock source of the timer. The use of an external clock has ramifications that must be understood. Because the external TCLK12 input is synchronized internally (sampled once per instruction cycle), the time TCLK12 changes to the time the timer increments will vary by as much as TCY (one instruction cycle). This will cause jitter in the duty cycle as well as the period of the PWM output.

This jitter will be ±TCY, unless the external clock is synchronized with the processor clock. Use of one of the PWM outputs as the clock source to the TCLKx input, will supply a synchronized clock.

In general, when using an external clock source for PWM, its frequency should be much less than the device frequency (Fosc).

© 1995 Microchip Technology Inc.

12.1.3.3.1 MAX RESOLUTION/FREQUENCY FOR EXTERNAL CLOCK INPUT

The use of an external clock for the PWM time-base (Timer1 or Timer2) limits the PWM output to a maximum resolution of 8-bits. The PWxDCL<7:6> bits must be kept cleared. Use of any other value will distort the PWM output. All resolutions are supported when internal clock mode is selected. The maximum attainable frequency is also lower. This is a result of the timing requirements of an external clock input for a timer (see the Electrical Specification section). The maximum PWM frequency, when the timers clock source is the RB4/TCLK12 pin, is shown in Table 12-3 (standard resolution mode).

12.2 Timer3

TMR3 is a 16-bit timer consisting of the TMR3H and TMR3L registers. TMR3H is the high byte of the timer and TMR3L is the low byte. This timer has an associated 16-bit period register (PR3H/CA1H:PR3L/CA1L). This period register can be software configured to be a second 16-bit capture register.

When the TMR3CS bit (TCON1<2>) is clear, the timer increments every instruction cycle (OSC/4). When TMR3CS is set, the timer increments on every falling edge of the RB5/TCLK3 pin. In either mode, the TMR3ON bit must be set for the timer to increment. When TMR3ON is clear, the timer will not increment or set the TMR3IF bit.

TMR3 has two modes of operation, depending on the CA1/$\overline{PR3}$ bit (TCON2<3>). These modes are:

- One capture and one period register mode
- Dual capture register mode

The PIC17C4X has up to two 16-bit capture registers that capture the 16-bit value of TMR3 when events are detected on capture pins. There are two capture pins (RB0/CAP1 and RB1/CAP2), one for each capture register. The capture pins are multiplexed with PORTB pins. An event can be:

- a rising edge
- a falling edge
- 4 rising edges
- 16 rising edges

Each 16-bit capture register has an interrupt flag associated with it. The flag is set when a capture is made. The capture module is truly part of the Timer3 block. Figure 12-7 and Figure 12-8 show the block diagrams for the two modes of operation.

TABLE 12-4: REGISTERS/BITS ASSOCIATED WITH PWM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 ??--
10h, Bank 3	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as '0', ? = value depends on conditions.
Note 1: Shaded cells are not used by PWM.

12.2.1 ONE CAPTURE AND ONE PERIOD REGISTER MODE

In this mode registers PR3H/CA1H and PR3L/CA1L constitute a 16-bit period register. A block diagram is shown in Figure 12-7. The timer increments until it equals the period register and then resets to 0000h. TMR3 Interrupt Flag bit (TMR3IF) is set at this point. This interrupt can be disabled by clearing the TMR3 Interrupt Enable bit (TMR3IE). TMR3IF must be cleared in software.

This mode is selected if control bit CA1/PR3 is clear. In this mode, the Capture1 register, consisting of high byte (PR3H/CA1H) and low byte (PR3L/CA1L), is configured as the period control register for TMR3. Capture1 is disabled in this mode, and the corresponding Interrupt bit CA1IF is never set. Timer3 increments until it equals the value in the period register and then resets to 0000h.

Capture2 is active in this mode. The CA2ED1 and CA2ED0 bits determine the event on which capture will occur. The possible events are:

- Capture on every falling edge
- Capture on every rising edge
- Capture every 4th rising edge
- Capture every 16th rising edge

When a capture takes place, an interrupt flag is latched into the CA2IF bit. This interrupt can be enabled by setting the corresponding mask bit CA2IE. The Peripheral Interrupt Enable bit (PEIE) must be set and the Global Interrupt Disable bit (GLINTD) must be cleared for the interrupt to be acknowledged. The CA2IF interrupt flag bit must be cleared in software.

When the capture prescale select is changed, the prescaler is not reset and an event may be generated. Therefore, the first capture after such a change will be ambiguous. However, it sets the time-base for the next capture. The prescaler is reset upon chip reset.

Capture pin RB1/CAP2 is a multiplexed pin. When used as a port pin, Capture2 is not disabled. However, the user can simply disable the Capture2 interrupt by clearing CA2IE. If RB1/CAP2 is used as an output pin, the user can activate a capture by writing to the port pin. This may be useful during development phase to emulate a capture interrupt.

The input on capture pin RB1/CAP2 is synchronized internally to internal phase clocks. This imposes certain restrictions on the input waveform (see the Electrical Specification section for timing).

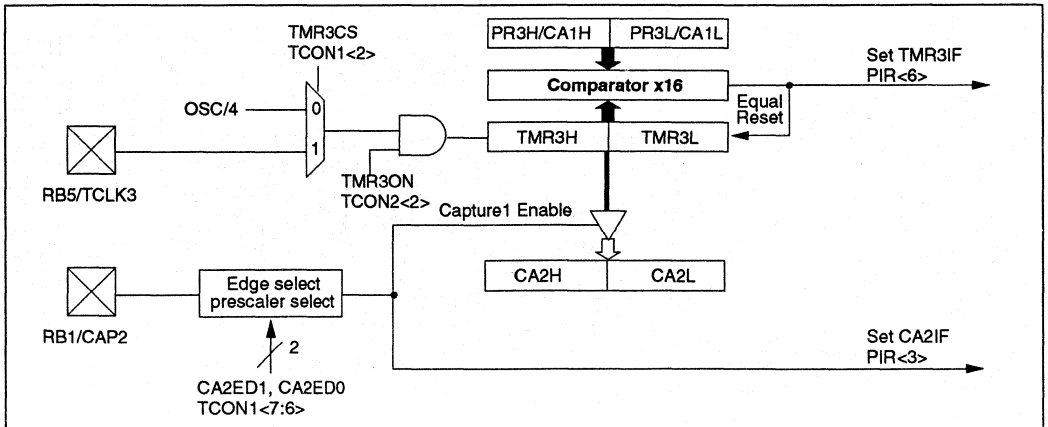
The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another "event" has occurred on the RB1/CA2 pin. The new event will not transfer the Timer3 value to the capture register, protecting the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The recommended sequence to read capture registers and capture overflow flag bits is shown in Example 12-1.

EXAMPLE 12-1: SEQUENCE TO READ CAPTURE REGISTERS

```
MOVLB   3               ; Select Bank 3
MOVPF   CA2L, LO_BYTE   ; Read Capture2 low
                        ; byte, store in LO_BYTE
MOVPF   CA2H, HI_BYTE   ; Read Capture2 high
                        ; byte, store in HI_BYTE
MOVPF   TCON2, STAT_VAL ; Read TCON2 into file
                        ; STAT_VAL
```

FIGURE 12-7: TIMER3 WITH ONE CAPTURE AND ONE PERIOD REGISTER BLOCK DIAGRAM



© 1995 Microchip Technology Inc.

12.2.2 DUAL CAPTURE1 REGISTER MODE

This mode is selected by setting CA1/$\overline{PR3}$. A block diagram is shown in Figure 12-8. In this mode, TMR3 runs without a period register and increments from 0000h to FFFFh and rolls over to 0000h. The Timer3 interrupt Flag (TMR3IF) is set on this roll over. The TMR3IF bit must be cleared in software.

Registers PR3H/CA1H and PR3L/CA1L make a 16-bit capture register (Capture1). It captures events on pin RB0/CAP1. Capture mode is configured by the CA1ED1 and CA1ED0 bits. Capture1 Interrupt Flag bit (CA1IF) is set on the capture event. The corresponding interrupt mask bit is CA1IE . The Capture1 overflow status bit is CA1OVF.

The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another "event" has occurred on the RB1/CA2 pin. The new event will not transfer the Timer3 value to the capture register which protects the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The operation of the Capture1 feature is identical to Capture2 (as described in Section 12.2.1).

2

FIGURE 12-8: TIMER3 WITH TWO CAPTURE REGISTERS BLOCK DIAGRAM

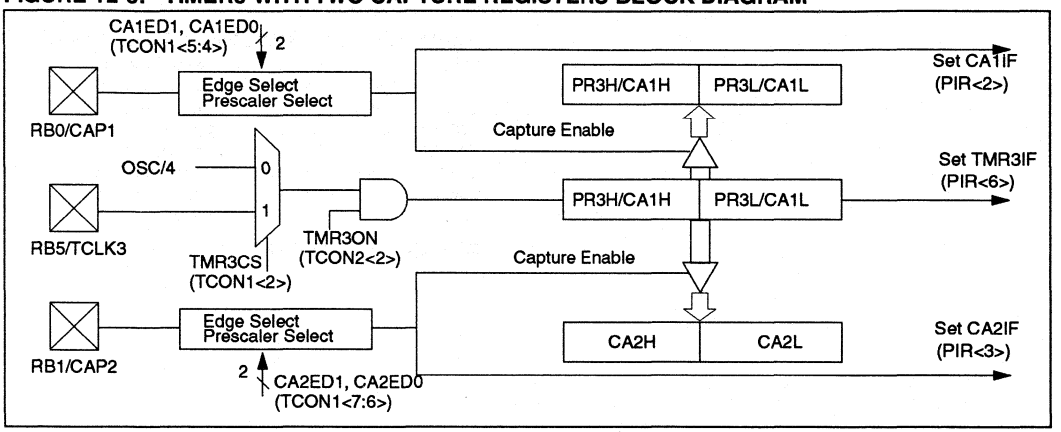


TABLE 12-5: REGISTERS ASSOCIATED WITH CAPTURE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
12h, Bank 2	TMR3L	Timer3 low byte								xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	Timer3 high byte								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	TO	PD	—	—	--11 11--	--11 ??--
16h, Bank 2	PR3L/CA1L	Timer3 period register, low byte/capture1 register, low byte								xxxx xxxx	uuuu uuuu
17h, Bank 2	PR3H/CA1H	Timer3 period register, high byte/capture1 register, high byte								xxxx xxxx	uuuu uuuu
14h, Bank 3	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h, Bank 3	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as '0'. ? - Value depends on condition.
Note 1: Shaded cells are not used by Capture.

12.2.3 EXTERNAL CLOCK INPUT FOR TIMER3

When TMR3CS is set, the 16-bit TMR3 increments on the falling edge of clock input TCLK3. The input on the RB5/TCLK3 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on TCLK3 to the time TMR3 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section. Figure 12-9 shows the timing diagram when operating from an external clock.

12.2.4 READING/WRITING TIMER3

Since Timer3 is a 16-bit timer and only 8-bits at a time can be read or written, care should be taken when reading or writing while the timer is running. The best method to read or write the timer is to stop the timer, perform any read or write operation, and then restart Timer3 (using the TMR3ON bit). However, if it is necessary to keep Timer3 free-running, care must be taken. For writing to the 16-bit Timer3, Example 12-2 may be used. For reading the 16-bit Timer3, Example 12-3 may be used.

EXAMPLE 12-2: WRITING TO TMR3

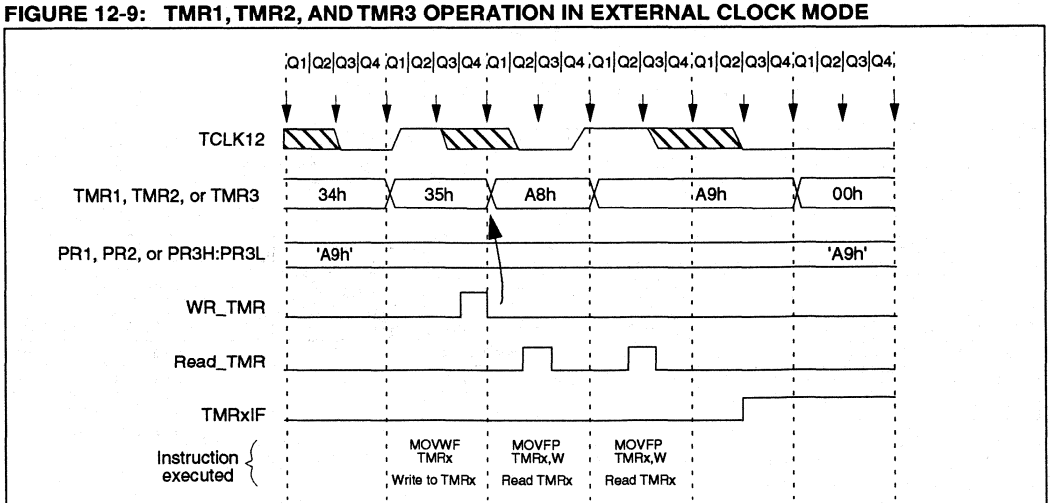
```
BSF   CPUSTA, GLINTD ; Disable interrupt
MOVFP RAM_L, TMR3L   ;
MOVFP RAM_H, TMR3H   ;
BCF   CPUSTA, GLINTD ; Done, enable interrupt
```

EXAMPLE 12-3: READING FROM TMR3

```
MOVPF   TMR3L, TMPLO    ;read low tmr0
MOVPF   TMR3H, TMPHI    ;read high tmr0
MOVFP   TMPLO, WREG     ;tmplo -> wreg
CPFSLT  TMR3L, WREG     ;tmr0l < wreg?
RETFIE                  ;no then return
MOVPF   TMR3L, TMPLO    ;read low tmr0
MOVPF   TMR3H, TMPHI    ;read high tmr0
RETFIE                  ;return
```

Interrupts must be disabled during this subroutine.

FIGURE 12-9: TMR1, TMR2, AND TMR3 OPERATION IN EXTERNAL CLOCK MODE



Note 1: TCLK12 is sampled in Q2 and Q4.
2: ↓ indicates a sampling point.
3: The latency from TCLK12 ↓ to timer increment is between 2 Tosc and 6 Tosc.

© 1995 Microchip Technology Inc.

FIGURE 12-10: TMR1, TMR2, AND TMR3 OPERATION IN TIMER MODE

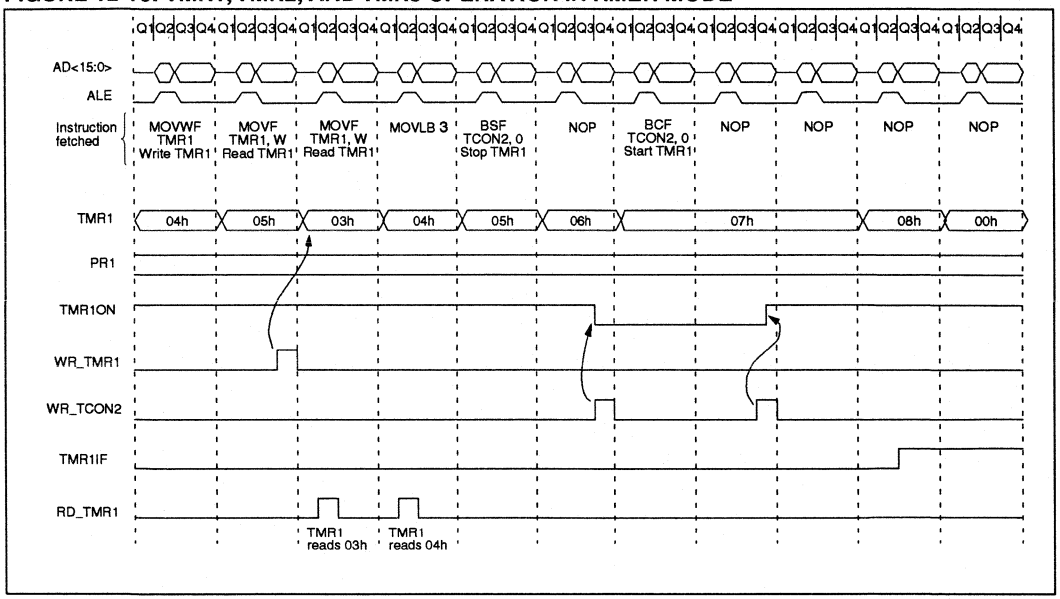


TABLE 12-6: SUMMARY OF TMR1, TMR2, AND TMR3 REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{PR3}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2								xxxx xxxx	uuuu uuuu
12h, Bank 2	TMR3L	Timer3 low byte								xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	Timer3 high byte									
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	TO	PD	—	—	--11 11--	--11 ??--
14h, Bank 2	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
16h, Bank 2	PR3L/CA1L	Timer3 period register, low byte/capture1 register, low byte								xxxx xxxx	uuuu uuuu
17h, Bank 2	PR3H/CA1H	Timer3 period register, high byte/capture1 register, high byte								xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
14h, Bank 3	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h, Bank 3	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as '0'. ? - Value depends on condition.
Note 1: Shaded cells are not used by TMR1, TMR2 or TMR3.

NOTES:

© 1995 Microchip Technology Inc.

13.0 SERIAL COMMUNICATION INTERFACE (SCI) MODULE

The Serial Communication Interface (SCI) module is a serial I/O module. The SCI (USART) can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc. The SCI can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

The SPEN (RCSTA<7>) bit has to be set in order to configure RC6 and RC7 as the Serial Communication Interface.

The SCI module will control the direction of the RA4/RX/DT and RA5/TX/CK pins, depending on the states of the SCI configuration bits in the RCSTA and TXSTA registers. The bits that control I/O direction are:

- SPEN
- TXEN
- SREN
- CREN
- CSRC

The Transmit Status And Control Register is shown in Figure 13-1, while the Receive Status And Control Register is shown in Figure 13-2.

2

FIGURE 13-1: TXSTA REGISTER (ADDRESS: 15H, BANK 0)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0	U - 0	R - 1	R/W - x
CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8
bit7							bit0

R = Readable bit
W = Writable bit
-n = Value at POR reset
(x = unknown)

bit 7: **CSRC**: Clock Source Select bit
Synchronous mode:
1 = Master Mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)
Asynchronous mode:
Don't care

bit 6: **TX8/9**: Transmit Data Length bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5: **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
SREN/CREN overrides TXEN in SYNC mode

bit 4: **SYNC**: SCI mode Select bit
(Synchronous/Asynchronous)
1 = Synchronous mode
0 = Asynchronous mode

bit 3-2: **Unimplemented**, reads as '0'

bit 1: **TRMT**: Transmit Shift Register (TSR) Empty bit
1 = TSR empty
0 = TSR full

bit 0: **TXD8**: 9th bit of transmit data (can be used to calculated the parity in software)

© 1995 Microchip Technology Inc.

FIGURE 13-2: RCSTA REGISTER (ADDRESS: 13H, BANK 0)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	U - 0	R - 0	R - 0	R - x
SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8
bit7							bit 0

R = Readable bit
W = Writable bit
-n = Value at POR reset
(x = unknown)

bit 7: **SPEN**: Serial Port Enable bit
1 = Configures RC7/RX/DT and RC6/TX/CK pins as serial port pins
0 = Serial port disabled

bit 6: **RC8/9**: Receive Data Length bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit
This bit enables the reception of a single byte. After receiving the byte, this bit is automatically cleared.
Synchronous mode:
1 = Enable reception
0 = Disable reception
Note: This bit is ignored in synchronous slave reception.
Asynchronous mode:
Don't care

bit 4: **CREN**: Continuous Receive Enable bit
This bit enables the continuous reception of serial data.
Asynchronous mode:
1 = Enable reception
0 = Disables reception
Synchronous mode:
1 = Enables continuous reception until CREN is cleared (CREN overrides SREN)
0 = Disables continuous reception

bit 3: **Unimplemented**, reads as '0'

bit 2: **FERR**: Framing Error bit
1 = Framing error (Updated by reading RCREG)
0 = No framing error

bit 1: **OERR**: Overrun Error bit
1 = Overrun (Cleared by clearing CREN)
0 = No overrun error

bit 0: **RCD8**: 9th bit of receive data (can be the software calculated parity bit)

© 1995 Microchip Technology Inc.

FIGURE 13-3: SCI TRANSMIT

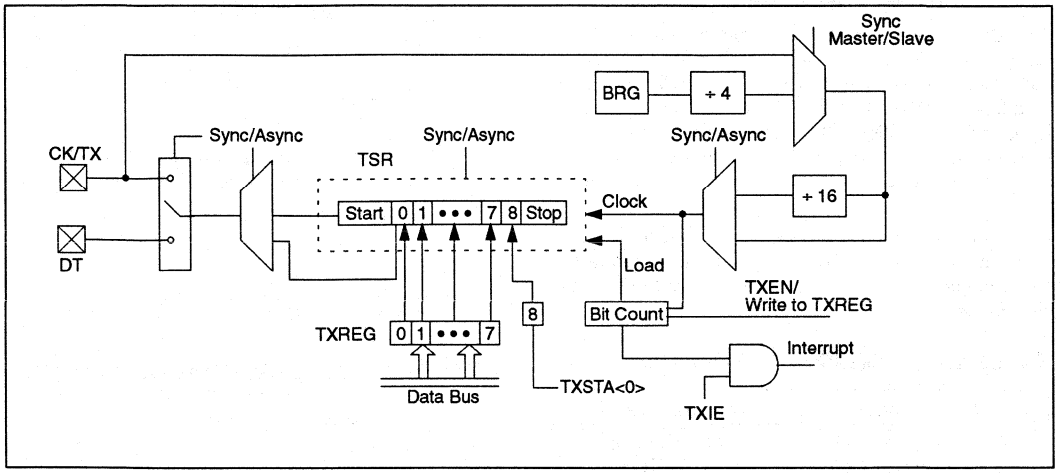
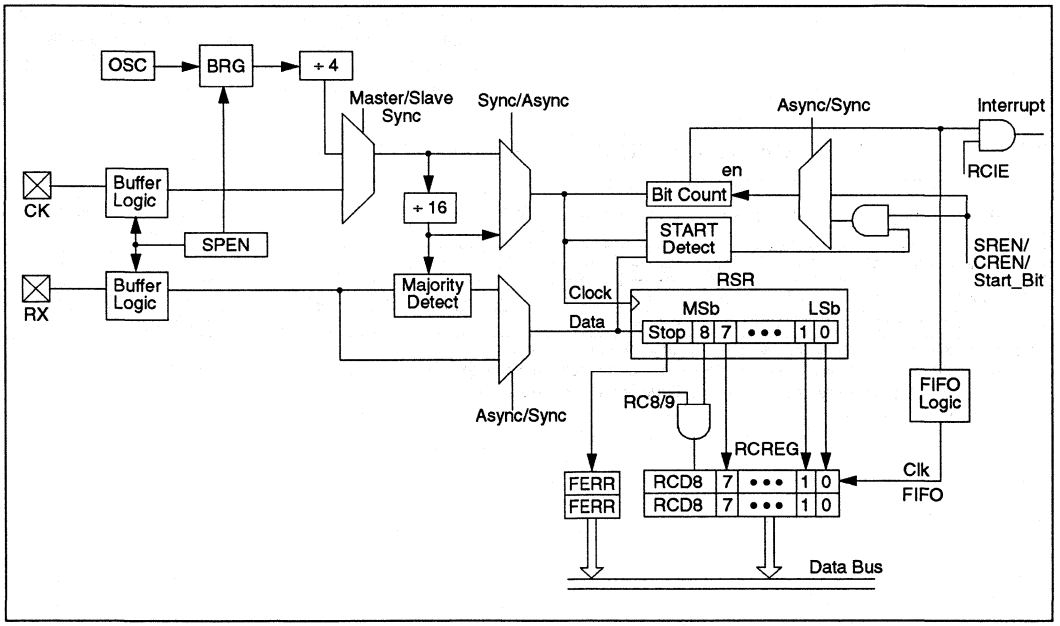


FIGURE 13-4: SCI RECEIVE



13.1 SCI Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the SCI. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. Table 13-1 shows the formula for computation of the baud rate for different SCI modes. These only apply when the SCI is in synchronous master mode (internal clock) and asynchronous mode.

Given the desired baud rate and Fosc, the nearest integer value between 0 and 255 can be calculated using the formula below. The error in baud rate can then be determined.

TABLE 13-1: BAUD RATE FORMULA

SYNC	Mode	Baud Rate
0	Asynchronous	Fosc/(64(X+1))
1	Synchronous	Fosc/(4(X+1))

X = value in SPBRG (0 to 255)

Example 13-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz
Desired Baud Rate = 9600
SYNC = 0

EXAMPLE 13-1: CALCULATING BAUD RATE ERROR

Desired Baud rate=Fosc / (64 (X + 1))

9600 = 16000000 /(64 (X + 1))

X = 25.042 = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

$$\text{Error} = \frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}}$$

= (9615 - 9600) / 9600

= 0.16%

Writing a new value to the SPBRG, causes the BRG timer to be reset (or cleared), this guarantees that the BRG does not wait for a timer overflow before outputting the new baud rate.

TABLE 13-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used by the Baud Rate Generator.

© 1995 Microchip Technology Inc.

TABLE 13-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 25 MHz		SPBRG value (decimal)	Fosc = 20 MHz		SPBRG value (decimal)	Fosc = 16 MHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	NA	—	—	NA	—	—	NA	—	—
19.2	NA	—	—	19.53	+1.73	255	19.23	+0.16	207
76.8	77.16	+0.47	80	76.92	+0.16	64	76.92	+0.16	51
96	96.15	+0.16	64	96.15	+0.16	51	95.24	-0.79	41
300	297.62	-0.79	20	294.1	-1.96	16	307.69	+2.56	12
500	480.77	-3.85	12	500	0	9	500	0	7
HIGH	6250	—	0	5000	—	0	4000	—	0
LOW	24.41	—	255	19.53	—	255	15.625	—	255

BAUD RATE (K)	Fosc = 10 MHz		SPBRG value (decimal)	Fosc = 7.159 MHz		SPBRG value (decimal)	Fosc = 5.068 MHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.766	+1.73	255	9.622	+0.23	185	9.6	0	131
19.2	19.23	+0.16	129	19.24	+0.23	92	19.2	0	65
76.8	75.76	-1.36	32	77.82	+1.32	22	79.2	+3.13	15
96	96.15	+0.16	25	94.20	-1.88	18	97.48	+1.54	12
300	312.5	+4.17	7	298.3	-0.57	5	316.8	+5.60	3
500	500	0	4	NA	—	—	NA	—	—
HIGH	2500	—	0	1789.8	—	0	1267	—	0
LOW	9.766	—	255	6.991	—	255	4.950	—	255

BAUD RATE (K)	Fosc = 3.579 MHz		SPBRG value (decimal)	Fosc = 1 MHz		SPBRG value (decimal)	Fosc = 32.768 kHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	NA	—	—	NA	—	—	0.303	+1.14	26
1.2	NA	—	—	1.202	+0.16	207	1.170	-2.48	6
2.4	NA	—	—	2.404	+0.16	103	NA	—	—
9.6	9.622	+0.23	92	9.615	+0.16	25	NA	—	—
19.2	19.04	-0.83	46	19.24	+0.16	12	NA	—	—
76.8	74.57	-2.90	11	83.34	+8.51	2	NA	—	—
96	99.43	_3.57	8	NA	—	—	NA	—	—
300	298.3	-0.57	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	894.9	—	0	250	—	0	8.192	—	0
LOW	3.496	—	255	0.976	—	255	0.032	—	255

© 1995 Microchip Technology Inc.

TABLE 13-4: BAUD RATES FOR ASYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 25 MHz		SPBRG value (decimal)	Fosc = 20 MHz		SPBRG value (decimal)	Fosc = 16 MHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	1.221	+1.73	255	1.202	+0.16	207
2.4	2.396	0.14	162	2.404	+0.16	129	2.404	+0.16	103
9.6	9.53	-0.76	40	9.469	-1.36	32	9.615	+0.16	25
19.2	19.53	+1.73	19	19.53	+1.73	15	19.23	+0.16	12
76.8	78.13	+1.73	4	78.13	+1.73	3	83.33	+8.51	2
96	97.65	+1.73	3	104.2	+8.51	2	NA	—	—
300	390.63	+30.21	0	312.5	+4.17	0	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	390.6	—	0	312.5	—	0	250	—	0
LOW	1.53	—	255	1.221	—	255	0.977	—	255

BAUD RATE (K)	Fosc = 10 MHz		SPBRG value (decimal)	Fosc = 7.159 MHz		SPBRG value (decimal)	Fosc = 5.068 MHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	NA	—	—	NA	—	—	0.31	+3.13	255
1.2	1.202	+0.16	129	1.203	_0.23	92	1.2	0	65
2.4	2.404	+0.16	64	2.380	-0.83	46	2.4	0	32
9.6	9.766	+1.73	15	9.322	-2.90	11	9.9	-3.13	7
19.2	19.53	+1.73	7	18.64	-2.90	5	19.8	+3.13	3
76.8	78.13	+1.73	1	NA	—	—	79.2	+3.13	0
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	156.3	—	0	111.9	—	0	79.2	—	0
LOW	0.610	—	255	0.437	—	255	0.309	—	255

BAUD RATE (K)	Fosc = 3.579 MHz		SPBRG value (decimal)	Fosc = 1 MHz		SPBRG value (decimal)	Fosc = 32.768 kHz		SPBRG value (decimal)
	KBAUD	%ERROR		KBAUD	%ERROR		KBAUD	%ERROR	
0.3	0.301	+0.23	185	0.300	+0.16	51	0.256	-14.67	1
1.2	1.190	-0.83	46	1.202	+0.16	12	NA	—	—
2.4	2.432	+1.32	22	2.232	-6.99	6	NA	—	—
9.6	9.322	-2.90	5	NA	—	—	NA	—	—
19.2	18.64	-2.90	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	55.93	—	0	15.63	—	0	0.512	—	0
LOW	0.218	—	255	0.061	—	255	0.002	—	255

© 1995 Microchip Technology Inc.

13.2 SCI Asynchronous Mode

In this mode, the SCI uses standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The SCI's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock 64x of the bit shift rate. Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

The asynchronous mode is selected by clearing the SYNC bit (TXSTA<4>).

The SCI Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

13.2.1 SCI ASYNCHRONOUS TRANSMITTER

The SCI transmitter block diagram is shown in Figure 13-3. The heart of the transmitter is the transmit shift register (TSR). The shift register obtains its data from the read/write transmit buffer (TXREG). TXREG is loaded with data in software. The TSR is not loaded until the stop bit has been transmitted from the previous load. As soon as the stop bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one TCY at the end of the current BRG cycle), the TXREG is empty and an interrupt bit, TXIF (PIR<1>) is set. This interrupt can be enabled or disabled by the TXIE bit (PIE<1>). TXIF will be set regardless of TXIE and cannot be reset in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of the TXREG, the TRMT (TXSTA<1>) bit shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty.

Note: The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 13-5). The transmission can also be started by first loading TXREG and then setting TXEN. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 13-6). Clearing TXEN during a transmission will cause the transmission to be aborted. This will reset the transmitter and the RA5/TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to TXD8 (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG. This is because a data write to TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty).

Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by configuring the bits SYNC = 0 and SPEN = 1.
3. If interrupts are desired, then the TXIE bit should be set.
4. If 9-bit transmission is desired, then the TX8/9 bit should be set.
5. Load data to the TXREG register.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
7. Enable the transmission by setting TXEN (starts transmission).

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner then doing these two events in the opposite order.

Note: To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

© 1995 Microchip Technology Inc.

FIGURE 13-5: ASYNCHRONOUS MASTER TRANSMISSION

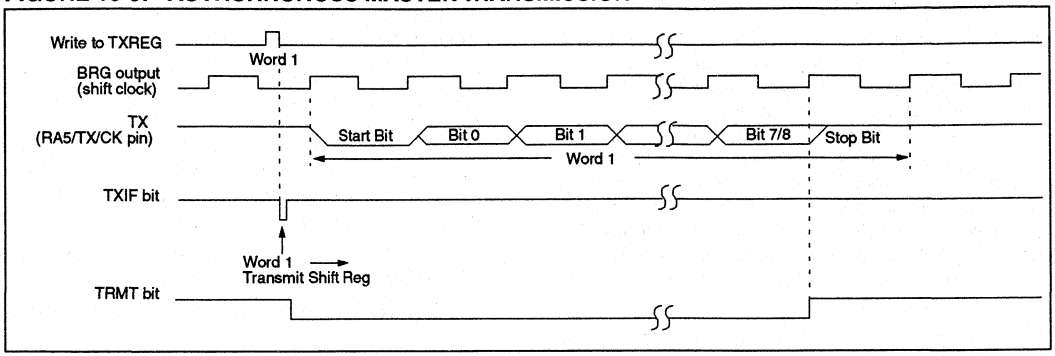
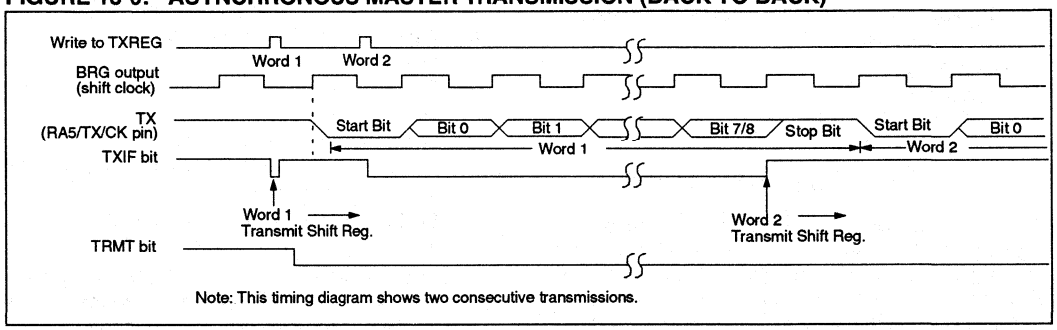


FIGURE 13-6: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 13-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
16h, Bank 0	TXREG	Serial port transmit register								xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used for asynchronous transmission.

© 1995 Microchip Technology Inc.

13.2.2 SCI ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 13-4. The data comes in the RA4/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once the asynchronous mode is selected, reception is enabled by setting CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the stop bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR<0>) is set. The actual interrupt can be enabled or disabled by the RCIE (PIE<0>) bit. RCIF is a read only bit which is reset by the hardware. It is cleared when RCREG has been read and is empty. RCREG is a double buffered register; (i.e. it is a two deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR. On detection of the stop bit of the third byte, if the RCREG is still full, then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software which is done by resetting the receive logic (CREN is set). If the OERR bit is set, transfers from the RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The framing error bit FERR (RCSTA<2>) is set if a stop bit is not detected.

Note: The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG register will load RCD8 and FERR with new values; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RCD8 information.

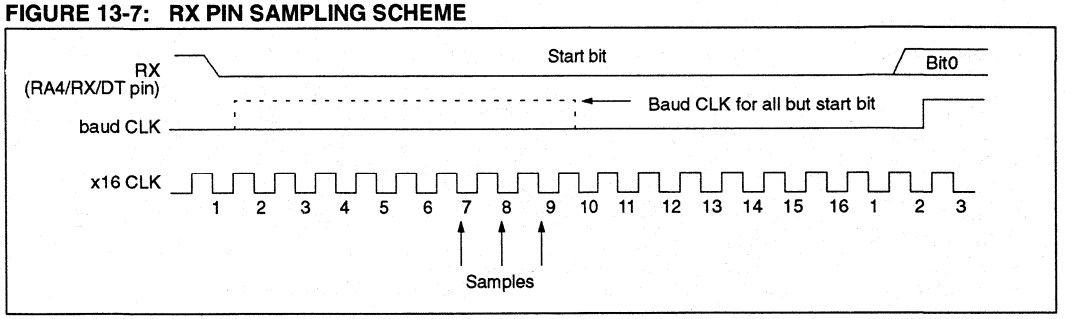
13.2.3 SAMPLING

The data on the RA4/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RA4/RX/DT pin. The sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (see Figure 11-3). These sample points have no relationship to the first falling edge of the start bit.

The x16 clock is a free running clock, and the three sample points occur at a frequency of every 16 falling edges.

2

FIGURE 13-7: RX PIN SAMPLING SCHEME

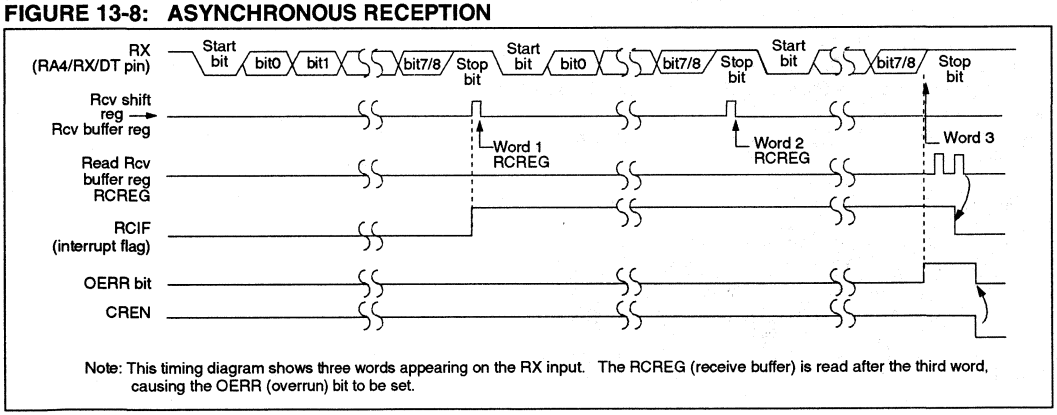


Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by configuring SYNC = 0 and SPEN = 1.
3. If interrupts are desired, then the RCIE bit should be set.
4. If 9-bit reception is desired, then the RX8/9 bit should be set.
5. Enable the reception by setting CREN.
6. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit is set.
7. Read RCSTA to get the ninth bit (if enabled) and FERR bit to determine if any error occurred during reception.
8. Read the 8-bit received data by reading RCREG.
9. If an overrun error occurred, clear the error by clearing the OERR bit.

Note: To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

FIGURE 13-8: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

TABLE 13-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used for asynchronous reception.

© 1995 Microchip Technology Inc.

13.3 SCI Synchronous Master Mode

In Master Synchronous mode, the data is transmitted in a half-duplex manner; i.e. transmission and reception do not occur at the same time: when transmitting data, the reception is inhibited and vice versa. The synchronous mode is entered by setting the SYNC (TXSTA<4>) bit. In addition, the SPEN (RCSTA<7>) bit is set in order to configure the RA5 and RA4 I/O ports to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting the CSRC (TXSTA<7>) bit.

13.3.1 SCI SYNCHRONOUS MASTER TRANSMISSION

The SCI transmitter block diagram is shown in Figure 13-3. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer TXREG. TXREG is loaded with data in software. The TSR is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from TXREG (if available). Once TXREG transfers the data to the TSR (occurs in one TCY at the end of the current BRG cycle), TXREG is empty and the TXIF (PIR<1>) bit is set. This interrupt can be enabled or disabled by the TXIE bit (PIE<1>). TXIF will be set regardless of TXIE and cannot be cleared in software. It will reset only when new data is loaded into TXREG. While TXIF indicates the status of TXREG, TRMT (TXSTA<1>) shows the status of the TSR. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting the TXEN (TXSTA<5>) bit. The actual transmission will not occur until TXREG has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the RA5/TX/CK pin. Data out is stable around the falling edge of the synchronous clock (Figure 13-10). The transmission can also be started by first loading TXREG and then setting TXEN. This is advantageous when slow baud rates are selected, since BRG is kept in RESET when TXEN=CREN=SREN=clear. Setting the TXEN bit will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR is empty, so a transfer to TXREG will result in an immediate transfer to the TSR, resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. The RA4/RX/DT and RA5/TX/CK pins will revert to hi-impedance. If either CREN or SREN are set during a transmission, the transmission is aborted and the RA4/RX/DT pin reverts to a hi-impedance state (for a reception). The RA5/TX/CK pin will remain an output if CSRC = 1 (internal clock). The transmitter logic is not reset, although it is disconnected from the pins. In order to reset the transmitter, the user has to clear TXEN. If the SREN bit is set (to interrupt an ongoing transmission and receive a single word), then after the single word is received, SREN will be cleared and the serial port will revert back to transmitting, since the TXEN bit is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, TXEN should be cleared.

In order to select 9-bit transmission, the TX8/9 (TXSTA<6>) bit should be set and the ninth bit should be written to TXD8 (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to TXREG. This is because a data write to TXREG can result in an immediate transfer of the data to the TSR (if the TSR is empty). If the TSR was empty and TXREG was written before writing the "new" TXD8, the "present" value of TXD8 is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

1. Initialize the SPBRG register for the appropriate baud rate (see Baud Rate Generator Section for details).
2. Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
3. Make sure CREN = SREN = 0 (these bits override transmission when set).
4. If interrupts are desired, then the TXIE bit should be set.
5. If 9-bit transmission is desired, then the TX8/9 bit should be set.
6. Start transmission by loading data to the TXREG register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
8. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner then doing these two events in the opposite order.

Note:	To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

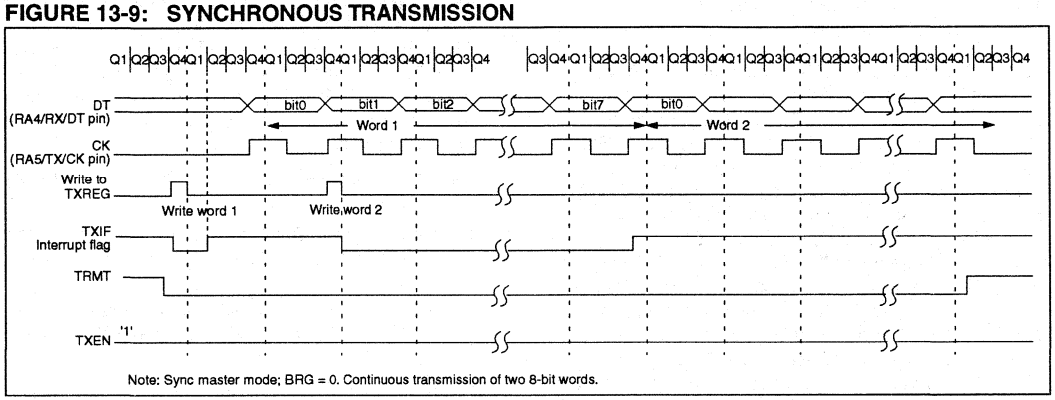
© 1995 Microchip Technology Inc.

TABLE 13-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
16h, Bank 0	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

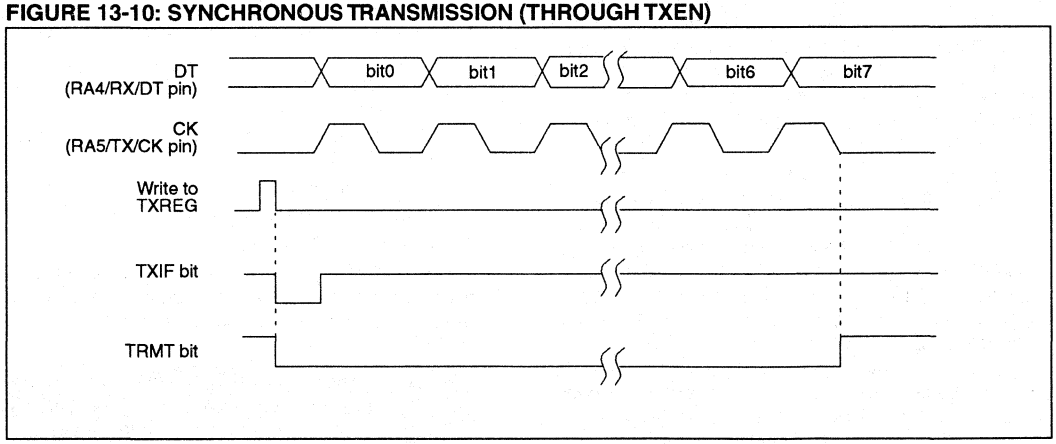
Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used for synchronous master transmission.

FIGURE 13-9: SYNCHRONOUS TRANSMISSION



Note: Sync master mode; BRG = 0. Continuous transmission of two 8-bit words.

FIGURE 13-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



© 1995 Microchip Technology Inc.

13.3.2 SCI SYNCHRONOUS MASTER RECEPTION

Once synchronous mode is selected, reception is enabled by setting either the SREN (RCSTA<5>) bit or the CREN (RCSTA<4>) bit. Data is sampled on the RA4/RX/DT pin on the falling edge of the clock. If SREN is set, then only a single word is received. If CREN is set, the reception is continuous until CREN is reset. If both bits are set, then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR<0>) is set. The actual interrupt can be enabled or disabled by the RCIE (PIE<0>) bit. RCIF is a read only bit which is reset by the hardware. In this case it is reset when RCREG has been read and is empty. RCREG is a double buffered register; i.e., it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR. On the clocking of the last bit of the third byte, if RCREG is still full, then the overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software. This is done by clearing the CREN bit. If OERR bit is set, transfers from RSR to RCREG are inhibited, so it is essential to clear OERR bit if it is set. The 9th receive bit is buffered the same way as the receive data. Reading RCREG will load RCD8 with a new value; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RCD8 information.

Steps to follow when setting up a Synchronous Master Reception:

1. Initialize the SPBRG register for the appropriate baud rate. See Section 13.1 for details.
2. Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 1.
3. If interrupts are desired, then the RCIE bit should be set.
4. If 9-bit reception is desired, then the RX8/9 bit should be set.
5. If a single reception is required, set SREN. For continuous reception set CREN.
6. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit were set.
7. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading RCREG.
9. If any error occurred, clear the error by clearing CREN.

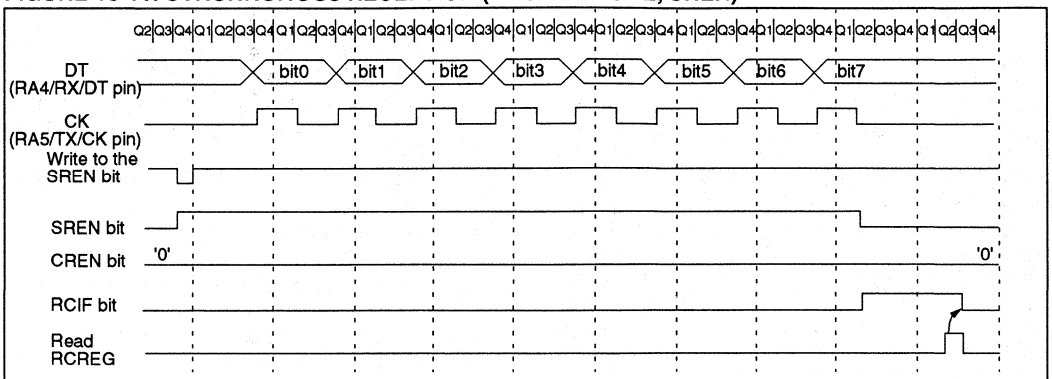
Note: To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

TABLE 13-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through MCLR and Watchdog Timer time-out reset.
2: Shaded cells are not used for synchronous master reception.

FIGURE 13-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates SYNC master mode with SREN = 1.

© 1995 Microchip Technology Inc.

13.4 SCI Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the RA5/TX/CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC (TXSTA<7>) bit.

13.4.1 SCI SYNCHRONOUS SLAVE TRANSMIT

The operation of the sync master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to TXREG and then the SLEEP instruction executes, the following will occur. The first word will immediately transfer to the TSR and transmit. The second word will remain in TXREG. TXIF will not be set. When the first word has been shifted out of TSR, TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
2. Make CREN = 0.
3. If interrupts are desired, then the TXIE bit should be set.
4. If 9-bit transmission is desired, then the TX8/9 bit should be set.
5. Start transmission by loading data to TXREG.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TXD8.
7. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner then doing these two events in the opposite order.

Note: To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

13.4.2 SCI SYNCHRONOUS SLAVE RECEPTION

Operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to RCREG and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by configuring the bits SYNC = 1, SPEN = 1 and CSRC = 0.
2. If interrupts are desired, then the RCIE bit should be set.
3. If 9-bit reception is desired, then the RX8/9 bit should be set.
4. To enable reception, set CREN = 1.
5. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
6. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading RCREG.
8. If any error occurred, clear the error by clearing CREN.

To abort reception, either clear the SPEN bit, the SREN bit (when in single receive mode), or the CREN bit (when in continuous receive mode). This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

© 1995 Microchip Technology Inc.

TABLE 13-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
16h, Bank 0	TXREG	TX7	TX6	TX5	TX4	TX3	TX2	TX1	TX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used for synchronous slave transmission.

TABLE 13-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
16h, Bank1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank0	RCSTA	SPEN	RC8/9	SREN	CREN	—	FERR	OERR	RCD8	0000 -00x	0000 -00u
14h, Bank0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX8/9	TXEN	SYNC	—	—	TRMT	TXD8	0000 --1x	0000 --1u
17h, Bank0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, reads as a '0'.
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used for synchronous slave reception.

© 1995 Microchip Technology Inc.

14.0 SPECIAL FEATURES OF THE CPU

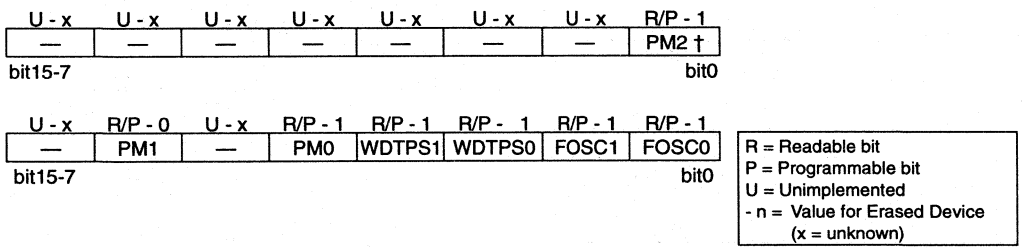
What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. The PIC17CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- OSC selection
- Reset
  - Power-On Reset (POR)
  - Power-Up Timer (PWRT)
  - Oscillator Start-Up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection

The PIC17CXX has a Watchdog Timer which can be shut off only through EPROM bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-Up Timer (OST), intended to keep the chip in RESET until the crystal oscillator is stable. The other is the Power-Up Timer (PWRT), which provides a fixed delay of 96 ms (nominal) on power up only, designed to keep the part in RESET while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake from SLEEP through external reset, watchdog timer time-out or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LF crystal option saves power. Configuration bits are used to select various options. This configuration word has the format shown in Figure 14-1.

FIGURE 14-1: CONFIGURATION WORD



bit 15-9: **Unimplemented**, reads as a '1'

bit 8,6,4: **PM2, PM1, PM0**, Processor Mode Select bits
111 =Microprocessor Mode
110 =Extended microcontroller mode
101 =Microcontroller mode
000 =Code protected microcontroller mode

bit 7, 5: **Unimplemented**, reads as a '0'

bit 3-2: **WDTPS1:WDTPS0**, WDT Postscaler Select bits
11 = WDT enabled, postscaler = 1
10 = WDT enabled, postscaler = 256
01 = WDT enabled, postscaler = 64
00 = WDT disabled, 16-bit overflow timer

bit 1-0: **FOSC1:FOSC0**, Oscillator Select bits
11 = EC oscillator
10 = XT oscillator
01 = RC oscillator
00 = LF oscillator

† This bit does not exist on the PIC17C42. Reading this bit will return an unknown value (x).

© 1995 Microchip Technology Inc.

14.1 Configuration Bits

The PIC17CXX has seven configuration locations (see Table 14-1). These locations can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. Any write to a configuration location, regardless of the data, will program that configuration bit. A TABLWT instruction is required to write to program memory locations. The configuration bits can be read by using the TABLRD instructions. Reading any configuration location between FE00h and FE07h will read the low byte of the configuration word (see Figure 14-1) into the TABLATL register. The TABLATH register will be FFh. Reading a configuration location between FE08h and FE0Fh will read the high byte of the configuration word into the TABLATL register. The TABLATH register will be FFh.

Addresses FE00h thru FE0Fh are only in the program memory space for microcontroller and code protected microcontroller modes. A device programmer will be able to read the configuration word in any processor mode. See programming specifications for more detail.

TABLE 14-1: CONFIGURATION LOCATIONS

Bit	Address
FOSC0	FE00h
FOSC1	FE01h
WDTPS0	FE02h
WDTPS1	FE03h
PM0	FE04h
PM1	FE06h
PM2 †	FE08h †

† This location does not exist on the PIC17C42.

Note: When programming the desired configuration locations, they must be programmed in ascending order. Starting with address FE00h.

14.2 Oscillator Configurations

14.2.1 OSCILLATOR TYPES

The PIC17CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

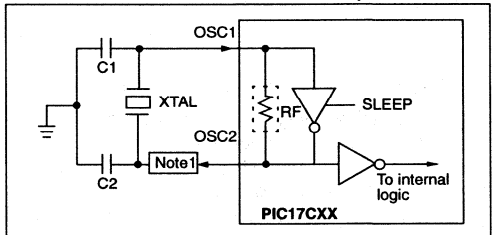
- LF: Low Power Crystal
- XT: Crystal/Resonator
- EC: External Clock Input
- RC: Resistor/Capacitor

14.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT or LF modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 14-2). The PIC17CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

For frequencies above 20 MHz, it is common for the crystal to be an overtone mode crystal. Use of overtone mode crystals require a tank circuit to attenuate the gain at the fundamental frequency. Figure 14-3 shows an example of this.

FIGURE 14-2: CRYSTAL OR CERAMIC RESONATOR OPERATION (XT OR LF OSC CONFIGURATION)

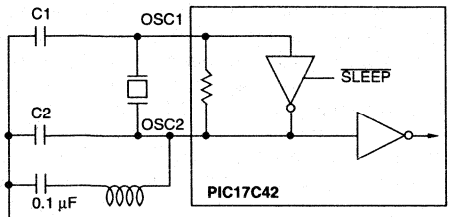


See Table 14-2 and Table 14-3 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

© 1995 Microchip Technology Inc.

FIGURE 14-3: CRYSTAL OPERATION, OVERTONE CRYSTALS (XT OSC CONFIGURATION)



To filter the fundamental frequency

$$\frac{1}{LC2} = (2\pi f)^2$$

Where f = tank circuit resonant frequency. This should be midway between the fundamental and the 3rd overtone frequencies of the crystal.

TABLE 14-2: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Oscillator Type	Resonator Frequency	Capacitor Range C1 = C2
LF	455 kHz 2.0 MHz	15 - 68 pF 10 - 33 pF
XT	4.0 MHz 8.0 MHz 16.0 MHz	22 - 68 pF 33 - 100 pF 33 - 100 pF

Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

Resonators Used:

455 kHz	Panasonic EFO-A455K04B	+/-0.3%
2.0 MHz	Murata Erie CSA2.00MG	+/-0.5%
4.0 MHz	Murata Erie CSA4.00MG	+/-0.5%
8.0 MHz	Murata Erie CSA8.00MT	+/-0.5%
16.0 MHz	Murata Erie CSA16.00MX	+/-0.5%

Resonators used did not have built-in capacitors.

TABLE 14-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Freq	C1	C2
LF	32 kHz[1] 1 MHz 2 MHz	100-150 pF 10-33 pF 10-33 pF	100-150 pF 10-33 pF 10-33 pF
XT	2 MHz 4 MHz 8 MHz [2] 16 MHz 25 MHz	47-100 pF 15-68 pF 15-47 pF TBD 15-47 pF	47-100 pF 15-68 pF 15-47 pF TBD 15-47 pF

Higher capacitance increases the stability of the oscillator but also increases the start-up time and the oscillator current. These values are for design guidance only. Rs may be required in XT mode to avoid overdriving the crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values for external components.

Note 1: For VDD > 4.5V, C1 = C2 ≈ 30 pf is recommended.

2: Rs of 330Ω is required for a capacitor combination of 15/15 pF.

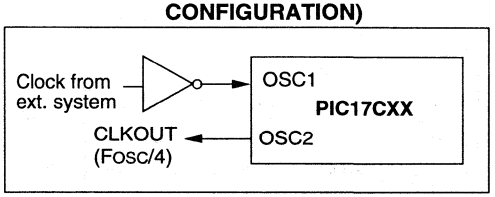
Crystals Used:

32.768 kHz	Epson C-001R32.768K-A	± 20 PPM
1.0 MHz	ECS ECS-10-13-2	± 50 PPM
2.0 MHz	ECS ECS-20-S-1	± 50 PPM
4.0 MHz	ECS ECS-40-S-4	± 50 PPM
8.0 MHz	ECS ECS-80-S-4	± 50 PPM
16.0 MHz	TBD	TBD
25 MHz	CTS CTS25M	± 50 PPM

14.2.3 EXTERNAL CLOCK OSCILLATOR

In the EC oscillator mode, the OSC1 input can be driven by CMOS drivers. In this mode, the OSC1/CLKIN pin is hi-impedance and the OSC2/CLKOUT pin is the CLKOUT output (4 TOSC).

FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (EC OSC CONFIGURATION)



14.2.4 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

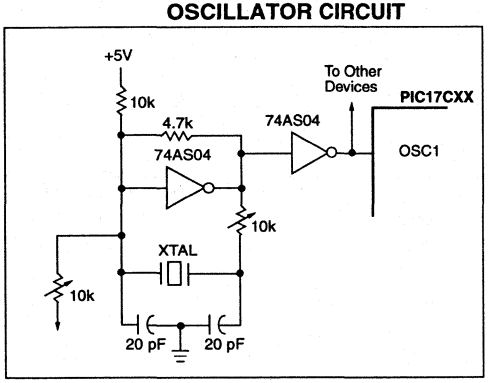
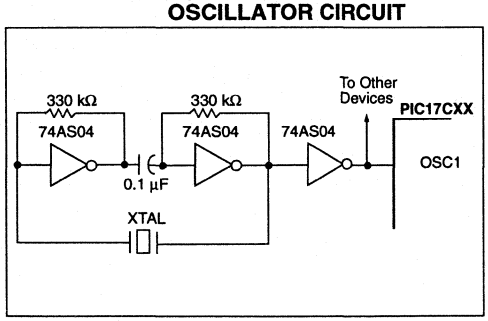


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



14.2.5 RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-6 shows how the R/C combination is connected to the PIC17CXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

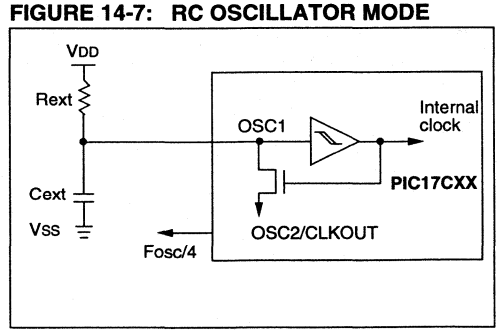
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 18.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 18.0 for variation of oscillator frequency due to VDD for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-2 for waveform).

FIGURE 14-7: RC OSCILLATOR MODE



© 1995 Microchip Technology Inc.

14.3 Watchdog Timer (WDT)

The Watchdog Timer's function is to recover from software malfunction. The WDT uses an internal free running on-chip RC oscillator for its clock source. This does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation and SLEEP mode, a WDT time-out generates a device RESET. The WDT can be permanently disabled by programming the configuration bits WDTPS1:WDTPS0 as '00' (Section 14.1).

Under normal operation, the WDT must be cleared on a regular interval. This time is less the the minimum WDT overflow time. Not clearing the WDT in this timeframe will cause the WDT to overflow and reset the device.

14.3.1 WDT PERIOD

The WDT has a nominal time-out period of 12 ms, (with postscaler = 1). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a postscaler with a division ratio of up to 1:256 can be assigned to the WDT. Thus, typical time-out periods up to 3.0 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler (if assigned to the WDT) and prevent it from timing out thus generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a WDT time-out.

14.3.2 CLEARING THE WDT AND POSTSCALER

The WDT and postscaler are cleared when:

- The device is in the reset state
- A SLEEP instruction is executed
- A CLRWDT instruction is executed
- Wake-up from SLEEP by an interrupt

The WDT counter/postscaler will start counting on the first edge after the device exits the reset state.

14.3.3 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT postscaler) it may take several seconds before a WDT time-out occurs.

The WDT and postscaler is the Power-Up timer during the Power-On Reset sequence.

14.3.4 WDT AS NORMAL TIMER

When the WDT is selected as a normal timer, the clock source is the device clock. Neither the WDT nor the postscaler are directly readable or writable. The overflow time is 65536 Tosc cycles. On overflow, the $\overline{TO}$ bit is cleared (device is not reset). The CLRWDT instruction can be used to set the $\overline{TO}$ bit. This allows this timer to be a simple overflow timer. When in sleep, this timer is stopped.

© 1995 Microchip Technology Inc.

FIGURE 14-8: WATCHDOG TIMER BLOCK DIAGRAM

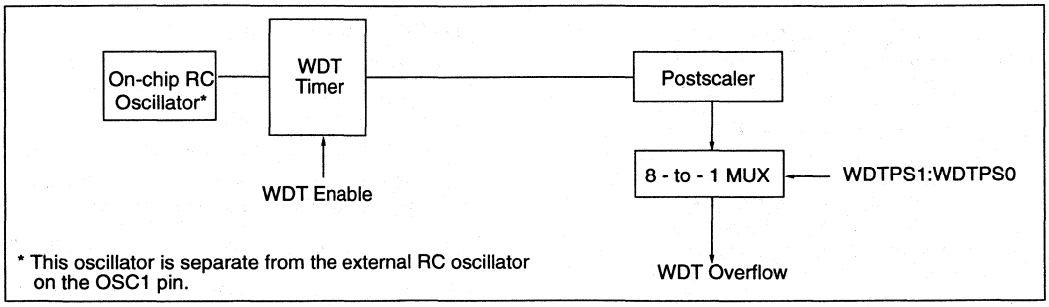


TABLE 14-4: REGISTERS/BITS ASSOCIATED WITH THE WATCHDOG TIMER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on all other resets (Note1)
—	Config	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0	(Note 3)	(Note 3)
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{TO}$	$\overline{PD}$	—	—	--11 11--	--11 ??--

Legend: - = Unimplemented, read as '0', ? - Value depends on condition.
Note 1: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and Watchdog Timer time-out reset.
2: Shaded cells are not used by the WDT.
3: This value will be as the device was programmed, or if unprogrammed, will read as all '1's.

© 1995 Microchip Technology Inc.

14.4 Power-Down Mode (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction. This clears the Watchdog Timer and postscaler (if enabled). The $\overline{PD}$ bit is cleared and the $\overline{TO}$ bit is set (in the CPUSTA register). In sleep mode, the oscillator driver is turned off. The I/O ports maintain their status (driving high, low, or hi-impedance).

The $\overline{MCLR}$/VPP pin must be at a logic high level (VIHMC). A WDT time-out RESET does not drive the $\overline{MCLR}$/VPP pin low.

14.4.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

- A POR reset
- External reset input on $\overline{MCLR}$/VPP pin
- WDT time-out reset (if WDT was enabled)
- Interrupt from RA0/INT pin, RB port change, T0CKI interrupt, or some Peripheral Interrupts

The following peripheral interrupts can wake-up from SLEEP:

- Capture1 interrupt
- Capture2 interrupt
- SCI synchronous slave transmit interrupt
- SCI synchronous slave receive interrupt

Other peripherals can not generate interrupts since during SLEEP, no on-chip Q clocks are present.

Any reset event will cause a device reset. Any interrupt event is considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the CPUSTA register can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ bit is cleared if WDT time-out occurred (and caused wake-up).

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GLINTD bit. If the GLINTD bit is set (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GLINTD bit is clear (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt vector address. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

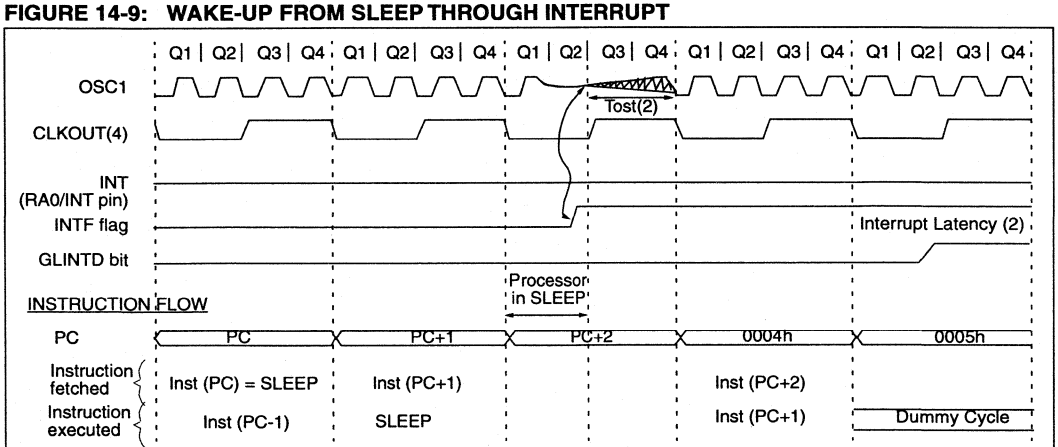
Note: If the global interrupts are disabled (GLINTD is set), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from sleep. The $\overline{TO}$ bit is set, and the $\overline{PD}$ bit is cleared.

The WDT is cleared when the device wake from SLEEP, regardless of the source of wake-up.

14.4.1.1 WAKE-UP DELAY

When the oscillator type is configured in XT or LF mode, the Oscillator Start-Up Timer (OST) is activated on wake-up. The OST will keep the device in reset for 1024 TOSC. This needs to be taken into account when considering the interrupt response time when coming out of SLEEP.

FIGURE 14-9: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: XT or LF oscillator mode assumed.
2: Tost = 1024 Tosc (drawing not to scale). This delay will not be there for RC osc mode.
3: When GLINTD = 0 processor jumps to interrupt routine after wake-up. If GLINTD = 1, execution will continue in line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

14.4.2 MINIMIZING CURRENT CONSUMPTION

To minimize current consumption, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should be at VDD or VSS. The contributions from on-chip pull-ups on PORTB should also be considered, and disabled when possible.

14.5 Code Protection

The code in the program memory can be protected by selecting the microcontroller in code protected mode (PM2, PM1, PM0 = '000').

Note:	PM2 does not exist on the PIC17C42. To select code protected microcontroller mode, PM1, PM0 = '00'.

In this mode, instructions that are in the on-chip program memory space, can continue to read or write the program memory. An instruction that is executed outside of the internal program memory range will be inhibited from writing to or reading from program memory.

Note:	Microchip does not recommend code protecting windowed devices. This may inhibit the device from being able to be reprogrammed.

© 1995 Microchip Technology Inc.

15.0 INSTRUCTION SET SUMMARY

Each PIC17CXX instruction set consists of 58 instructions. Each instruction is a 16-bit word divided into an OPCODE and one or more operands. The opcode specifies the instruction type, while the operand(s) further specify the operation of the instruction. The PIC17CXX instruction set can be grouped into three types:

- byte-oriented
- bit-oriented
- literal and control operations.

These formats are shown in Figure 15-1.

Table 15-1 shows the field descriptions for the opcodes. These descriptions are useful for understanding the opcodes in Table 15-2 and in each specific instruction descriptions.

byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' = '0', the result is placed in the WREG register. If 'd' = '1', the result is placed in the file register specified by the instruction.

bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

The instruction set is highly orthogonal and is grouped into:

- byte-oriented operations
- bit-oriented operations
- literal and control operations

All instructions are executed within one single instruction cycle, unless:

- a conditional test is true
- the program counter is changed as a result of an instruction
- a table read or a table write instruction is executed (in this case, the execution takes two instruction cycles with the second cycle executed as a NOP)

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 25 MHz, the normal instruction execution time is 160 ns. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 320 ns.

TABLE 15-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (00h to FFh)
p	Peripheral register file address (00h to 1Fh)
i	Table pointer control i = '0' (do not change) i = '1' (increment after instruction execution)
t	Table byte select t = '0' (perform operation on lower byte) t = '1' (perform operation on upper byte literal field, constant data)
WREG	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= '0' or '1') The assembler will generate code with x = '0'. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select 0 = store result in WREG 1 = store result in file register f Default is d = '1'
u	Unused, encoded as '0'
s	Destination select 0 = store result in file register f and in the WREG 1 = store result in file register f Default is s = '1'
label	Label name
C, DC, Z, OV	ALU status bits Carry, Digit Carry, Zero, Overflow
GLINTD	Global Interrupt Disable bit (CPUSTA<4>)
TBLPTR	Table Pointer (16-bit)
TBLAT	Table Latch (16-bit) consists of high byte (TBLATH) and low byte (TBLATL)
TBLATL	Table Latch low byte
TBLATH	Table Latch high byte
TOS	Top of Stack
PC	Program Counter
BSR	Bank Select Register
WDT	Watchdog Timer Counter
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination either the WREG register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

© 1995 Microchip Technology Inc.

Table 15-2 lists the instructions recognized by the MPASM assembler.

Note 1: Any unused opcode is Reserved. Use of any reserved opcode may cause unexpected operation.

Note 2: The shaded instructions are not available in the PIC17C42

All instruction examples use the following format to represent a hexadecimal number:

0xhh

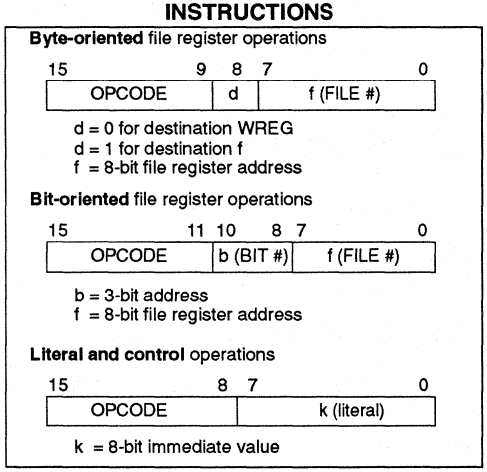
where h signifies a hexadecimal digit.

To represent a binary number:

0000 0100b

where b signifies a binary string.

FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS



15.1 Special Function Registers as Source/Destination

The PIC17C4X's orthogonal instruction set allows read and write of all file registers, including special function registers. There are some special situations the user should be aware of:

15.1.1 ALUSTA AS DESTINATION

If an instruction writes to ALUSTA, the Z, C, DC and OV bits may be set or cleared as a result of the instruction and overwrite the original data bits written. For example, executing `CLRF ALUSTA` will clear register ALUSTA, and then set the Z bit leaving `0000 0100b` in the register.

15.1.2 PCL AS SOURCE OR DESTINATION

Read, write or read-modify-write on PCL may have the following results:

Read PC:	PCH → PCLATH; PCL → dest
Write PCL:	PCLATH → PCH; 8-bit destination value → PCL
Read-Modify-Write:	PCL→ ALU operand PCLATH → PCH; 8-bit result → PCL

Where PCH = program counter high byte (not an addressable register), PCLATH = Program counter high holding latch, dest = destination, WREG or f.

15.1.3 BIT MANIPULATION

All bit manipulation instructions are done by first reading the entire register, operating on the selected bit and writing the result back (read-modify-write). The user should keep this in mind when operating on special function registers, such as ports.

© 1995 Microchip Technology Inc.

TABLE 15-2: PIC 17CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-bit Opcode MSb LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS						
ADDWF	f,d	ADD WREG to f	1	0000 111d ffff ffff	OV,C,DC,Z	
ADDWFC	f,d	ADD WREG and Carry bit to f	1	0001 000d ffff ffff	OV,C,DC,Z	
ANDWF	f,d	AND WREG with f	1	0000 101d ffff ffff	Z	
CLRF	f,s	Clear f, or Clear f and Clear WREG	1	0010 100s ffff ffff	None	3
COMF	f,d	Complement f	1	0001 001d ffff ffff	Z	
CPFSEQ	f	Compare f with WREG, skip if f = WREG	1 (2)	0011 0001 ffff ffff	None	6,8
CPFSGT	f	Compare f with WREG, skip if f > WREG	1 (2)	0011 0010 ffff ffff	None	2,6,8
CPFSLT	f	Compare f with WREG, skip if f < WREG	1 (2)	0011 0000 ffff ffff	None	2,6,8
DAW	f,s	Decimal Adjust WREG Register	1	0010 111s ffff ffff	C	3
DECF	f,d	Decrement f	1	0000 011d ffff ffff	OV,C,DC,Z	
DECFSZ	f,d	Decrement f, skip if 0	1 (2)	0001 011d ffff ffff	None	6,8
DCFSNZ	f,d	Decrement f, skip if not 0	1 (2)	0010 011d ffff ffff	None	6,8
INCF	f,d	Increment f	1	0001 010d ffff ffff	OV,C,DC,Z	
INCFSZ	f,d	Increment f, skip if 0	1 (2)	0001 111d ffff ffff	None	6,8
INFSNZ	f,d	Increment f, skip if not 0	1 (2)	0010 010d ffff ffff	None	6,8
IORWF	f,d	Inclusive OR WREG with f	1	0000 100d ffff ffff	Z	
MOVFP	f,p	Move f to p	1	011p pppp ffff ffff	None	
MOVPF	p,f	Move p to f	1	010p pppp ffff ffff	Z	
MOVWF	f	Move WREG to f	1	0000 0001 ffff ffff	None	
MULWF	f	Multiply WREG with f	1	0011 0100 ffff ffff	None	
NEGW	f,s	Negate WREG	1	0010 110s ffff ffff	OV,C,DC,Z	1,3
NOP	—	No Operation	1	0000 0000 0000 0000	None	
RLCF	f,d	Rotate left f through Carry	1	0001 101d ffff ffff	C	
RLNCF	f,d	Rotate left f (no carry)	1	0010 001d ffff ffff	None	
RRCF	f,d	Rotate right f through Carry	1	0001 100d ffff ffff	C	
RRNCF	f,d	Rotate right f (no carry)	1	0010 000d ffff ffff	None	
SETF	f,s	Set f	1	0010 101s ffff ffff	None	3
SUBWF	f,d	Subtract WREG from f	1	0000 010d ffff ffff	OV,C,DC,Z	1
SUBWFB	f,d	Subtract WREG from f with Borrow	1	0000 001d ffff ffff	OV,C,DC,Z	1
SWAPF	f,d	Swap f	1	0001 110d ffff ffff	None	
TABLRD	t,i,f	Table Read	2 (3)	1010 10ti ffff ffff	None	7
TABLWT	t,i,f	Table Write	2	1010 11ti ffff ffff	None	5

Legend: Refer to Table 15-1 for opcode field descriptions, Shaded instructions are not available in the PIC17C42.

Note 1: 2's Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two cycle instruction.

6: Two cycle instruction when condition is true, else single cycle instruction.

7: Two cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.

8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.

© 1995 Microchip Technology Inc.

TABLE 15-2: PIC 17CXX INSTRUCTION SET (CONT.)

Mnemonic, Operands		Description	Cycles	16-bit Opcode				Status Affected	Notes
				MSb			LSb		
TLRD	**t,f**	Table Latch Read	1	1010	00tx	ffff	ffff	None	
TLWT	**t,f**	Table Latch Write	1	1010	01tx	ffff	ffff	None	
TSTFSZ	**f**	Test f, skip if 0	1 (2)	0011	0011	ffff	ffff	None	6,8
XORWF	**f,d**	Exclusive OR WREG with f	1	0000	110d	ffff	ffff	Z	
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f,b**	Bit Clear f	1	1000	1bbb	ffff	ffff	None	
BSF	**f,b**	Bit Set f	1	1000	0bbb	ffff	ffff	None	
BTFSC	**f,b**	Bit test, skip if clear	1 (2)	1001	1bbb	ffff	ffff	None	6,8
BTFSS	**f,b**	Bit test, skip if set	1 (2)	1001	0bbb	ffff	ffff	None	6,8
BTG	**f,b**	Bit Toggle f	1	0011	1bbb	ffff	ffff	None	
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	ADD literal to WREG	1	1011	0001	kkkk	kkkk	OV,C,DC,Z	
ANDLW	**k**	AND literal with WREG	1	1011	0101	kkkk	kkkk	Z	
CALL	**k**	Subroutine Call	2	111k	kkkk	kkkk	kkkk	None	7
CLRWDT	**—**	Clear Watchdog Timer	1	0000	0000	0000	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Unconditional Branch	2	110k	kkkk	kkkk	kkkk	None	7
IORLW	**k**	Inclusive OR literal with WREG	1	1011	0011	kkkk	kkkk	Z	
LCALL	**k**	Long Call	2	1011	0111	kkkk	kkkk	None	4,7
MOVLB	**k**	Move literal to low nibble in BSR	1	1011	1000	uuuu	kkkk	None	
MOVLR	**k**	Move literal to high nibble in BSR	1	1011	101x	kkkk	uuuu	None	
MOVLW	**k**	Move literal to WREG	1	1011	0000	kkkk	kkkk	None	
MULLW	**k**	Multiply literal with WREG	1	1011	1100	kkkk	kkkk	None	
RETFIE	**—**	Return from interrupt (and enable interrupts)	2	0000	0000	0000	0101	GLINTD	7
RETLW	**k**	Return literal to WREG	2	1011	0110	kkkk	kkkk	None	7
RETURN	**—**	Return from subroutine	2	0000	0000	0000	0010	None	7
SLEEP	**—**	Enter SLEEP Mode	1	0000	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	**k**	Subtract WREG from literal	1	1011	0010	kkkk	kkkk	OV,C,DC,Z	
XORLW	**k**	Exclusive OR literal with WREG	1	1011	0100	kkkk	kkkk	Z	

Legend: Refer to Table 15-1 for opcode field descriptions, Shaded instructions are not available in the PIC17C42.

Note 1: 2's Complement method.

2: Unsigned arithmetic.

3: If s = '1', only the file is affected: If s = '0', both the WREG register and the file are affected; If only the Working register (WREG) is required to be affected, then f = WREG must be specified.

4: During an LCALL, the contents of PCLATH are loaded into the MSB of the PC and kkkk kkkk is loaded into the LSB of the PC (PCL)

5: Multiple cycle instruction for EPROM programming when table pointer selects internal EPROM. The instruction is terminated by an interrupt event. When writing to external program memory, it is a two cycle instruction.

6: Two cycle instruction when condition is true, else single cycle instruction.

7: Two cycle instruction except for TABLRD to PCL (program counter low byte) in which case it takes 3 cycles.

8: A "skip" means that instruction fetched during execution of current instruction is not executed, instead an NOP is executed.

© 1995 Microchip Technology Inc.

ADDLW **ADD Literal to WREG**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (WREG) + k → (WREG)

Status Affected: OV, C, DC, Z

Encoding:

1011	0001	kkkk	kkkk

Description: The contents of WREG are added to the eight bit literal 'k' and the result is placed in WREG.

Words: 1

Cycles: 1

Example:
```
ADDLW   0x15
```
Before Instruction
WREG = 0x10

After Instruction
WREG = 0x25

ADDWF **ADD WREG to f**

Syntax: [*label*] ADDWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) + (f) → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0000	111d	ffff	ffff

Description: Add WREG to register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ADDWF   REG, 0
```
Before Instruction
WREG = 0x17
REG = 0xC2

After Instruction
WREG = 0xD9
REG = 0xC2

ADDWFC **ADD WREG and Carry bit to f**

Syntax: [*label*] ADDWFC f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) + (f) + C → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0001	000d	ffff	ffff

Description: Add WREG, the Carry Flag and data memory location 'f'. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed in data memory location 'f'.

Words: 1

Cycles: 1

Example:
```
ADDWFC   REG  0
```
Before Instruction
Carry bit = 1
REG = 0x02
WREG = 0x4D

After Instruction
Carry bit = 0
REG = 0x02
WREG = 0x50

ANDLW **And Literal with WREG**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (WREG) .AND. (k) → (WREG)

Status Affected: Z

Encoding:

1011	0101	kkkk	kkkk

Description: The contents of WREG are AND'ed with the eight bit literal 'k'. The result is placed in WREG.

Words: 1

Cycles: 1

Example:
```
ANDLW   0x5F
```
Before Instruction
WREG = 0xA3

After Instruction
WREG = 0x03

© 1995 Microchip Technology Inc.

ANDWF **AND WREG with f**

Syntax: [*label*] ANDWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) .AND. (f) → (dest)

Status Affected: Z

Encoding:

0000	101d	ffff	ffff

Description: The contents of WREG are AND'ed with register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
ANDWF   REG, 1
```
Before Instruction
WREG = 0x17
REG = 0xC2
After Instruction
WREG = 0x17
REG = 0x02

BSF **Bit Set f**

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: 1 → (f<b>)

Status Affected: None

Encoding:

1000	0bbb	ffff	ffff

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example:
```
BSF     FLAG_REG, 7
```
Before Instruction
FLAG_REG= 0x0A
After Instruction
FLAG_REG= 0x8A

BCF **Bit Clear f**

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

1000	1bbb	ffff	ffff

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example:
```
BCF     FLAG_REG,  7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BTFSC **Bit Test, skip if Clear**

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 255$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1001	1bbb	ffff	ffff

Description: If bit 'b' in register 'f' is 0 then the next instruction is skipped.
If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSC   FLAG,1
FALSE   :
TRUE    :
```
Before Instruction
PC = address (HERE)
After Instruction
if FLAG<1> = 0;
PC = address (TRUE)
if FLAG<1> = 1;
PC = address (FALSE)

© 1995 Microchip Technology Inc.

BTFSS **Bit Test, skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ b < 7

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

1001	0bbb	ffff	ffff

Description: If bit 'b' in register 'f' is 1 then the next instruction is skipped.
If bit 'b' is 1, then the next instruction fetched during the current instruction execution, is discarded and an NOP is executed instead, making this a 2 cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    BTFSS   FLAG,1
FALSE   :
TRUE    :
```
Before Instruction
PC = address (HERE)
After Instruction
if FLAG<1> = 0;
PC = address (FALSE)
if FLAG<1> = 1;
PC = address (TRUE)

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: 0 ≤ k ≤ 4095

Operation: PC+ 1→ TOS, k → PC<12:0>,
k<12:8> → PCLATH<4:0>;
PC<15:13> → PCLATH<7:5>

Status Affected: None

Encoding:

111k	kkkk	kkkk	kkkk

Description: Subroutine call within 8K page. First, return address (PC+1) is pushed onto the stack. The thirteen bit value is loaded into PC bits<12:0>. Then the upper-eight bits of the PC are copied into PCLATH. Call is a two-cycle instruction.
See LCALL for calls outside 8K memory space.

Words: 1

Cycles: 2

Example:
```
HERE    CALL    THERE
```
Before Instruction
PC = Address (HERE)
After Instruction
PC = Address (THERE)
TOS = Address (HERE + 1)

BTG **Bit Toggle f**

Syntax: [*label*] BTG f,b

Operands: 0 ≤ f ≤ 255
0 ≤ b < 7

Operation: (f<b>) → (f<b>)

Status Affected: None

Encoding:

0011	1bbb	ffff	ffff

Description: Bit 'b' in data memory location 'f' is inverted.

Words: 1

Cycles: 1

Example:
```
BTG     PORTC,  4
```
Before Instruction:
PORTC = 0111 0101 [0x75]
After Instruction:
PORTC = 0110 0101 [0x65]

CLRF **Clear f**

Syntax: [*label*] CLRF f,s

Operands: 0 ≤ f ≤ 255

Operation: 00h → f, s ∈ [0,1]
00h → dest

Status Affected: None

Encoding:

0010	100s	ffff	ffff

Description: Clears the contents of the specified register(s).
s = 0: Data memory location 'f' and WREG are cleared.
s = 1: Data memory location 'f' is cleared.

Words: 1

Cycles: 1

Example:
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00

© 1995 Microchip Technology Inc.

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT postscaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0100

Description: CLRWDT instruction resets the watchdog timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example: CLRWDT

Before Instruction
WDT counter = ?

After Instruction
WDT counter = 0x00
WDT Postscaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: $(\bar{f}) \rightarrow$ (dest)

Status Affected: Z

Encoding:

0001	001d	ffff	ffff

Description: The contents of register 'f' are complemented. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example: COMF REG1,0

Before Instruction
REG1 = 0x13

After Instruction
REG1 = 0x13
WREG = 0xEC

© 1995 Microchip Technology Inc.

CPFSEQ **Compare f with WREG, skip if f = WREG**

Syntax: [*label*] CPFSEQ f

Operands: $0 \le f \le 255$

Operation: (f) – (WREG), skip if (f) = (WREG) (unsigned comparison)

Status Affected: None

Encoding:

0011	0001	ffff	ffff

Description: Tests the contents of data memory location 'f' to the contents of WREG.

The subtraction is unsigned.

If 'f' = WREG then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Example:

```
HERE    CPFSEQ REG
NEQUAL  :
EQUAL   :
```

Before Instruction
PC Address = HERE
WREG = ?
REG = ?

After Instruction
if REG = WREG;
PC = Address (EQUAL)
if REG ≠ WREG;
PC = Address (NEQUAL)

CPFSGT **Compare f with WREG, skip if f > WREG**

Syntax: [*label*] CPFSGT f

Operands: $0 \le f \le 255$

Operation: (f) – (WREG), skip if (f) > (WREG) (unsigned comparison)

Status Affected: None

Encoding:

0011	0010	ffff	ffff

Description: Tests the contents of data memory location 'f' to the contents of the W register.

The subtraction is unsigned.

If the contents of 'f' > the contents of WREG then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Example:

```
HERE     CPFSGT, REG
NGREATER :
GREATER  :
```

Before Instruction
PC = Address (HERE)
WREG = ?

After Instruction
if REG > WREG;
PC = Address (GREATER)
if REG ≤ WREG;
PC = Address (NGREATER)

2

CPFSLT — **Compare f with WREG, skip if f < WREG**

Syntax: [*label*] CPFSLT f

Operands: $0 \le f \le 255$

Operation: (f) – (WREG), skip if (f) < (WREG) (unsigned comparison)

Status Affected: None

Encoding:

0011	0000	ffff	ffff

Description: Tests the contents of data memory location 'f' to the contents of WREG.

The subtraction is unsigned.

If the contents of 'f' < the contents of WREG, then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Example:

```
HERE    CPFSLT, REG
NLESS   :
LESS    :
```

Before Instruction
PC = Address (HERE)
W = ?

After Instruction
if REG < WREG;
PC = Address (LESS)
if REG ≥ WREG;
PC = Address (NLESS)

DAW — **Decimal Adjust WREG Register**

Syntax: [*label*] DAW f,s

Operands: $0 \le f \le 255$
$s \in [0,1]$

Operation:
if [WREG<3:0> >9] .OR. [DC = 1] then
WREG<3:0> + 6 → f<3:0>, s<3:0>;
else
WREG<3:0> → f<3:0>, s<3:0>;

if [WREG<7:4> >9] .OR. [C = 1] then
WREG<7:4> + 6 → f<7:4>, s<7:4>
else
WREG<7:4> → f<7:4>, s<7:4>

Status Affected: C

Encoding:

0010	111s	ffff	ffff

Description: DAW adjusts the eight bit value in WREG resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

s = 0: Result is placed in Data memory location 'f' and WREG.

s = 1: Result is placed in Data memory location 'f'.

Words: 1

Cycles: 1

Example1:

```
DAW   REG1, 0
```

Before Instruction
WREG = 0xA5
REG1 = ??
C = 0
DC = 0

After Instruction
WREG = 0x05
REG1 = 0x05
C = 1
DC = 0

Example 2:

Before Instruction
WREG = 0xCE
REG1 = ??
C = 0
DC = 0

After Instruction
WREG = 0x24
REG1 = 0x24
C = 1
DC = 0

© 1995 Microchip Technology Inc.

DECF **Decrement f**

Syntax: [*label*] DECF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – 1 → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0000	011d	ffff	ffff

Description: Decrement register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:
```
DECF    CNT,   1
```

Before Instruction
CNT = 0x01
Z = 0

After Instruction
CNT = 0x00
Z = 1

DECFSZ **Decrement f, skip if 0**

Syntax: [*label*] DECFSZ f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – 1 → (dest);
skip if result = 0

Status Affected: None

Encoding:

0001	011d	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE        DECFSZ    CNT, 1
            GOTO      LOOP
CONTINUE
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT - 1
if CNT = 0;
PC = Address (CONTINUE)
if CNT ≠ 0;
PC = Address (HERE+1)

© 1995 Microchip Technology Inc.

DECFSNZ **Decrement f, skip if not 0**

Syntax: [*label*] DECFSNZ f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – 1 → (dest);
skip if not 0

Status Affected: None

Encoding:

0010	011d	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

If the result is not 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:

```
HERE    DECFSNZ  TEMP, 1
ZERO    :
NZERO   :
```

Before Instruction
TEMP_VALUE = ?

After Instruction
TEMP_VALUE = TEMP_VALUE - 1,
if TEMP_VALUE = 0;
PC = Address (ZERO)
if TEMP_VALUE ≠ 0;
PC = Address (NZERO)

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 8191$

Operation: k → PC<12:0>;
k<12:8> → PCLATH<4:0>,
PC<15:13> → PCLATH<7:5>

Status Affected: None

Encoding:

110k	kkkk	kkkk	kkkk

Description: GOTO allows an unconditional branch anywhere within an 8K page boundary. The thirteen bit immediate value is loaded into PC bits <12:0>. Then the upper eight bits of PC are loaded into PCLATH. GOTO is always a two-cycle instruction.

Words: 1

Cycles: 2

Example:

```
GOTO THERE
```

After Instruction
PC = Address (THERE)

© 1995 Microchip Technology Inc.

INCF **Increment f**

Syntax: [*label*] INCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) + 1 → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0001	010d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
INCF    CNT, 1
```

Before Instruction
CNT = 0xFF
Z = 0
C = ?

After Instruction
CNT = 0x00
Z = 1
C = 1

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) + 1 → (dest)
skip if result = 0

Status Affected: None

Encoding:

0001	111d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

If the result is 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    INCFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT + 1
if CNT = 0;
PC = Address (ZERO)
if CNT ≠ 0;
PC = Address (NZERO)

© 1995 Microchip Technology Inc.

INCFSNZ **Increment f, skip if not 0**

Syntax: [*label*] INCFSNZ f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) + 1 → (dest), skip if not 0

Status Affected: None

Encoding:

0010	010d	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.
If the result is not 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE    INFSNZ  REG, 1
ZERO
NZERO
```
Before Instruction
REG = REG
After Instruction
REG = REG + 1
if REG = 1;
PC = Address (ZERO)
if REG = 0;
PC = Address (NZERO)

IORLW **Inclusive OR Literal with WREG**

Syntax: [*label*] IORLW k

Operands: $0 \le k \le 255$

Operation: (WREG) .OR. (k) → (WREG)

Status Affected: Z

Encoding:

1011	0011	kkkk	kkkk

Description: The contents of WREG are OR'ed with the eight bit literal 'k'. The result is placed in WREG.

Words: 1

Cycles: 1

Example:
```
IORLW   0x35
```
Before Instruction
WREG = 0x9A
After Instruction
WREG = 0xBF

© 1995 Microchip Technology Inc.

IORWF **Inclusive OR WREG with f**

Syntax: [*label*] IORWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) .OR. (f) → (dest)

Status Affected: Z

Encoding:

0000	100d	ffff	ffff

Description: Inclusive OR WREG with register 'f'. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Example:
```
IORWF  RESULT, 0
```
Before Instruction
RESULT = 0x13
WREG = 0x91

After Instruction
RESULT = 0x13
WREG = 0x93

LCALL **Long Call**

Syntax: [*label*] LCALL k

Operands: $0 \le k \le 255$

Operation: PC + 1 → TOS;
k → PCL, (PCLATH) → PCH

Status Affected: None

Encoding:

1011	0111	kkkk	kkkk

Description: LCALL allows unconditional subroutine call to anywhere within the 64k program memory space.

First, the return address (PC + 1) is pushed onto the stack. A 16-bit destination address is then loaded into the program counter. The lower 8-bits of the destination address is embedded in the instruction. The upper 8-bits of PC is loaded from PC high holding latch, PCLATH.

Words: 1

Cycles: 2

Example:
```
MOVLW  HIGH(SUBROUTINE)
MOVPF  WREG, PCLATH
LCALL  LOW(SUBROUTINE)
```
Before Instruction
SUBROUTINE = 16-bit Address
PC = ?

After Instruction
PC = Address (SUBROUTINE)

2

MOVFP **Move f to p**

Syntax: [*label*] MOVFP f,p

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (f) → (p)

Status Affected: None

Encoding:

011p	pppp	ffff	ffff

Description: Move data from data memory location 'f' to data memory location 'p'. Location 'f' can be anywhere in the 256 word data space (00h to FFh) while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful special situation).

MOVFP is particularly useful to transfer a data memory location to a peripheral register (such as the transmit buffer or an I/O port). Both 'f' and 'p' can be indirectly addressed.

Words: 1

Cycles: 1

Example:
```
MOVFP   REG1, REG2
```
Before Instruction
REG1 = 0x33,
REG2 = 0x11

After Instruction
REG1 = 0x33,
REG2 = 0x33

MOVLB **Move Literal to low nibble in BSR**

Syntax: [*label*] MOVLB k

Operands: $0 \le k \le 15$

Operation: k → (BSR<3:0>)

Status Affected: None

Encoding:

1011	1000	uuuu	kkkk

Description: The constant is loaded in the Bank Select Register (BSR). Only the low 4 bits of the Bank Select Register are affected. The upper half of the BSR is unchanged. The assembler will encode the "u" fields as '0'.

Words: 1

Cycles: 1

Example:
```
MOVLB   0x5
```
Before Instruction
BSR register = 0x22

After Instruction
BSR register = 0x25

Note: For the PIC17C42, only the low four bits of the BSR register are physically implemented. The upper nibble is read as '0'.

© 1995 Microchip Technology Inc.

MOVLR **Move Literal to high nibble in BSR**

Syntax: [*label*] MOVLR k

Operands: $0 \le k \le 15$

Operation: k → (BSR<7:4>)

Status Affected: None

Encoding:

1011	101x	kkkk	uuuu

Description: The constant is loaded into the most significant 4 bits of the Bank Select Register (BSR). Only the high 4 bits of the Bank Select Register are affected. The lower half of the BSR is unchanged. The assembler will encode the "u" fields as 0.

Words: 1

Cycles: 1

Example:
```
MOVLR   5
```
Before Instruction
BSR register = 0x22
After Instruction
BSR register = 0x52

Note: This instruction is not available in the PIC17C42 device.

MOVLW **Move Literal to WREG**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (WREG)

Status Affected: None

Encoding:

1011	0000	kkkk	kkkk

Description: The eight bit literal 'k' is loaded into WREG.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
WREG = 0x5A

MOVPF **Move p to f**

Syntax: [*label*] MOVPF p,f

Operands: $0 \le f \le 255$
$0 \le p \le 31$

Operation: (p) → (f)

Status Affected: Z

Encoding:

010p	pppp	ffff	ffff

Description: Move data from data memory location 'p' to data memory location 'f'. Location 'f' can be anywhere in the 256 byte data space (00h to FFh) while 'p' can be 00h to 1Fh.

Either 'p' or 'f' can be WREG (a useful special situation).

MOVPF is particularly useful for transferring a peripheral register (e.g., the timer or an I/O port) to a data memory location.

Words: 1

Cycles: 1

Example:
```
MOVPF   REG1, REG2
```
Before Instruction
REG1 = 0x11
REG2 = 0x33
After Instruction
REG1 = 0x11
REG2 = 0x11

MOVWF **Move WREG to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 255$

Operation: (WREG) → (f)

Status Affected: None

Encoding:

0000	0001	ffff	ffff

Description: Move data from WREG to register 'f'. Location 'f' can be anywhere in the 256 word data space.

Words: 1

Cycles: 1

Example:
```
MOVWF   REG
```
Before Instruction
WREG = 0x4F
REG = 0xFF
After Instruction
WREG = 0x4F
REG = 0x4F

© 1995 Microchip Technology Inc.

MULLW	Multiply Literal with WREG
Syntax:	[*label*] MULLW k
Operands:	0 ≤ k ≤ 255
Operation:	(k x WREG) → ProdH:ProdL
Status Affected:	None
Encoding:	1011 | 1100 | kkkk | kkkk
Description:	An unsigned multiplication is carried out between the contents of WREG and the 8-bit literal 'k'. The 16 bit result is placed in ProdH:ProdL register pair. ProdH contains the high byte. WREG is unchanged. None of the status flags are affected. Note that overflow or carry is not possible in this operation. Zero result is possible but not detected.
Words:	1
Cycles:	1
Example:	`MULLW 0xC4`

Before Instruction
- WREG = 0xE2
- PRODH = ?
- PRODL = ?

After Instruction
- WREG = 0xC4
- PRODH = 0xAD
- PRODL = 0x08

Note: This instruction is not available in the PIC17C42 device.

MULWF	Multiply WREG with f
Syntax:	[*label*] MULWF f
Operands:	0 ≤ f ≤ 255
Operation:	(WREG x f) → ProdH:ProdL
Status Affected:	None
Encoding:	0011 | 0100 | ffff | ffff
Description:	An unsigned multiplication is carried out between the contents of WREG and the register file location 'f'. The 16-bit result is stored in ProdH:ProdL register pair. ProdH contains the high byte. Both WREG and 'f' are unchanged. None of the status flags are affected. Note that overflow or carry is not possible in this operation. Zero result is possible but not detected.
Words:	1
Cycles:	1
Example:	`MULWF REG`

Before Instruction
- WREG = 0xC4
- REG = 0xB5
- PRODH = ?
- PRODL = ?

After Instruction
- WREG = 0xC4
- REG = 0xB5
- PRODH = 0x8A
- PRODL = 0x94

Note: This instruction is not available in the PIC17C42 device.

© 1995 Microchip Technology Inc.

NEGW **Negate W**

Syntax: [*label*] NEGW f,s

Operands: $0 \le F \le 255$
$s \in [0,1]$

Operation: $\overline{WREG} + 1 \rightarrow (f)$;
$\overline{WREG} + 1 \rightarrow s$

Status Affected: OV, C, DC, Z

Encoding:

0010	110s	ffff	ffff

Description: WREG is negated using two's complement. If 's' is 0 the result is placed in WREG and data memory location 'f'. If 's' is 1 the result is placed only in data memory location 'f'.

Words: 1

Cycles: 1

Example:
```
NEGW    REG,0
```
Before Instruction
WREG = 0011 1010 [0x3A],
REG = 1010 1011 [0xAB]

After Instruction
WREG = 1100 0111 [0xC6]
REG = 1100 0111 [0xC6]

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:

RETFIE **Return from Interrupt**

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → (PC);
0 → GLINTD;
PCLATH is unchanged.

Status Affected: GLINTD

Encoding:

0000	0000	0000	0101

Description: Return from Interrupt. Stack is POP'ed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by clearing the GLINTD bit. GLINTD is the global interrupt disable bit (CPUSTA<4>).

Words: 1

Cycles: 2

Example:
```
RETFIE
```
After Interrupt
PC = TOS
GLINTD = 0

RETLW **Return Literal to WREG**

Syntax: [*label*] RETLW k

Operands: $0 \le k \le 255$

Operation: k → (WREG); TOS → (PC);
PCLATH is unchanged

Status Affected: None

Encoding:

1011	0110	kkkk	kkkk

Description: WREG is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Example:
```
   CALL TABLE ; WREG contains table
              ;  offset value
              ; WREG now has
               ;  table value
   :
TABLE
   ADDWF PC   ; WREG = offset
   RETLW k0   ; Begin table
   RETLW k1   ;
   :
   :
   RETLW kn   ; End of table
```
Before Instruction
WREG = 0x07

After Instruction
WREG = value of k7

2

RETURN **Return from Subroutine**

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC;
PCLATH is unchanged

Status Affected: None

Encoding:

0000	0000	0000	0010

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter.

Words: 1

Cycles: 2

Example:
```
RETURN
```
After Interrupt
PC = TOS

RLCF **Rotate Left f through Carry**

Syntax: [*label*] RLCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

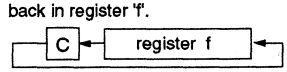
Operation: f<n> → d<n+1>;
f<7> → C;
C → d<0>

Status Affected: C

Encoding:

0001	101d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RLCF        REG,0
```
Before Instruction
REG = 1110 0110
C = 0

After Instruction
REG = 1110 0110
WREG = 1100 1100
C = 1

© 1995 Microchip Technology Inc.

RLNCF **Rotate Left f (no carry)**

Syntax: [*label*] RLNCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

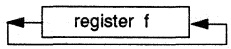
Operation: f<n> → d<n+1>;
f<7> → d<0>

Status Affected: None

Encoding:

0010	001d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RLNCF       REG, 1
```
Before Instruction
C = 0
REG = 1110 1011
After Instruction
C =
REG = 1101 0111

RRCF **Rotate Right f through Carry**

Syntax: [*label*] RRCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

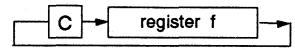
Operation: f<n> → d<n-1>;
f<0> → C;
C → d<7>

Status Affected: C

Encoding:

0001	100d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RRCF            REG1,0
```
Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
WREG = 0111 0011
C = 0

2

RRNCF **Rotate Right f (no carry)**

Syntax: [*label*] RRNCF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

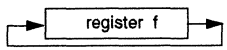
Operation: f<n> → d<n-1>;
f<0> → d<7>

Status Affected: None

Encoding:

0010	000d	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the right. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example 1:
```
RRNCF   REG, 1
```
Before Instruction
WREG = ?
REG = 1101 0111
After Instruction
WREG = 0
REG = 1110 1011

Example 2:
```
RRNCF   REG, 0
```
Before Instruction
WREG = ?
REG = 1101 0111
After Instruction
WREG = 1110 1011
REG = 1101 0111

SETF **Set f**

Syntax: [*label*] SETF f,s

Operands: $0 \le f \le 255$
$s \in [0,1]$

Operation: FFh → f;
FFh → d

Status Affected: None

Encoding:

0010	101s	ffff	ffff

Description: If 's' is 0, both the data memory location 'f' and WREG are set to FFh. If 's' is 1 only the data memory location 'f' is set to FFh.

Words: 1

Cycles: 1

Example1:
```
SETF    REG, 0
```
Before Instruction
REG = 0xDA
WREG = 0x05
After Instruction
REG = 0xFF
WREG = 0xFF

Example2:
```
SETF    PORTD, 1
```
Before Instruction
REG = 0xDA
WREG = 0x05
After Instruction
REG = 0xFF
WREG = 0x05

© 1995 Microchip Technology Inc.

SLEEP **Enter SLEEP mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT postscaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0011

Description: The power down status bit ($\overline{PD}$) is cleared. The time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its prescaler are cleared.

The processor is put into SLEEP mode with the oscillator stopped.

Words: 1

Cycles: 1

Example:

```
SLEEP
```

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?

After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared

SUBLW **Subtract WREG from Literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k – (WREG) → (WREG)

Status Affected: OV, C, DC, Z

Encoding:

1011	0010	kkkk	kkkk

Description: WREG is subtracted from the eight bit literal 'k'. The result is placed in WREG.

Words: 1

Cycles: 1

Example 1:

```
SUBLW   0x02
```

Before Instruction
WREG = 1
C = ?

After Instruction
WREG = 1
C = 1 ; result is positive
Z = 1

Example 2:

Before Instruction
WREG = 2
C = ?

After Instruction
WREG = 0
C = 1 ; result is zero
Z = 0

Example 3:

Before Instruction
WREG = 3
C = ?

After Instruction
WREG = FF ; (2's complement)
C = 0 ; result is negative
Z = 1

2

© 1995 Microchip Technology Inc.

SUBWF **Subtract WREG from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – (W) → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0000	010d	ffff	ffff

Description: Subtract WREG from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWF   REG1, 1
```
Before Instruction
```
REG1  =  3
WREG  =  2
C     =  ?
```
After Instruction
```
REG1  =  1
WREG  =  2
C     =  1    ; result is positive
Z     =  1
```
Example 2:

Before Instruction
```
REG1  =  2
WREG  =  2
C     =  ?
```
After Instruction
```
REG1  =  0
WREG  =  2
C     =  1    ; result is zero
Z     =  0
```
Example 3:

Before Instruction
```
REG1  =  1
WREG  =  2
C     =  ?
```
After Instruction
```
REG1  =  FF
WREG  =  2
C     =  0    ; result is negative
Z     =  1
```

SUBWFB **Subtract WREG from f with Borrow**

Syntax: [*label*] SUBWFB f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (f) – (W) – $\overline{C}$ → (dest)

Status Affected: OV, C, DC, Z

Encoding:

0000	001d	ffff	ffff

Description: Subtract WREG and the carry flag (borrow)from register 'f' (2's complement method). If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example 1:
```
SUBWFB  REG1, 1
```
Before Instruction
```
REG1  =  0x19   (0001 1001)
WREG  =  0x0D   (0000 1101)
C     =  1
```
After Instruction
```
REG1  =  0x0B   (0000 1011)
WREG  =  0x0D   (0000 1101)
C     =  1      ; result is positive
Z     =  1
```
Example2:
```
SUBWFB  REG1,0
```
Before Instruction
```
REG1  =  0x1B   (0001 1011)
WREG  =  0x1A   (0001 1010)
C     =  1
```
After Instruction
```
REG1  =  0x1B   (0001 1011)
WREG  =  0x00
C     =  1      ; result is zero
Z     =  0
```
Example3:
```
SUBWFB  REG1,1
```
Before Instruction
```
REG1  =  0x03   (0000 0011)
WREG  =  0x0E   (0000 1101)
C     =  1
```
After Instruction
```
REG1  =  0xF4   (1111 0100) [2's comp]
WREG  =  0x0E   (0000 1101)
C     =  0      ; result is negative
Z     =  1
```


© 1995 Microchip Technology Inc.

SWAPF **Swap f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: f<3:0> → dest<7:4>;
f<7:4> → dest<3:0>

Status Affected: None

Encoding:

0001	110d	ffff	ffff

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Example:
```
SWAPF   REG,   0
```
Before Instruction
REG = 0x53
After Instruction
REG = 0x35

TABLRD **Table Read**

Syntax: [*label*] TABLRD t,i,f

Operands: $0 \le f \le 255$
$i \in [0,1]$
$t \in [0,1]$

Operation: If t = 1,
TBLATH → f;
if t = 0,
TBLATL → f;
Prog Mem (TBLPTR) → TBLAT;
if i = 1,
TBLPTR + 1 → TBLPTR

Status Affected: None

Encoding:

1010	10ti	ffff	ffff

Description:
1. A byte of the table latch (TBLAT) is moved to register file 'f'.
if t = 0: the high byte is moved;
if t = 1: the low byte is moved
2. Then the contents of the program memory location pointed to by the 16-bit Table Pointer (TBLPTR) is loaded into the 16-bit Table Latch (TBLAT).
3. if i = 1: TBLPTR is incremented;
if i = 0: TBLPTR is not incremented

Words: 1

Cycles: 2 (3 cycle if f = PC)

Example1:
```
TABLRD  1, 1, REG ;
```
Before Instruction
REG = 0x53
TBLATH = 0xAA
TBLATL = 0x55
TBLPTR = 0xA356
MEMORY(TBLPTR) = 0x1234
After Instruction (table write completion)
REG = 0xAA
TBLATH = 0x12
TBLATL = 0x34
TBLPTR = 0xA357
MEMORY(TBLPTR) = 0x5678

Example2:
```
TABLRD  0, 0, REG ;
```
Before Instruction
REG = 0x53
TBLATH = 0xAA
TBLATL = 0x55
TBLPTR = 0xA356
MEMORY(TBLPTR) = 0x1234
After Instruction (table write completion)
REG = 0x55
TBLATH = 0x12
TBLATL = 0x34
TBLPTR = 0xA356
MEMORY(TBLPTR) = 0x1234

© 1995 Microchip Technology Inc.

TABLWT **Table Write**

Syntax: [*label*] TABLWT t,i,f

Operands: $0 \le f \le 255$
$i \in [0,1]$
$t \in [0,1]$

Operation: If t = 0,
f → TBLATL;
if t = 1,
f → TBLATH;
TBLAT → Prog Mem (TBLPTR);
if i = 1,
TBLPTR + 1 → TBLPTR

Status Affected: None

Encoding:

1010	11ti	ffff	ffff

Description:
1. Load value in 'f' into 16-bit table latch (TBLAT)
   if t = 0: load into low byte;
   if t = 1: load into high byte
2. The contents of TBLAT is written to the program memory location pointed to by TBLPTR
   If TBLPTR points to external program memory location, then the instruction takes two cycles.
   If TBLPTR points to an internal EPROM location, then the instruction is terminated when an interrupt is received.

Note: The $\overline{MCLR}/V_{PP}$ pin must be at the programming voltage for successful programming.
If $\overline{MCLR}/V_{PP} = V_{DD}$
the programming sequence will be executed, but will not be successful (although the memory location may be disturbed)

3. The TBLPTR can be automatically incremented
   if i = 0; TBLPTR is not incremented
   if i = 0; TBLPTR is incremented

Words: 1

Cycles: 2 (many if write is to on-chip EPROM program memory)

TABLWT (Cont.) **Table Write**

Example1: `TABLWT 0, 1, REG`

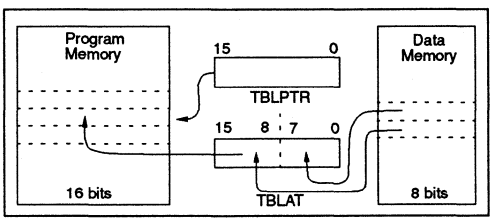
Before Instruction
REG	=	0x53
TBLATH	=	0xAA
TBLATL	=	0x55
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0xFFFF

After Instruction (table write completion)
REG	=	0x53
TBLATH	=	0x53
TBLATL	=	0x55
TBLPTR	=	0xA357
MEMORY(TBLPTR)	=	0x5355

Example 2: `TABLWT 1, 0, REG`

Before Instruction
REG	=	0x53
TBLATH	=	0xAA
TBLATL	=	0x55
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0xFFFF

After Instruction (table write completion)
REG	=	0x53
TBLATH	=	0xAA
TBLATL	=	0x53
TBLPTR	=	0xA356
MEMORY(TBLPTR)	=	0xAA53



© 1995 Microchip Technology Inc.

TLRD **Table Latch Read**

Syntax: [*label*] TLRD t,f

Operands: $0 \le f \le 255$
$t \in [0,1]$

Operation: If t = 0,
TBLATL → f;
if t = 1,
TBLATH → f

Status Affected: None

Encoding:

1010	00tx	ffff	ffff

Description: Read data from 16-bit table latch (TBLAT) into file register 'f'. Table Latch is unaffected.
If t = 1; high byte is read
if t = 0; low byte is read
This instruction is used in conjunction with TABLRD to transfer data from program memory to data memory.

Words: 1

Cycles: 1

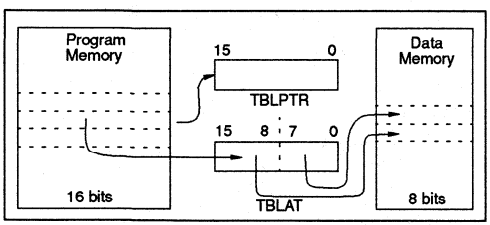
Example: `TLRD t, RAM`

Before Instruction
t = 0
RAM = ?
TBLAT = 0x00AF (TBLATH = 0x00) (TBLATL = 0xAF)

After Instruction
RAM = 0xAF
TBLAT = 0x00AF (TBLATH = 0x00) (TBLATL = 0xAF)

Before Instruction
t = 1
RAM = ?
TBLAT = 0x00AF (TBLATH = 0x00) (TBLATL = 0xAF)

After Instruction
RAM = 0x00
TBLAT = 0x00AF (TBLATH = 0x00) (TBLATL = 0xAF)



TLWT **Table Latch Write**

Syntax: [*label*] TLWT t,f

Operands: $0 \le f \le 255$
$t \in [0,1]$

Operation: If t = 0,
f → TBLATL;
if t = 1,
f → TBLATH

Status Affected: None

Encoding:

1010	01tx	ffff	ffff

Description: Data from file register 'f' is written into the 16-bit table latch (TBLAT).
If t = 1; high byte is written
if t = 0; low byte is written
This instruction is used in conjunction with TABLWT to transfer data from data memory to program memory.

Words: 1

Cycles: 1

Example: `TLWT t, RAM`

Before Instruction
t = 0
RAM = 0xB7
TBLAT = 0x0000 (TBLATH = 0x00) (TBLATL = 0x00)

After Instruction
RAM = 0xB7
TBLAT = 0x00B7 (TBLATH = 0x00) (TBLATL = 0xB7)

Before Instruction
t = 1
RAM = 0xB7
TBLAT = 0x0000 (TBLATH = 0x00) (TBLATL = 0x00)

After Instruction
RAM = 0xB7
TBLAT = 0xB700 (TBLATH = 0xB7) (TBLATL = 0x00)

© 1995 Microchip Technology Inc.

TSTFSZ **Test f, skip if 0**

Syntax: [*label*] TSTFSZ f

Operands: $0 \le f \le 255$

Operation: skip if f = 0

Status Affected: None

Encoding:

0011	0011	ffff	ffff

Description: If 'f' = 0, the next instruction, fetched during the current instruction execution, is discarded and an NOP is executed making this a two-cycle instruction.

Words: 1

Cycles: 1 (2)

Example:

```
HERE    TSTFSZ  CNT
NZERO     :
ZERO      :
```

Before Instruction
PC = Address(HERE)

After Instruction
if CNT = 0x00,
PC = Address (ZERO)
if CNT ≠ 0x00,
PC = Address (NZERO)

XORLW **Exclusive OR Literal with WREG**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (WREG) .XOR. k → (WREG)

Status Affected: Z

Encoding:

1011	0100	kkkk	kkkk

Description: The contents of WREG are XOR'ed with the eight bit literal 'k'. The result is placed in WREG.

Words: 1

Cycles: 1

Example:

```
XORLW  0xAF
```

Before Instruction
WREG = 0xB5

After Instruction
WREG = 0x1A

XORWF **Exclusive OR WREG with f**

Syntax: [*label*] XORWF f,d

Operands: $0 \le f \le 255$
$d \in [0,1]$

Operation: (WREG) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

0000	110d	ffff	ffff

Description: Exclusive OR the contents of WREG with register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in the register 'f'.

Words: 1

Cycles: 1

Example:

```
XORWF   REG, 1
```

Before Instruction
REG = 0xAF
WREG = 0xB5

After Instruction
REG = 0x1A
WREG = 0xB5

© 1995 Microchip Technology Inc.

16.0 DEVELOPMENT SUPPORT

16.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER® Real-Time In-Circuit Emulator
- PRO MATE™ Universal Programmer
- PICSTART® Low-Cost Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- MPASM Assembler
- MPSIM Software Simulator
- C Compiler (MP-C)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

16.2 PICMASTER: High Performance Universal In-Circuit Emulator

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16C5X, PIC16CXX and PIC17CXX families. A PICMASTER System configuration is shown in Figure 16-1.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new PIC16C5X, PIC16CXX and PIC17CXX microcontrollers.

The Emulator System is designed to operate on PC compatible 386 (and better) machines in the Microsoft Windows™ 3.x environment. Thus, allowing the operator access to a wide range of supporting software and accessories.

The PICMASTER has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The AT platform and Windows 3.x environment was chosen to best make these features available to you, the end user.

The PICMASTER Universal Emulator System consists primarily of four major components:

- Host-Interface Card
- Emulator Control Pod
- Target-Specific Emulator Probe
- PC-Host Emulation Control Software

The Windows 3.x operating system allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

PICMASTER emulation can operate in one window, while a text editor is running in a second window.

PC-Host Emulation Control software takes full advantage of Dynamic Data Exchange (DDE), a feature of Windows 3.x. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

Under Windows 3.x, two or more PICMASTER emulators can be run simultaneously from the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

The PICMASTER probes specifications are shown in Table 16-1.

FIGURE 16-1: PICMASTER SYSTEM CONFIGURATION

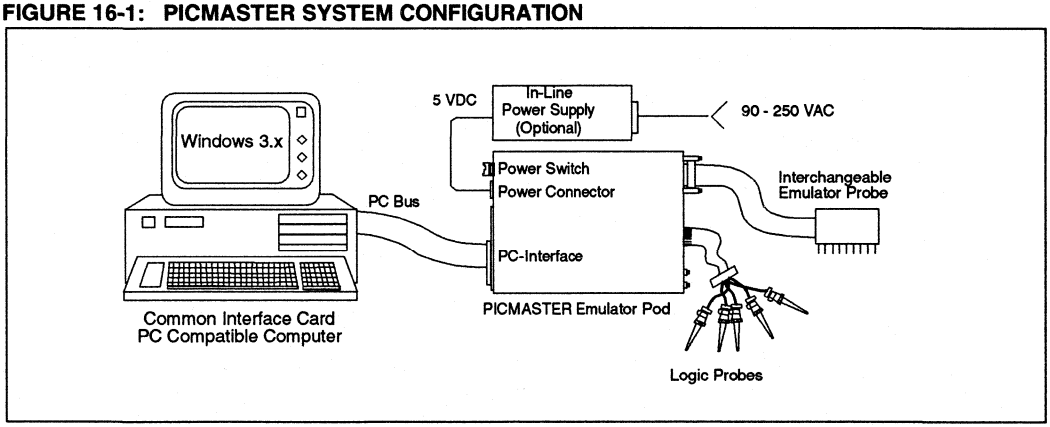


TABLE 16-1: PICMASTER PROBE SPECIFICATION

PICMASTER Probe	Devices Supported	PROBE	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10 MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10 MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C54A, PIC16CR54, PIC16C55, PIC16C56, PIC16C57, PIC16CR57A, PIC16C58A, and PIC16CR58A	20 MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10 MHz	4.5V - 5.5V
PROBE-16F	PIC16C65*, PIC16C73 and PIC16C74	10 MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10 MHz	4.5V - 5.5V
PROBE-16H	PIC16C620, PIC16C621 and PIC16C622	10 MHz	4.5V - 5.5V
PROBE-17A	PIC17C42	16 MHz	4.5V - 5.5V

* PROBE-16F indirectly supports the PIC16C65.

© 1995 Microchip Technology Inc.

16.3 PRO MATE: Universal Programmer

The PRO MATE Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE has programmable VDD and VPP supplies which allows it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE can read, verify or program PIC16C5X, PIC16CXX and PIC17CXX devices. It can also set fuse configuration and code-protect bits in this mode.

In PC-hosted mode, the PRO MATE connects to the PC via one of the COM (RS-232) ports. PC based user-interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. Full screen display and editing of data, easy selection of fuse configuration and part type, easy selection of VDD min, VDD max and VPP levels, load and store to and from disk files (Intel® hex format) are some of the features of the software. Essential commands such as read, verify, program and blank check can be issued from the screen. Additionally, serial programming support is possible where each part is programmed with a different serial number, sequential or random.

The PRO MATE has a modular "programming socket module". Different socket modules are required for different processor types and/or package types.

PRO MATE supports all PIC16C5X, PIC16CXX and PIC17CXX processors.

16.4 PICSTART Low-Cost Development System

The PICSTART programmer is an easy to use, very low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. A PC-based user interface software makes using the programmer simple and efficient. The user interface is full-screen and menu-based. PICSTART is not recommended for production programming.

16.5 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C84, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

16.6 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

© 1995 Microchip Technology Inc.

16.7 Assembler (MPASM)

The MPASM Cross Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC16C5X, PIC16CXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER).

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a full feature directive language represented by four basic classes of directives:

- **Data Directives** are those that control the allocation of memory and provide a way to refer to data items symbolically, i.e., by meaningful names.
- **Listing Directives** control the MPASM listing display. They allow the specification of titles and subtitles, page ejects and other listing control.
- **Control Directives** permit sections of conditionally assembled code.
- **Macro Directives** control the execution and data allocation within macro body definitions.

16.8 Software Simulator (MPSIM)

The MPSIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode. MPSIM fully supports symbolic debugging using MP-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

16.9 C Compiler (MP-C)

The MP-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

The MP-C Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada. If you have any questions, please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

16.10 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP Edition, for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

16.11 Development Systems

For convenience, the development tools are packaged into comprehensive systems as listed in Table 16-2.

TABLE 16-2: DEVELOPMENT SYSTEM PACKAGES

Item	Name	System Description
1.	PICMASTER System	PICMASTER In-Circuit Emulator, PRO MATE Programmer, Assembler, Software Simulator, Samples and your choice of Target Probe.
2.	PICSTART System	PICSTART Low-Cost Prototype Programmer, Assembler, Software Simulator and Samples.
3.	PRO MATE System	PRO MATE Universal Programmer, full featured stand-alone or PC-hosted programmer, Assembler, Simulator

© 1995 Microchip Technology Inc.

17.0 PIC17C42 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to +125°C

Storage temperature ..... -65°C to +150°C

Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V

Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) ..... -0.6V to +14V

Voltage on RA2 and RA3 with respect to V_{SS} ..... -0.6V to +12V

Voltage on all other pins with respect to V_{SS} ..... -0.6V to V_{DD} + 0.6V

Total power dissipation (Note 1) 1.0W

Maximum current out of V_{SS} pin(s) - Total ..... 250 mA

Maximum current into V_{DD} pin(s) - Total ..... 200 mA

Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA

Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA

Maximum output current sunk by any I/O pin (except RA2 and RA3) ..... 35 mA

Maximum output current sunk by RA2 or RA3 pins ..... 60 mA

Maximum output current sourced by any I/O pin ..... 20 mA

Maximum current sunk by PORTA and PORTB (combined) ..... 150 mA

Maximum current sourced by PORTA and PORTB (combined) ..... 100 mA

Maximum current sunk by PORTC, PORTD and PORTE (combined) ..... 150 mA

Maximum current sourced by PORTC, PORTD and PORTE (combined) ..... 100 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 17-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	17C42-16	17C42-25
RC	VDD: 4.5V to 5.5V IDD: 6 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 4 MHz max.	VDD: 4.5V to 5.5V IDD: 6 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 4 MHz max.
XT	VDD: 4.5V to 5.5V IDD: 24 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 16 MHz max.	VDD: 4.5V to 5.5V IDD: 38 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 25 MHz max.
EC	VDD: 4.5V to 5.5V IDD: 24 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 16 MHz	VDD: 4.5V to 5.5V IDD: 38 mA max. IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 25 MHz max.
LF	VDD: 4.5V to 5.5V IDD: 150 µA max. at 32 kHz (WDT enabled) IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 2 MHz max.	VDD: 4.5V to 5.5V IDD: 150 µA max. at 32 kHz (WDT enabled) IPD: 5 µA max. at 5.5V (WDT disabled) Freq: 2 MHz max

© 1995 Microchip Technology Inc.

17.1 DC CHARACTERISTICS: PIC17C42-16 (Commercial, Industrial) PIC17C42-25 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.5V to 5.5V				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	4.5	–	5.5	V	
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to guarantee Power-On Reset	–	VSS	–	V	See section on Power-On Reset for details
D004	SVDD	VDD rise rate to guarantee Power-On Reset	0.060*	–	–	mV/ms	See section on Power-On Reset for details
D010	IDD	Supply Current (Note 2)	–	3	6	mA	FOSC = 4 MHz (Note 4)
D011			–	6	12 *	mA	FOSC = 8 MHz
D012			–	11	24 *	mA	FOSC = 16 MHz
D013			–	19	38	mA	FOSC = 25 MHz
D014			–	95	150	µA	FOSC = 32 kHz WDT enabled (EC osc configuration)
D020	IPD	Power Down Current (Note 3)	–	10	40	µA	VDD = 5.5V, WDT enabled
D021			–	< 1	5	µA	VDD = 5.5V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD or VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
Current consumed from the oscillator and I/O's driving external capacitive or resistive loads need to be considered.
For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD / (2 • R).
For capacitive loads, The current can be estimated (for an individual I/O pin) as (CL • VDD) • f
CL = Total capacitive load on the I/O pin; f = average frequency on the I/O pin switches.
The capacitive currents are most significant when the device is configured for external execution (includes extended microcontroller mode).

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

17.2 DC CHARACTERISTICS: PIC17C42-16 (Commercial, Industrial)
PIC17C42-25 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in Section 17.1				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Input Low Voltage					
	VIL	I/O ports					
D030		with TTL buffer	VSS	–	0.8	V	
D031		with Schmitt Trigger buffer	VSS	–	0.2 VDD	V	
D032		$\overline{MCLR}$, OSC1 (in EC and RC mode)	VSS	–	0.2 VDD	V	Note1
D033		OSC1 (in XT, and LF mode)	–	0.5 VDD	–	V	
		Input High Voltage					
	VIH	I/O ports		–			
D040		with TTL buffer	2.0	–	VDD	V	
D041		with Schmitt Trigger buffer	0.8 VDD	–	VDD	V	
D042		$\overline{MCLR}$	0.8 VDD	–	VDD	V	Note1
D043		OSC1 (XT, and LF mode)	–	0.5 VDD	–	V	
D050	VHYS	Hysteresis of Schmitt Trigger inputs	0.15 VDD*	–	–	V	
		Input Leakage Current (Notes 2, 3)					
D060	IIL	I/O ports (except RA2, RA3)	–	–	±1	µA	VSS ≤ VPIN ≤ VDD, I/O Pin at hi-impedance PORTB weak pull-ups disabled
D061		$\overline{MCLR}$	–	–	±2	µA	VPIN = VSS or VPIN = VDD
D062		RA2, RA3			±2	µA	VSS ≤ VRA2, VRA3 ≤ 12V
D063		OSC1, TEST	–	–	±1	µA	VSS ≤ VPIN ≤ VDD
D064		$\overline{MCLR}$	–	–	10	µA	VMCLR = VPP = 12V (when not programming)
D070	IPURB	PORTB weak pull-up current	60	200	400	µA	VPIN = VSS, $\overline{RBPU}$ = 0

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The $\overline{MCLR}$/Vpp pin may be kept in this range at times other than programming, but this is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

© 1995 Microchip Technology Inc.

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD range as described in Section 17.1				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Output Low Voltage					
D080	VOL	I/O ports (except RA2 and RA3)	–	–	0.1 VDD	V	IOL = 4 mA
D081		with TTL buffer	–	–	0.4	V	IOL = 6 mA, VDD = 4.5V Note 6
D082		RA2 and RA3	–	–	3.0	V	IOL = 60.0 mA, VDD = 5.5V
D083		OSC2/CLKOUT (RC and EC osc modes)	–	–	0.4	V	IOL = 2 mA, VDD = 4.5V
		Output High Voltage (Note 3)					
D090	VOH	I/O ports (except RA2 and RA3)	0.9 VDD	–	–	V	IOH = -2 mA
D091		with TTL buffer	2.4	–	–	V	IOH = -6.0 mA, VDD = 4.5V Note 6
D092		RA2 and RA3	–	–	12	V	Pulled-up to externally applied voltage
D093		OSC2/CLKOUT (RC and EC osc modes)	2.4	–	–	V	IOH = -5 mA, VDD = 4.5V
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin	–	–	25 ††	pF	In EC or RC osc modes when OSC2 pin is outputting CLKOUT. external clock is used to drive OSC1.
D101	CIO	All I/O pins and OSC2 (in RC mode)	–	–	50 ††	pF	
D102	CAD	System Interface Bus (PORTC, PORTD and PORTE)	–	–	100 ††	pF	In Microprocessor or Extended Microcontroller mode
		Internal Program Memory Programming Specs (Note 4)					
D110	VPP	Voltage on MCLR/VPP pin	12.5	–	13.5	V	Note 5
D111	VDDP	Supply voltage during programming	4.75	5.0	5.25	V	
D112	IPP	Current into MCLR/VPP pin	–	25 ‡	50 ‡	mA	
D113	IDDP	Supply current during programming	–	–	30 ‡	mA	
D114	TPROG	Programming pulse width	10	100	1000	µs	Terminated via internal/external interrupt or a reset

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The MCLR/Vpp pin may be kept in this range at times other than programming, but this is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

© 1995 Microchip Technology Inc.

17.3 Timing Parameter Symbology

The timing parameter symbols have been created using one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

Lowercase symbols (pp) and their meanings:

pp			
ad	Address/Data	ost	Oscillator Start-Up Timer
al	ALE	pwrt	Power-Up Timer
cc	Capture1 and Capture2	rb	PORTB
ck	CLKOUT or clock	rd	$\overline{RD}$
dt	Data in	rw	$\overline{RD}$ or $\overline{WR}$
in	INT pin	t0	T0CKI
io	I/O port	t123	TCLK12 and TCLK3
mc	$\overline{MCLR}$	wdt	Watchdog Timer
oe	$\overline{OE}$	wr	$\overline{WR}$
os	OSC1		

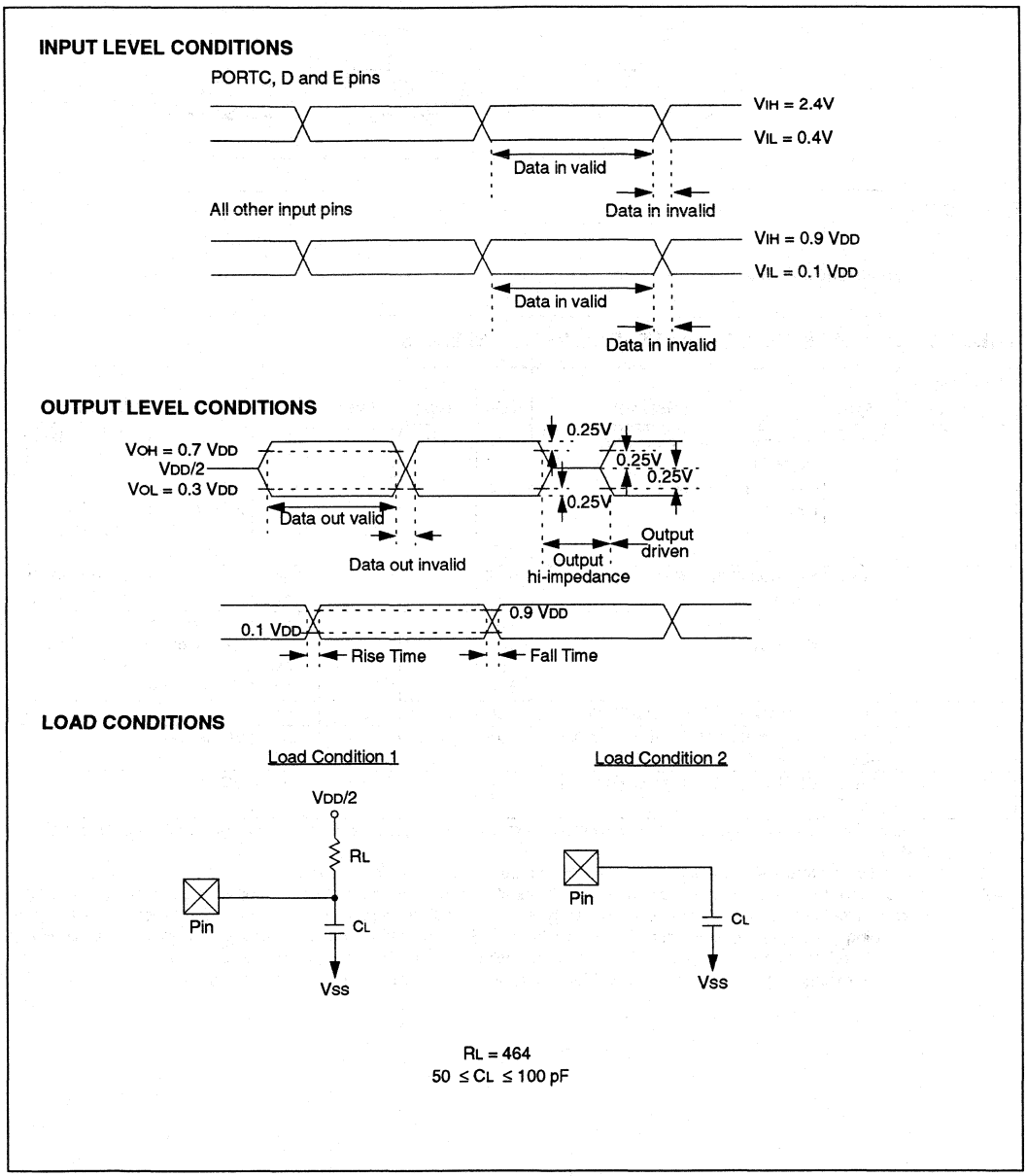
Uppercase symbols and their meanings:

S			
D	Driven	L	Low
E	Edge	P	Period
F	Fall	R	Rise
H	High	V	Valid
I	Invalid (Hi-impedance)	Z	High Impedance

© 1995 Microchip Technology Inc.

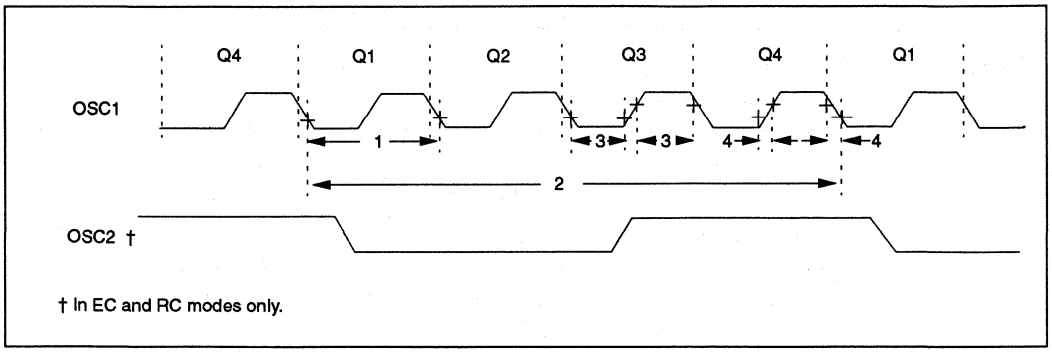
FIGURE 17-1: PARAMETER MEASUREMENT INFORMATION

All timings are measure between high and low measurement points as indicated in the figures below.



17.4 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING



† In EC and RC modes only.

TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency**	DC	—	16	MHz	EC osc mode - PIC17C42-16
		(Note 1)	DC	—	25	MHz	- PIC17C42-25
		Oscillator Frequency	DC	—	4	MHz	RC osc mode
		(Note 1)	1	—	16	MHz	XT osc mode - PIC17C42-16
			1	—	25	MHz	- PIC17C42-25
			DC	—	2	MHz	LF osc mode
1	Tosc	**External CLKIN Period**	62.5	—	—	ns	EC osc mode - PIC17C42-16
		(Note 1)	40	—	—	ns	- PIC17C42-25
		Oscillator Period	250	—	—	ns	RC osc mode
		(Note 1)	62.5	—	1,000	ns	XT osc mode - PIC17C42-16
			40	—	1,000	ns	- PIC17C42-25
			500	—	—	ns	LF osc mode
2	Tcy	**Instruction Cycle Time (Note 1)**	160	4/Fosc	DC	ns	
3	TosL, TosH	Clock in (OSC1) High or Low Time	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	5 ‡	ns	EC oscillator

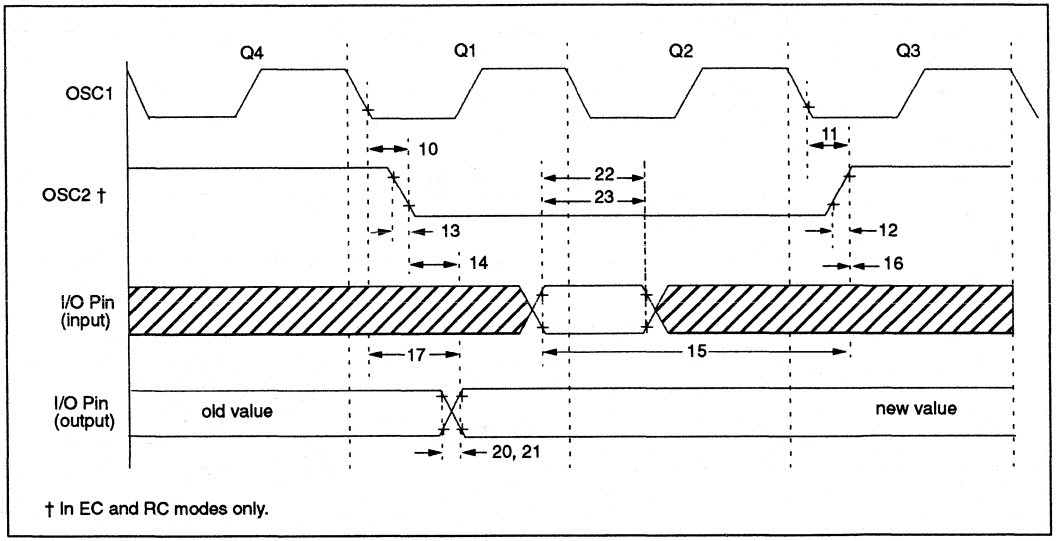
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 1995 Microchip Technology Inc.

FIGURE 17-3: CLKOUT AND I/O TIMING



† In EC and RC modes only.

TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	—	15 ‡	30 ‡	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	—	15 ‡	30 ‡	ns	Note 1
12	TckR	CLKOUT rise time	—	5 ‡	15 ‡	ns	Note 1
13	TckF	CLKOUT fall time	—	5 ‡	15 ‡	ns	Note 1
14	TckH2ioV	CLKOUT↑ to Port out valid	—	—	0.5 TCY+20‡	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑	0.25 TCY+25 ‡	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0 ‡	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	—	100 ‡	ns	
20	TioR	Port output rise time	—	10 ‡	35 ‡	ns	
21	TioF	Port output fall time	—	10 ‡	35 ‡	ns	
22	TinHL	INT pin high or low time	25 *	—	—	ns	
23	TrbHL	RB<7:0> change INT high or low time	25 *	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Measurements are taken in EC Mode where OSC2 output is 4 x Tosc.

FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

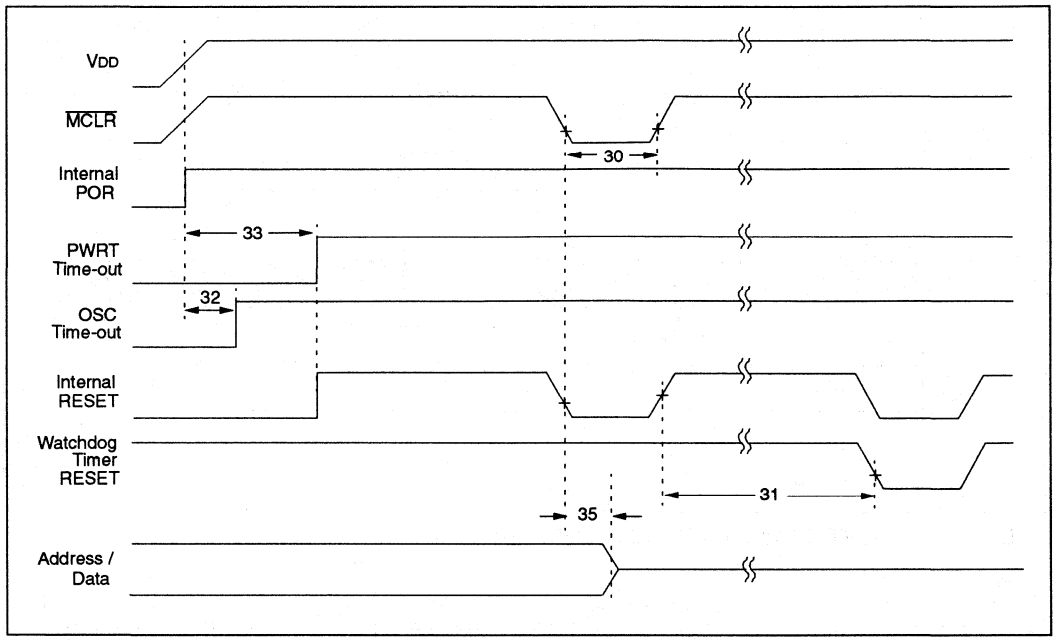


TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100 *	—	—	ns	
31	Twdt	Watchdog Timer Timeout Period (Prescale = 1)	5 *	12	25 *	ms	
32	Tost	Oscillation Start-Up Timer Period		1024 Tosc §		ms	Tosc = OSC1 period
33	Tpwrt	Power-Up Timer Period	40 *	96	200 *	ms	
35	TmcL2adl	$\overline{MCLR}$ to System Interface bus (AD<15:0>) invalid	—	—	100 *	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

§ This specification guaranteed by design.

© 1995 Microchip Technology Inc.

FIGURE 17-5: TIMER0 CLOCK TIMINGS

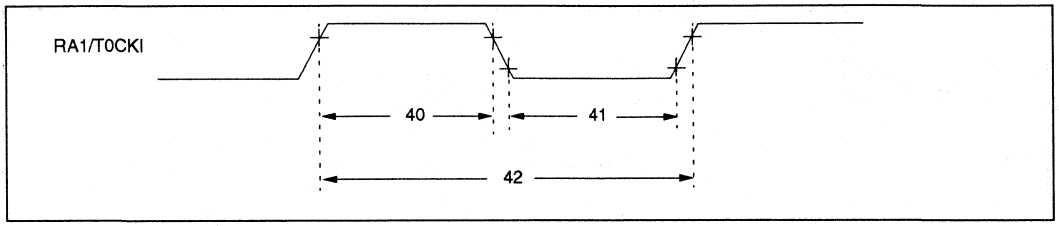


TABLE 17-5: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		TCY + 40 § / N	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS

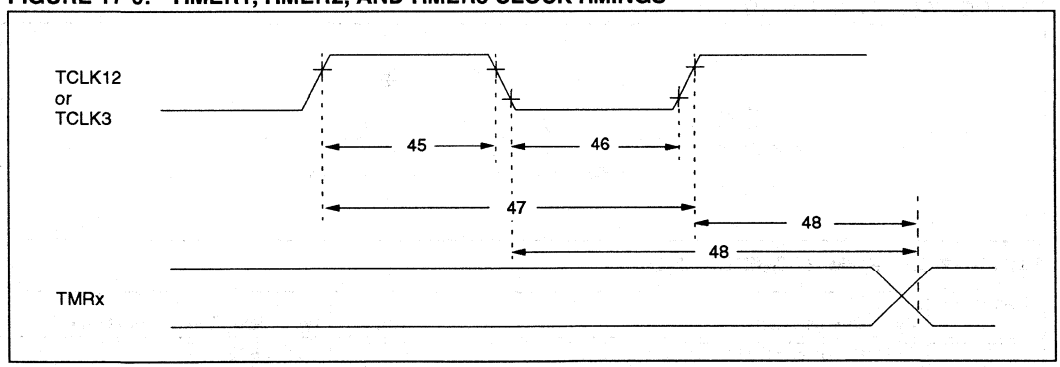


TABLE 17-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ †	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5 TCY + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5 TCY + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	TCY + 40 § / N	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrI	Delay from selected External Clock Edge to Timer increment	2 Tosc §		6 Tosc §		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 17-7: CAPTURE TIMINGS

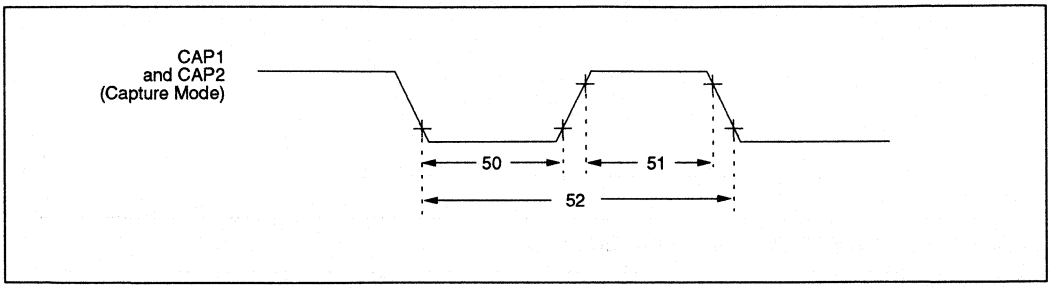


TABLE 17-7: CAPTURE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
50	TccL	Capture1 and Capture2 input low time	10 *	—	—	ns	
51	TccH	Capture1 and Capture2 input high time	10 *	—	—	ns	
52	TccP	Capture1 and Capture2 input period	2 Tcy § / N	—	—	ns	N = prescale value (4 or 16)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 17-8: PWM TIMINGS

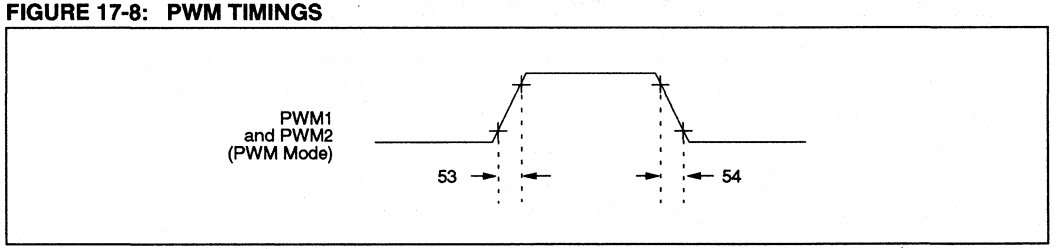


TABLE 17-8: PWM REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
53	TccR	PWM1 and PWM2 output rise time	—	10 *	35 *§	ns	
54	TccF	PWM1 and PWM2 output fall time	—	10 *	35 *§	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

© 1995 Microchip Technology Inc.

FIGURE 17-9: SCI MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

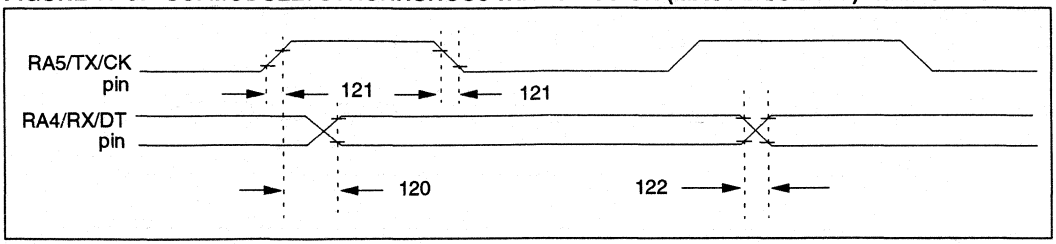


TABLE 17-9: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE) Clock high to data out valid	—	—	65	ns	
121	TckRF	Clock out rise time and fall time (Master Mode)	—	10	35	ns	
122	TdtRF	Data out rise time and fall time	—	10	35	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-10: SCI MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

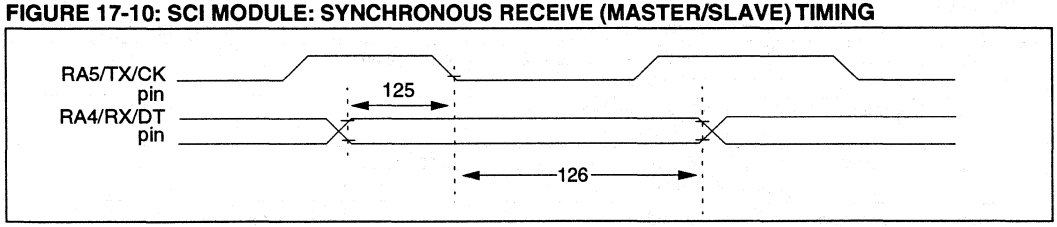


TABLE 17-10: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data hold before CK↓ (DT hold time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-11: MEMORY INTERFACE WRITE TIMING

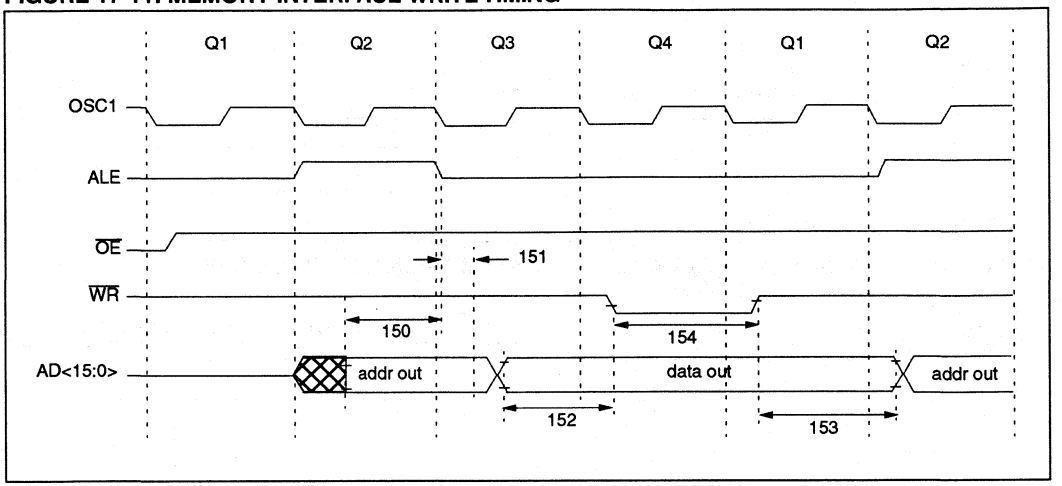


TABLE 17-11: MEMORY INTERFACE WRITE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25 Tcy-30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	—	—	ns	
152	TadV2wrL	Data out valid to $\overline{WR}$↓ (data setup time)	0.25 Tcy-40	—	—	ns	
153	TwrH2adI	$\overline{WR}$↑ to data out invalid (data hold time)	—	0.25 Tcy §	—	ns	
154	TwrL	$\overline{WR}$ pulse width	—	0.25 Tcy §	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification is guaranteed by design.

© 1995 Microchip Technology Inc.

FIGURE 17-12: MEMORY INTERFACE READ TIMING

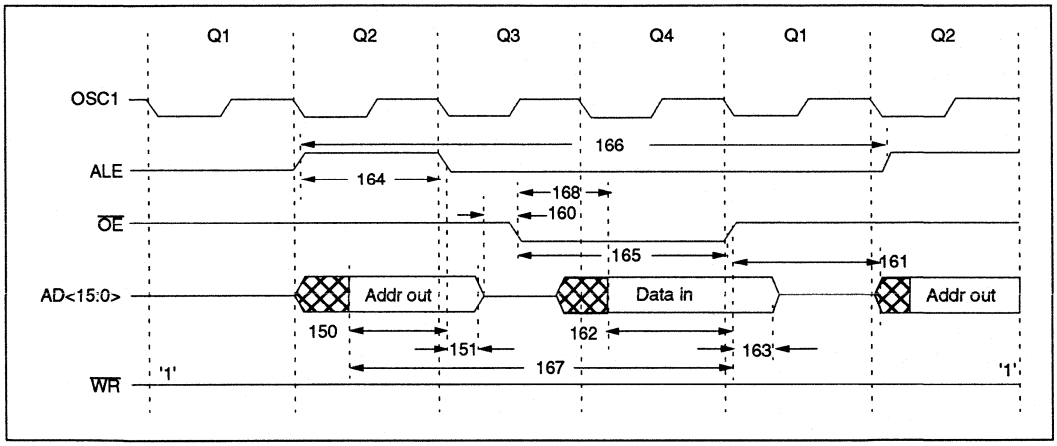


TABLE 17-12: MEMORY INTERFACE READ REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25 Tcy-30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	—	—	ns	
160	TadZ2oeL	AD<15:0> high impedance to $\overline{OE}$↓	10	—	—	ns	
161	ToeH2adD	$\overline{OE}$↑ to AD<15:0> driven	0.25 Tcy-15	—	—	ns	
162	TadV2oeH	Data in valid before $\overline{OE}$↑ (data setup time)	35	—	—	ns	
163	ToeH2adI	$\overline{OE}$↑to data in invalid (data hold time)	0	—	—	ns	
164	TalH	ALE pulse width	—	0.25 Tcy §	—	ns	
165	ToeL	$\overline{OE}$ pulse width	0.5 Tcy-35 §	—	—	ns	
166	TalH2alH	ALE↑ to ALE↑ (cycle time)	—	Tcy §	—	ns	
167	Tacc	Address access time	—	—	0.75 Tcy-40	ns	
168	Toe	Output enable access time ($\overline{OE}$ low to Data Valid)	—	—	0.5 Tcy - 60	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

NOTES:

© 1995 Microchip Technology Inc.

18.0 PIC17C42 DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

TABLE 18-1: PIN CAPACITANCE PER PACKAGE TYPE

Pin Name	Typical Capacitance (pF)			
	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP
All pins, except $\overline{MCLR}$, VDD, and Vss	10	10	10	10
$\overline{MCLR}$ pin	20	20	20	20

FIGURE 18-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

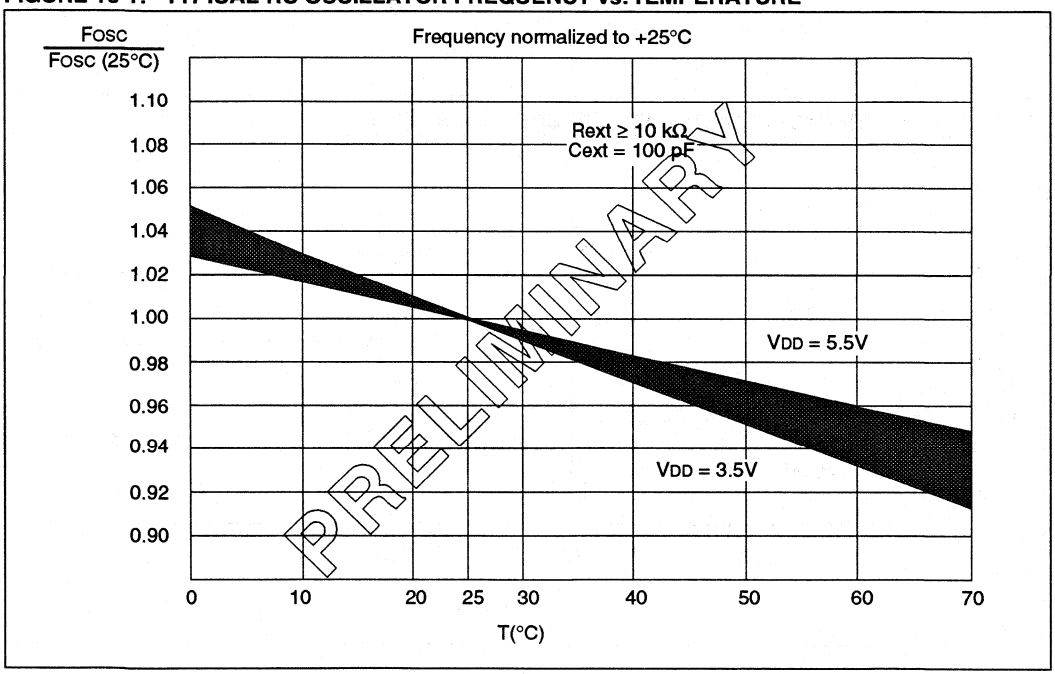


FIGURE 18-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

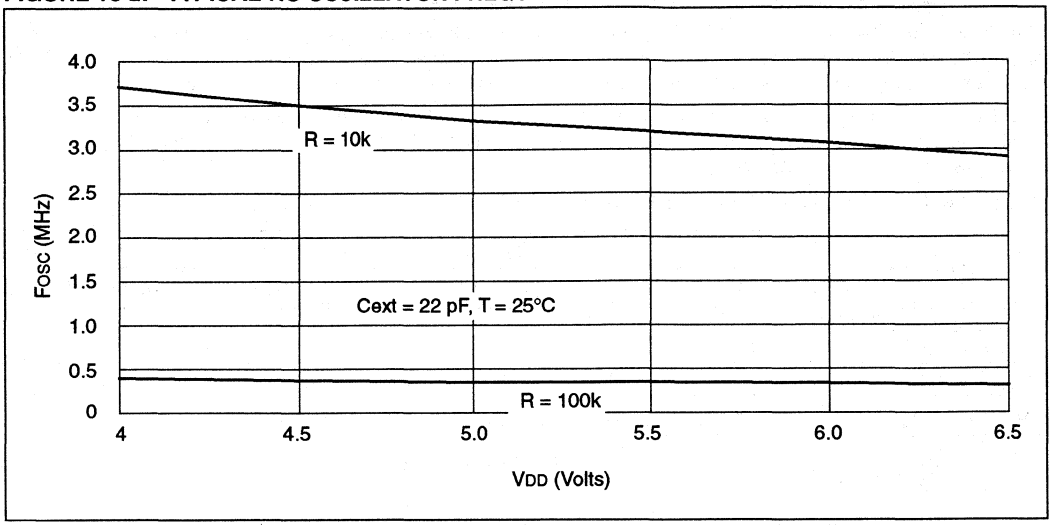
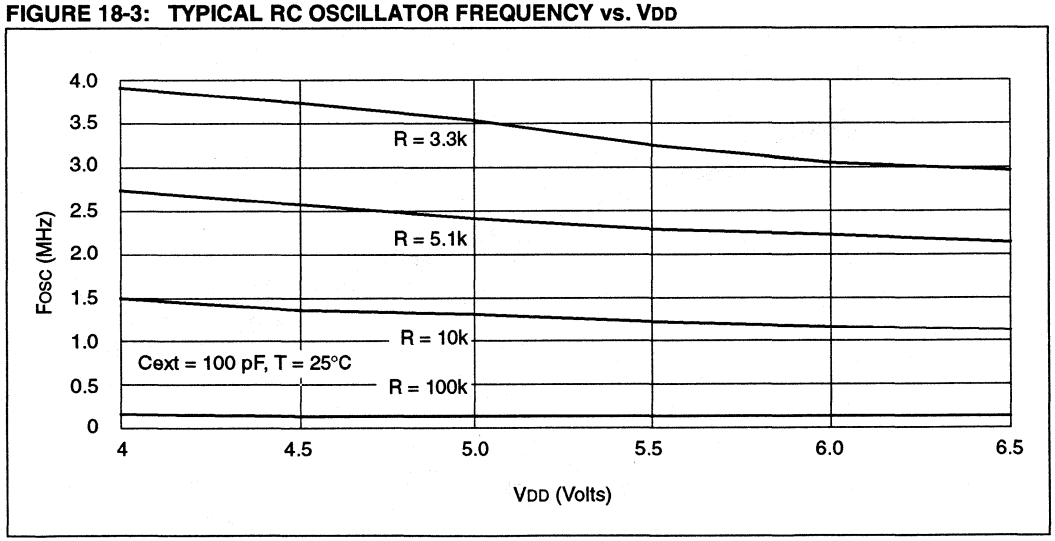


FIGURE 18-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 18-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

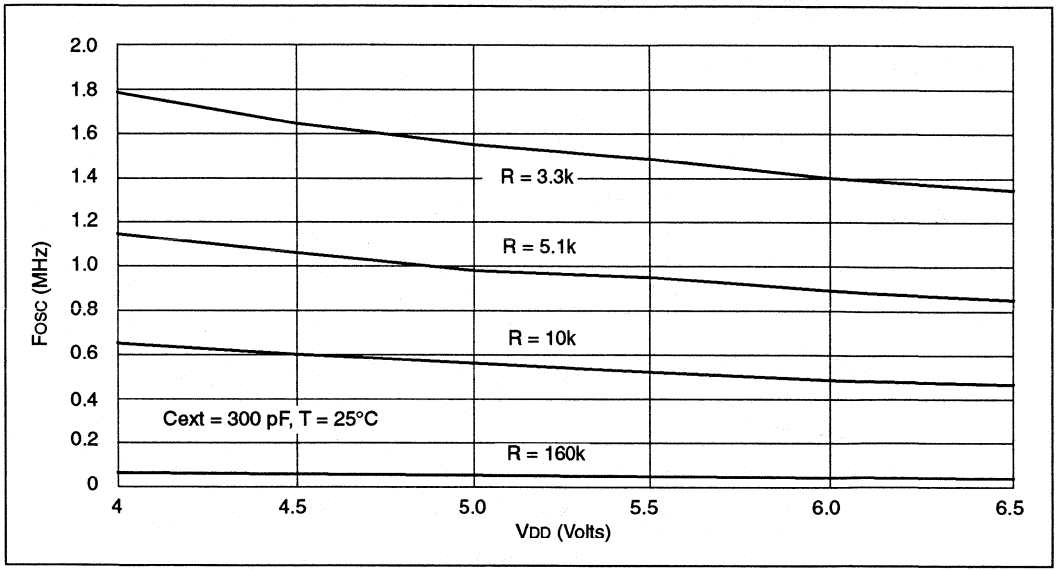


TABLE 18-2: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
22 pF	10k	3.33 MHz	± 12%
	100k	353 kHz	± 13%
100 pF	3.3k	3.54 MHz	± 10%
	5.1k	2.43 MHz	± 14%
	10k	1.30 MHz	± 17%
	100k	129 kHz	± 10%
300 pF	3.3k	1.54 MHz	± 14%
	5.1k	980 kHz	± 12%
	10k	564 kHz	± 16%
	160k	35 kHz	± 18%

© 1995 Microchip Technology Inc.

FIGURE 18-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

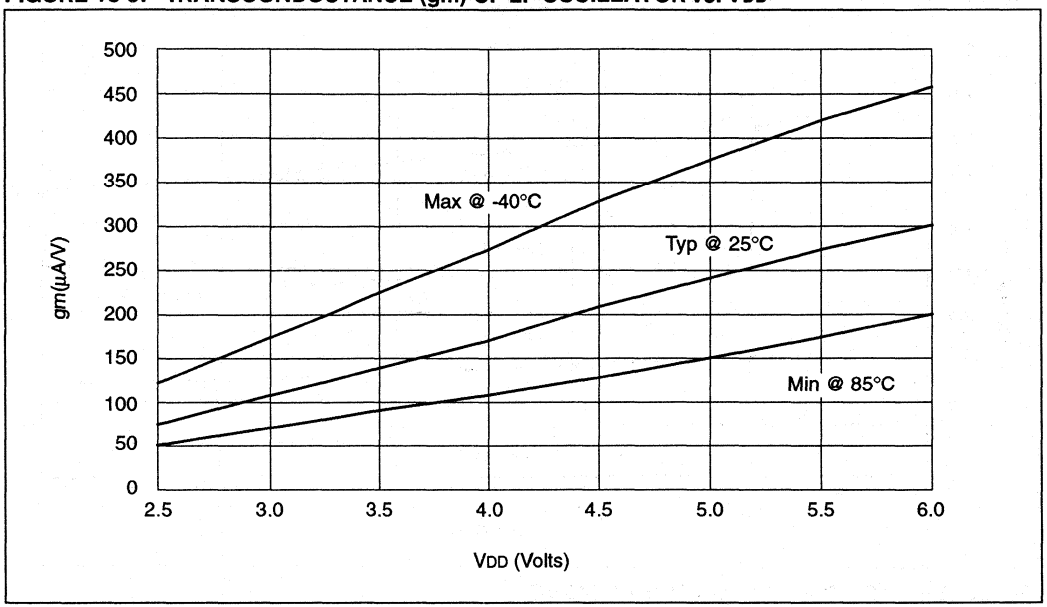
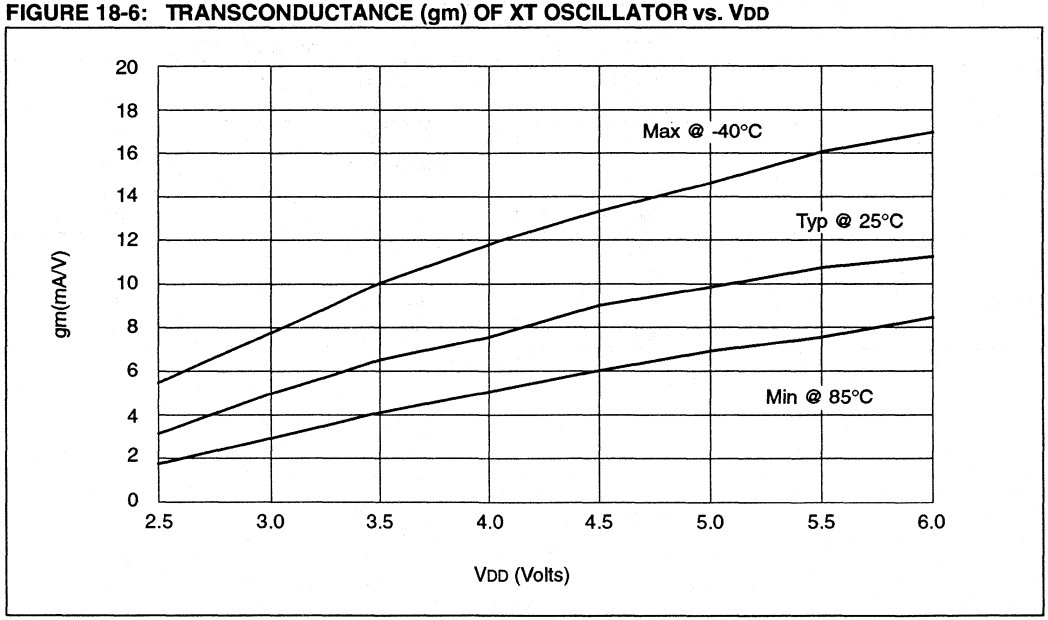


FIGURE 18-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 18-7: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

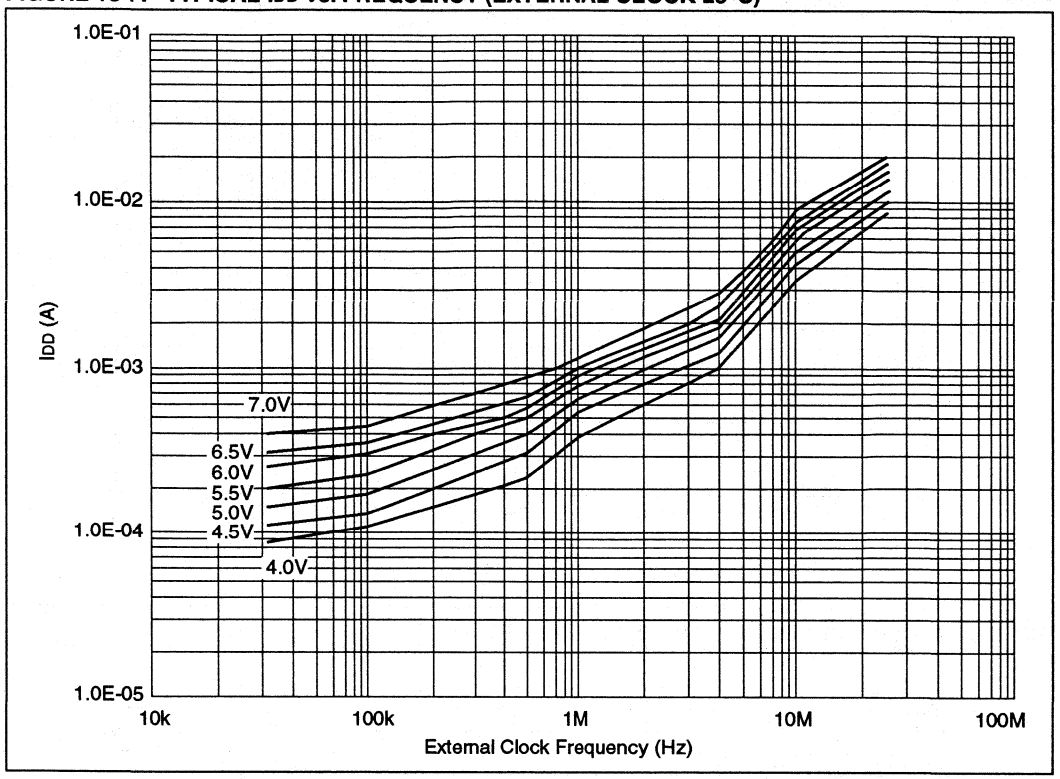


FIGURE 18-8: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK 125°C TO -40°C)

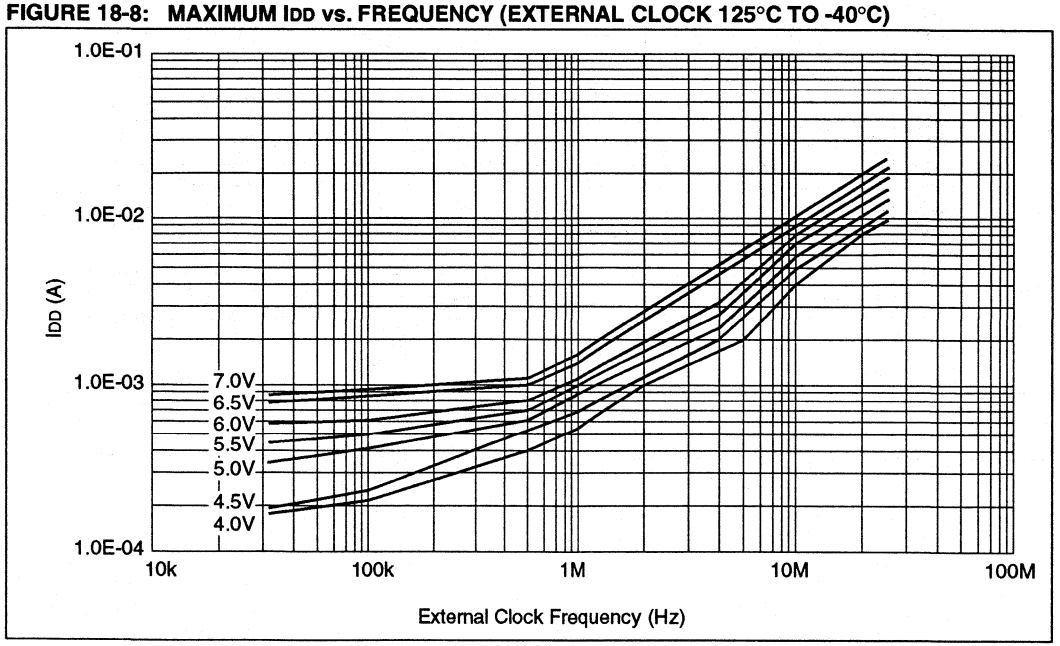


FIGURE 18-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

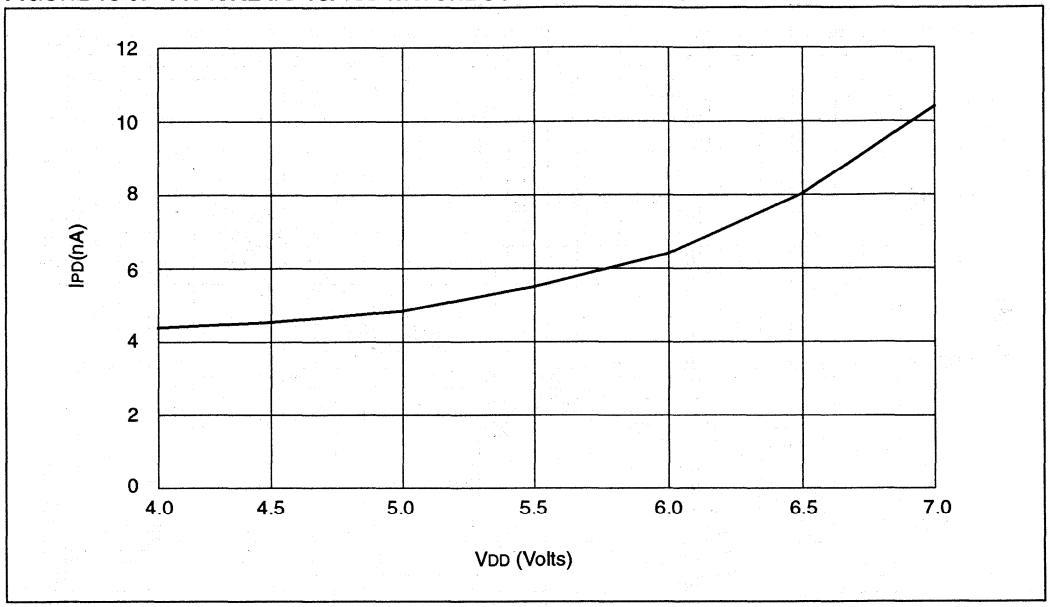
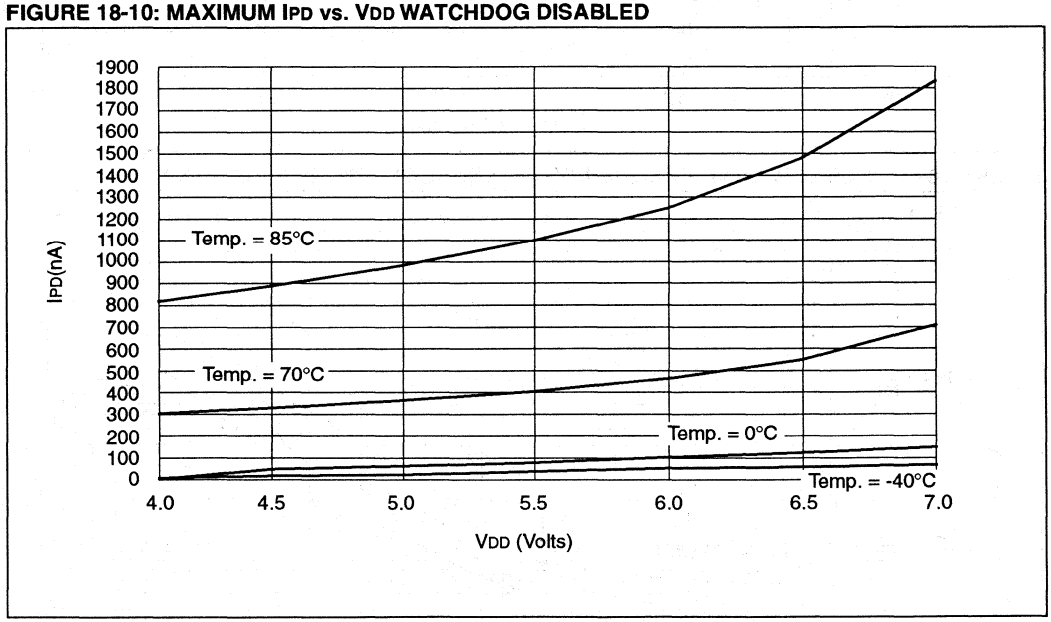


FIGURE 18-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED



© 1995 Microchip Technology Inc.

FIGURE 18-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

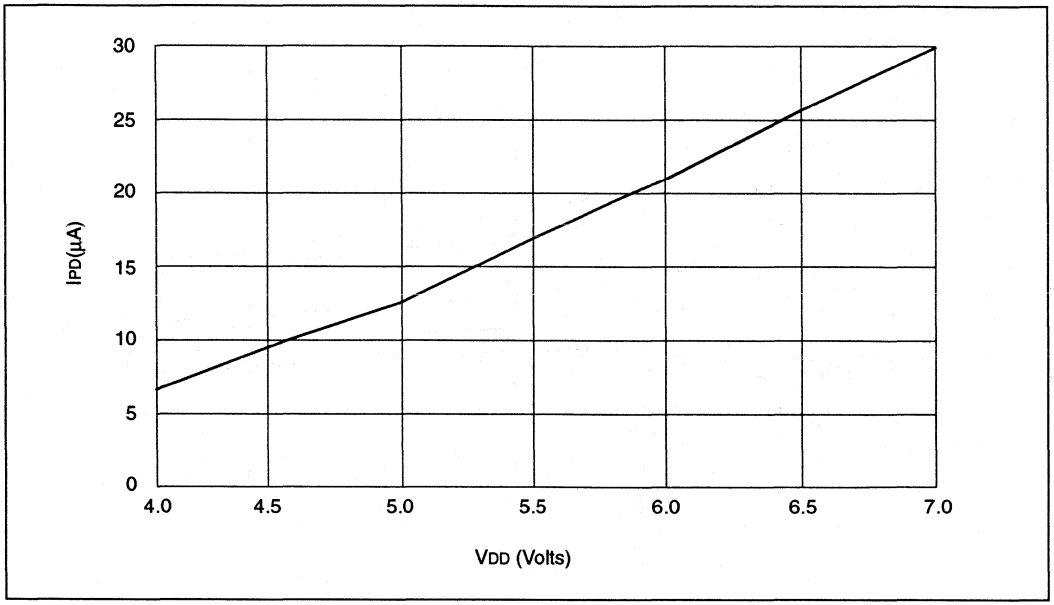


FIGURE 18-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED

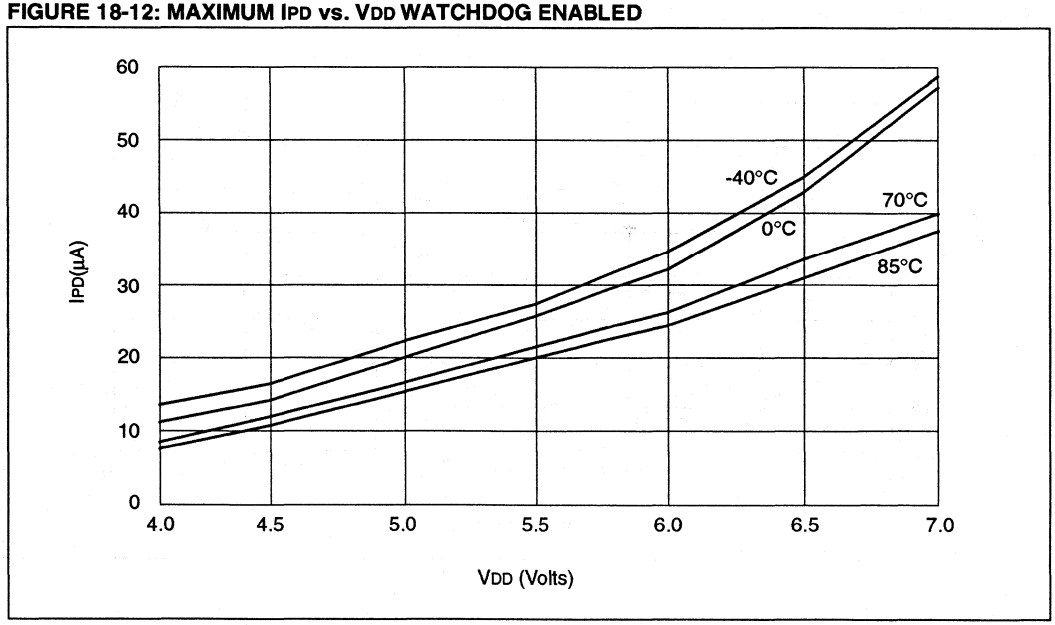


FIGURE 18-13: WDT TIMER TIME-OUT PERIOD vs. VDD

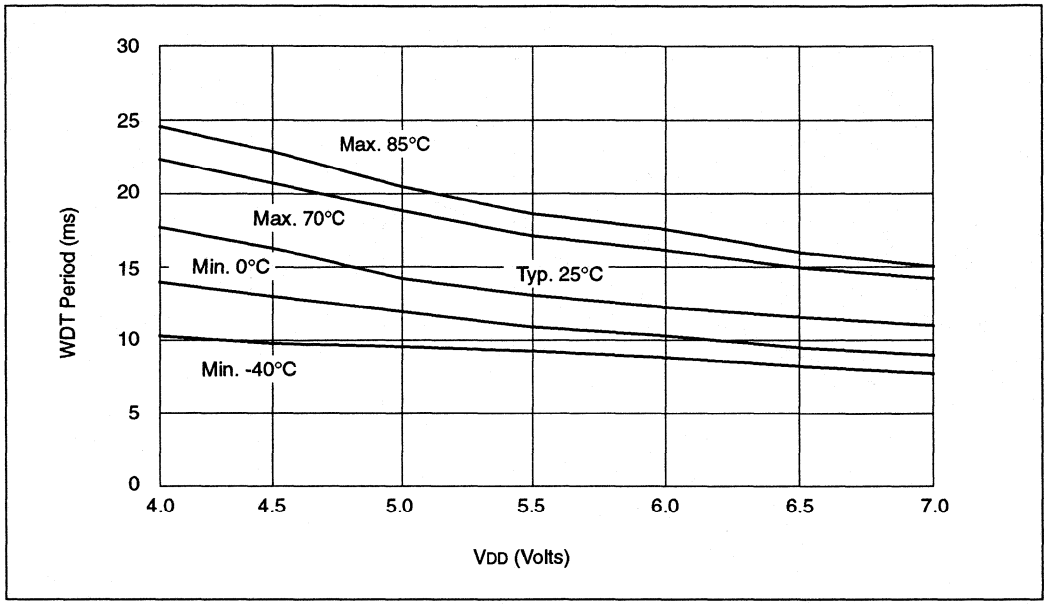
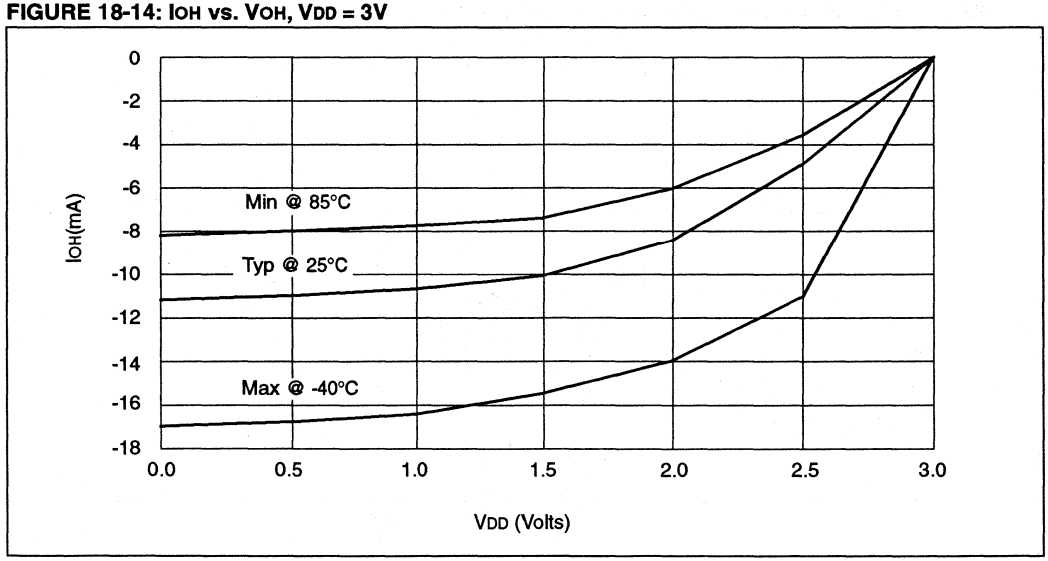


FIGURE 18-14: IOH vs. VOH, VDD = 3V



© 1995 Microchip Technology Inc.

FIGURE 18-15: I_{OH} vs. V_{OH}, V_{DD} = 5V

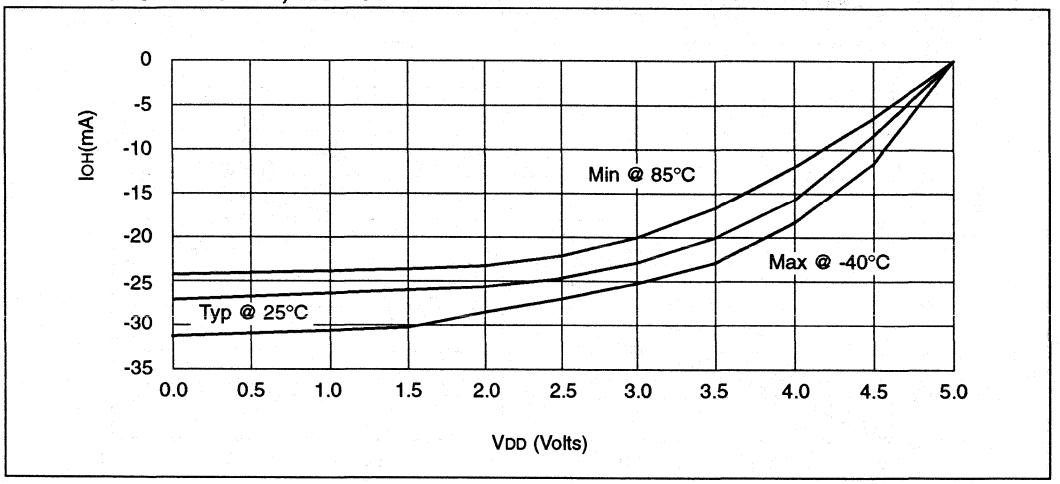
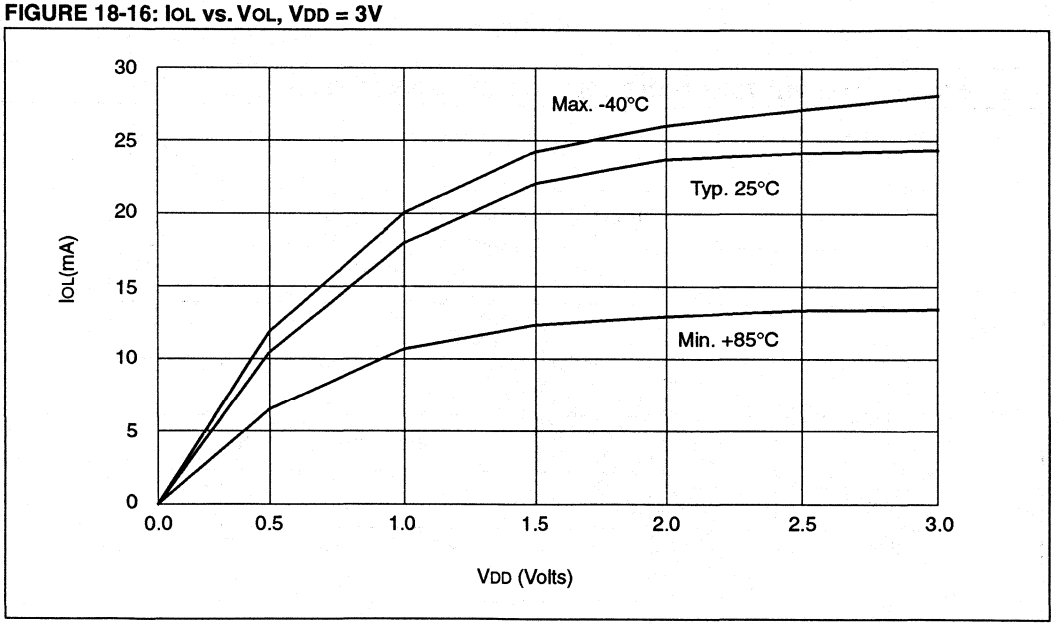


FIGURE 18-16: I_{OL} vs. V_{OL}, V_{DD} = 3V



© 1995 Microchip Technology Inc.

FIGURE 18-17: IOL vs. VOL, VDD = 5V

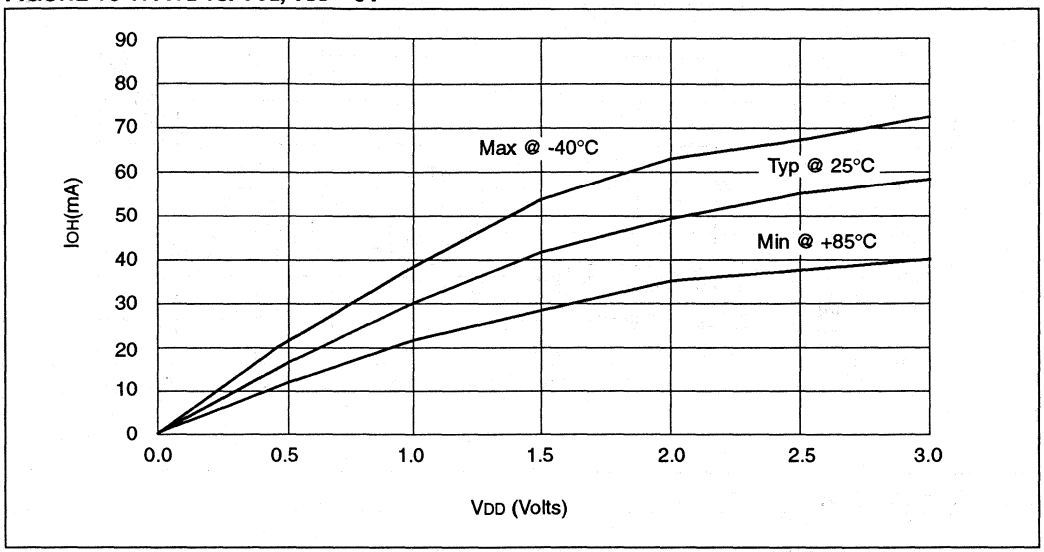
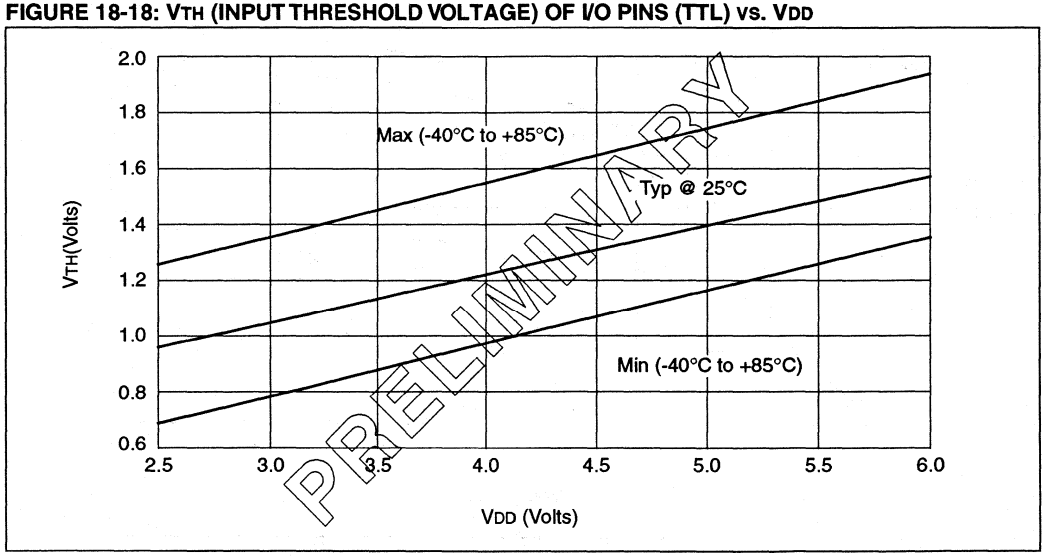


FIGURE 18-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. VDD



PRELIMINARY

© 1995 Microchip Technology Inc.

FIGURE 18-19: VTH, VIL of I/O PINS (SCHMITT TRIGGER) vs. VDD

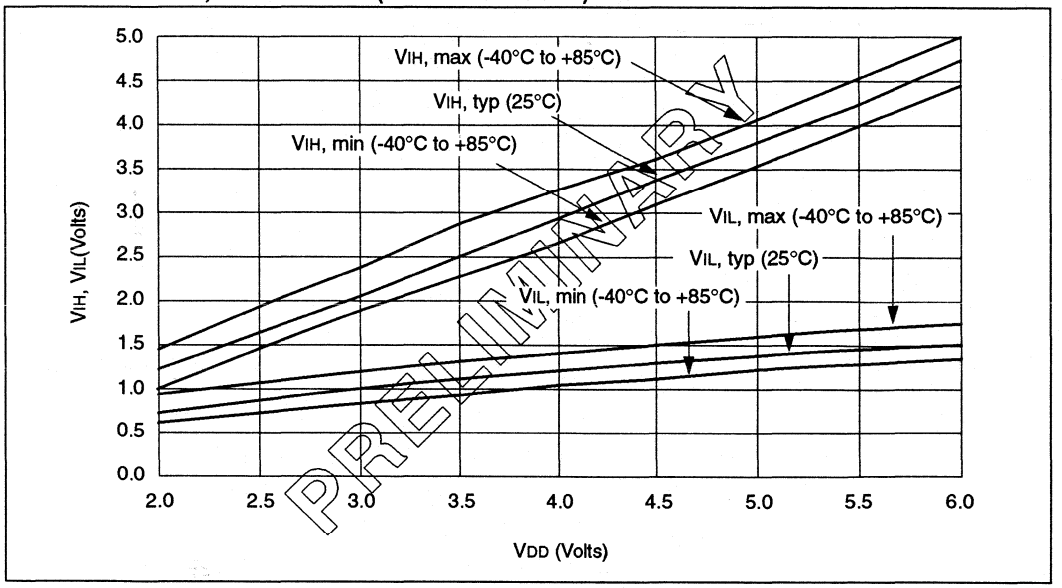
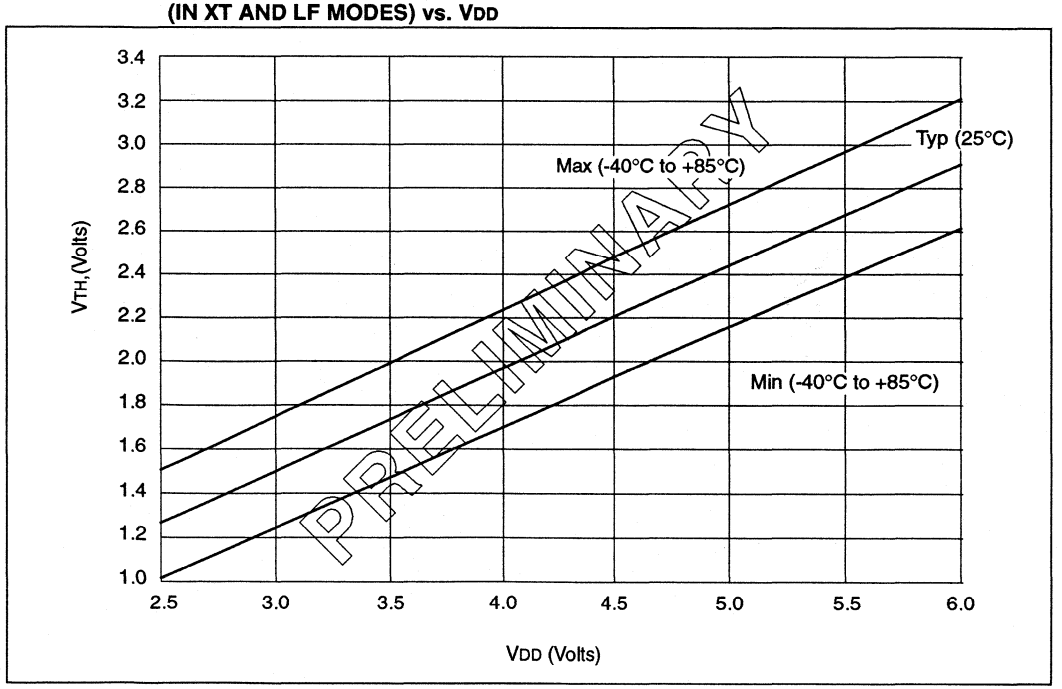


FIGURE 18-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



NOTES:

© 1995 Microchip Technology Inc.

19.0 PIC17C43 AND PIC17C44 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias ..... -55 to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) ..... -0.6V to +14V
Voltage on RA2 and RA3 with respect to V_{SS} ..... -0.6V to +14V
Voltage on all other pins with respect to V_{SS} ..... -0.6V to V_{DD} +0.6V
Total power dissipation (Note 1) 1.0W
Maximum current out of V_{SS} pin(s) - total ..... 250 mA
Maximum current into V_{DD} pin(s) - total ..... 200 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_0 < 0 or V_0 > V_{DD}) ..... ±20 mA
Maximum output current sunk by any I/O pin (except RA2 and RA3) ..... 35 mA
Maximum output current sunk by RA2 or RA3 pins ..... 60 mA
Maximum output current sourced by any I/O pin ..... 20 mA
Maximum current sunk by PORTA and PORTB (combined) ..... 150 mA
Maximum current sourced by PORTA and PORTB (combined) ..... 100 mA
Maximum current sunk by PORTC, PORTD and PORTE (combined) ..... 150 mA
Maximum current sourced by PORTC, PORTD and PORTE (combined) ..... 100 mA

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

TABLE 19-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	17C43-16 17C44-16	17C43-25 17C44-25	17LC43-08 17LC44-08
RC	VDD: 4.5V to 6.0V IDD: 6 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 4 MHz Max.	VDD: 4.5V to 6.0V IDD: 6 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 4 MHz Max.	VDD: 2.5V to 6.0V IDD: 6 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 4 MHz Max.
XT	VDD: 4.5V to 6.0V IDD: 24 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 16 MHz Max.	VDD: 4.5V to 6.0V IDD: 38 mA Max. IPD: 5 µA Max. at 4V WDT disabled Freq: 25 MHz Max.	VDD: 2.5V to 6.0V IDD: 12 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 8 MHz Max.
EC	VDD: 4.5V to 6.0V IDD: 24 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 16 MHz Max	VDD: 4.5V to 6.0V IDD: 38 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 25 MHz Max.	VDD: 2.5V to 6.0V IDD: 12 mA Max. IPD: 5 µA Max. at 6V WDT disabled Freq: 8 MHz Max.
LF	VDD: 4.5V to 6.0V IDD: 95 µA typ. at 32 kHz IPD: < 1 µA typ. at 6V WDT disabled Freq: 2 MHz Max.	VDD: 4.5V to 6.0V IDD: 95 µA typ. at 32 kHz IPD: < 1 µA typ. at 6V WDT disabled Freq: 2 MHz Max.	VDD: 2.5V to 6.0V IDD: 150 µA Max. at 32 kHz IPD: 5 µA Max. at 6.0V WDT disabled Freq: 2 MHz Max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that guarantees the specifications required.

 © 1995 Microchip Technology Inc.

19.1 DC CHARACTERISTICS: PIC17C43/C44-16 (Commercial, Industrial) PIC17C43/C44-25 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 4.5V to 6.0V				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	4.5	–	6.0	V	
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to guarantee Power-On Reset	–	VSS	–	V	See section on Power-On Reset for details
D004	SVDD	VDD rise rate to guarantee Power-On Reset	0.060 *	–	–	mV/ms	See section on Power-On Reset for details
D010	IDD	Supply Current (Note 2)	–	3	6	mA	FOSC = 4 MHz (Note 4)
D011			–	6	12 *	mA	FOSC = 8 MHz
D012			–	11	24 *	mA	FOSC = 16 MHz
D013			–	19	38	mA	FOSC = 25 MHz
D014			–	95	150	µA	FOSC = 32 kHz, WDT enabled (EC osc configuration)
D020	IPD	Power Down Current (Note 3)	–	10	40	µA	VDD = 6.0V, WDT enabled
D021			–	< 1	5	µA	VDD = 6.0V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD or VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
Current consumed from the oscillator and I/O's driving external capacitive or resistive loads needs to be considered.
For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD / (2 • R).
For capacitive loads, The current can be estimated (for an individual I/O pin) as (CL • VDD) • f
CL = Total capacitive load on the I/O pin; f = average frequency the I/O pin switches.
The capacitive currents are most significant when the device is configured for external execution (includes extended microcontroller mode).

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

19.2 DC CHARACTERISTICS: PIC17LC43/LC44 (Commercial, Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial Operating voltage VDD = 2.5V to 6.0V				
Parameter No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	2.5	–	6.0	V	
D002	VDR	RAM Data Retention Voltage (Note 1)	1.5 *	–	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to guarantee Power-On Reset	–	VSS	–	V	See section on Power-On Reset for details
D004	SVDD	VDD rise rate to guarantee Power-On Reset	0.060 *	–	–	mV/ms	See section on Power-On Reset for details
D010	IDD	Supply Current (Note 2)	–	3	6	mA	FOSC = 4 MHz (Note 4)
D011			–	6	12 *	mA	FOSC = 8 MHz
D014			–	95	150	µA	FOSC = 32 kHz, WDT disabled (EC osc configuration)
D020	IPD	Power Down Current (Note 3)	–	10	40	µA	VDD = 6.0V, WDT enabled
D021			–	< 1	5	µA	VDD = 6.0V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1=external square wave, from rail to rail; all I/O pins tristated, pulled to VDD or VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.
Current consumed from the oscillator and I/O's driving external capacitive or resistive loads needs to be considered.
For the RC oscillator, the current through the external pull-up resistor (R) can be estimated as: VDD / (2 • R).
For capacitive loads, The current can be estimated (for an individual I/O pin) as (CL • VDD) • f
CL = Total capacitive load on the I/O pin; f = average frequency the I/O pin switches.
The capacitive currents are most significant when the device is configured for external execution (includes extended microcontroller mode)

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula IR = VDD/2Rext (mA) with Rext in kOhm.

 © 1995 Microchip Technology Inc.

19.3 DC CHARACTERISTICS: PIC17C43/C44-16 (Commercial, Industrial)
PIC17C43/C44-25 (Commercial, Industrial)
PIC17LC43/LC44-08 (Commercial, Industrial)

Standard Operating Conditions (unless otherwise stated)
Operating temperature
DC CHARACTERISTICS -40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in Section 19.1

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Input Low Voltage					
	VIL	I/O ports					
D030		with TTL buffer	VSS	–	0.8	V	PIC17C43/C44
			VSS	–	0.2 VDD	V	PIC17LC43/LC44
D031		with Schmitt Trigger buffer	VSS	–	0.2 VDD	V	
D032		$\overline{MCLR}$, OSC1 (in EC and RC mode)	VSS	–	0.2 VDD	V	Note1
D033		OSC1 (in XT, and LF mode)	–	0.5 VDD	–	V	
		Input High Voltage					
	VIH	I/O ports		–			
D040		with TTL buffer	2.0	–	VDD	V	PIC17C43/C44
			1+0.2 VDD	–	VDD	V	PIC17LC43/LC44
D041		with Schmitt Trigger buffer	0.8 VDD	–	VDD	V	
D042		$\overline{MCLR}$	0.8 VDD	–	VDD	V	Note1
D043		OSC1 (XT, and LF mode)	–	0.5 VDD	–	V	
D050	VHYS	Hysteresis of Schmitt Trigger inputs	0.15 VDD *	–	–	V	
		Input Leakage Current (Notes 2, 3)					
D060	IIL	I/O ports (except RA2, RA3)	–	–	±1	µA	VSS ≤ VPIN ≤ VDD, I/O Pin at hi-impedance PORTB weak pull-ups disabled
D061		$\overline{MCLR}$	–	–	±2	µA	VPIN = VSS or VPIN = VDD
D062		RA2, RA3			±2	µA	VSS ≤ VRA2, VRA3 ≤ 12V
D063		OSC1, TEST	–	–	±1	µA	VSS ≤ VPIN ≤ VDD
D064		$\overline{MCLR}$	–	–	10	µA	VMCLR = VPP = 12V (when not programming)
D070	IPURB	PORTB weak pull-up current	60	200	400	µA	VPIN = VSS, $\overline{RBPU}$ = 0 4.5V ≤ VDD ≤ 6.0V

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The $\overline{MCLR}$/Vpp pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature
-40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in Section 19.1

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Output Low Voltage					
D080	VOL	I/O ports (except RA2 and RA3)					IOL = VDD/1.250 mA
			–	–	0.1 VDD	V	4.5 V ≤ VDD ≤ 6.0 V
			–	–	0.1 VDD *	V	VDD = 2.5 V
D081		with TTL buffer	–	–	0.4	V	IOL = 6 mA, VDD = 4.5V Note 6
D082		RA2 and RA3	–	–	3.0	V	IOL = 60.0 mA, VDD = 6.0V
D083		OSC2/CLKOUT	–	–	0.4	V	IOL = 2 mA, VDD = 4.5V
D084		(RC and EC osc modes)	–	–	0.1 VDD *	V	IOL = VDD/2.500 mA (PIC17LC43/LC44 only)
		Output High Voltage (Note 3)					
D090	VOH	I/O ports (except RA2 and RA3)					IOH = -VDD/2.500 mA
			0.9 VDD	–	–	V	4.5 V ≤ VDD ≤ 6.0 V
			0.9 VDD *	–	–	V	VDD = 2.5 V
D091		with TTL buffer	2.4	–	–	V	IOH = -6.0 mA, VDD=4.5V Note 6
D092		RA2 and RA3	–	–	12	V	Pulled-up to externally applied voltage
D093		OSC2/CLKOUT	2.4	–	–	V	IOH = -5 mA, VDD = 4.5V
D094		(RC and EC osc modes)	0.9 VDD *	–	–	V	IOH = -VDD/2.500 mA (PIC17LC43/LC44 only)
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2/CLKOUT pin	–	–	25 ††	pF	In EC or RC osc modes when OSC2 pin is outputting CLKOUT. external clock is used to drive OSC1.
D101	CIO	All I/O pins and OSC2 (in RC mode)	–	–	50 ††	pF	
D102	CAD	System Interface Bus (PORTC, PORTD and PORTE)	–	–	100 ††	pF	In Microprocessor or Extended Microcontroller mode

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

te 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The $\overline{MCLR}$/Vpp pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

 © 1995 Microchip Technology Inc.

DC CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature
-40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in Section 19.1

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
		Internal Program Memory Programming Specs (Note 4)					
D110	Vpp	Voltage on MCLR/VPP pin	12.5	–	13.5	V	Note 5
D111	Vddp	Supply voltage during programming	4.75	5.0	5.25	V	
D112	Ipp	Current into MCLR/VPP pin	–	25 ‡	50 ‡	mA	
D113	Iddp	Supply current during programming	–	–	30 ‡	mA	
D114	Tprog	Programming pulse width	10	100	1000	µs	Terminated via internal/external interrupt or a reset

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The MCLR/Vpp pin may be kept in this range at times other than programming, but is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

19.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST (I²C specifications only)
4. Ts (I²C specifications only)

T			
F	Frequency	T	Time

Lowercase symbols (pp) and their meanings:

pp			
ad	Address/Data	ost	Oscillator Start-Up Timer
al	ALE	pwrt	Power-Up Timer
cc	Capture1 and Capture2	rb	PORTB
ck	CLKOUT or clock	rd	$\overline{RD}$
dt	Data in	rw	$\overline{RD}$ or $\overline{WR}$
in	INT pin	t0	T0CKI
io	I/O port	t123	TCLK12 and TCLK3
mc	$\overline{MCLR}$	wdt	Watchdog Timer
oe	$\overline{OE}$	wr	$\overline{WR}$
os	OSC1		

Uppercase symbols and their meanings:

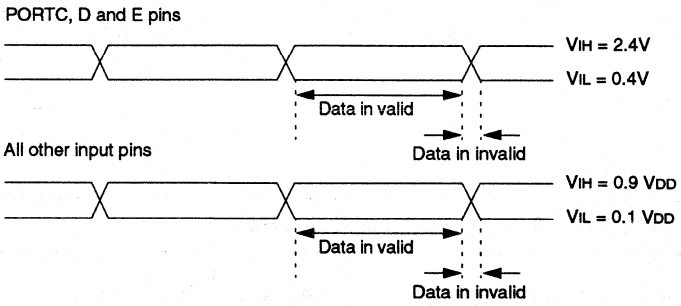
S			
D	Driven	L	Low
E	Edge	P	Period
F	Fall	R	Rise
H	High	V	Valid
I	Invalid (Hi-impedance)	Z	High Impedance

 © 1995 Microchip Technology Inc.

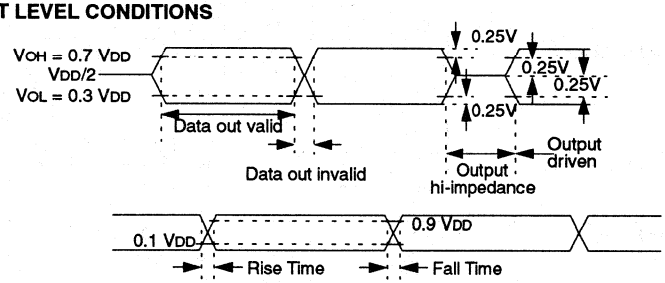
FIGURE 19-1: PARAMETER MEASUREMENT INFORMATION

All timings are measure between high and low measurement points as indicated in the figures below.

INPUT LEVEL CONDITIONS

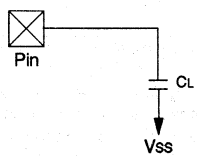


OUTPUT LEVEL CONDITIONS



LOAD CONDITIONS

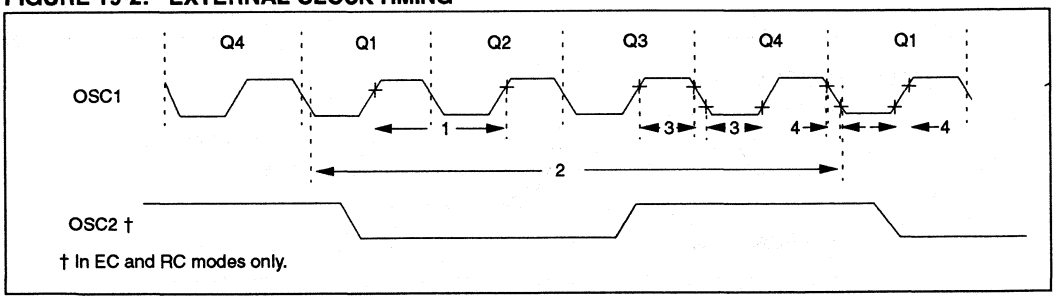
Load Condition 1



$50 \leq C_L \leq 100$ pF

19.5 Timing Diagrams and Specifications

FIGURE 19-2: EXTERNAL CLOCK TIMING



† In EC and RC modes only.

TABLE 19-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	**External CLKIN Frequency (Note 1)**	DC	—	8	MHz	EC osc mode - PIC17LC43/44-08
			DC	—	16	MHz	- PIC17C43/44-16
			DC	—	25	MHz	- PIC17C43/44-25
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			1	—	8	MHz	XT osc mode - PIC17LC43/44-08
			1	—	16	MHz	- PIC17C43/44-16
			1	—	25	MHz	- PIC17C43/44-25
			DC	—	2	MHz	LF osc mode
1	Tosc	**External CLKIN Period (Note 1)**	125	—	—	ns	EC osc mode - PIC17LC43/44-08
			62.5	—	—	ns	- PIC17C43/44-16
			40	—	—	ns	- PIC17C43/44-25
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			125	—	1,000	ns	XT osc mode - PIC17LC43/44-08
			62.5	—	1,000	ns	- PIC17C43/44-16
			40	—	1,000	ns	- PIC17C43/44-25
			500	—	—	ns	LF osc mode
2	Tcy	**Instruction Cycle Time (Note 1)**	160	4/Fosc	DC	ns	
3	TosL, TosH	Clock in (OSC1) high or low time	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	Clock in (OSC1) rise or fall time	—	—	5 ‡	ns	EC oscillator

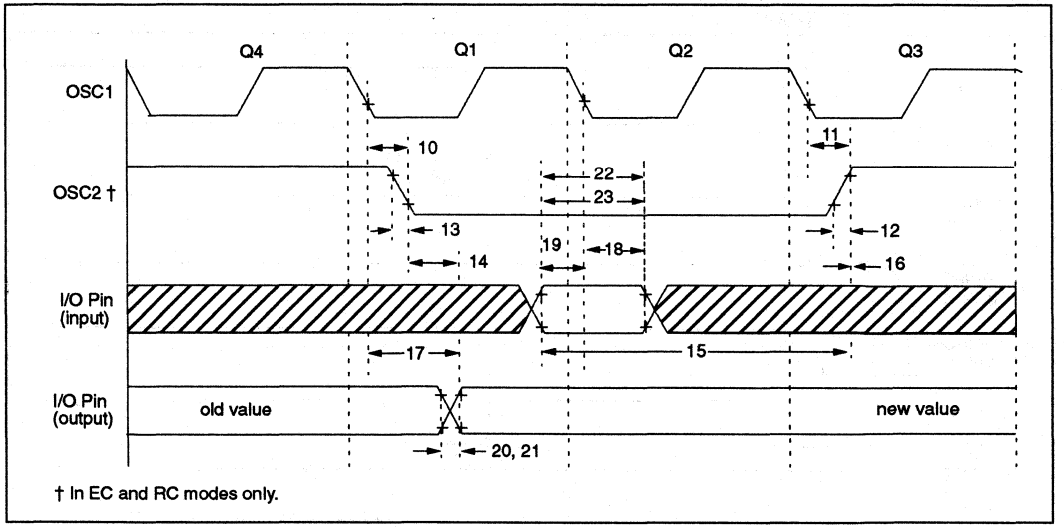
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

 © 1995 Microchip Technology Inc.

FIGURE 19-3: CLKOUT AND I/O TIMING



† In EC and RC modes only.

TABLE 19-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↓ to CLKOUT↓	—	15 ‡	30 ‡	ns	Note 1
11	TosH2ckH	OSC1↓ to CLKOUT↑	—	15 ‡	30 ‡	ns	Note 1
12	TckR	CLKOUT rise time	—	5 ‡	15 ‡	ns	Note 1
13	TckF	CLKOUT fall time	—	5 ‡	15 ‡	ns	Note 1
14	TckH2ioV	CLKOUT ↑ to Port out valid	—	—	0.5 TCY+20 ‡	ns	Note 1
15	TioV2ckH	Port in valid before CLKOUT↑ 17C43/44 17LC43/44	0.25 TCY+25 ‡ 0.25 TCY+50 ‡	— —	— —	ns ns	Note 1 Note 1
16	TckH2ioI	Port in hold after CLKOUT↑	0 ‡	—	—	ns	Note 1
17	TosH2ioV	OSC1↓ (Q1 cycle) to Port out valid	—	—	100 ‡	ns	
18	TosH2ioI	OSC1↓ (Q2 cycle) to Port input invalid (I/O in hold time)	0 ‡	—	—	ns	
19	TioV2osH	Port input valid to OSC1↓ (I/O in setup time)	30 ‡	—	—	ns	
20	TioR	Port output rise time	—	10 ‡	35 ‡	ns	
21	TioF	Port output fall time	—	10 ‡	35 ‡	ns	
22	TinHL	INT pin high or low time	25 *	—	—	ns	
23	TrbHL	RB<7:0> change INT high or low time	25 *	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

Note 1: Measurements are taken in EC Mode where CLKOUT output is 4 x Tosc.

FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

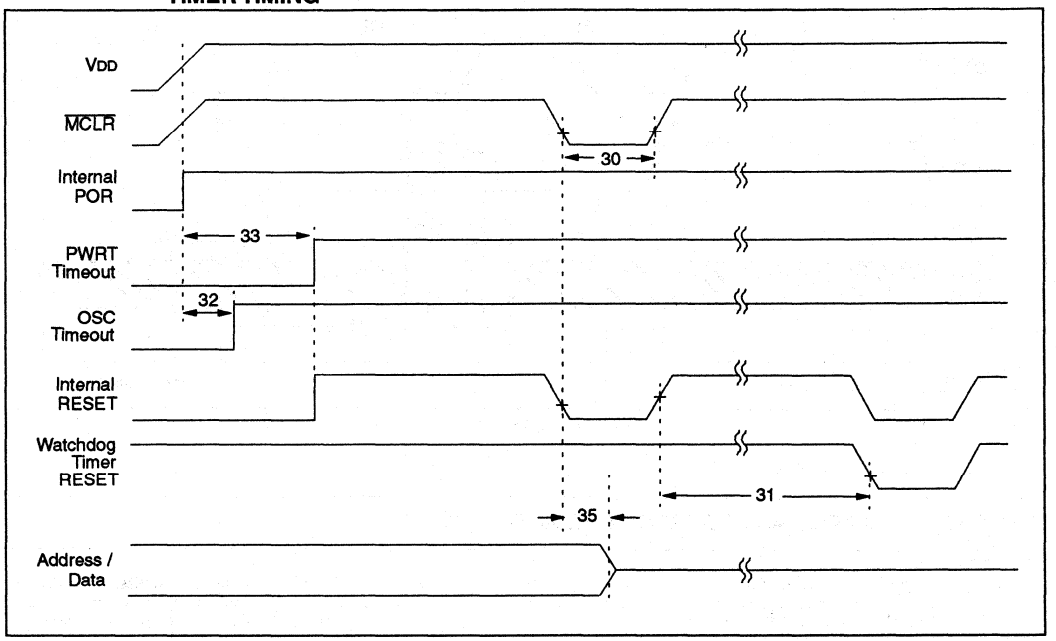


TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	100 *	—	—	ns	VDD = 5V
31	Twdt	Watchdog Timer Timeout Period (Prescale = 1)	5 *	12	25 *	ms	VDD = 5V
32	Tost	Oscillation Start-Up Timer Period		1024 Tosc §		ms	Tosc = OSC1 period
33	Tpwrt	Power-Up Timer Period	40 *	96	200 *	ms	VDD = 5V
35	TmcL2adl	$\overline{MCLR}$ to System Interface bus (AD<15:0>) invalid	—	—	100 *	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

‡ These parameters are for design guidance only and are not tested, nor characterized.

§ This specification guaranteed by design.

 © 1995 Microchip Technology Inc.

FIGURE 19-5: TIMER0 CLOCK TIMINGS

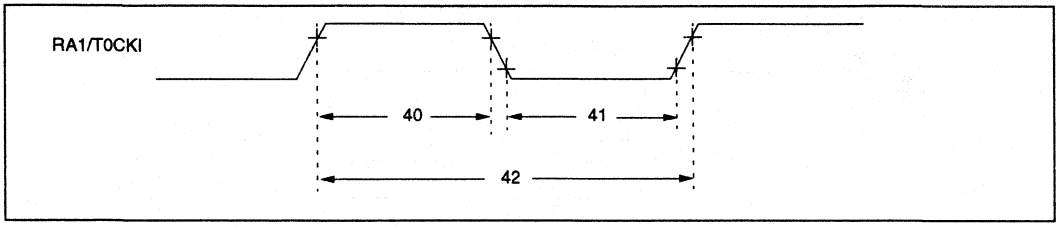


TABLE 19-5: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 Tcy + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 Tcy + 20 §	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		Tcy + 40 § N	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK TIMINGS

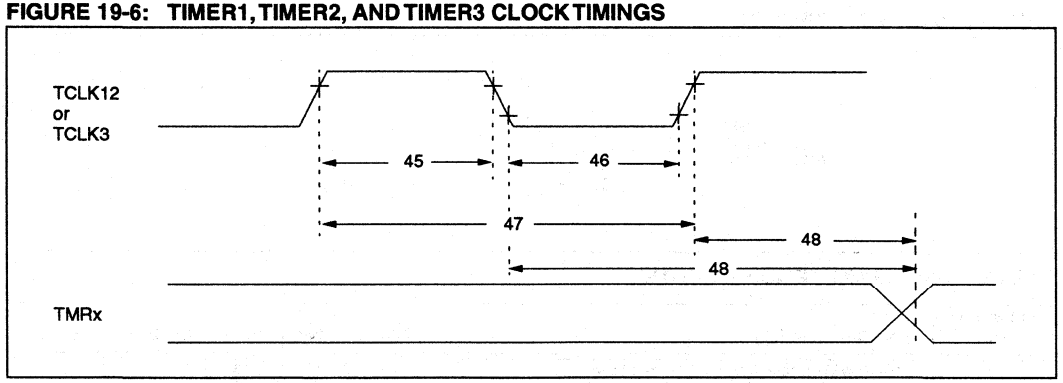


TABLE 19-6: TIMER1, TIMER2, AND TIMER3 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ †	Max	Units	Conditions
45	Tt123H	TCLK12 and TCLK3 high time	0.5 Tcy + 20 §	—	—	ns	
46	Tt123L	TCLK12 and TCLK3 low time	0.5 Tcy + 20 §	—	—	ns	
47	Tt123P	TCLK12 and TCLK3 input period	Tcy + 40 § N	—	—	ns	N = prescale value (1, 2, 4, 8)
48	TckE2tmrI	Delay from selected External Clock Edge to Timer increment	2 Tosc §		6 Tosc §		

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 19-7: CAPTURE TIMINGS

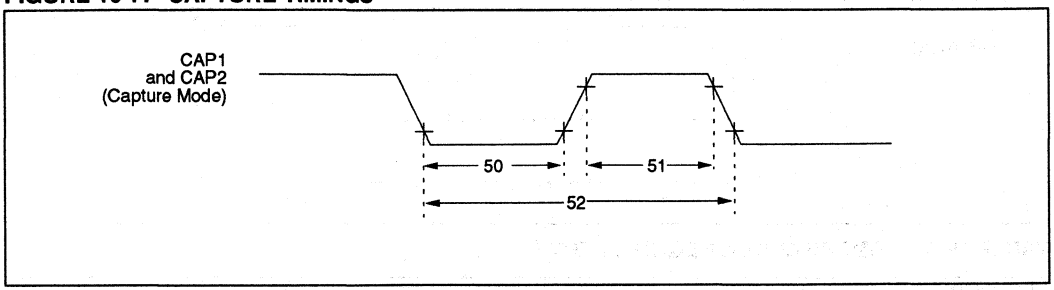


TABLE 19-7: CAPTURE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
50	TccL	Capture1 and Capture2 input low time	10 *	—	—	ns	
51	TccH	Capture1 and Capture2 input high time	10 *	—	—	ns	
52	TccP	Capture1 and Capture2 input period	2 Tcy § / N	—	—	ns	N = prescale value (4 or 16)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

FIGURE 19-8: PWM TIMINGS

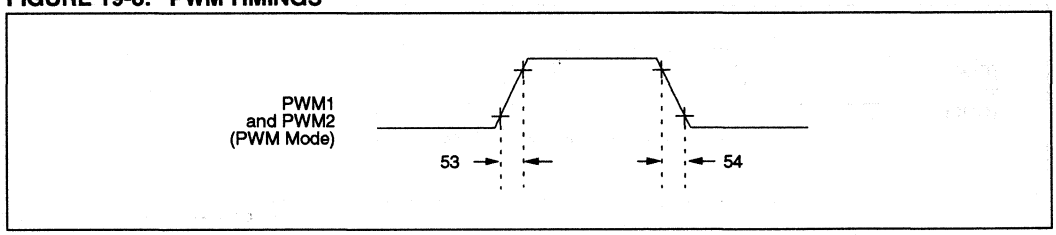


TABLE 19-8: PWM REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
53	TccR	PWM1 and PWM2 output rise time	—	10 *	35 *§	ns	
54	TccF	PWM1 and PWM2 output fall time	—	10 *	35 *§	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

 © 1995 Microchip Technology Inc.

FIGURE 19-9: SCI MODULE: SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

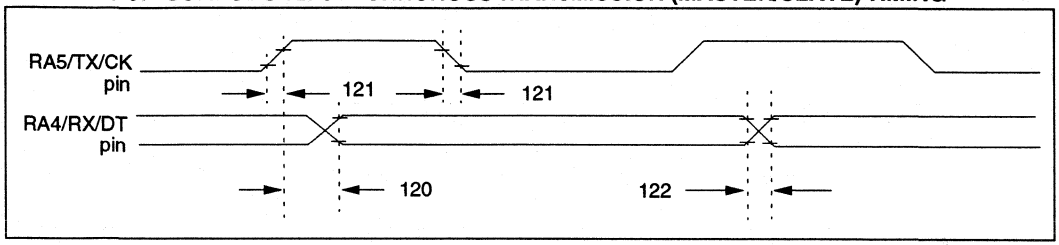


TABLE 19-9: SERIAL PORT SYNCHRONOUS TRANSMISSION REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
120	TckH2dtV	SYNC XMIT (MASTER & SLAVE)						
		Clock high to data out valid	17C43/44	—	—	50	ns	
			17LC43/44	—	—	75	ns	
121	TckRF	Clock out rise time and fall time (Master Mode)	17C43/44	—	—	25	ns	
			17LC43/44	—	—	40	ns	
122	TdtRF	Data out rise time and fall time	17C43/44	—	—	25	ns	
			17LC43/44	—	—	40	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-10: SCI MODULE: SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

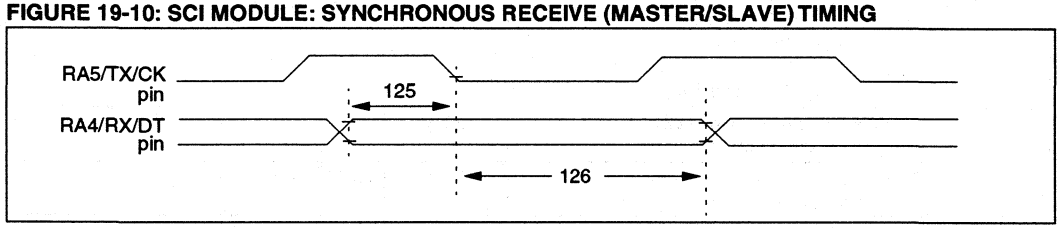


TABLE 19-10: SERIAL PORT SYNCHRONOUS RECEIVE REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
125	TdtV2ckL	SYNC RCV (MASTER & SLAVE) Data hold before CK↓ (DT hold time)	15	—	—	ns	
126	TckL2dtl	Data hold after CK↓ (DT hold time)	15	—	—	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 19-11: MEMORY INTERFACE WRITE TIMING (NOT SUPPORTED IN PIC17LC4X DEVICES)

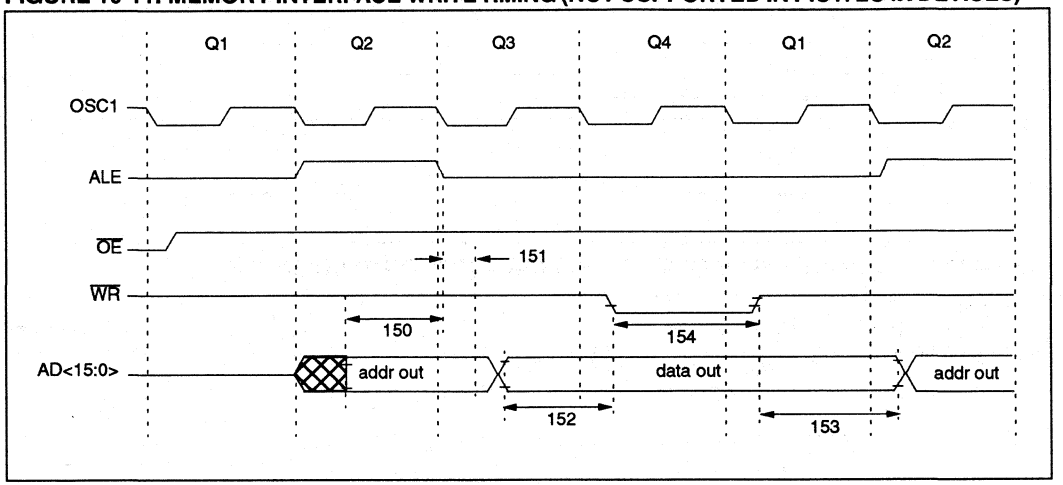


TABLE 19-11: MEMORY INTERFACE WRITE REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25 Tcy-30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	—	—	ns	
152	TadV2wrL	Data out valid to WR↓ (data setup time)	0.25 Tcy-40	—	—	ns	
153	TwrH2adI	WR↑ to data out invalid (data hold time)	—	0.25 TCY §	—	ns	
154	TwrL	WR pulse width	—	0.25 TCY §	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

 © 1995 Microchip Technology Inc.

FIGURE 19-12: MEMORY INTERFACE READ TIMING (NOT SUPPORTED IN PIC17LC4X DEVICES)

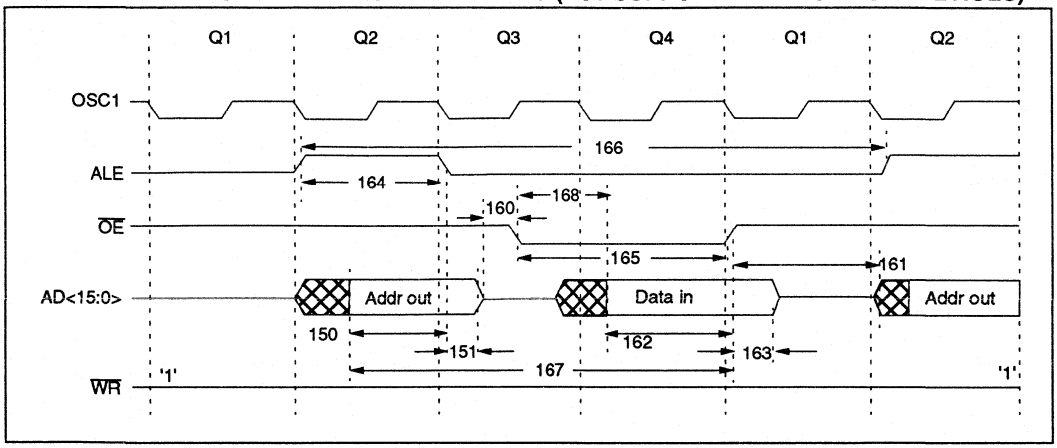


TABLE 19-12: MEMORY INTERFACE READ REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25 Tcy-30	—	—	ns	
151	TalL2adI	ALE↓ to address out invalid (address hold time)	0	—	—	ns	
160	TadZ2oeL	AD<15:0> hi-impedance to $\overline{OE}$↓	10	—	—	ns	
161	ToeH2adD	$\overline{OE}$↑ to AD<15:0> driven	0.25 Tcy-15	—	—	ns	
162	TadV2oeH	Data in valid before $\overline{OE}$↑ (data setup time)	35	—	—	ns	
163	ToeH2adI	$\overline{OE}$↑to data in invalid (data hold time)	0	—	—	ns	
164	TalH	ALE pulse width	—	0.25 Tcy §	—	ns	
165	ToeL	$\overline{OE}$ pulse width	0.5 Tcy-35 §	—	—	ns	
166	TalH2alH	ALE↑ to ALE↑(cycle time)	—	Tcy §	—	ns	
167	Tacc	Address access time	—	—	0.75 Tcy-30	ns	
168	Toe	Output enable access time ($\overline{OE}$ low to Data Valid)	—	—	0.5 Tcy - 45	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

§ This specification guaranteed by design.

NOTES:

 © 1995 Microchip Technology Inc.

20.0 PIC17C43 AND PIC17C44 DC AND AC CHARACTERISTICS

The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g. outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. "Typical" represents the mean of the distribution while "max" or "min" represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

TABLE 20-1: PIN CAPACITANCE PER PACKAGE TYPE

Pin Name	Typical Capacitance (pF)			
	40-pin DIP	44-pin PLCC	44-pin MQFP	44-pin TQFP
All pins, except $\overline{MCLR}$, VDD, and VSS	10	10	10	10
$\overline{MCLR}$ pin	20	20	20	20

FIGURE 20-1: TYPICAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE

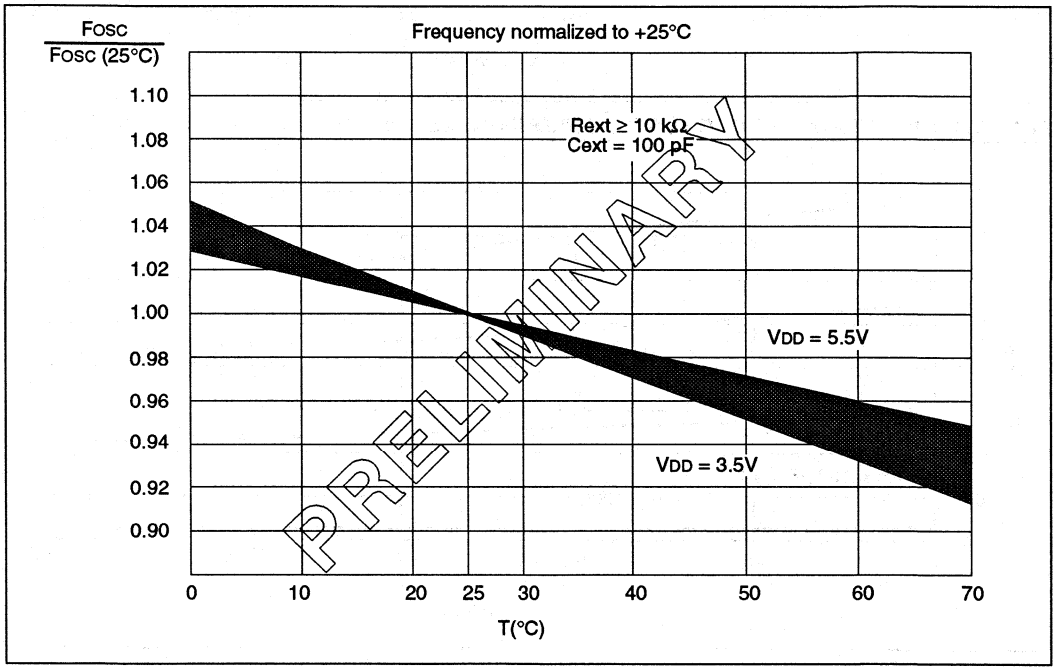


FIGURE 20-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

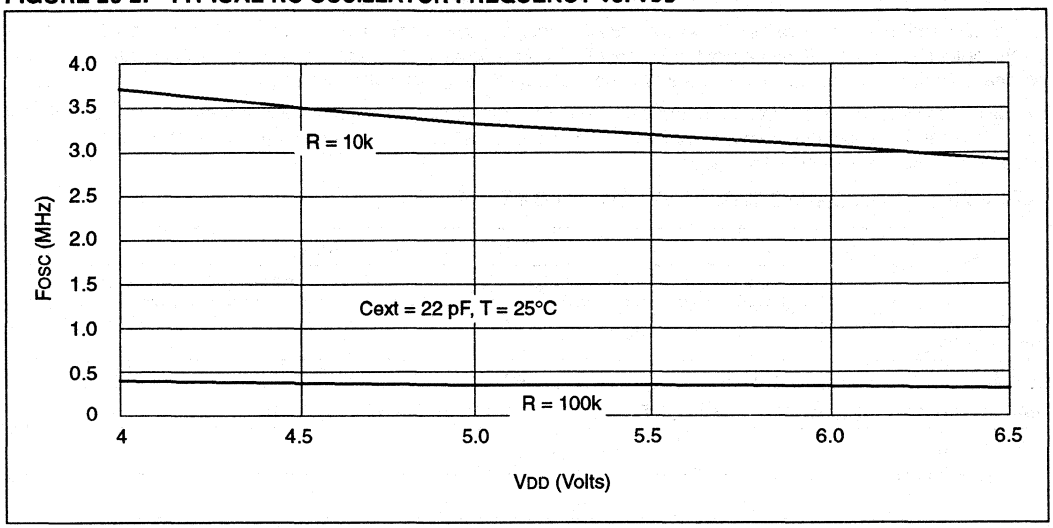
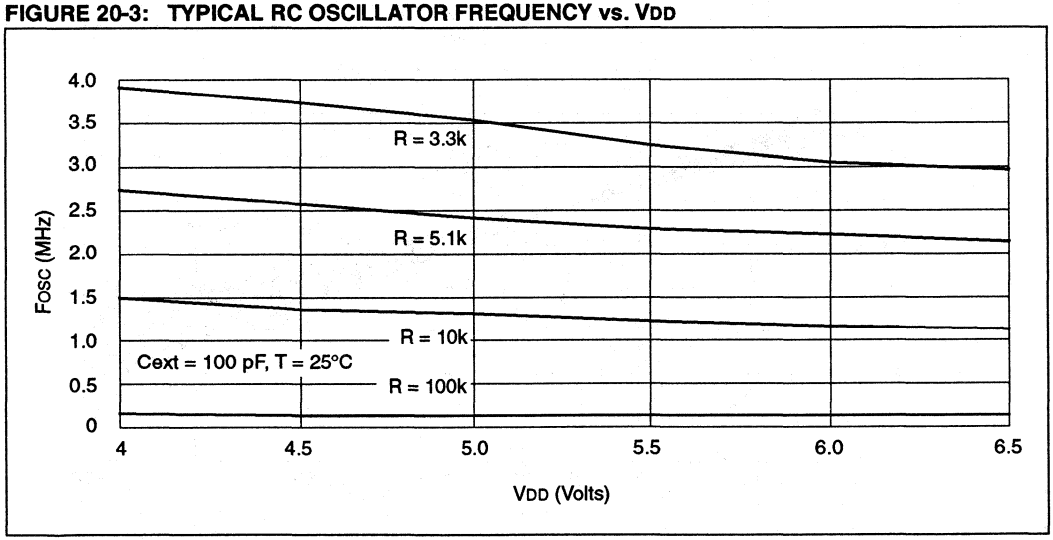


FIGURE 20-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 20-4: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD

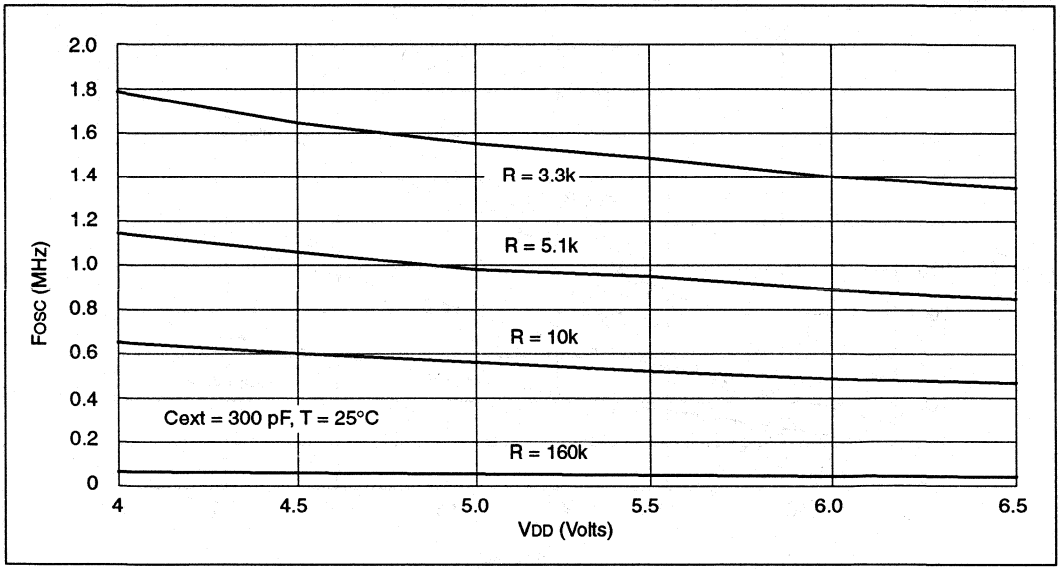


TABLE 20-2: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average Fosc @ 5V, 25°C	
22 pF	10k	3.33 MHz	± 12%
	100k	353 kHz	± 13%
100 pF	3.3k	3.54 MHz	± 10%
	5.1k	2.43 MHz	± 14%
	10k	1.30 MHz	± 17%
	100k	129 kHz	± 10%
300 pF	3.3k	1.54 MHz	± 14%
	5.1k	980 kHz	± 12%
	10k	564 kHz	± 16%
	160k	35 kHz	± 18%

FIGURE 20-5: TRANSCONDUCTANCE (gm) OF LF OSCILLATOR vs. VDD

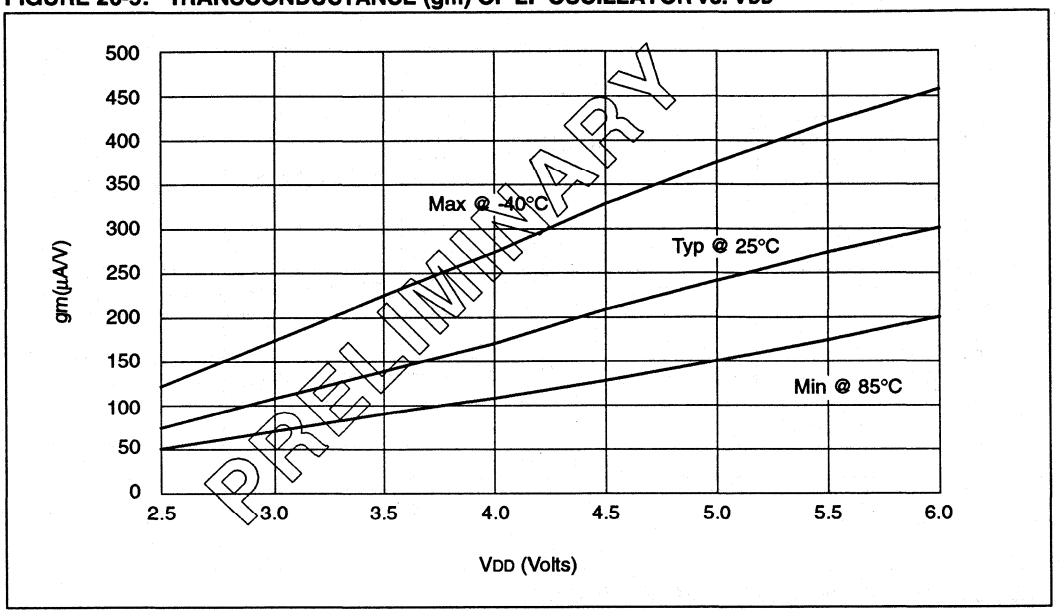
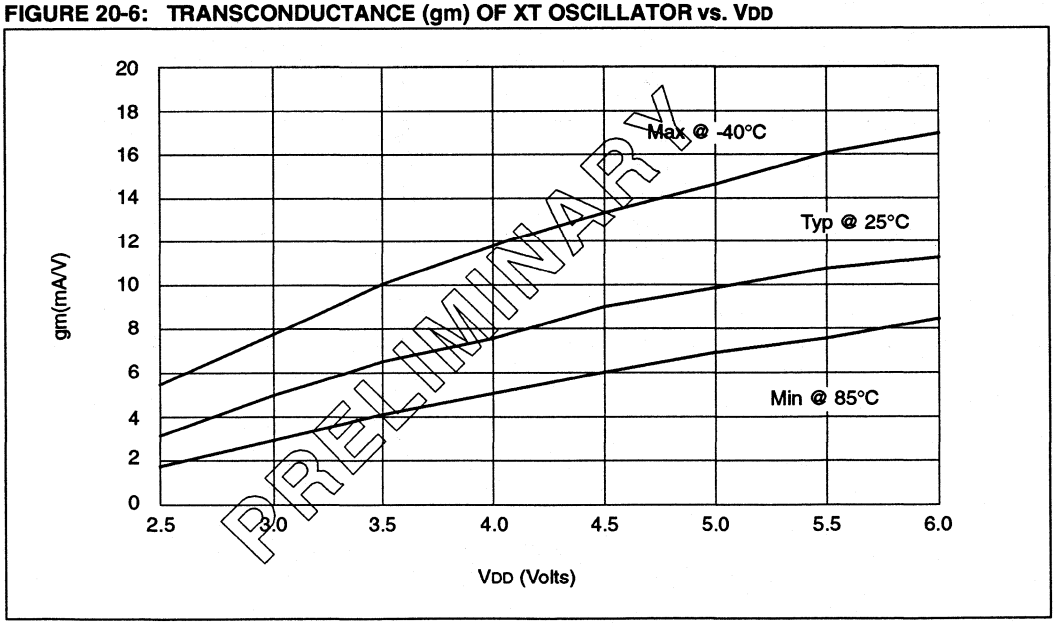


FIGURE 20-6: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR vs. VDD



© 1995 Microchip Technology Inc.

FIGURE 20-7: TYPICAL IDD vs. FREQUENCY (EXTERNAL CLOCK 25°C)

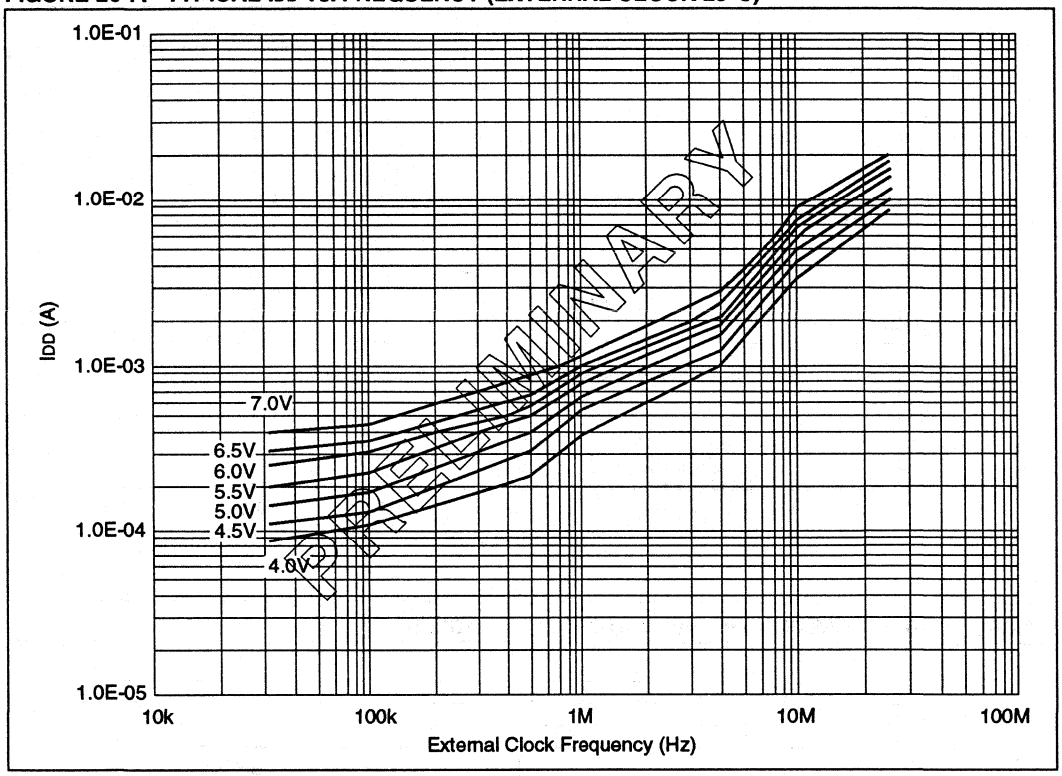


FIGURE 20-8: MAXIMUM IDD vs. FREQUENCY (EXTERNAL CLOCK 125°C TO -40°C)

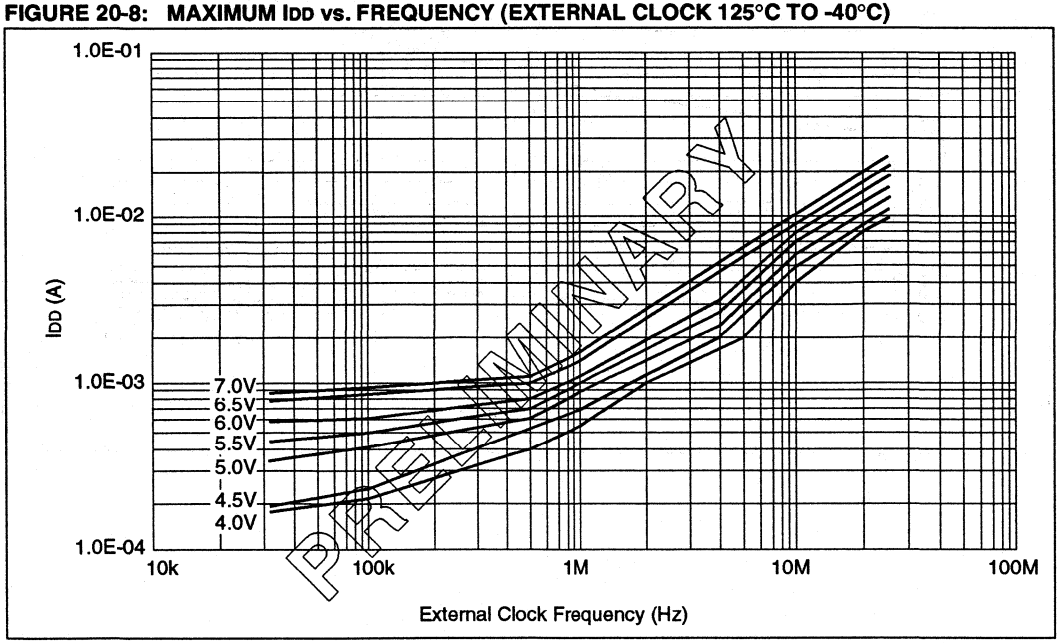


FIGURE 20-9: TYPICAL IPD vs. VDD WATCHDOG DISABLED 25°C

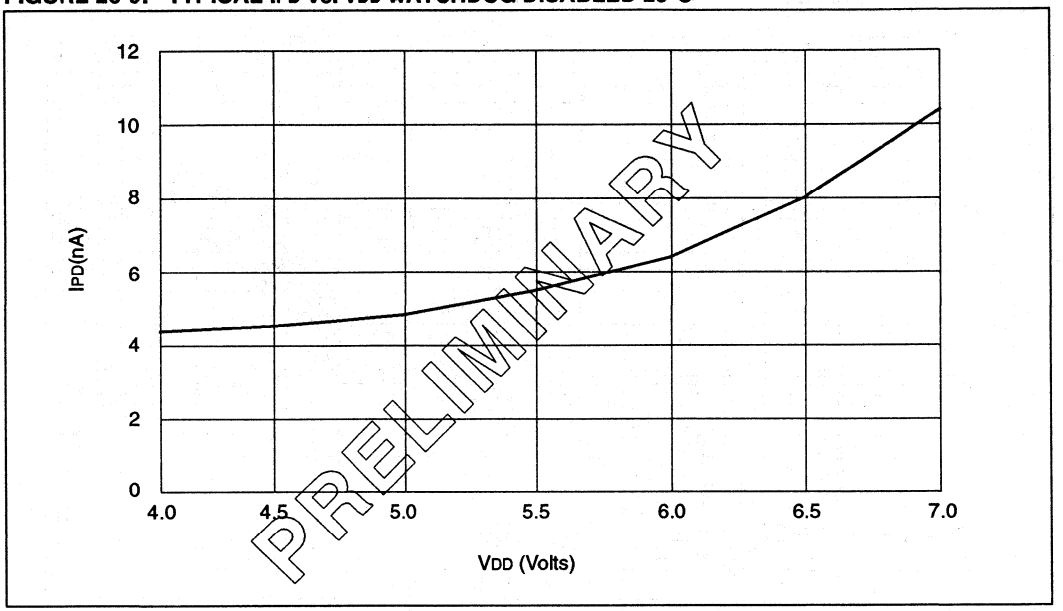
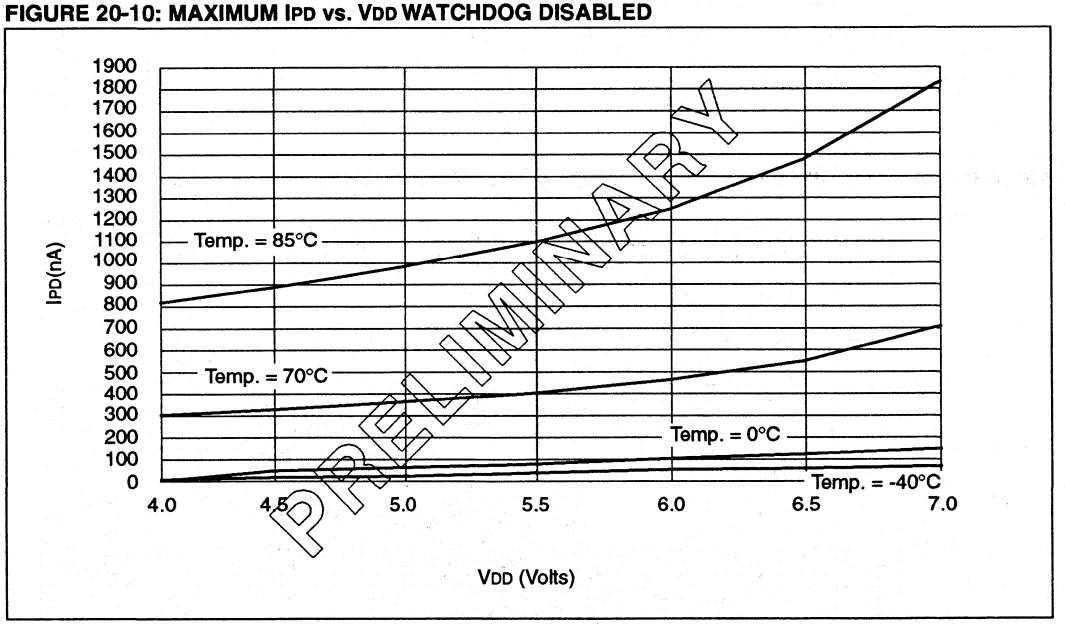


FIGURE 20-10: MAXIMUM IPD vs. VDD WATCHDOG DISABLED



© 1995 Microchip Technology Inc.

FIGURE 20-11: TYPICAL IPD vs. VDD WATCHDOG ENABLED 25°C

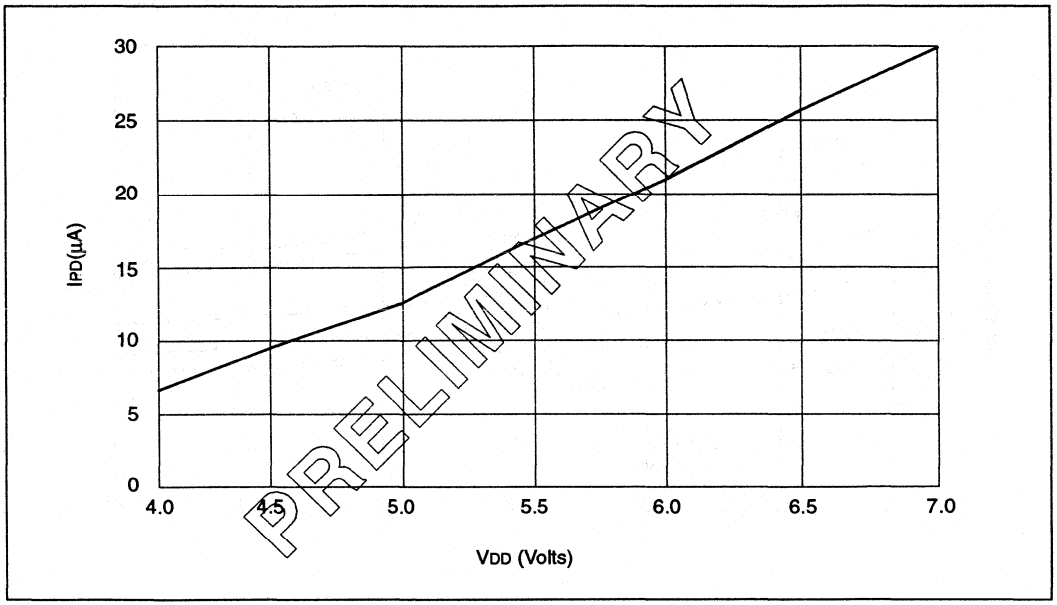


FIGURE 20-12: MAXIMUM IPD vs. VDD WATCHDOG ENABLED

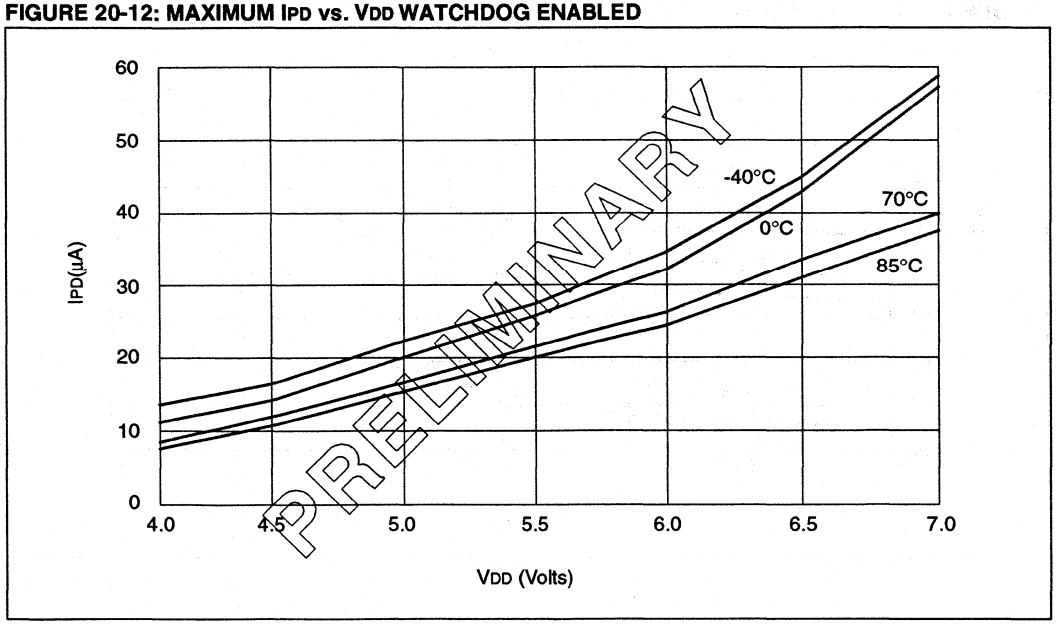


FIGURE 20-13: WDT TIMER TIME-OUT PERIOD vs. VDD

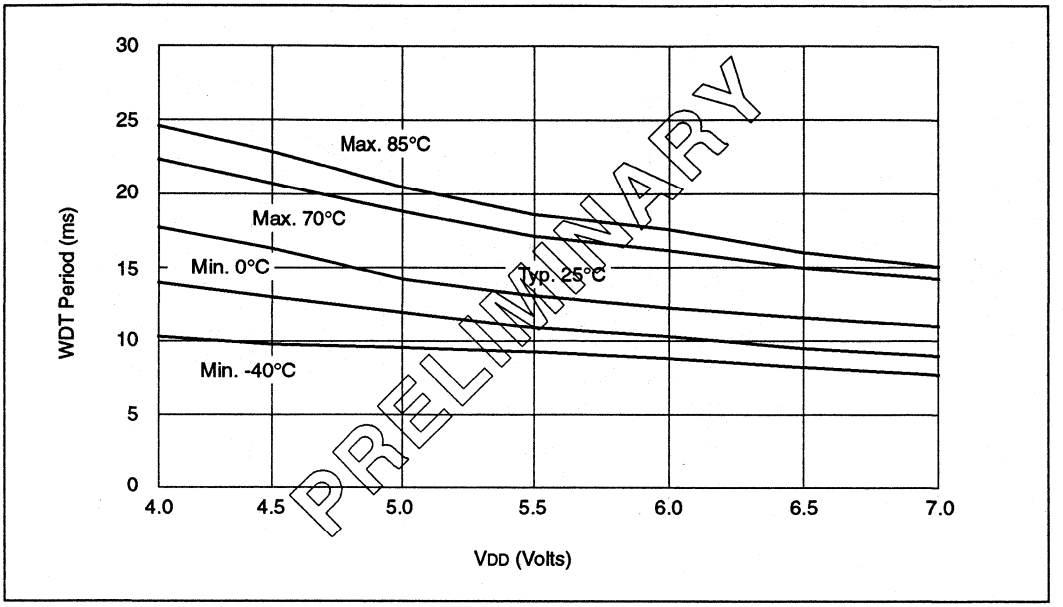
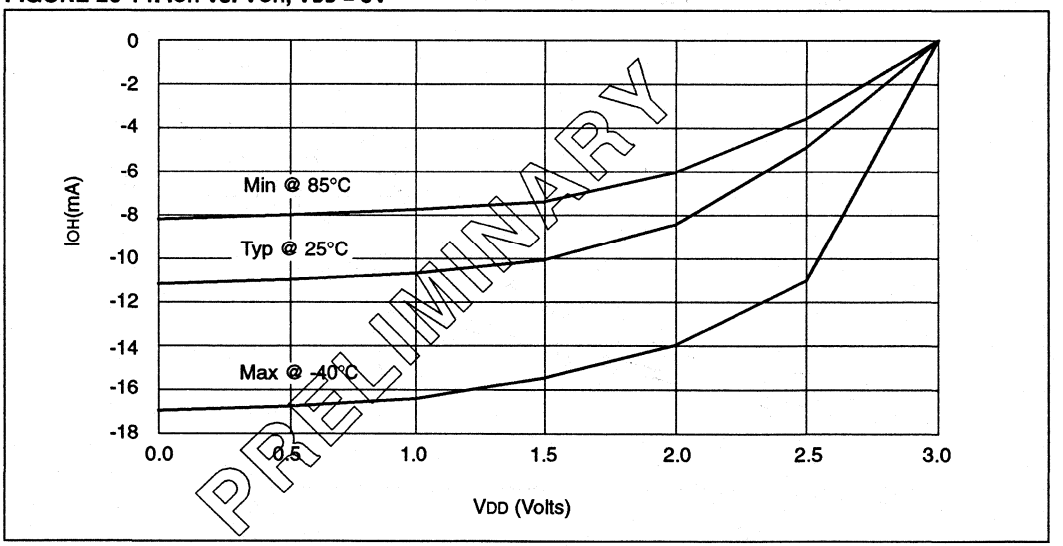


FIGURE 20-14: IOH vs. VOH, VDD = 3V



© 1995 Microchip Technology Inc.

FIGURE 20-15: IOH vs. VOH, VDD = 5V

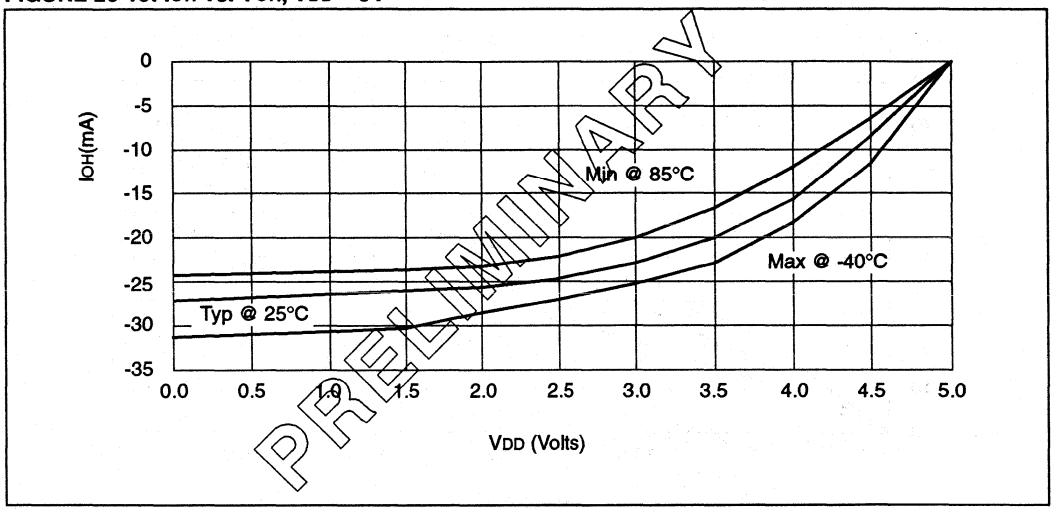


FIGURE 20-16: IOL vs. VOL, VDD = 3V

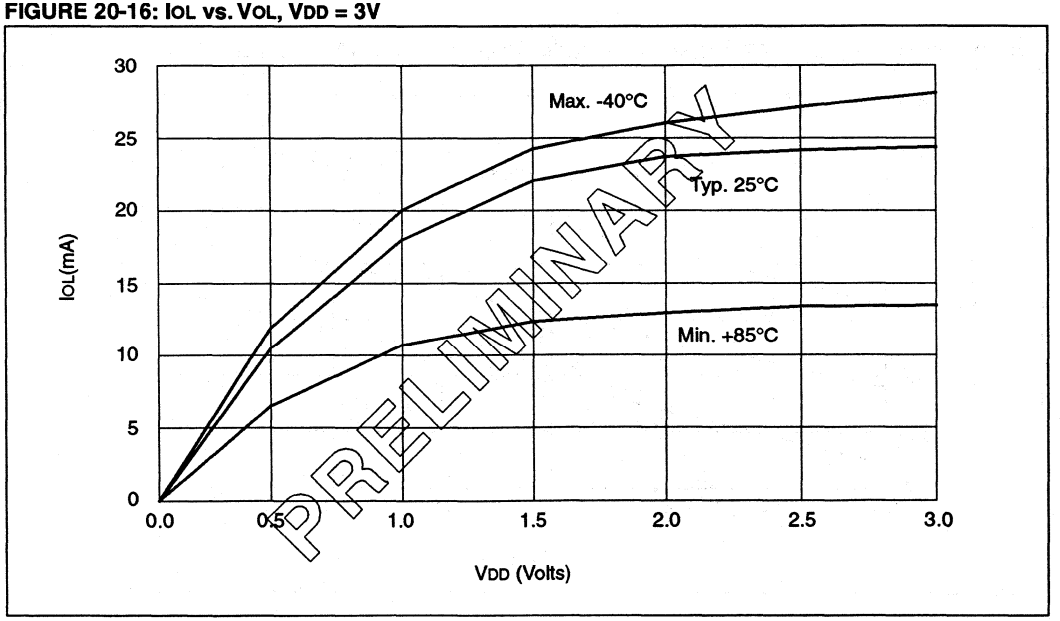


FIGURE 20-17: IOL vs. VOL, VDD = 5V

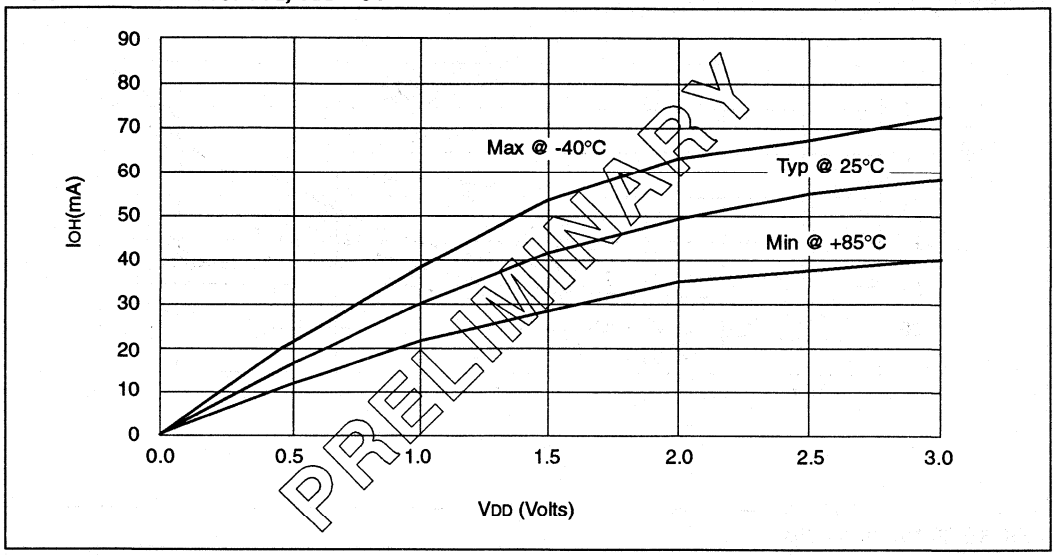
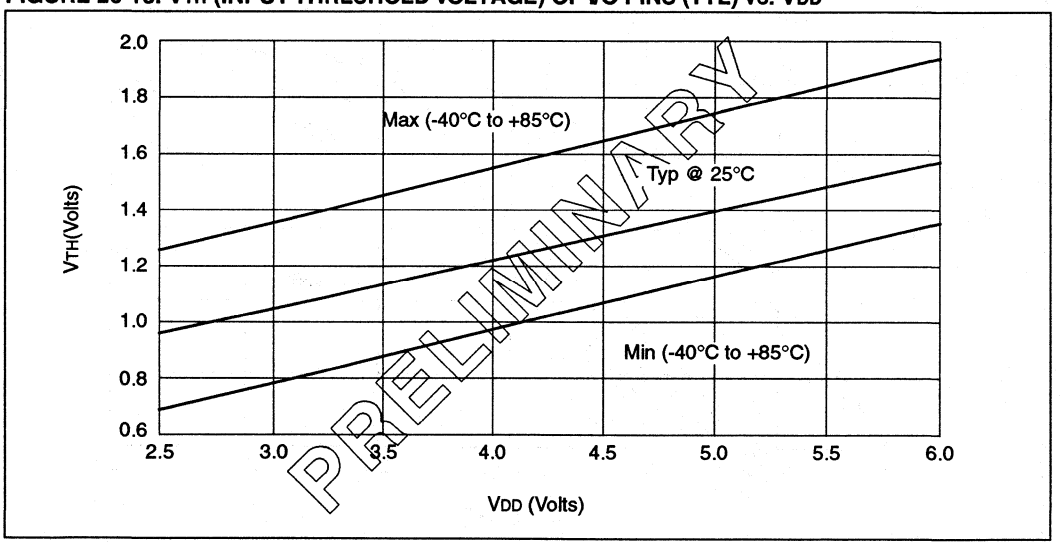
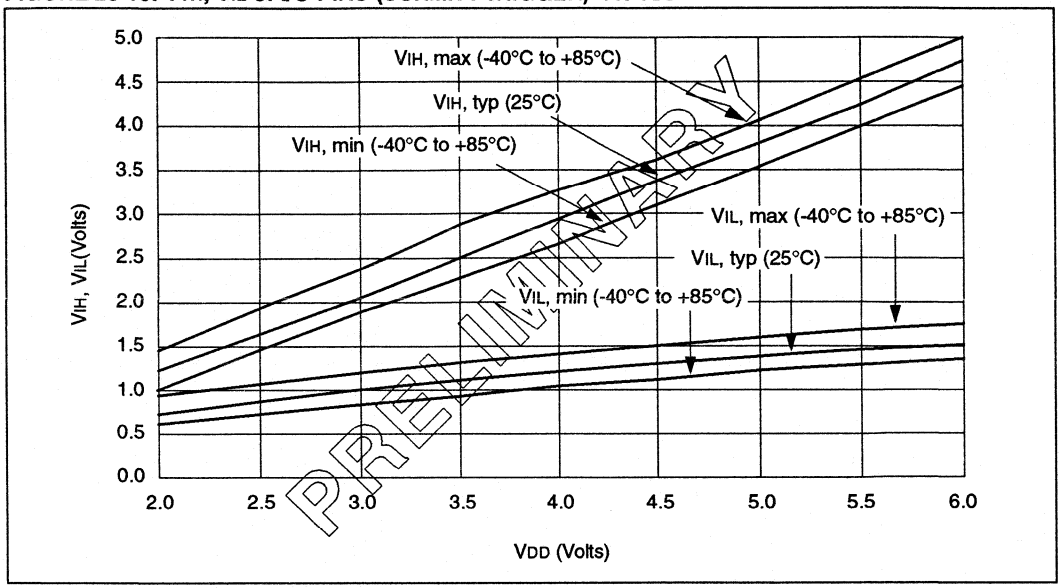


FIGURE 20-18: VTH (INPUT THRESHOLD VOLTAGE) OF I/O PINS (TTL) vs. VDD



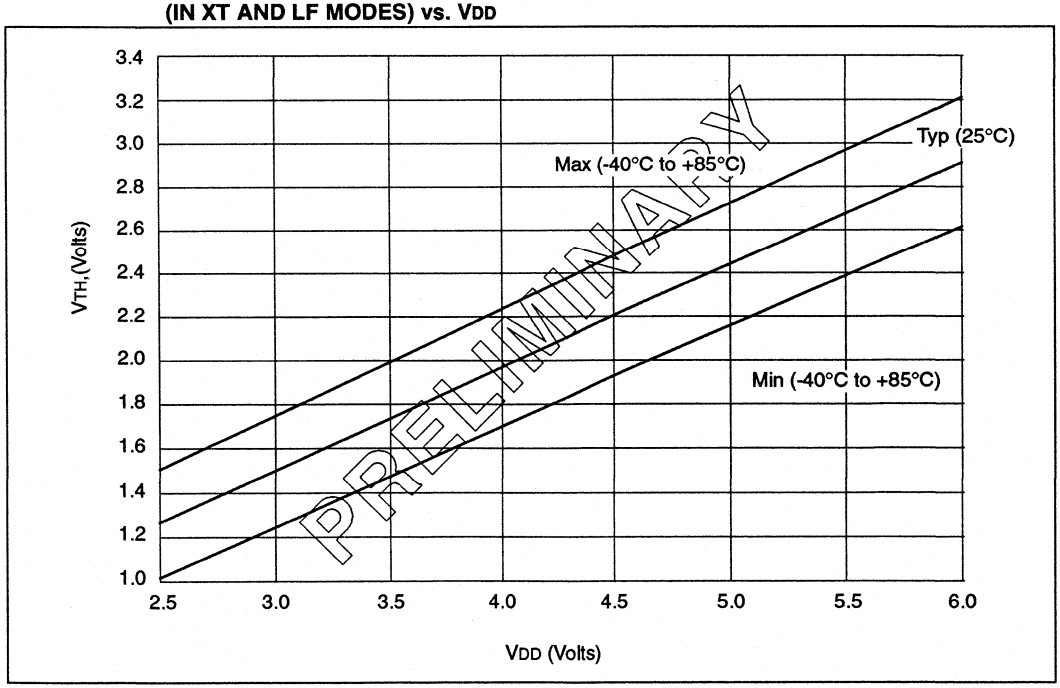
© 1995 Microchip Technology Inc.

FIGURE 20-19: VTH, VIL of I/O PINS (SCHMITT TRIGGER) vs. VDD



PRELIMINARY

FIGURE 20-20: VTH (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT AND LF MODES) vs. VDD



PRELIMINARY

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

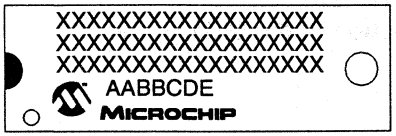
21.0 PACKAGING INFORMATION

For package dimensions,
please refer to the packaging section of the Data Book

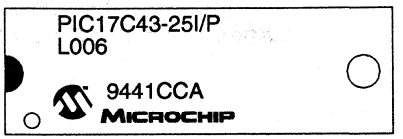
© 1995 Microchip Technology Inc.

21.1 Package Marking Information

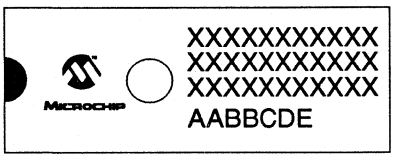
40-Lead PDIP



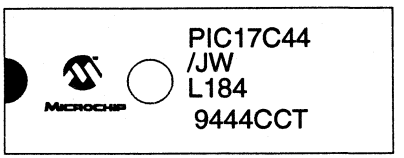
Example



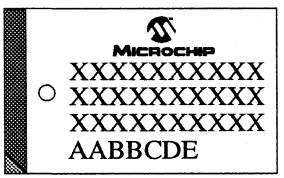
40 Lead CERDIP Windowed



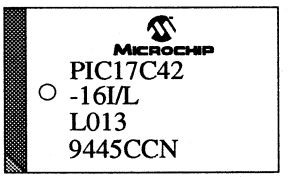
Example



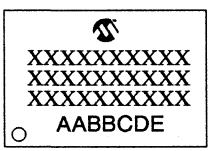
44-Lead PLCC



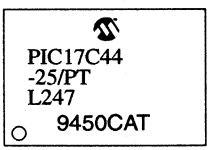
Example



44-Lead MQFP



Example



Legend:	MM...M	Microchip part number information
	XX...X	Customer specific information*
	AA	Year code (last 2 digits of calendar year)
	BB	Week code (week of January 1 is week '01')
	C	Facility code of the plant at which wafer is manufactured C = Chandler, Arizona, U.S.A., S = Tempe, Arizona, U.S.A.
	D	Mask revision number
	E	Assembly code of the plant or country of origin in which part was assembled
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.	

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

© 1995 Microchip Technology Inc.

APPENDIX A: :MODIFICATIONS

The following is the list of modifications over the PIC16CXX microcontroller family:

1. Instruction word length is increased to 16-bit. This allows larger page sizes both in program memory (8 Kwords verses 2 Kwords) and register file (256 bytes versus 128 bytes).
2. Four modes of operation: microcontroller, protected microcontroller, extended microcontroller, and microprocessor.
3. 22 new instructions.
   The `MOVF`, `TRIS` and `OPTION` instructions have been removed.
4. 4 new instructions for transferring data between data memory and program memory. This can be used to "self program" the EPROM program memory.
5. Single cycle data memory to data memory transfers possible (`MOVPF` and `MOVFP` instructions). These instructions do not affect the Working register (WREG).
6. W register (WREG) is now directly addressable.
7. A PC high latch register (PCLATH) is extended to 8-bits. The PCLATCH register is now both readable and writable.
8. Data memory paging is redefined slightly.
9. DDR registers replaces function of TRIS registers.
10. Multiple Interrupt vectors added. This can decrease the latency for servicing the interrupt.
11. Stack size is increased to 16 deep.
12. BSR register for data memory paging.
13. Wake up from SLEEP operates slightly differently.
14. The Oscillator Start-Up Timer (OST) and Power-Up Timer (PWRT) operate in parallel and not in series.
15. PORTB interrupt on change feature works on all eight port pins.
16. TMR0 is 16-bit plus 8-bit prescaler.
17. Second indirect addressing register added (FSR1 and FSR2). Configuration bits can select the FSR registers to auto-increment, auto-decrement, remain unchanged after an indirect address.
18. Hardware multiplier added (8 x 8 → 16-bit) (PIC17C43 and PIC17C44 only).
19. Peripheral modules operate slightly differently.
20. Oscillator modes slightly redefined.
21. Control/Status bits and registers have been placed in different registers and the control bit for globally enabling interrupts has inverse polarity.
22. Addition of a test mode pin.
23. In-circuit serial programming is not implemented.

APPENDIX B: COMPATIBILITY

To convert code written for PIC16CXX to PIC17CXX, the user should take the following steps:

1. Remove any `TRIS` and `OPTION` instructions, and implement the equivalent code.
2. Separate the interrupt service routine into its four vectors.
3. Replace:
   ```
   MOVF    REG1, W
   ```
   with:
   ```
   MOVFP   REG1, WREG
   ```
4. Replace:
   ```
   MOVF    REG1, W
   MOVWF   REG2
   ```
   with:
   ```
   MOVPF   REG1, REG2 ; Addr(REG1)<20h
   ```
   or
   ```
   MOVFP   REG1, REG2 ; Addr(REG2)<20h
   ```

   Note: If REG1 and REG2 are both at addresses greater then 20h, two instructions are required.
   ```
   MOVFP   REG1, WREG ;
   MOVPF   WREG, REG2 ;
   ```
5. Ensure that all bit names and register names are updated to new data memory map location.
6. Verify data memory banking.
7. Verify mode of operation for indirect addressing.
8. Verify peripheral routines for compatibility.
9. Weak pull-ups are enabled on reset.

To convert code from the PIC17C42 to the PIC17C43 or PIC17C44, the user should take the following steps.

1. If the hardware multiply is to be used, ensure that any variables at address 18h and 19h are moved to another address.
2. Ensure that the upper nibble of the BSR was not written with a non-zero value. This may cause unexpected operation since the RAM bank is no longer 0.
3. The disabling of global interrupts has been enhanced so there is no additional testing of the GLINTD bit after a `BCF CPUSTA, GLINTD` instruction.

© 1995 Microchip Technology Inc.

APPENDIX C: WHAT'S NEW

The conversion of this Data Sheet into the desktop publishing software package, The structure of the document has been made consistent with other data sheets. This ensures that important topics are covered across all PIC16/17 families. Here is an overview of new features:

- Data Sheet Structure / Outline
- Section on Table Reads and Table Writes
- Characterization results of the PIC17C42
- Hardware multiplier description
- New devices
- Three new instructions (PIC17C43 and PIC17C44 only)
- New electrical specification format

APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This allows control bits that have the same function, to have the same name (regardless of processor family). Care must still be taken, since they may not be at the same special function register address. The following shows the register and bit names that have been changed:

TABLE 21-1: REGISTER NAME CHANGES

OLD NAME	NEW NAME
W	WREG
RTCSTA	T0STA

TABLE 21-2: BIT NAME CHANGES

OLD NAME	NEW NAME
PEIR	PEIF
RTXIR	T0CKIF
T0IR	T0IF
IRB	RBIF
TM3IR	TMR3IF
TM2IR	TMR2IF
TM1IR	TMR1IF
CA2IR	CA2IF
CA1IR	CA1IF
TBMT	TXIF
RBFL	RCIF
INTIR	INTF
IEB	RBIE
RTXIE	T0CKIE
T/$\overline{C}$	T0CS
RTPS<3:0>	PS<3:0>
TMR1C	TMR1CS
TMR2C	TMR2CS
TMR3C	TMR3CS
16/$\overline{8}$	T16
$\overline{PUEB}$	$\overline{RBPU}$
RTEDG	T0SE
FPPM<1:0>	PM<1:0>
FWDT<1:0>	WDTPS<1:0>

- BSR register operation
- Instruction set descriptions have examples
- Timing specifications have been numbered

© 1995 Microchip Technology Inc.

APPENDIX E: PIC16/17 MICROCONTROLLERS

TABLE E-1: PIC17CXX FAMILY OF DEVICES

	Clock	Memory			Peripherals			Features			
	Maximum Frequency of Operation (MHz)	EPROM	RAM Data Memory (bytes)	Program Memory (bytes)	Timer Module(s)	Captures PWMs	Serial Port(s) (SCI)	External Interrupts	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC17C42	25	2K	232		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	4.5-5.5	55	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C43*	25	4K	454		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC17C44	25	8K	454		TMR0, TMR1, TMR2, TMR3	2 2	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP, 44-pin PLCC, 44-pin QFP

* Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC17C4X devices can also operate in microprocessor and external microcontroller modes.

3: PORTB has software-configurable weak pull-ups.

TABLE E-2: PIC16CXX FAMILY OF DEVICES

	Clock	Memory				Peripherals						Features				
	Maximum Frequency of Operation (MHz)	Program Memory EPROM	Program Memory EEPROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, SCI)	Parallel Slave Port	Analog to Digital Converter (8-bit)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out	Packages
PIC16C61	20	1K	—	36	—	TMR0	—	—	—	—	—	—	3	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C62*	20	2K	—	128	—	TMR0, TMR1, TMR2	2	SPI/I²C	—	—	—	—	10	22	2.5-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C63*	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	—	—	—	10	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C64	20	2K	—	128	—	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	—	—	—	8	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C65	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	—	—	—	11	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C620*	20	512	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C621*	20	1K	—	80	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C622	20	2K	--	128	—	TMR0	—	—	—	—	2	Yes	4	13	3.0-6.0	Yes	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C71	20	1K	—	36	—	TMR0	—	—	—	4 ch	—	—	4	13	3.0-6.0	—	18-pin DIP, 18-pin SOIC
PIC16C73	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	—	5 ch	—	—	11	22	3.0-6.0	—	28-pin SDIP, 28-pin SOIC
PIC16C74	20	4K	—	192	—	TMR0, TMR1, TMR2	2	SPI/I²C/ SCI	Yes	8 ch	—	—	12	33	3.0-6.0	—	40-pin DIP, 44-pin PLCC, 44-pin QFP
PIC16C84	10	—	1K	36	64	TMR0	—	—	—	—	—	—	4	13	2.0-6.0	—	18-pin DIP, 18-pin SOIC

* *Please contact your local sales office for availability of these devices.

Note 1: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

2: The PIC16CXX Timer1 has its own oscillator circuit and can operate asynchronously to the device. Timer1 can increment while the device is in SLEEP mode. This allows a Real Time Clock to be implemented.

3: PORTB has software-configurable weak pull-ups.

© 1995 Microchip Technology Inc.

TABLE E-3: PIC16C5X FAMILY OF DEVICES

	Clock	Memory				Peripherals	Features		
	Maximum Frequency of Operation (MHz)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C54A	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR54	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16C55	20	512	—	25	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20-pin SSOP
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16CR57A	20	—	2K	72	TMR0	20	2.0-6.25	33	28-pin DIP, 28-pin SOIC, 28 pin SSOP
PIC16C58A	20	2K	—	73	TMR0	12	2.5-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP
PIC16CR58A	20	—	2K	73	TMR0	12	2.0-6.25	33	18-pin DIP, 18-pin SOIC, 20 pin SSOP

Note: All PIC16/17 Family devices have Power-On Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.

© 1995 Microchip Technology Inc.

E.1 Pin Compatibility

Devices that have the same package type; and VDD, VSS, and $\overline{MCLR}$ pin locations, are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-4: PIN COMPATIBILE DEVICES

Pin Compatible Devices	Package
PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C71, PIC16C84, PIC16C54, PIC16C54A, PIC16CR54, PIC16C56, PIC16C58A, PIC16CR58A	18 pin
PIC16C62, PIC16C63, PIC16C73	28 pin
PIC16C55, PIC16C57, PIC16CR57A	28 pin
PIC17C42, PIC17C43, PIC17C44	40 pin
PIC16C64, PIC16C65, PIC16C74	40 pin

© 1995 Microchip Technology Inc.

APPENDIX F: ERRATA FOR PIC17C42 SILICON

The PIC17C42 devices that you have received have the following anomalies. At present there is no intention for future revisions to the present PIC17C42 silicon. If these cause issues for the application, it is recommended that you select the PIC17C43 device.

1. When the Oscillator Start-Up Timer (OST) is enabled (in LF or XT oscillator modes), any interrupt that wakes the processor may cause a WDT reset. This occurs when the WDT is greater than or equal to 50% time-out period when the SLEEP instruction is executed. This will not occur in either the EC or RC oscillator modes.

   Work-arounds

a) Always ensure that the CLRWDT instruction is executed before the WDT increments past 50% of the WDT period. This will keep the "false" WDT reset from occurring.
b) When using the WDT as a normal timer (WDT disabled), ensure that the WDT is less than or equal to 50% time-out period when the SLEEP instruction is executed. This can be done by monitoring the $\overline{TO}$ bit for changing state from set to clear. Example 1 shows putting the PIC17C42 to sleep.

EXAMPLE F-1: PIC17C42 TO SLEEP

```
      BTFSS   CPUSTA, TO  ; TO = 0?
      CLRWDT              ; YES, WDT = 0
LOOP  BTFSC   CPUSTA, TO  ; WDT rollover?
      GOTO    LOOP        ; NO, Wait
      SLEEP               ; YES, goto Sleep
```

2. When the clock source of Timer1 or Timer2 is selected to external clock, the overflow interrupt flag will be set twice, once when the timer equals the period, and again when the timer value is reset to 0h. If the latency to clear TMRxIF is greater than the time to the next clock pulse, no problems will be noticed. If the latency is less than the time to the next timer clock pulse, the interrupt will be serviced twice.

   Work-arounds

a) Ensure that the timer has rolled over to 0h before clearing the flag bit.
b) Clear the timer in software. Clearing the timer in software causes the period to be one count less than expected.

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

INDEX

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

2

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF EXAMPLES]

LIST OF FIGURES

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

LIST OF TABLES

LIST OF EQUATIONS

2

CONNECTING TO MICROCHIP BBS

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allows multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations:

1. Set your modem to 8 bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<ENTER↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type **+,** depress **<ENTER↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS,** depress **< ENTER↵ >** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

NETWORK, depress **< ENTER↵ >**
and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

Trademarks:

PICMASTER and PICSTART are registered trademarks of Microchip Technology Incorporated. PIC is a registered trademark of Microchip Technology Incorporated in the U.S.A.

PRO MATE, *fuzzy*LAB, the Microchip logo and name are trademarks of Microchip Technology Incorporated.

*fuzzy*TECH is a registered trademark of Inform Software Corporation.

I^2C is a trademark of Philips Corporation.

IBM, IBM PC-AT are registered trademarks of International Business Machines Corp.

Pentium is a trademark of Intel Corporation.

MS-DOS and Microsoft Windows are registered trademarks of Microsoft Corporation. Windows is a trademark of Microsoft Corporation.

CompuServe is a registered trademark of CompuServe Incorporated.

All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC17C4X Literature Number: DS30412A

Questions:

1. What are the best features of this document? _____

2. How does this document meet your hardware and software development needs? _____

3. Do you find the organization of this data sheet easy to follow? If not, why? _____

4. What additions to the data sheet do you think would enhance the structure and subject? _____

5. What deletions from the data sheet could be made without affecting the overall usefullness? _____

6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

8. How would you improve our software, systems, and silicon products? _____

© 1995 Microchip Technology Inc.

PIC17C4X Product Identification System

To order or obtain information, e.g., on pricing or delivery refer to the factory or the listed sales office.

PART NO. -XX X /XX XXX

Pattern:		QTP, SQTP, ROM Code (factory specified) or Special Requirements. Blank for OTP and Windowed devices
Package:	JW	= Windowed CERDIP
	P	= PDIP (600 mil)
	PQ	= MQFP (Metric PQFP PIC17C42 only)
	PT	= TQFP
	L	= PLCC
Temperature Range:	-	= 0°C to +70°C
	I	= -40°C to +85°C
Frequency Range:	08	= 8 MHz
	16	= 16 MHz
	25	= 25 MHz
Device	PIC17C44	:Standard VDD range
	PIC17C44T	:(Tape and Reel)
	PIC17LC44	:Extended VDD range
	PIC17LC44T	:(Tape and Reel)

Examples

a) PIC17C42 - 16/P
Commercial Temp.,
PDIP Package,
16 MHz,
normal VDD limits

b) PIC17LC44 - 08/PT
Commercial Temp.,
TQFP package,
8 MHz,
extended VDD limits

c) PIC17C43 - 25I/P
Industrial Temp.,
PDIP package,
25 MHz,
normal VDD limits

Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office (see below)
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.
For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

*Information contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation of warranty is given and no liability is assumed by Microchip Technology Inc. with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. The Microchip logo and name are trademarks of Microchip Technology Incorporated. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.

© 1995 Microchip Technology Inc.

MICROCHIP

SECTION 3
PIC16/17 MICROCONTROLLER PROGRAMMING SPECIFICATIONS

3

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

MICROCHIP

PIC16C5X

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C54
- PIC16C54A
- PIC16C55
- PIC16C56
- PIC16C57
- PIC16C58A
- PIC16CR54
- PIC16CR57A
- PIC16CR58A

INTRODUCTION

Overview

The PIC16C5X Series is a family of single-chip CMOS microcontrollers with on-chip EPROM for program storage. The programming specification also applies to ROM products for verification only.

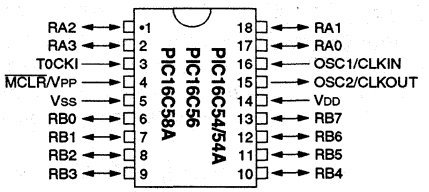
Due to the special architecture of these microcontrollers (12-bit wide instruction word) and the low pin counts (starting at 18 pins), the EPROM programming methodology is different from that of standard (byte-wide) EPROMs (e.g., 27C256).

The PIC16C5X Series can be programmed by applying the 12-bit wide data word to the 12 available I/O pins while the address is generated by the on-chip Program Counter. The $\overline{MCLR}$/VPP pin provides the programming supply voltage (VPP). Programming/verify chip enable is controlled by the T0CKI pin while the OSC1 pin controls the Program Counter.

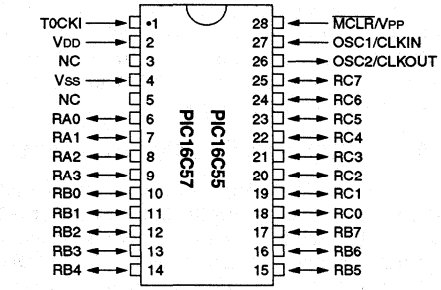
This document describes all the programming details of the PIC16C5X Series and the requirements for programming equipment to be used from programming prototypes in the engineering lab up to high volume programming on the factory floor.

PIN CONFIGURATIONS

PDIP, SOIC, CERDIP Window



PDIP, SOIC, CERDIP Window



PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C54/54A/56/58a, PIC16C55/57

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
T0CKI	$\overline{PROG}$/VER	I	Program pulse input/verify pulse input
RA0 - RA3	D0 - D3	I/O	Data input/output
RB0 - RB7	D4 - D11	I/O	Data input/output
OSC1	INCPC	I	Increment Program Counter input
$\overline{MCLR}$/VPP	VPP	P	Programming Power
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = input, O = Output, P = Power

© 1995 Microchip Technology Inc.

1.0 PROGRAM/VERIFY MODES

The PIC16C5X Series uses the internal Program Counter (PC) to generate the EPROM address. VPP is supplied through the $\overline{MCLR}$ pin.

The T0CKI pin acts as chip enable, alternating between programming and verifying.

The OSC1 pin is used for incrementing the PC.

Data is applied to, or can be read on PORTA and PORTB (MSB on RB7, LSB on RA0).

The programming/verify mode is entered by raising the level on the $\overline{MCLR}$ pin from VIL to VHH (= VPP) while the T0CKI pin is held at VIH and the OSC1 pin is held at VIL.

The Program Counter now has the value "0xFFF", because $\overline{MCLR}$ was at VIL before. This condition selects the configuration word as the very first EPROM location to be accessed after entering the program/verify mode.

Since the $\overline{MCLR}$ pin was initially at VIL, the device is in the reset state (the I/O pins are in the reset state).

Incrementing the PC once (by pulsing the OSC1 pin) selects location "0x000" of the user program memory. Afterwards all other memory locations from 001h through end of memory can be addressed by incrementing the PC .

If the Program Counter has reached the last address of the user memory area (e.g. "0x1FF" for the PIC16C54/55), and is incremented again, the on-chip special EPROM area will be addressed. (See Figure 1-2 to determine where the special EPROM area is located for the various PIC16C5X devices).

1.1 Program/Verify without PC Increment

After entering the program/verify mode, pulsing the T0CKI pin LOW programs the data present on PORTA and PORTB into the memory location selected by the Program Counter. The duration of the T0CKI LOW time determines the length of the programming pulse.

Pulsing the T0CKI pin LOW again without changing the signals on $\overline{MCLR}$ and OSC1 puts the contents of the selected memory location out on PORTA and PORTB for verification of a successful programming cycle. This verification pulse on T0CKI can be much shorter than the programming pulse. If the programming was not successful, T0CKI can be pulsed LOW again to apply another programming pulse, followed again by a shorter T0CKI LOW pulse for another verification cycle.

This sequence can be repeated as many times as required until the programming is successful.

1.2 Verify with PC Increment

If a verification cycle shows that programming was successful, the Program Counter can be incremented by keeping the T0CKI input at a HIGH level while pulsing the OSC1 input HIGH. When both T0CKI and OSC1 are HIGH, the contents of the selected memory location is put out on Ports A and B (= Verify). The falling edge of OSC1 will increment the Program Counter.

A fast VERIFY- ONLY with automatic increment of the PC can be performed by entering the program/verify mode as described above and then clocking the OSC1 input. If OSC1 is HIGH, the selected memory location is output on Ports A and B, while the falling edge of OSC1 will increment the Program Counter. Thus, the first memory location to be verified after entering the program/verify mode, is the configuration word. The next location is 000h followed by 001h and so on. The program memory location "N" can be reached by generating "N + 1" falling edges on OSC1. When OSC1 is brought HIGH again, the contents of address "N" are output on Ports A and B as long as OSC1 stays HIGH.

1.3 Programming/Verifying Configuration Word

The configuration word is logically mapped at program memory location "0xFFF". The PC points to the configuration word after $\overline{MCLR}$ pin goes from LOW to VHH (HIGH). The configuration word can be programmed or verified using the techniques described in Section 1.1 and Section 1.2.

If PC is incremented, the next location it will point to is "0x000" in user memory. Incrementing PC 4096 times will not allow the user to point to the configuration word. The only way to point to it again is to reset and re-enter program mode.

© 1995 Microchip Technology Inc.

1.4 Programming Method

The programming technique is described in the following section. It is designed to guarantee good programming margins. It does, however, require a variable power supply for VCC.

1.4.1 PROGRAMMING METHOD DETAILS

Essentially, this technique includes the following steps:

1. Perform blank check at VDD = VDD min. Report failure. The device may not be properly erased.
2. Program location with pulses (100µs typically) and verify after each pulse at VDD = VDDP: where VDDP = VDD range required during programming (4.5V - 5.5V).

a) Programming condition:

VPP = 13.0V to 13.25V

VDD = VDDP = 4.5V to 5.5V

VPP must be ≥ VDD + 7.25V to keep "programming mode" active.

b) Verify condition:

VDD = VDDP

VPP ≥ VDD + 7.5V but not to exceed 13.25V

If location fails to program after "N" pulses, (suggested maximum program pulses of 25) then report error as a programming failure.

Note: Device must be verified at minimum and maximum specified operating voltages as specified in the data sheet.

3. Once location passes "Step 2", apply 3X over-programming, i.e., apply three times the number of pulses that were required to program the location. This will guarantee a solid programming margin. The overprogramming should be made "software programmable" for easy updates.
4. Program all locations.
5. Verify all locations (using speed verify mode) at VDD = VDD min.
6. Verify all locations at VDD = VDD max.

VDD min. is the minimum operating voltage spec. for the part. VDD max. is the maximum operating voltage spec. for the part.

1.4.2 SYSTEM REQUIREMENTS

Clearly, to implement this technique, the most stringent requirements will be that of the power supplies:

VPP: VPP can be a fixed 13.0V to 13.25V supply. It must not exceed 14.0V to avoid damage to the pin and should be current limited to approximately 100mA.

VDD: 2.0V to 6.5V with 0.25V granularity. Since this method calls for verification at different VDD values, a programmable VDD power supply is needed.

Current Requirement: 40mA maximum

Microchip may release PIC16C5Xs in the future with different VDD ranges which make it necessary to have a programmable VDD.

It is important to verify an EPROM at the voltages specified in this method to remain consistent with Microchip's test screening. For example, a PIC16C5X specified for 4.5V to 5.5V should be tested for proper programming from 4.5V to 5.5V.

Note: Any programmer not meeting the programmable VDD requirement and the verify at VDD max. and VDD min. requirement may only be classified as "prototype" or "development" programmer but not a production programmer.

1.4.3 SOFTWARE REQUIREMENTS

Certain parameters should be programmable (and therefore easily modified) for easy upgrade.

a) Pulse width, current value 100µs.
b) Maximum number of pulses, current limit 25.
c) Number of over-programming pulses: should be = (A • N) + B, where N = number of pulses required in regular programming. In our current algorithm A = 3, B = 0.

1.5 Programming Pulse Width

Normal EPROM Cells: When programming one word of EPROM, a programming pulse width (TPW) of 100µs is recommended.

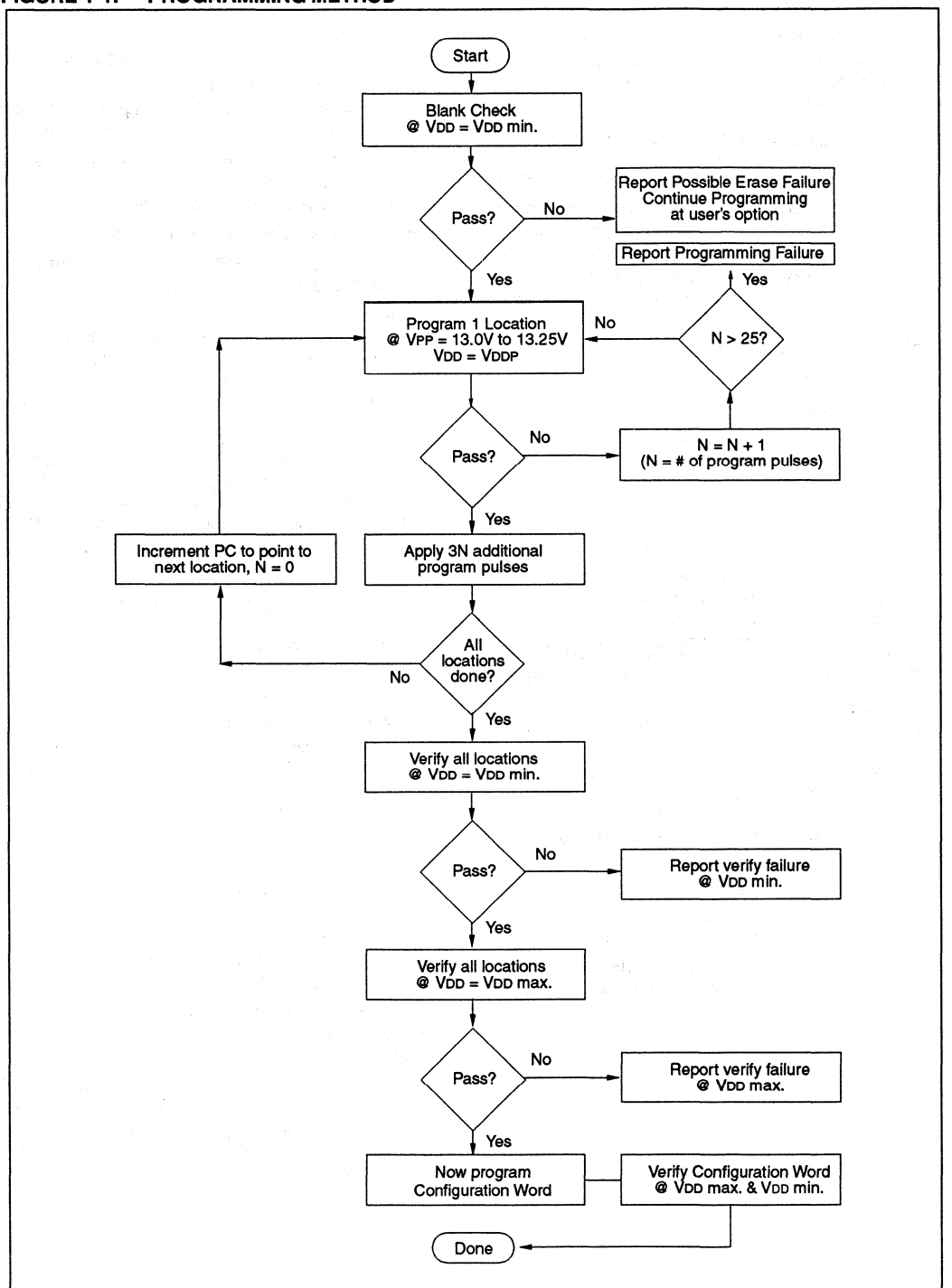
The maximum number of programming attempts should be limited to 25 per word.

After the first successful verify, the same location should be over-programmed with 3X over-programming.

Configuration Word: The configuration word for oscillator selection, WDT (watchdog timer) disable and code protection, requires a programming pulse width (TPWF) of 10ms. A series of 100µs pulses is preferred over a single 10ms pulse.

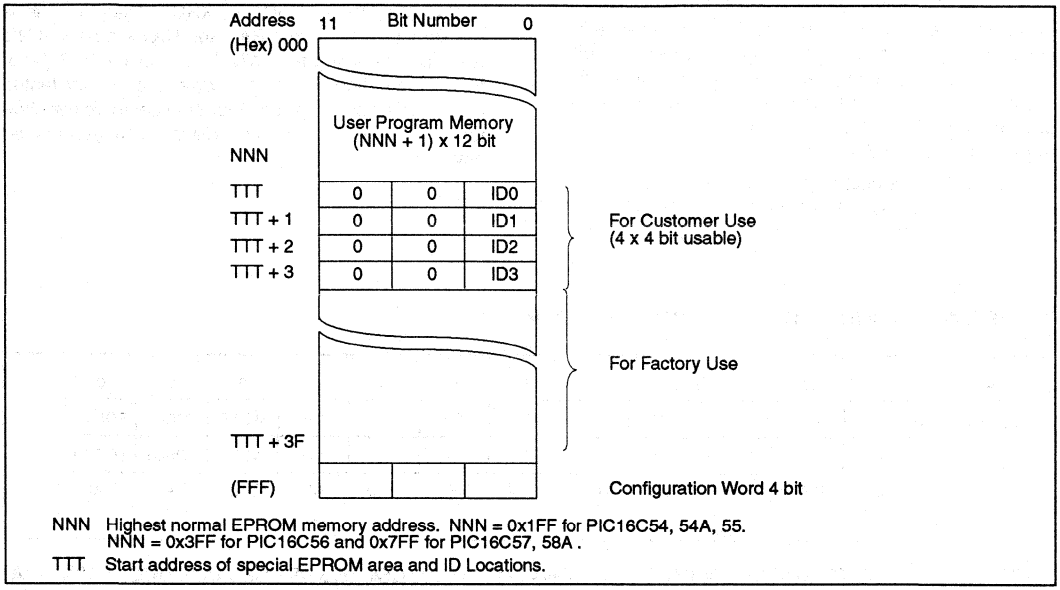
© 1995 Microchip Technology Inc.

FIGURE 1-1: PROGRAMMING METHOD



© 1995 Microchip Technology Inc.

FIGURE 1-2: PIC16C5X SERIES PROGRAM MEMORY MAP IN PROGRAM/VERIFY MODE



1.6 Special Memory Locations

The ID Locations area is only enabled if the device is in a test or programming/verify mode. Thus, in normal operation mode only the memory location 0x000 to 0xNNN will be accessed and the Program Counter will just roll over from address 0xNNN to 0x000 when incremented.

The configuration word can only be accessed immediately after $\overline{MCLR}$ going from VIL to VHH. The Program Counter will be set to all '1's upon $\overline{MCLR}$ = VIL. Thus, it has the value "0xFFF" when accessing the configuration EPROM. Incrementing the Program Counter once by pulsing OSC1 causes the Program Counter to roll over to all '0's. Incrementing the Program Counter 4K times after reset ($\overline{MCLR}$ = VIL) does not allow access to the configuration EPROM.

1.6.1 CUSTOMER ID CODE LOCATIONS

Per definition, the first four words (address TTT to TTT + 3) are reserved for customer use. It is recommended that the customer use only the four lower order bits (bits 0 through 3) of each word and filling the eight higher order bits with '0's.

A user may want to store an identification code (ID) in the ID locations and still be able to read this code after the code protection bit was programmed. This is possible if the ID code is only four bits long per memory location, is located in the least significant nibble boundary of the 12-bit word, and the remaining eight bits are all '0's.

EXAMPLE 1-1: CUSTOMER CODE 0xD1E2

The Customer ID code "0xD1E2" should be stored in the ID locations 200-203 like this:

```
200:  0000 0000 1101
201:  0000 0000 0001
202:  0000 0000 1110
203:  0000 0000 0010
```

Reading these four memory locations, even with the code protection bit programmed would still output on Port A the bit sequence "1101", "0001", "1110", "0010" which is "0xD1E2".

Note: Microchip will assign a unique pattern number for QTP and SQTP requests and for ROM devices. This pattern number will be unique and traceable to the submitted code.

2.0 CONFIGURATION WORD

The configuration word is the very first memory location which is accessed after entering the program/verify mode of the PIC16C5X. It contains the two bits for the selection of the oscillator type, the watchdog timer enable bit, and the code protection bit. All other bits (4 through 11) are read as '1's.

One-Time-Programmable (OTP) devices may have the oscillator configuration bits "FOSC0" and "FOSC1" set by the factory and are tested accordingly. The packages are marked "PIC16C5XHS", "PIC16C5XXT", "PIC16C5XLP", or "PIC16C5XRC". Therefore, it is essential that the inputs RA0 and RA1 are held at '1's when programming the "WDTE" and/or the "CP" bit of the configuration word. Otherwise, the factory tested and selected oscillator configuration could be overwritten and the functionality of the device is not guaranteed any more.

FIGURE 2-1: CONFIGURATION WORD BIT MAP

Bit Number:	11	10	9	8	7	6	5	4	3	2	1	0
PIC16CR58A	CP	CP	CP	CP	CP	CP	CP	CP	CP	WDTE	FOSC1	FOSC0
Other PIC16C5X	—	—	—	—	—	—	—	—	CP	WDTE	FOSC1	FOSC0
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	RA3	RA2	RA1	RA0

TABLE 2-1: CONFIGURATION BIT FUNCTIONALITY (PIC16C54/55/56/57/C54A/C58A/CR54/CR57A)

RA3-RA11 CP	RA2 WDTE	RA1 FOSC1	RA0 FOSC2	Function	Remarks
1	x	x	x	Memory Unprotected	Default
0	x	x	x	Memory protected	
x	1	x	x	Watchdog Timer enabled	Default
x	0	x	x	Watchdog Timer disabled	
x	x	1	1	RC Oscillator	Default
x	x	1	0	HS - High Speed Crystal	
x	x	0	1	XT - Standard Crystal	
x	x	0	0	LP - Low Frequency Crystal	

Legend: 1= Erased (apply HIGH Level to I/O pin during program)
0 = Written (apply LOW Level to I/O pin during program).
x = Don't Care

© 1995 Microchip Technology Inc.

TABLE 2-2: CONFIGURATION WORD

PIC16C54 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C54A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR54 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled	Read Unscrambled
ID Words [0x200 : 0x203]	Read Scrambled	Read Unscrambled
[0x040 : 0x1FF]	Read Scrambled	Read Unscrambled
[0x000 : 0x03F]	Read Scrambled	Read Unscrambled

PIC16C55 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x200 : 0x203]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x1FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C56 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x400 : 0x403]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x3FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C57 (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x800 : 0x803]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x7FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR57A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled	Read Unscrambled
ID Words [0x800 : 0x803]	Read Scrambled	Read Unscrambled
[0x040 : 0x7FF]	Read Scrambled	Read Unscrambled
[0x000 : 0x03F]	Read Scrambled	Read Unscrambled

© 1995 Microchip Technology Inc.

PIC16C58A (CP enable pattern: XXXXXXXX0XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
ID Words [0x800 : 0x803]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
[0x040 : 0x7FF]	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
[0x000 : 0x03F]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16CR58A (CP enable pattern: 000000000XXX)

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFFF)	Read Unscrambled	Read Unscrambled
ID Words [0x800 : 0x803]	Read Unscrambled	Read Unscrambled
[0x040 : 0x7FF]	Read Disabled	Read Unscrambled
[0x000 : 0x03F]	Read Unscrambled	Read Unscrambled

Legend: X = Don't care

2.1 Code Protection

The program code written into the EPROM can be protected by writing to the "CP" bit of the configuration word. All memory locations starting at 0x40 and above are protected against programming. It is still possible to program locations 0x00 through 0x3F, the ID locations, and the configuration word.

Note: Locations [0x000 : 0x03F] are not secure after code protection.

2.1.1 PROGRAMMING LOCATIONS 0x000 TO 0x03F AFTER CODE PROTECTION

In a code protected part, these locations will program with the exception of the PIC16CRXX devices. They will read back scrambled data, with the exception of PIC16CR54A and PIC16CR58A. In any event, the programmer cannot verify the device once it is code protected.

In code protected parts, specifically PIC16C54/ C54A/ CR54/C55/C56/C57 devices, the contents of the program memory cannot be read out in a way that the program code can be reconstructed. A location when read out will read as: 0000 0000 xxxx where xxxx is the XOR of the three nibbles.

For example, if the memory location contains 0xC04 (movlw 4), after code protection the output will be 0x008.

In addition, all memory locations starting at 0x40 and above are protected against programming. It is still possible to program locations 0x000 through 0x03F and the configuration word. However, performing a verify with activated code protection logic puts a 4-bit wide "checksum" on PORTA while the 8-bits of PORTB are read as '0's. The checksum is computed as follows:

The four high order bits of an instruction word are "XOR'ed" with the four middle and the four low order bits, and the result is transferred to PORTA. All memory locations are affected.

To program location 0x000 to 0x03F in a code protected part, the programmer should program one nibble at a time and verify the result through the XOR'ed output. For example, to program a location with 0xA93, first program the location with 0xFF3, verify checksum to be 0x003; then program the location with 0xF93 and verify the XOR'ed output to be 0x00C and finally program the location with 0xA93 and verify the read-out to be 0x006.

2.2 CHECKSUM COMPUTATION

2.2.1 CHECKSUM

Checksum is calculated by reading the contents of the PIC16C5X memory locations and adding up the opcodes up to the maximum user addressable location, e.g., 0x1FF for the PIC16C54/55. Any carry bits exceeding 16-bits are neglected. Finally, the configuration word (appropriately masked) is added to the checksum. Checksum computation for each member of the PIC16C5X devices is shown in Table 2-3.

© 1995 Microchip Technology Inc.

TABLE 2-3: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	0x723 at 0 and max address
PIC16C54	OFF	SUM[0x000:0x1FF] + CFGW & 0xFFF	0x0DFF	0xFC47
	ON	SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x1E07	0x1DF5
PIC16C54A	OFF	SUM[0x000:0x1FF] + CFGW & 0xFFF	0x0DFF	0xFC47
	ON	SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x1E07	0x1DF5
PIC16CR54	OFF	SUM[0x000:0x1FF] + CFGW & 0xFFF	0x0DFF	—
	ON	SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x1E07	—
PIC16C55	OFF	SUM[0x000:0x1FF] + CFGW & 0xFFF	0x0DFF	0xFC47
	ON	SUM_XOR4[0x000:0x1FF] + CFGW & 0x00F	0x1E07	0x1DF5
PIC16C56	OFF	SUM[0x000:0x3FF] + CFGW & 0xFFF	0x0BFF	0xFA47
	ON	SUM_XOR4[0x000:0x3FF] + CFGW & 0x00F	0x3C07	0x3BF5
PIC16C57	OFF	SUM[0x000:0x7FF] + CFGW & 0xFFF	0x07FF	0xF647
	ON	SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x7807	0x77F5
PIC16CR57A	OFF	SUM[0x000:0x7FF] + CFGW & 0xFFF	0x07FF	—
	ON	SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x7807	—
PIC16C58A	OFF	SUM[0x000:0x7FF] + CFGW & 0xFFF	0x07FF	0xF647
	ON	SUM_XOR4[0x000:0x7FF] + CFGW & 0x00F	0x7807	0x77F5
PIC16CR58A	OFF	SUM[0x000:0x7FF] + CFGW & 0xFFF	0x07FF	—
	ON	SUM[0x000:0x3F] + CFGW & 0xFFF + SUM_ID	0x800E	—

Legend: CFGW = Configuration Word
SUM[a:b] = Sum of locations a through b inclusive
SUM_XOR4[a:b] = XOR of the four high order bits with the four middle and the four low of memory location order bits summed over the locations a through b inclusive. For example, location_a = 0x123 and location_b = 0x456, then SUM_XOR [location_a : location_b] = 0x0007.
SUM_ID = ID locations masked by 0xF then made into a 16-bit value with ID0 as the most significant nibble. For example, ID0 = 0x1, ID1 = 0x2, ID3 = 0x3, ID4 = 0x4, then SUM_ID = 0x1234.
*Checksum = Sum of all individual expressions **modulo** [0xFFFF]

2.3 Embedding Configuration Word and ID Information in the Hex File

To allow portability of code, a PIC16C5X programmer is required to read the configuration word and ID locations from the hex file when loading the hex file. If configuration word information was not present in the hex file then a simple warning message may be issued. Similarly, while saving a hex file, all configuration word and ID information must be included. Configuration word should have the address of 0xFFF. ID locations are mapped at addresses described in Section 1.6.1 and Table 2-2. An option to not include this information may be provided.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

3.0 PIC16C5X HEX DATA FORMATS

MPASM assembler is capable of outputting several different object file formats, suitable for a variety of programmers. A PIC16C5X programmer must be able to accept and send data in one of the following formats: 8-bit Split Format (.HXL/.HXH) and Intel® HEX Format (.HEX). The 8-bit merged (INHX8M) format is preferred.

3.1 8-bit Split Format (.HXL/.HXH)

The Intellec™ split 8-bit file format produces two output files: .HXL and .HXH. The format is the same as the normal 8-bit format, except that the low bytes of the data word are stored in the .HXL file, and the high bytes of the data word are stored in the .HXH file.

EXAMPLE 3-1:

<FILENAME>. HXL:

```
:0A0000000000000000000000000F6
:0A0000000000000000000000000F6
:1000190000284068A8E8C82868A989EA28086ABFAA
:10002900E0E82868BFE8C8080808034303E8E8FFDO
:03003900FFFF19AD
:00000001FF
```

EXAMPLE 3-2:

<FILENAME>. HXH:

```
:0A0000000000000000000000000F6
:0A0000000000000000000000000F6
:1000190000000000000000010101010102020202CA
:1000290002020303030303040404040506070708 83
:0300390008080AAA
:00000001FF
```

3.2 Intel HEX Format (.HEX)

This format produces one 8-bit HEX file with a low byte, high byte combination. Since each address can only contain 8 bits in this format, all addresses are doubled. This format is useful for transferring PIC16C5X series object code to third party EPROM programmers.

EXAMPLE 3-3:

<FILENAME>.HEX

```
:0400100000000000EC
:10000000000000000000000000000000000000000000F0
:0400100000000000EC
:1000320000002800400068 00A800E800C80028016D
:100042006801A9018901EA01280208026A02BF02C5
:10005200E002E80228036803BF03E803C8030804B8
:1000620008040804030443050306E807E807FF0839
:06007200FF08FF08190A57
:00000001FF
```

3.3 8-Bit Word Format

Each data record begins with a nine character prefix and ends with a two character checksum. Each record has the following format:

```
:BBAAAATTHHHH....HHHCC
```

Where:

BB Two-digit hexadecimal byte count representing the number of data words that will appear on the line.

AAAA Four-digit hexadecimal address representing the starting address for the data record.

TT Two-digit record type that will always be '00' except for the end-of-file record which is set to '01'.

HH Two-digit hexadecimal data word.

CC Two-digit hexadecimal checksum that is the two's compliment of the sum of all preceding bytes in the record including the prefix.

© 1995 Microchip Technology Inc.

4.0 PROGRAM/VERIFY MODE ELECTRICAL CHARACTERISTICS

4.1 DC Program Characteristics

TABLE 4-1: DC CHARACTERISTICS (TA = +10°C TO +40°C)

Characteristics	Symbol	Min.	Typ.	Max.	Units	Conditions
Supply voltage during programming	VDDP	4.75	5.0	5.25	V	Note 1
Supply Current (from VDD)	IDDP			25.0	mA	VDD = 5.0V, Fosc1 = 5MHz
Supply Voltage during verify	VDDV	VDD min		VDD max		
Voltage on $\overline{MCLR}$ to stay in Program/Verify Modes	VHH1	VDD + 3		VDD+7.25	V	
Voltage on $\overline{MCLR}$ during programming	VHH2	12.5		13.5	V	
Supply current from programming voltage source	IHH			100	mA	
Current into $\overline{MCLR}$ pin during programming (T0CKI=0)	IHH2		10.0	25.0	mA	VHH = 13.5V, VDD = 6.0V
Input Low Voltage	VIL	VSS		0.15VDD	V	
Input High Voltage	VIH	0.85VDD	5.0	VDD	V	

Note 1: Device must be verified at minimum and maximum operating voltages specified in the data sheet.

4.2 AC Program and Test Mode Characteristics

TABLE 4-2: AC CHARACTERISTICS (TA = +10°C TO +40°C, VDD = 5.0V ± 5%)

Characteristics	Symbol	Min	Typ	Max	Units	Conditions
$\overline{MCLR}$ Rise Time	TR	0.15	1.0	5	µs	
$\overline{MCLR}$ Fall Time	TF	0.5	2.0	5	µs	
Program Mode Setup Time	TPS	1.0			µs	
Data Access Time	TACC			250	ns	
Data Setup Time	TDS	1.0			µs	
Data Hold Time	TDH	1.0			µs	
Output Enable Time	TOE	0		100	ns	
Output Disable Time	TOZ	0		100	ns	
Programming Pulse Width	TPW	10.0			µs	
Programming Pulse Width	TPWF		10,000		µs	Configuration Word only
Recovery Time	TRC	10.0			µs	
Frequency on OSC1	FOSC	DC		5	MHz	For incrementing of the PC

© 1995 Microchip Technology Inc.

4.3 Timing Diagrams

FIGURE 4-1: PROGRAMMING AND VERIFY TIMING WAVEFORM

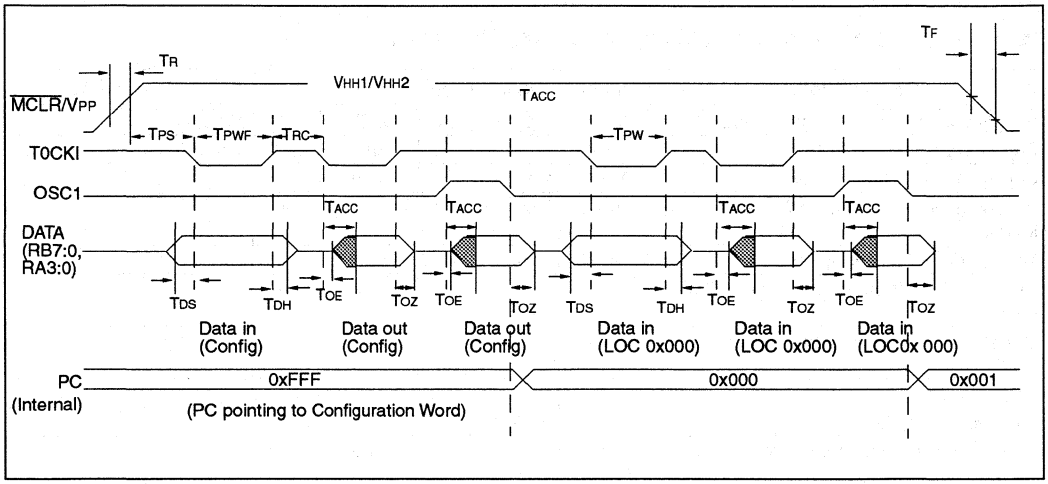
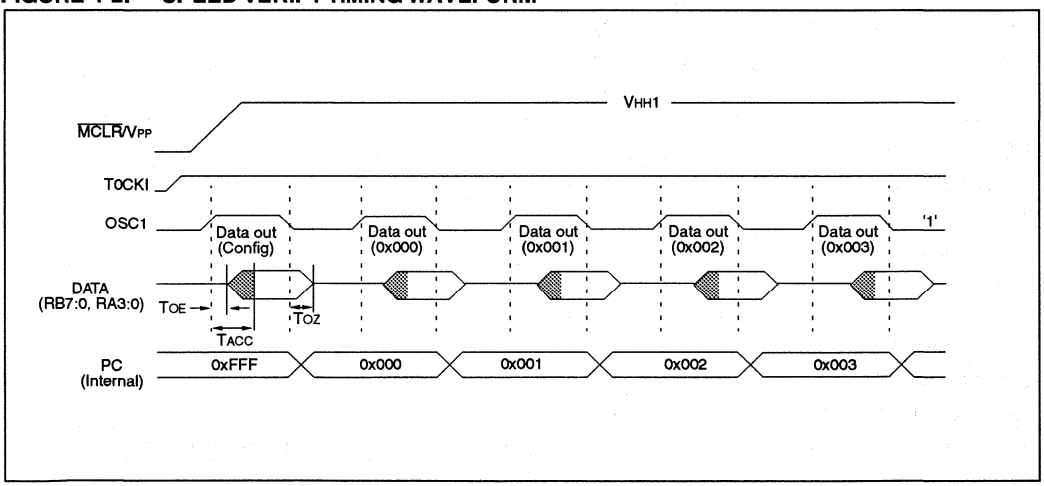


FIGURE 4-2: SPEED VERIFY TIMING WAVEFORM



© 1995 Microchip Technology Inc.

PIC16C6X/7X

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C61
- PIC16C62
- PIC16C64
- PIC16C65
- PIC16C71
- PIC16C73
- PIC16C74
- PIC16C620
- PIC16C621
- PIC16C622

1.0 PROGRAMMING THE PIC16C6X/7X

The PIC16C6X/7X can be programmed using the serial method. In serial mode the PIC16C6X/7X can be programmed while in the users system. This allows for increased design flexibility.

1.1 Hardware Requirements

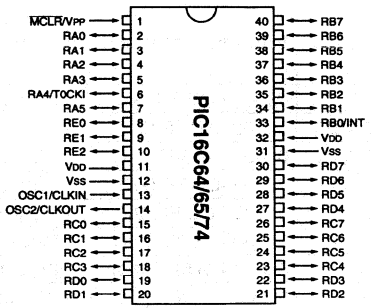
The PIC16C6X/7X requires two programmable power supplies, one for VDD (2.0V to 6.5V recommended) and one for VPP (12V to 14V). Both supplies should have a minimum resolution of 0.25V.

1.2 Programming Mode

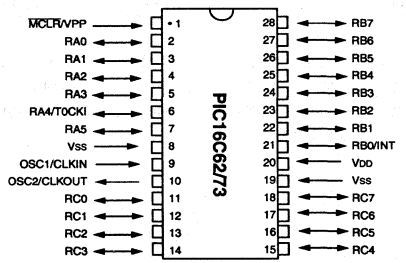
The programming mode for the PIC16C6X/7X allows programming of user program memory, special locations used for ID, and the configuration word for the PIC16C6X/7X.

PIN CONFIGURATIONS

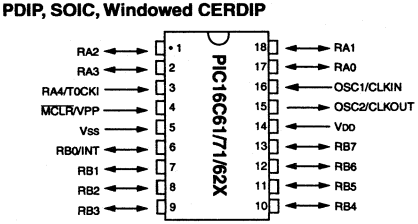
PDIP, Windowed CERDIP



PDIP, SOIC, Windowed CERDIP **(300 mil)**



PDIP, SOIC, Windowed CERDIP



Note: Peripheral pinout functions are not shown (see datasheets for full pinout information).

PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C61/620/621/622/62/64/65/71/73/74

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
MCLR/VPP	VPP	P	Programming Power
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = input, O = Output, P = Power

2.0 PROGRAM MODE ENTRY

2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8K). Table 2-1 shows actual implementation of program memory in the PIC16C6X/7X family.

TABLE 2-1: IMPLEMENTATION OF PROGRAM MEMORY IN THE PIC16C6X/7X

Device	Program Memory Size	Access to Program Memory
PIC16C61	0x000-0x3FF (1K)	PC<9:0>
PIC16C620	0x000 -0x1FF (0.5K)	PC<8:0>
PIC16C621	0x000 - 0x3FF (1K)	PC<9:0>
PIC16C622	0x000 -0x7FF (2K)	PC<10:0>
16C62	0x000 - 0x7FF (2K)	PC <10:0>
16C64	0x000 - 0x7FF (2K)	PC<10:0>
16C65	0x000 - 0xFFF (4K)	PC<11:0>
16C71	0x000 - 0x3FF (1K)	PC<9:0>
16C73	0x000 - 0xFFF (4K)	PC<11:0>
16C74	0x000 - 0xFFF (4K)	PC<11:0>

When the PC reaches the last location of the implemented program memory, it will wrap around and address a location within the physically implemented memory (see Figure 2-1).

In programming mode the program memory space extends from 0x0000 to 0x3FFF, with the first half (0x0000-0x1FFF) being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF to 0x2000 to 0x3FFF and wrap around to 0x2000 (not to 0x0000). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and reenter program/verify mode, as described in Section 2.2.

In the configuration memory space, 0x2000-0x207F or 0x2000-0x20FF are utilized. When in configuration memory, as in the user memory, the 0x2000-0x2XFF segment is repeatedly accessed as PC exceeds 0x2XFF (see Figure 2-1).

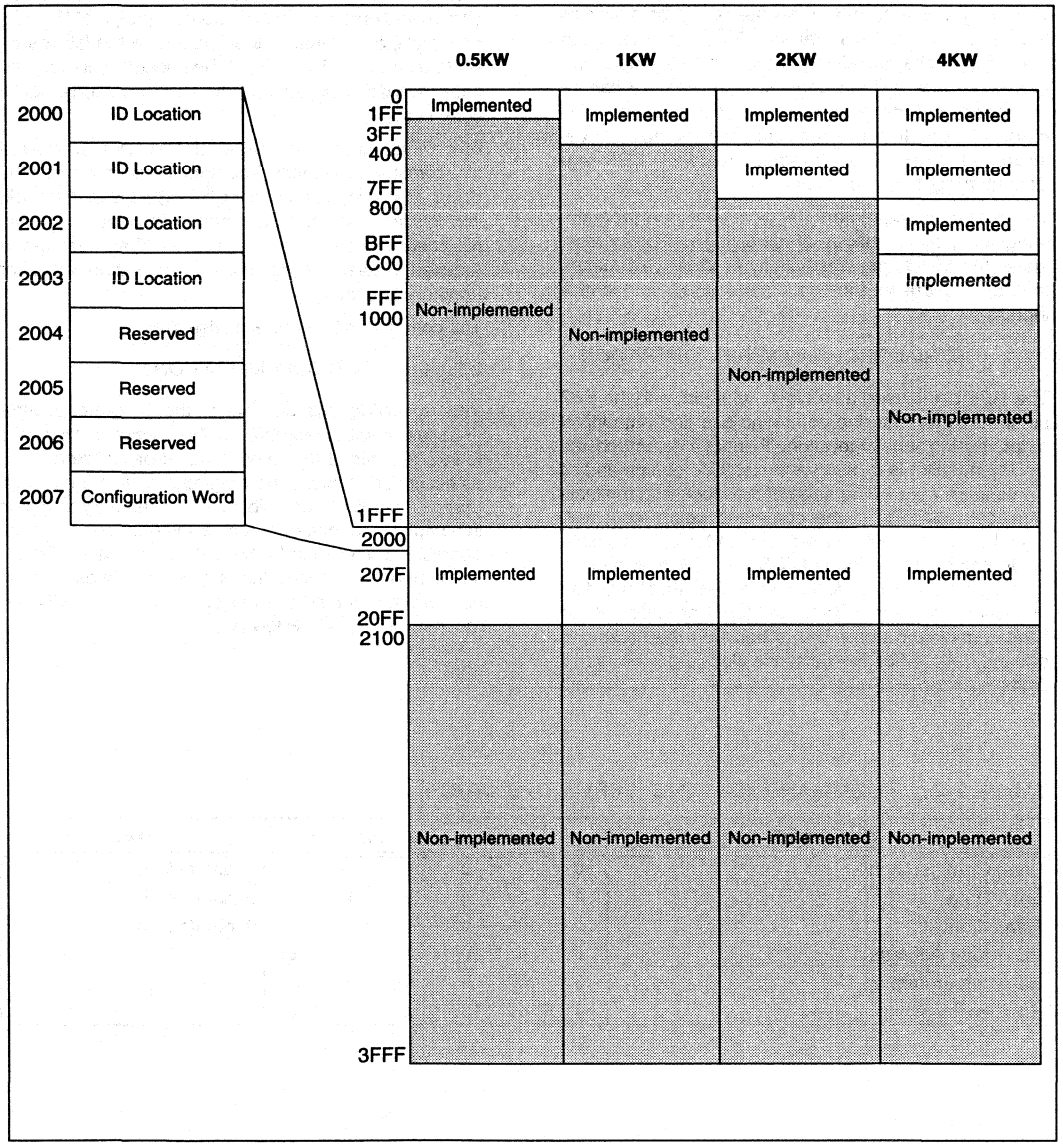
A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000 : 0x2003]. It is recommended that the user use only the four least significant bits of each ID location. In some devices, the ID locations read-out in a scrambled fashion after code protection is enabled. For these devices, it is recommended that ID location is written as "11 1111 1000 bbbb" where 'bbbb' is ID information.

In other devices, the ID locations read out normally, even after code protection. To understand how the devices behave, refer to Table 3-1.

To understand the scrambling mechanism after code protection, refer to Section 4.0.1.

© 1995 Microchip Technology Inc.

FIGURE 2-1: PROGRAM MEMORY MAPPING



© 1995 Microchip Technology Inc.

2.2 Program/Verify Mode

The program/verify mode is entered by holding pins RB6 and RB7 low while raising $\overline{MCLR}$ pin from VIL to VIHH (high voltage). Once in this mode the user program memory and the configuration memory can be accessed and programmed in serial fashion. The mode of operation is serial, and the memory that is accessed is the user program memory. RB6 is a Schmitt trigger input in this mode.

The sequence that enters the device into the programming/verify mode places all other logic into the reset state (the $\overline{MCLR}$ pin was initially at VIL). This means that all I/O are in the reset state (High impedance inputs).

2.2.1 SERIAL PROGRAM/VERIFY OPERATION

The RB6 pin is used as a clock input pin, and the RB7 pin is used for entering command bits and data input/output during serial operation. To input a command, the clock pin (RB6) is cycled six times. Each command bit is latched on the falling edge of the clock with the least significant bit (lsb) of the command being input first. The data on pin RB7 is required to have a minimum setup and hold time (see AC/DC specs) with respect to the falling edge of the clock. Commands that have data associated with them (read and load) are specified to have a minimum delay of 1µs between the command and the data. After this delay the clock pin is cycled 16 times with the first cycle being a start bit and the last cycle being a stop bit. Data is also input and output lsb first. Therefore, during a read operation the lsb will be transmitted onto pin RB7 on the rising edge of the second cycle, and during a load operation the lsb will be latched on the falling edge of the second cycle. A minimum 1µs delay is also specified between consecutive commands.

All commands are transmitted lsb first. Data words are also transmitted lsb first. The data is transmitted on the rising edge and latched on the falling edge of the clock. To allow for decoding of commands and reversal of data pin configuration, a time separation of at least 1µs is required between a command and a data word (or another command).

The commands that are available are:

2.2.1.1 LOAD CONFIGURATION

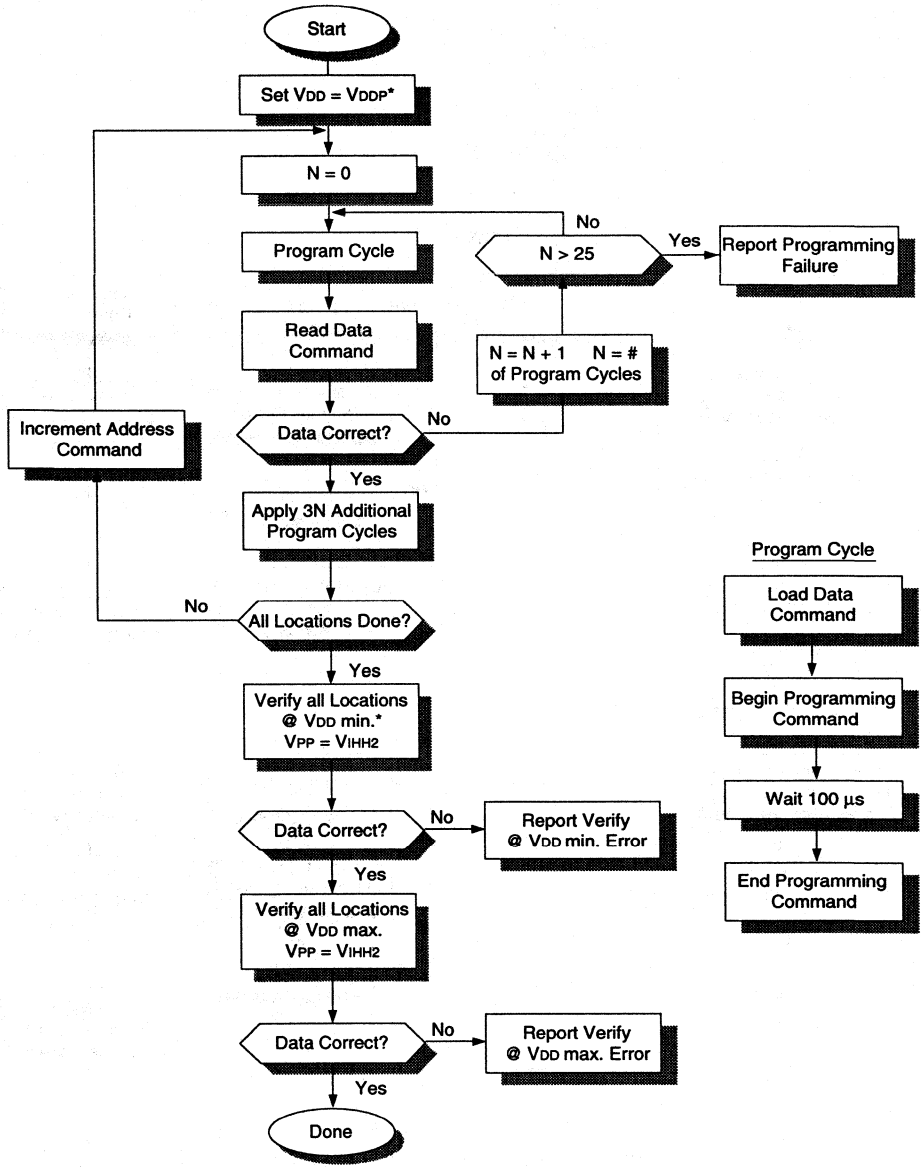
After receiving this command, the program counter (PC) will be set to 0x2000. By then applying 16 cycles to the clock pin, the chip will load 14-bits a "data word" as described above, to be programmed into the configuration memory. A description of the memory mapping schemes for normal operation and configuration mode operation is shown in Figure 2-1. After the configuration memory is entered, the only way to get back to the user program memory is to exit the program/verify test mode by taking $\overline{MCLR}$ low (VIL).

TABLE 2-2: COMMAND MAPPING (SERIAL OPERATION)

Command	Mapping (msb ....lsb)						Data
Load Configuration	0	0	0	0	0	0	0, data(14), 0
Load Data	0	0	0	0	1	0	0, data(14), 0
Read Data	0	0	0	1	0	0	0, data(14), 0
Increment Address	0	0	0	1	1	0	
Begin programming	0	0	1	0	0	0	
End Programming	0	0	1	1	1	0	

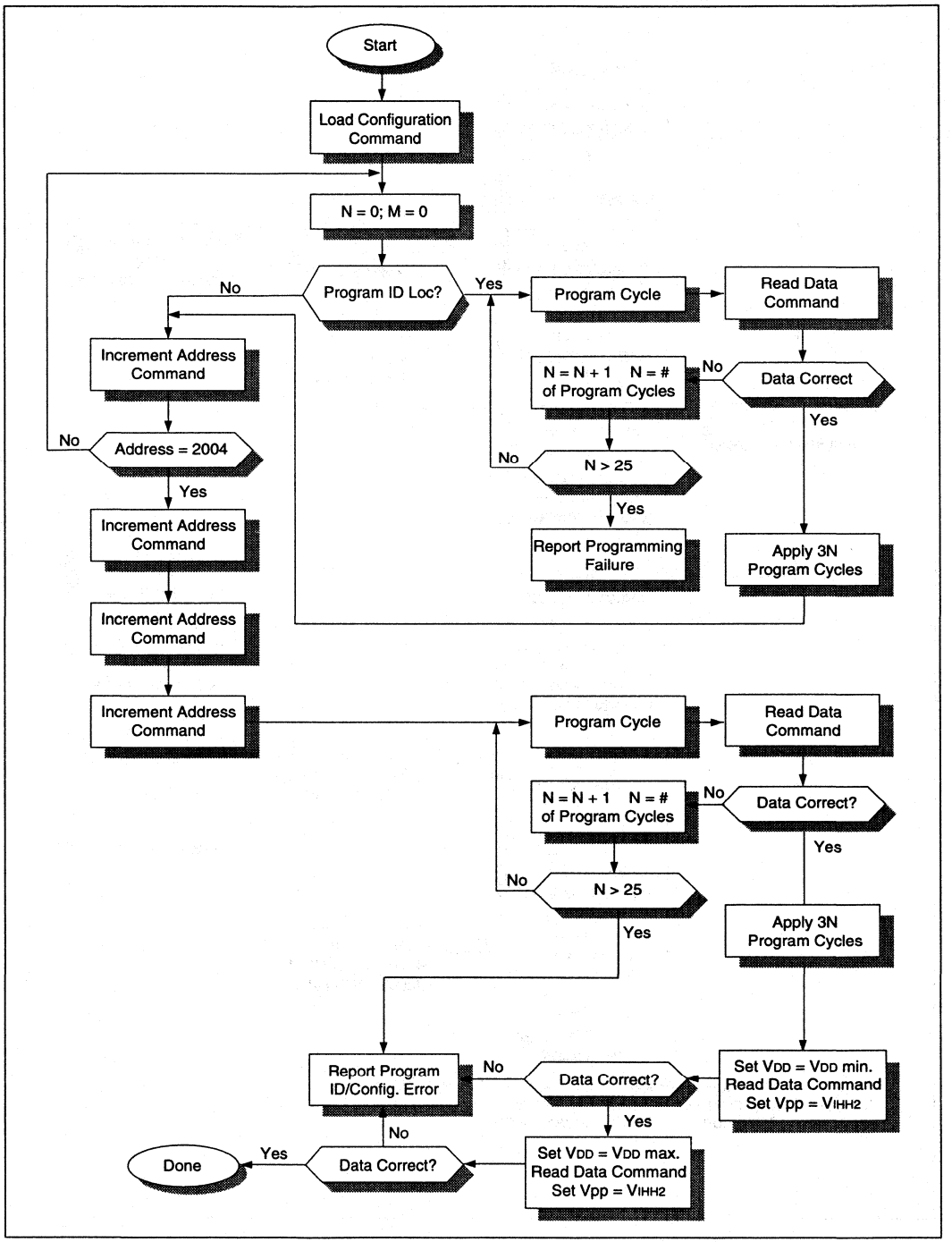
© 1995 Microchip Technology Inc.

FIGURE 2-2: PROGRAM FLOW CHART - PIC16C6X/7X PROGRAM MEMORY



* VDDP = VDD range for programming (typically 4.75V - 5.25V).
VDD min. = Minimum VDD for device operation.
VDD max. = Maximum VDD for device operation.

FIGURE 2-3: PROGRAM FLOW CHART - PIC16C6X/7X CONFIGURATION WORD & ID LOCATIONS



© 1995 Microchip Technology Inc.

2.2.1.2 LOAD DATA

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied, as described previously. A timing diagram for the load data command is shown in Figure 2-4.

2.2.1.3 READ DATA

After receiving this command, the chip will transmit data bits out of the memory currently accessed starting with the second rising edge of the clock input. The RB7 pin will go into output mode on the second rising clock edge, and it will revert back to input mode (hi-impedence) after the 16th rising edge. A timing diagram of this command is shown in Figure 2-5.

2.2.1.4 INCREMENT ADDRESS

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 2-6.

FIGURE 2-4: LOAD DATA COMMAND (PROGRAM/VERIFY)

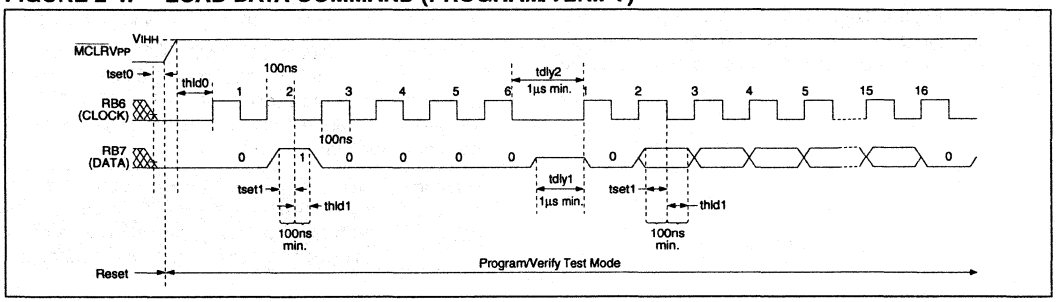


FIGURE 2-5: READ DATA COMMAND (PROGRAM/VERIFY)

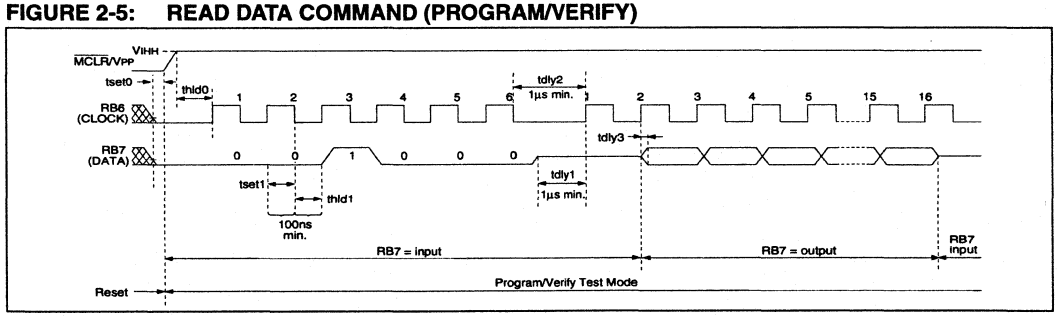
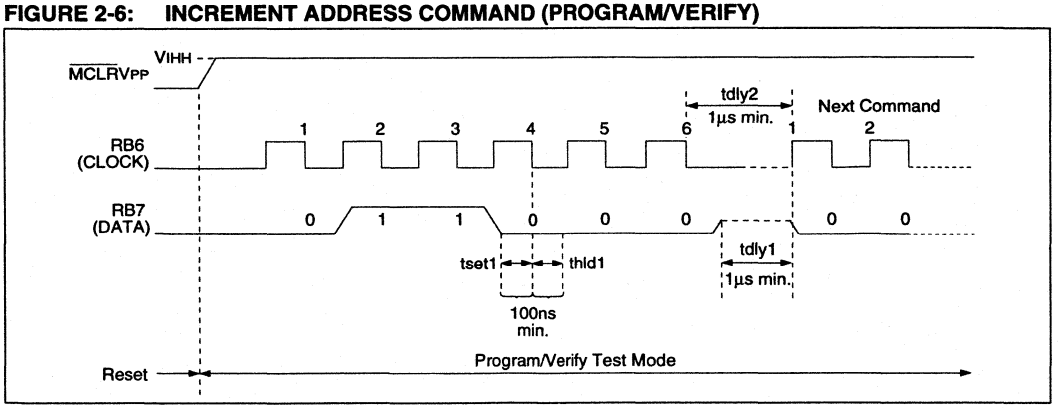


FIGURE 2-6: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



2.2.1.5 BEGIN PROGRAMMING

A load command (load configuration or load data) must be given before every begin programming command. Programming of the appropriate memory (test program memory or user program memory) will begin after this command is received and decoded. Programming should be performed with a series of 100µs programming pulses. A programming pulse is defined as the time between the begin programming command and the end programming command.

2.2.1.6 END PROGRAMMING

After receiving this command, the chip stops programming the memory (configuration program memory or user program memory) that it was programming at the time.

2.3 Programming Algorithm Requires Variable VDD

The PIC16C6X/7X uses an intelligent algorithm. The algorithm calls for program verification at VDD (min.) as well as VDD (max.). Verification at VDD (min.) guarantees good "erase margin". Verification at VDD (max.) guarantees good "program margin".

The actual programming must be done with VDD in the VDDP range (4.75 - 5.25V).

VDDP = VCC range required during programming.

VDD min. = minimum operating VDD spec for the part.

VDD max.= maximum operating VDD spec for the part.

Programmers must verify the PIC16C6X/7X at its specified VDD max. and VDD min. levels. Since Microchip may introduce future versions of the PIC16C6X/7X with a broader VDD range, it is best that these levels are user selectable (defaults are ok).

Note:	Any programmer not meeting these requirements may only be classified as "prototype" or "development" programmer but not a "production" quality programmer.

© 1995 Microchip Technology Inc.

3.0 CONFIGURATION WORD

The PIC16C6X/7X family members have several configuration bits. These bits can be programmed (reads '0') or left unprogrammed (reads '1') to select various device configurations. Figure 3-1 provides an overview of configuration bits.

FIGURE 3-1: CONFIGURATION WORD BIT MAP

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1
PIC16C61/71	—	—	—	—	—	—	—	—	—	CP	PWRTE	WDTE	FOSC1
PIC16C6X/7X	—	—	—	—	—	—	Reserved	Reserved	CP1	CP0	PWRTE	WDTE	FOSC1
PIC16C62X	CP1	CP0	CP1	CP0	CP1	CP0	Reserved	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1

bit 6: **Reserved**
write as '0' for PIC16C6X/7X

5-4**CP1:CP0**, Code Protect

Device	CP1	CP0	Code Protection
PIC16C62 PIC16C64 PIC16C65 PIC16C73 PIC16C74	0	0	All memory protected
	0	1	Upper 3/4 memory protected
	1	0	Upper 1/2 memory protected
	1	1	Code protection off
PIC16C61 PIC16C71	—	0	On
	—	1	Off
PIC16C620	0	0	All memory protected
	0	1	Do not use
	1	0	Do not use
	1	1	Code protection off
PIC16C621	0	0	All memory protected
	0	1	Upper 1/2 memory protected
	1	0	Do not use
	1	1	Code protection off
PIC16C622	0	0	All memory protected
	0	1	Upper 3/4 memory protected
	1	0	Upper 1/2 memory protected
	1	1	Code protection off

bit 6: **BODEN**, Brown Out Enable Bit

bit 4: **PWRTE**, Power Up Timer Enable Bit
PIC16C61/62/64/65/71/73/74:
1 = Power up timer enabled
0 = Power up timer disabled
PIC16C620/621/622:
0 = Power up timer enabled
1 = Power up timer disabled

bit 3-2:**WDTE**, WDT Enable Bit
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>**, Oscillator Selection Bit
11: RC oscillator
10: HS oscillator
01: XT oscillator
00: LP oscillator

© 1995 Microchip Technology Inc.

3.1 Embedding Configuration Word and ID Information in the Hex File

To allow portability of code, the programmer is required to read the configuration word and ID locations from the hex file when loading the hex file. If configuration word information was not present in the hex file then a simple warning message may be issued. Similarly, while saving a hex file, configuration word and ID information must be included. An option to not include this information may be provided.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

TABLE 3-1: CONFIGURATION WORD

PIC16C61

To code protect:

- Protect all memory `XXXXXXXXX0XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C620

To code protect:

- Protect all memory `0000001100XXXX`
- No code protection `1111111111XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read all 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C621

To code protect:

- Protect all memory `0000001100XXXX`
- Protect upper 1/2 memory `0101011101XXXX`
- No code protection `1111111111XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read all 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C622

To code protect:

- Protect all memory `0000001100XXXX`
- Protect upper 3/4 memory `0101011101XXXX`
- Protect upper 1/2 memory `1010101110XXXX`
- No code protection `1111111111XXXX`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read all 0's, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

© 1995 Microchip Technology Inc.

PIC16C62

To code protect:

- Protect all memory XXXXXX1100XXXX
- Protect upper 3/4 memory XXXXXX1101XXXX
- Protect upper 1/2 memory XXXXXX1110XXXX
- No code protection XXXXXX1111XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C64

To code protect:

- Protect all memory XXXXXX1100XXXX
- Protect upper 3/4 memory XXXXXX1101XXXX
- Protect upper 1/2 memory XXXXXX1110XXXX
- No code protection XXXXXX1111XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C65

To code protect:

- Protect all memory XXXXXX1100XXXX
- Protect upper 3/4 memory XXXXXX1101XXXX
- Protect upper 1/2 memory XXXXXX1110XXXX
- No code protection XXXXXX1111XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C71

To code protect:

- Protect all memory XXXXXXXXX0XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segmentz	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

© 1995 Microchip Technology Inc.

PIC16C73

To code protect:

- Protect all memory XXXXXX1100XXXX
- Protect upper 3/4 memory XXXXXX1101XXXX
- Protect upper 1/2 memory XXXXXX1110XXXX
- No code protection XXXXXX1111XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

PIC16C74

To code protect:

- Protect all memory XXXXXX1100XXXX
- Protect upper 3/4 memory XXXXXX1101XXXX
- Protect upper 1/2 memory XXXXXX1110XXXX
- No code protection XXXXXX1111XXXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Unprotected memory segment	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled
Protected memory segment	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations (0x2000 : 0x2003)	Read Unscrambled, Write Enabled	Read Unscrambled, Write Enabled

Legend: X = Don't care

4.0 CODE PROTECTION

The program code written into the EPROM can be protected by writing to the CP0 & CP1 bits of the configuration word.

In PIC16C61/71 it is still possible to program locations 0x000 through 0x03F, after code protection. For all other devices, writing to all protected memory is disabled.

4.0.1 PROGRAMMING LOCATIONS 0X0000 TO 0X03F AFTER CODE PROTECTION

For PIC16C61/71 devices, once code protection is enabled, all program memory locations read out in a scrambled fashion. The ID locations and the configuration word also read out in a scrambled fashion. Further programming is disabled for locations 0x040 and above. It is possible to program the ID locations and the configuration word.

For PIC16C61/71 devices, program memory locations 0x000 through 0x03F are essentially not protected, i.e., these locations can be further programmed after code protection is enabled. However, since the data reads out in a scrambled fashion, to correctly overprogram these locations, the programmer must program seven bits at a time. For example, to program 0x3AD2 ("11 1010 1101 0010") in a blank location, first program the location with "11 1111 1101 0010" and verify scrambled output to be "xx xxxx x101 0010". Next, program the location with "11 1010 1101 0010" and verify scrambled output to be "xx xxxx x101 1000".

For all other PIC16C6X/7X devices, once code protection is enabled, all protected segments read '0's (or "garbage values") and are prevented from further programming. All unprotected segments, including ID locations and configuration word, read normally. These can be programmed.

4.1 Checksum

Checksum is calculated by reading the contents of the PIC16CXX memory locations and adding up the opcodes up to the maximum addressable location, e.g., 0x3FF for the PIC16C61. Then ID and configuration locations are added. Any carry bits exceeding 16-bits are neglected. Checksum computations for each member of PIC16CXX devices is shown in Table 4-1.

© 1995 Microchip Technology Inc.

TABLE 4-1: CHECKSUM COMPUTATION

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and max address
PIC16C61	OFF	SUM[0x000:0x3FF] + CFGW & 0x3FFF	0x3BFF	0x07CD
	ON	SUM_XNOR7[0x000:0x3FF] + CFGW & 0x007F	0xFC6F	0xFC15
PIC16C620	OFF	SUM[0x000:0x3FF] + CFGW & 0x3F7F	0x3DFF	0x094D
	ALL	SUM_ID + CFGW & 0x3F7F	0x3FFE	0x099C
PIC16C621	OFF	SUM[0x000:0x3FF] + CFGW & 0x3F7F	0x3BFF	0x074D
	1/2	SUM[0x000:0x1FF] + CFGW & 0x3F7F + SUM_ID	0x3A7E	0x0093
	ALL	CFGW & 0x3F7F + SUM_ID	0x3C7E	0x079C
PIC16C622	OFF	SUM[0x000:0x7FF] + CFGW & 0x3F7F	0x37FF	0x034D
	1/2	SUM[0x000:0x3FF] + CFGW & 0x3F7F + SUM_ID	0x347E	0x0FA3
	3/4	SUM[0x000:0x1FF] + CFGW & 0x3F7F + SUM_ID	0x367E	0xF093
	ALL	CFGW & 0x3F7F + SUM_ID	0x387E	0x039C
PIC16C62	OFF	SUM[0x000:0x7FF] + CFGW & 0x3FBF	0x37BF	0x038D
	1/2	SUM[0x000:0x3FF] + SUM_XNOR7[0x400:0x7FF] + CFGW & 0x3FBF	0x37AF	0x1D69
	3/4	SUM[0x000:0x1FF] + SUM_XNOR7[0x200:0x7FF] + CFGW & 0x3FBF	0x379F	0x1059
	ALL	SUM_XNOR7[0x000:0x7FF] + CFGW & 0x3FBF	0x378F	0x3735
PIC16C64	OFF	SUM[0x000:0x7FF] + CFGW & 0x3FBF	0x37BF	0x038D
	1/2	SUM[0x000:0x3FF] + SUM_XNOR7[0x400:0x7FF] + CFGW & 0x3FBF	0x37AF	0x1D69
	3/4	SUM[0x000:0x1FF] + SUM_XNOR7[0x200:0x7FF] + CFGW & 0x3FBF	0x379F	0x1D59
	ALL	SUM_XNOR7[0x000:0x7FF] + CFGW & 0x3FBF	0x378F	0x3735
PIC16C65	OFF	SUM[0x000:0xFFF] + CFGW & 0x3FBF	0x2FBF	0xFB8D
	1/2	SUM[0x000:0x7FF] + SUM_XNOR7[0x800:FFF] + CFGW & 0x3FBF	0x2FAF	0x1569
	3/4	SUM[0x000:0x3FF] + SUM_XNOR7[0x400:FFF] + CFGW & 0x3FBF	0x2F9F	0x1559
	ALL	SUM_XNOR7[0x000:0xFFF] + CFGW & 0x3FBF	0x2F8F	0x2F35
PIC16C71	OFF	SUM[0x000:0x3FF] + CFGW & 0x3FFF	0x3BFF	0x07CD
	ON	SUM_XNOR7[0x000:0x3FF] + CFGW & 0x3FFF	0xFC6F	0xFC15
PIC16C73	OFF	SUM[0x000:0xFFF] + CFGW & 0x3FBF	0x2FBF	0xFB8D
	1/2	SUM[0x000:0x7FF] + SUM_XNOR7[0x800:FFF] + CFGW & 0x3FBF	0x2FAF	0x0569
	3/4	SUM[0x000:0x3FF] + SUM_XNOR7[0x400:FFF] + CFGW & 0x3FBF	0x2F9F	0x1559
	ALL	SUM_XNOR7[0x000:0xFFF] + CFGW & 0x3FBF	0x2F8F	0x2F35
PIC16C74	OFF	SUM[0x000:0xFFF] + CFGW & 0x3FBF	0x2FBF	0xFB8D
	1/2	SUM[0x000:0x7FF] + SUM_XNOR7[0x800:FFF] + CFGW & 0x3FBF	0x2FAF	0x1569
	3/4	SUM[0x000:0x3FF] + SUM_XNOR7[0x400:FFF] + CFGW & 0x3FBF	0x2F9F	0x1559
	ALL	SUM_XNOR7[0x000:0xFFF] + CFGW & 0x3FBF	0x2F8F	0x2B35

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a through b inclusive]
SUM_XNOR7[a:b] = XNOR of the seven high order bits of memory location with the seven low order bits summed over locations a through b inclusive. For example, location_a = 0x123 and location_b = 0x456, then SUM_XNOR7 [location_a : location_b] = 0x001F.
SUM_ID = ID locations masked by 0xF then made into a 16-bit value with ID0 as the most significant nibble. For example, ID0 = 0x12, ID1 = 0x37, ID2 = 0x4, ID3 = 0x26, then SUM_ID = 0x2746.
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]

TABLE 4-2: AC/DC Timing Requirements for Program/Verify Mode

AC/DC Characteristics, Power Supply Pins	Standard Operating Conditions Operating Temperature +10°C ≤ TA ≤ +40°C, unless otherwise stated Operating Voltage 4.5V ≤ VDD ≤ 5.5V, unless otherwise stated.					
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
General						
Supply voltage during programming	VDDP	4.75	5.0	5.25	V	
Supply voltage during verify	VDDV	VDD min.		VDD max.	V	Note 1

Note 1: Program must be verified at the minimum and maximum VDD limits for the part.
Note 2: VIHH must be greater than VDD + 4.5V to stay in programming/verify mode.

TABLE 4-2: AC/DC Timing Requirements for Program/Verify Mode (Continued)

AC/DC Characteristics, Power Supply Pins	Standard Operating Conditions Operating Temperature +10°C ≤ TA ≤ +40°C, unless otherwise stated Operating Voltage 4.5V ≤ VDD ≤ 5.5V, unless otherwise stated.					
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Supply current (from VDD) during programming	IDDP			20	mA	
Programming supply current (from VPP)	IPP			50	mA	
Voltage on $\overline{MCLR}$/VPP during programming	VIHH	12.5		13.5	V	Note 2
Voltage on $\overline{MCLR}$/VPP during verify	VIHH2	VDD + 4.0		13.5		
$\overline{MCLR}$/VPP rise time (VSS to VHH) for test mode entry	tVHHR			1.0	µs	
(RB6, RB7) input high level	VIH1	0.8 VDD			V	Schmitt trigger input
(RB6, RB7) input low level	VIL1	0.2 VDD			V	Schmitt trigger input
Serial Program Verify						
Data in setup time before clock ↓	tset1	100			ns	
Data in hold time after clock ↓	thld1	100			ns	
Data input not driven to next clock input (delay required between command/data or command/command)	tdly1	1.0			µs	
Delay between clock ↓ to clock ↑ of next command or data	tdly2	1.0			µs	
Clock ↑ to date out valid (during read data)	tdly3	80			ns	

Note 1: Program must be verified at the minimum and maximum VDD limits for the part.
Note 2: VIHH must be greater than VDD + 4.5V to stay in programming/verify mode.

© 1995 Microchip Technology Inc.

PIC16C84

EEPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16C84

1.0 PROGRAMMING THE PIC16C84

The PIC16C84 is programmed using the serial method. The serial mode will allow the PIC16C84 to be programmed while in the users system. This allows for increased design flexibility.

1.1 Hardware Requirements

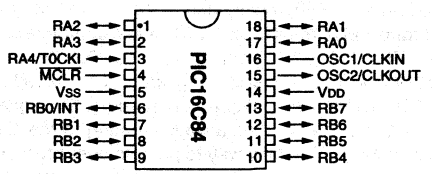
The PIC16C84 requires one programmable power supply for VDD (4.5V to 5.5V) and a VPP of 12V to 14V. Both supplies should have a minimum resolution of 0.25V.

1.2 Programming Mode

The programming mode for the PIC16C84 allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

PIN CONFIGURATION

PDIP, SOIC



PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16C84

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
MCLR	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = input, O = Output, P = Power

*In PIC16C84, programming high voltage is internally generated. To activate the programming mode, high voltage needs to be applied to MCLR input. This means that MCLR does not draw any significant current.

© 1995 Microchip Technology Inc.

2.0 PROGRAM MODE ENTRY

2.1 User Program Memory Map

The user memory space extends from 0x0000 to 0x1FFF (8K), of which 1K (0x0000 - 0x03FF) is physically implemented. In actual implementation the on-chip user program memory is accessed by the lower 10-bits of the PC, with the upper 3-bits of the PC ignored. Therefore if the PC is greater than 0x3FF, it will wrap around and address a location within the physically implemented memory. (See Figure 2-1).

In programming mode the program memory space extends from 0x0000 to 0x3FFF, with the first half (0x0000-0x1FFF) being user program memory and the second half (0x2000-0x3FFF) being configuration memory. The PC will increment from 0x0000 to 0x1FFF to 0x2000 to 0x3FFF and wrap around to 0x2000 (not to 0x0000). Once in configuration memory, the highest bit of the PC stays a '1', thus always pointing to the configuration memory. The only way to point to user program memory is to reset the part and reenter program/verify mode as described in Section 2.3.

In the configuration memory space, 0x2000-0x200F are physically implemented. Locations beyond 0x200F will physically access user memory. (See Figure 2-1).

2.2 ID Locations

A user may store identification information (ID) in four ID locations. The ID locations are mapped in [0x2000 : 0x2003]. It is recommended that the user use only the four least significant bits of each ID location. In some devices, the ID locations read-out in a scrambled fashion after code protection is enabled. For these devices, it is recommended that ID location is written as "11 1111 1000 bbbb" where 'bbbb' is ID information.

In other devices, the ID locations read out normally, even after code protection. To understand how the devices behave, refer to Table 3-1.

To understand the scrambling mechanism after code protection, refer to Section 3.1.

© 1995 Microchip Technology Inc.

FIGURE 2-1: PROGRAM MEMORY MAPPING

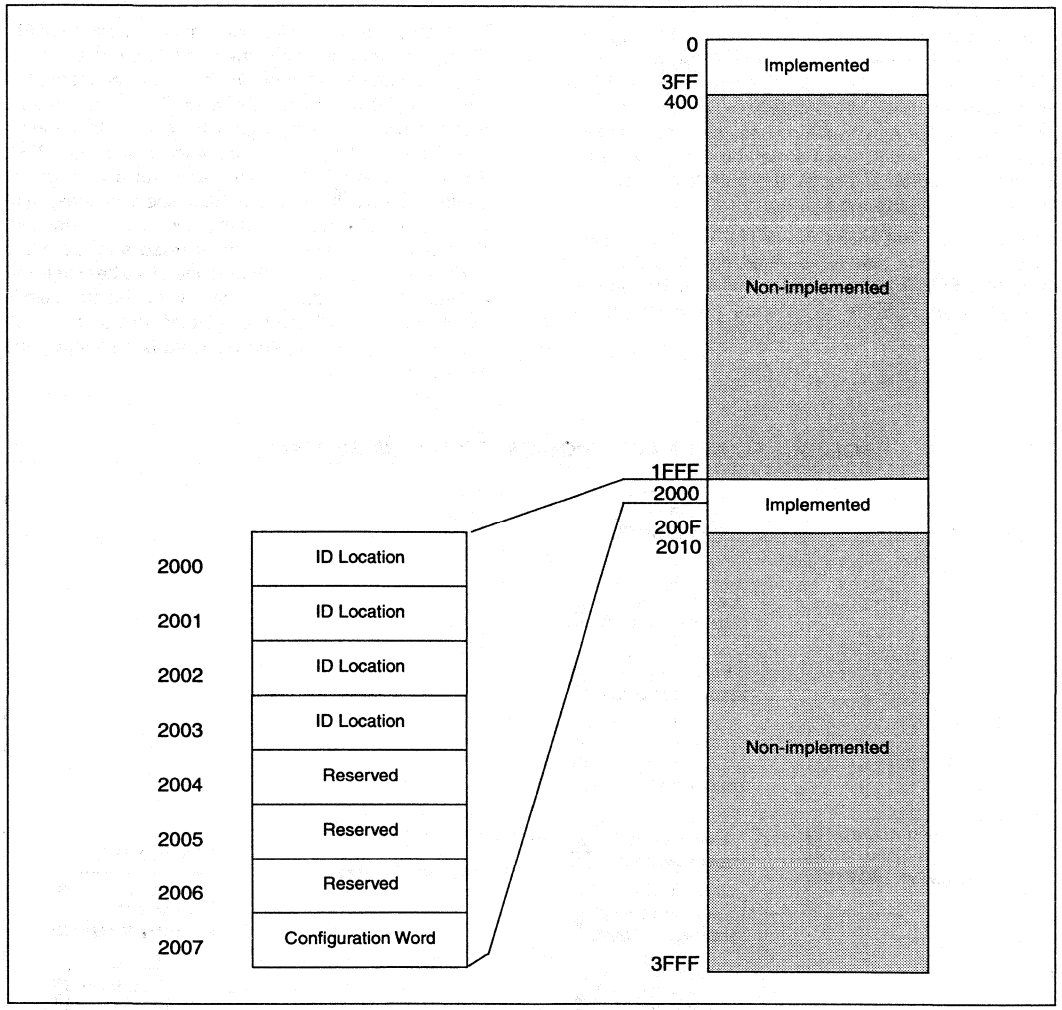


TABLE 2-1: COMMAND MAPPING (SERIAL OPERATION)

Command	Mapping (msb ... lsb)						Data
Load Configuration	0	0	0	0	0	0	0, data (14), 0
Load Data for Program Memory	0	0	0	0	1	0	0, data (14), 0
Read Data from Program Memory	0	0	0	1	0	0	0, data (14), 0
Increment Address	0	0	0	1	1	0	
Begin Programming	0	0	1	0	0	0	
Load Data for Data Memory	0	0	0	0	1	1	0, data (14), 0
Read Data from Data Memory	0	0	0	1	0	1	0, data (14), 0
Bulk Erase Program Memory	0	0	1	0	0	1	
Bulk Erase Data Memory	0	0	1	0	1	1	

© 1995 Microchip Technology Inc.

2.3 Program/Verify Mode

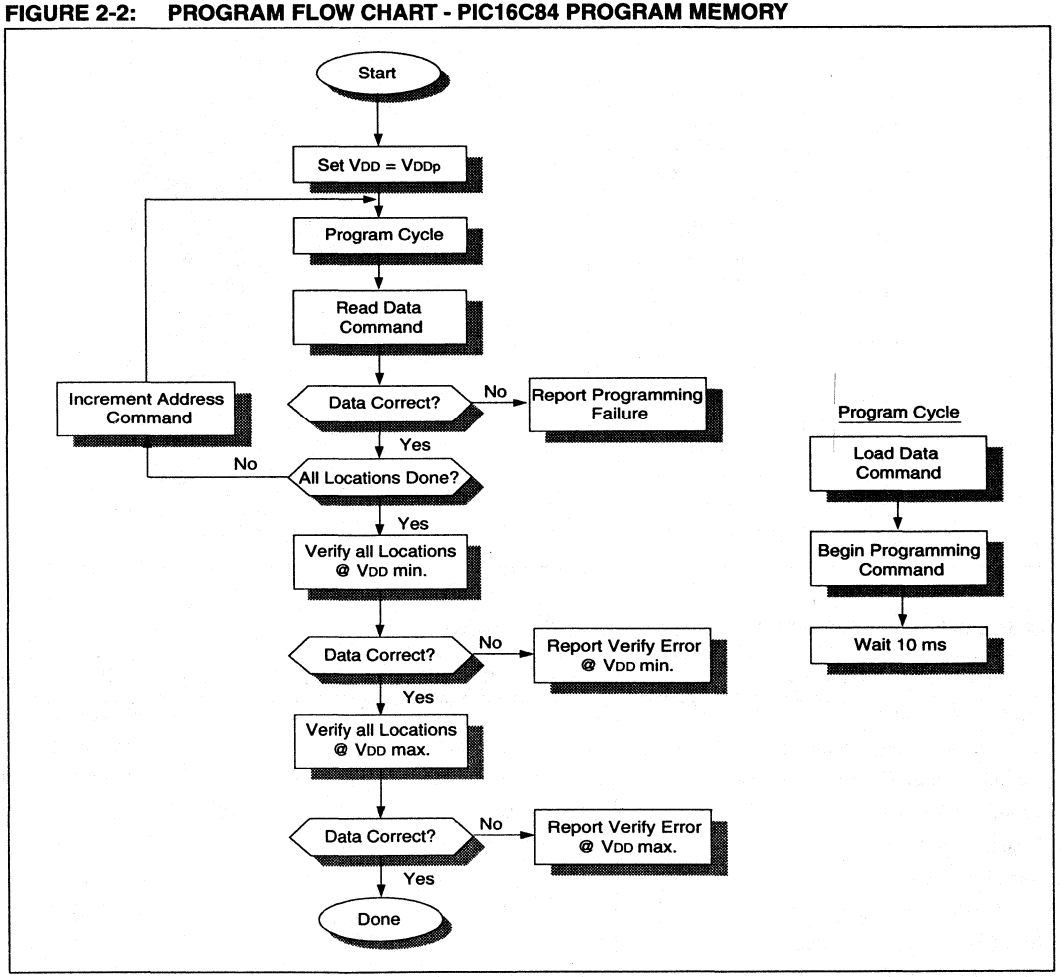
The program/verify mode is entered by holding pins RB6 and RB7 low while raising $\overline{MCLR}$ pin from V_{IL} to V_{IHH} (high voltage). Once in this mode the user program memory and the configuration memory can be accessed and programmed in serial fashion. The mode of operation is serial, and the memory that is accessed is the user program memory. RB6 and RB7 are Schmitt Trigger Inputs in this mode.

The sequence that enters the device into the programming/verify mode places all other logic into the reset state (the $\overline{MCLR}$ pin was initially at V_{IL}). This means that all I/O are in the reset state (High impedance inputs).

2.3.1 SERIAL PROGRAM/VERIFY OPERATION

The RB6 pin is used as a clock input pin, and the RB7 pin is used for entering command bits and data input/output during serial operation. To input a command, the clock pin (RB6) is cycled six times. Each command bit is latched on the falling edge of the clock with the least significant bit (lsb) of the command being input first. The data on pin RB7 is required to have a minimum setup and hold time (see AC/DC specifications) with respect to the falling edge of the clock. Commands that have data associated with them (read and load) are specified to have a minimum delay of ~~us~~ between the command and the data. After this delay, the clock pin is cycled 16 times with the first cycle being a start bit and the last cycle being a stop bit. Data is also input and output lsb first.

FIGURE 2-2: PROGRAM FLOW CHART - PIC16C84 PROGRAM MEMORY



© 1995 Microchip Technology Inc.

Therefore, during a read operation the lsb will be transmitted onto pin RB7 on the rising edge of the second cycle, and during a load operation the lsb will be latched on the falling edge of the second cycle. A minimum 1µs delay is also specified between consecutive commands.

All commands are transmitted lsb first. Data words are also transmitted lsb first. The data is transmitted on the rising edge and latched on the falling edge of the clock. To allow for decoding of commands and reversal of data pin configuration, a time separation of at least ~~us~~ is required between a command and a data word (or another command).

The commands that are available are:

2.3.1.1 LOAD CONFIGURATION

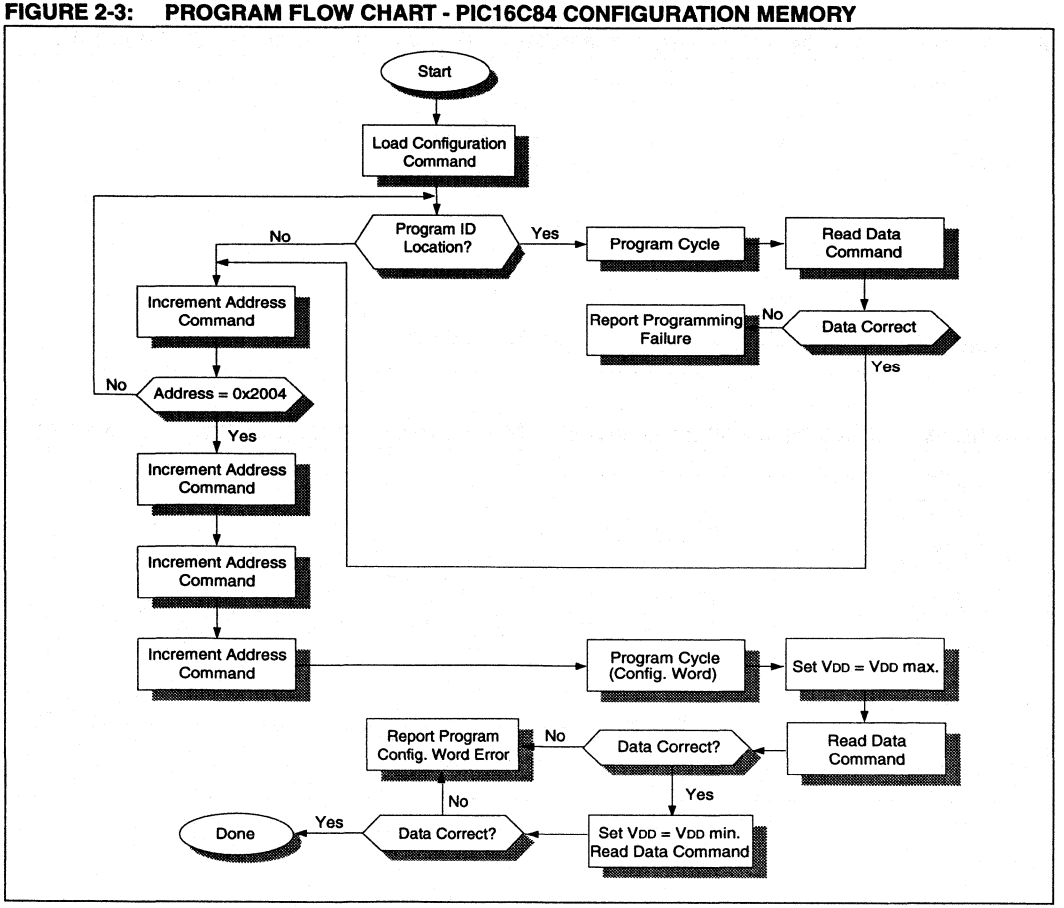
After receiving this command, the program counter (PC) will be set to 0x2000. By then applying 16 cycles to the clock pin, the chip will load 14-bits in a "data word", as described above, to be programmed into the configuration memory. A description of the memory mapping schemes of the program memory for normal operation and configuration mode operation is shown in Figure 2-1. After the configuration memory is entered, the only way to get back to the user program memory is to exit the program/verify test mode by taking $\overline{MCLR}$ low (V_{IL}).

2.3.1.2 LOAD DATA FOR PROGRAM MEMORY

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied, as described previously. A timing diagram for the load data command is shown in Figure 2-4.

3

FIGURE 2-3: PROGRAM FLOW CHART - PIC16C84 CONFIGURATION MEMORY



2.3.1.3 LOAD DATA FOR DATA MEMORY

After receiving this command, the chip will load in a 14-bit "data word" when 16 cycles are applied. However, the data memory is only 8-bits wide, and thus only the first 8-bits of data after the start bit will be programmed into the data memory. It is still necessary to cycle the clock the full 16 cycles in order to allow the internal circuitry to reset properly. The data memory contains 64 words. Only the lower 8-bits of the PC are decoded by the data memory, and therefore if the PC is greater than 0x3F, it will wrap around and address a location within the physically implemented memory.

2.3.1.4 READ DATA FROM PROGRAM MEMORY

After receiving this command, the chip will transmit data bits out of the program memory (user or configuration) currently accessed starting with the second rising edge of the clock input. The RB7 pin will go into output mode on the second rising clock edge, and it will revert back to input mode (hi-impedance) after the 16th rising edge. A timing diagram of this command is shown in Figure 2-5.

2.3.1.5 READ DATA FROM DATA MEMORY

After receiving this command, the chip will transmit data bits out of the data memory starting with the second rising edge of the clock input. The RB7 pin will go into output mode on the second rising edge, and it will revert back to input mode (hi-impedance) after the 16th rising edge. As previously stated, the data memory is 8-bits wide, and therefore, only the first 8-bits that are output are actual data.

2.3.1.6 INCREMENT ADDRESS

The PC is incremented when this command is received. A timing diagram of this command is shown in Figure 2-6.

FIGURE 2-4: LOAD DATA FOR PROGRAM MEMORY COMMAND (SERIAL PROGRAM/VERIFY)

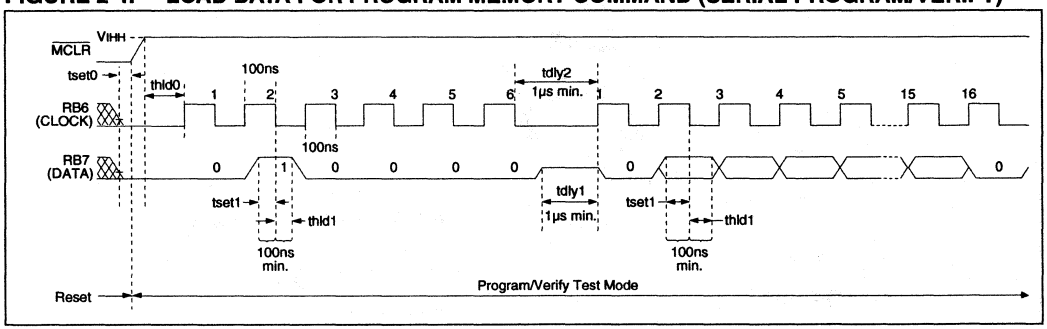
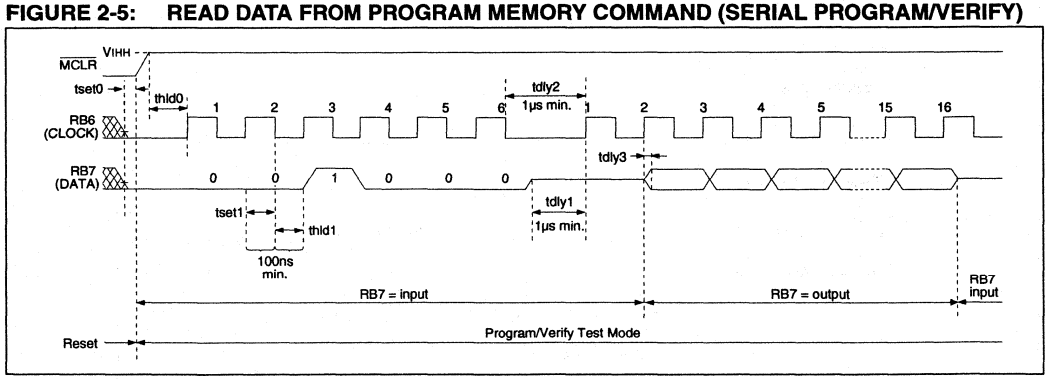
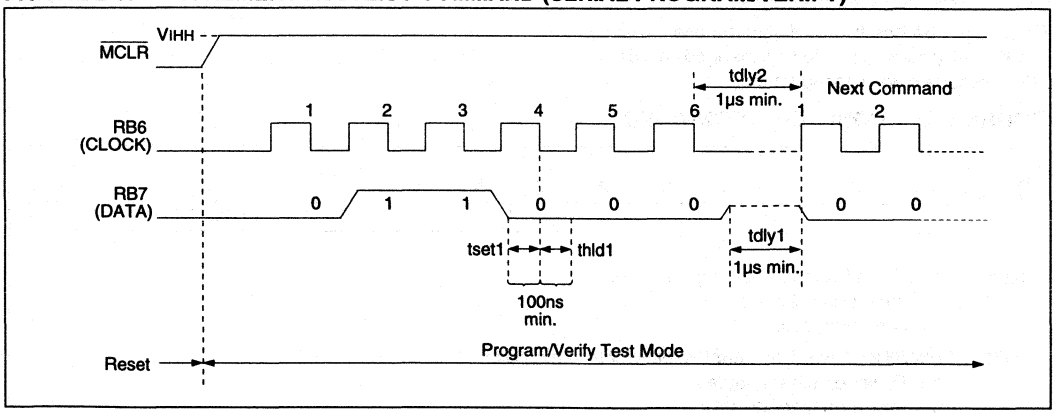


FIGURE 2-5: READ DATA FROM PROGRAM MEMORY COMMAND (SERIAL PROGRAM/VERIFY)



© 1995 Microchip Technology Inc.

FIGURE 2-6: INCREMENT ADDRESS COMMAND (SERIAL PROGRAM/VERIFY)



3

2.3.1.7 BEGIN PROGRAMMING

A load command must be given before every begin programming command. Programming of the appropriate memory (test program memory, user program memory or data memory) will begin after this command is received and decoded. An internal timing mechanism executes an erase before write. The user must allow 10ms for programming to complete. No "end programming" command is required.

2.3.1.8 BULK ERASE PROGRAM MEMORY

After this command is performed, the next program command will erase the entire program memory. The erase time is specified to be 10ms.

If the address is pointing to user memory, only the user memory will be erased.

If the address is pointing to the test program memory (0x2000 - 0x200F), then both the user memory and the test memory will be erased. The configuration word will not be erased, even if the address is pointing to location 0x2007.

2.3.1.9 BULK ERASE DATA MEMORY

After this command is performed, the next program command will erase the entire data memory. The erase time is specified to be 10ms.

2.4 Programming Algorithm Requires Variable VDD

The PIC16C84 uses an intelligent algorithm. The algorithm calls for program verification at VDD (min.) as well as VDD (max.). Verification at VDD (min.) guarantees good "erase margin". Verification at VDD (max) guarantees good "program margin".

The actual programming must be done with VDD in the VDDP range (4.5 - 5.5V).

VDDP = VCC range required during programming.

VDD min. = minimum operating VDD spec for the part.

VDD max.= maximum operating VDD spec for the part.

Programmers must verify the PIC16C84 at its specified VDD max. and VDD min. levels. Since Microchip may introduce future versions of the PIC16C84 with a broader VDD range, it is best that these levels are user selectable (defaults are ok).

Note:	Any programmer not meeting these requirements may only be classified as "prototype" or "development" programmer but not a "production" quality programmer.

3.0 CONFIGURATION WORD

The PIC16C84 has five configuration bits. These bits can be set (reads '0') or left unchanged (reads '1') to select various device configurations.

FIGURE 3-1: CONFIGURATION WORD

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CP	PWRTE	WDTE	FOSC1	FOSC0

bit 4: **CP,** Code Protection Configuration Bit
1 = code protection off
0 = code protection on

bit 3: **PWRTE,** Power Up Timer Enable Configuration Bit
1 = Power up timer enabled
0 = Power up timer disabled

bit 3-2: **WDTE,** WDT Enable Configuration Bits
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>,** Oscillator Selection Configuration Bits
11: RC oscillator
10: HS oscillator
01: XT oscillator
00: LP oscillator

3.1 Code Protection

For PIC16C84 devices, once code protection is enabled, all program memory locations read out in a scrambled fashion. The ID locations and the configuration word also read out in a scrambled fashion. Further programming is disabled for the entire program memory as well as data memory. It is possible to program the ID locations and the configuration word.

3.1.1 DISABLING CODE-PROTECTION

It is recommended that the following procedure be performed before any other programming is attempted. It is also possible to turn code protection off (code protect bit = 1) using this procedure; however, ***all data within the program memory and the data memory will be erased when this procedure is executed, and thus, the security of the data or code is not compromised.***

Procedure to disable code protect:

a) Execute load configuration (with a '1' in bit 4, code protect).
b) Increment to configuration word location (0x2007)
c) Execute command (000001)
d) Execute command (000111)
e) Execute 'Begin Programming' (001000)
f) Wait 10ms
g) Execute command (000001)
h) Execute command (000111)

© 1995 Microchip Technology Inc.

3.2 Embedding Configuration Word and ID Information in the Hex File

To allow portability of code, the programmer is required to read the configuration word and ID locations from the hex file when loading the hex file. If configuration word information was not present in the hex file then a simple warning message may be issued. Similarly, while saving a hex file, all configuration word and ID information must be included. An option to not include this information may be provided.

Specifically for the PIC16C84, the EEPROM data memory should also be embedded in the hex file (see Section 4.0).

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

TABLE 3-1: CHECKSUM COMPUTATION

TABLE 3-2:

Device	Code Protect	Checksum*	Blank Value	0x25E6 at 0 and max address
PIC16C84	OFF	SUM[0x000:0x3FF] + CFGW & 0x3FFF	0x3BFF	0x07CD
	ON	SUM_XNOR7[0x000:0x3FF] + CFGW & 0x007F	0xFC6F	0xFC15

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_XNOR7[a:b] = XNOR of the seven high order bits of memory location with the seven low order bits summed over locations a through b inclusive. For example, location_a = 0x123 and location_b = 0x456, then SUM_XNOR7 [location_a : location_b] = 0x001F.
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]

TABLE 3-3: CONFIGURATION WORD

PIC16C84

To code protect: XXXXXXXX0XXX

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0x2007)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory.	Read Scrambled, Write Disabled	Read Unscrambled, Write Enabled
ID Locations [0x2000 : 0x2003]	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled

Legend: X = Don't care

© 1995 Microchip Technology Inc.

4.0 EMBEDDING DATA EEPROM CONTENTS IN HEX FILE

The programmer should be able to read data EEPROM information from a hex file and conversely (as an option) write data EEPROM contents to a hex file along with program memory information and fuse information.

The 64 data memory locations are logically mapped starting at address 0x2100. The format for data memory storage is one data byte per address location, lsb aligned.

TABLE 4-1: AC/DC TIMING REQUIREMENTS FOR PROGRAM/VERIFY TEST MODE

AC/DC Characteristics, Power Supply Pins	Standard Operating Conditions Operating Temperature +10°C ≤ TA ≤ +70°C, unless otherwise stated Operating Voltage 4.5V ≤ VDD ≤ 5.5V, unless otherwise stated.					
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions/Comments**
Supply voltage during programming	VDDP	4.5	5.0	5.5	V	
Supply voltage during verify	VDDV	VDD min.		VDD max.	V	Note 1
High voltage on $\overline{MCLR}$ for test mode entry	VIHH	12		14.0	V	Note 2
Supply current (from VDD) during program/verify	IDDP			50	mA	
Supply current from VIHH (on $\overline{MCLR}$)	IHH			200	µA	
$\overline{MCLR}$ rise time (VSS to VHH) for test mode entry	tVHHR			1.0	µs	
(RB6, RB7) input high level	VIH1	0.8 VDD			V	Schmitt trigger input
(RB6, RB7) input low level $\overline{MCLR}$ (test mode selection	VIL1	0.2 VDD			V	Schmitt trigger input
RB6, RB7 setup time (before pattern setup time)	tset0	100			ns	
Data in setup time before clock ↓	tset1	100			ns	
Data in hold time after clock ↓	thld1	100			ns	
Data input not driven to next clock input (delay required between command/data or command/command)	tdly1	1.0			µs	
Delay between clock ↓ to clock ↑ of next command or data	tdly2	1.0			µs	
Clock to data out valid (during read data)	tdly3	80			ns	

Note 1: Program must be verified at the minimum and maximum VDD limits for the part.
Note 2: VIHH must be higher than VDD + 4.5V to stay in programming/verify mode.

© 1995 Microchip Technology Inc.

EPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC17C42
- PIC17C43
- PIC17C44

1.0 PROGRAMMING THE PIC17CXX

The PIC17CXX is fundamentally programmed using the TABLWT instruction with the table pointer pointing to an internal EPROM location. Therefore, a user can program an EPROM location while executing code (even from internal EPROM).

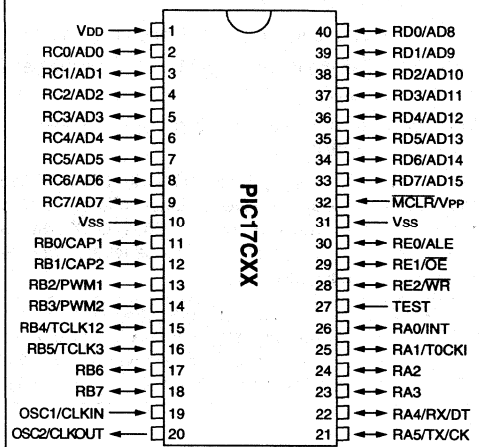
For the convenience of a programmer developer, a "program & verify" routine is provided in the on-chip test program memory space, the program resides in ROM and not EPROM. Therefore, it is not erasable. The "program/verify" routine allows the user to load any address, program a location, verify a location or increment to the next location. It allows variable programming pulse width.

1.1 Hardware Requirements

Since the PIC17CXX under programming is actually executing code from "boot ROM", clock must be provided to the part. Furthermore, the PIC17CXX under programming may have any oscillator configuration (EC, XT, LF or RC). Therefore, the external clock driver must be able to overdrive pulldown in RC mode. CMOS drivers are required since the OSC1 input has a Schmitt trigger input with levels (typically) of 0.2VDD and 0.8VDD. See the PIC17C4X datasheet (DS30412A) for exact specifications.

PIN CONFIGURATIONS

40L PDIP, CERDIP WINDOW PINOUT



PIN DESCRIPTIONS (DURING PROGRAMMING): PIC17C42/43/44

Pin Name	During Programming		
	Pin Name	Pin Type	Pin Description
RA <0:4>	RA <0:4>	I	Necessary in programming mode
TEST	TEST	I	Must be set to "high" to enter programming mode
RB <7:0>	PAD <15:8>	I/O	Address & data: high byte
RC <7:0>	PAD <7:0>	I/O	Address & data: low byte
MCLR/VPP	VPP	P	Programming Power
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = input, O = Output, P = Power

The PIC17CXX requires two programmable power supplies, one for VDD (2.5V to 6.0V recommended) and one for VPP (13 ± 0.5V). Both supplies should have a minimum resolution of 0.25V.

The PIC17CXX uses an intelligent algorithm. The algorithm calls for program verification at VDD (min.) as well as VDD (max.). Verification at VDD (min.) guarantees good "erase margin". Verification at VDD (max) guarantees good "program margin". Three times (3X) additional pulses will increase program margin then beyond VDD (max.) and insure safe operation in user system.

The actual programming must be done with VDD in the VDDP range (4.75 - 5.25V).

VDDP = VDD range required during programming.

VDD min. = minimum operating VDD spec for the part. (2.5V)

VDD max. = maximum operating VCC spec for the part. (6.0V)

Programmers must verify the PIC17CXX at its specified VDD max. and VDD min. levels. Since Microchip may introduce future versions of the PIC17CXX with a broader VDD range, it is best that these levels are user selectable (defaults are ok).

Note:	Any programmer not meeting these requirements may only be classified as "prototype" or "development" programmer but not a "production" quality programmer.

2.0 HOW TO ENTER PROGRAMMING MODE

To execute the programming routine, the user must hold TEST pin high, RA2, RA3 must be low and RA4 must be high (after power-up) while keeping $\overline{MCLR}$ low and then raise $\overline{MCLR}$ pin from VIL to VIHH (VDD or VPP). This will force FFE0h in the program counter and execution will begin at that location (the beginning of the boot code) following reset. Execution is forced to Internal mode by overriding the fuse configuration. The code protect bit is not overwritten. The program immediately polls PORT RB<7:0> to determine a branch address. Presenting E1h on PORT RB will cause the program to jump to and execute the "program/verify" routine.

All unused pins during programming are in high impedance state.

PORTB (RB) has internal weak pull-ups which are active during the programming mode. When TEST pin is high, Power-up timer (PWRT) and Oscillator Start-up Timers (OST)are disabled.

2.1 Program/Verify Mode

The program/verify mode is intended for full-feature programmers. This mode offers the following capabilities:

a) Load any arbitrary 16-bit address to start program and/or verify at that location.

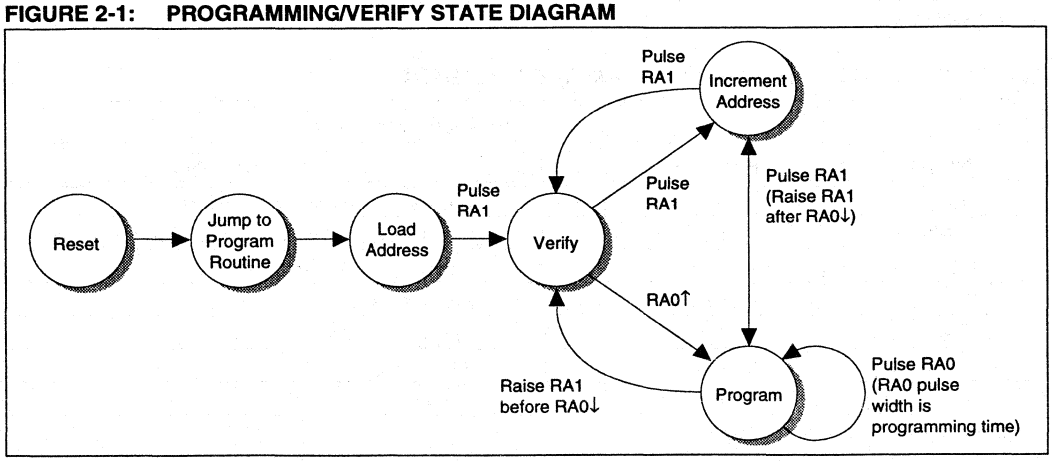
b) Increment address to program/verify the next location.

c) Allows arbitrary length programming pulse width.

d) Following a "verify" allows option to program the same location or increment and verify the next location.

e) Following a "program" allows options to program the same location again, verify the same location or to increment and verify the next location.

FIGURE 2-1: PROGRAMMING/VERIFY STATE DIAGRAM



© 1995 Microchip Technology Inc.

2.1.1 LOADING NEW ADDRESS

The program allows new address to be loaded right out of reset. A 16-bit address is presented on ports RB (high byte) and RC (low byte) and the RA1 is pulsed ($0 \rightarrow 1$, then $1 \rightarrow 0$). The address is latched on the rising edge of RA1. See timing diagrams for details. After loading an address, the program automatically goes into a "verify cycle". To load a new address at any time, the PIC17C4X must be reset and the programming mode re-entered.

2.1.2 VERIFY (OR READ) MODE

"Verify mode" can be entered from "Load address" mode, "program mode" or "verify mode". In verify mode pulsing RA1 will turn on PORTS RB and RC output drivers and output the 16-bit value from the current location. Pulsing RA1 again will increment location count and be ready for the next verify cycle. Pulsing RA0 will begin a program cycle.

2.1.3 PROGRAM CYCLE

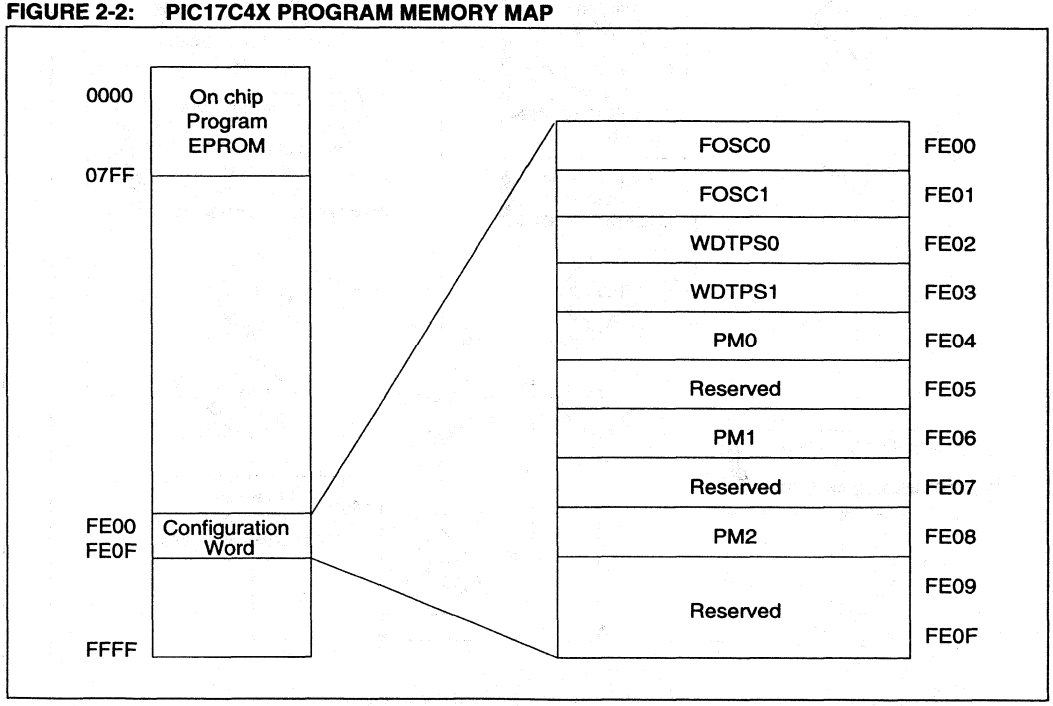
"Program cycle" is entered from "verify cycle" or program cycle" itself. After a verify, pulsing RA0 will begin a program cycle. 16-bit data must be presented on PORTS RB (high byte) and RC (low byte) before RA0 is raised.

The data is sampled 3 TCY after the rising edge of RA0. Programming continues for the duration of RA0 pulse.

At the end of programming the user can choose one of three different routes. If RA1 is kept low and RA0 is pulsed again, the same location will be programmed again. This is useful for applying over programming pulses. If RA1 is raised before RA0 falling edge, then a verify cycle is started without address increment. Raising RA1 after RA0 goes low will increment address and begin verify cycle on the next address.

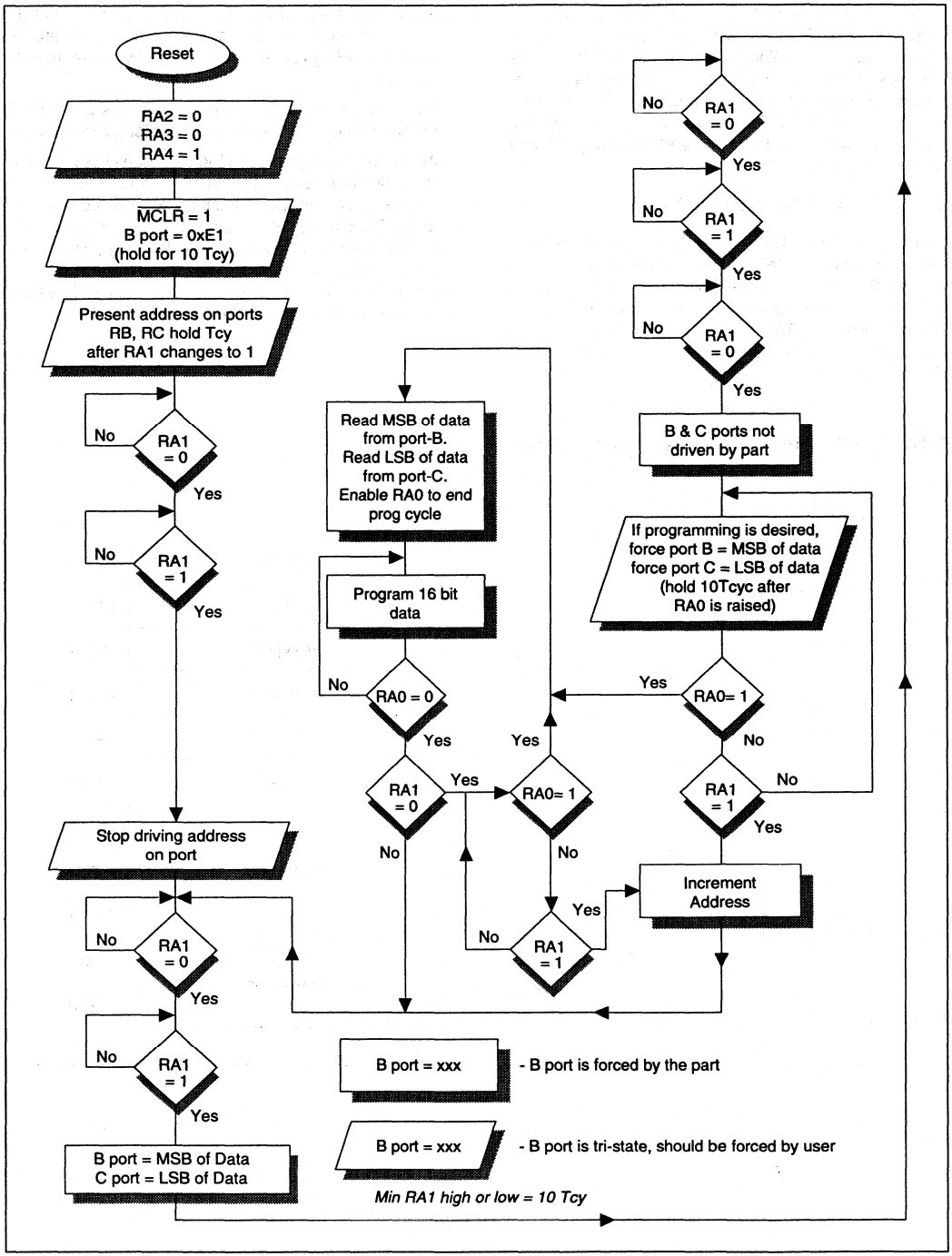
3

FIGURE 2-2: PIC17C4X PROGRAM MEMORY MAP



3.0 PROGRAMMING SPECIFICATIONS

FIGURE 3-1: PROGRAMMING ROUTINE FLOWCHART



© 1995 Microchip Technology Inc.

FIGURE 3-2: PROGRAMMING AND VERIFY TIMINGS I

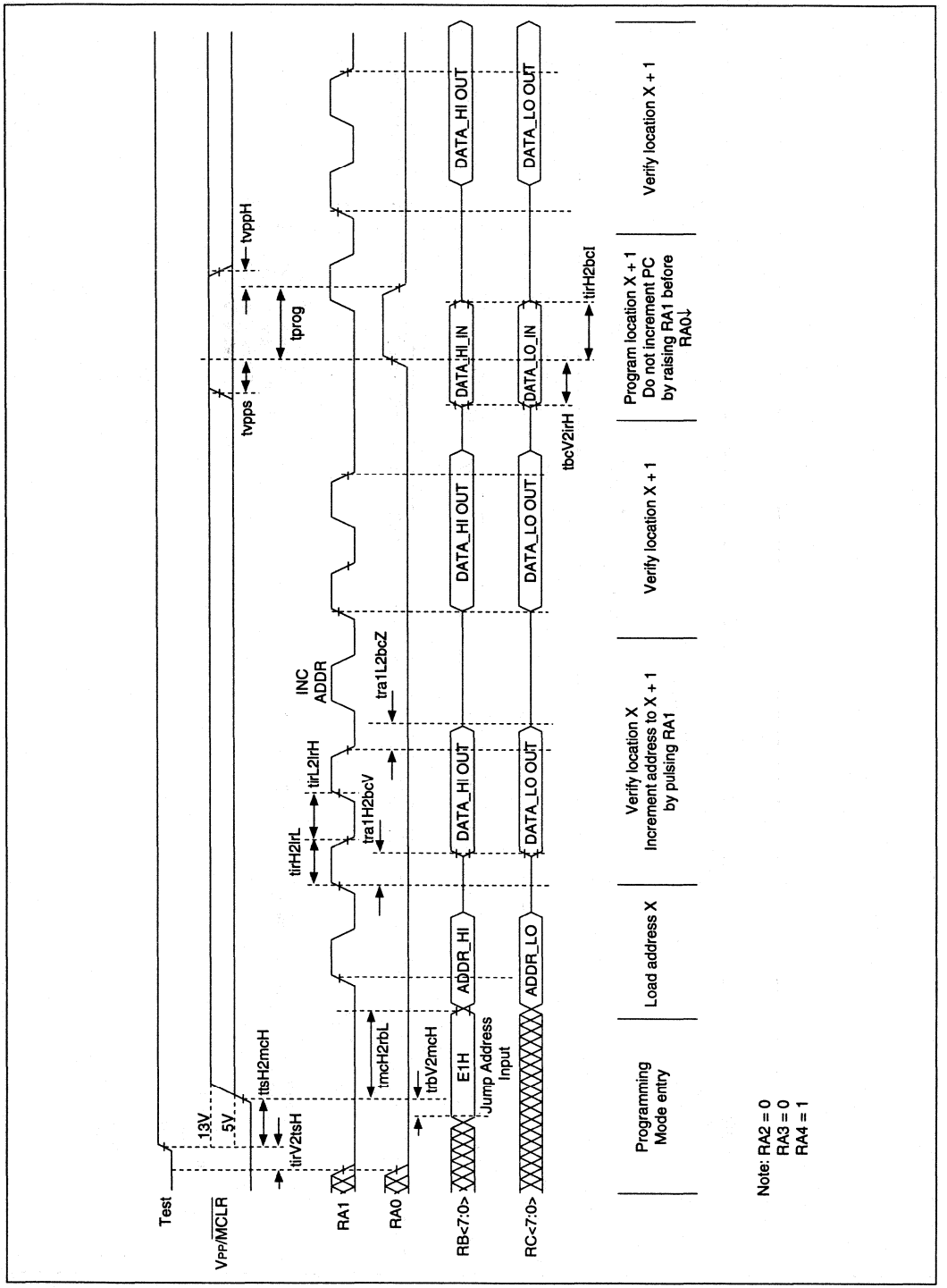
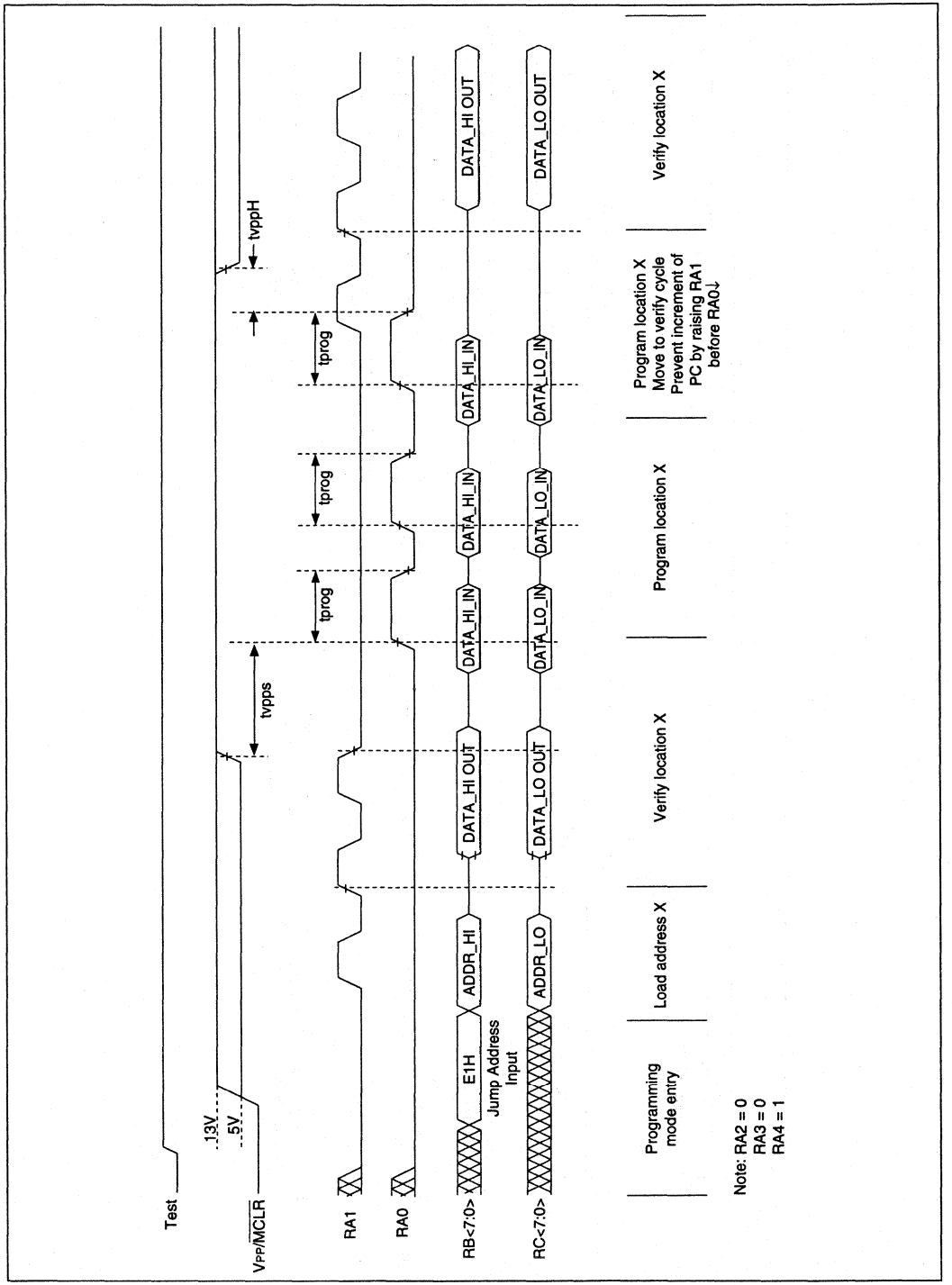


FIGURE 3-3: PROGRAMMING AND VERIFY TIMINGS II



© 1995 Microchip Technology Inc.

FIGURE 3-4: PROGRAMMING AND VERIFY TIMINGS III

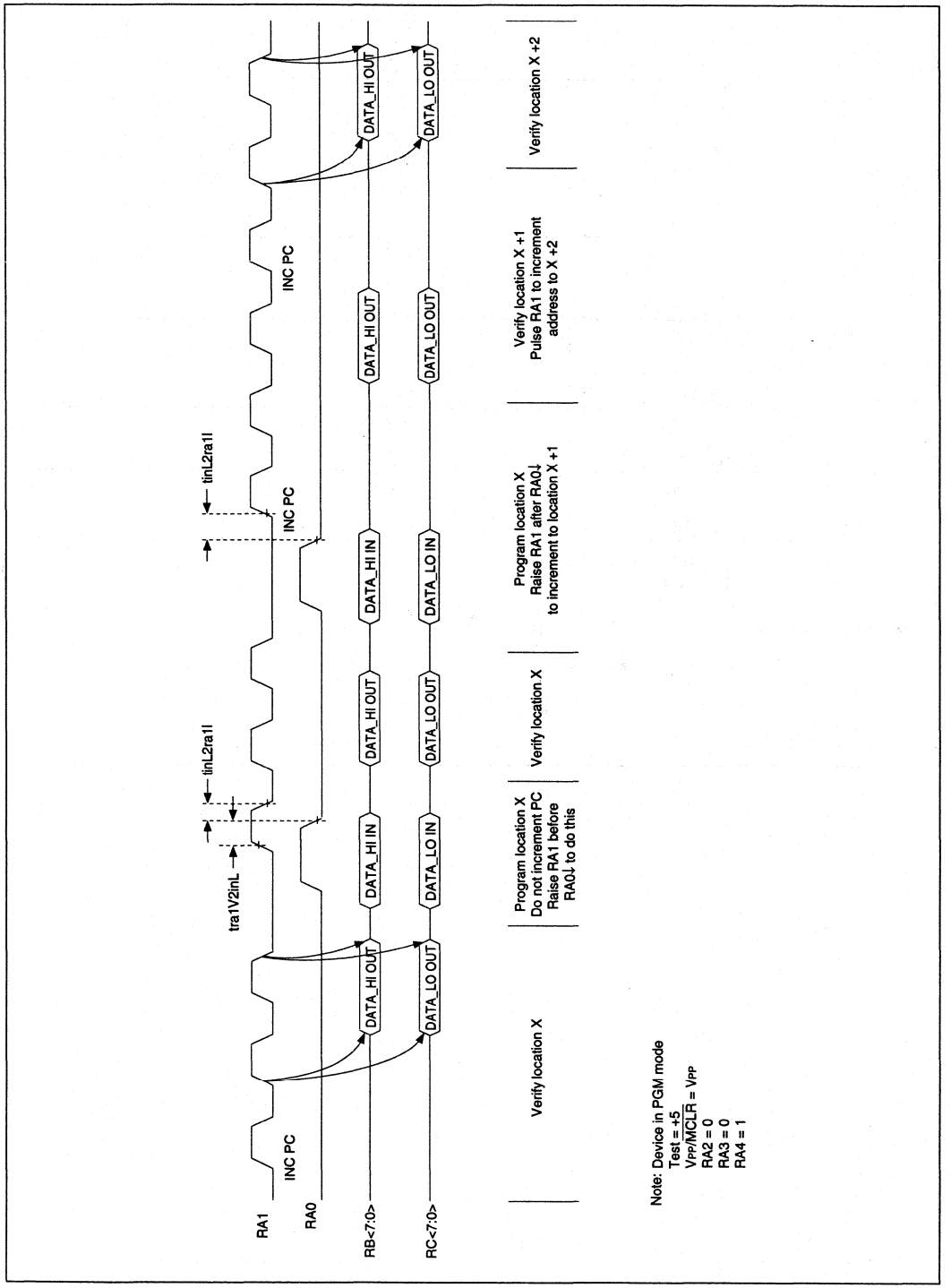
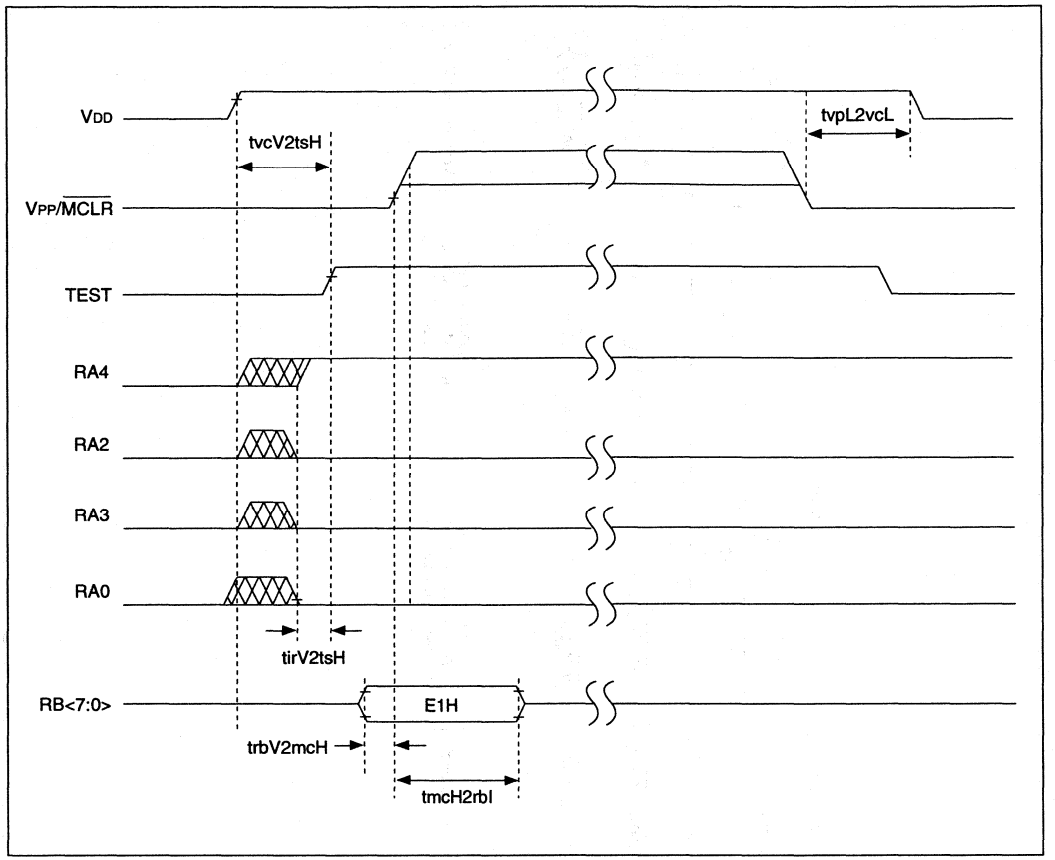
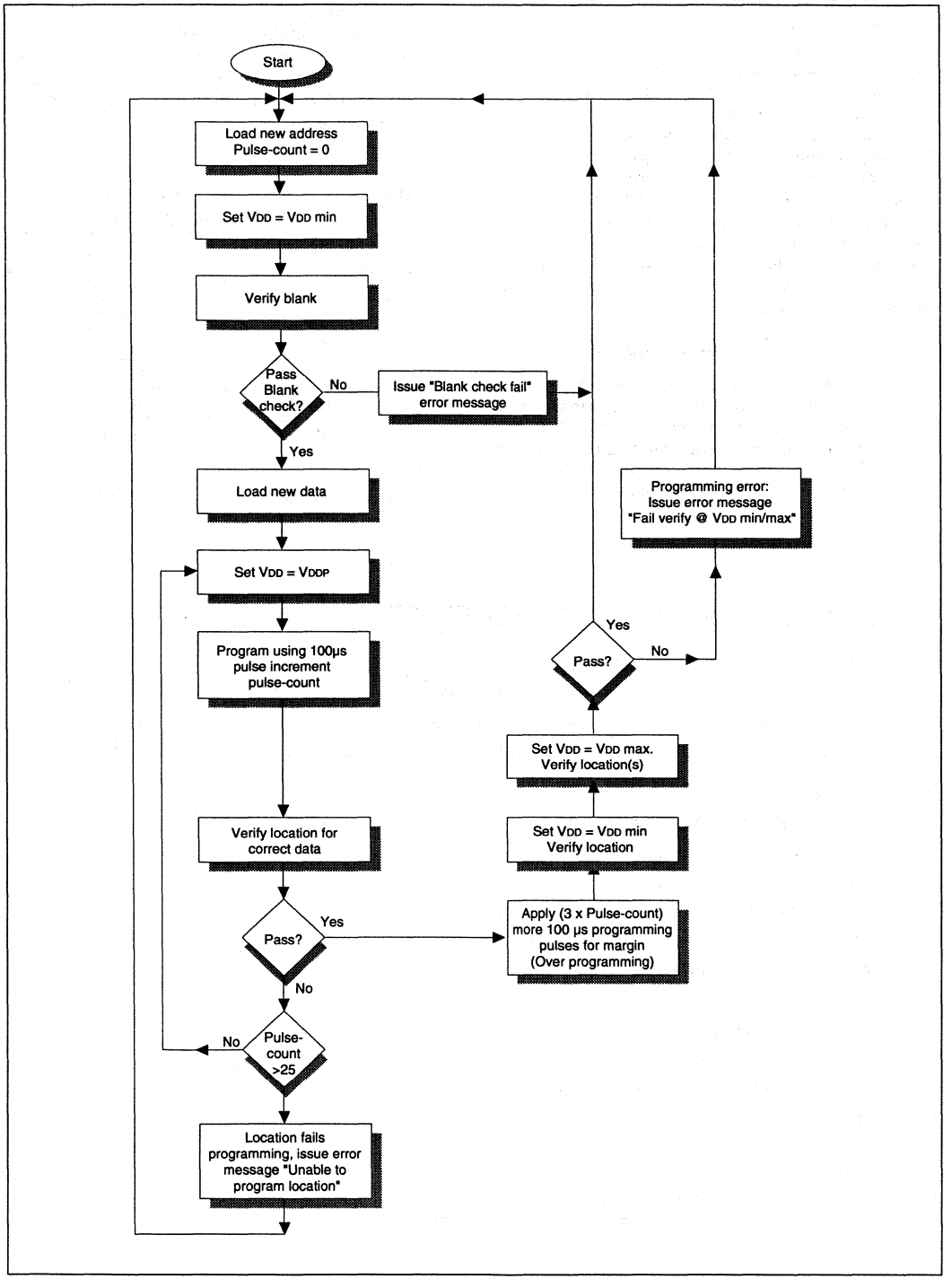


FIGURE 3-5: POWER-UP/DOWN SEQUENCE FOR PROGRAMMING



© 1995 Microchip Technology Inc.

FIGURE 3-6: RECOMMENDED PROGRAMMING ALGORITHM



4.0 CONFIGURATION WORD

Configuration bits are mapped into program memory. Each bit is assigned one memory location. In erased condition a bit will read as '1'. To program a bit, the user needs to write to the memory address. The data is immaterial; the very act of writing will program the bit. The configuration word locations are shown in Table 4-1. **The programmer should not program the reserved locations to avoid unpredictable results and to be compatible with future variations of the PIC17C4X. It is also mandatory that configuration locations are programmed in the strict order starting from the first location (0xFE00) and ending with the last (0xFE08). Unpredictable results may occur if the sequence is violated.**

4.1 Reading Configuration Word

The PIC17CXX has seven configuration locations (see Table 4-1). These locations can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. Any write to a configuration location, regardless of the data, will program that configuration bit. Reading any configuration location between 0xFE00 and 0xFE07 will place the low byte of the configuration word (see Table 4-2) into PAD<7:0> (PORTC). PAD<15:8> (PORTB) will be set to 0xFF. Reading a configuration location between 0xFE08 and 0xFE0F will place the high byte of the configuration word into PAD<7:0> (PORTC). PAD<15:8> (PORTB) will be set to 0xFF.

TABLE 4-1: CONFIGURATION BIT PROGRAMMING LOCATIONS

Bit	Address
FOSC0	0xFE00
FOSC1	0xFE01
WDTPS0	0xFE02
WDTPS1	0xFE03
PM0	0xFE04
PM1	0xFE06
PM2†	0xFE08

†This location does not exist on the PIC17C42.

TABLE 4-2: READ MAPPING OF CONFIGURATION BITS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	—	PM1	—	PM0	WDTPS1	WDTPS0	FOSC1	FOSC0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	—	—	—	—	—	—	—	PM2

—=Unused

PM<2:0>, Processor Mode Select bits

111=Microprocessor Mode

110=Extended microcontroller mode

101 =Microcontroller mode

000 =Code protected microcontroller mode

WDTPS<1:0>, WDT Prescaler Select bits.

11 =WDT enabled, postscaler = 0

10 =WDT enabled, postscaler = 256

01=WDT enabled, postscaler = 64

00=WDT disabled, 16-bit overflow timer

FOSC<1:0>, Oscillator Select bits

11=EC oscillator

10=XT oscillator

† This bit does not exist on PIC17C42.

© 1995 Microchip Technology Inc.

4.2 Embedding Configuration Word Information in the Hex File

To allow portability of code, a PIC17C4X programmer is required to read the configuration word locations from the hex file when loading the hex file. If configuration word information was not present in the hex file then a simple warning message may be issued. Similarly, while saving a hex file, all configuration word information must be included. An option to not include the configuration word information may be provided. When embedding configuration word information in the hex file, it should be to address FE00h.

Microchip Technology Inc. feels strongly that this feature is important for the benefit of the end customer.

TABLE 4-3:

Device	Code Protect	Checksum*	Blank Value	0xC0DE at 0 and max address
PIC17C42	MP mode	SUM[0x000:0x7FF] + CFGW & 0xFFFF	0xF7FF	0x79BD
	MC mode	SUM[0x000:0x7FF] + CFGW & 0xFFFF	0xF7EF	0x79AD
	EMC mode	SUM[0x000:0x7FF] + CFGW & 0xFFFF	0xF7BF	0x797D
	PMC mode	SUM_XNOR8[0x000:0x7FF] + CFGW & 0xFFFF	0xF7AF	0xF723
PIC17C43	MP mode	SUM[0x000:0xFFF] + CFGW • 0x015F	0xF15F	0x731D
	MC mode	SUM[0x000:0xFFF] + CFGW • 0x015F	0xF14F	0x730D
	EMC mode	SUM[0x000:0xFFF] + CFGW • 0x015F	0xF11F	0x72DD
	PMC mode	SUM_XNOR8[0x000:0xFFF] + CFGW • 0x015F	0xF00F	0xB3D3
PIC17C44	MP mode	SUM[0x000:0x1FFF] + CFGW • 0x015F	0xE15F	0x631D
	MC mode	SUM[0x000:0x1FFF] + CFGW • 0x015F	0xE14F	0x630D
	EMC mode	SUM[0x000:0x1FFF] + CFGW • 0x015F	0xE11F	0x62DD
	PMC mode	SUM_XNOR8[0x000:0x1FFF] + CFGW • 0x015F	0xE00F	0xA3D3

Legend: CFGW = Configuration Word
SUM[a:b] = [Sum of locations a to b inclusive]
SUM_XNOR8(a:b) = [Sum of 8-bit wide XNOR copied into upper and lower byte, of locations a to b inclusive]
*Checksum = [Sum of all the individual expressions] **MODULO** [0xFFFF]

TABLE 4-4: CONFIGURATION WORD

PIC17C42

To code protect:

- Protect all memory `xxxxxxx0x0x0xxxx`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFE00)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read Scrambled, Write Disabled*	Read Unscrambled, Write Enabled

PIC17C43

To code protect:

- Protect all memory `xxxxxxx0x0x0xxxx`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFE00)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read Scrambled, Write Disabled*	Read Unscrambled, Write Enabled

PIC17C44

To code protect:

- Protect all memory `xxxxxxx0x0x0xxxx`

Program Memory Segment	R/W in Protected Mode	R/W in Unprotected Mode
Configuration Word (0xFE00)	Read Scrambled, Write Enabled	Read Unscrambled, Write Enabled
All memory	Read Scrambled, Write Disabled*	Read Unscrambled, Write Enabled

Legend: X = Don't care

*Write to on-chip EPROM memory is disabled. The only way these locations can be programmed is if a TABLWT instruction is issued from an "on-chip" program memory space to program an on-chip memory location.

© 1995 Microchip Technology Inc.

5.0 AC/DC SPECIFICATIONS FOR PROGRAMMING

	Standard Operating Conditions Operating Temperature +10°C ≤ TA ≤ +70°C, unless otherwise stated Operating Voltage 4.75V ≤ VDD ≤ 5.25V, unless otherwise stated.					
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Programming voltage on VPP/$\overline{MCLR}$ pin	VPP	12.5		13.5	V	Note 1
Programming current on VPP/$\overline{MCLR}$ pin	IPP		25	50	mA	Note 3
Supply voltage during programming	VDDP	4.75	5.0	5.25	V	
Osc/clockin frequency during programming	FOSCP	4		10	MHz	
Instruction cycle	TCY	1		0.4	µs	TCY = 4/FOSCP
Supply current during programming	IDDP			30	mA	Freq = 10MHz, VDD = 5.5V Note 3
Supply voltage during verify	VDDV	VDD min.		VDD max.	V	Note 2
RA0, RA1, RA2, RA3, RA4 setup before TEST↑	tirV2tsH	1			µs	
TEST↑ to $\overline{MCLR}$↑	ttsH2mcH	1			µs	
RC<7:0>, RB<7:0> valid to RA1 or RA0↑ :Address/Data input setup time	tbcV2irH	0			µs	
RA1 or RA0↑ to RB<7:0>, RC<7:0> invalid ; Address data hold time;	tirH2bcI	10 TCY			µs	
RT↓ to RB<7:0>, RC<7:0> high impedance	t0ckiL2rbcZ			8 TCY		
RA1↑ to data out valid	t0ckiH2bcV			10 TCY		
Programming pulse width	tprog	10	100	1000	µs	
RA0, RA1 high pulse width	tirH2irL	10 TCY			µs	
RA0, RA1 low pulse width	tirL2irH	10 TCY			µs	
RA1↑ before INT↓ (to go from prog cycle to verify w/o increment)	t0ckiV2inL	0			µs	
RA1 valid after RA0 (to select increment or no increment going from program to verify cycle	tinL2rtI	10 TCY			µs	
VPP setup time before RA0↑	tvpps	100			µs	Note 1
VPP hold time after INT↓	tvpph	0			µs	Note 1
VDD stable to TEST↑	tvdV2tsH	10			ms	
RB input (E1h) valid to VPP/$\overline{MCLR}$↑	trbV2mcH	0			µs	
RB input (E1h) hold after VPP/$\overline{MCLR}$↑	tmcH2rbI	10 TCY			ns	
VDD power down after VPP power down	tvpL2vdL	10			ms	

Note 1: VPP/$\overline{MCLR}$ pin can be kept at VPP level (12.5V - 13.5V) at times other than programming.
Note 2: Program must be verified at the minimum and maximum VDD limits for the part.
Note 3: These parameters are for design guidance only and are not tested nor characterized.

© 1995 Microchip Technology Inc.

SQTP™ Specification for PIC16C5X/5XA

Serialized Quick Turn Programming Specification for PIC16C5X/5XA

OVERVIEW

Serialization is a method of programming PIC16C5X and PIC16C5XA microcontrollers whereby each chip is programmed with a slightly different code. Typically, all locations are programmed with the same basic code except for a few continuous bytes which are programmed with a different number (referred to as 'key' or 'ID number' or 'serial number') in each member. Typical applications of such programming are remote transmitters for car alarms or garage door openers where each unit must have a different access code.

Microchip offers a flexible SQTP program, whereby a customer can simply specify the nature of serialization. The 'serial number' generation and programming will be taken care of by the factory.

1.0 DEVICES CURRENTLY SUPPORTED

Other device types are being added. Please consult a Microchip representative or Microchip sales person.

Device	Oscillator Type	Package
PIC16C54	XT, RC, LP, HS	PDIP, SOIC, SSOP
PIC16C55	XT, RC, LP, HS	PDIP, SOIC, SSOP
PIC16C56	XT, RC, LP, HS	PDIP, SOIC, SSOP
PIC16C57	XT, RC, LP, HS	PDIP, SOIC, SSOP
PIC16C54A	XT, RC, LP, HS	PDIP, SOIC, SSOP
PIC16C58A	XT, RC, LP, HS	PDIP, SOIC, SSOP

2.0 SERIALIZATION SCHEME SUPPORTED

2.1 Locations:

The serial number must reside in continuous locations with up to eight locations used. Furthermore these locations must be coded as 8NN (RETLW NN, where NN=8-bit random code) in the finished product. For details on how the RETLW instruction is typically used for serialization purposes, please see Appendix A. The customer code must be supplied without the serial code in these locations. These locations must be 8FFh in the customer code provided to Microchip. Microchip will insert the serial code at these locations during programming. Hex files must be in Intel hex 8-bit merged format. See Appendix B for details.

2.2 Numbering Schemes:

Random: Truly random numbers are generated. However, there is no guarantee that the numbers will be non-repeating although the probability of such an occurrence will be infinitesimally small for a reasonably large field.

Pseudo-Random*: Pseudo-random sequences of requested length (e.g. 32-bit long if four locations are used) starting with a 'seed value' selected by the factory. The customer may optionally specify the starting value.

Pseudo-random sequences, by definition are non-repeating. See Appendix C for polynomials used to generate the numbers.

Sequential: Sequential numbers are generated. User specifies the "starting number" and an increment value. In sequential numbering, the least significant digit is in the lowest memory location. The increment value must be between 1 and 255.

Numbers are always in hex and not in BCD or any other format.

© 1995 Microchip Technology Inc.

3.0 PROGRAMMING SEQUENCE

The factory will program the "basic code" first, then program the serial number and finally program the code-protection fuse. Program memory will be verified at each stage except after code protection. Optionally, the factory may choose to program the "basic code" and the "serial number" at the same time. The customer may specify an ID number (four hex digits) to be programmed in the ID locations or elect to leave them unprogrammed.

4.0 SAMPLES

Three (3) verification samples will be provided. These will be programmed with factory selected random or sequential codes in the serialization locations. The three parts will be programmed with three different serial codes. If order entry has been completed, then the samples will reflect the first three codes. If code protection is requested, then one of the three samples will be code protected.

5.0 THE FOLLOWING LIMITATIONS APPLY TO THE SQTP PROGRAM

1. During shipment of serialized parts, no particular sequence can be guaranteed.
2. In sequential or pseudo-random numbering scheme, there may be missing serial numbers (e.g. due to QC sampling).
3. A list of serial numbers programmed can not be provided, nor will such a list be generated or maintained by Microchip.
4. For sequential and pseudo-random numbering schemes, Microchip will maintain last number used in last shipment and use the next number as the starting number for the next shipment. The customer should be prepared to provide a "new starting number" in the event the flow is disrupted due to unforeseen events.

© 1995 Microchip Technology Inc.

APPENDIX A:

Implementing a table in the program memory of PIC16C5X and PIC16C5XA:

The PIC16C5X and the PIC16C5XA family uses Harvard architecture, in which the program memory is separate from data memory. All instructions operate on data that is fetched from the register file or data memory. Since there are no instructions to read from or write to the program memory, simply storing data words in program memory is of no use. There is, however, a simple and elegant way to implement constant tables in the program memory by using the RETLW instruction. This instruction returns from a subroutine as well as loads an 8-bit constant into the W register. The following example shows how to get a byte of "serial information" from the table stored at location 000h in PIC16C54:

```
            ORG    0        ;store serial numbers
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh
            RETLW  0FFh      ;end of serial
                             ;numbers
     •
     •
main_prog   ORG    XYZ      ;This is main program
     •
     •
     MOVLW  byte_num        ;byte_num = 0 for 1st
                            ;byte
     CALL   get_1byte;
     •
     •
get_1byte   MOVWF  PC       ;write W to program
                            ;counter
                            ;W = offset = 0 for
                            ;1st byte
                            ;end of get_1byte sub
                            ;routine
     •
     •
     •
     END
```

The next example shows how a serial number may reside at location other than 000h.

```
main_prog ORG    XYZ    ;This is main program
     •
     •
     MOVLW byte_num      ;byte_num = 0 for 1st
                         ;byte
     CALL  get_1byte;
     •
     •
get_1byte ADDWFPC       ;W = offset
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;
          RETLW  0ffh ;end of serial
                      ;numbers
     •
     •
     END
```


© 1995 Microchip Technology Inc.

APPENDIX B:

Standard hex file format for serial programming:

The hex file containing the 'serial numbers' will be in Intel hex 8-bit format. Since the PIC16C5X and the PIC16C5XA have 12-bit data words, all addresses are doubled in this hex format. Each line of the hex file will be for a new part. Each line can contain only up to 16 bytes (i.e. eight PIC16C5X, PIC16C5XA instruction words). The format is as follows:

```
:NNAAAATTHHHHHH.....HHCC
```

where:

NN = byte count on current line (max 10h allowed)

AAAA = address in four hex digits

TT = record type, always 00 except 01 for EOF

HH = Two digit hex data byte

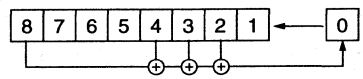
CC = Two digit hex checksum

APPENDIX C:

Pseudo-random numbers are generated using modulo-2 primitive polynomials. This method guarantees to produce a sequence of maximal length, i.e., cycle through all possible sequence of n bits before it repeats. By providing a seed value as the initial bit pattern (the only combination not used is all 0's), one can get 2n-1 random bits before the sequence repeats itself. Microchip will only support pseudo-random serial numbers for bit lengths 8, 16, 24, 32, 40, 48, 56 and 64 (i.e., 1-8 locations). The polynomials used are:

8 bit:	x8 + x4 + x3 + x2 + 1
16 bit:	x16 + x5 + x3 + x2 + 1
24 bit:	x24 + x4 + x3 + x + 1
32 bit:	x32 + x7 + x5 + x3 + x2 + x + 1
40 bit:	x40 + x5 + x4 + x3 + 1
48 bit:	x48 + x7 + x5 + x4 + x2 + x + 1
56 bit:	x56 + x7 + x4 + x2 + 1
64 bit:	x64 + x4 + x3 + x + 1

To implement the 8-bit polynomial requires XORing the non-zero bits of the polynomial (shown as a shift register below) and shift on the resetting bit back into the shift register.



© 1995 Microchip Technology Inc.

SECTION 4
LOGIC PRODUCTS

4

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

MICROCHIP

AY0438

32-Segment CMOS LCD Driver

FEATURES

- Drives up to 32 LCD segments of arbitrary configuration
- CMOS process for: wide supply voltage range, low- power operation, high-noise immunity, wide temperature range
- CMOS and TTL-compatible inputs
- Electrostatic discharge protection on all pins
- Cascadable
- On-chip oscillator
- Requires only three control lines

APPLICATIONS

- Industrial displays
- Consumer product displays
- Telecom product displays
- Automotive dashboard displays

DESCRIPTION

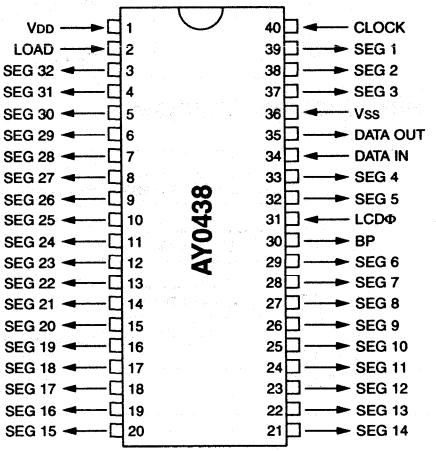
The AY0438 is a CMOS integrated device that drives a liquid crystal display, usually under microprocessor control. The part acts as a smart peripheral that drives up to 32 LCD segments. It needs only three control lines due to its serial input construction. It latches the data to be displayed and relieves the microprocessor from the task of generating the required waveforms.

The AY0438 can drive any standard or custom parallel drive LCD display, whether it be field effect or dynamic scattering; 7-, 9-, 14- or 16-segment characters; decimals; leading + or -; or special symbols. Several AY0438 devices can be cascaded. The AC frequency of the LCD waveforms can either be supplied by the user or generated by attaching a capacitor to the LCD input, which controls the frequency of an internal oscillator.

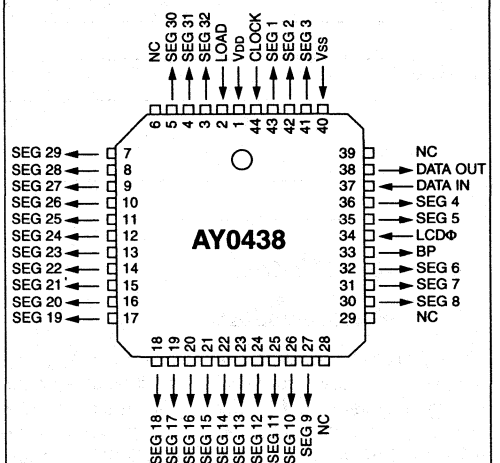
The AY0438 is available in 40-lead dual in-line plastic and 44-lead PLCC packages. Unpackaged dice are also available.

PIN CONFIGURATION

40-Lead Dual In-line



44 PLCC



4

© 1995 Microchip Technology Inc.

FIGURE 1: PIN DESCRIPTIONS

Pin # (PDIP Only)	Name	Direction	Description
1	VDD	-	Supply voltage
2	Load	Input	Latch data from registers
3-29, 32, 33, 37-39	Seg 1-32	Output	Direct drive outputs
30	BP	Output	Backplane drive output
31	LCDΦ	Input	Backplane drive input
34	Data In	Input	Data input to shift register
35	Data Out	Output	Data output from shift register
36	Vss	Ground	Ground
40	Clock	Input	System clock input

FIGURE 2: BLOCK DIAGRAM

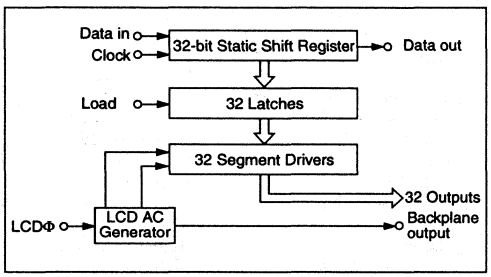


FIGURE 3: BACKPLANE AND SEGMENT OUTPUT

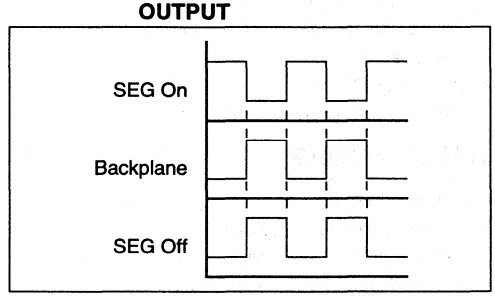
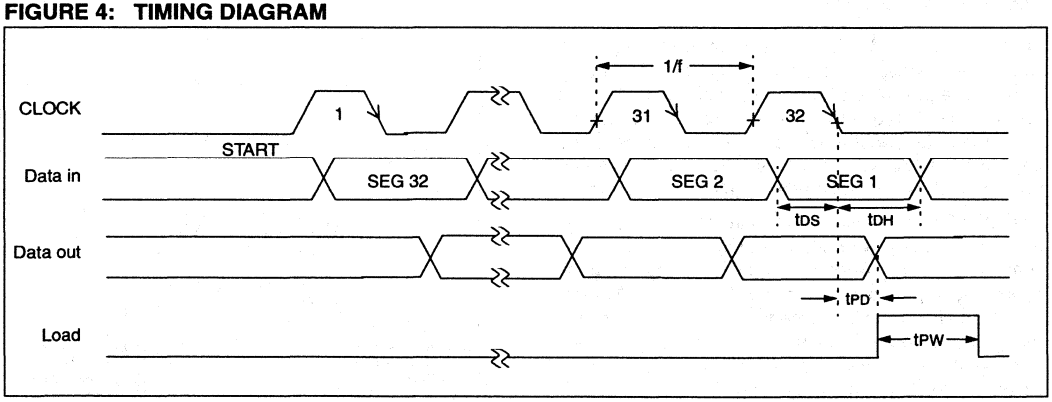


FIGURE 4: TIMING DIAGRAM



1.0 OPERATION:

1.1 Data In and Clock

The shift register shifts and outputs on the falling edge of the clock. Every clock falling edge does a logical left shift. As an example, if 32 clock pulses are supplied as in Figure 4, then the data input at the first clock will output at SEG 32, and the last data input (# 32) will output at SEG 1 when a LOAD signal is enabled (Figure 2). It is recommended that a complete 32 bit transfer be done every time the outputs are updated. A logic 1 at the Data In causes the corresponding segment to be enabled or visible, i.e. the output at Segment Output is 180° out-of-phase with the Backplane output (Figure 3).

1.2 Load

A logic 1 at the Load input (Figure 2) causes the parallel load of the data in the shift register into the latches that control the segment drivers. If the Load signal is tied high, then the latches become transparent and the segment drivers are always connected to the shift registers.

© 1995 Microchip Technology Inc.

1.3 LCDϕ

LCDϕ can be driven by an external signal or by connecting a capacitor between LCDϕ and ground (GND), which will enable the on-chip oscillator required to generate the backplane output voltage. Figure 5 shows the relationship between capacitance value and output frequency. Leaving the LCDϕ input unconnected is not recommended. When driven by an external clock, the backplane output is in phase with the input clock. When cascading two AY0438 devices (Figure 6 and Figure 7), the backplane output can be generated using a capacitor to GND on the first AY0438. This backplane output can then be connected to the LCDϕ input of the second AY0438. The backplane output of the second device is then used to drive the backplane of the LCD module.

FIGURE 5: OSCILLATOR FREQUENCY GRAPH (TYPICAL @ 25°C)

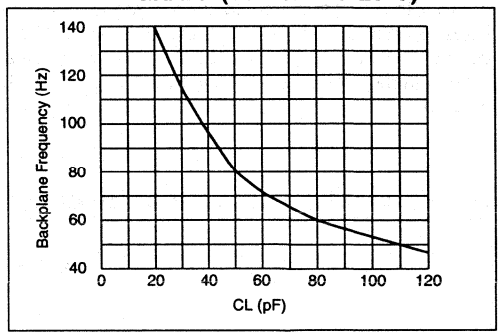


FIGURE 6: CASCADING TWO AY0438 DEVICES

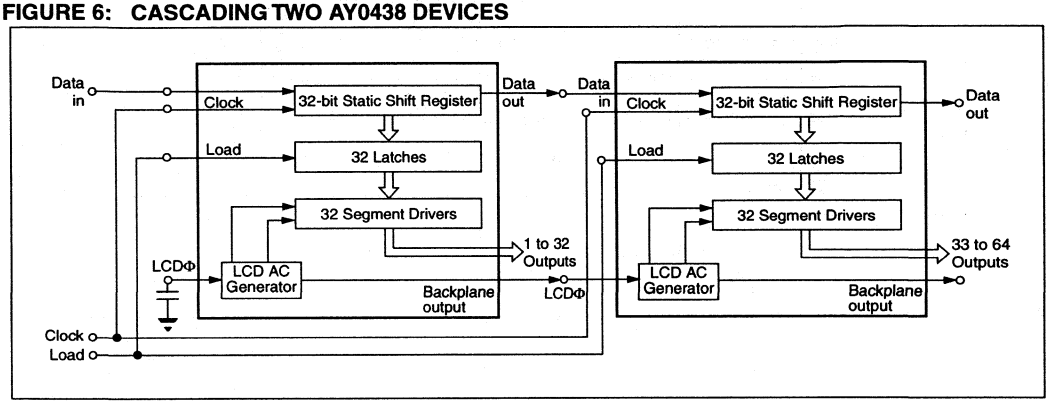
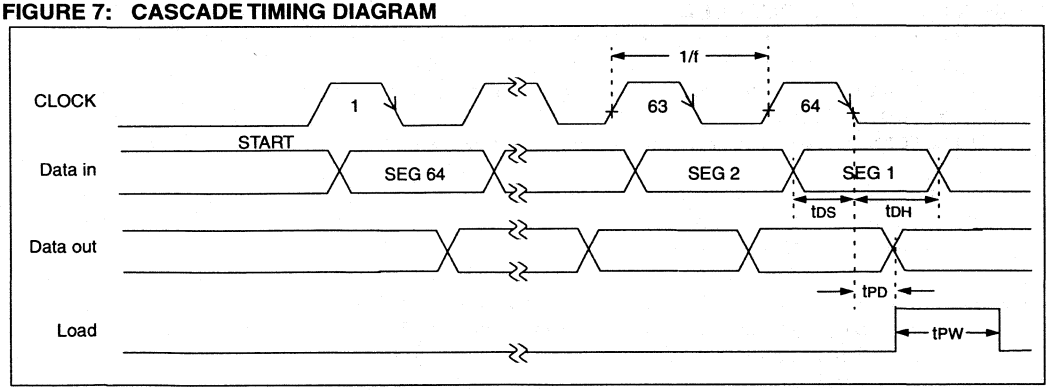


FIGURE 7: CASCADE TIMING DIAGRAM



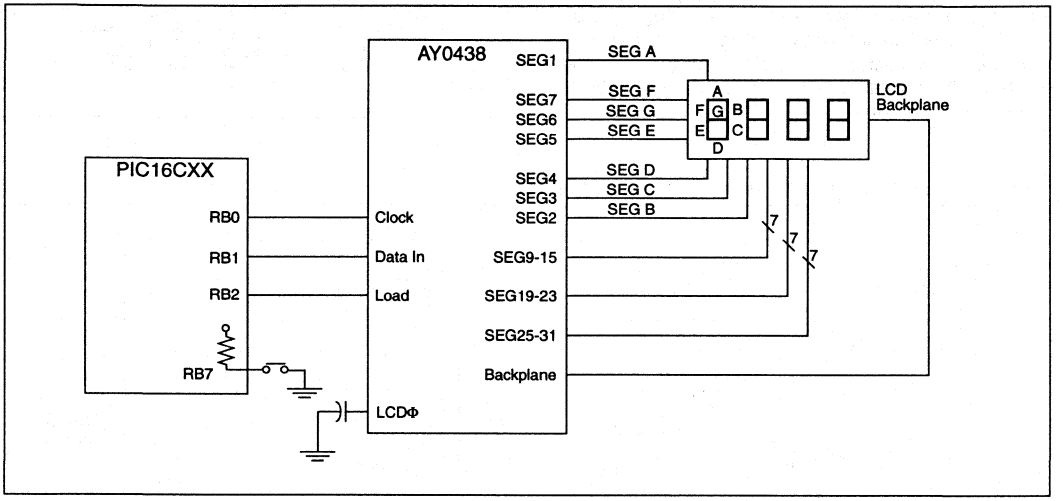
1.4 General

In order to avoid any race conditions, the Data In and Load signals should not be changed during a falling edge of the Clock. Figure 4 and Figure 7 show a typical timing diagram for a 32 segment and 64 segment LCD module.

1.5 Interfacing to a LCD Module and PIC16CXX Device

Figure 8 shows a typical layout of an AY0438 connected to a LCD module and interfaced to a PIC16CXX family device. Example 1 lists code used to program the PIC16CXX device. This code was complied using MPASM.

FIGURE 8: INTERFACING TO A LCD MODULE AND PIC16CXX DEVICE



EXAMPLE 1: EXAMPLE CODE

```
;*************************************************************************
;This program shows an interface between a PIC16CXX device
;and the AY0438 LCD controller to control a 7 Segment
;4 digit LCD module.
;The PIC16CXX interface to the AY0438 Hardware:
;
;       PORTB bit 0 --> CLK
;       PORTB bit 1 --> DATA IN
;       PORTB bit 2 --> LOAD
;
;The LCD module is connected to the AY0438 as follows:
;       Most Significant digit  --> seg1 to seg7
;       3rd Significant digit   --> seg9 to seg15
;       2nd Significant digit   --> seg17 to seg 23
;       Least Significant digit --> seg25 to seg 31
;
```


© 1995 Microchip Technology Inc.

```
;The DP are not connected, but can be connected to seg8, 16, 24 & 32.
;For each digit, the segments are connected as:
;       Seg A --> seg(8*n + 1)
;       Seg B --> seg(8*n + 2)
;       Seg C --> seg(8*n + 3)
;       Seg D --> seg(8*n + 4)
;       Seg E --> seg(8*n + 5)
;       Seg F --> seg(8*n + 6)
;       Seg G --> seg(8*n + 7)
;where n = 0, 1, 2 and 3 for MSD, 3rdSD, 2ndSD and LSD respectively.
;The firmware uses the values in registers:
;       MSD, THRDSD, SCNDSD and LSD   to determine the values to be
;pulsed to the AY0438.
;In this example, a pushbutton connected to PORTB bit 7
;is checked periodically to see if it has been pressed. If so,
;the LCD values in locations MSD to LSD are updated.
;*****************************************************************************
        list p=16c71,f=inhx8m
;
;
MSD     equ     0x20
THRDSD  equ     0x21
SCNDSD  equ     0x22
LSD     equ     0x23
count   equ     0x24
temp    equ     0x25
PORTB   equ     0x06
#define CLK     PORTB,0
#define DATAIN  PORTB,1
#define LOAD    PORTB,2
#define UPDATELCD PORTB,7
w       equ     0
STATUS  equ     0x03
C       equ     0
RP0     equ     5
OPTION  equ     0x81
RBPU    equ     7
PCL     equ     0x02
PCLATH  equ     0x0A
;
;
        org     0
        goto    start
        org     0x10
;
;This DecodeValue table must reside in page 0 for this program to work
;
DecodeValue
        addwf   PCL
        retlw   B'00111111'     ;decode for 0
        retlw   B'00000110'     ;decode for 1
        retlw   B'01011011'     ;decode for 2
        retlw   B'01001111'     ;decode for 3
        retlw   B'01100110'     ;decode for 4
        retlw   B'01101101'     ;decode for 5
```

```
        retlw   B'01111101'     ;decode for 6
        retlw   B'00000111'     ;decode for 7
        retlw   B'01111111'     ;decode for 8
        retlw   B'01101111'     ;decode for 9
;
;
start
        clrf    PORTB
        bsf     STATUS,RP0      ;set portb 0,1&2 as outputs
        movlw   B'11111000'     ;       /
        movwf   PORTB           ;       /
        bcf     OPTION,RBPU     ;enable pull-up for switch
        bcf     STATUS,RP0
wait
        btfsc   UPDATELCD       ;see if update switch is low
        goto    wait            ;no then wait
        bcf     LOAD            ;make sure load is disabled
        movf    LSD,w           ;get least significant value
        clrf    PCLATH          ;PCH = 0
        call    DecodeValue     ;decode the value
        call    Send8           ;serially output the seg values
        movf    SCNDSD,w        ;get 2nd significant digit
        call    DecodeValue     ;decode it
        call    Send8           ;serially output it
        movf    THRDSD,w        ;get 3rd significant digit
        call    DecodeValue     ;decode it
        call    Send8
        movf    MSD,w           ;get Most significant value
        call    DecodeValue     ;decode it
        call    Send8           ;serially send it
        bsf     LOAD            ;toggle the LOAD line
        bcf     LOAD            ;to enable the latches
KeyReleased
        btfss   UPDATELCD       ;wait for key to be released
        goto    KeyReleased
        goto    wait            ;repeat loop.
;
;Send8, sends the 8 bits in the W register
Send8
        movwf   temp            ;save in temp
        movlw   .8              ;init count
        movwf   count           ;to 8
sendloop
        bcf     DATAIN          ;make sure DATAIN is low
        rrf     temp            ;rotate value through carry
        btfsc   STATUS,C        ;if bit clear then skip
        bsf     DATAIN          ;else set data bit
        bsf     CLK             ;toggle clock
        bcf     CLK             ;       /
        decfsz  count           ;see if 8 done
        goto    sendloop        ;no then do all
        return                  ;else return

        end
```


© 1995 Microchip Technology Inc.

2.0 ELECTRICAL CHARACTERISTICS

Maximum Ratings*

VDD ..... -0.3V to +12V

Inputs (CLK, Data In, Load) ..... VCC to VDD +0.3V

LCDΦ Input ..... -0.3V to VDD +0.3V

Power Dissipation ..... 250 mW

Storage Temperature ..... -65°C to +125°C

Operating Temperature Industrial ..... -40°C to +85°C

* Exceeding these ratings could cause permanent damage to the device. This is a stress rating only and functional operation of this device at these conditions is not implied. Operating ranges are specified in Standard Conditions. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Data labeled "typical" is presented for design guidance only and is not guaranteed.

TABLE 2: DC CHARACTERISTICS

VDD = +5V unless otherwise noted, TA = 40°C to +85°C

Characteristics		Sym	Min	Typ	Max	Units	Conditions
Supply Voltage		VDD	+3.0	—	+8.5	V	
Supply Current		IDD	—	25	60	µA	LCDΦ OSC < 15 kHz
			—	13	30	µA	LCDΦ OSC < 100 Hz
Input High Level		VIH	0.5 VDD	—	VDD	V	
Input Low Level	Clock	VIL1	0	—	0.1 VDD	V	3.0V ≤ VDD ≤ 8.5V
	Data,	VIL2	0	—	0.1 VDD	V	3.0V ≤ VDD ≤ 8.5V
Input Leakage Current	Load	IL	—	0.01	±10	µA	VIN = 0V and +5.0V
Input Capacitance		CI	—	—	5.0	pF	VDD = +5.0V
Segment Output Voltage		VOH	0.8 VDD	—	VDD	V	IOH = -100 µA
		VOL	0	—	0.1 VDD	V	IOL = 100 µA
LCDΦ Input High Level		VIN	0.9 VDD	—	VDD	V	
LCDΦ Input Low Level		VIL	0	—	0.1 VDD	V	
LCDΦ Input Leakage		IL	—	—	10	µA	VIN = 0V and +5.0V
							VDD = +5.0V

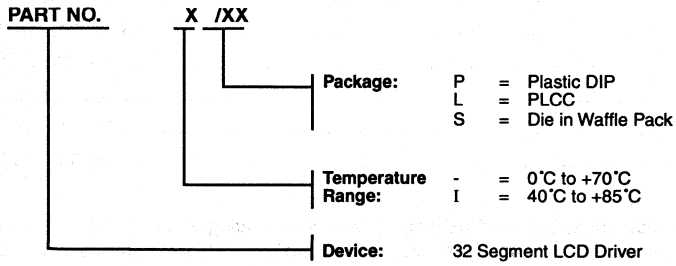
TABLE 3: AC CHARACTERISTICS

Characteristics	Sym	Min	Typ	Max	Units	Conditions
Clock Rate	f	DC	—	1.5	MHz	50% duty cycle
Data Set-up Time	tDS	150	—	—	nsec	Data change to Clk falling edge
Data Hold Time	tDH	50	—	—	nsec	
Load Pulse Width	tPW	175	—	—	nsec	
Data Out Prop. Delay	tPD	—	—	500	nsec	CL = 55 pF

© 1995 Microchip Technology Inc.

AY0438 Product Identification System

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.



Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office.
2. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
3. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

© 1995 Microchip Technology Inc.

SECTION 5
DEVELOPMENT SYSTEMS AND SOFTWARE TOOLS

DEVELOPMENT SYSTEMS:

SOFTWARE TOOLS:

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

System Support

Development System Selection Chart

	PIC16/17 Microcontroller																		
PRODUCT	**PIC16C54**	**PIC16C54A**	**PIC16C55**	**PIC16C56**	**PIC16C57**	**PIC16C58A**	**PIC16C620**	**PIC16C621**	**PIC16C622**	**PIC16C61**	**PIC16C64**	**PIC16C65**	**PIC16C71**	**PIC16C73**	**PIC16C74**	**PIC16C84**	**PIC17C42**	**PIC17C43**	**PIC17C44**
Emulator Systems																			
PICMASTER®-16B													X						
PICMASTER-16C																X			
PICMASTER-16D	X	X	X	X	X	X													
PICMASTER-16E											X								
PICMASTER-16F												X		X	X				
PICMASTER-16G										X									
PICMASTER-16H							X	X	X										
PICMASTER-17																	X		
Development Kits																			
*fuzzy*TECH®-MP	X	X	X	X	X	X					X		X		X	X	X		
PICSTART®-16B1	X	X	X	X	X	X				X			X			X			
PICSTART-16C											X	X		X	X				
PRO MATE™	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Emulator Probe Kits																			
PICPROBE-16B													X						
PICPROBE-16C																X			
PICPROBE-16D	X	X	X	X	X	X													
PICPROBE-16E											X								
PICPROBE-16F												X		X	X				
PICPROBE-16G										X									
PICPROBE-16H							X	X	X										
PICPROBE-17A																	X		
Demonstration Boards																			
PICDEM-1	X	X	X	X	X	X	X	X	X	X			X			X	X	X	X
PICDEM-2											X	X		X	X				

5

© 1995 Microchip Technology Inc.

Compatible & Universal				
Product	**Description**	**PIC16C5X**	**PIC16CXX**	**PIC17CXX**
MPASM	Universal Assembler	X	X	X
MPSIM	Software Simulator	X	X	X
MP-C	C Compiler	X	X	X
fuzzyTECH®	Fuzzy Logic Development Kit	X	X	X
PICSTART®	Low Cost Development Kit	X	X	—
PRO MATE™	Universal Programer/Development Kit	X	X	X
PICMASTER®	Universal In-Circuit Emulator	X	X	X

© 1995 Microchip Technology Inc.

MICROCHIP BBS

Microchip Bulletin Board Service

Get current information and help on Microchip's Bulletin Board Service (BBS)! Microchip wants to provide you with the most responsive service possible. To accomplish this, the systems team monitors the BBS, posting the latest component data and software tool updates, providing technical help and embedded systems insights, and discussing how Microchip products provide project solutions. Extend your technical groups staff with microcontroller and memory experts through Microchip's BBS communication channel.

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information And Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are: 1-800-755-2345 for U.S. and most of Canada, and 1-602-786-7302 for the rest of the world.

These phone numbers are also listed on the "Important Information" sheet that is shipped with all development systems. The hot line message is updated whenever a new software version is added to the Microchip BBS, or when a new upgrade kit becomes available.

CONNECTING TO MICROCHIP

Connect worldwide to the Microchip BBS using the CompuServe® communications network. In most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore, **you do not need CompuServe membership to join Microchip's BBS.**

There is **no charge** for connecting to the BBS, except for a toll charge to the CompuServe access number, where applicable. You do not need to be a CompuServe member to take advantage of this connection (you never actually log in to CompuServe).

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress **<Enter↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress **<Enter↵>** and `Host Name:` will appear.
5. Type **MCHIPBBS**, depress **<Enter↵>** and you will be connected to the Microchip BBS.

In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type **NETWORK**, depress **<Enter↵>** and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

USING THE BULLETIN BOARD

The bulletin board is a multifaceted tool. It can provide you with information on a number of different topics.

- Special Interest Groups
- Files
- Mail
- Bug Lists

Special Interest Groups

Special Interest Groups, or SIGs as they are commonly referred to, provide you with the opportunity to discuss issues and topics of interest with others that share your interest or questions. SIGs may provide you with information not available by any other method because of the broad background of the PIC16/17 user community.

There are SIGs for most Microchip systems, including:

- MPASM
- PICMASTER®
- PRO MATE™
- PICSTART®
- Utilities
- Bugs
- MPSIM
- TRUE GAUGE™
- *fuzzy*TECH®-MP
- ASSP
- MTE1122

These groups are monitored by the Microchip staff.

© 1995 Microchip Technology Inc.

Note: The SIGs provide you with the opportunity to discuss issues and exchange ideas. Technical support and urgent questions should be referred to your local distributor, sales representative or FAE. They are your first level of support.

Files

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. Users can contribute files for distribution on the BBS. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

Mail

The BBS can be used to distribute mail to other users of the service. This is one way to get answers to your questions and problems from the Microchip staff, as well as keeping in touch with fellow Microchip users worldwide.

Consider mailing the moderator of your SIG, or the SYSOP, if you have ideas or questions about Microchip products, or the operation of the BBS.

Software Releases

Software products released by Microchip are referred to by version numbers. Version numbers use the form:

```
xx.yy.zz <status>
```

Where `xx` is the major release number, `yy` is the minor number, and `zz` is the intermediate number. The `status` field displays one of the following categories:

- Alpha
- Intermediate
- Beta
- Released

Production releases are numbered with major, and minor version numbers like:

```
3.04 Released
```

Alpha, Beta and Intermediate releases are numbered with the major, minor and intermediate numbers:

```
3.04.01 Alpha
```

Alpha Release

Alpha designated software is engineering software that has not been submitted to any quality assurance testing. In general, this grade of software is intended for software development team access only, but may be sent to selected individuals for conceptual evaluation.

Intermediate Release

Intermediate released software represents changes to a released software system and is designated as such by adding an intermediate number to the version number. Intermediate changes are represented by:

- Bug Fixes
- Special Releases
- Feature Experiments

Intermediate released software does not represent our most tested and stable software. Typically, it will not have been subject to a thorough and rigorous test suite, unlike production released versions. Therefore, users should use these versions with care, and only in cases where the features provided by an intermediate release are required.

Intermediate releases are primarily available through the BBS.

Beta Release

Preproduction software is designated as Beta. Beta software is sent to Applications Engineers and Consultants, FAEs, and select customers. The Beta Test period is limited to a few weeks. Software that passes Beta testing without having significant flaws, will be production released. Flawed software will be evaluated, repaired, and updated with a new revision number for a subsequent Beta trial.

Production Release

Production released software is software shipped with tool products. Example products are PRO MATE, PICSTART, and PICMASTER. The Major number is advanced when significant feature enhancements are made to the product. The minor version number is advanced for maintenance fixes and minor enhancements. Production released software represents Microchip's most stable and thoroughly tested software.

There will always be a period of time when the Production Released software is not reflected by products being shipped until stocks are rotated. You should always check the BBS for the current production release.

© 1995 Microchip Technology Inc.

PICMASTER® System

PICMASTER Universal In-Circuit Emulator System

5

SYSTEM FEATURES

General:

- Complete Hi-Performance PC-based Microcontroller Development System for the PIC16/17 families.
- For use on PC-compatible 386, and 486 machines under Microsoft® Windows™ 3.X environment.
- Assembler and Simulator Software, Emulator System, and EPROM Programmer unit, sample kit, and demonstration hardware and software provide a complete microcontroller product development environment.

Emulator System:

- Hi-Performance In-Circuit Emulation of Microchip Microcontrollers.
- Real-time instruction emulation.
- Single and Multiple instruction step execution.
- Program Memory emulation and memory mapping capability up to 64K words. Instruction execution can be mapped into either emulation memory or user prototype memory.
- Real-time trace memory capture of 40 bits of information for each instruction cycle in an 8Kx40 trace buffer. Trace region can range from 0 to 64K in any address combinations.
- Real-time trace data can be captured and displayed without halting emulation.
- Unlimited number of hardware breakpoints can be set anywhere in the program memory.
- External Break with "AND"/"OR" capability with internal breakpoints.
- Multiprocessor emulation capability. Up to eight PICMASTER emulators can be synchronized on a single PC, for multi-processor development.
- Extended 48-bit cycle counter.
- Trigger Output available on any range of addresses.
- Full Symbolic Debug Capability. Symbolic display and alter of all register files, special purpose registers, stack registers, and bank registers.
- Selectable Internal Emulator Clock or User Target (Prototype) System Clock.
- User selectable internal or external Power Supply (provided).

PICMASTER is a registered trademark of Microchip Technology Inc.
Windows is a trademark of Microsoft Corp.

© 1995 Microchip Technology Inc.

EPROM Programmer System:

- PRO MATE™ Device Programmer unit for all current PIC16/17 products.
- Operates as a Stand-alone Unit or in Conjunction with a PC-compatible host system.
- Performs READ, PROGRAM, and VERIFY functions in Stand-alone mode.
- PC Host Software provides file display and editing, file transfer to and from programmer unit, device serialization, and program voltage calibration.

Macro Assembler:

- Provides translation of Assembler source code to object code for the PIC16/17 family of microcontrollers.
- Macro-assembly and conditional assembly capability.
- Produces Object files, Listing files, Symbol files, and special files required for symbolic debug with the PICMASTER Emulator System.
- Binary / Hex output formats: INHX8S, INHX8M, INHX16, and PICMASTER.

Simulator:

MPSIM is a discrete event software simulator designed to imitate operation of PIC16/17 microcontrollers. It allows the user to debug software that will use any of these microcontrollers. At any instruction boundary, MPSIM also allows the user to examine and/or modify any data area within the processor, or provide external stimulus to any of the pins. Other MPSIM features include program load and save, disassembler, symbolic debug, and execution trace and breakpoints.

Demo Board:

The PICDEM Demonstration Board provides a user with a simple hardware tool through which software can be exercised and debugged. A step-by-step tutorial enables first-time users of PICMASTER to become familiar with all the features of the emulator. A generous prototype area allows the user to build additional hardware for their project.

SYSTEM DESCRIPTION

The PICMASTER Universal In-Circuit Emulator System is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC16/17 family. The PICMASTER system currently supports the PIC16C54, PIC16C54A, PIC16C55, PIC16C56, PIC16C57 and PIC16C58A at clock frequencies of 20MHz; the PIC16C61, PIC16C620, PIC16C621, PIC16C622, PIC16C64, PIC16C71, PIC16C73, PIC16C74, PIC16C84 to 10MHz; and the PIC17C42 at 16MHz.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new microcontroller architectures with data and program memory paths to 16 bits.

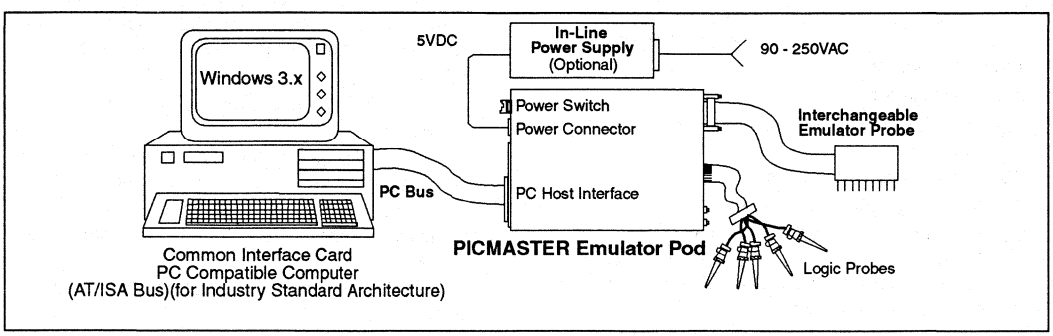
The Emulator System is designed to operate PC-compatible machines ranging from 286-AT® class ISA-bus systems through the new Pentium™ machines. The development software runs in the Microsoft Windows 3.X environment, allowing the operator access to a wide range of supporting software and accessories.

Provided with the PICMASTER System is a high performance, real-time In-Circuit Emulator, a microcontroller programmer unit, a macro assembler program, and a simulator program. Sample programs are provided to help quickly familiarize the user with the development system and the PIC16/17 microcontroller families.

Coupled with the user's choice of text editor, the system is ready for development of products containing any of Microchip's microcontroller products.

A "Quick Start" Product Sample Pak containing user programmable parts is included for additional convenience (only devices supported by the specified probe header).

Microchip provides additional customer support to developers through an electronic Bulletin Board System (BBS). Customers have access to the latest updates in software as well as application source code examples. Consult your local sales representative for information on accessing the BBS system.



PRO MATE is a trademark of Microchip Technology Inc.
AT is a registered trademark of IBM Corp.
Pentium is a trademark of Intel Corporation.

© 1995 Microchip Technology Inc.

Host System Requirements:

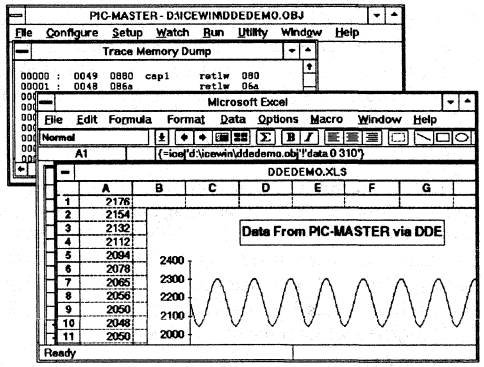
The PICMASTER has been designed as a real-time emulation system with advanced features generally found on more expensive development tools. The IBM PC-compatible platform and Windows 3.X environment was chosen to best make these features available to you the end user. To properly take advantages of these features, PICMASTER requires installation on a system having the following minimum configuration:

- PC/AT-compatible machine: 286, 386SX, 386DX, or 486 with ISA, EISA Bus or Pentium machines.
- EGA, VGA, 8514/A, Hercules graphic card (EGA or higher recommended).
- MS-DOS® / PC-DOS version 5.0 or greater.
- Microsoft® Windows version 3.0 or greater operating in either standard or 386 enhanced mode).
- 1 Mbyte RAM (2 Mbytes recommended).
- One 3.5" floppy disk drive.
- Approximately 10 Mbytes of hard disk (1 Mbyte required for PICMASTER, remainder for Windows 3.X system).
- One 8-bit PC/AT (ISA) I/O expansion slot (half size)
- Microsoft mouse or compatible (highly recommended).

Emulator System Components:

The PICMASTER Emulator Universal System consists primarily of four major components:

- **Host-Interface Card:** The PC Host Interface Card connects the emulator system to a PC compatible system. This high-speed parallel interface requires a single half-size standard AT / ISA slot in the host system. A 37-conductor cable connects the interface card to the external Emulator Control Pod.
- **Emulator Control Pod:** The Emulator Control Pod contains all emulation and control logic common to all microcontroller devices. Emulation memory, trace memory, event and cycle timers, and trace/breakpoint logic are contained here. The Pod controls and interfaces to an interchangeable target-specific emulator probe via a 14" precision ribbon cable.
- **Target-specific Emulator Probe:** A probe specific to microcontroller family to be emulated is installed on the ribbon cable coming from the control pod. This probe configures the universal system for emulation of a specific microcontroller. Currently, the PIC16C5X family, PIC16CXX family, and the PIC17C42 microcontrollers are supported. Future microcontroller probes will be available as they are released.



- **PC Host Emulation Control Software:** Host software necessary to control and provide a working user interface is the last major component of the system. The emulation software runs in the Windows 3.X environment, and provides the user with full display, alter, and control of the system under emulation. The Control Software is also universal to all microcontroller families.

  The Windows 3.X operating System allows the developer to take full advantage of the many powerful features and functions of the PICMASTER system.

  PICMASTER emulation can operate in one window, while a text editor is running in a second window. Dynamic Data Exchange (DDE), a feature of Windows 3.X, will be available in this and future versions of the software. DDE allows data to be dynamically transferred between two or more Windows programs. With this feature, data collected with PICMASTER can be automatically transferred to a spreadsheet or database program for further analysis.

  Under Windows 3.X, up to eight PICMASTER emulators can run simultaneously on the same PC making development of multi-microcontroller systems possible (e.g., a system containing a PIC16CXX processor and a PIC17CXX processor).

PRO MATE Device Programmer:

The PRO MATE Programmer system included in the PICMASTER Development System provides the product developer with the ability to program (transfer) the developer's software into PIC16/17 microcontrollers.

The programmer unit comes complete with accessories for use with a PC host computer. Supplied are interface cables and connectors to a standard PC serial port, a power supply unit, and host operating software.

The PRO MATE Programmer will work in either standalone mode, or in PC host connected mode. Connected to a PC host, many more features are available to the user.

5

MS-DOS is a registered trademark of Microsoft Corp.

© 1995 Microchip Technology Inc.

STAND-ALONE MODE

Stand-alone mode is useful in situations where a PC may not be available or even required, such as in the field or in a lab production environment. In stand-alone mode the following programming functions are available:

VERIFY:

VERIFY performs two functions. For a programmed part, the device in the programming socket will be compared to the program data stored in internal memory. If the data and fuse settings are correct, VERIFIED will be displayed. VERIFY will also confirm that erased parts are blank. A device in the socket will display ERASED if all programmable locations are blank.

PROGRAM:

In stand-alone mode, devices inserted into the programmer socket will be programmed with data currently stored in memory. Pressing the PROGRAM key will cause the unit to program and verify both the program memory and the device fuses. If all program successfully, PGM OKAY will be displayed.

READ:

A pre-programmed device placed in the programmer socket can be read into the programmer unit by pressing the READ key. Program and fuse data will be read and stored into internal memory. Various options exist with the READ function.

PC HOST CONNECT MODE

When the PRO MATE is connected to a host PC system, many more options and conveniences are available to the user. Host mode allows full interactive control over the PRO MATE unit. A full screen, user-friendly software program is provided to fully assist the user.

As in stand-alone mode, parts may be Read, Programmed, Blank checked, and Verified. Also, all fuses and ID locations may be specified. In addition, other features available in host-mode are:

Editing

A large screen buffer editing facility allows the user to change and program location in hexadecimal. Complete program and fuse data can be loaded and saved to DOS disk files. Files generated by the Assembler program are directly loadable into programmer memory.

VDD and VPP Adjust

The programming environment voltage settings of VDD max, VDD min, and VPP can be set and altered only on PC host mode. The voltage settings allow the user to program the part in the environment that the part will be used. The part will be programmed at VDD max and verified at VDD min. VPP is the programming voltage.

PICMASTER PROBE Specifications

Table 1 shows the current probe specifications for the PICMASTER In-Circuit Emulator. The devices are supported regardless of program memory type (ROM, EPROM or EEPROM), process technology or voltage range. That is, selecting the PROBE that supports the PIC16C54 (Probe-16D) also supports the PIC16CR54, PIC16C54A and the PIC16LC54A devices. The probe would also support other variations as they become available (such as PIC16CR54A).

TABLE 1: PICMASTER PROBE SPECIFICATIONS

PICMASTER Probe	Devices Supported	Probe	
		Maximum Frequency	Operating Voltage
PROBE-16B	PIC16C71	10MHz	4.5V - 5.5V
PROBE-16C	PIC16C84	10MHz	4.5V - 5.5V
PROBE-16D	PIC16C54, PIC16C55, PIC16C56, PIC16C57, PIC16C58A	20MHz	4.5V - 5.5V
PROBE-16E	PIC16C64	10MHz	4.5V - 5.5V
PROBE-16F	PIC16C65 PIC16C74 PIC16C73	10MHz	4.5V - 5.5V
PROBE-16G	PIC16C61	10MHz	4.5V - 5.5V
PROBE-16H	PIC16C620 PIC16C621 PIC16C622	10MHz	4.5V - 5.5V
PROBE - 17A	PIC17C42	16MHz	4.5V - 5.5V

© 1995 Microchip Technology Inc.

Sales and Support

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NUMBER	DESCRIPTION
EM167011	Complete PICMASTER-16B System for PIC16C71
EM167012	Complete PICMASTER-16B System for PIC16C71 without Programmer
EM167013	Complete PICMASTER-16C System for PIC16C84
EM167014	Complete PICMASTER-16C System for PIC16C84 without Programmer
EM167015	Complete PICMASTER-16D System for PIC16C5X
EM167016	Complete PICMASTER-16D System for PIC16C5X without Programmer
EM167017	Complete PICMASTER-16E System for PIC16C64
EM167018	Complete PICMASTER-16E System for PIC16C64 without Programmer
EM167019	Complete PICMASTER-16F System for PIC16C65, PIC16C74/C73
EM167020	Complete PICMASTER-16F System for PIC16C65, PIC16C74/C73 without Programmer
EM167021	Complete PICMASTER-16G System for PIC16C61
EM167022	Complete PICMASTER-16G System for PIC16C61 without Programmer
EM167023	Complete PICMASTER-16H System for PIC16C62X
EM167024	Complete PICMASTER-16H System for PIC16C62X without Programmer
EM177001	Complete PICMASTER-17 System for PIC17C42
EM177004	Complete PICMASTER-17 System for PIC17C42 without Programmer

© 1995 Microchip Technology Inc.

NOTES:

© 1995 Microchip Technology Inc.

MICROCHIP

PRO MATE™

Universal Device Programmer

SYSTEM FEATURES

Device Programmer System:

- PRO MATE Programmer unit for the PIC16C5X, PIC16CXX, PIC17CXX Microcontroller family.
- Operates as a Stand-alone Unit or in Conjunction with a PC Compatible host system.
- READS, PROGRAMS, and VERIFIES in Stand-alone mode.
- PC Host Software provides file display and editing, and transfer to and from Programmer unit
- Communicates with PC via RS-232
- Modular socket modules provide easy migration from one PIC16/17 microcontroller product to another.

SYSTEM DESCRIPTION

PRO MATE Programmer:

The PRO MATE Programmer system provides the product developer with the ability to program user software into PIC16C5X, PIC16CXX, PIC17CXX CMOS microcontrollers.

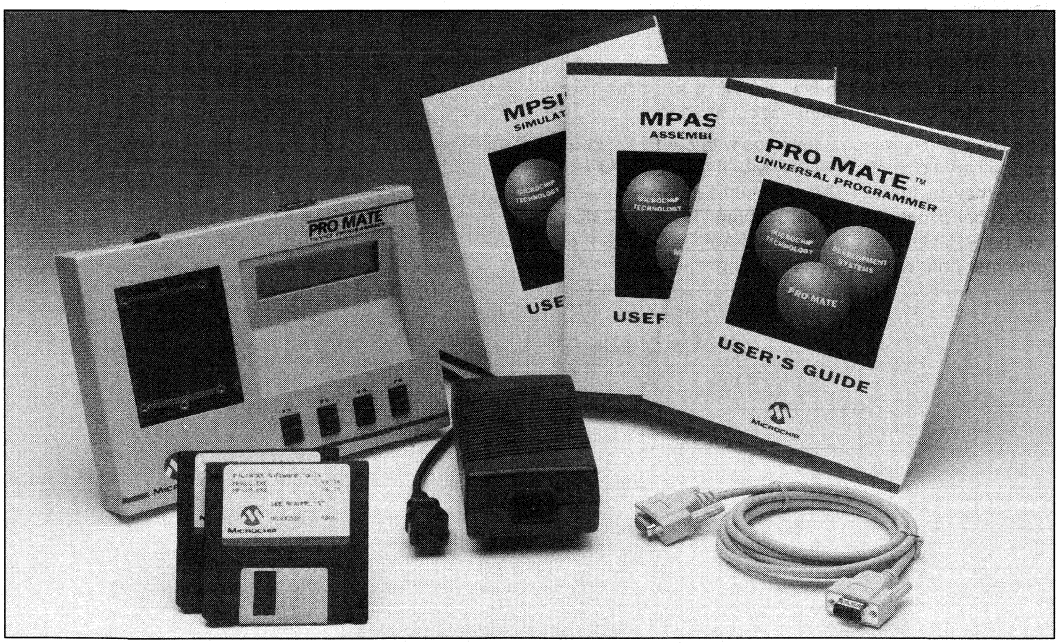
PRO MATE is also supplied with a discrete event software simulator (MPSIM) and a Universal PIC16/17 Macro assembler (MPASM).

The programmer unit comes complete with accessories to be used with the PC host computer. Supplied are interface cables and connectors to a standard PC serial port, a universal input power supply unit, and host operating software.

The PRO MATE Programmer will work in either stand-alone mode, or in PC host connected mode. Connected to a PC host, many more features are available to the user.

The modular socket module design allows users to easily migrate between PIC16/17 devices at the lowest possible cost.

5



PRO MATE is a trademark of Microchip Technology Inc.

© 1995 Microchip Technology Inc.

STAND-ALONE MODE

Stand-alone mode is useful in situations where a PC may not be available or even required, such as in the field or in a lab production environment. In stand-alone mode the following programming functions are available:

VERIFY

VERIFY performs two functions. For a programmed part, the device in the programming socket will be compared to the program data stored in internal memory. If the data and fuse settings are correct, VERIFIED will be displayed. VERIFY will also confirm that erased parts are blank. A device in the socket will display ERASED if all programmable locations are blank.

PROGRAM

In stand-alone mode, devices inserted into the programmer socket will be programmed with data currently stored in memory. Pressing the PROGRAM key will cause the unit to program and verify both the program memory and the device fuses. If all program successfully, PGM OKAY will be displayed.

READ

A pre-programmed device placed in the programmer socket can be read into the programmer unit by pressing the READ key. Program and fuse data will be read and stored into internal memory. Various options exist with the READ function.

PC HOST CONNECT MODE

The PRO MATE provides a very user friendly user interface which allows complete control over the programming session.

The PRO MATE host software is a DOS windowed environment with full mouse support to allow the user to point and click when entering commands.

The Host Software communicates with the PRO MATE via the serial port of the PC. Any of the four (COM 1-4) ports may be used. The communication is done at 19200 baud to insure fast throughput. Communication will be established with the PRO MATE Device Programmer prior to any transfers taking place.

Serialization is done by generating a serialization file, and then using that file to serialize locations in the PIC16/17 microcontroller. Once a serialization file is generated, it may be used over different programming sessions. Serial numbers are automatically marked as used when a PIC16/17 is programmed successfully with that serial number.

Complete control over the programming environment is also provided. Control over the programming and verify voltage of V_{DD} insures that the Microcontroller will perform in the desired environment. Programming (V_{PP}) voltage is also adjustable to insure complete compatibility with future programming algorithms.

Macro Assembler:

- Provides translation of Assembler source code to object code for all PIC16/17 microcontroller product family.
- Macro-Assembly capability.
- Provides Object files, Listing files, Symbol files, and special files required for symbolic debug with the PIC16/17 Emulator System.
- Output formats: INHX8S and INHX8M.

Simulator:

- Instruction-level Simulator of the PIC16/17 microcontroller product family.
- For PC-compatible systems running the MS-DOS® operating system.
- Full screen simulation user interface.
- Symbolic debugging capability.
- I/O stimulus input capability.

MS-DOS is a registered trademark of Microsoft Corp.

© 1995 Microchip Technology Inc.

PRO MATE SOCKET MODULE CROSS-REFERENCE

	Pin Count	DIP	SOIC	SSOP	PLCC	MQFP
PIC16C54	18/20	AC164001	AC164002	AC164015	—	—
PIC16C54A	18/20	AC164001	AC164002	AC164015	—	—
PIC16CR54	18/20	AC164001	AC164002	AC164015	—	—
PIC16C55	28	AC164001	AC164002	AC164015	—	—
PIC16C56	18/20	AC164001	AC164002	AC164015	—	—
PIC16C57	28	AC164001	AC164002	AC164015	—	—
PIC16C57A	28	AC164001	AC164002	AC164015	—	—
PIC16C58A	18/20	AC164001	AC164002	AC164015	—	—
PIC16C61	18	AC164010	AC164010	—	—	—
PIC16C64	40/44	AC164012	—	—	AC164013	AC164014
PIC16C65	40/44	AC164012	—	—	AC164013	AC164014
PIC16C620	18/20	AC164010	AC164010	AC164018	—	—
PIC16C621	18/20	AC164010	AC164010	AC164018	—	—
PIC16C622	18/20	AC164010	AC164010	AC164018	—	—
PIC16C71	18	AC164010	AC164010	—	—	—
PIC16C73	28	AC164012	AC164017	—	—	—
PIC16C74	40/44	AC164012	—	—	AC164013	AC164014
PIC16C84	18	AC164010	AC164010		—	—
PIC17C42	40/44	AC174001	—	—	AC174002	AC174004
PIC17C44	40/44	AC174001	—	—	AC174002	—

© 1995 Microchip Technology Inc.

SALES AND SUPPORT

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

Programmer Part Number	Description
DV007001	Programmer Kit as described above

Socket Part Number	Description
AC164001	PIC16C54 thru C58A 18- & 28-Lead PDIP Socket Module
AC164002	PIC16C54 thru C58A 18- & 28-Lead SOIC Socket Module
AC164010	PIC16C61, PIC16C62X, PIC16C71, PIC16C84, 18-Lead PDIP/SOIC Socket Module
AC164012	PIC16C62, PIC16C64, PIC16C65, PIC16C73, PIC16C74, 40-Lead PDIP Socket Module
AC164013	PIC16C64, PIC16C65, PIC16C74, 44-Lead PLCC Socket Module
AC164014	PIC16C64, PIC16C65, PIC16C74, 44-Lead PQFP Socket Module
AC164015	PIC16C54 through PIC16C58A, 20 and 28-Lead SSOP Socket Module
AC174001	PIC17C42, PIC17C43, PIC17C44, 40-Lead PDIP Socket Module
AC174002	PIC17C42, PIC17C43, PIC17C44, 44-Lead PLCC Socket Module
AC174004	PIC17C42 44-Lead QFP Socket Module
AC164017	PIC16C73, 28-Lead SOIC Socket Module
AC164018	PIC16C62X 20-Lead SSOP Socket Module

Socket modules are sold separately.

PICSTART®-16B1

PIC16CXX Low-Cost Microcontroller Development System

SYSTEM FEATURES

EPROM Programmer System:

- EPROM Development Programmer unit for the PIC16C5X and selected PIC16CXX Microcontroller family members. Supports PIC16C54, PIC16C54A, PIC16C55, PIC16C56, PIC16C58A, PIC16C57, PIC16C61, PIC16C71, PIC16C84.
- Operates with a PC-compatible host system.
- READS, PROGRAMS, and VERIFIES EPROM Memory.
- PC Host Software provides file display and editing, and transfer to and from Programmer unit.
- Universal power supply
- RS-232 interface cable

Macro Assembler:

- Provides translation of Assembler source code to object code for all PIC16CXX microcontroller product family.
- For PC-compatible systems running the MS-DOS® operating system.
- Macro-Assembly capability.
- Provides Object files, Listing files, Symbol files, and special files required for symbolic debug with the PIC16CXX Emulator System.
- Output formats: INHX8S and INHX8M.

Simulator:

- Instruction-level Simulator of the PIC16CXX microcontroller product family.
- For PC-compatible systems running the MS-DOS® operating system.
- Full screen simulation user interface.
- Symbolic debugging capability.
- I/O stimulus input capability.

"Quick Start" Sample Kit:

- Provides the User / Developer with a sample kit of PIC16CXX parts for initial prototype use.

5

PICSTART is a registered trademark of Microchip Technology Inc.
MS-DOS is a registered trademark of Microsoft Corp.

© 1995 Microchip Technology Inc.

PICSTART®-16B1

SYSTEM DESCRIPTION

The PICSTART-16B1 Development System provides the product development engineer with an alternative low-cost introductory microcontroller design tool set for the PIC16CXX family where full real-time emulation is not required. The equipment in the PICSTART-16B1 system operates on any PC compatible machine running the MS-DOS/PC-DOS operating system.

Provided in the System is an MS-DOS-based Software Simulator program (MPSIM), a microcontroller EPROM programmer, and a macro assembler program (MPASM).

Sample software programs to be run on the simulator are provided to help the user to quickly become familiar with the development system and the PIC16CXX microcontroller line.

The user need only provide his or her own preferred text editor and the system is ready for development of end products using the PIC16C54, PIC16C55, PIC16C56, PIC16C57, PIC16C58A, PIC16C61, PIC16C71 or PIC16C84 microcontrollers.

A "Quick Start" PIC16CXX Product Sample Pak containing user programmable parts is also included.

Microchip provides additional customer support to developers through an electronic Bulletin Board System (BBS). Customers have access to the latest updates in software as well as application source code examples. Consult your local sales representative for information on accessing the BBS.

PICSTART-16B1 Development Programmer:

The Microchip device programmer system included in the PICSTART-16B1 Development System provides the product developer with the ability to program user software into PIC16CXX EPROM microcontrollers. It is designed to be a development programmer and not recommended for use in a production environment.

The programmer unit connects to a standard PC serial port.

A full screen, user-friendly software program is provided for full interactive control over the programmer. Parts may be Read, Programmed, Blank checked and Verified. Also, all fuses and ID locations may be specified.

A large screen buffer editing facility allows the user to change and program location in hexadecimal. Complete program data can be loaded and saved to DOS disk files. Files generated by the MPASM Assembler program are directly loadable into programmer memory.

MPSIM Simulator:

The MPSIM Simulator program provides the developer with an instruction and limited I/O simulator software program for debugging PIC16/17 assembler code.

The simulator is meant for use with smaller projects not requiring precise, more extensive development equipment. Many applications can be developed by using a simulator program alone.

The MPSIM Simulator has the following features to assist in the debugging of software/firmware for the user.

Program Load/Save

Commands exist to load assembled object file programs into simulation memory. Conversely, programs may be saved from program simulation memory back to the PC disk.

Display & Alter

Provisions are made to display and alter Program Memory, Register Files, and status register bits. Also simulator information such as cycle times, elapsed time, and step count can be displayed.

Utility Functions

Various utility functions exist which assist the user in operating the simulator. Memory and registers can be cleared by command. Memory can be searched to find occurrences of instructions, register use, and ASCII data.

Disassembler

Program memory can be disassembled showing both hexadecimal data and instruction mnemonics for specified address ranges.

Symbolic Debugging

The simulator provides for symbolic referencing to aid and simplify debugging. The symbol table may be displayed. New symbols defined and unwanted symbols deleted.

Execution and Trace

During program execution, address ranges, registers, register contents, and others can be traced.

Breakpoints

The user may specify up to 512 breakpoints at any one time.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part number, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DV163003	PICSTART-16B1 DEVELOPMENT SYSTEM

© 1995 Microchip Technology Inc.

MICROCHIP

PICSTART®-16C

PIC16CXX Low-Cost Microcontroller Development System

SYSTEM FEATURES

EPROM Programmer System:

- EPROM Programmer unit for the PIC16CXX Microcontroller family. Supports the PIC16C64, PIC16C65, PIC16C73 and the PIC16C74.
- Operates with a PC-compatible host system.
- READS, PROGRAMS, and VERIFIES EPROM Memory.
- PC Host Software provides file display and editing, and transfer to and from Programmer unit.
- Universal power supply
- RS-232 interface cable

Macro Assembler:

- Provides translation of Assembler source code to object code for all PIC16CXX microcontroller product family.
- Macro-Assembly capability.
- For PC-compatible systems running the MS-DOS® operating system.
- Provides Object files, Listing files, Symbol files, and special files required for symbolic debug with the PIC16CXX Emulator System.
- Output formats: INHX8S and INHX8M.

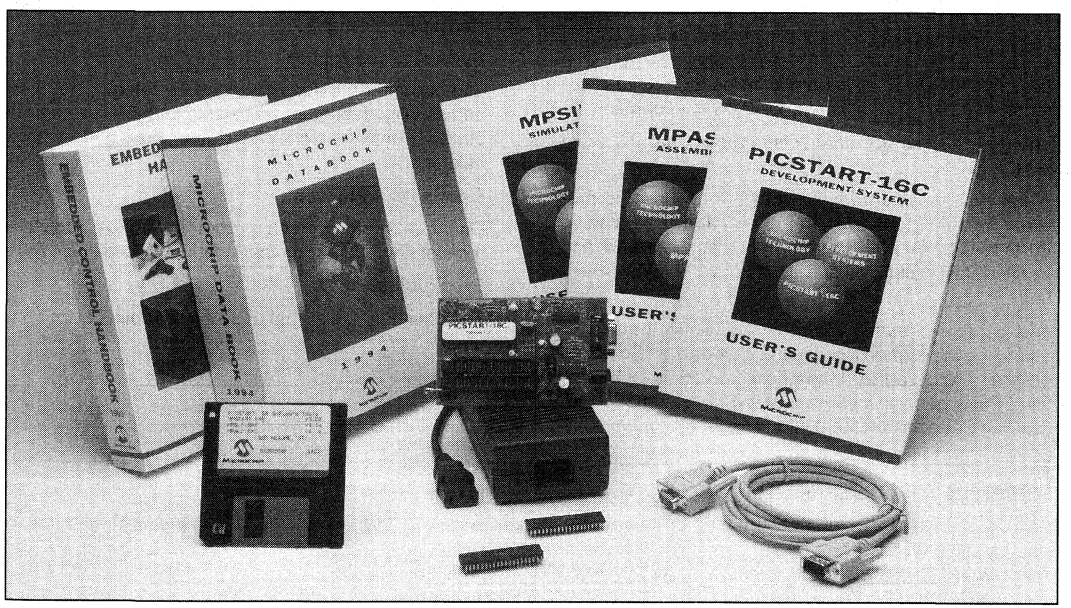
Simulator:

- Instruction-level Simulator of the PIC16/17 microcontroller product family.
- For PC-compatible systems running the MS-DOS® operating system.
- Full screen simulation user interface.
- Symbolic debugging capability.
- I/O stimulus input capability.

"Quick Start" Sample Kit:

- Provides the User / Developer with a sample kit of the supported PIC16CXX parts for initial prototype use.

5



PICSTART is a registered trademark of Microchip Technology Inc.
MS-DOS is a registered trademark of Microsoft Corp.

© 1995 Microchip Technology Inc.

PICSTART®-16C

SYSTEM DESCRIPTION

The PICSTART-16C Development System provides the product development engineer with an alternative low-cost introductory microcontroller design tool set for the PIC16CXX family where full real-time emulation is not required. The equipment in the PICSTART-16C system operates on any PC compatible machine running the MS-DOS/PC-DOS operating system.

Provided in the System is an MS-DOS-based Software Simulator program (MPSIM), a microcontroller EPROM programmer, and a macro assembler program (MPASM).

Sample software programs to be run on the simulator are provided to help the user to quickly become familiar with the development system and the PIC16CXX microcontroller line.

The user need only provide his or her own preferred text editor and the system is ready for development of end products using the PIC16C64, PIC16C65, PIC16C73, or the PIC16C74.

A "Quick Start" PIC16CXX Product Sample Pak containing user programmable parts is also included.

Microchip provides additional customer support to developers through an electronic Bulletin Board System (BBS). Customers have access to the latest updates in software as well as application source code examples. Consult your local sales representative for information on accessing the BBS.

PICSTART-16C Development Programmer:

The Microchip device programmer system included in the PICSTART-16C Development System provides the product developer with the ability to program user software into PIC16CXX EPROM microcontrollers. It is designed to be a development programmer and not recommended for use in a production environment.

The programmer unit connects to a standard PC serial port.

A full screen, user-friendly software program is provided for full interactive control over the programmer. Parts may be Read, Programmed, Blank checked, and Verified. Also, all fuses and ID locations may be specified.

A large screen buffer editing facility allows the user to change and program location in hexadecimal. Complete program data can be loaded and saved to DOS disk files. Files generated by the MPASM Assembler program are directly loadable into programmer memory.

MPSIM Simulator:

The MPSIM Simulator program provides the developer with an instruction and limited I/O simulator software program for debugging PIC16/17 assembler code.

The simulator is meant for use with smaller projects not requiring precise more extensive development equipment. Many applications can be developed by using a simulator program alone.

The MPSIM Simulator has the following features to assist in the debugging of software/firmware for the user.

Program Load/Save

Commands exist to load assembled object file programs into simulation memory. Conversely, programs may be saved from program simulation memory back to the PC disk.

Display & Alter

Provisions are made to display and alter Program Memory, Register Files, and status register bits. Also simulator information such as cycle times, elapsed time, and step count can be displayed.

Utility Functions

Various utility functions exist which assist the user in operating the simulator. Memory and registers can be cleared by command. Memory can be searched to find occurances of instructions, register use, and ASCII data.

Disassembler

Program memory can be disassembled showing both hexadecimal data and instruction mnemonics for specified address ranges.

Symbolic Debugging

The simulator provides for symbolic referencing to aid and simplify debugging. The symbol table may be displayed. New symbols defined and unwanted symbols deleted.

Execution and Trace

During program execution, address ranges, registers, register contents, and others can be traced.

Breakpoints

The user may specify up to 512 breakpoints at any one time.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part number, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DV163002	PICSTART-16C DEVELOPMENT SYSTEM

© 1995 Microchip Technology Inc.

MICROCHIP

PICDEM-1

Low-Cost PIC16/17 Demonstration Board

PRODUCT INFORMATION

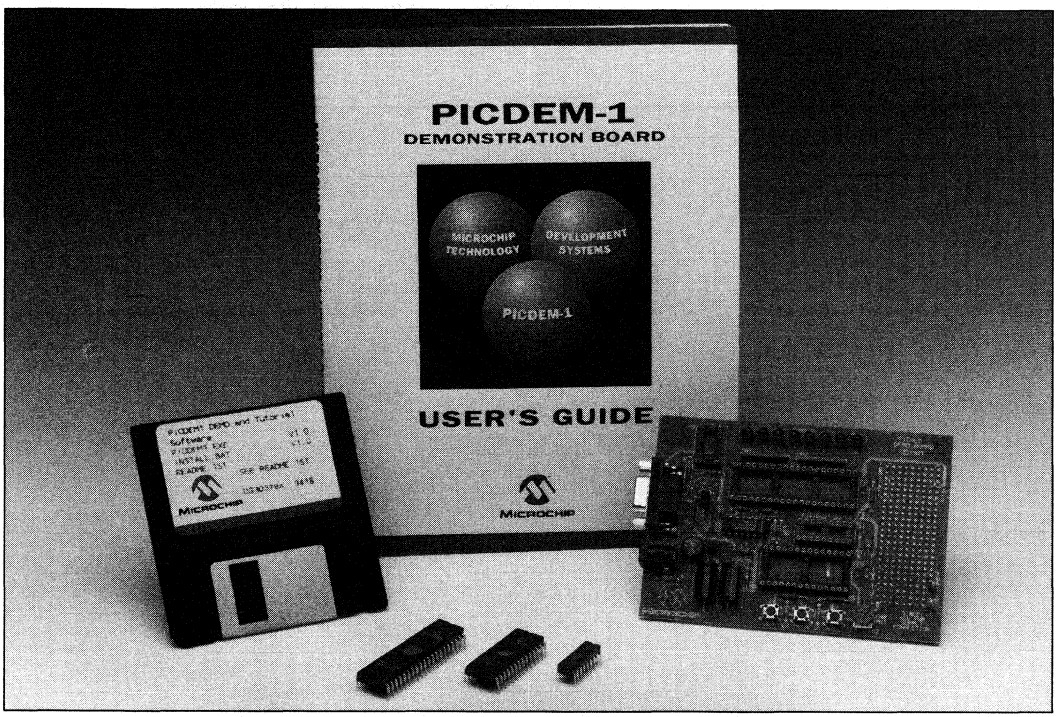
The PICDEM-1 is a simple board which demonstrates the capabilities of several Microchip microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58), PIC16C61, PIC16C62X, PIC16C71, PIC16C84, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5" disk. The users can program the samples (one each of PIC17C42, PIC16C71 and PIC16C55) provided with the PICDEM-1, on a PRO MATE™ or PICSTART® programmer and easily debug/test the sample code, or the user can connect the PICDEM-1 with the PICMASTER emulator and download the sample code to the emulator and debug/test the code. Additionally, a generous 200-hole prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s).

FEATURES:

Hardware:

- 40-pin, 28-pin and 18-pin Precision sockets for all supported microcontrollers.
- On board +5V regulator and filter rectifier for direct input from 9V AC/DC wall adapter.
- RS-232 socket and associated hardware for direct connection to RS-232 interface.
- 5K pot to simulate analog input for PIC16C71.
- Three push button Key for external stimulus and RESET.
- Eight bright LEDs connect to PORTB, help in displaying 8-bit binary values on PORTB.
- Socket for "canned" crystal Oscillator.
- Unpopulated holes provided for Xtal connection
- Jumper to disconnect on board RC Oscillator.
- 200-hole prototype area for user's hardware.

5



PICSTART is a registered trademark, and PRO MATE is a trademark of Microchip Technology Inc.

© 1995 Microchip Technology Inc.

Software:

- Program for PIC16C71 to demonstrate on-chip A/D features.
- Program for PIC16C84 to demonstrate on-chip EEPROM.
- Program for PIC17C42 to demonstrate on-chip USART.
- Program for PIC16C5X to demonstrate key input capability.
- All demo programs supplied on 3.5" disk,
- Additional programs available on Microchip's BBS.

DOCUMENTATION

- A comprehensive User's Guide with easy to follow step-by-step Getting Started and a Tutorial.
- Schematics for the entire circuit.

SAMPLES

Several UV erasable devices supplied are included. The device types may change from time to time. The supplied devices are typically:

- PIC17C42
- PIC16C71
- PIC16C55

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DM163001	Low-cost Demonstration Board for PIC16C5X, PIC16C61, PIC16C62X, PIC16C71, PIC16C84, PIC17C42, PIC17C43 and PIC17C44.

© 1995 Microchip Technology Inc.

MICROCHIP

PICDEM-2

Low-Cost PIC16CXX Demonstration Board

PRODUCT INFORMATION

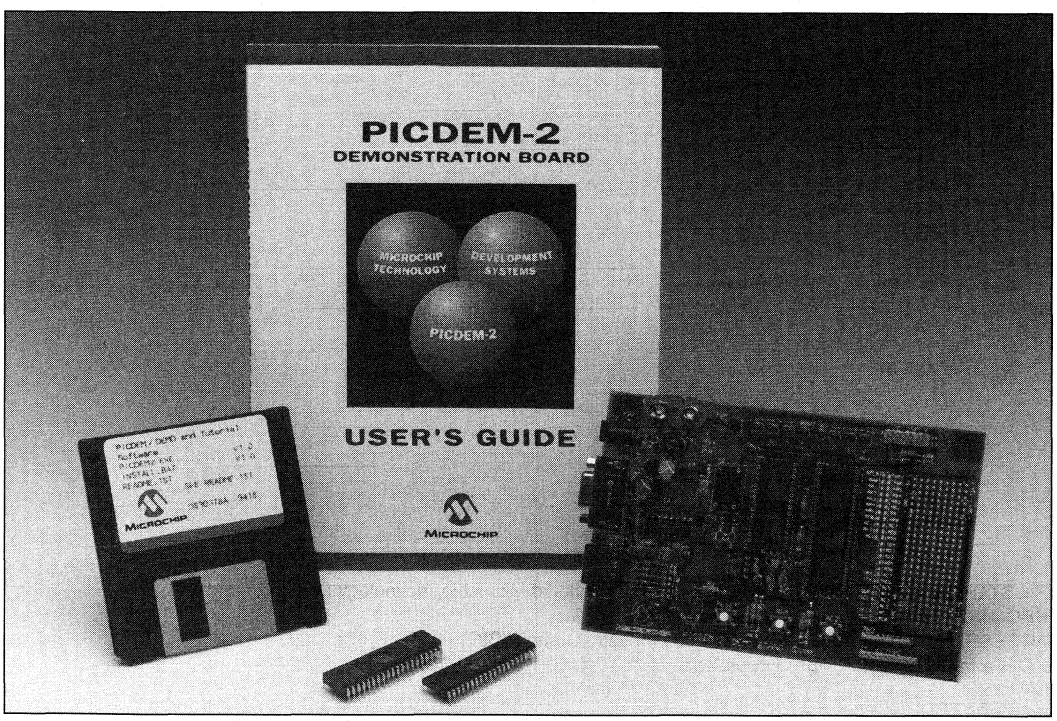
The PICDEM-2 is a simple board which demonstrates the capabilities of several Microchip microcontrollers, including PIC16C64/65 and PIC16C73/74. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5" disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE™ or PICSTART® programmer and easily debug and test the sample code. The PICDEM-2 is also usable with the PICMASTER® emulator, and all of the sample programs can be run and modified using the PICMASTER. Additionally, a generous prototype area is available for user hardware.

FEATURES:

Hardware:

- 40- and 28-pin DIP sockets
- On board +5V regulator for direct input from 9V AC/DC wall adapter or 9V battery.
- RS-232C socket and associated hardware for direct connection to RS-232C interface.
- 5K pot for analog inputs for the PIC16C73/74
- Three push button keys for external stimulus and RESET.
- Eight bright LEDs connected to PORTB for displaying 8-bit binary values.
- Socket for "canned" crystal oscillator.
- Unpopulated holes provided for crystal connection
- 128 x 8 Serial EEPROM.
- LCD module header.
- Keyboard header.
- Unpopulated holes for ACCESS.bus™ connector.
- Jacks for connection of 9V battery.
- Jumper to disconnect on-board RC oscillator.
- Prototype area for user hardware.

5



© 1995 Microchip Technology Inc.

Software:

- Program for PIC16C74 to demonstrate on-chip A/D feature.
- Program for PIC16C64 to demonstrate I^2C Serial EEPROM usage.
- All demo programs supplied on 3.5" disk.
- Additional programs available on Microchip's BBS.

DOCUMENTATION:

- A comprehensive User's Guide with easy to follow, step-by-step Getting Started and Tutorial.
- Full schematics.

Samples:

Several UV erasable devices supplied are included. The device types may change from time to time. The supplied devices are typically:

- PIC16C64
- PIC16C74

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DM163002	Low-cost Demonstration Board for PIC16C64, PIC16C65, PIC16C73 AND PIC17C74

PICSTART and PICMASTER are registered trademarks of Microchip Technology Inc.
PRO MATE IIs a trademark of Microchip Technology Inc.
ACCESS.bus is a trademark of Access.bus Industry Group (ABIG).
I^2C is a trademark of Philips Corporation.

© 1995 Microchip Technology Inc.

MPASM Universal Assembler

Universal PIC16/17 Microcontroller Assembler Software

This product brief describes the technical aspects of the PIC16/17 Assembler. The MPASM Cross Assembler is a PC hosted symbolic assembler. It supports all microcontroller series, including the PIC16C5X, PIC16CXX and PIC17CXX families.

MPASM offers fully featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

MPASM allows full symbolic debugging from the Microchip Universal Emulator System (PICMASTER®).

MPASM REQUIREMENTS

MPASM will run on any IBM PC/AT® or compatible computer running DOS 5.0 or later.

MPASM ASSEMBLER FEATURES

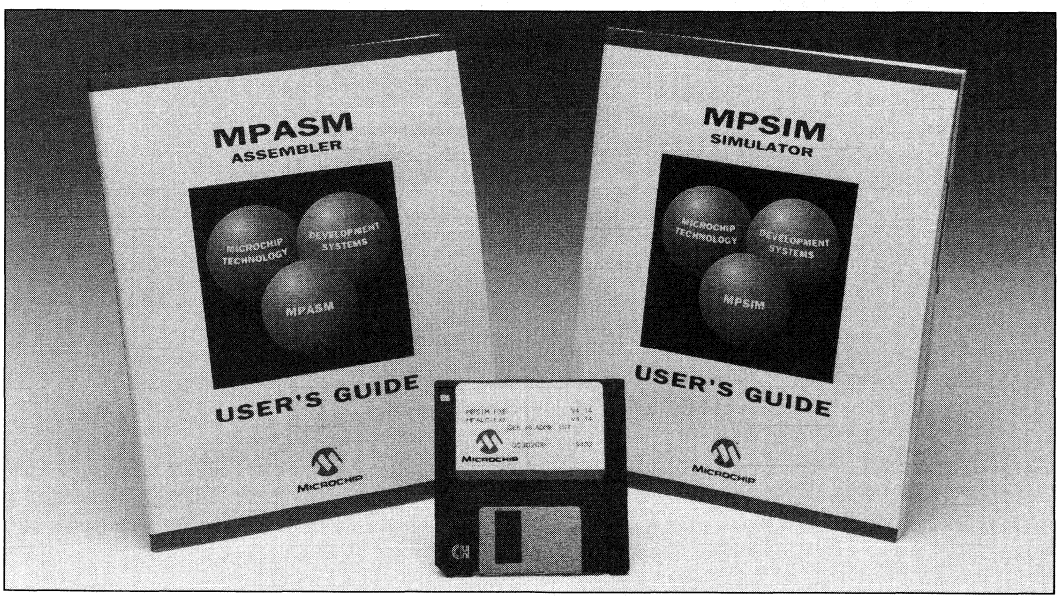
MPASM supports the 12-bit PIC16C5X, the 14-bit PIC16CXX, and the 16-bit PIC17CXX cores.

All instructions are single-word and single-cycle, except for branches, which execute in two cycles. Most instructions operate on one or more operands.

MPASM have the following features to assist in developing software for specific user applications:

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro Assembly Capability
- Provides Object, Listing, Symbol and special files required for debugging with one of the Microchip Emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.
- Output formats: INHX8S, INHX8M, INHX32 and relocatable objects.

5



PICMASTER is a registered trademark of Microchip Technology Inc.
IBM PC/AT is a registered trademark of IBM Corp.

© 1995 Microchip Technology Inc.

MPASM Universal Assembler

MPASM DIRECTIVE LANGUAGE

MPASM provides a full featured directive language represented by four basic classes of directives:

- Data Directives are those that control the allocation of memory and provide a way to refer to data items symbolically, by meaningful names.
- Listing Directives control the MPASM listing display. They allow the specification of titles and subtitles, page ejects and other listing control.
- Control Directives permit sections of conditionally assembled code.
- Macro Directives control the execution and data allocation within macro body definitions.

MPASM INSTRUCTION SET

MPASM supports the entire instruction set of the PIC16C5X, PIC16CXX and PIC17CXX microcontrollers, as represented in the following four classes of instructions:

- Data Move Operations
- Arithmetic and Logical Operations
- Bit Manipulation Operations
- Special Control Operations

The Microchip microcontroller set is used to operate on data located in any of the file registers, including the I/O registers. There are:

- Data Transfer Operations
- Logical Operations
- Rotate Operations

MPASM provides bit level file register operations to manipulate and test individual bits in any addressable register, literal and control operations permitting operations on literals and branches to subroutines in program memory.

The Microchip microcontroller instruction sets allow read and write of special function registers such as the PC and status registers.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
SW165002	MPSIM/MPASM Simulator and Assembler Software and Documentation

© 1995 Microchip Technology Inc.

MPSIM Simulator

PIC16/17 Microcontroller Simulator

MPSIM is a discrete event simulator software application designed to imitate operation of the PIC16/17 microcontrollers. It allows the user to debug software that will use any of these microcontrollers.

At any instruction boundary, you may examine and/or modify any data area within the processor, or provide external stimulus to any of the pins. MPSIM gives you a solid, low cost, source-level debug tool to help you through the early design verification stages of your project.

MPSIM REQUIREMENTS

MPSIM requires an IBM PC/AT® or compatible computer running DOS version 5.0 or later. The PC needs a 3-1/2" floppy disk drive and at least 256K of main memory; MPSIM.EXE occupies roughly 150K. Recommended is a hard disk drive with 5 Mb of available storage.

MPSIM SIMULATOR

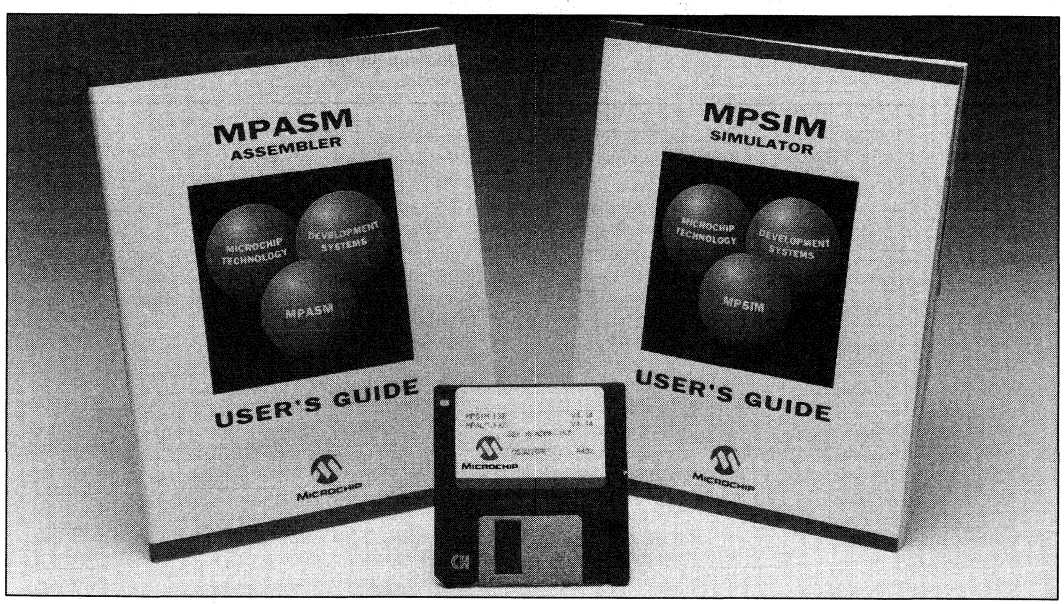
The MPSIM Simulator program provides the developer with an instruction and limited I/O simulator software program for debugging Microchip microcontroller assembler code.

The simulator is meant for use with smaller projects not requiring precise, more extensive development equipment. Many applications can be developed by using a simulator program alone.

The PIC16CXX and PIC17CXX families support various peripherals and interrupts. MPSIM generally simulates interrupts and most peripheral functions. However, certain peripheral functions are not supported (such as A/D conversion or serial I/O).

The MPSIM Simulator has the following features to assist in the debugging of software / firmware for the user:

5



IBM PC/AT is a registered trademark of IBM Corp.

© 1995 Microchip Technology Inc.

MPSIM Simulator

Program Load / Save

Commands exist to load assembled object file programs into simulation memory. Conversely, programs may be saved from program simulation memory back to the PC disk.

Display and Alter

Provisions are made to display and alter Program Memory, Register Files and status register bits. Also, simulator information such as cycle times, elapsed time, and step count can be displayed.

Disassembler

Program memory can be disassembled showing both hexadecimal data and instruction mnemonics for specified address ranges.

Utility Functions

Various utility functions exist which assist the user in operating the simulator. Memory and registers can be cleared by command. Memory can be searched to find occurrences of instructions, register use and ASCII data.

Symbolic Debugging

The simulator provides for symbolic referencing to aid and simplify debugging. The symbol table may be displayed. New symbols defined and unwanted symbols deleted.

Execution and Trace

During program execution, a number of items can be traced. Address ranges, registers and register contents and others.

Breakpoints

The user may specify up to 512 breakpoints at any one time.

Assembler Support

MPSIM works with Microchip's MPASM Universal Assembler.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
SW165002	MPSIM/MPASM Simulator and Assembler Software and Documentation

© 1995 Microchip Technology Inc.

MICROCHIP

MP-C

C Compiler

MP-C C COMPILER FOR PIC16/17

This product brief describes the technical aspects of the **MP-C** Code Development System for PIC16/17 micro-controllers developed by Byte Craft Limited.

The **MP-C** Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the PICMASTER Universal Emulator memory display (emulator software versions 1.13 and later).

MP-C is fast and efficient. You can quickly produce stand-alone single-chip microcontroller applications. These, taken with its other advantages make the Byte Craft **MP-C** Code Development System the first choice in intelligent compiler technology.

MP-C Requirements

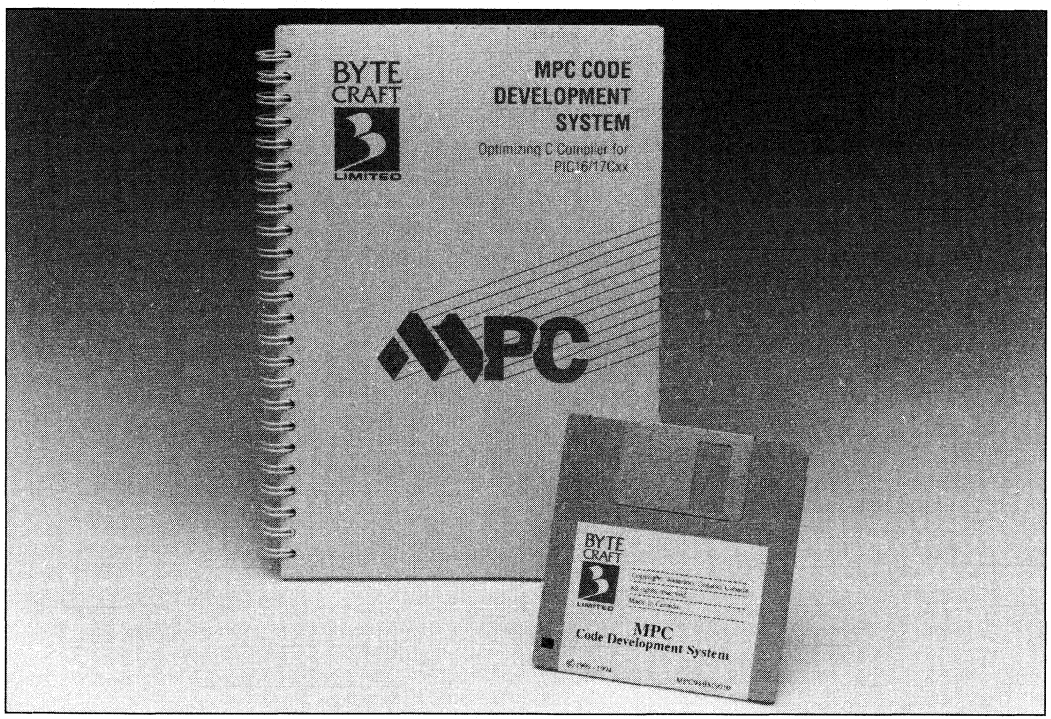
The compiler will run on any IBM PC, PC/XT®, PC/AT® or compatible computer, running DOS 5.0 or later.

MP-C Code Development System Features

MP-C supports the 12-bit PIC16C5X, the 14-bit PIC16CXX, and 16-bit PIC17CXX cores. It is a rule-based compiler with expert systems tailored to each of these platforms for optimal efficiency.

The compiler generates executable code directly from the compile process. There is no need for an extra step to assemble code generated by the compiler.

5



IBM PC/XT and PC/AT are registered trademarks of IBM Corporation.

© 1995 Microchip Technology Inc.

MP-C has the following features to assist in developing PIC16/17 software for specific user applications:

- Provides Object, Listing, Symbol and special files required for debugging with other Microchip Development systems.
- Supports interrupt routines
- Checks source against target hardware definitions
- Generates efficient, tight object code
- Includes a linker and built-in macro assembler
- 'C' enhancements specific to the PIC16/17 families' instruction sets.
- Output formats: INHX8S, INHX8M, and INHX32.

MP-C Microprocessor Specific Extensions

The **MP-C** Code Development System includes common 'C' enhancements such as ROM arrays, binary constants and case statements together with functions specific to the PIC16/17 architecture.

- **Binary Constants** of the form 0b0101110 which are logical extensions to the conventional 0x1a3b style of hexadecimal constants. You may also use 0B as leading characters.
- **Case Statements** are supported well by the PIC16/17 instruction set and the compiler provides a superset of the standard 'C' case statement. For example, case 4,5:, case '0'..'9', and complex case statements are allowed.
- **Processor Specific Functions** that are specific to the PIC16/17 family. For example NOP() and SLEEP() produce the equivalent PIC16/17 instruction.
- **"At" or @ Extension** allows you to fix a variable to a specific address in memory, for example: int N @ 0x0C.

SALES AND SUPPORT

The **MP-C** Code Development System is supplied directly by Byte Craft Limited of Waterloo, Ontario, Canada.

If you have any questions please contact your regional Microchip FAE or Microchip technical support personnel at (602) 786-7627.

© 1995 Microchip Technology Inc.

MICROCHIP

*fuzzy*TECH®-MP

Fuzzy Logic Development System for PIC16/17

*fuzzy*TECH-MP FOR MICROCHIP PIC16/17

This product brief describes the technical aspects of the *fuzzy*TECH-MP Fuzzy Logic Development System for PIC16/17 microcontrollers developed by INFORM Software Corporation specifically for Microchip.

The *fuzzy*TECH-MP Development System comes in two versions. The first, the Explorer, contains everything you need to gain a comprehensive working knowledge about fuzzy-logic system design. It is easy-to-use, all graphic editors and tools guide you step-by-step through the development phases of fuzzy systems. The Explorer supports two input variables and one output variable.

The full-featured *fuzzy*TECH-MP Edition offers all of the capabilities of the Explorer, plus it has the additional flexibility of eight input variables and four output variables for designing more complex systems. The full features are enabled with a hardware key lock attached to the parallel port of the PC.

Included in both versions is *fuzzy*LAB™, a fully functional demonstration board, to give customers hand-on experience with fuzzy logic systems implementation. *fuzzy*LAB is a simple heating thermostat consisting of a PWM-controlled resistor configured to heat a thermistor to a preset temperature. Using the two fuzzy algorithms provided, a designer can set a target temperature and observe the thermostat response to the set point.

Both systems generate assembly code compatible with the MPASM, Microchip's Universal Assembler, that can be integrated into your application. Examining this code provides you with further insights into the fabrics of fuzzy logic systems.

5

© 1995 Microchip Technology Inc.

*fuzzy*TECH-MP System Requirements

*fuzzy*TECH-MP will run on any IBM PC® (386 or higher) or compatible computer, running DOS 4.1 or later, and Microsoft Windows® 3.0 or later. Because *fuzzy*TECH-MP makes extensive use of graphics, a color graphic monitor (VGA) is required, and higher resolutions of 800 x 600 or 1024 x 768 are recommended.

What is Fuzzy Logic?

Fuzzy logic is a technology that enhances mode-based system designs using both intuition and engineering heuristics. Fuzzy logic uses elements of everyday language to represent desired system behavior, thus circumventing the need for rigorous mathematical modeling.

It is an efficient way of designing, optimizing and maintaining highly complex systems transparently.

Fuzzy Logic Applications

Fuzzy logic finds its home in unique applications:

- When no adequate mathematical model for a given problem is readily apparent.
- When non-linearities, time constraints or multiple parameters exist.
- When engineering know-how about the given problem is available or can be acquired during the design process.

The *fuzzy*TECH-MP Implementation

*fuzzy*TECH-MP provides the following standard features:

- Windows Compatible with full graphical user interface
- 8-Input variables (2 for the Explorer version)
- 4-Output variables (1 for the Explorer version)
- 8-Bit resolution on input and output variables
- 16-Bit computation resolution for the PIC16CXX and PIC17CXX microcontrollers
- No theoretical limit on rules, antecedents and linguistic conjunctions (chip limitations will place a practical limit on these)
- MAX-MIN and MAX-DOT inference methods
- CoM and MoM defuzzification methods
- MPASM Compatible
- PICMASTER® Compatible

*fuzzy*TECH-MP

*fuzzy*TECH-MP is available directly from Microchip Technology and its authorized distributors. Contact your local sales office for more information.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DV005001	*fuzzy*TECH-MP EXPLORER
DV005002	*fuzzy*TECH-MP EDITION

*fuzzy*TECH is a registered trademark of INFORM Software Corporation.
PICMASTER is a registered trademark of Microchip Technology Inc. *fuzzy*LAB is a trademark of Microchip.
IBM-PC is a registered trademark of IBM Corporation.
Microsoft Windows is a registered trademark of Microsoft Corporation.

© 1995 Microchip Technology Inc.

MICROCHIP

SECTION 6
SHORTFORM CATALOG TO OTHER MICROCHIP PRODUCTS

6

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

MICROCHIP

MEMORY

Shortform Catalog to Non-Volatile Memory Products

The following shortform catalog will provide an overview to Microchip Technology Inc.'s Serial EEPROM, Parallel EEPROM and EPROM products. This includes: I^2C™ Serial EEPROM, Microwire™ Serial EEPROM, Specialty Serial EEPROM and EPROM devices with complementary development systems.

For detailed information on these non-volatile memory devices and development systems, please refer to the ***Microchip Non-Volatile Memory Products Data Book.*** (A copy can be obtained by calling your local Microchip sales office.)

6

© 1995 Microchip Technology Inc.

I²C™ SERIAL EEPROM PRODUCT SPECIFICATIONS*

Density	Organization	Part Number	Supply Voltage	Package	Features
1K Bit	1024 bits (128 x 8)	**24AA01**	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC	8-byte page, maximum write time 10 ms/page, 10 million erase/write cycles, 100/400KHz clock, hardware write-protect.
2K Bit	2048 bits	**24AA02**	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC	8-byte page, maximum write time 10 ms/page, 1 million erase/write cycles, 100/400KHz clock, hardware write-protect.
4K Bit	4096 bits (512 x 8)	**24AA04**	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10 ms/page, 1 million erase/write cycles, 100/400KHz clock, hardware write-protect.
8K Bit	8192 bits (1K x 8)	**24AA08**	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10ms/page, 1 million erase/write cycles, 100/400KHz clock, hardware write-protect.
16K Bit	16384 bits (2K x 8)	**24AA16**	+1.8V to +5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10ms/page, 10 million erase/write cycles, 100/400KHz clock, hardware write-protect.
16K Bit	16384 bits (2K x 8)	**24AA164**	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC	16 byte page, cascadable up to 8 devices on same bus, up to 400KHz clock. 10 million erase/write cycles.
32K Bit	32768 bits (4K x 8)	**24AA32**	+1.8V to 6.0V	8-Lead PDIP 8-Lead SOIC	64 byte input cache, maximum write time 10ms/page (8-byte page), variable page size, 10 million erase/write cycles for high endurance block, up to 400 KHz clock, up to 8 devices on same bus.
64K Bit	64736 bits (8K x 8)	**24AA65**	+1.8V to 6.0V	8-Lead PDIP 8-Lead SOIC	64 byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, up to 400 KHz clock, up to 8 devices on same bus.
1K Bit	1024 bits (128 x 8)	**24LC01B***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	8-byte page, maximum write time 10ms/page, 10 million erase/write cycles, 400KHz clock, hardware write-protect.
2K Bit	2048 bits	**24LC02B***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	8-byte page, maximum write time 10ms/page, 1 million erase/write cycles, 400KHz clock, hardware write-protect.
4K Bit	4096 bits (512 x 8)	**24LC04B***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10ms/page, 1 million erase/write cycles, 400KHz clock, hardware write-protect.
8K Bit	8192 bits (1K x 8)	**24LC08B***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10ms/page, 1 million erase/write cycles, 400KHz clock, hardware write-protect.
16K Bit	16384 bits (2K x 8)	**24LC16B***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	16-byte page, maximum write time 10ms/page, 10 million erase/write cycles, 400KHz clock, hardware write-protect.

*Available in standard commercial and industrial temperature ranges.

© 1995 Microchip Technology Inc.

I²C™ SERIAL EEPROM PRODUCT SPECIFICATIONS* (CONTINUED)

Density	Organization	Part Number	Supply Voltage	Package	Features
16K Bit	16384 bits (2K x 8)	**24LC164***	+2.5V to 5.5V	8-Lead PDIP 8-Lead SOIC	16 byte page, cascadable up to 8 devices on same bus, up to 400KHz clock. 10 million erase/write cycles.
32K Bit	32768 bits (4K x 8)	**24LC32***	+2.5V to +6.0V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, up to 400KHz clock, up to 8 devices on same bus, split endurance. 10 million erase/write cycles for high endurance block.
64K Bit	64736 bits (8K x 8)	**24LC65***	+2.5V TO +6.0V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, software write-protect, up to 400KHz clock, up to 8 devices on same bus.
1K Bit	1024 bits (128 x 8)	**24C01A****	+5.0V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only,** other applications should use 24LC01B. 2-byte page, typ. write time 1ms per byte, 1 million erase/write cycles.
2K Bit	2048 bits	**24C02A****	+5.0V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only,** other applications should use 24LC02B. 2-byte page, typ. write time 1ms per byte, 1 million erase/write cycles.
4K Bit	4096 bits (512 x 8)	**24C04A****	+5.0V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	**For fast byte write (1ms) or E-temp applications only,** other applications should use 24LC04B. 8-byte page, typ. write time 1ms per byte, 1 million erase/write cycles.
8K Bit	8192 bits (1K x 8)	**24C08B****	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	Available in automotive (E) temp. only.
16K Bit	16384 bits (2K x 8)	**24C16B****	+4.5 to +5.5V	8-Lead PDIP 8-Lead SOIC	Available in automotive (E) temp. only.
32K Bit	32768 bits (4K x 8)	**24C32***	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, up to 400KHz clock, up to 8 devices on same bus, split endurance. 10 million erase/write cycles for high endurance block.
64K Bit	64736 bits (8K x 8)	**24C65***	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, software write-protect, up to 400 kHz clock, up to 8 devices on same bus.
1K Bit	1024 bits (128 x 8)	**85C72****	+5.0V	8-Lead PDIP 8-Lead SOIC	2-byte page, typical write time 1ms per byte, 1 million erase/write cycles, 100KHz clock, can connect up to 8 devices on the same bus. CMOS version of PCD8572.

*Available in standard commercial and industrial temperature ranges.
**Available in standard commercial, industrial and automotive temperature ranges.

© 1995 Microchip Technology Inc.

I²C™ SERIAL EEPROM PRODUCT SPECIFICATIONS* (CONTINUED)

Density	Organization	Part Number	Supply Voltage	Package	Features
2K Bit	2048 bits	**85C82****	+5.0V	8-Lead PDIP 8-Lead SOIC	2-byte page, typical write time 1ms per byte, 1 million erase/write cycles, can connect up to 8 devices on the same bus. CMOS version of PCD8582.
4K Bit	4096 bits (512 x 8)	**85C92****	+5.0V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	8-byte page, typical write time 1ms per byte, 1 million erase/write cycles, can connect up to 4 devices on the same bus and hardware write-protect. Memory upgrade of PCD8582.

For higher densities, see Smart Serial section on page 6.

MICROWIRE™ SERIAL EEPROM PRODUCT SPECIFICATIONS*

Density	Organization	Part Number	Supply Voltage	Package	Features
1K Bit	128 x 8 or 64 x 16	**93AA46*****	+1.8V to +5.5V	8-Lead PDIP 8-Lead SOIC	Hardware READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 1 million erase/write cycles, and sequential read function.
2K Bit	256 x8 or 128 x 16	**93AA56*****	+1.8V to +5.5V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, and sequential read function.
4K Bit	512 x 8 or 256 x 16	**93AA66*****	+1.8V to +5.5V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, and sequential read function.
1K Bit	128 x 8 or 64 x 16	**93LC46***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Hardware READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 1 million erase/write cycles, and sequential read function.
2K Bit	256 x8 or 128 x 16	**93LC56***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, and sequential read function.
4K Bit	512 x 8 or 256 x 16	**93LC66***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, and sequential read function.
1K Bit	64 x 16	**93LC46B***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Hardware READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 1 million erase/write cycles, sequential read function, and x16 organization hard-wired by factory.
2K Bit	128 x 16	**93LC56B***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, sequential read function, and x16 organization hard-wired by factory.

*Available in standard commercial and industrial temperature ranges.
**Available in standard commercial, industrial and automotive temperature ranges.
***Available in commercial temperature range.

© 1995 Microchip Technology Inc.

MICROWIRE™ SERIAL EEPROM PRODUCT SPECIFICATIONS* (CONTINUED)

Density	Organization	Part Number	Supply Voltage	Package	Features
4K Bit	256 x 16	**93LC66B***	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC	Software READY/BUSY signal, 2MHz max. clock rate, 4ms per word typical write time, 10 million erase/write cycles, sequential read function, and x16 organization hard-wired by factory.
256 Bit	16 x 16	**93C06****	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only. All other applications see 93LC46.** Software READY/BUSY signal. 1MHz max. clock rate.
1K Bit	64 x 16 or 256 x 16	**93C46****	+4.0V to +5.5V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only. All other applications see 93LC46.** Software READY/BUSY signal, 1MHz max. clock rate.
2K Bit	256 x 8 or 256 x 16	**93C56****	+4.0V to +5.5V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only. All other applications see 93LC56.** Software READY/BUSY signal, 2MHz max. clock rate. 1 million erase/write cycles.
4K Bit	512 x 8 or 256 x 16	**93C66****	+4.0V to +5.5V	8-Lead PDIP 8-Lead SOIC	**For fast byte write (1ms) or E-temp applications only. All other applications see 93LC66.** Software READY/BUSY signal, 2MHz max. clock rate. 1 million erase/write cycles.

6

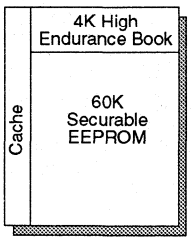
SPECIALTY SERIAL EEPROM PRODUCT SPECIFICATIONS

Density	Organization	Part Number	Supply Voltage	Package	Features
16K Bit	16384 bits (2K x 8)	**24AA174*****	+1.8V to 5.5V	8-Lead PDIP 8-Lead SOIC	16K bit + 16 byte OTP security page. 400KHz clock, 10 million erase/write cycles guaranteed, cascadable up to 8 devices on same bus.
		24LC174*	+2.5V to 5.5V	8-Lead PDIP 8-Lead SOIC	16K bit + 16 byte OTP security page. 400KHz clock, 10 million erase/write cycles guaranteed, cascadable up to 8 devices on same bus.
1K Bit	1024 bits (Dual Clock)	**24LC21***	+2.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	Dual mode operation. 400KHz clock. Maximum Write time 10ms/page. 1 million erase/write cycles. DDC1/DDC compatible.

*Available in standard commercial and industrial temperature ranges.
**Available in standard commercial, industrial and automotive temperature ranges.
Available in commercial temperature range.

© 1995 Microchip Technology Inc.

SPECIALTY SERIAL EEPROM PRODUCT SPECIFICATIONS (CONTINUED)

Density	Organization	Part Number	Supply Voltage	Package	Features
64K Bit	8K x 8 	**24AA65**	+1.8V to +6.0V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, software write-protect, up to 400 kHz clock, up to 8 devices on same bus.
		24LC65*	+2.5V TO +6.0V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, software write-protect, up to 400 kHz clock, up to 8 devices on same bus.
		24C65*	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	64-byte input cache, maximum write time 10ms/page (8-byte page), variable page size, split-endurance, 10 million erase/write cycles for high endurance block, software write-protect, up to 400 kHz clock, up to 8 devices on same bus.
1K Bit	128 x 8 or 64 x 16	**59C11****	+4.5V to +5.5V	8-Lead PDIP 8-Lead SOIC	4-wire bus, external READY/BUSY signal. 1MHz max. clock rate, max. 2ms per word max. write time and 1 million erase/write cycles.
2K Bit	128 x 16	**93LCS56**	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	Software write protection, 2MHz max. clock rate, 4ms per word typical write time, 1 million erase/write cycles, sequential read function and x16 organization hard-wired by factory.
4K Bit	256 x 16	**93LCS66**	+2.0V to +6.0V	8-Lead PDIP 8-Lead SOIC 14-Lead SOIC	Software write protection signal, 2MHz max. clock rate, 4ms per word typical write time, 1 million erase/write cycles, sequential read function and x16 organization hard-wired by factory.

PARALLEL EEPROM PRODUCT SPECIFICATIONS

Density	Organization	Part Number	Access Time (ns)	Package	Features
4K Bit	512 x 8	**28C04A** **28C04A** **28C04A**	250 200 150	24-Lead PDIP 32-Lead PLCC	100,000 erase/write cycles, 1ms write cycle time, DATA POLLING signal, fast byte write time (200µs) option.
16K Bit	2K x 8	**28C16A** **28C16A** **28C16A**	250 200 150	24-Lead PDIP 28-Lead TSOP 28-Lead VSOP 32-Lead PLCC	100,000 erase/write cycles, 1ms write cycle time, DATA POLLING signal, fast byte write time (200µs) option.
16K Bit	2K x 8	**28C17A** **28C17A** **28C17A**	250 200 150	28-Lead PDIP 28-Lead SOIC 28-Lead TSOP 28-Lead VSOP 32-Lead PLCC	100,000 erase/write cycles, 1ms write cycle time, DATA POLLING signal, and READY/BUSY signal, fast byte write time (200µs) option.

*Available in standard commercial and industrial temperature ranges.
**Available in standard commercial, industrial and automotive temperature ranges.

© 1995 Microchip Technology Inc.

PARALLEL EEPROM PRODUCT SPECIFICATIONS (CONTINUED)

Density	Organization	Part Number	Access Time (ns)	Package	Features
64K Bit	8K x 8	**28C64A**	250	28-Lead PDIP	100,000 erase/write cycles, 1ms write cycle time, DATA POLLING signal and READY/BUSY signal, fast byte write time (200µs) option.
		28C64A	200	28-Lead SOIC	
		28C64A	150	28-Lead TSOP	
				28-Lead VSOP	
				32-Lead PLCC	

Note: Some package/speed/temperature combinations may not be available. Please consult your authorized Microchip Representative.

EPROM PRODUCT SPECIFICATIONS

Density	Organization	Part Number	Access Time (ns)	Operation	Supply Voltage	Package	Features
64K Bit	8K x 8	**27C64-25**	250	Static	+5.0V	28-Lead DIP, PDIP	Factory programming available.
		27C64-20	200			32-Lead PLCC	
		27C64-17	170			32-Lead LCC	
		27C64-15	150			28-Lead SOIC	
		27C64- 12	120			28-Lead TSOP	
64K Bit	8K x 8	**27LV64-30**	300	Static	+3.0V to 5.5V	28-Lead DIP, PDIP	27C64 compatible
		27LV64-25	250			32-Lead PLCC	
		27LV64-20	200			32-Lead LCC	
						28-Lead SOIC	
						28-Lead TSOP	
						28-Lead VSOP	
128K Bit	16K x 8	**27C128-25**	250	Static	+5.0V	28-Lead DIP, PDIP	Factory programming available.
		27C128-20	200			32-Lead PLCC	
		27C128-17	170			32-Lead LCC	
		27C128-15	150			28-Lead SOIC	
		27C128-12	120				
256K Bit	32K x 8	**27C256-20**	200	Static	+5.0V	28-Lead DIP, PDIP	Factory programming available.
		27C256-15	150			32-Lead PLCC	
		27C256-12	120			32-Lead LCC	
		27C256-10	100			28-Lead SOIC	
		27C256-90	90			28-Lead TSOP	
						28-Lead VSOP	
256K Bit	32K x 8	**27HC256-90**	90	Static	+5.0V	28-Lead DIP, PDIP	27C256 compatible
		27HC256-70	70			32-Lead PLCC	
		27HC256-55	55			32-Lead LCC	
						28-Lead SOIC	
						28-Lead TSOP	
						28-Lead VSOP	
256K Bit	32K x 8	**27LV256-30**	300	Static	+3.0V to 5.5V	28-Lead DIP, PDIP	27C256 compatible
		27LV256-25	250			32-Lead PLCC	
		27LV256-20	200			32-Lead LCC	
						28-Lead SOIC	
						28-Lead TSOP	
						28-Lead VSOP	
256K Bit	16K x 16	**27HC1616-70**	70	Static	+5.0V	40-Lead DIP	JEDEC approved worldwide pinout
		27HC1616-55	55			44-Lead LCC	

6

© 1995 Microchip Technology Inc.

EPROM PRODUCT SPECIFICATIONS (CONTINUED)

Density	Organization	Part Number	Access Time (ns)	Operation	Supply Voltage	Package	Features
512K Bit	64K x 8	**27C512A-15** **27C512A-12** **27C512A-10** **27C512A-90** **27C512A-70**	150 120 100 90 70	Static	+5.0V	28-Lead DIP, PDIP 32-Lead PLCC 32-Lead LCC 28-Lead SOIC 28-Lead TSOP 28-Lead VSOP	Factory programming available.
512K Bit	64K x 8	**27LV512-30** **27LV512-25** **27LV512-20**	300 250 200	Static	+3.0V to 5.5V	28-Lead DIP, PDIP 32-Lead PLCC 32-Lead LCC 28-Lead SOIC 28-Lead TSOP 28-Lead VSOP	27C512 compatible
256K Bit	32K x 8	**27HC256L-90**	90	Static	+5.0V	28-Lead DIP, PDIP 32-Lead PLCC 32-Lead LCC 28-Lead SOIC 28-Lead TSOP 28-Lead VSOP	27C256 compatible

SERIAL EPROM PRODUCT SPECIFICATIONS

Density	Organization	Part Number	Operation	Supply Voltage	Package	Features
36K Bit	36K x 1	**37LV36**	Static	+3.0 to 6.0	8-Lead SOIC 8-Lead PDIP 20-Lead PLCC	Operational equivalent to Xilinx XC1736
65K Bit	65K x 1	**37LV65**	Static	+3.0 to 6.0	8-Lead SOIC 8-Lead PDIP 20-Lead PLCC	Operational equivalent to Xilinx XC1765
128K Bit	128K x 1	**37LV128**	Static	+3.0 to 6.0	8-Lead SOIC 8-Lead PDIP 20-Lead PLCC	Operational equivalent to Xilinx XC17128

Note: Some package/speed/temperature combinations may not be available. Please consult your authorized Microchip Representative.

DEVELOPMENT SYSTEMS

Description	Part Number	Features
Total Endurance™ Disk	**SW242001**	Application oriented, predictive software model. Models the performance of erase/write cycle endurance (FIT, PPM levels and operating life) of Microchip Serial EEPROMs with DOS or Windows interface.
Serial [Designer's Kit] EEPROM	**DV243001**	Complete designer's kit for systems using Microchip's Serial EEPROMs. **Designer's Kit** includes Total Endurance Disk, SEEVAL™ programming and evaluation board, *Serial EEPROM Handbook*, power supply and all necessary cabling. • RS-232 connection to IBM® compatible PC • Windows and DOS based software to erase, write and fully exercise all Microchip Serial EEPROMs • Special Functions: user selectable security, programmable endurance and special pinouts • All Microchip Serial EEPROMs from 256 bits to 64K bits supported

© 1995 Microchip Technology Inc.

Shortform Catalog to Application-Specific Standard Products

The following shortform catalog will provide an overview to Microchip Technology Inc.'s application-specific standard products including the TrueGauge™ Intelligent Battery Management Family, PC pointing device family, energy management controller, PICSEE 8-bit microcontrollers and development systems.

For further information on application-specific standard products, please refer to individual product data sheets. (Copies can be obtained by calling your local Microchip sales office.)

© 1995 Microchip Technology Inc.

TRUEGAUGE™ INTELLIGENT BATTERY MANAGEMENT

	Function/Description	Part Number	Package	Features
Integrated Battery Capacity Monitoring and Charge Controller	The **MTA11200 TrueGauge Battery "Fuel Gauge" and Charge Controller IC** is a simple full-featured solution to battery monitoring and charging. It is designed to operate with either NiCd, NiMH or lead acid battery packs. The MTA11200 digitally integrates battery charge and discharge current to determine the battery state of charge. The MTA11200 is ideally suited for use in portable computers, portable video equipment, cellular phones, and other products relying on rechargeable battery technology. It excels in applications where an accurate "fuel gauge" is desired to prevent interruption in use, or data loss due to insufficient battery power.	**MTA11200**	28 Lead	• Low-cost • Operates with NiCd, NiMH or lead acid battery pack • From 3.0 volts to 25VDC • Real-time RS-232 interface provides battery data on remaining capacity, total capacity, battery voltage, current and temperature • Five levels of overcharge protection • Automatic measurement of battery capacity and request of condition cycles • Logs battery information such as number of charge cycles, over temperature, under temperature, and over voltage conditions
Programmable Intelligent Battery I.C.	The **MTA14000** is a high performance mixed-signal microcontroller based on Microchip's powerful 8-bit RISC core that enables real-time measurement and processing of battery parameters including voltage, charge current, discharge current, temperature, and total number of cycles. It supports 4096 words of program memory, 192 bytes of RAM, 11 interrupts, 38 special function hardware registers and eight levels of hardware stack. The MTA14000 is ideally suited for use in smart battery controllers, battery chargers, uninterruptable power supply controllers, smart sensors, HVAC controllers and data acquisition.	**MTA14000**	28 Lead	• RISC core • 35 single word instructions • Fully code compatible with Microchip's standard PIC16/17 microcontroller family • 4K Program Memory, 192 bytes RAM, 11 interrupts, eight levels of stack • 8-channel analog-to-digital converter with programmable resolution up to 16 bits • Two 3-decade digital-to-analog converters • Multiple power down controls for analog circuits • Synchronous Serial Port compatible with I^2C™, ACCESS.bus™, System Management Bus • I/O pins with individual direction control allowing for support of any other communications interface such as RS-232 and one-wire

© 1995 Microchip Technology Inc.

TRUEGAUGE™ DEVELOPMENT TOOLS

Function/Description		Part Number	Features
TrueGauge Development Tool	The **MTA11200 TrueGauge Intelligent Battery Management IC** is support by a user friendly tool for system development. The DV114001 operates under Microsoft Windows™. This development tool enables the management of all phases of product development including inception, debugging and maintenance. System design verification can be accomplished before a hardware prototype needs to be built, thus reducing time and cost. The user interface provides a graphically-oriented development environment. The data logging feature saves measured data into a file that can be imported to Excel®.	**DV114001**	The TrueGauge development tool is a tool for system development under Windows. The development tool kit contains the following: • NiCd battery with TrueGauge module • NiMH battery with TrueGauge module • Stand-alone TrueGauge module • Charger/Discharger Interface Board • Universal power supply with power cord • PC Interface Cable with DB9-DB25 converter • Design/Verification software on a 3.5" diskette • MTA11200 and 24LC01B product samples • MTA11200 data sheet • *TrueGauge Development Tool User's Guide*

PC POINTING DEVICES

Function/Description		Part Number	Package	Features
Mouse Controller	The **MTA41XXX** Mouse Controllers are the heart of a simple, low-cost mouse or trackball solution. The MTA41XXX family supports all Apple® Computer and IBM® PC-compatible formats.	**MTA41300**	18 Lead	Low-cost mouse controller with support for IBM PS/2®-compliant or Microsoft® serial-format-compatible.* The MTA41300 controller supports 2-button mouse or trackball operation. Packaging is available in 300 mil wide PDIP and SOIC
		MTA41110	18 Lead	Low-cost, low power mouse controller with complete support for IBM PS/2 interface format. Like the MTA41300, the MTA41110 controller supports 2-button mouse or trackball operation, but unique software features of the MTA41110 allow for direct input from optical encoders without the need for external comparators. LED strobing is also supported by the MTA41110 for low-power applications.
		MTA41120	18 Lead	Same as MTA41110 except offers complete support for Apple Computer ADB™ interface.

* The code in this product was not developed or licensed by Microsoft Corporation.

6

© 1995 Microchip Technology Inc.

ENERGY MANAGEMENT DEVICES

	Function/Description	Part Number	Package	Features
Energy Management Controller	The **MTE1122** Energy Management Controller combines Microchip's proprietary PIC16/17 8-bit RISC microcontroller technology with a unique, patent pending power management firmware algorithm in a single package. This device, by monitoring and controlling the supply requirements into an AC induction motor, effectively reduces the power consumed by the motor. The MTE1122 is available in both plastic DIP and space-saving SOIC packages, and operates over commercial and industrial ranges.	**MTE1122**	18 Lead	• Low cost • Reduces power consumption of AC induction motors • Protects against brownouts and power surges

© 1995 Microchip Technology Inc.

PICSEE FAMILY OF MICROCONTROLLERS WITH SERIAL EEPROM

Function/Description		Part Number	Package	Features
8-Bit Microcontroller with Serial EEPROM	The **PICSEE™** offers the unique combination of an EPROM-based microcontroller and a Serial EEPROM in a single package. The MTA81010 is a multi-chip module that integrates a PIC16C54 low-cost, high-performance, 8-bit, fully-static, EPROM-based CMOS microcontroller with a 24LC01B 1024-bit Serial EEPROM configured 128K x 8. The integration of these two popular chips into a single package reduces system cost, board area and inventory.	**MTA81010**	28 Lead	• 512 x 12 EPROM • 1K EEPROM • 32 bytes of RAM registers • 12 bidirectional I/O lines • RTCC timer/counter • Free running watchdog timer and load protection fuse • EEPROM is configured as 8-byte page • Maximum write time 10ms • 100K erase write cycles minimum • 400KHz clock, hardware write-protect • Available in the oscillator configurations RC, XT, LP • Frequency range: from 25KHz through 4MHz down to 2.5 volt operation • Temperature ranges: 0°C to +70°C, –40°C to +85°C, and –40°C to +125°C • Packaging: available in 600 mil wide PDIP, CERDIP (Windowed), SSOP and 300 mil wide SOIC
		MTA85XXX	20 Lead SSOP	• 512 or 2048 x 12 on-chip EPROM • 1K or 2K EEPROM • 25 or 72 x 8 general purpose registers (SRAM) • 7 special function hardware registers • 12 I/O pins with individual direction control • 8-bit real time clock/counter (RTCC) with 8-bit programmable prescaler • Clock frequencies available: 4MHz, 10MHz

© 1995 Microchip Technology Inc.

PICSEE FAMILY DEVELOPMENT TOOLS

	Function/Description	Part Number	Package	Features
PICSEE Development Kit	**PICSEESTART** is a very low-cost entry-level development system for the PICSEE microcontroller. It is a combination of the PICSEE Adapter Kit and the PICSTART-16B1 Development Kit.	**DV813001 PICSEESTART-81A**	28 Lead	• PICSEE Adapter Kit • PICSTART-16B1 • Product sample kit • Complete system documentation
		DV853001 PICSEESTART-85A	20 Lead	• PICSEE Adapter Kit • PICSTART-16B1 • Product sample kit • Complete system documentation

PICSEE FAMILY PROGRAMMERS

	Function/Description	Part Number	Package	Features
PICSEEKIT	The **PICSEEKITs** are programmer adapters for use in conjunction with PRO MATE or PICSTART programmers. Included is the in-circuit emulation adapter board for the PICMASTER-16A.	**AC812001 PICSEEKIT-81A**	28 Lead	Supports programming of all PDIP, SOIC and JW MTA81010 devices.
		AC814003	28 Lead	Programming adapter sockets for DIP and SOIC devices.
		AC852001 PICSEEKIT-85A	20 Lead	MTA85XXX programming adapter and emulation kit.
		AC854001	20 Lead	20-lead SSOP programming adapter socket.

© 1995 Microchip Technology Inc.

SECTION 7
QUALITY, RELIABILITY AND ENDURANCE

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

MICROCHIP

Product Quality

A CORPORATE COMMITMENT

Microchip Technology Inc. has evolved a culture where a commitment to quality is an integral part. By empowering every employee to be responsible for the quality of their work, the entire corporation is involved in the quality process. This interaction creates an environment for continuous improvement throughout the organization. The benefits of the system are then not only enhanced product quality and reliability but also product services.

THE CHALLENGE OF COMPLEXITY

Integrating an Ideal

Microchip's quality programs and business plan are vertically integrated and touch all levels of the company. From the top down, the President and CEO actively lead programs to ensure continuous improvement is a perpetual process. Improvement and cross functional teams work to enhance performance at every department level. Incorporating the improvement objectives into the business plan creates a unity of purpose and mandates that the two merge as one measurement.

Determination to be the Best

A fundamental concept at Microchip is the commitment to continuous improvement. All areas are constantly looking for ways to improve every aspect of the company. This has allowed products and processes to become world class in quality and reliability. These programs are the foundation for success.

PROCESS TECHNOLOGY

All the products manufactured at Microchip make use of a common N-Well CMOS baseline process to which modules are added in order to create the specific functions required by the product (EEPROM, Microcontroller, Logic and EPROM).

The baseline process, which has been in Manufacturing for the last 8 years, uses minimum dimensions of 1.5µm, 360Å gate oxide thickness, N+ doped polysilicon gates and arsenic implanted source-drain diffusions for the N-channel devices.

A more advanced process uses minimum dimensions of 1µm, 250Å gate oxide thickness, polycide gate and LDD junction for the N-channel devices. A double level metal module can be added to both processes.

All of these devices utilize a proprietary passivation suitable for a wide variety of package types. Microchip's processes have been developed with reliability and manufacturability as their primary goals.

EEPROM

Microchip's CMOS floating gate EEPROM technology produces a non-volatile memory cell by storing or removing charge from the floating gate. Charge is transferred bidirectionally to the floating gate by Fowler-Nordheim tunneling through a sub-10 nm oxide over the drain of the transistor. This technology produces a memory cell with a typical endurance of $> 10^7$ cycles and greater than 40 years of data retention. (See EEPROM application note for details).

EPROM

This technology uses a non-volatile memory cell which stores charge on a self aligned floating gate. Electrons are provided to the floating gate via hot electron injection from the drain depletion region. Each byte can typically be programmed in 100 microseconds, and can retain that data for more than 10 years with unlimited reads. Block erasing is accomplished with a high intensity UV source through the package window. Windowed parts can be erased and reprogrammed more than 100 times.

Microcontroller and Logic

Logic products are built on a variety of Microchip's processes and their derivatives. These products have process modules for production of controllers that feature ROM, Analog, EPROM, and EEPROM. By utilizing the standard processing modules, the designs meld these technologies and their flexibility while maintaining the high quality and reliability standards expected.

QUALITY

Design for Quality and Reliability

Product reliability is designed into all Microchip processes and products. Design margins are established to guarantee every product can be produced economically, error-free and within the tolerances of the manufacturing process. Product Introduction Teams representing manufacturing, engineering, quality and product divisions ensure that exacting standards are met for each specific product.

© 1995 Microchip Technology Inc.

Documentation and Procurement Specifications

Microchip's documentation control program assures the correct and current document always is available at the point of use. Active documents are revision coded and serialized. Procurement specifications bear the same requirements. These document control procedures, which are common in the industry for military and high reliability products, are employed by Microchip, system wide.

In Line Controls and Process Assessment

Product integrity is assured by sampling and inspection plans performed in-line. This enables Microchip to control and improve product quality levels as product moves through the manufacturing operation. Microchip's acceptance sampling plans in assembly emphasize the attempt to eliminate defective product as it is discovered. Acceptance and sampling plans are based on proprietary low fraction defective (<1000ppm) quality statistics.

To determine whether a process is within normal manufacturing variation, statistical techniques are put to work at selected process steps. In-process controls are performed by operators in the wafer fabrication and assembly operations. Operators take immediate corrective action if a process step is out of its control limit. Through these in-line controls the true capability of a process is generated. (See Appendix A:In Line Controls)

Control of Customer Quality is attained through a statistical program based on minimum defect capability levels. These levels are defined as the error levels associated with the circuit design and science limitations of the chemistry and physics of processing.

Material controls prevent defective piece parts from getting into the line. Microchip's assembly material control sample plan is typical of the emphasis placed on safeguards. (See Appendix B:Material Controls Package).

Testing for Margin

Microchip conducts a product's initial test under stringent requirements. All quality assurance tests are run to tighter limits than customer specifications. As part of an outgoing quality assurance program, products are tested at least two machine tolerances tighter than those limits specified by the customer on every parameter. Margin testing accounts for normal tolerances of any particular test system and provides the assurance that Microchip's products meet a customer's specifications.

Variation from Expectation

Microchip works to make variation from target as small as possible. The better process is the one that holds the narrowest dispersion. Processes are targeted to maintain Cpk's of >1.5 and currently have typical values of >2.0. Higher process capability values are continually strived for indicating that better process control is being obtained.

Outgoing Quality

Quality Control samples all outgoing product from final testing. These samples measure in-line defect levels after screens have been applied. Root cause analysis follows, initiating technical change to effect continuous improvement.

Programmability Yield

Using programmable devices adds a complexity to the Quality Level interpretation. It is not unlikely that some programmable devices will not program. The programmability yield is dependent on (but not limited to): programmers, technology, array size, and handling.

Any device that does not program properly will not be used in the end system. Therefore, programmability yields should not be used to calculate AQLs.

For convenience, Microchip offers programming services for certain devices. This service is an advantage to the customer since it not only eliminates programmability rejects, but also reduces the handling of the parts. See the individual data sheets for details on our Quick-Turnaround-Production (QTP) service.

RELIABILITY

Process Qualification

No priority is more important than the one where processes under which Microchip products are built operate without fail. All products are stressed beyond normal use limits when undergoing high temperature operating life and retention bake tests. This is done to ensure that the devices meet the strictest reliability guidelines and will maintain industry low failure rates.

Package Qualification

Package qualification measures a component's ability to withstand extreme thermal and mechanical stresses. All products are stressed to high level industrial specifications to ensure reliability.

Ongoing Sampling of Key Reliability Variables

Microchip conducts accelerated mechanical tests, operating life tests and memory retention tests to explore the many ways failures might occur. Data obtained from continuous testing is used to identify potential reliability problems and for defining action courses to improve product. Microchip's reliability knowledge is shared with customers. This data is available for use in customer's own quality and reliability improvement programs and is published in regular quarterly and yearly reports.

© 1995 Microchip Technology Inc.

RELIABILITY CONCEPTS

Definition

Reliability is the probability of a system or circuit performing its predefined function adequately under specific conditions for a given period of time. Thus, the reliability of a microcircuit is a function of both stress conditions and the time of operation.

The reliability (or probability of survival) range runs from 0 (no chance of survival) to 1 (no chance of failure). Current microelectronic circuits are manufactured and controlled to such tight specifications that reliability figures for the total operation time approaching 1 (i.e., 0.9999) are common. As a result, the complement of reliability, or the failure probability, is more often quoted in current literature.

The failure rate is the rate at which failures occur on units surviving to a specific number of hours of operation. Failure rates per unit circuit-hour would generally be very small. To avoid reporting such small numbers, failure rates have been defined for greater circuit-hours. One thousand circuit-hours is defined as one circuit operating for one thousand hours, or 1,000 circuits operating for 1 hour, etc. The numbers of circuit-hours is the number of circuits multiplied by the number of operation hours for each circuit.

Two methods to define failure rate are commonly used:

- Percent failures per thousand circuit-hours
- Absolute failures per billion circuit-hours, or FITs.

Note that a failure rate of 0.0001%/1000 hours and 1 FIT are equivalent numbers.

Bathtub Curve: Failure Rate Over Time

The generic representational graph of failure rate vs. time takes the shape of a bathtub curve. (See Figure 1).

The early failure rate (infant mortality) period starts from initial operation (time T0) and decreases as time goes on.

Time T1 signifies the end of the infant mortality period. The next phase of the curve occurs between time T1 and T2. This long period of time is distinguished by a nearly constant and very low failure rate. After T2 is passed, the failure rate starts to increase slowly. This last phase of failure rate vs. time is known as the wear-out period.

Temperature Dependency

In order to establish failure rates in a reasonable time, it is necessary to accelerate the incidence of the failure modes. Higher environmental stress levels than those encountered under normal conditions are needed. The accelerating parameter most employed is junction temperature, although voltage and humidity, for example, are also used. Higher temperatures are capable of accelerating many common failure modes dramatically.

Arrhenius Equation

A number of mathematical models were developed to quantify the relationship between accelerated failure rates and increased junction temperatures. The one model most commonly used employs the Arrhenius Equation. It is as follows:

$$AF = e^{x}, \text{ where } x = \frac{E_A}{K}\left[\frac{1}{T_N} - \frac{1}{T_A}\right]$$

AF = Acceleration Factor (non-dimensional)

e = 2.718281828... (non-dimensional constant)

E_A = Activation energy level (electron volts)

k = Boltzmann's constant = 8.6172 x 10-5 (electron-volts/degree Kelvin)

T_N = Normal junction temperature (degrees Kelvin)

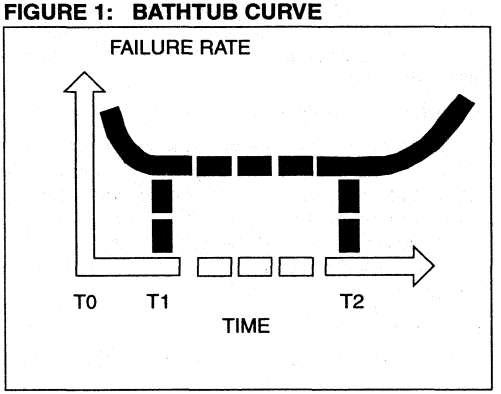
T_A = Accelerated junction temperature (degrees Kelvin)

Thus, the time to achieve a certain probability of failure at time T1 under temperature TN can be compressed to T1 divided by AF at the accelerated temperature, TA.

Note that for true acceleration, the acceleration factor AF is independent of the probability of the fail point specified.

AF, the dependent variable of the Arrhenius Equation is a function of several variables. TN and TA are specified for the situation under consideration. EA is a function of the particular mode of failure, and is determined by experimental evaluation.

FIGURE 1: BATHTUB CURVE



7

© 1995 Microchip Technology Inc.

Activation Energy Level

Activation energy levels in semiconductors generally are in the 0.3 - 1.1 electron-volt range. Each failure mode has its own activation energy. Some typical examples are:

FAILURE MECHANISM	E_A (eV)
Oxide/Dielectric Breakdown	0.3
Electromigration	0.5 to 0.7
Surface Related Contamination	1.0
Intermetallics	1.0
Floating Gate Charge Loss	0.6 to 1.2
Hot Electron Trapping	-.1
Charge Trapping	0.12

A compromise value of 0.6 electron-volts is often used when there is no specific information relating to the failure modes being accelerated.

RELIABILITY TESTS

Operating Life Test

The Operating Life Test is run under dynamic bias conditions where inputs are clocked. The test is conducted at high temperature to accelerate the failure mechanisms. The normal temperature for the test is +125°C for 1,000 hours. Readouts occur at 168 and 1,000 hours. Early hour failures are usually associated with manufacturing defects or otherwise marginal material.

Retention Bake

The Retention Bake Test is performed to accelerate data loss on floating gate devices. The test consists of unbiased baking at elevated temperature. Usually the test lasts for 1,000 hours at +150°C. The failure mechanism that is accelerated is charge leakage from a stored element.

Endurance Cycling

Endurance Cycling establishes the number of times a device can be programmed and erased. Normally the test is conducted at rated temperature conditions and is followed by retention bake. The standard cycling at Microchip is done at 85°C using a page cycle mode and is followed by a bake of both a checkerboard and an inverse checkerboard of 48 hours at 150°C.

Temperature Cycle

The Temperature Cycle test simulates stresses which occur to systems during power up/power down sequences. The test is intended to reveal any deficiencies resulting from thermal expansion mismatch of the die/package structure. Normally the test is conducted by cycling between -65°C and +150°C in an air ambient. Duration for the test is typically 500 cycles for both plastic and ceramic packages. Endpoint criteria are both electrical and visual/mechanical.

Thermal Shock

The Thermal Shock test is similar to the Temperature Cycle test except that the ambient during cycling is liquid-to-liquid. This stimulates rapid thermal environmental changes. The mechanisms accelerated are identical to those in the Temperature Cycle test except that the Thermal Shock test is a more accelerated test with temperatures normally +125°C to -55°C. The number of cycles are 500 for qualification testing.

Autoclave

The autoclave test determines the survivability of devices in molded plastic packages to a hot, humid environment. The test exposes unbiased, plastic packaged devices to saturated steam at 121°C and 15 pounds per square inch (one atmosphere) gauge pressure. The 168 or more hours of testing allows moisture to penetrate into the die. Chemical corrosion of the die metallization may occur if ionic contaminants are present and the die surface protection is deficient or damaged. Charge leaks from floating devices usually happen before a corrosion mechanism develops.

Temperature Humidity Test

The Temperature Humidity test determines the survivability of devices in molded plastic packages functioning in a hot, humid environment. By convention, test conditions are 85°C with 85% relative humidity. The parts are biased to lend themselves to electrochemical corrosion. The duration of the test is usually 1,000 hours or more. The test checks the adequacy of the die surface protection and the plastic's lack of ionic impurities. The applied bias is 5 volts on alternating pins or set up for minimum power to reduce internal heating and consequent moisture evaporation on the device. Similar to the Autoclave test, charge loss on floating gate devices is a principle failure mechanism.

HAST

The Highly Accelerated Stress Test is similar to the Temperature Humidity Test but with more stringent temperature exposure. Devices are subjected to 130°C with 85% relative humidity and an alternating bias of 5 volts and ground on device pins. The duration of the test is 168 hours. This tests for ionic contamination and corrosion, but floating gate devices may also fail for charge loss, due to the high temperature.

© 1995 Microchip Technology Inc.

QUALIFICATION CATEGORIES

In general, qualification is required for new design, major changes in old design, process or material when either wafer fabrication or package assembly operations are affected. Cross functional teams which include reliability develop new products for introduction. In other areas, Microchip utilizes the concept of a Change Control Board which meets regularly to establish which criteria is to be used for all specific proposed changes. This board is made up of representative leaders of various groups and departments throughout Microchip to insure all concerns are heard early during the process.

QUALIFICATION PROGRAMS

Qualifications guarantee changes to or new processes and technologies are properly evaluated for reliability performance.

Reliability Monitoring

Microchip's reliability monitoring program is a comprehensive effort to measure the reliability of all process families with strict regularity. The program strives to improve performance through failure analysis and corrective action. Numerous screening procedures are used and estimates of product life and expected failure rates are provided.

Typical tests and frequencies include:

- Die Monitor on selected product for -
  - Dynamic Life
  - Retention Bake
  - Endurance
- Periodic (weekly, monthly and quarterly) package monitors to evaluate:
  - Mechanical stresses
  - Alignment
  - Temperature and moisture stresses
  - Corrosion resistance
  - Marking permanency

© 1995 Microchip Technology Inc.

APPENDIX A: IN LINE CONTROLS

TABLE 1: CONTROLS - PLASTIC PACKAGE ASSEMBLY

Operation	Action	Sample Plan	Responsibility		Referenced MIL-STD
			Quality	Prod	
Die Visual	Reject defectives 100% rescreen per LTPD	10% sample LTPD 10	— X	X —	MIL-STD-883C Method 2010
Wafer Saw	Machine Shut Down	One kerf per lot	X	—	MIL-STD-883C Method 2010
Die Attach	Machine Shut Down	4X/Lot/Machine LTPD 15	X	—	N/A
Wire Bond	Machine Shut Down	1% AQL each 1/2 shift	X	—	MIL-STD-883C Method 2010
Post Wire Bond	Reject defectives 100% rescreen per LTPD	LTPD 15	X	—	MIL-STD-883C Method 2010
Mold Press	Machine Shut Down	One sample /4 hrs	X	—	N/A
Die Plating	Reject defectives 100% rescreen per LTPD	Every 4 hrs LTPD 10	X	—	N/A
Trim and Form	Reject defectives 100% rescreen per LTPD	Once/ 2 hrs LTPD 10	X	—	N/A
External Visual and Documentation Verification	Reject defectives 100% rescreen per LTPD	100% LTPD 2	— X	X —	MIL-STD-883C Method 2010, Method 2016

© 1995 Microchip Technology Inc.

TABLE 2: CONTROLS - CERAMIC PACKAGE ASSEMBLY

Operation	Action	Sample Plan	Responsibility		Referenced MIL-STD
			Quality	Prod	
Die Visual	Reject defectives 100% rescreen per LTPD	10% LTPD 10	— X	X —	MIL-STD-883C Method 2010
Wafer Saw	Machine Shut Down	One kerf per lot	X	—	MIL-STD-883C Method 2010
Die Attach	Machine Shut Down	Non-destruct each 2 hrs destruct each shift	X	—	MIL-STD-883C Method 2010
Wire Bond	Machine Shut Down	4X/shift/machine	X	—	MIL-STD-883C Method 2010
Preseal Visual	Reject defectives 100% rescreen per LTPD	100% LTPD 15	— X	X —	MIL-STD-883C Method 2010
Package Seal	Machine Shut Down	LTPD 15	X	—	N/A
Environmental Stress Centrifuge Temp Cycle	Machine Shut Down	LTPD 5 84(0) 84(0)	X	—	MIL-STD-883C Method 2001 Method 1010
Fine Leak	Reject defectives 100% rescreen per LTPD	LTPD 5	X	—	MIL-STD-883C Method 1014
Gross Leak	Reject defectives 100% rescreen per LTPD	LTPD 5	X	—	MIL-STD-883C Method 1014
Lead Trim	Reject defectives 100% rescreen per LTPD	100% LTPD 2	— X	X —	MIL-STD-883C Method 2009
External Visual and Documentation Verification	Reject defectives 100% rescreen per LTPD	100% LTPD 2	— X	X —	MIL-STD-883C Method 2010, Method 2016

7

APPENDIX B: MATERIAL CONTROLS PACKAGE

TABLE 3: MATERIALS CONTROLS - PLASTIC PACKAGE ASSEMBLY

Operation	Action	Sample Plan	Responsibility		Referenced MIL-STD
			Quality	Prod	
Lead Frame	Reject defectives 100% rescreen per LTPD	Visual, LTPD 2 Functional, LTPD 10 and material spec	X	—	N/A
Die Attach Epoxy	Reject	Functional, LTPD 15 and material spec	X	—	N/A
Gold Wire	Reject	Per material spec	X	—	N/A
Molding Compound	Reject	Spiral flow, 3X/lot Functional, 1X/lot and material spec	X	—	N/A

TABLE 4: MATERIALS CONTROLS - CERAMIC PACKAGE ASSEMBLY

Operation	Action	Sample Plan	Responsibility		Referenced MIL-STD
			Quality	Prod	
Base/Lead Frame	Reject 100% rescreen per LTPD	Visual, LTPD 10 Functional, LTPD 10 Bake test, LTPD 15 Dimensions, LTPD 50 and material spec	X	—	MIL-M-38510
Package	Reject 100% rescreen per LTPD	Visual, LTPD 10 Functional, LTPD 10 Bake test, LTPD 15 Dimensions, LTPD 50 and material spec	X	—	MIL-M-38510
Preform	Reject	Visual, LTPD 10 Functional, LTPD 15	X	—	MIL-M-38510
Bond Wire	Reject	Per material spec 2 spools/lot	X	—	MIL-M-38510
Lid	Reject	Visual, LTPD 7 Functional, LTPD 10 and material spec	X	—	MIL-M-38510

© 1995 Microchip Technology Inc.

MICROCHIP

Product Reliability

OVERVIEW

Microchip Technology Inc.'s products provide competitive leadership in quality and reliability, with demonstrated performance of less than 100 FITs (Failures in Time) operating life for most products. The designed-in reliability of Microchip's products are supported by ongoing reliability data monitors. This document presents current data for your use - to provide you with results you can count on.

The test descriptions included in this document explain Microchip's quality and reliability system. The product data demonstrates its results.

The customer's quality requirements are Microchip's top priority. Ongoing customer feedback and device performance monitoring drive Microchip, leading to continuing improvements in the long-term quality and reliability.

FAILURE RATE CALCULATION

Extended field life is simulated by using high ambient temperature. In the semiconductor technology, high temperatures dramatically accelerate the mechanisms leading to component failure. Using performance results at different temperatures, an activation energy is determined using the Arrhenius equation. For each type of failure mechanism, the activation energy expresses the degree to which temperature increases the failure rate.

The activation energy values determined by Microchip Technology agree closely with those published in the literature. For complex CMOS devices in production at Microchip Technology, an activation energy of 0.6 eV has been shown to be representative of typical failures on operating life. This activation energy also applies to some of our retention bake failures, though most are 1.2eV. By definition, failure is reached when a device no longer meets the data sheet specifications as a direct result of the reliability test environment to which it was exposed. Common failure modes for CMOS integrated circuits are identified for each test environment.

To establish a field failure rate, the acceleration factor is applied to the device operating hours observed at high temperature stress and extrapolated to a failure rate at 55°C ambient temperature in still air.

The actual failure rate experienced could be considerably less than that calculated if lower device temperatures occur in the application board, such as would be the case if a fan, a heat sink, or air flow by convection is used.

Environment	Typical Failure Mechanism
Operating Life	Process parameter drift/shift Metal electromigration Internal leakage path Lifted bond/ball bond chip-out
Temperature Cycle	Lifted bond/ball chip-out Cracked die or surface cracks Bond pad corrosion
Biased-Humidity	Internal circuit corrosion
Autoclave	Inter-pin leakage Charge loss
High Temp. Bake	Charge loss
High Temp. Reverse Bias	Charge gain, Parameter drift/shift

DEFINITIONS

FIT (Failure In Time): Expresses the estimated field failure rate in number of failures per billion power-on device-hours. 100 FITS equals 0.01% fail per 1,000 device-hours.

Operating Life Test: The device is dynamically exercised at a high ambient temperature (usually 125°C) to quickly simulate field life. Derating from high temperature, an ambient use condition failure rate can be calculated.

Temperature Cycle: The devices are exposed to severe extremes of temperature in an alternating fashion (-65°C for 15 minutes, 150°C for 15 minutes per cycle). Package strength, bond quality and consistency of assembly process are stressed using this environment.

Biased-Humidity: Moisture and bias are used to accelerate corrosion-type failures in plastic packages. The conditions include 85°C ambient temperature with 85% relative humidity. Typical bias voltage is +5 volts and ground on alternating pins.

Autoclave (pressure cooker): Using a pressure of one atmosphere above atmospheric pressure, plastic packaged devices are exposed to moisture at 121°C. The pressure forces moisture permeation of the package and accelerates related failure mechanisms, if present, on the device.

7

© 1995 Microchip Technology Inc.

Thermal Shock: Exposes devices to extreme temperatures from -55°C to +125°C by alternate immersion in liquid media.

Retention Bake: A 150°C temperature stress is used to accelerate charge loss in the memory cell and measure the data retention on the EPROM and EEPROM portions of the circuitry.

HAST: Moisture, extreme heat and bias are used to accelerate corrosion and contamination in plastic packages. The conditions are 130°C and 85% relative humidity. Typical bias voltage is +5 Volts and ground on alternating pins.

RELIABILITY CONTROL SYSTEM

A comprehensive qualification system ensures that released products are designed, processed, packaged and tested to meet both design functionality and strict reliability objectives. Once qualified, a reliability monitor system ensures that wafer fabrication and assembly process performance is stable over time. A set of baseline specifications is maintained that states which changes require requalification. These process changes can only be made after successful demonstration of reliability performance. This system results in reliable field performance, while enabling the smooth phase-in of improved designs and product capability.

RELIABILITY DATA SUMMARY

Introduction

This section provides a reliability summary of Microchip Technology's product. Included is reliability data and packaging information obtained over the recent past.

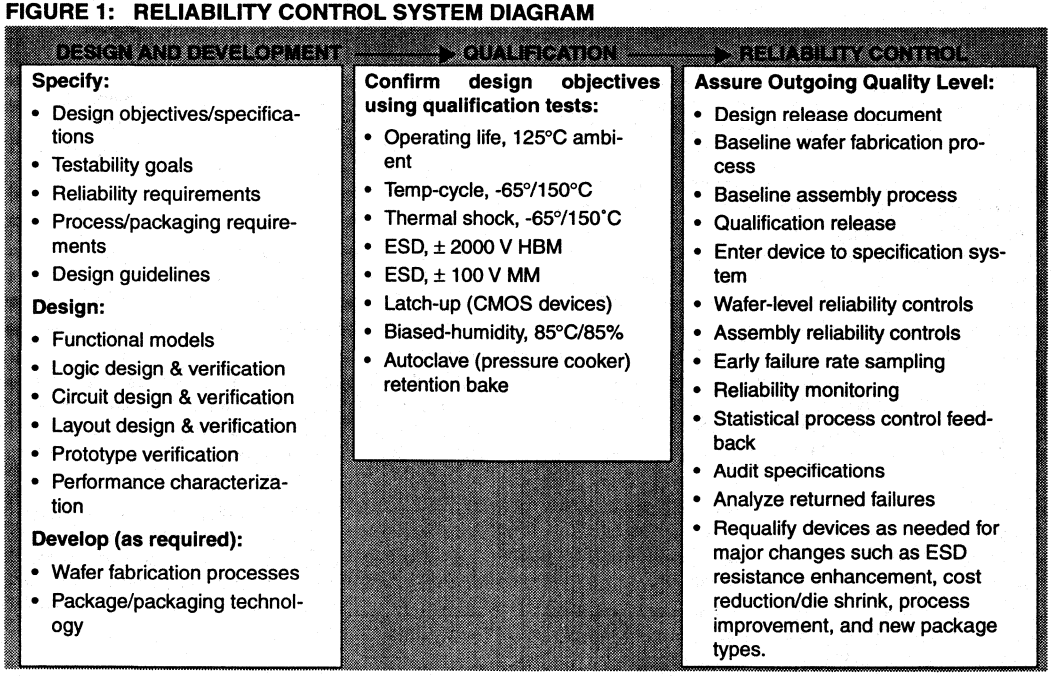
Plastic Package Characteristics and Codes

As part of an on going product program, Microchip Technology will apply its Quality and Reliability process in evaluating the latest developments in plastic packaging technology, and implement the highest reliability materials and assembly techniques. The plastic packages that are currently available from Microchip are listed in the table below.

Package Description Identification Code

Package Description	Identification Code
Plastic Leadless Chip Carrier	L
Plastic Dual In Line (600)	P
Plastic Dual In Line (300)	SP
Plastic SOIC (.150)	SL/SN
Plastic SOIC (.207)	SM
Plastic SOIC (.300)	SO
Plastic TSOP (8 x 20mm)	TS
Plastic SSOP (.207)	SS

FIGURE 1: RELIABILITY CONTROL SYSTEM DIAGRAM



© 1995 Microchip Technology Inc.

HIGH TEMPERATURE (125°C) DYNAMIC LIFE TEST

Graph set for EEPROM, PIC16/17 and EPROM for all conditions

High temperature dynamic life testing accelerates random failure modes which would occur in user applications. Voltage bias and address signals are used to exercise the device in a manner similar to user systems.

FIGURE 2: EEPROM DYNAMIC LIFE

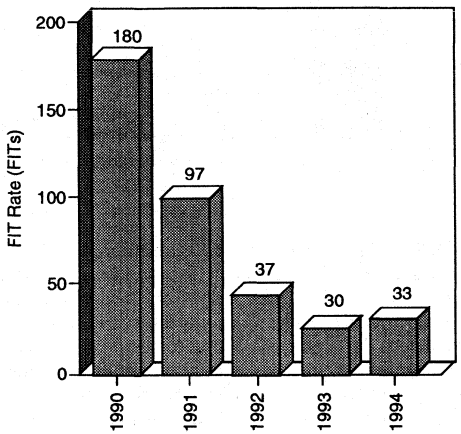


FIGURE 3: EPROM DYNAMIC LIFE

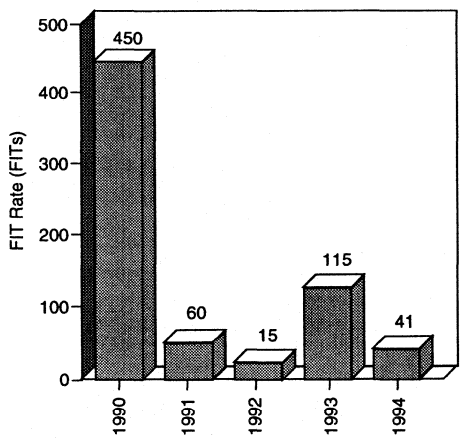
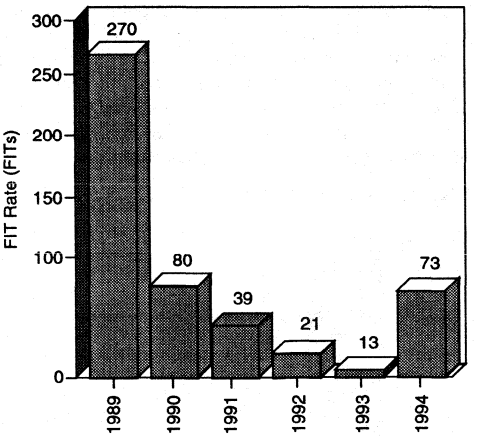


FIGURE 4: PIC16/17 MICROCONTROLLER DYNAMIC LIFE



7

DATA RETENTION BAKE

Data storage in applicable devices is done by developing a charge on the floating gate structure in the memory cell. Charge loss in this cell structure results in a conversion of zeroes to ones. In order to evaluate the level of this type of failure, devices are subjected to a 150°C bake. This bake accelerates charge loss in the memory cell and 1000 hours at 150°C is equivalent to approximately 13.5 years in the field at 55°C.

FIGURE 5: EEPROM RETENTION BAKE

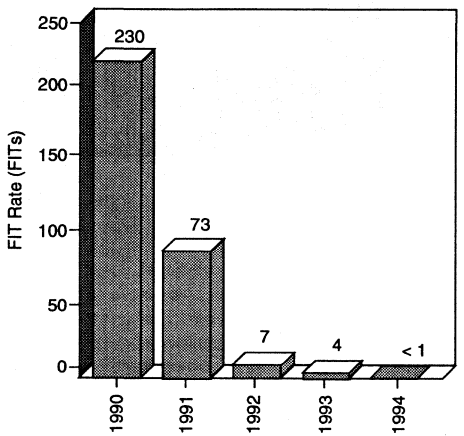


FIGURE 6: EPROM RETENTION BAKE

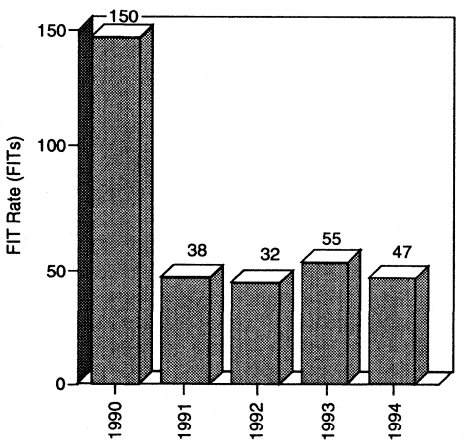
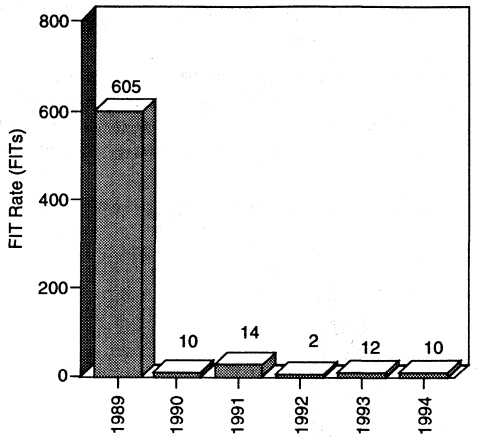


FIGURE 7: PIC16/17 MICROCONTROLLER RETENTION BAKE



NOTE: Representation of reliability data typically shows calendar year grouping along the x-axis, except for 1993 which includes only first, second and third quarters. This provides the equal time interval normally expected for graphical presentation. However, Chi-square statistics demand equivalent device-hours for fair interval comparison. Such data grouping assures that relatively small sample sizes do not indicate unrepresentative FIT rates.

© 1995 Microchip Technology Inc.

BIASED 85°/85% R.H.

Microchip Technology evaluates plastic encapsulated devices ability to withstand high temperature, high humidity environments while under electrical bias. This is done by utilizing the industry standard test method known as 85/85. This stress is designed to create corrosion of the metal or internal device leakage if ionic contaminants are present but also may cause charge loss in memory cells.

Operating Hours				
Package	**24**	**168**	**504**	**1008**
PDIP	0/5152	2/5152	2/5150	4/5148
PLCC	0/3909	5/3909	2/3904	3/3902
SOIC	0/4827	5/4827	1/4694	0/4693
TSOP	0/377	0/377	0/377	0/377

PCT (AUTOCLAVE)

Originally, this test was designed to evaluate corrosion of bond pads due to penetrating moisture combining with contaminant residue on the metal surface. The corrosion failure rate for this test has become nearly zero and a new failure mode has surfaced. This is memory cell charge loss due to moisture penetration along the floating gate allowing a conduction path for removal of stored charge. This moisture path is between the seal of the metal and the passivation which can then be traced to the substrate near the edge of the floating gate. This failure type is the primary mode in the data provided.

Operating Hours		
Package	**24**	**168**
PDIP	0/10199	5/10199
PLCC	0/4978	4/4978
SOIC	0/8764	7/8764
TSOP	0/234	2/234

TEMPERATURE CYCLING

This thermal tests evaluates air to air rapid temperature change evaluating built in material stresses. This is a worst case simulation of system power up/ power down and is based on stringent military packaging requirements.

Operating Results			
Package	**15 Cycles**	**100 Cycles**	**500 Cycles**
PDIP	0/3112	3/2514	3/2054
PLCC	0/1413	0/1275	1/907
SOIC	0/1895	0/1665	0/837
TSOP	0/234	0/96	0/96
SSOP	0/240	0/194	0/94
VSOP	0/46	0/0	0/0

THERMAL SHOCK

Thermal shock is the most extreme case of temperature cycling by using liquid immersion for the technique to change the device environment. This accelerates any stress related failures with the rapidly changing gradient. After the temperature stressing a constant force centrifuge test is also preformed prior to final electrical testing to further uncover any defects that may have occurred under stress.

Operating Results			
Package	**15 Cycles**	**100 Cycles**	**500 Cycles**
PDIP	0/5700	2/3386	0/2682
PLCC	0/2365	2/1487	7/1292
SOIC	0/3422	0/2332	0/2056
TSOP	0/330	0/234	0/96
SSOP	0/234	0/140	0/94
VSOP	0/46	0/46	0/0

HAST (130°/85% R.H.)

Highly Accelerated Stress Testing evaluates plastic encapsulated devices' ability to withstand extreme high temperature, high humidity environments while under electrical bias. This is done by a new method known as HAST. This stress is designed to create corrosion of the metal or internal device leakage if ionic contaminants are present but also may cause charge loss in memory cells.

Operating Results		
Package	**48 Hours**	**168 Hours**
PDIP	1/3707	1/2030
PLCC	2/1542	0/1170
SOIC	5/2784	3/2779

© 1995 Microchip Technology Inc.

PRODUCT RELIABILITY DATA

CMOS PIC16/17								
		Operating Hours						
Device	Operation	24	168	504	1008	Fails	Device Hours	FITS 60% CL@55°C
PIC16C57	DLT	0/11648	11/11648	0/5316	0/5316	11	6,422,304	47
PIC16C56	DLT	0/9192	1/9192	1/3672	1/3671	3	4,628,736	22
PIC16C55	DLT	0/9215	1/9215	1/4562	2/4561	4	5,380,200	23
PIC16C54	DLT	1/13424	1/13423	3/7334	1/7331	6	8,415,792	21
PC16C84	DLT	0/1305	0/1305	0/1023	1/1023	1	1,078,560	45
PIC16C57	BAKE	0/13092	0/13092	1/2788	0/2787	1	4,541,376	4
PIC16C56	BAKE	0/11136	2/11136	0/2280	0/2280	2	3,786,048	7
PIC16C55	BAKE	0/9887	1/9887	1/2823	1/2822	3	4,032,336	9
PIC16C54	BAKE	0/15938	2/15938	1/4689	0/4688	3	6,616,344	5
PIC16C84	BAKE	0/485	0/485	0/485	0/485	0	488,880	< 1
EEPROM								
		Operating Hours						
Device	Operation	24	168	504	1008	Fails	Device Hours	FITS 60% CL@55°C
24CXX	DLT	0/18597	1/18597	2/5735	4/5733	7	7,941,696	25
93CXX	DLT	2/6993	4/6991	0/2601	0/2601	6	3,359,664	52
24LCXX	DLT	0/15164	3/15164	2/8427	3/8425	8	9,626,232	24
93LCXX	DLT	0/7771	5/7771	3/3188	2/3185	10	3,983,448	69
28C16	DLT	0/2681	2/2681	0/1071	0/1071	2	1,350,048	55
28C64	DLT	1/5800	4/5799	0/2586	0/2586	5	3,146,640	48
24CXX	BAKE	2/14345	1/14343	1/3600	2/3599	6	5,433,960	< 1
93CXX	BAKE	0/6320	0/6320	0/1425	0/1425	0	2,258,760	< 1
24LCXX	BAKE	0/11842	0/11842	0/5004	1/5004	1	6,192,816	< 1
93LCXX	BAKE	0/6969	0/6969	0/3192	0/3192	0	3,852,072	< 1
28C16	BAKE	1/3840	2/3839	2/797	1/795	6	1,314,600	< 1
28C64	BAKE	0/3942	26/3942	1/1144	1/1143	28	1,623,216	1
EPROM								
		Operating Hours						
Device	Operation	24	168	504	1008	Fails	Device Hours	FITS 60% CL@55°C
27HC256	DLT	0/2316	1/2316	1/1166	0/1165	2	1,368,528	54
27C256	DLT	1/9666	4/9665	0/2028	0/2028	5	3,327,408	45
27C512	DLT	0/3658	4/3658	0/1260	0/1260	4	1,672,944	75
27HC256	BAKE	0/2656	5/2656	1/627	0/626	6	972,888	64
27C256	BAKE	2/9905	3/9903	0/2478	1/2478	6	3,745,560	17
27C512	BAKE	2/3334	16/3332	1/570	2/569	21	1,038,912	188

Operation Legend: DLT - Dynamic Life Test (125°C) Bake -Retention Bake (150°C)

© 1995 Microchip Technology Inc.

EEPROM Endurance

INTRODUCTION

A unique feature of non-volatile memory devices is the dual requirement both to change and to maintain data states. It is this combination of requirements that provides the contrasting nature that defines the complexities involved in change and maintaining such change. Anything that enhances the physics to allow a data state change in contrast degrades the retention of that change. It also holds that any retention enhancements inhibit the data changing capabilities. A balance must be struck between the combinations to achieve the field requirements of customer applications.

Erase/Write cycling has many variables which greatly effect the lifetime of the device. To accurately make comparisons between specifications and the actual requirements, or any other comparisons, these factors must be well understood and taken into account.

TECHNOLOGY OVERVIEW

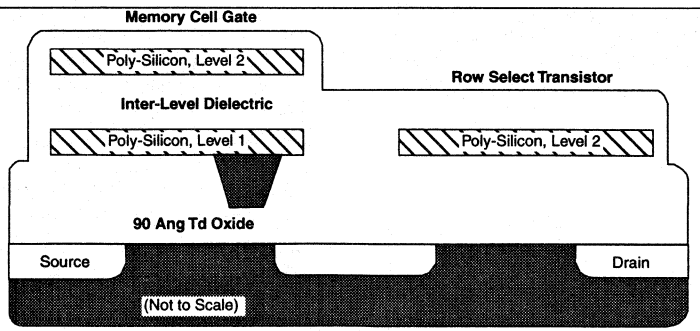
Silicon Technology

The basic technology employed by Microchip Technology for EEPROM's is a FLOTOX structure as drawn below. This is a an industry standard architecture base which has been enhanced by Microchip to provide improvements to the quality and reliability of the devices produced.

Circuit Technology

These cells are then structured in either an 8 or 16 bit word organization for data storage with between 32 words (256 byte device) and 8K words (64K device) using standard binary decoding schemes found industry wide on memory devices. Data can then be transmitted either into the device for storage or read from the device when needed along a single DATA pin, (I^2C bus) or a dual Data in/Data out configuration (3-wire bus). The device has no restriction on the number of read cycles that can be processed per byte without damage but the storage process does have finite limitations.

Currently two different schemes of error correction are being utilized on Microchip EEPROM's. The 24CXX, 93CXX, 85CXX, 59CXX and 28CXX device types utilize a modified Hamming code redundancy scheme with four parity bits per eight bit byte. This has been the industry standard correction scheme for enhancement of cycling lifetimes by eliminating single bit per word errors. An alternative approach has been developed utilizing an AND cell concept of redundant memory cells. This further enhances the write/erase lifetime over the Hamming code and has been implemented on the 24LCXX and 93LCXX circuits.



	Read	Write	Erase	Standby
Bit Line	1.6 volts	18.0 volts	0.0 volts	0.0 volts
Row Select Gate	5.0 volts	20.0 volts	0.0 volts	0.0 volts
Memory Cell Gate	5.0 volts	0.0 volts	20.0 volts	0.0 volts
Common Source	0.0 volts	Float	0.0 volts	0.0 volts

© 1995 Microchip Technology Inc.

Reliability Endurance

The endurance failure rate curve for the Microchip devices is presented in the standard form for this curve from EEPROM FLOTOX manufacturers. Microchip does write/erase cycle all EEPROM devices prior to shipment to remove the infant mortality endurance failures from the population. This characteristic curve, with two failure increase sections, is shown below for reference. Both sections have single bits failing as the dominant mechanism.

The first of the failure increase sections is usually related to breakdown of oxides from latent oxide defects that are inherent to any process. These oxides have reached a time dependent dielectric breakdown condition and permanently rupture. This generally characterizes the first 200K write/erase cycles under any conditions. The second curve is the standard trap up of electrons within the tunnel dielectric which closes the write/erase threshold window until the device no longer adequately programs or erases.

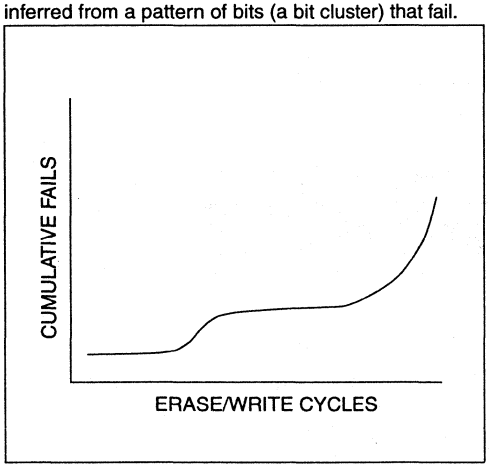
This curve depends on multiple parameters, but the trap up failure increase portion often does not occur until ten million write/erase cycles or greater.

The first failures from endurance cycling and the long term end of life failures are due to different mechanisms.

The failures from 200K to 500K cycles have historically been attributed to "fast trap up" around the industry. Analysis at Microchip has shown that these failures are not actually trap up but oxide breakdown in nature. They most often manifest themselves as single bit charge loss or charge pump failure, both due to the formation of a conductive path within the gate oxide layer.

These oxide breakdown failures can be related to defects of three types of categories.

Type 1 is a residual chemical stain left behind on the wafer after processing due to an inadequate rinse. These are very difficult to physically detect and are best inferred from a pattern of bits (a bit cluster) that fail.



Type 2 is a physical defect which can be found upon microscopic or SEM analysis resulting in the failure. This is most often a particle, polysilicon nodule or metal short.

Type 3 is a physical defect with low activation energy that cannot be detected until end of life evaluation because of it's change in state (subsequent consumption) during latter processing steps.

The end of life mechanism is called "oxide trap up". This is where the tunnel dielectric oxide layer loses it's ability to pass charge and begins to retain some portion of the charge that it passes to the floating gate. These excess electrons within the oxide act as a charge shield, resulting in insufficient charge movement while significantly raising the voltage required to continue transmitting a constant charge level. Since the programming voltage is not adjustable this results in less charge movement for either the write or erase state. These states, whether charged negatively or charged positively, approach a central point and become indistinguishable to the detection circuitry of the device. This results in a failure to read the correct pattern, (impossible to distinguish between a programmed one and a programmed zero) beginning with the extreme voltage values of the operating specification.

Microchip strives to offer the lowest failure rate for both early life and wearout fails. Early fails in Microchip products are in the PPM range, and wearout does not set in most applications until after ten million cycles.

MEASUREMENT OF CYCLING

Microchip Technology defines a device lifetime in the strictest sense, that is, stated lifetime has to exhibit the truest correlation to customer results. All units shipped from Microchip have Error Correction circuitry engaged for customer use. Error Correction amends any one error per byte for Hamming and one error per bit for AND cell which allows the device to read correct data. An endurance failure is determined when any one bit is not capable of being correctly written and maintained in that state indefinitely.

The device lifetime is defined when a specified percentage of devices, (Microchip currently uses a cumulative 2.5 percent), have a customer detectable error under worst case operating conditions.

Ongoing monitors are acquired from every wafer lot of material manufactured for shipment. Samples are subjected to byte cycling of a checkerboard pattern at 85°C in rapid succession to the specified number of guaranteed cycles. These units then are baked in both checkerboard and inverse checkerboard forms and electrically tested to ensure that data sheet requirements are met. This data is accumulated on a monthly basis and reviewed to measure both results of continuous improvement programs and conformance against the device standards.

© 1995 Microchip Technology Inc.

ENDURANCE VARIABLES

a) **Temperature:** Within the FLOTOX technology, temperature has an inverse effect on the endurance of a EEPROM device. The activation energy of these cycling failures is approximately 0.15 eV. The long term trap up portion of the curve is worsened by temperature to a greater extent than the early fails due to the difficult failure mechanisms that are activated.

b) **Delay between cycles:** This has been reported in the industry as having an enhanced effect on the lifetime of EEPROM devices. For some technologies this does have a positive effect, however, this is not strictly the case for FLOTOX manufacturers. While the second failure rate increase period (associated with end of life) may be impacted by this due to a decreased rate of electron trapping, the first failure rate is actually not impacted by this variable.

c) **Write timing:** The decrease in write time to the device correlates directly with write/erase cycling failure rate of the device. This shorter pulse reduces the cell time at voltage which then provides an enhanced life prior to the occurrence of a time dependent oxide breakdown. It also passes fewer electrons through the oxide providing less potential trapping possibilities while maintaining adequate margins for the written state.

   Please note that the rise time of the signal, which the customer does not have control over is also a dominant effect.

d) **Vcc voltage:** The higher voltages generate higher fields within the device. This causes more stress which is offset by the operational increase of internal timers and actually shortens the write time of the device.

   Lower voltages this has the opposite effects on the individual parameters except in cycles obviously where the write timer is externally controlled. This overall effect is minor compared to the others in magnitude on the failure rate curve and is variable over the customer operating range with a maximum at Vcc=5.5 Volts.

e) **Pattern effect:** The pattern that is programmed in the device does play a first order role in the overall lifetime. The act of programming a nonvolatile memory inflicts damage on the device that cannot be repaired. This damage is the result of exposure to high electric fields which over a period of time either breakdown or trap up the effected oxide causing failures. The act of writing a cell from a one to a zero provides the maximum amount of stress by exercising the charge pump and passing electrons through the tunnel dielectric. (Please note that to write a zero even from a zero state causes an automatic byte erase prior to the write converting the bit to a one and returning it to it's original state!) Conversely writing a one from a one then passes no charge through the cell and therefore does no damage to the cell but does stress the charge pump.

   From an array standpoint, this would allow a checkerboard/ inverse checkerboard patterned device to endure twice the number of write cycles that an all zero patterned device would last. In general this appears to be approximately correct but does neglect the charge pump and other peripheral wearout mechanisms.

f) **Cycling mode:** Three modes exist in Microchip devices that are primarily used for endurance evaluations. The byte write mode (one single byte written at a time) is the standard mode used by the customer in the field. A second technique exists called block mode which exercises all the cells of the array simultaneously. The lifetime expectations are approximately ten times as long for these block cycled devices as equivalently cycled byte cycled circuits based on experimental findings. This effect has been traced back to the rise time of the programming signal at the memory cell. The faster this voltage rise occurs, the more damage occurs and the shorter the lifetime. The block mode has a much slower rise time given the entire array being utilized provides a much larger resistive/capacitive load which slows the signal rise ultimately resulting in the greater lifetime. Please note that the page mode which can be utilized by the customers falls between the block and byte modes with respect to failure rate. Page mode offers the best balance of endurance and write cycle times.

g) **Array Size:** This effect is a direct result of how fast most devices will fail due to a single bit not working simply due to the number of bits involved. This is not exactly double the failure rate with a doubling of the memory size since some circuits fail within the charge pump or decoding circuitry sections and are therefore not directly related to array size.

© 1995 Microchip Technology Inc.

FIELD RESULTS

Microchip Technology, after significant experimentation, has developed a model of the Endurance failure rates as a function of all the variables listed above. This model is available to the customer in the form of a diskette called Total Endurance™. This allows the customer to bypass confusing information and conditions other than their application and directly predict the failures seen in their application conditions within a few percent. This also allows the customer to adjust operating parameters and immediately evaluate the impact on the results of the final system. Results for a typical application (obtained from the Total Endurance model are listed at right for reference).

Device	Application Life	Cum Percent
24C02A	10 years	154 PPM
24C04A	10 years	172 PPM
93C46	10 years	136 PPM
24LC02B	10 years	108 PPM
24LC04B	10 years	131 PPM
24LC16B	10 years	668 PPM
93LC46B	10 years	374 PPM
93LC56B	10 years	108 PPM
24LC65HE	10 years	131 PPM
24LC65SE	10 years	4061 PPM

Typical conditions used are 25°C, Byte mode operation with 24 cycles per day, Vcc = 4.5 Volts with a random pattern writing one quarter of the array at each occasion. The failure rates quoted are the expected failure rate at the end of the application life using an unlimited number of read cycles. For more information on Endurance, it is recommended that the user obtain a copy of Total Endurance.

© 1995 Microchip Technology Inc.

SECTION 8
PACKAGING INFORMATION

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

PACKAGING

Commercial/Industrial Outlines and Parameters

PART NUMBER SUFFIX DESIGNATIONS:

XXXXXXXXXX - XX X/XX XXX

Examples:

27C256T - 15I/J
PIC16C54 - RCI/SO

QTP, SQTP, or ROM Code or Special Requirements

Case Outline

D = Ceramic
J = CERDIP (with window if EPROM) - all products except Microcontrollers
K = LCC (Ceramic Leadless Chip Carrier, not thermally enhanced)
L = PLCC (Plastic Leaded Chip Carrier)
P = Plastic DIP
S = Die in Waffle Pack
W = Die in Wafer Form
CB = COB (Chip-on-Board)
JN = CERDIP, no window - for Microcontrollers only
JW = CERDIP, windowed - for Microcontrollers only
PQ = Plastic Quad Flat Pack (PQFP) - Metric
SJ = Skinny CERDIP
SL = 14-Lead Small Outline 150 mil
SM = 8-Lead Small Outline 207 mil
SN = 8-Lead Small Outline 150 mil
SO = Small Outline 300 mil
SP = Skinny Plastic Carrier
SS = Shrink Small Outline Package
TS = Thin Small Outline (TSOP) 8 mm x 20 mm
VS = Very Small Outline (VSOP) 8 mm x 13.4 mm

Process Temperature

Blank = 0°C to +70°C
I = -40°C to +85°C
E = -40°C to +125°C

Speed (EPROM/High Density EEPROM)

-55 = 55 ns
-70 = 70 ns
-90 = 90 ns
-10 = 100 ns
-12 = 120 ns
-15 = 150 ns
-17 = 170 ns
-20 = 200 ns
-25 = 250 ns
-30 = 300 ns

Crystal Frequency Designator for PIC16/17 Microcontrollers

LP = Low Power Crystal
RC = Resistor Capacitor
XT = Standard Crystal/Resonator
HS = High Speed Crystal
-04 = 4 MHz
-10 = 10 MHz
-16 = 16 MHz
-20 = 20 MHz
-25 = 25 MHz

OPTION

- = twc = 1 ms
F = twc = 200 µs
X = Rotated pinout
T = Tape and Reel

Device Type (Up to 10 Digits)

C = Indicates CMOS
LC = Indicates Low Power CMOS
AA = 1.8V
LV = Low Voltage
HC = High Speed
LCS = Low Power Security

© 1995 Microchip Technology Inc.

MICROCHIP

Packaging Diagrams and Parameters

Table of Contents

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Ceramic Side Brazed Dual In-Line Family

Symbol List for Ceramic Side Brazed Dual In-Line Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body (lid)
A1	Distance between seating plane and base plane
A2	Distance from base plane to highest point of body (lid)
A3	Base body thickness
B	Width of terminal leads
B1	Width of terminal lead shoulder which locate seating plane (standoff geometry optional)
C	Thickness of terminal leads
D	Largest overall package parameter of length
D1	Body length parameter - end lead center to end lead center
E	Largest overall package width parameter outside of lead
E1	Body width parameters not including leads
eA	Linear spacing of true minimum lead position center line to center line
eB	Linear spacing between true lead position outside of lead to outside of lead
e1	Linear spacing between center lines of body standoffs (terminal leads)
L	Distance from seating plane to end of lead
N	Total number of potentially usable lead positions
S	Distance from true position center line of Number 1 lead to the extremity of the body
S1	Distance from other end lead edge positions to the extremity of the body

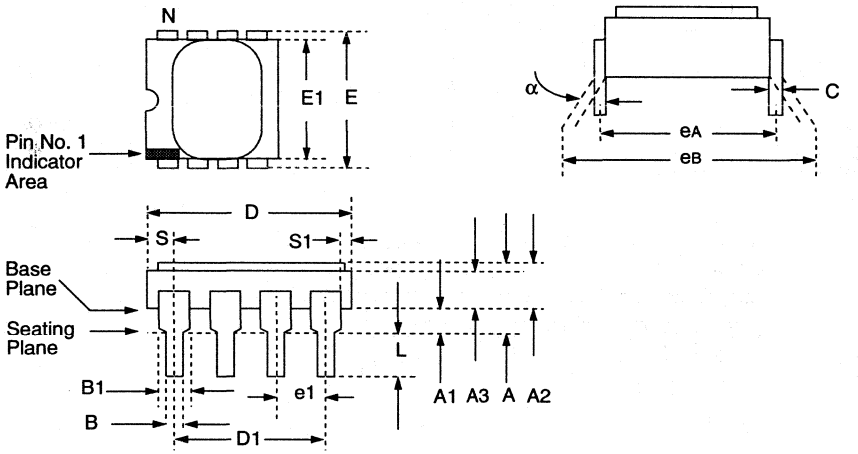
Notes:

1. Controlling parameter: inches.
2. Parameter "e1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by board hole size.
4. Parameter "B1" is nominal.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 8-Lead Ceramic Side Brazed Dual In-Line (300 mil)

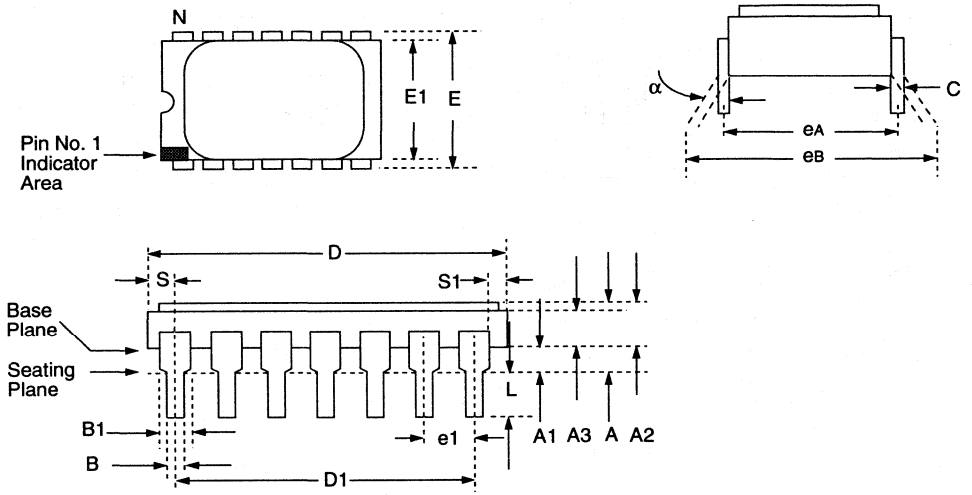


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.302	3.937		0.130	0.155	
A1	0.635	1.143		0.025	0.045	
A2	2.032	2.794		0.080	0.110	
A3	1.778	2.413		0.070	0.095	
B	0.406	0.508		0.016	0.020	
B1	1.371	1.371	Typical	0.054	0.054	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	13.004	13.412		0.512	0.528	
D1	7.416	7.824	Reference	0.292	0.308	Reference
E	7.569	8.230		0.298	0.324	
E1	7.112	7.620		0.280	0.300	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.620	9.652		0.300	0.380	
L	3.302	3.810		0.130	0.150	
N	8	8		8	8	
S	2.540	3.048		0.100	0.120	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 14-Lead Ceramic Side Brazed Dual In-line (300 mil)

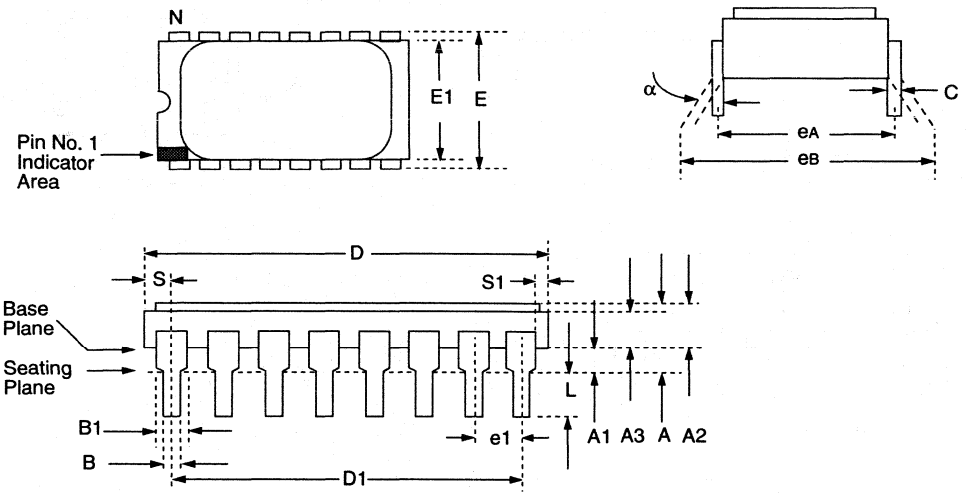


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.302	4.064		0.130	0.160	
A1	0.635	1.143		0.025	0.045	
A2	2.032	2.794		0.080	0.110	
A3	1.778	2.413		0.070	0.095	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.203	0.305	Typical	0.008	0.012	Typical
D	18.796	19.228		0.740	0.757	
D1	15.036	15.444	Reference	0.592	0.608	Reference
E	7.620	8.382		0.300	0.330	
E1	7.061	7.570		0.278	0.298	
e1	2.362	2.744	Typical	0.093	0.108	Typical
eA	7.366	7.874	Reference	0.290	0.310	Reference
eB	7.620	9.652		0.300	0.380	
L	3.175	4.191		0.125	0.165	
N	14	14		14	14	
S	–	2.490		–	0.098	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 16-Lead Ceramic Side Brazed Dual In-line (300 mil)

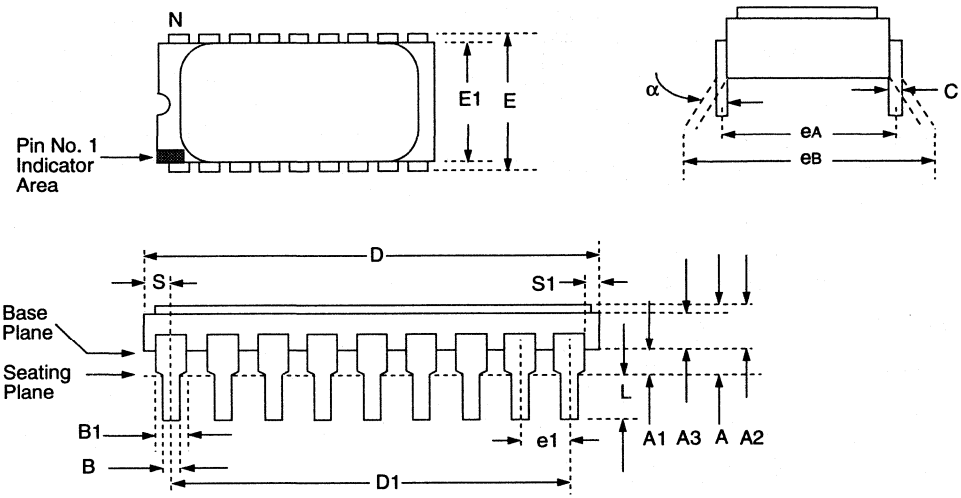


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.302	4.064		0.130	0.160	
A1	0.635	1.143		0.025	0.045	
A2	2.032	2.794		0.080	0.110	
A3	1.778	2.413		0.070	0.095	
B	0.406	0.508		0.016	0.020	
B1	1.371	1.371	Typical	0.054	0.054	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	19.812	20.574		0.780	0.810	
D1	17.653	17.907	Reference	0.695	0.705	Reference
E	7.620	8.382		0.300	0.330	
E1	7.162	7.470		0.282	0.294	
e1	2.413	2.667	Typical	0.095	0.105	Typical
eA	7.366	7.874	Reference	0.290	0.310	Reference
eB	7.620	9.652		0.300	0.380	
L	3.175	4.191		0.125	0.165	
N	16	16		16	16	
S	–	2.032		–	0.080	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 18-Lead Ceramic Side Brazed Dual In-line (300 mil)

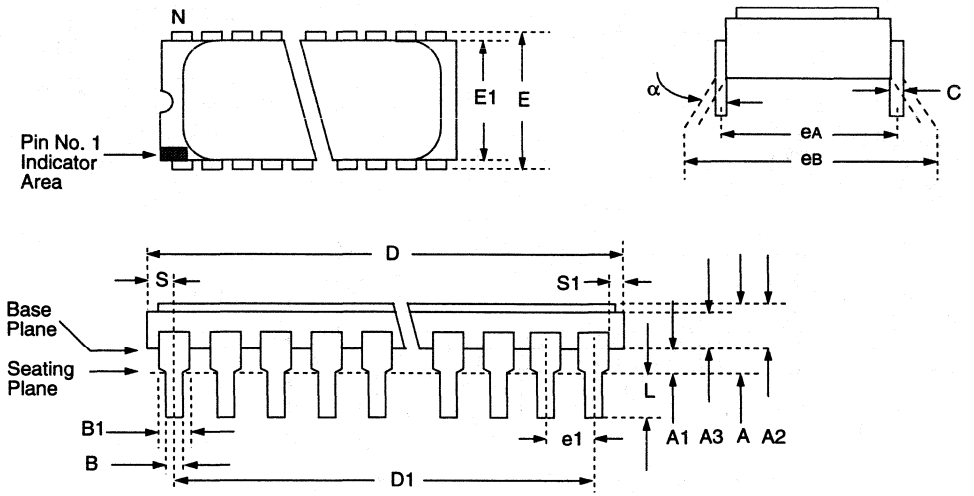


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.302	4.064		0.130	0.160	
A1	0.635	1.143		0.025	0.045	
A2	2.032	2.794		0.080	0.110	
A3	1.778	2.413		0.070	0.095	
B	0.406	0.508		0.016	0.020	
B1	1.371	1.371	**Typical**	0.054	0.054	**Typical**
C	0.228	0.305	**Typical**	0.009	0.012	**Typical**
D	22.352	23.114		0.880	0.910	
D1	20.193	20.447	**Reference**	0.795	0.805	**Reference**
E	7.620	8.382		0.300	0.330	
E1	7.061	7.570		0.278	0.298	
e1	2.413	2.667	**Typical**	0.095	0.105	**Typical**
eA	7.366	7.874	**Reference**	0.290	0.310	**Reference**
eB	7.620	9.652		0.300	0.380	
L	3.175	4.191		0.125	0.165	
N	18	18		18	18	
S	–	2.490		–	0.098	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 22-Lead Ceramic Side Brazed Dual In-line (400 mil)



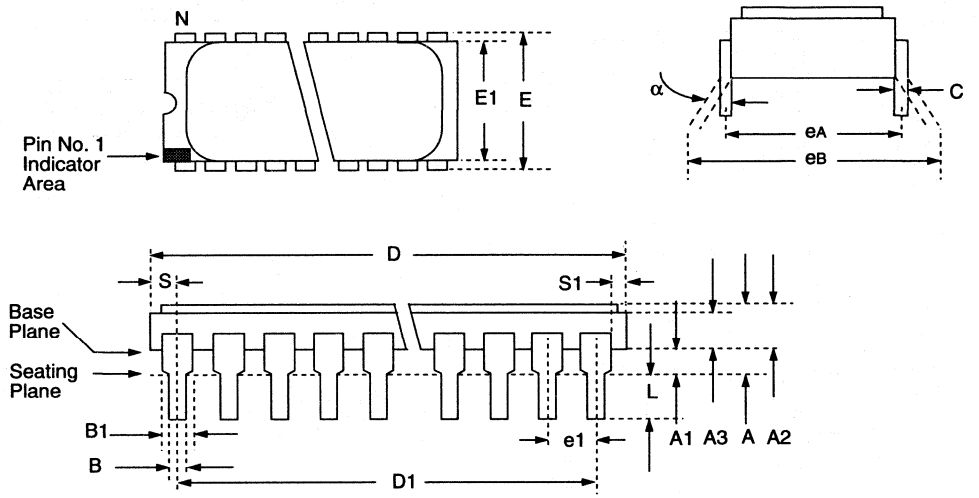
Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	2.667	4.064		0.105	0.160	
A1	0.711	1.220		0.028	0.048	
A2	2.032	3.302		0.080	0.130	
A3	1.778	2.921		0.070	0.115	
B	0.431	0.585		0.017	0.023	
B1	1.016	1.016	Typical	0.040	0.040	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	27.152	27.864		1.069	1.091	
D1	25.296	25.604	Reference	0.992	1.008	Reference
E	10.160	10.922		0.400	0.430	
E1	9.728	9.983		0.383	0.393	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	9.906	10.414	Reference	0.390	0.410	Reference
eB	10.160	12.192		0.400	0.480	
L	3.175	4.191		0.125	0.165	
N	22	22		22	22	
S	–	2.032		–	0.080	
S1	0.127	–		0.005	–	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic Side Brazed Dual In-line (600 mil)

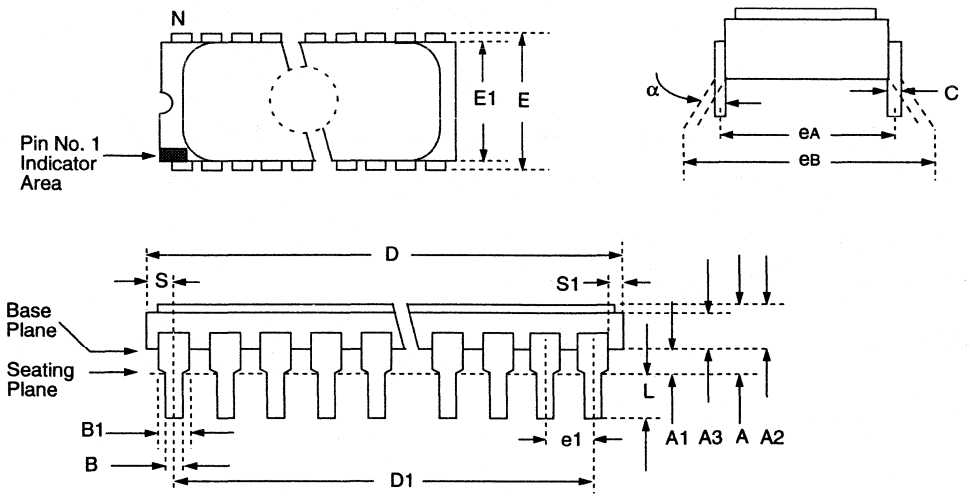


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.445		0.120	0.175	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.778	2.540		0.070	0.100	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	30.175	30.745	*	1.188	1.212	
D1	27.736	28.144	Reference	1.092	1.108	Reference
E	14.986	16.002		0.590	0.630	
E1	14.782	14.936		0.582	0.588	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	14.986	15.748	Reference	0.590	0.620	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	24	24		24	24	
S	–	2.540		–	0.100	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic Side Brazed Dual In-line with Window (600 mil)

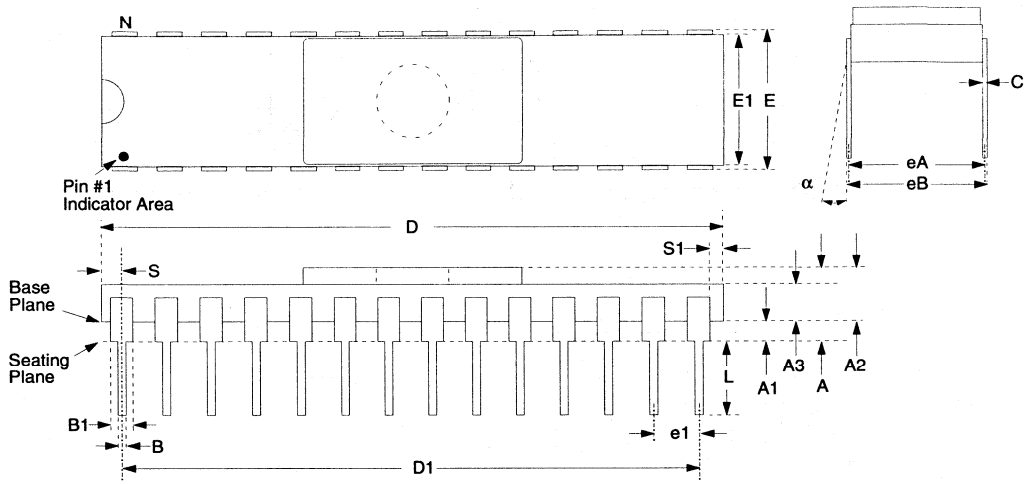


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.445		0.120	0.175	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.778	2.540		0.070	0.100	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	30.175	30.785		1.188	1.212	
D1	27.736	28.144	Reference	1.092	1.108	Reference
E	14.986	16.002		0.590	0.630	
E1	14.782	14.936		0.582	0.588	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	14.986	15.748	Reference	0.590	0.620	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	24	24		24	24	
S	–	2.540		–	0.100	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Side Brazed Dual In-Line with Window (300 mil)

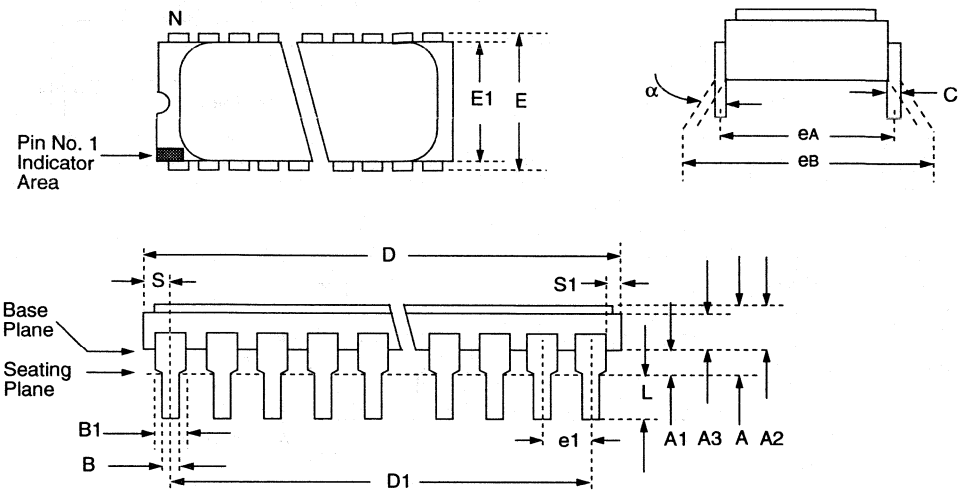


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.937	5.030		0.155	0.198	
A1	1.016	1.524		0.040	0.060	
A2	2.921	3.506		0.115	0.138	
A3	1.930	2.388		0.076	0.094	
B	0.406	0.508		0.016	0.020	
B1	1.219	1.321	Typical	0.048	0.052	
C	0.228	0.305	Typical	0.009	0.012	
D	35.204	35.916		1.386	1.414	
D1	32.893	33.147	Reference	1.295	1.305	
E	7.620	8.128		0.300	0.320	
E1	7.366	7.620		0.290	0.300	
e1	2.413	2.667	Typical	0.095	0.105	
eA	7.366	7.874	Reference	0.290	0.310	
eB	7.594	8.179		0.299	0.322	
L	3.302	4.064		0.130	0.160	
N	28	28		28	28	
S	1.143	1.397		0.045	0.055	
S1	0.533	0.737		0.021	0.029	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Side Brazed Dual In-line (600 mil)



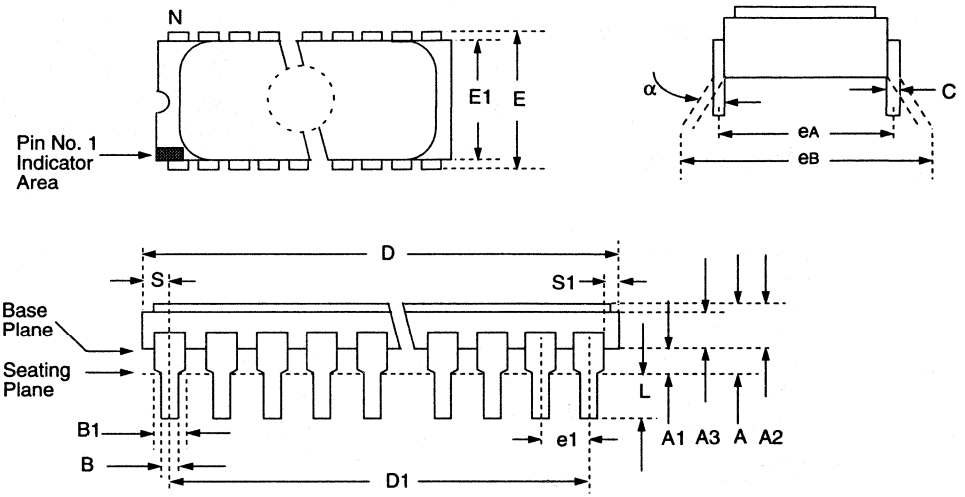
Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.064		0.120	0.160	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.778	2.540		0.070	0.100	
B	0.457	0.508		0.018	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	35.204	35.916		1.386	1.414	
D1	32.816	33.224	Reference	1.292	1.308	Reference
E	14.986	16.002		0.590	0.630	
E1	14.782	15.190		0.582	0.598	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	14.986	15.494	Reference	0.590	0.610	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	28	28		28	28	
S	–	2.540		–	0.100	
S1	0.127	–		0.005	–	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Side Brazed Dual In-line with Window (600 mil)

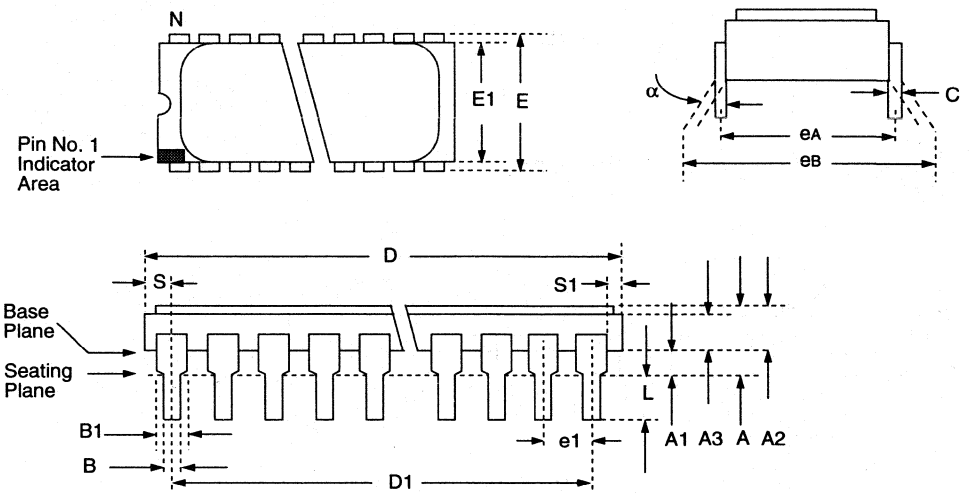


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.064		0.120	0.160	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.778	2.540		0.070	0.100	
B	0.457	0.508		0.018	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	35.204	35.916		1.386	1.414	
D1	32.816	33.224	Reference	1.292	1.308	Reference
E	14.986	16.002		0.590	0.630	
E1	14.782	15.190		0.582	0.598	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	14.986	15.494	Reference	0.590	0.610	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	28	28		28	28	
S	–	2.540		–	0.100	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 40-Lead Ceramic Side Brazed Dual In-line (600 mil)



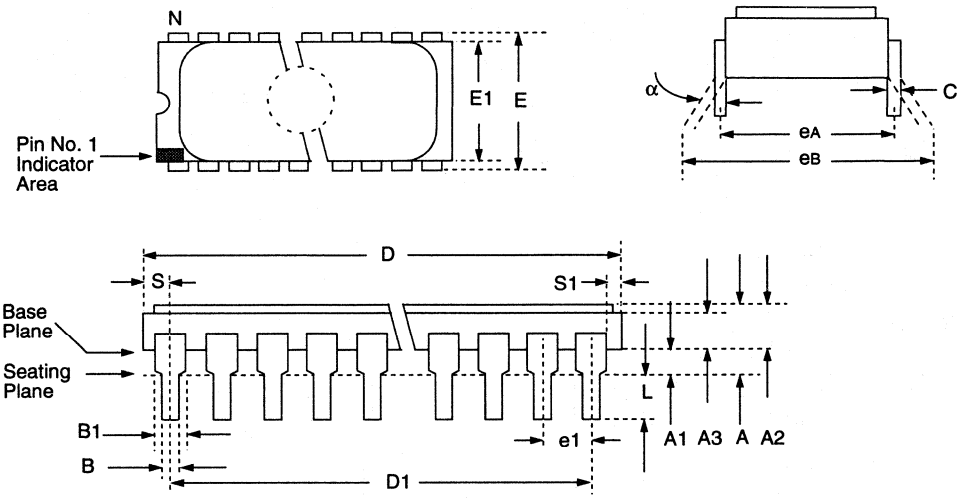
Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.445		0.120	0.175	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.829	2.235		0.072	0.088	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	50.546	51.308		1.990	2.020	
D1	48.056	48.463	Reference	1.892	1.908	Reference
E	15.240	16.256		0.600	0.640	
E1	14.478	15.748		0.570	0.620	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	40	40		40	40	
S	–	2.490		–	0.098	
S1	0.127	–		0.005	–	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 40-Lead Ceramic Side Brazed Dual In-line with Window (600 mil)

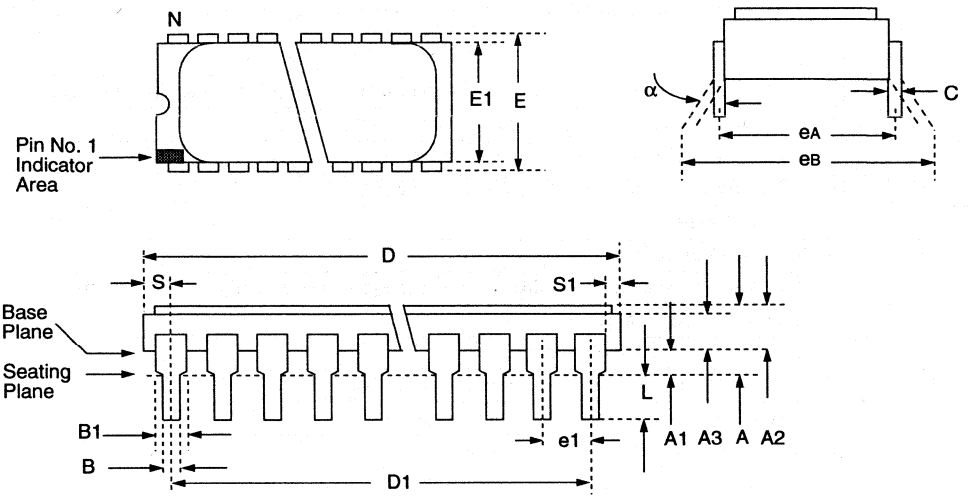


Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.445		0.120	0.175	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.829	2.235		0.072	0.088	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	50.546	51.308		1.990	2.020	
D1	48.056	48.463	Reference	1.892	1.908	Reference
E	15.240	16.256		0.600	0.640	
E1	14.478	15.748		0.570	0.620	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	40	40		40	40	
S	–	2.490		–	0.098	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 48-Lead Ceramic Side Brazed Dual In-line (600 mil)



Package Group: Ceramic Side Brazed Dual In-Line (CER)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.048	4.445		0.120	0.175	
A1	1.016	1.524		0.040	0.060	
A2	2.032	2.921		0.080	0.115	
A3	1.829	2.235		0.072	0.088	
B	0.406	0.508		0.016	0.020	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.228	0.305	Typical	0.009	0.012	Typical
D	60.350	61.570		2.376	2.424	
D1	58.216	58.624	Reference	2.292	2.308	Reference
E	15.240	16.256		0.600	0.640	
E1	14.478	15.748		0.570	0.620	
e1	2.336	2.744	Typical	0.092	0.108	Typical
eA	15.240	15.290	Reference	0.600	0.600	Reference
eB	14.986	16.256		0.590	0.640	
L	3.302	4.064		0.130	0.160	
N	48	48		48	48	
S	–	2.490		–	0.100	
S1	0.127	–		0.005	–	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Ceramic CERDIP Dual In-Line Family

Symbol List for Ceramic CERDIP Dual In-Line Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body (lid)
A1	Distance between seating plane and base plane
A2	Distance from base plane to highest point of body (lid)
A3	Base body thickness
B	Width of terminal leads
B1	Width of terminal lead shoulder which locate seating plane (standoff geometry optional)
C	Thickness of terminal leads
D	Largest overall package parameter of length
D1	Body length parameter - end lead center to end lead center
E	Largest overall package width parameter outside of lead
E1	Body width parameters not including leads
eA	Linear spacing of true minimum lead position center line to center line
eB	Linear spacing between true lead position outside of lead to outside of lead
e1	Linear spacing between center lines of body standoffs (terminal leads)
L	Distance from seating plane to end of lead
N	Total number of potentially usable lead positions
S	Distance from true position center line of Number 1 lead to the extremity of the body
S1	Distance from other end lead edge positions to the extremity of the body

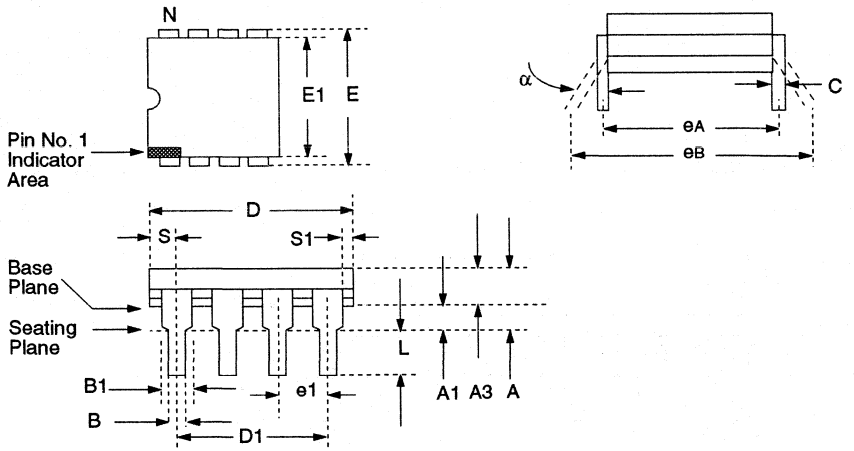
Notes:

1. Controlling parameter: inches.
2. Parameter "e1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by board hole size.
4. Parameter "B1" is nominal.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 8-Lead Ceramic CERDIP Dual In-line (300 mil)



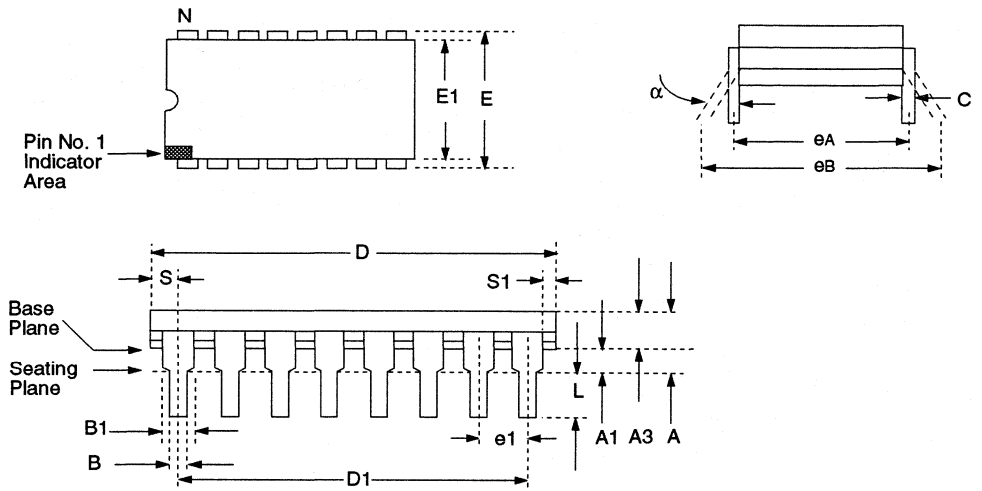
Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.080		—	0.200	
A1	0.381	1.524		0.015	0.060	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	9.398	10.287		0.370	0.405	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	7.620	8.255		0.300	0.325	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	7.366	8.128	Reference	0.290	0.320	Reference
eB	7.620	10.160		0.300	0.400	
L	3.175	3.810		0.125	0.150	
N	8	8		8	8	
S	0.508	1.397		0.020	0.055	
S1	0.381	1.270		0.015	0.050	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 16-Lead Ceramic CERDIP Dual In-line (300 mil)

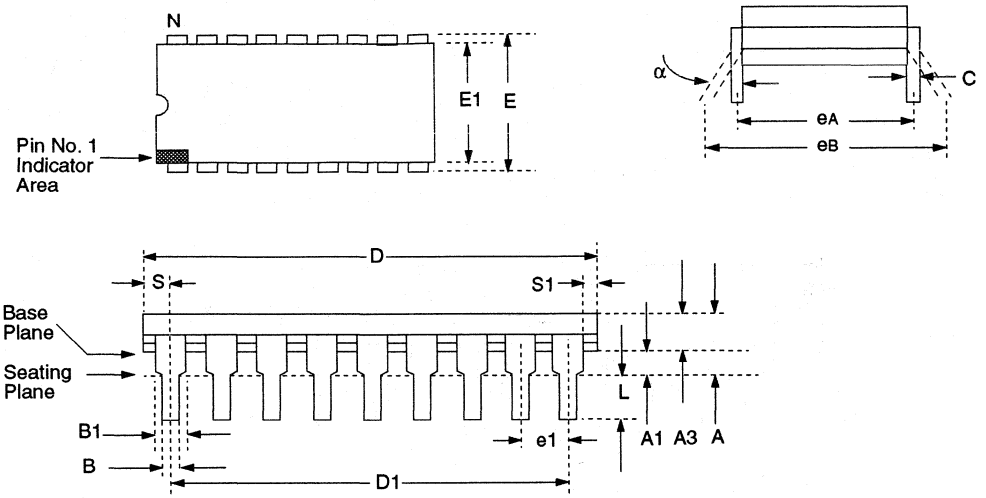


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.191	5.080		0.165	0.200	
A1	0.381	1.524		0.015	0.060	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	19.050	20.320		0.750	0.800	
D1	17.780	17.780	Reference	0.700	0.700	Reference
E	7.493	8.255		0.295	0.325	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	7.366	8.128	Reference	0.290	0.320	Reference
eB	7.62	10.160		0.300	0.400	
L	3.175	3.810		0.125	0.150	
N	16	16		16	16	
S	0.508	1.397		0.020	0.055	
S1	0.381	1.270		0.015	0.050	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 18-Lead Ceramic CERDIP Dual In-line (300 mil)



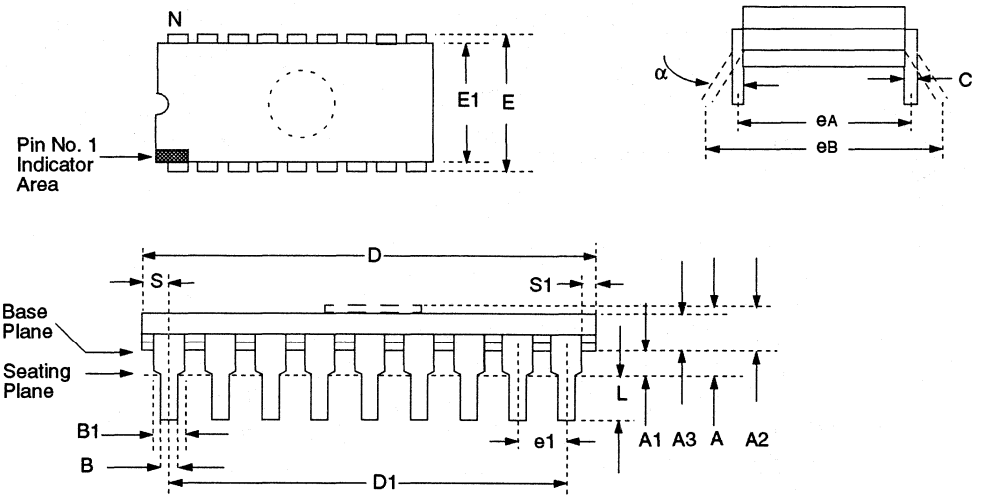
Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.080		—	0.200	
A1	0.381	1.778		0.015	0.070	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	22.352	23.622		0.880	0.930	
D1	20.320	20.320	Reference	0.800	0.800	Reference
E	7.620	8.382		0.300	0.330	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540		0.100	0.100	
eA	7.366	8.128	Typical	0.290	0.320	Typical
eB	7.620	10.160	Reference	0.300	0.400	Reference
L	3.175	3.810		0.125	0.150	
N	18	18		18	18	
S	0.508	1.397		0.020	0.055	
S1	0.381	1.270		0.015	0.050	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 18-Lead Ceramic CERDIP Dual In-line with Window (300 mil)

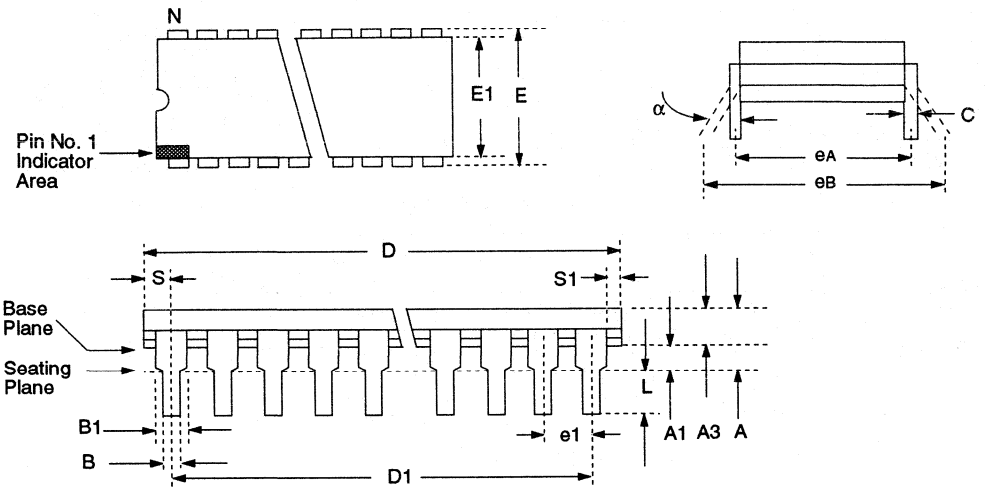


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.080		—	0.200	
A1	0.381	1.7780		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	22.352	23.622		0.880	0.930	
D1	20.320	20.320	Reference	0.800	0.800	Reference
E	7.620	8.382		0.300	0.330	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	7.366	8.128	Typical	0.290	0.320	Typical
eB	7.620	10.160		0.300	0.400	
L	3.175	3.810		0.125	0.150	
N	18	18		18	18	
S	0.508	1.397		0.020	0.055	
S1	0.381	1.270		0.015	0.050	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 22-Lead Ceramic CERDIP Dual In-line (400 mil)



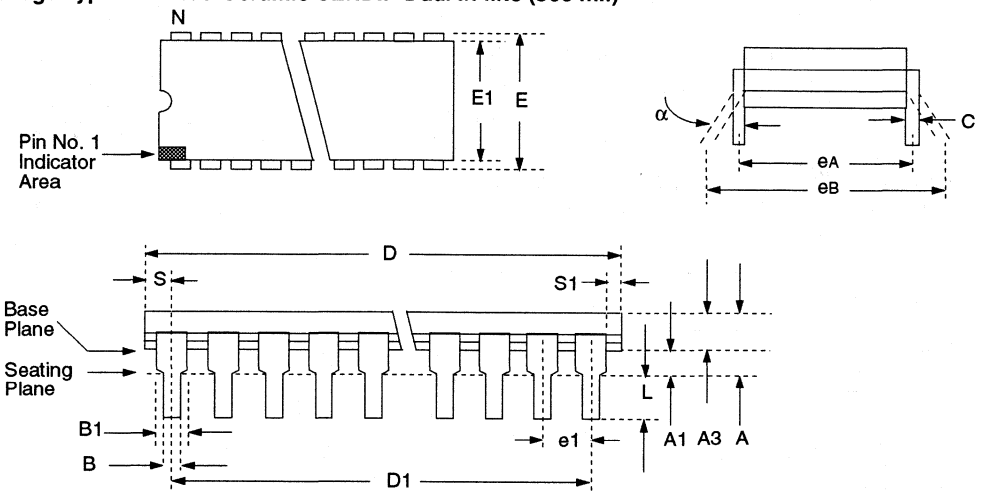
Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.715		—	0.225	
A1	0.381	1.778		0.015	0.070	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	26.670	27.940		1.050	1.100	
D1	25.400	25.400	Reference	1.000	1.000	Reference
E	10.160	10.922		0.400	0.430	
E1	8.890	10.414		0.350	0.410	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	9.906	10.668	Typical	0.390	0.420	Typical
eB	10.160	12.700		0.400	0.500	
L	3.175	3.810		0.125	0.150	
N	18	18		22	22	
S	—	1.270		—	0.050	
S1	0.127	1.270		0.005	0.050	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic CERDIP Dual In-line (300 mil)

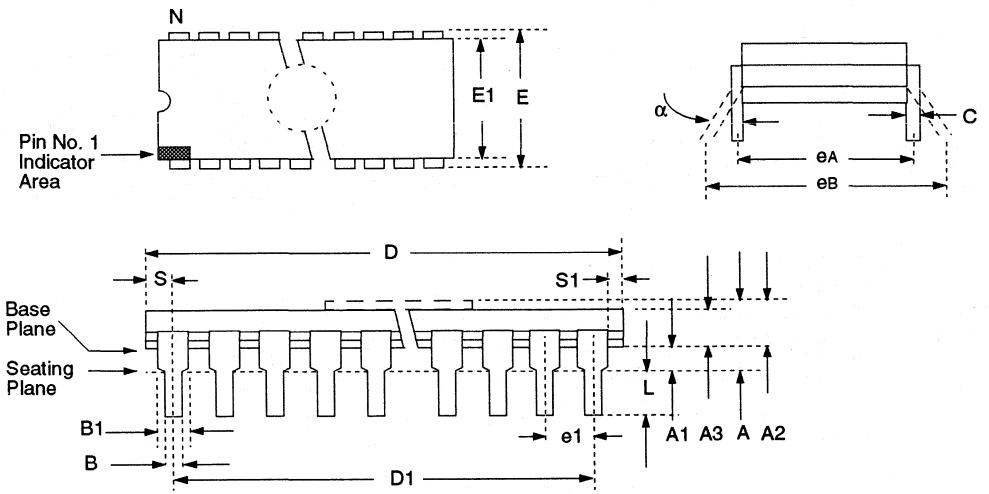


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.715		—	0.225	
A1	0.381	1.905		0.015	0.075	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	31.115	32.385		1.225	1.275	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	7.620	8.382		0.300	0.330	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	7.366	8.128	Typical	0.290	0.320	Typical
eB	7.620	11.430		0.300	0.450	
L	3.175	3.810		0.125	0.150	
N	24	24		24	24	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic CERDIP Dual In-line with Window (300 mil)

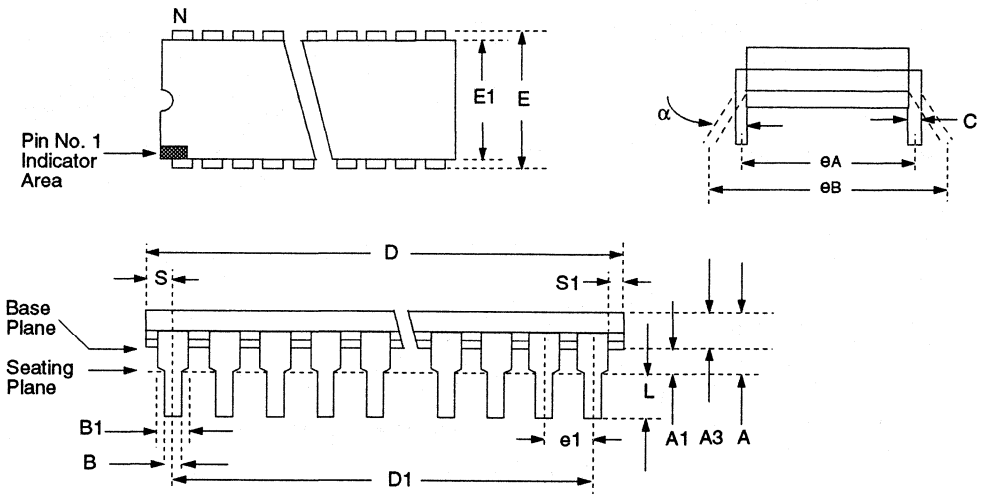


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.715		—	0.225	
A1	0.381	1.905		0.015	0.075	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	31.115	32.385		1.225	1.275	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	7.620	8.382		0.300	0.330	
E1	5.588	7.874		0.220	0.310	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	7.366	8.128	Typical	0.290	0.320	Typical
eB	7.620	11.430		0.300	0.450	
L	3.175	3.810		0.125	0.150	
N	24	24		24	24	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic CERDIP Dual In-line (600 mil)

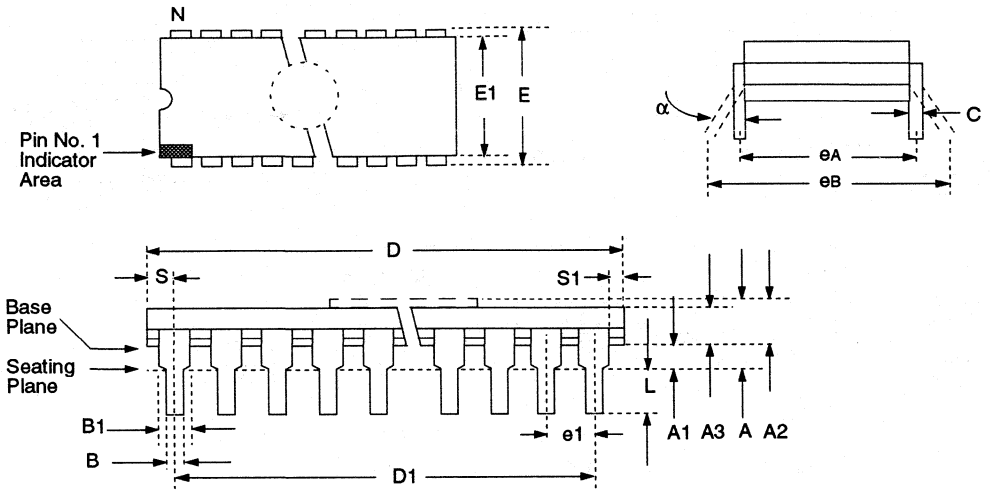


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.715		—	0.225	
A1	0.381	1.905		0.015	0.075	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	31.115	32.385		1.225	1.275	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	15.748	Typical	0.590	0.620	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	24	24		24	24	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Ceramic CERDIP Dual In-line with Window (600 mil)

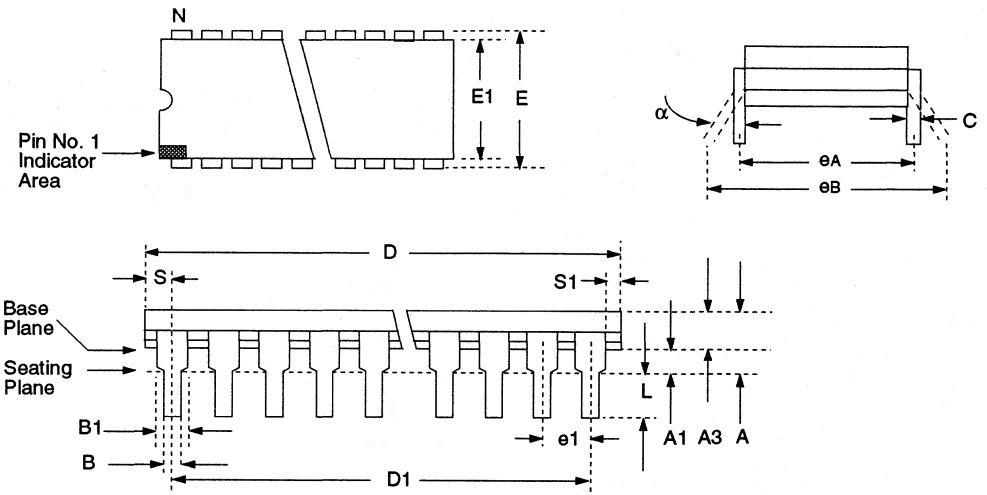


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.715		—	0.225	
A1	0.381	1.905		0.015	0.075	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	31.115	32.385		1.225	1.275	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	15.748	Typical	0.590	0.620	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	24	24		24	24	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic CERDIP Dual In-line (600 mil)

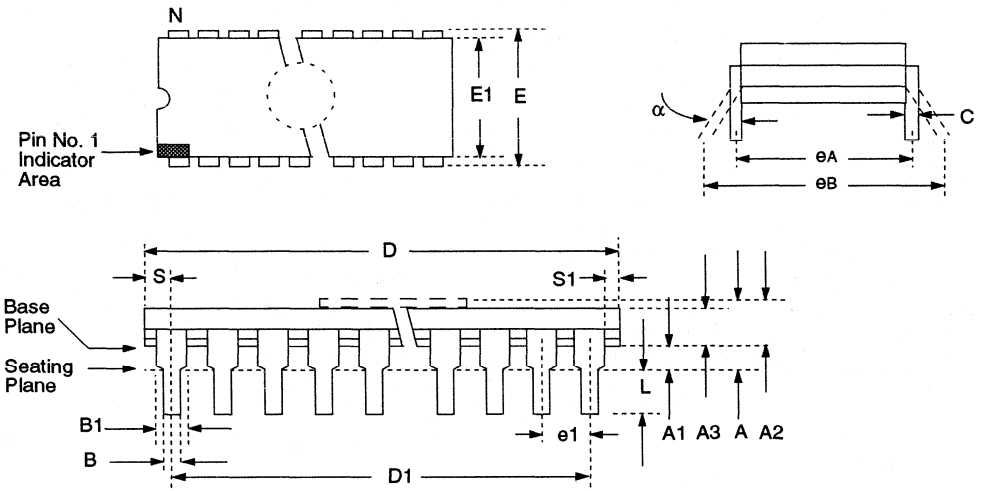


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.461		—	0.215	
A1	0.381	1.524		0.015	0.060	
A2	—	—	Ref. A3	—	—	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	36.195	37.465		1.425	1.475	
D1	33.020	33.020	Reference	1.300	1.300	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	15.748	Typical	0.590	0.620	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	28	28		28	28	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic CERDIP Dual In-line with Window (600 mil)

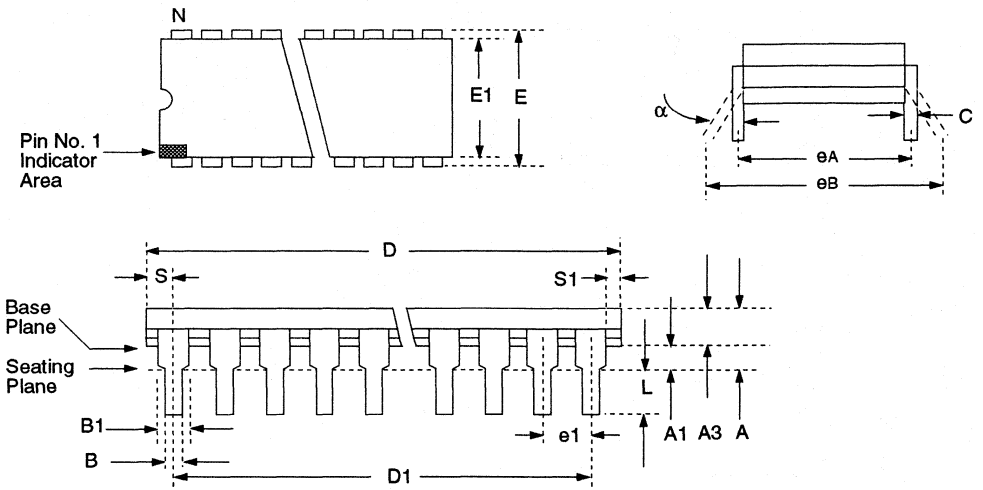


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	—	5.461		—	0.215	
A1	0.381	1.524		0.015	0.060	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	36.195	37.465		1.425	1.475	
D1	33.020	33.020	Reference	1.300	1.300	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	14.986	15.748	Reference	0.590	0.620	Reference
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	28	28		28	28	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 40-Lead Ceramic CERDIP Dual In-line (600 mil)

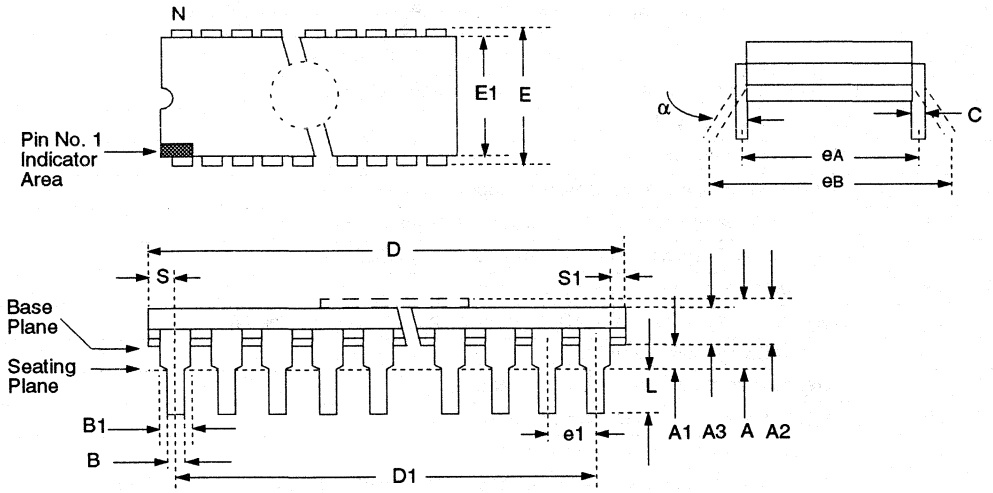


Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	–	–	Ref. A3	–	–	Ref. A3
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	14.986	16.002	Reference	0.590	0.630	Reference
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	40	40		40	40	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 40-Lead Ceramic CERDIP Dual In-line with Window (600 mil)



Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	Typical	0.050	0.065	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	Reference	0.100	0.100	Reference
eA	14.986	16.002	Typical	0.590	0.630	Typical
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
N	40	40		40	40	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Ceramic Flatpack Family

Symbol List for Ceramic Flatpack Package Parameters	
Symbol	**Description of Parameters**
A	Distance between seating plane to highest point of body (lid)
B	Width of terminal leads
C	Thickness of terminal leads
D	Largest overall package parameter of length
D1	Body length parameter - end lead center to end lead center
E	Largest overall package width parameter outside of lead
E2, E3	Body width parameters not including leads
e	Linear spacing between center lines of body standoffs (terminal leads)
H	Other package width parameter
L	Distance from seating plane to end of lead
N	Total number of potentially usable lead positions
Q	Distance between seating plane and lead
S	Distance from true position center line of Number 1 lead to the extremity of the body
S1	Distance from other end lead edge positions to the extremity of the body

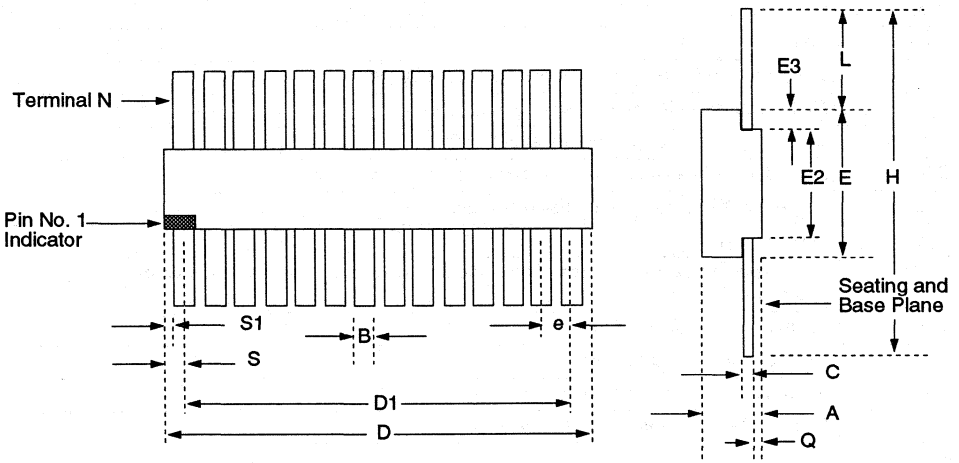
Notes:

1. Controlling parameter: inches.
2. Parameter "e1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by board hole size.
4. Parameters "B" and "C" are nominal.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Flatpack



Package Group: Ceramic Flatpack (CFPK)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	2.286	3.302		0.090	0.130	
B	0.381	0.482		0.015	0.019	Typical
C	0.076	0.153		0.003	0.006	Typical
D	17.780	18.796		0.700	0.740	
D1	16.306	16.714		0.642	0.658	Reference
E	9.652	10.668		0.380	0.420	
E2	4.572	–		0.180	–	
E3	0.762	–		0.030	–	
e	1.270	1.270	BSC	0.050	0.050	Typical
H	22.352	29.464		0.880	1.160	
L	6.350	9.398		0.250	0.370	
N	28	28		28	28	
Q	0.660	1.143		0.026	0.045	
S	0.889	1.016		0.035	0.040	
S1	0.254	0.381		0.010	0.015	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Ceramic Leadless Chip Carrier Family

Symbol List for Ceramic Leadless Chip Carrier Package Parameters	
Symbol	**Description of Parameters**
A	Thickness of base body
A1	Total package height
A2	Distance from base body to highest point of body (lid)
B	Width of terminal lead pin
D	Largest overall package parameter of length
D1, E1	Body length dimension - end lead center to end lead center
E	Largest overall package dimension of width
e	Linear spacing
e1	Linear spacing between edges of true lead positions (of corner terminal lead pads) lead corner to lead corner
h	Depth of major index feature
j	Width of minor index feature
L	Distance from package edge to end of effective pad
N	Total number of potentially usable lead positions

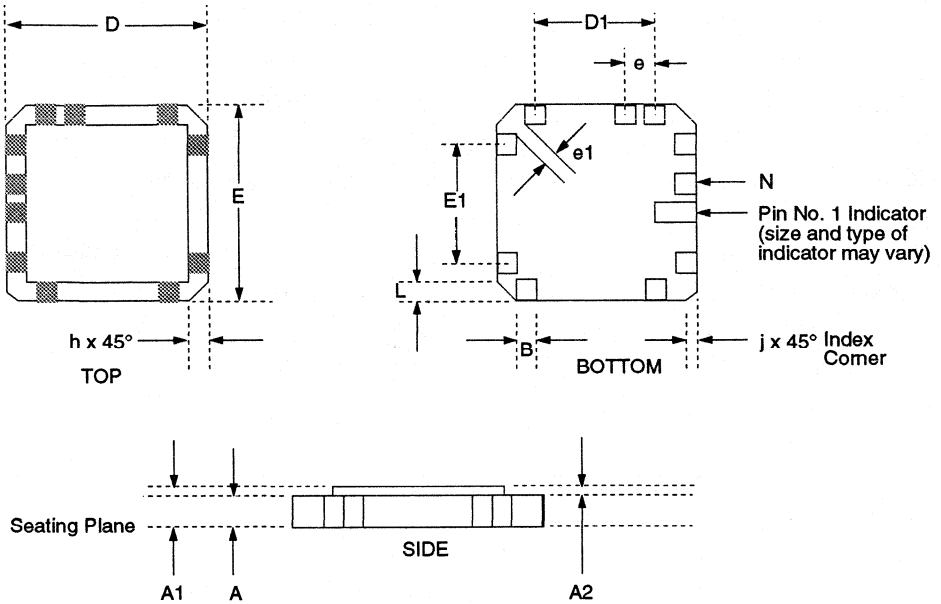
Notes:

1. Controlling dimension: inches.
2. Parameter "e1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by PC board hole size.
4. Parameter "B" is nominal.
5. Corner configuration optional.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Leadless Chip Carrier

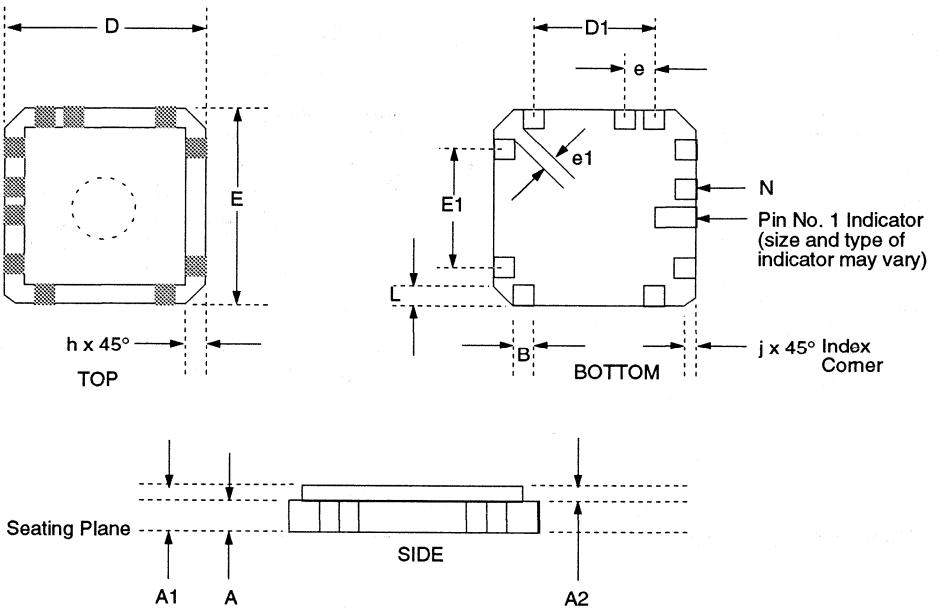


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	1.651	2.540		0.065	0.100	
A2	0.254	0.381		0.010	0.015	
B	0.558	0.712	Typical	0.022	0.028	Typical
D	11.226	11.684		0.442	0.460	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	11.226	11.684		0.442	0.460	
E1	7.620	7.620	Typical	0.300	0.300	Typical
e	1.270	1.270	Reference	0.050	0.050	Reference
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	28	28		28	28	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Ceramic Leadless Chip Carrier with Window

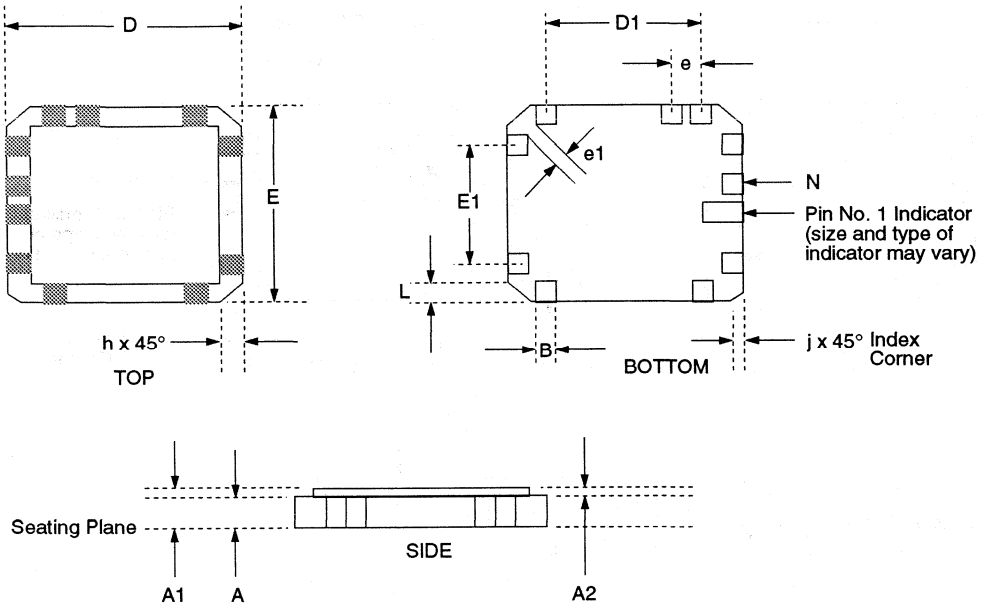


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	2.286	2.540		0.090	0.100	
A2	0.889	1.143		0.035	0.045	
B	0.558	0.712	Typical	0.022	0.028	Typical
D	11.226	11.684		0.442	0.460	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	11.226	11.684		0.442	0.460	
E1	7.620	7.620	Reference	0.300	0.300	Reference
e	1.270	1.270	Typical	0.050	0.050	Typical
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	28	28		28	28	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 32-Lead Ceramic Leadless Chip Carrier

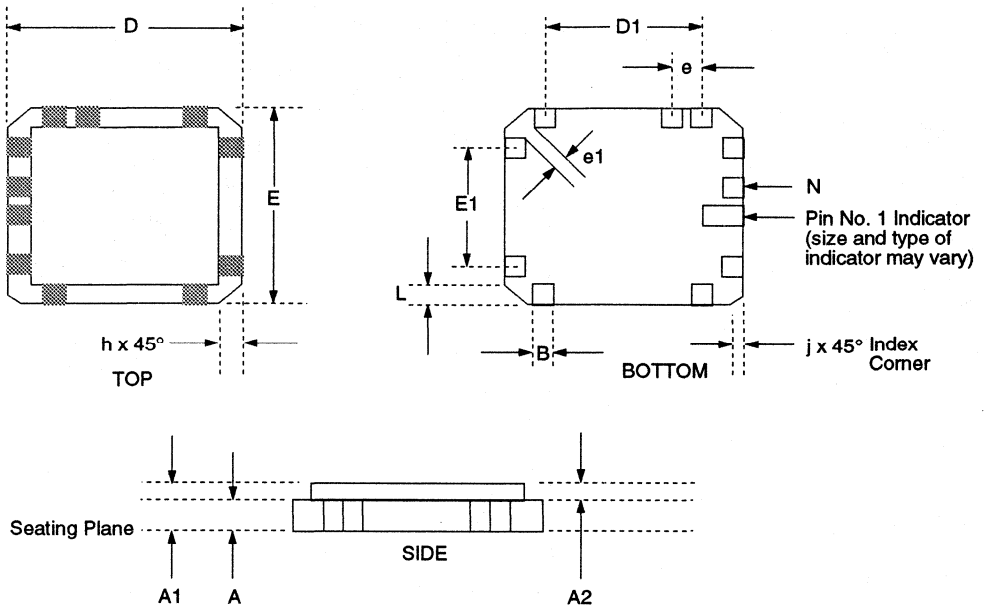


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	2.540	3.048		0.100	0.120	
A2	0.254	0.381		0.010	0.015	
B	0.635	0.661	Typical	0.025	0.026	Typical
D	13.716	14.224		0.540	0.560	
D1	9.982	10.338	Reference	0.393	0.407	Reference
E	11.226	11.634		0.442	0.458	
E1	7.442	7.798	Reference	0.293	0.307	Reference
e	1.270	1.270	Typical	0.050	0.050	Typical
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	32	32		32	32	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 32-Lead Ceramic Leadless Chip Carrier - FRIT

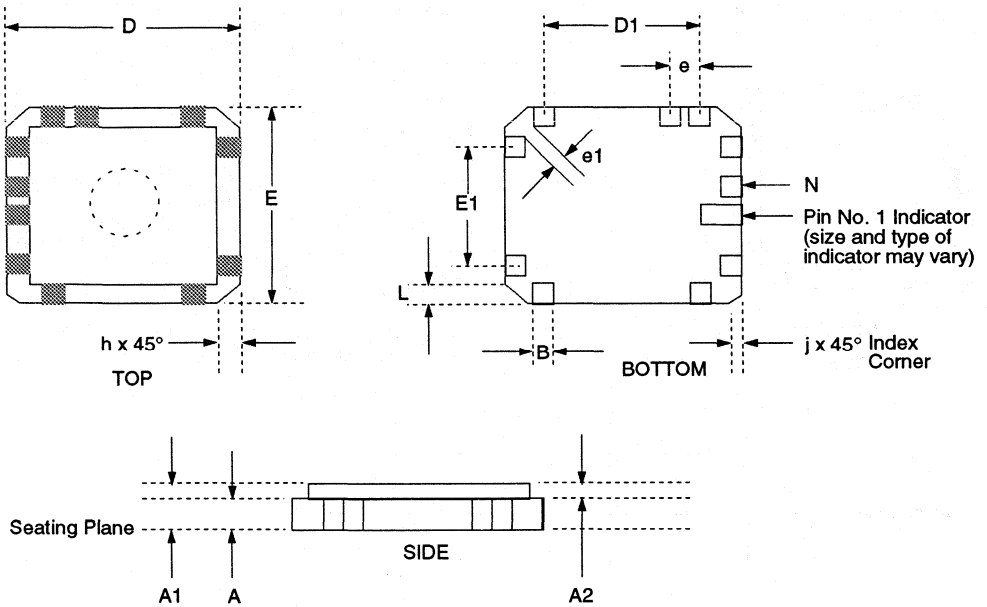


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	2.286	3.302		0.090	0.130	
A2	0.635	1.143		0.025	0.045	
B	0.558	0.712	Typical	0.022	0.028	Typical
D	13.716	14.224		0.540	0.560	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	11.226	11.634		0.442	0.458	
E1	10.160	10.160	Reference	0.400	0.400	Reference
e	1.270	1.270	Reference	0.050	0.050	Reference
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	32	32		32	32	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 32-Lead Ceramic Leadless Chip Carrier with Window

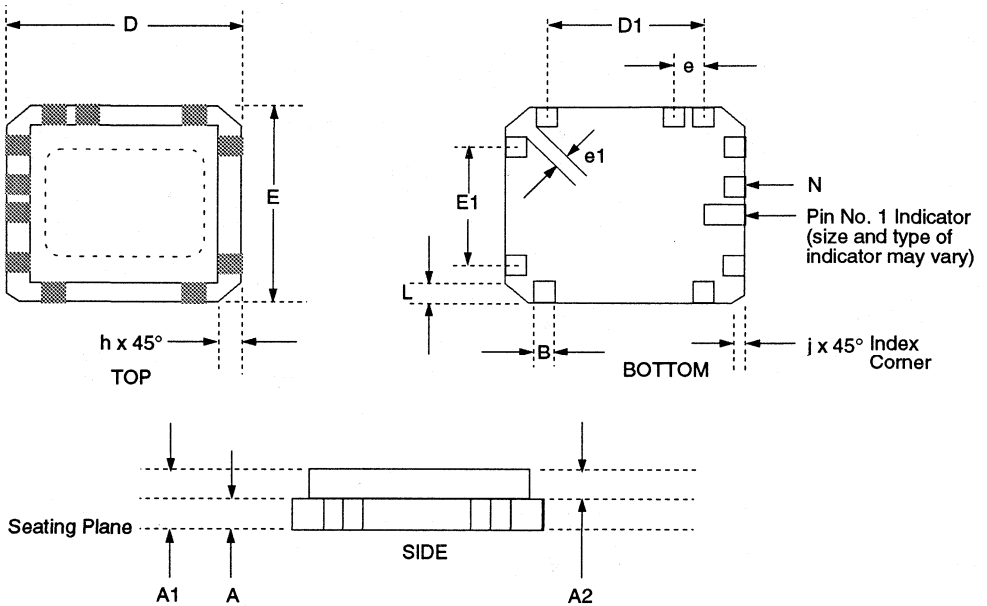


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	2.286	3.302		0.090	0.130	
A2	0.889	1.143		0.035	0.045	
B	0.558	0.712	Typical	0.022	0.028	Typical
D	13.716	14.224		0.540	0.560	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	11.226	11.634		0.442	0.458	
E1	10.160	10.160	Reference	0.400	0.400	Reference
e	1.270	1.270	Reference	0.050	0.050	Reference
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	32	32		32	32	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 32-Lead Ceramic Leadless FRIT-Seal Chip Carrier with Window

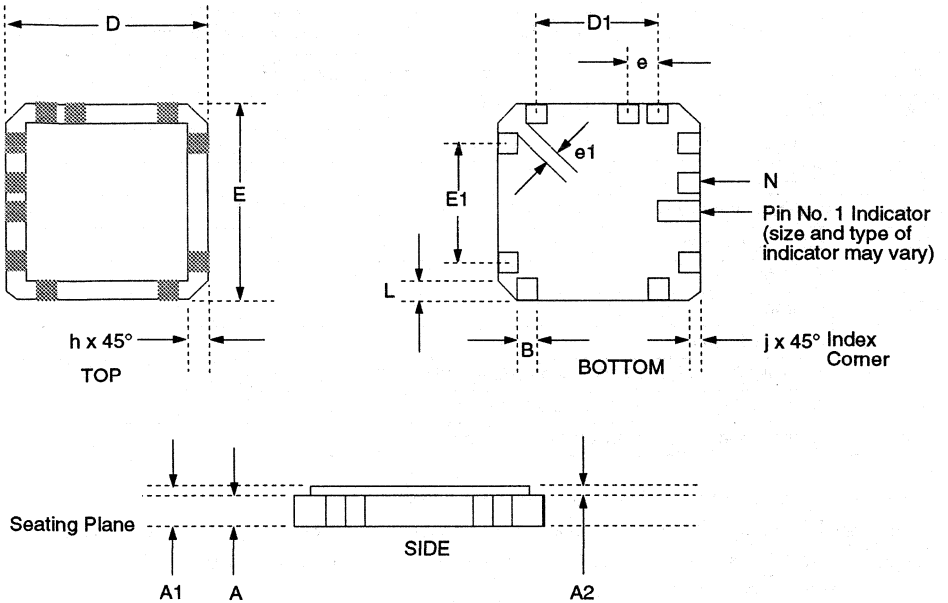


Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.397	2.159		0.055	0.085	
A1	2.286	3.302		0.090	0.130	
A2	0.889	1.143		0.035	0.045	
B	0.558	0.712	Typical	0.022	0.028	Typical
D	13.716	14.224		0.540	0.560	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	11.226	11.634		0.442	0.458	
E1	10.160	10.160	Reference	0.400	0.400	Reference
e	1.270	1.270	Typical	0.050	0.050	Typical
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	32	32		32	32	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 44-Lead Ceramic Leadless Chip Carrier



Package Group: Ceramic Leadless Chip Carrier (LCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	1.371	2.083		0.054	0.082	
A1	1.778	3.048		0.070	0.120	
A2	0.254	1.143		0.010	0.045	
B	0.584	0.712	Typical	0.023	0.028	Typical
D	16.256	16.815		0.640	0.662	
D1	12.700	12.700	Reference	0.500	0.500	Reference
E	16.256	16.815		0.640	0.662	
E1	12.700	12.700	Reference	0.500	0.500	Reference
e	1.270	1.270	Typical	0.050	0.050	Typical
e1	0.381	–	Typical	0.015	–	Typical
h	1.016	1.016	Reference	0.040	0.040	Reference
j	0.508	0.508	Reference	0.020	0.020	Reference
L	1.143	1.397	Typical	0.045	0.055	Typical
N	44	44		44	44	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Ceramic Leaded Chip Carrier Family

Symbol List for Ceramic Leaded Chip Carrier Package Parameters	
Symbol	**Description of Parameters**
A	Distance between seating plane to highest point of body
A1	Distance from lead shoulder to seating plane
CP	Seating plane coplanarity
D/E	Outside dimension
D1/E1	Body dimension
D2/E2	Footprint
D3/E3	Footprint
LT	Lead thickness
N	Total number of potentially usable lead positions
Nd	Total number of leads on short side (rectangular)
Ne	Total number of leads on long side (rectangular)

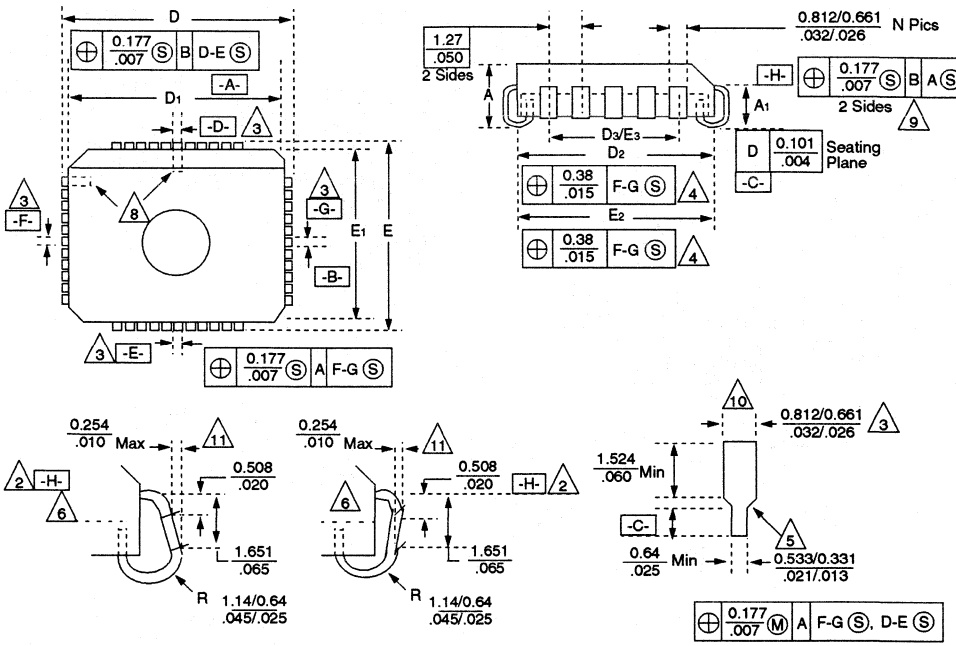
Notes:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane -H- located at top of parting line and coincident with top of lead. Where lead exits body.
3. Datums D-E and F-G to be determined where center leads exit body at datum plane -H-.
4. To be determined at seating plane -C-.
5. Transition is optional.
6. Square: Details of pin1 identifier are optional but must be located within one of the two zones indicated. If the number of terminals on a side is odd terminal 1 is the center terminal.
   Rectangle: Details of pin1 are optional but must be located within zone indicated. If the number of terminals on a side is odd, terminal 1 is the center terminal.
7. Location to datums -A- and -B- to be determined at plane -H-.
8. All dimensions and tolerances include lead trim offset and lead finish.
9. These two dimensions determine maximum angle of the lead for certain socket applications. If unit is intended to be socketed, it is advisable to review these dimensions with the socket supplier.
10. Controlling dimension: inches.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 68-Lead Ceramic Leaded Chip Carrier (Window)

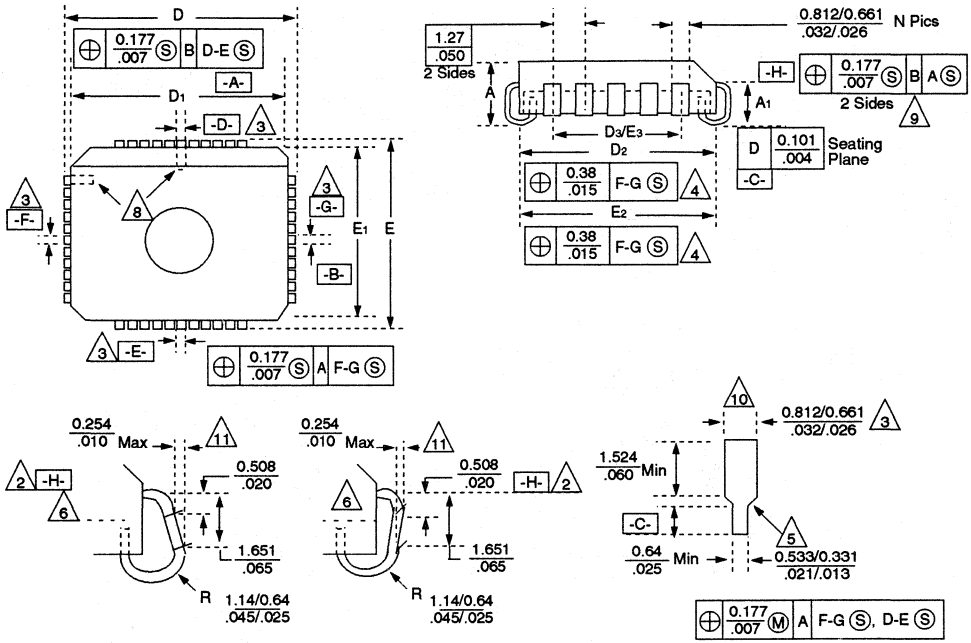


Package Group: Ceramic Leaded Chip Carrier (CLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.699		0.165	0.185	
A1	2.286	3.048		0.090	0.120	
D	24.968	25.222		0.983	0.993	
D1	23.977	24.333		0.944	0.958	
D2	22.860	23.876		0.900	0.940	
D3	20.320	-	Reference	0.800	-	Reference
E	24.968	25.222		0.983	0.993	
E1	23.977	24.333		0.944	0.958	
E2	22.860	23.876		0.900	0.940	
E3	20.320	-	Reference	0.800	-	Reference
N	68	-		68	-	
CP	-	0.102		-	0.004	
LT	0.152	0.204		0.006	0.008	

8

Packaging Diagrams and Parameters

Package Type: 84-Lead Ceramic Leaded Chip Carrier (Window)



Package Group: Ceramic Leaded Chip Carrier (CLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.699		0.165	0.185	
A1	2.286	3.048		0.090	0.120	
D	30.048	30.353		1.183	1.195	
D1	28.829	29.591		1.135	1.165	
D2	27.940	28.956		1.100	1.140	
D3	25.400	-	Reference	1.000	-	Reference
E	30.048	30.353		1.183	1.195	
E1	28.829	29.591		1.135	1.165	
E2	27.940	28.956		1.100	1.140	
E3	25.400	-	Reference	1.000	-	Reference
N	84	-		84	-	
CP	-	0.102		-	0.004	
LT	0.152	0.204		0.006	0.008	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Dual In-Line Family

Symbol List for Plastic In-Line Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body
A1	Distance between seating plane and base plane
A2	Base body thickness
B	Width of terminal leads
B1	Width of terminal lead shoulder which locate seating plane (standoff geometry optional)
C	Thickness of terminal leads
D	Largest overall package parameter of length
D1	Body length parameter - end lead center to end lead center
E	Largest overall package width parameter outside of lead
E1	Body width parameters not including leads
eA	Linear spacing of true minimum lead position center line to center line
eB	Linear spacing between true lead position outside of lead to outside of lead
e1	Linear spacing between center lines of body standoffs (terminal leads)
L	Distance from seating plane to end of lead
N	Total number of potentially usable lead positions
S	Distance from true position center line of Number 1 lead to the extremity of the body
S1	Distance from other end lead edge positions to the extremity of the body

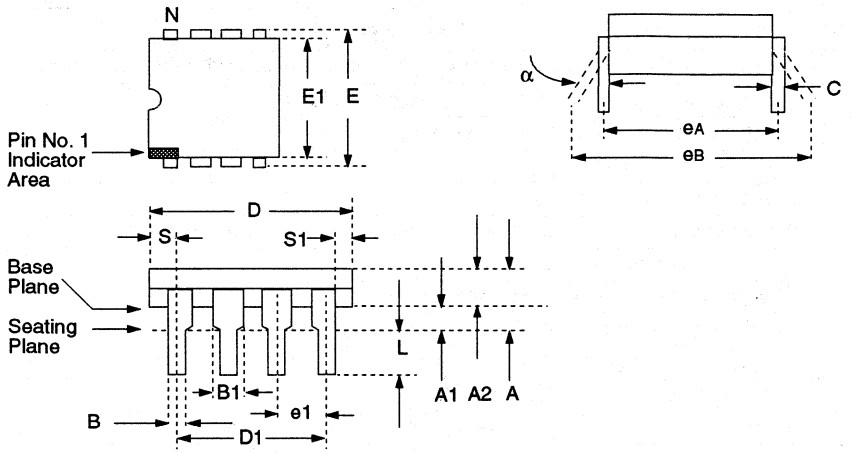
Notes:

1. Controlling parameter: inches.
2. Parameter "e1" ("e") is non-cumulative.
3. Seating plane (standoff) is defined by board hole size.
4. Parameter "B1" is nominal.
5. Details of pin Number 1 identifier are optional.
6. Parameters "D + E1" do not include mold flash/protrusions.
   Mold flash or protrusions shall not exceed .010 inches.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 8-Lead Plastic Dual In-line (300 mil)

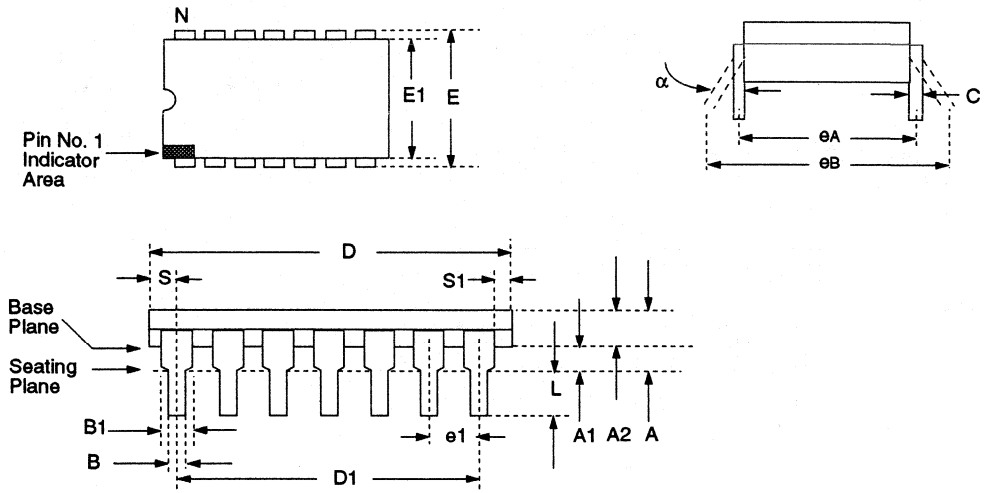


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.397	1.651		0.055	0.065	
C	0.203	0.381	Typical	0.008	0.015	Typical
D	9.017	10.922		0.355	0.430	
D1	7.620	7.620	Reference	0.300	0.300	Reference
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	8	8		8	8	
S	0.889	–		0.035	–	
S1	0.254	–		0.010	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 14-Lead Plastic Dual In-line (300 mil)

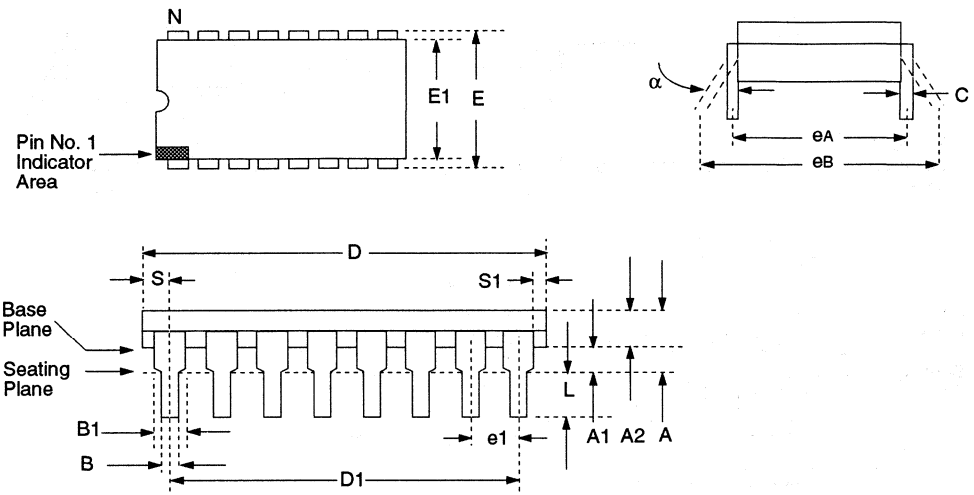


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	Reference	0.060	0.060	Reference
C	0.203	0.381	Typical	0.008	0.015	Typical
D	18.415	19.431		0.725	0.765	
D1	15.240	15.240	Reference	0.600	0.600	Reference
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	14	14		14	14	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 16-Lead Plastic Dual In-line (300 mil)

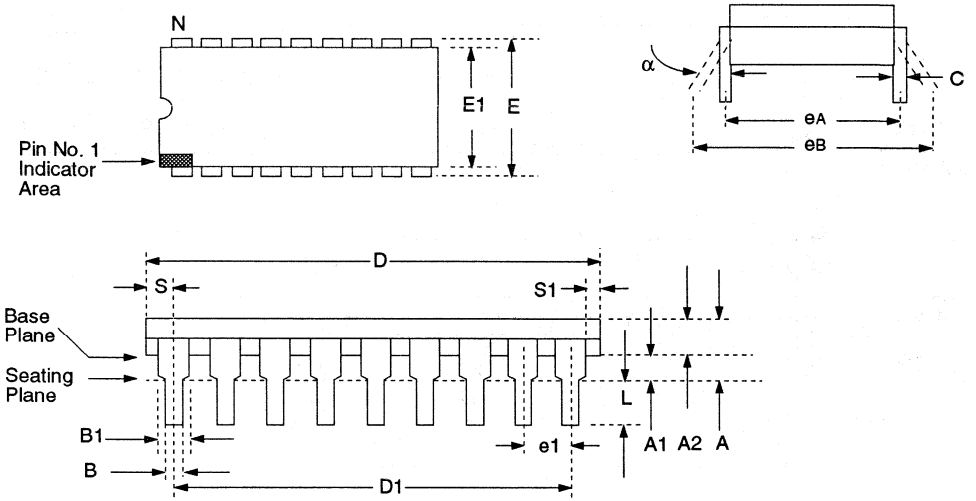


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	Reference	0.060	0.060	Reference
C	0.203	0.381	Typical	0.008	0.015	Typical
D	18.923	19.939		0.745	0.785	
D1	17.780	17.780	Reference	0.700	0.700	Reference
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	16	16		16	16	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 18-Lead Plastic Dual In-line (300 mil)

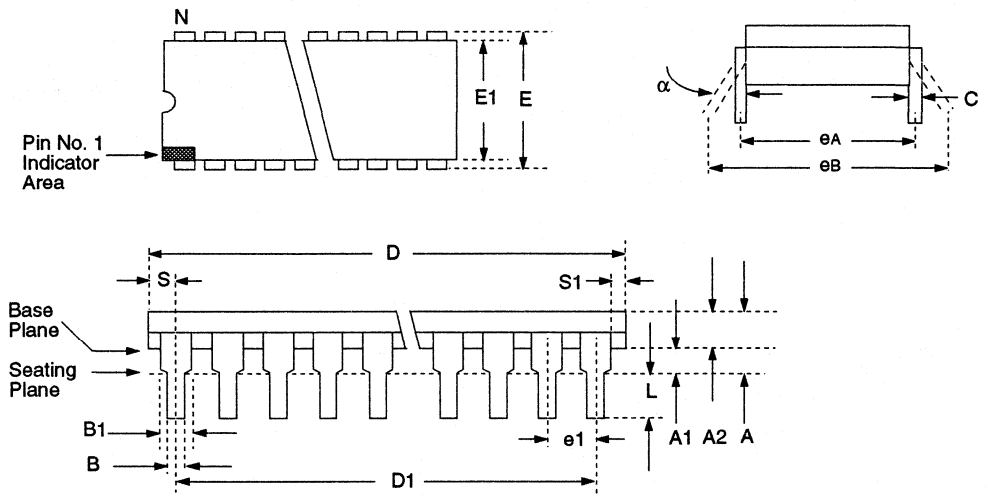


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	Reference	0.060	0.060	Reference
C	0.203	0.381	Typical	0.008	0.015	Typical
D	22.479	23.495		0.885	0.925	
D1	20.320	20.320	Reference	0.800	0.800	Reference
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	18	18		18	18	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 22-Lead Plastic Dual In-line (400 mil)

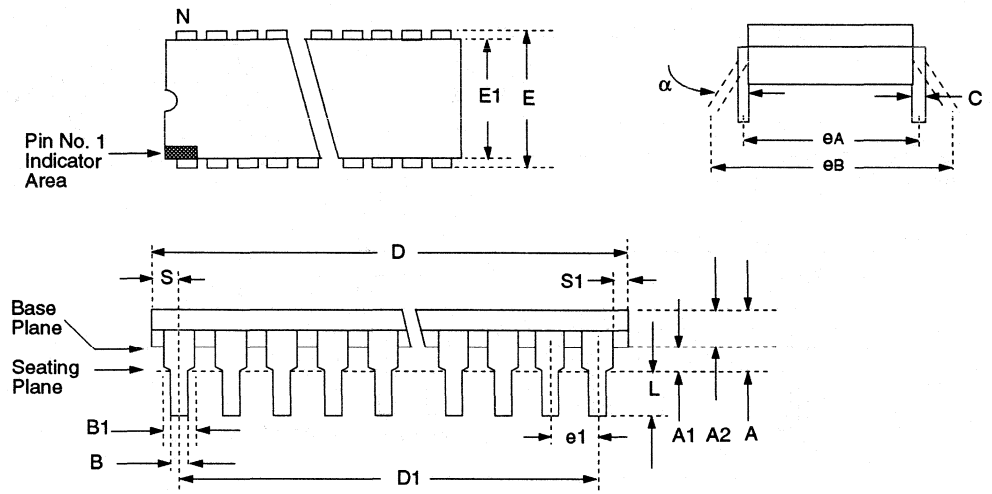


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.572		–	0.180	
A1	0.381	–		0.015	–	
A2	3.175	3.810		0.125	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	Reference	0.060	0.060	Reference
C	0.203	0.381	Typical	0.008	0.015	Typical
D	26.670	28.448		1.050	1.120	
D1	25.400	25.400	Reference	1.000	1.000	Reference
E	9.906	10.795		0.390	0.425	
E1	8.382	9.398		0.330	0.370	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	10.160	10.160	Reference	0.400	0.400	Reference
eB	10.160	12.192		0.400	0.480	
L	3.048	3.556		0.120	0.140	
N	22	22		22	22	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Plastic Dual In-line (600 mil)

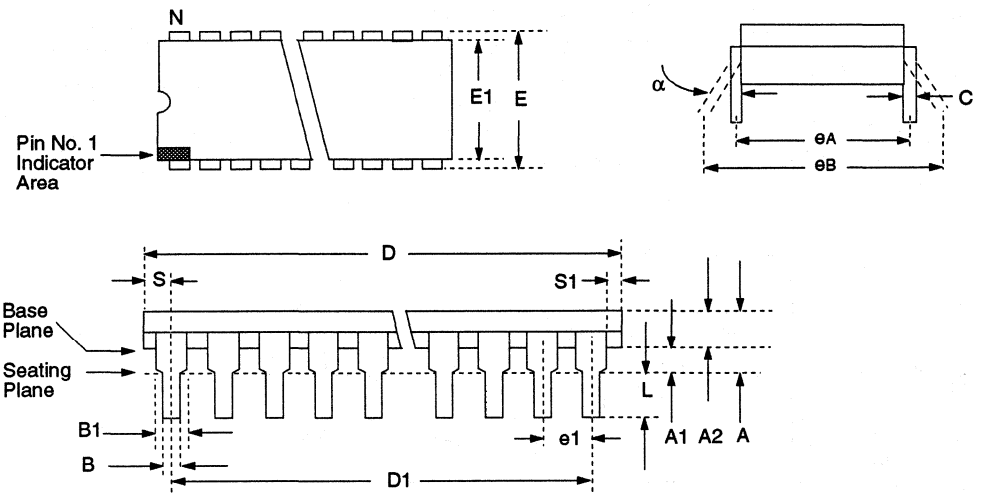


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.508	–		0.020	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	30.353	32.385		1.195	1.275	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	15.240	15.875		0.600	0.625	
E1	12.827	14.224		0.505	0.560	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	15.494	17.272		0.610	0.680	
L	3.048	3.556		0.120	0.140	
N	24	24		24	24	
S	0.889	–		0.035	–	
S1	0.127	–		0.005	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Plastic Dual In-line (300 mil)

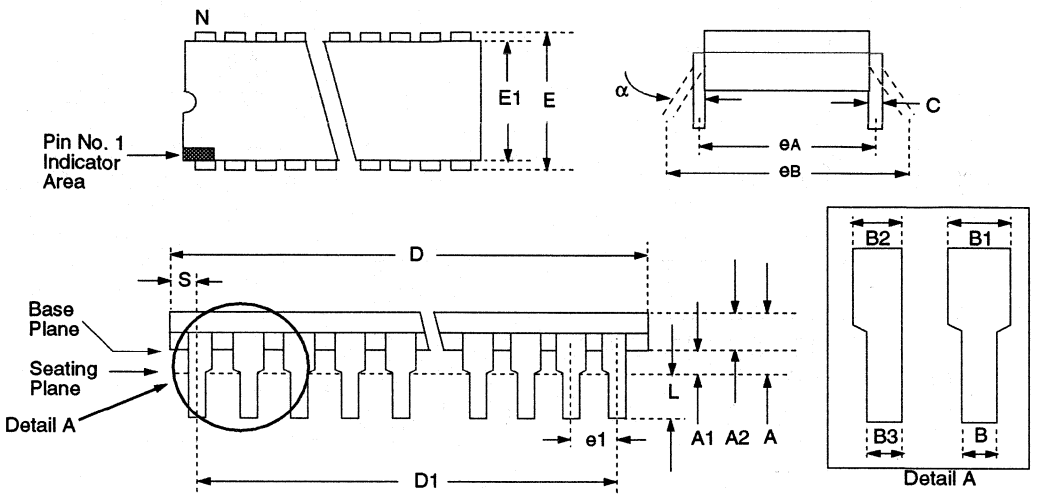


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	4.064		–	0.160	
A1	0.381	–		0.015	–	
A2	3.048	3.810		0.120	0.150	
B	0.355	0.559		0.014	0.022	
B1	1.524	1.524	Typical	0.060	0.060	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	31.242	32.258		1.230	1.270	
D1	27.940	27.940	Reference	1.100	1.100	Reference
E	7.620	8.255		0.300	0.325	
E1	6.096	7.112		0.240	0.280	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	7.620	7.620	Reference	0.300	0.300	Reference
eB	7.874	9.906		0.310	0.390	
L	3.048	3.556		0.120	0.140	
N	24	24		24	24	
S	0.889	–		0.035	–	
S1	0.381	–		0.015	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Dual In-line (300 mil)



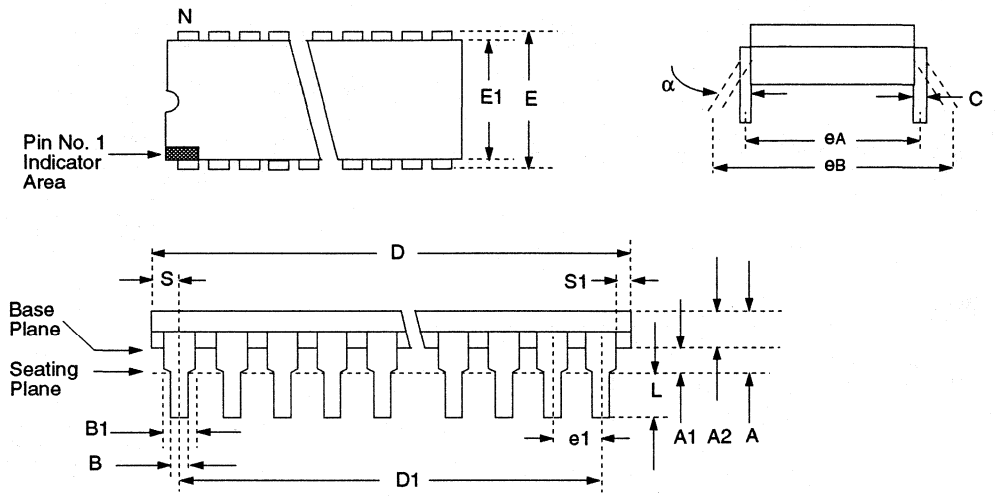
Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	3.632	4.572		0.143	0.180	
A1	0.381	–		0.015	–	
A2	3.175	3.556		0.125	0.140	
B	0.406	0.559		0.016	0.022	
B1	1.016	1.651	Typical	0.040	0.065	Typical
B2	0.762	1.016	4 places	0.030	0.040	4 places
B3	0.203	0.508	4 places	0.008	0.020	4 places
C	0.203	0.331	Typical	0.008	0.013	Typical
D	34.163	35.179		1.385	1.395	
D1	33.020	33.020	Reference	1.300	1.300	Reference
E	7.874	8.382		0.310	0.330	
E1	7.112	7.493		0.280	0.295	
e1	2.540	2.540	Typical	0.100	0.100	Typical
eA	7.874	7.874	Reference	0.310	0.310	Reference
eB	8.128	9.652		0.320	0.380	
L	3.175	3.683		0.125	0.145	
N	28	-		28	-	
S	0.584	1.220		0.023	0.048	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Dual In-line (600 mil)

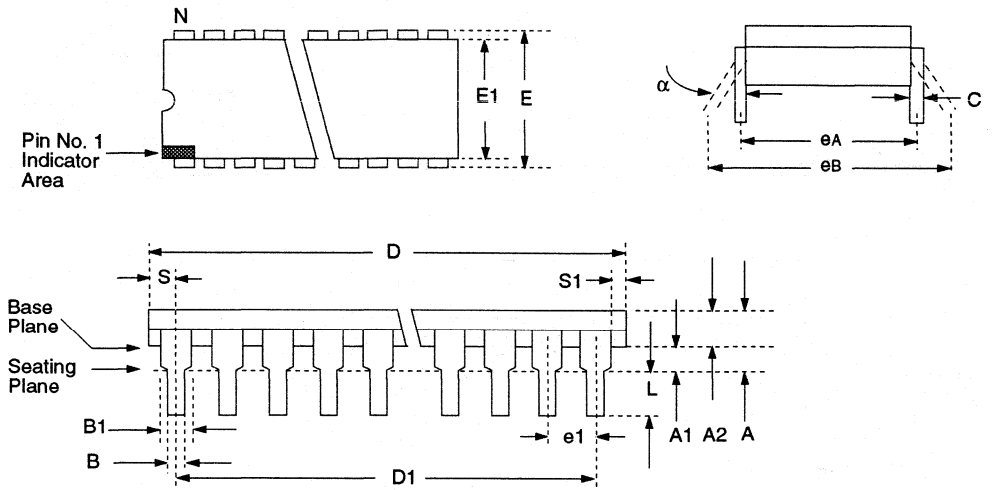


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.508	–		0.020	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	Typical	0.050	0.070	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	35.052	37.084		1.380	1.460	
D1	33.020	33.020	Reference	1.300	1.300	Reference
E	15.240	15.875		0.600	0.625	
E1	12.827	13.970		0.505	0.550	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	28	28		28	28	
S	0.889	–		0.035	–	
S1	0.508	–		0.020	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 40-Lead Plastic Dual In-line (600 mil)

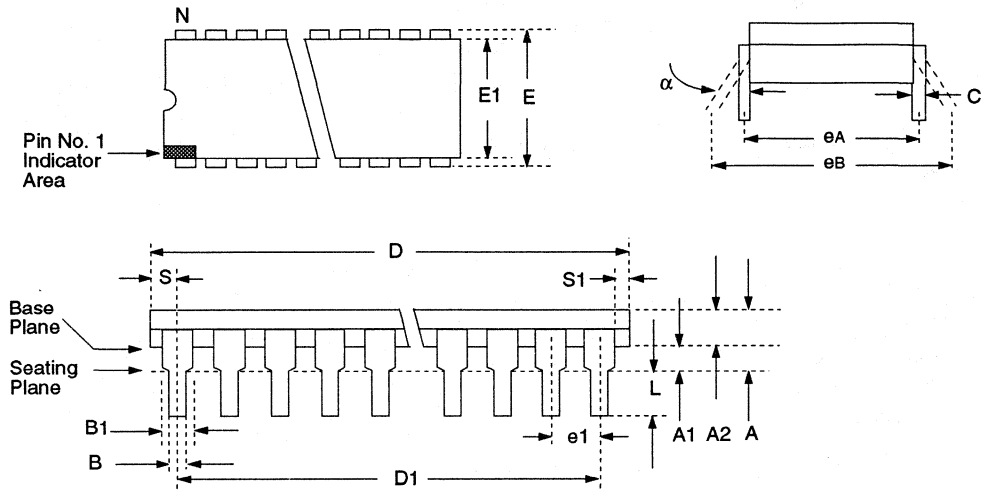


Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.381	–		0.015	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	Typical	0.050	0.070	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	40	40		40	40	
S	1.270	–		0.050	–	
S1	0.508	–		0.020	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 48-Lead Plastic Dual In-line (600 mil)



Package Group: Plastic Dual In-Line (PLA)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	–	5.080		–	0.200	
A1	0.381	–		0.015	–	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.270	Typical	0.050	0.050	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	61.468	62.230		2.420	2.450	
D1	58.420	58.420	Reference	2.300	2.300	Reference
E	15.240	15.875		0.600	0.625	
E1	13.716	14.224		0.540	0.560	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	48	48		48	48	
S	1.270	–		0.050	–	
S1	0.508	–		0.020	–	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Leaded Chip Carrier Family

Symbol List for Plastic Leaded Chip Carrier Package Parameters	
Symbol	**Description of Parameters**
A	Distance between seating plane to highest point of body
A1	Distance between lead shoulder to seating plane
CP	Seating plane coplanarity
D/E	Outside dimension
D1/E1	Plastic body dimension
D2/E2	Footprint
D3/E3	Footprint
LT	Lead thickness
N	Total number of potentially usable lead positions
Nd	Total number of leads on short side (rectangular)
Ne	Total number of leads on long side (rectangular)

Notes:

1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane -H- located at top of mold parting line and coincident with top of lead where lead exits plastic body.
3. Datums D-E and F-G to be determined where center leads exit plastic body at datum plane -H-.
4. To be determined at seating plane -C-.
5. Transition is optional.
6. Plastic body details between leads are optional.
7. Dimension D1 and E1 do not include mold protrusion. Allowable mold protrusion is .254mm/.010in. per side. Dimensions D and E include mold mismatch and are determined at parting line.
8. Square: Details of pin 1 identifier are optional but must be located within one of the two zones indicated. Rectangle: Details of pin 1 identifier are optional but must be located within zone indicated. If the number of terminals on a side is odd, terminal 1 is the center terminal.
9. Location of datums -A- and -B- to be determined at plane -H-.
10. All dimensions and tolerances include lead trim offset and lead finish.
11. These two dimensions determine maximum angle of the lead for certain socket applications. If unit is intended to be socketed, it is advisable to review these dimensions with the socket supplier.
12. Controlling dimension: inches.

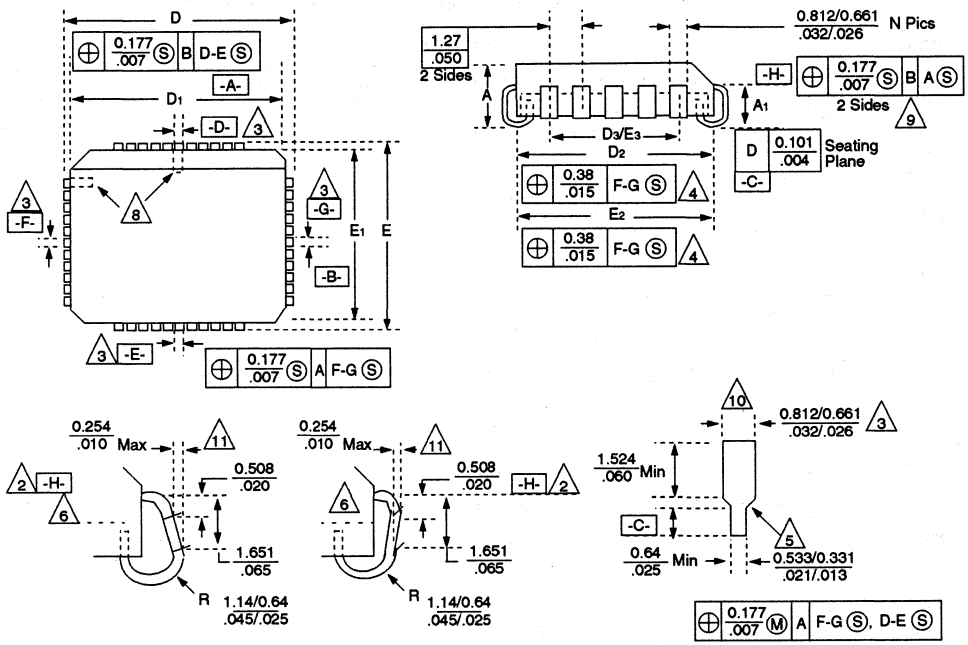
X. Sum of dambar protrusions to be 0.17 (.007) max. per lead.

Y. Feature is not required, but is optional at manufacturer's discretion.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Leaded Chip Carrier (Square)

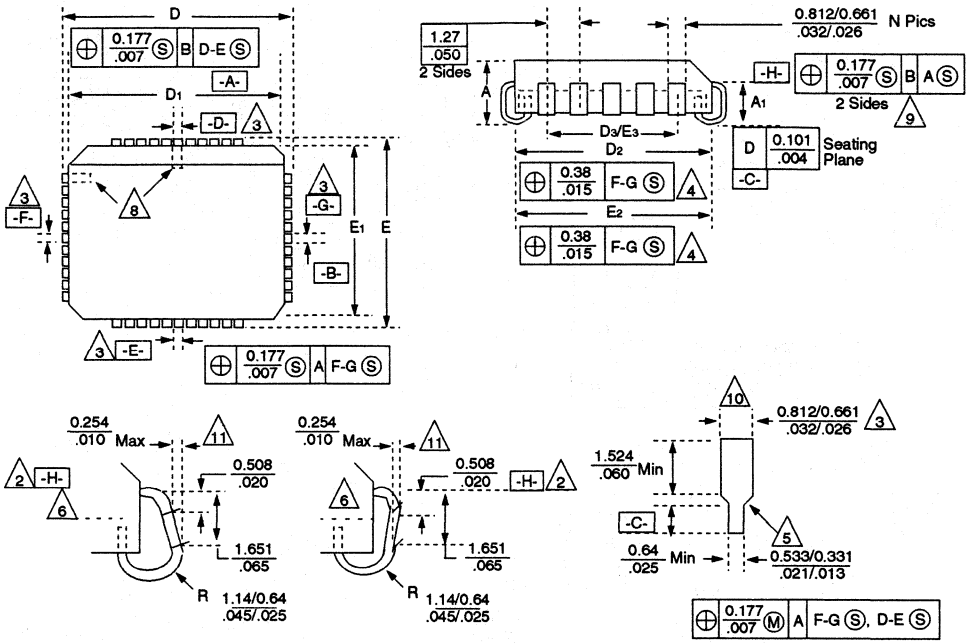


Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.572		0.165	0.180	
A1	2.413	2.921		0.095	0.115	
D	12.319	12.573		0.485	0.495	
D1	11.430	11.583		0.450	0.456	
D2	10.414	10.922		0.410	0.430	
D3	7.620	7.620	Reference	0.300	0.300	Reference
E	12.319	12.573		0.485	0.495	
E1	11.430	11.583		0.450	0.456	
E2	10.414	10.922		0.410	0.430	
E3	7.620	7.620	Reference	0.300	0.300	Reference
N	28	28		28	28	
CP	–	0.102		–	0.004	
LT	0.203	0.381		0.008	0.015	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 32-Lead Plastic Leaded Chip Carrier (Rectangle)



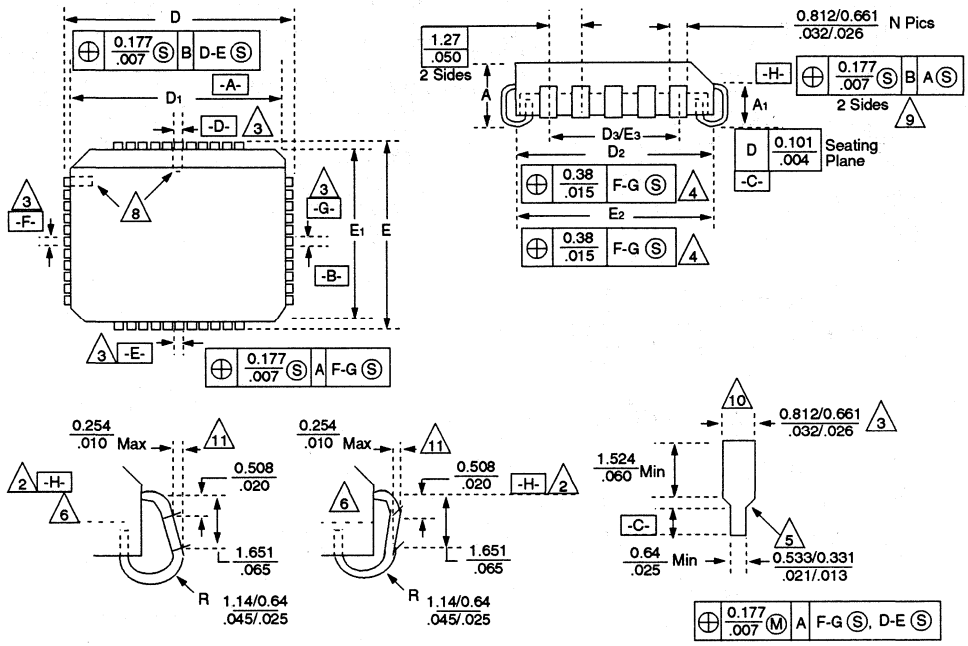
Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	3.048	3.556		0.120	0.140	
A1	1.905	2.413		0.075	0.095	
D	12.319	12.573		0.485	0.495	
D1	11.353	11.507		0.447	0.453	
D2	9.310	10.780		0.380	0.440	
D3	7.620	7.620	Reference	0.300	0.300	Reference
E	14.859	15.113		0.585	0.595	
E1	13.893	14.047		0.547	0.553	
E2	11.760	13.230		0.480	0.540	
E3	10.160	10.160	Reference	0.400	0.400	Reference
N	32	32		32	32	
Nd	7	7		7	7	
Ne	9	9		9	9	
CP	–	0.102		–	0.004	
LT	0.203	0.381		0.008	0.015	

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 44-Lead Plastic Leaded Chip Carrier (Square)

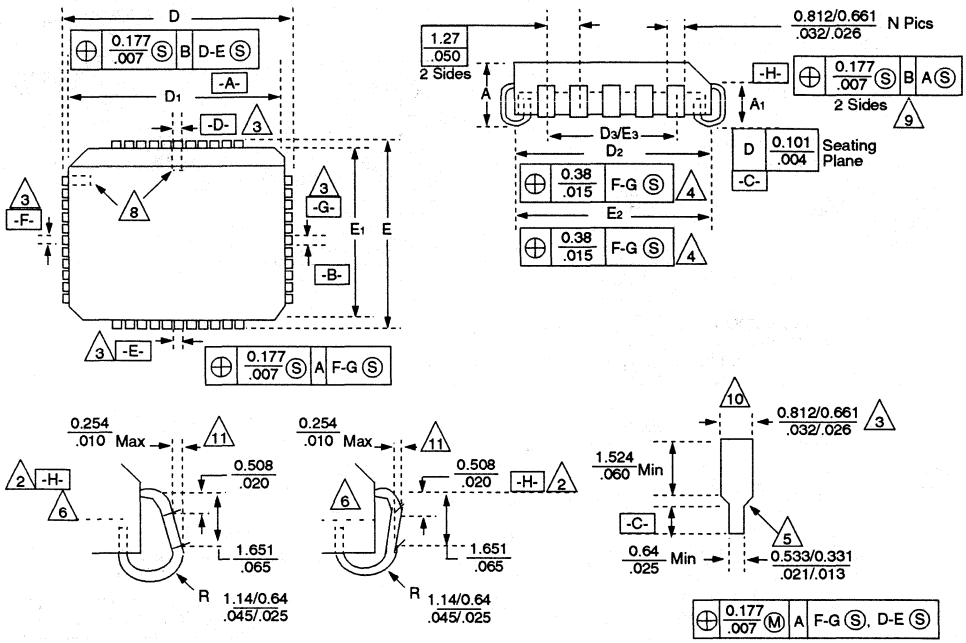


Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.572		0.165	0.180	
A1	2.413	2.921		0.095	0.115	
D	17.399	17.653		0.685	0.695	
D1	16.510	16.663		0.650	0.656	
D2	15.494	16.002		0.610	0.630	
D3	12.700	12.700	Reference	0.500	0.500	Reference
E	17.399	17.653		0.685	0.695	
E1	16.510	16.663		0.650	0.656	
E2	15.494	16.002		0.610	0.630	
E3	12.700	12.700	Reference	0.500	0.500	Reference
N	44	44		44	44	
CP	–	0.102		–	0.004	
LT	0.203	0.381		0.008	0.015	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 68-Lead Plastic Leaded Chip Carrier (Square)

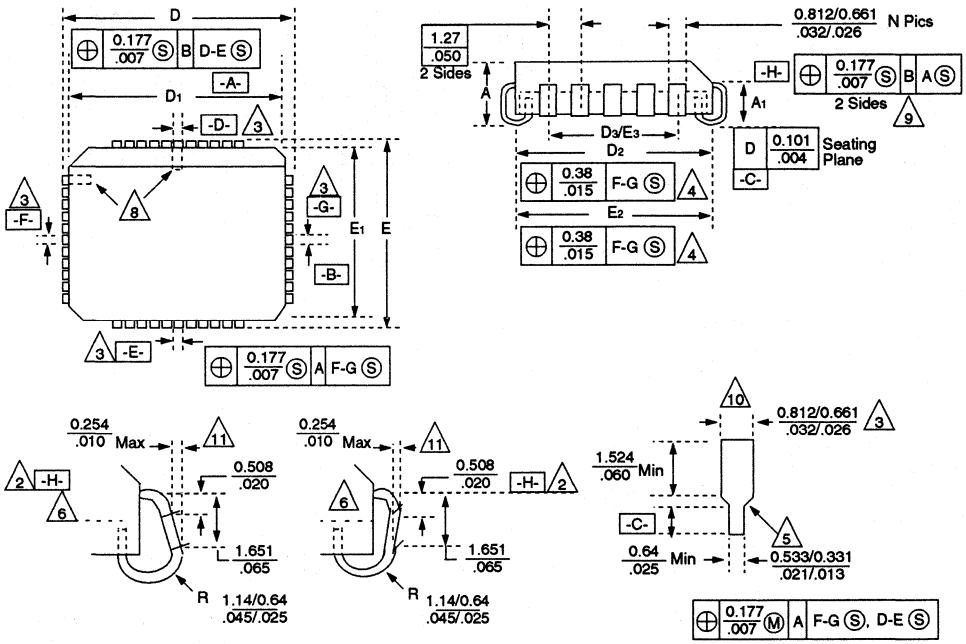


Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.699		0.165	0.185	
A1	2.286	2.794		0.090	0.110	
D	25.019	25.273		0.985	0.995	
D1	24.130	24.334		0.950	0.958	
D2	22.860	23.622		0.900	0.930	
D3	20.320	-	Reference	0.800	-	Reference
E	25.019	25.273		0.985	0.995	
E1	24.130	24.334		0.950	0.958	
E2	22.860	23.622		0.900	0.930	
E3	20.320	-	Reference	0.800	-	Reference
N	68	-		68	-	
CP	-	0.102		-	0.004	
LT	0.203	0.254		0.008	0.010	

8

Packaging Diagrams and Parameters

Package Type: 84-Lead Plastic Leaded Chip Carrier (Square)



Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.699		0.165	0.185	
A1	2.286	2.794		0.090	0.110	
D	30.099	30.353		1.185	1.195	
D1	29.210	29.414		1.150	1.158	
D2	27.940	28.702		1.100	1.130	
D3	25.400	-	Reference	1.000	-	Reference
E	30.099	30.353		1.185	1.195	
E1	29.210	29.414		1.150	1.158	
E2	27.940	28.702		1.100	1.130	
E3	25.400	-	Reference	1.000	-	Reference
N	84	-		84	-	
CP	-	0.102		-	0.004	
LT	0.203	0.254		0.008	0.010	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Small Outline Family

Symbol List for Small Outline Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body
A1	Distance between seating plane and base plane
B	Width of terminals
C	Thickness of terminals
D	Largest overall package parameter of length
E	Largest overall package width parameter not including leads
e	Linear spacing of true minimum lead position center line to center line
H	Largest overall package dimension of width
L	Length of terminal for soldering to a substrate
N	Total number of potentially usable lead positions
CP	Seating plane coplanarity

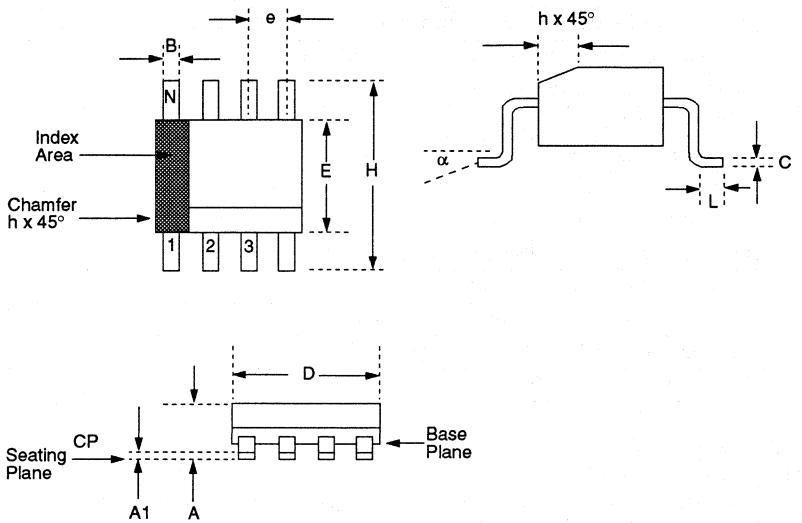
Notes:

1. Controlling parameter: inches.
2. All packages are gull wing lead form.
3. "D" and "E" are reference datums and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .006 package ends and .010 on sides.
4. The chamfer on the body is optional. If it is not present, a visual index feature must be located within the cross-hatched area to indicate pin1 position.
5. Terminal numbers are shown for reference.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 8-Lead Plastic Surface Mount (SOIC - Narrow, 150 mil Body)

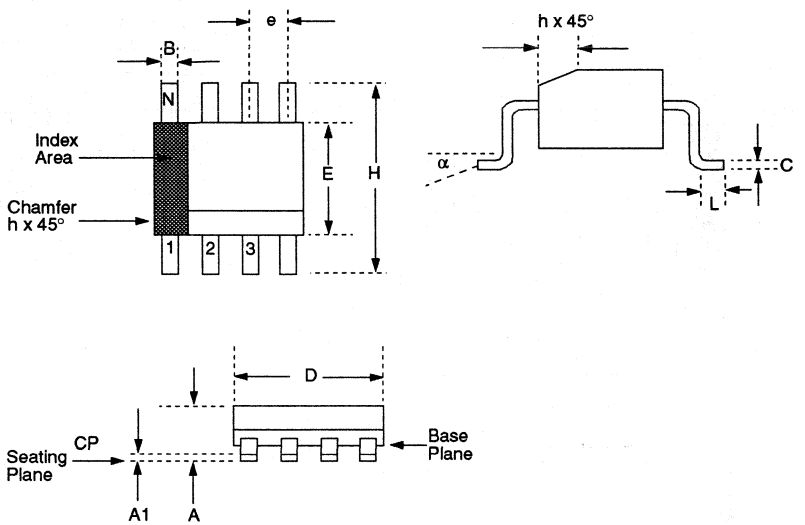


Package Group: Plastic SOIC (SN)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.371	1.728		0.054	0.068	
A1	0.101	0.249		0.004	0.010	
B	0.355	0.483		0.014	0.019	
C	0.190	0.249		0.007	0.010	
D	4.800	4.979		0.189	0.196	
E	3.810	3.988		0.150	0.157	
e	1.270	1.270	Typical	0.050	0.050	Typical
H	5.816	6.198		0.229	0.244	
h	0.381	0.762		0.015	0.030	
L	0.508	1.016		0.020	0.040	
N	8	8		8	8	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 8-Lead Plastic Surface Mount (SOIC - Medium, 200 mil Body)

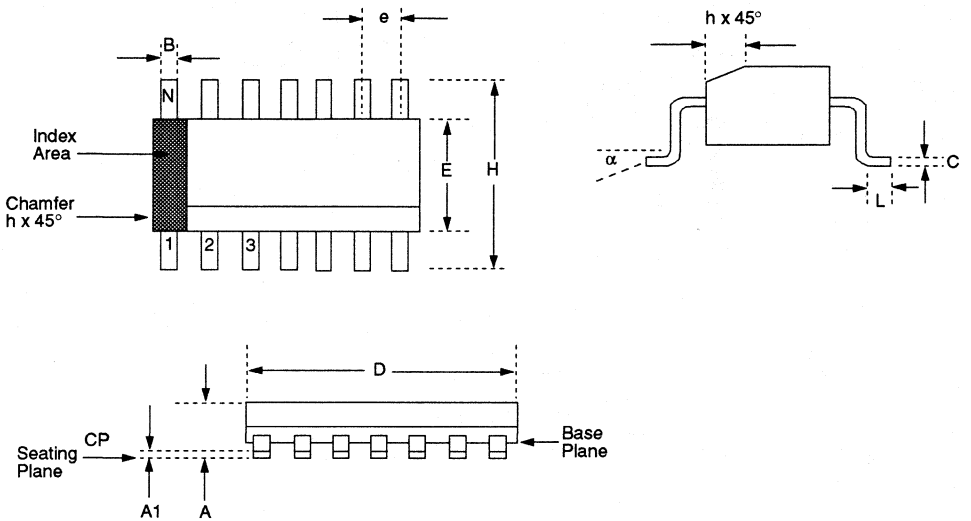


Package Group: Plastic SOIC (SM)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.778	2.032		0.070	0.080	
A1	0.101	0.249		0.004	0.010	
B	0.355	0.483		0.014	0.019	
C	0.190	0.249		0.007	0.010	
D	5.080	5.334		0.200	0.210	
E	5.156	5.411		0.203	0.213	
e	1.270	1.270	Reference	0.050	0.050	Reference
H*	7.670	8.103		0.302	0.319	
h	0.381	0.762		0.015	0.030	
L	0.508	1.016		0.020	0.040	
N	14	14		14	14	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 14-Lead Plastic Surface Mount (SOIC - Narrow, 150 mil Body)

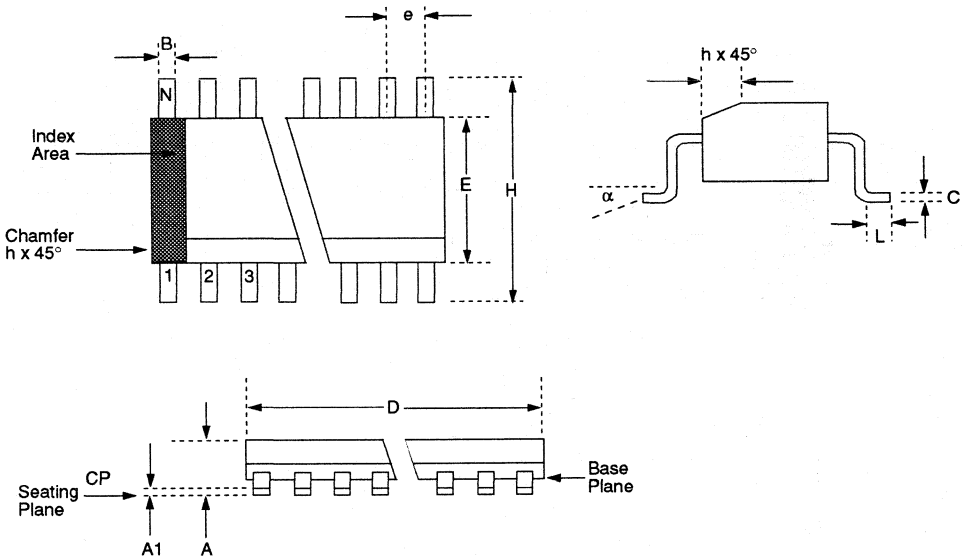


Package Group: Plastic SOIC (SL)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.371	1.728		0.054	0.068	
A1	0.101	0.249		0.004	0.010	
B	0.355	0.483		0.014	0.019	
C	0.190	0.249		0.008	0.010	
D	8.559	9.983		0.337	0.393	
E	3.810	3.988		0.150	0.157	
e	1.270	1.270	Reference	0.050	0.050	Reference
H	5.816	6.198		0.229	0.244	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	14	14		16	16	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 18-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body)

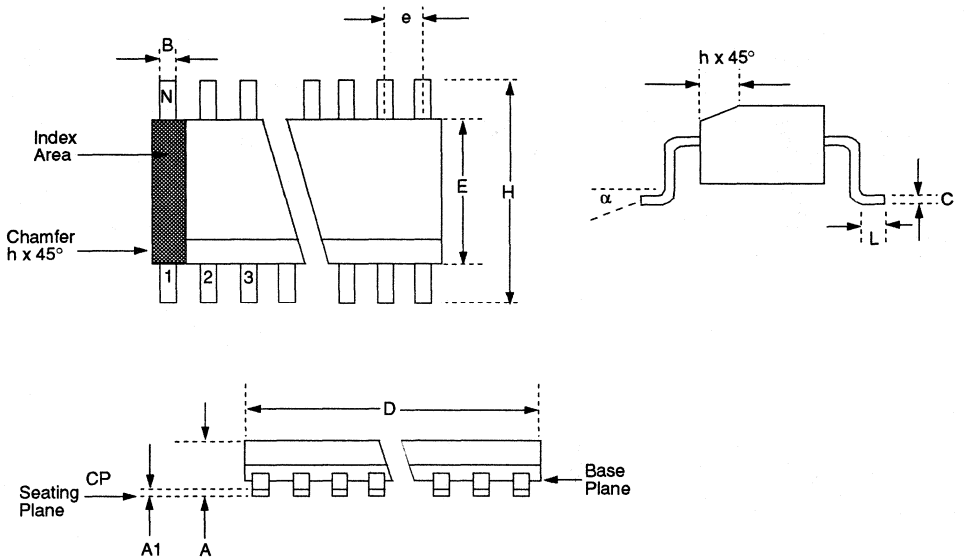


Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	11.353	11.735		0.447	0.462	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	Reference	0.050	0.050	Reference
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	18	18		18	18	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 24-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body)

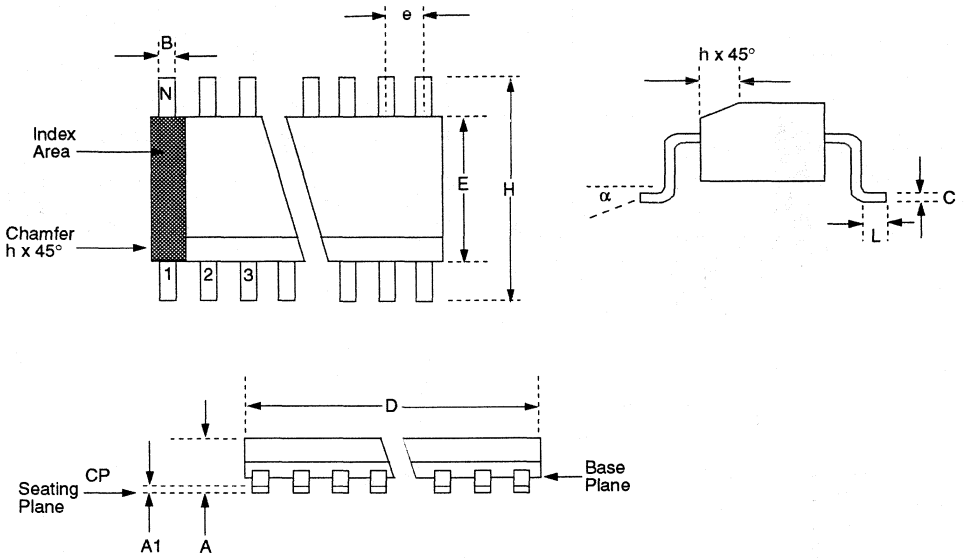


Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	15.214	15.596		0.599	0.614	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	Reference	0.050	0.050	Reference
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	24	24		24	24	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body)

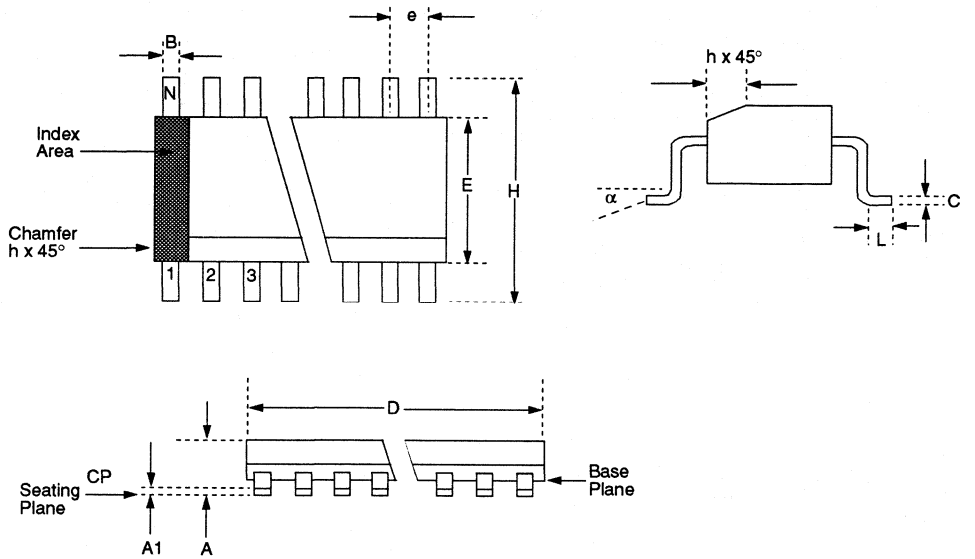


Package Group: Plastic SOIC (SO)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	2.362	2.642		0.093	0.104	
A1	0.101	0.300		0.004	0.012	
B	0.355	0.483		0.014	0.019	
C	0.241	0.318		0.009	0.013	
D	17.703	18.085		0.697	0.712	
E	7.416	7.595		0.292	0.299	
e	1.270	1.270	Typical	0.050	0.050	Typical
H	10.007	10.643		0.394	0.419	
h	0.381	0.762		0.015	0.030	
L	0.406	1.143		0.016	0.045	
N	28	28		28	28	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Surface Mount (SOIC - Wide, 330 mil Body)



Package Group: Plastic SOIC (SW)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0	8°		0	8°	
A	2.286	2.642		0.090	0.104	
A1	0.101	0.280		0.004	0.011	
B	0.355	0.508		0.014	0.020	
C	0.228	0.305		0.009	0.012	
D	17.780	18.085		0.700	0.712	
E	8.636	8.890		0.340	0.350	
e	1.270	1.270	Reference	0.050	0.050	Reference
H	11.760	12.116		0.463	0.477	
h	0.254	0.737		0.010	0.029	
L	0.508	1.067		0.020	0.042	
N	28	28		28	28	
CP	–	0.102		–	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Shrink Small Outline Family

Symbol List for Shrink Small Outline Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body
A1	Distance between seating plane and base plane
B	Width of terminals
C	Thickness of terminals
D	Largest overall package parameter of length
E	Largest overall package width parameter not including leads
e	Linear spacing of true minimum lead position center line to center line
H	Largest overall package dimension of width
L	Length of terminal for soldering to a substrate
N	Total number of potentially usable lead positions
CP	Seating plane coplanarity

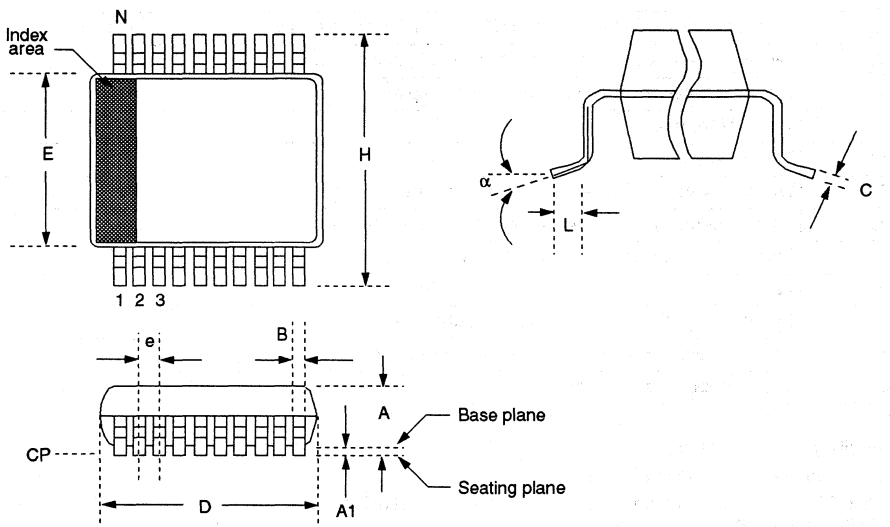
Notes:

1. Controlling parameter: mm.
2. All packages are gull wing lead form.
3. "D" and "E" are reference datums and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .015mm .006 package ends and .010" on sides.
4. A .25mm visual index feature must be located within the shaded area to indicate pin 1 position.
5. Terminal numbers are shown for reference.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 20-Lead Plastic Surface Mount (SSOP - 209 mil Body 5.30 mm)

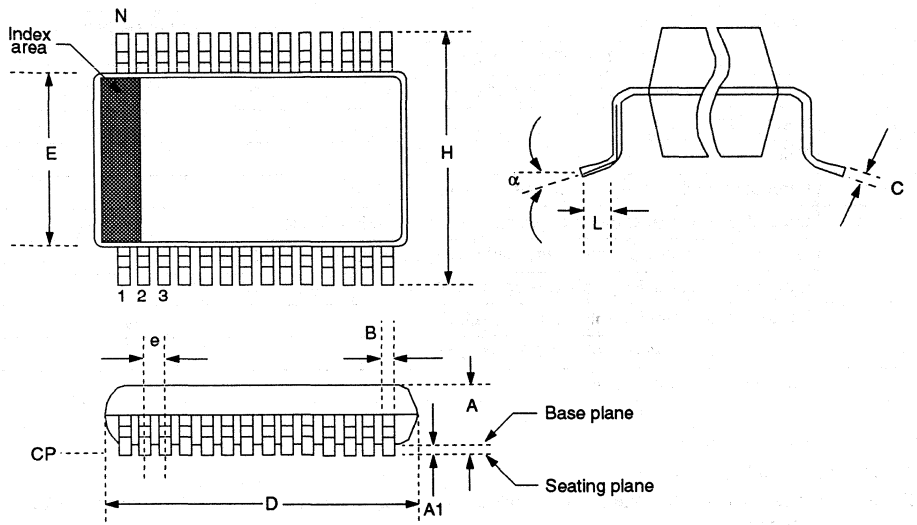


Package Group: Plastic SSOP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.730	1.990		0.068	0.078	
A1	0.050	0.210		0.002	0.008	
B	0.250	0.380		0.010	0.015	
C	0.130	0.220		0.005	0.009	
D	7.070	7.330		0.278	0.289	
E	5.200	5.380		0.205	0.212	
e	0.650	0.650	Reference	0.026	0.026	Reference
H	7.650	7.900		0.301	0.311	
L	0.550	0.950		0.022	0.037	
N	20	20		20	20	
CP	-	0.102		-	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Surface Mount (SSOP - 209 mil Body 5.30 mm)



Package Group: Plastic SSOP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	8°		0°	8°	
A	1.730	1.990		0.068	0.078	
A1	0.050	0.210		0.002	0.008	
B	0.250	0.380		0.010	0.015	
C	0.130	0.220		0.005	0.009	
D	10.070	10.330		0.396	0.407	
E	5.200	5.380		0.205	0.212	
e	0.650	0.650	Reference	0.026	0.026	Reference
H	7.650	7.900		0.301	0.311	
L	0.550	0.950		0.022	0.037	
N	28	28		28	28	
CP	-	0.102		-	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Thin Small Outline and Very Small Outline Families (TSOP, VSOP)

Symbol List for Thin Small Outline Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body
A1	Distance between seating plane and base plane
B	Width of terminals
C	Thickness of terminals
D	Largest overall package parameter of length
E	Largest overall package width parameter not including leads
e	Linear spacing of true minimum lead position center line to center line
H	Largest overall package dimension of width
L	Length of terminal for soldering to a substrate
N	Total number of potentially usable lead positions
CP	Seating plane coplanarity

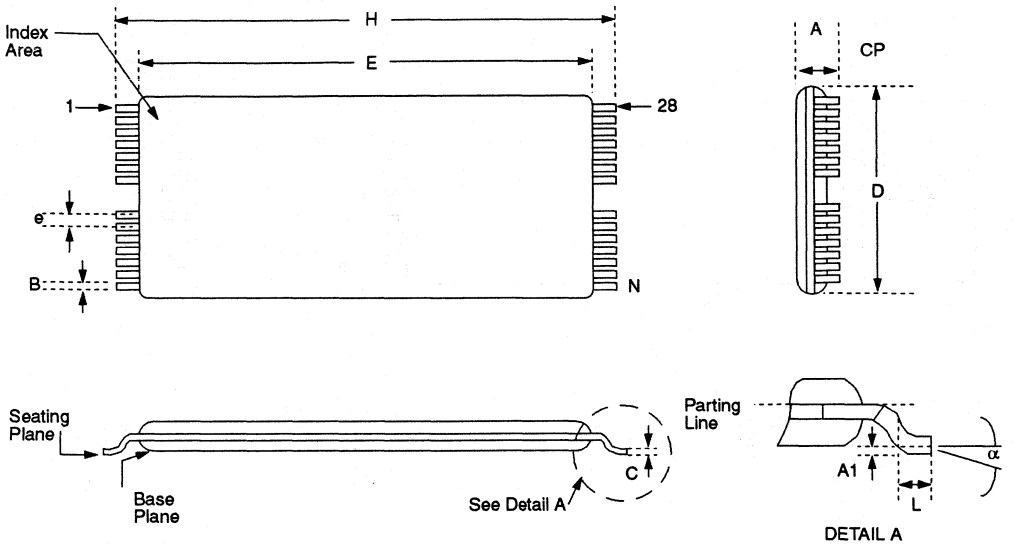
Notes:

1. Controlling parameter: inches.
2. All packages are gull wing lead form.
3. "D" and "E" are reference datums and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005 per side.
4. A visual index feature must be located within the crosshatched area to indicate pin 1 position.
5. Terminal numbers are shown for reference.

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Surface Mount (TSOP 8 x 20 mm)

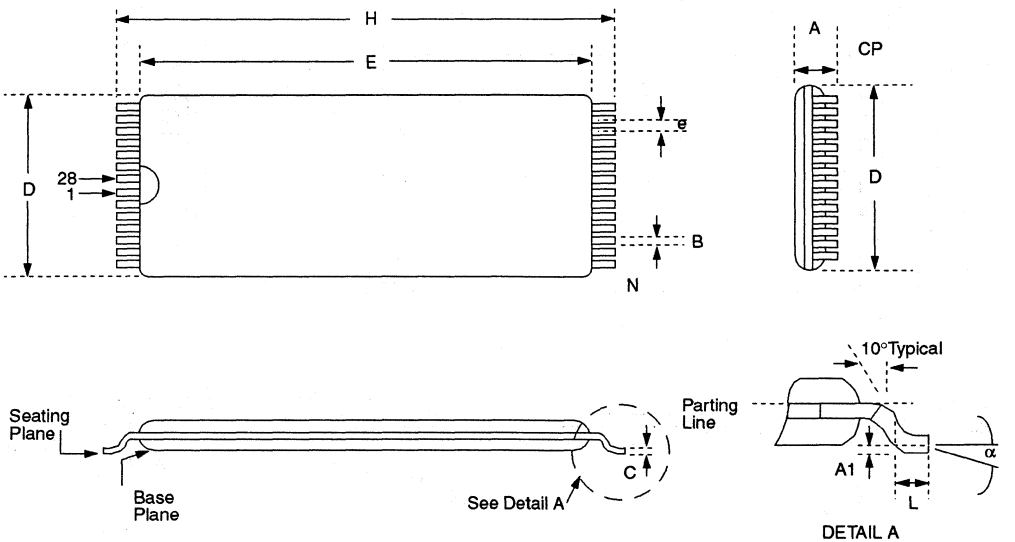


Package Group: Plastic TSOP (TS)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0	8°		0	8°	
A	-	1.190		-	0.047	
A1	0.050	0.150		0.002	0.006	
B	0.150	0.250		0.006	0.010	
C	0.100	0.200		0.004	0.008	
D	7.800	8.200		0.307	0.323	
E	18.290	18.490		0.720	0.728	
e	0.510	-	Reference	0.020	-	Reference
H	19.810	20.190		0.780	0.795	
L	0.410	0.610		0.016	0.024	
N	28	28		28	28	
CP	-	0.102		-	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 28-Lead Plastic Surface Mount (VSOP 8 x 13.4 mm))



Package Group: Plastic VSOP (VS)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0	8°		0	8°	
A	1.000	1.250		0.039	0.049	
A1	0.000	0.200		0.000	0.008	
B	0.150	0.300		0.006	0.012	
C	0.130	0.220		0.005	0.009	
D	7.900	8.100		0.311	0.319	
E	11.700	11.900		0.460	0.468	
e	0.550	-	Reference	0.022	-	Reference
H	13.100	13.700		0.516	0.539	
N	28	28		28	28	
L	0.300	0.700		0.012	0.027	
CP	-	0.102		-	0.004	

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Plastic Metric Quad Flatpack Family (MQFP)

Symbol List for Metric Plastic Quad Flatpack Package Parameters	
Symbol	**Description of Parameters**
α	Angular spacing between min. and max. lead positions measured at the gauge plane
A	Distance between seating plane to highest point of body
A1	Distance between seating plane and base plane
A2	Distance from base plane to highest point of body
b	Width of terminals
C	Thickness of terminals
D1/E1	Largest overall package parameter including leads
D/E	Largest overall package parameter including leads
D3/E3	Center of end lead to center of end lead
e	Linear spacing of true minimum lead position center line to center line
L	Length of terminal for soldering to a substrate
N	Total number of potentially usable lead positions
CP	Seating plane coplanarity

Notes:

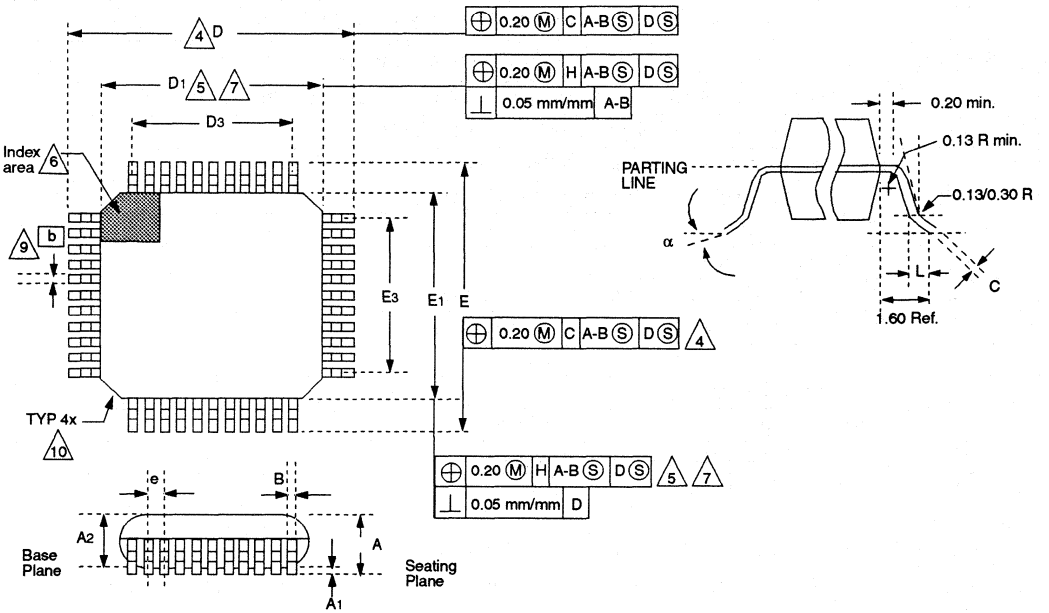
1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane -H- is located at bottom of mold parting line and coincident with bottom of lead, where lead exits body.
3. Datums A-B and -D- to be determined at datum plane -H-.
4. To be determined at seating plane -C-.
5. Dimension D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25mm per side. Dimensions D1 and E1 do not include mold mismatch and are determined at datum plane -H-.
6. Details of pin 1 identifier are optional but must be located within the zone indicated.
7. These dimensions to be determined at datum plane -H-.
8. All dimensions are in millimeters.
9. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius or the lead foot.
10. Exact shape of this feature is optional.
11. N is the number of leads.
12. Controlling parameter: millimeters.
13. All packages are gull wing lead form.

8

© 1995 Microchip Technology Inc.

Packaging Diagrams and Parameters

Package Type: 44-Lead Plastic Surface Mount (MQFP 10x10 mm Body 1.6/0.15 mm Lead Form))



Package Group: Plastic MQFP						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	7°		0°	7°	
A	2.000	2.350		0.078	0.093	
A1	0.050	0.250		0.002	0.010	
A2	1.950	2.100		0.768	0.083	
b	0.300	0.450	Typical	0.011	0.018	Typical
C	0.150	0.180		0.006	0.007	
D	12.950	13.450		0.510	0.530	
D1	9.900	10.100		0.390	0.398	
D3	8.000	8.000	Reference	0.315	0.315	Reference
E	12.950	13.450		0.510	0.530	
E1	9.900	10.100		0.390	0.398	
E3	8.000	8.000	Reference	0.315	0.315	Reference
e	0.800	0.800		0.031	0.032	
L	0.730	1.030		0.028	0.041	
N	44	44		44	44	
CP	0.102	–		0.004	–	

© 1995 Microchip Technology Inc.

MICROCHIP

Devices in Die/Wafer Form - Non-Volatile Memory

INTRODUCTION

Microchip Technology Inc.'s non-volatile memory devices are available in wafer form and in die form. All products sold as die or wafers have been characterized and qualified according to the requirements of Microchip Technology Inc. Specifications SPI-41014, "Characterization and Qualification of Integrated Circuits", and QCI-39000, "Worldwide Quality Conformance Requirements".

PRODUCT INTEGRITY

Product supplied in die or wafer form will be 100 percent visually inspected to the criteria defined in Microchip Technology Inc. Specification, QCI-30014, "Standard Visual Inspection Procedure of Dice Prior to Assembly for Commercial Products"

Die/Wafer thickness is 18 mils or 21 mils depending on product.

Wafers at reduced thicknesses are also available.

CAUTION

Some EEPROM products use EEPROM cells for device configuration. Exposure to ultra-violet light or x-rays must be avoided. Exposure to ultra-violet light or x-rays may cause the device to operate improperly.

These products are susceptible to damage from electro-static discharge. Extreme care is urged in the handling and assembly of these products.

BONDABILITY OF DIE TO SUBSTRATE

Dice are capable of bonding either by using a gold eutectic bond to a gold plated pedestal containing 60 micro-inches of gold, or by using an electrically conductive adhesive (e.g. epoxy) to any substrate.

BONDABILITY OF WIRES TO DIE

Dice shall be capable of thermosonic gold or ultrasonic wire bonding such that the minimum conditions of MIL-STD 883, Method 2011 on "Bond Strength (Destructive Bond pull Test)" are met.

PAD METALLIZATION

Pad metallization is silicon doped aluminum.

BACK SIDE PREPARATION

Die and wafer back sides are backlapped and are without any gold coating.

ELECTRICAL

Dice are guaranteed to fully meet data sheet specifications at the commercial temperature range of 0°C to 70°C.

The die back side is grounded through contacts internal to the part. Thus, it is permitted to float the die mounting surface. It is recommended that the die mounting surface be grounded in multi-chip assemblies.

PACKAGING

Die shipped by Microchip Technology Inc. are placed in a "waffle pack" with sufficient cavity area to restrain the die while maintaining their orientation. Lint free paper inserts are placed over the waffle packs and each pack is secured with a plastic locking clip. Groups of waffle packs are assembled into sets for shipment. A label with lot number, quantity, part number and packing date is placed on each waffle pack.

Wafers shipped by Microchip Technology Inc. are separated by lint free paper and dry packed in a shipping container of appropriate size which is labeled with lot number, quantity, part number and packing date.

8

© 1994 Microchip Technology Inc.

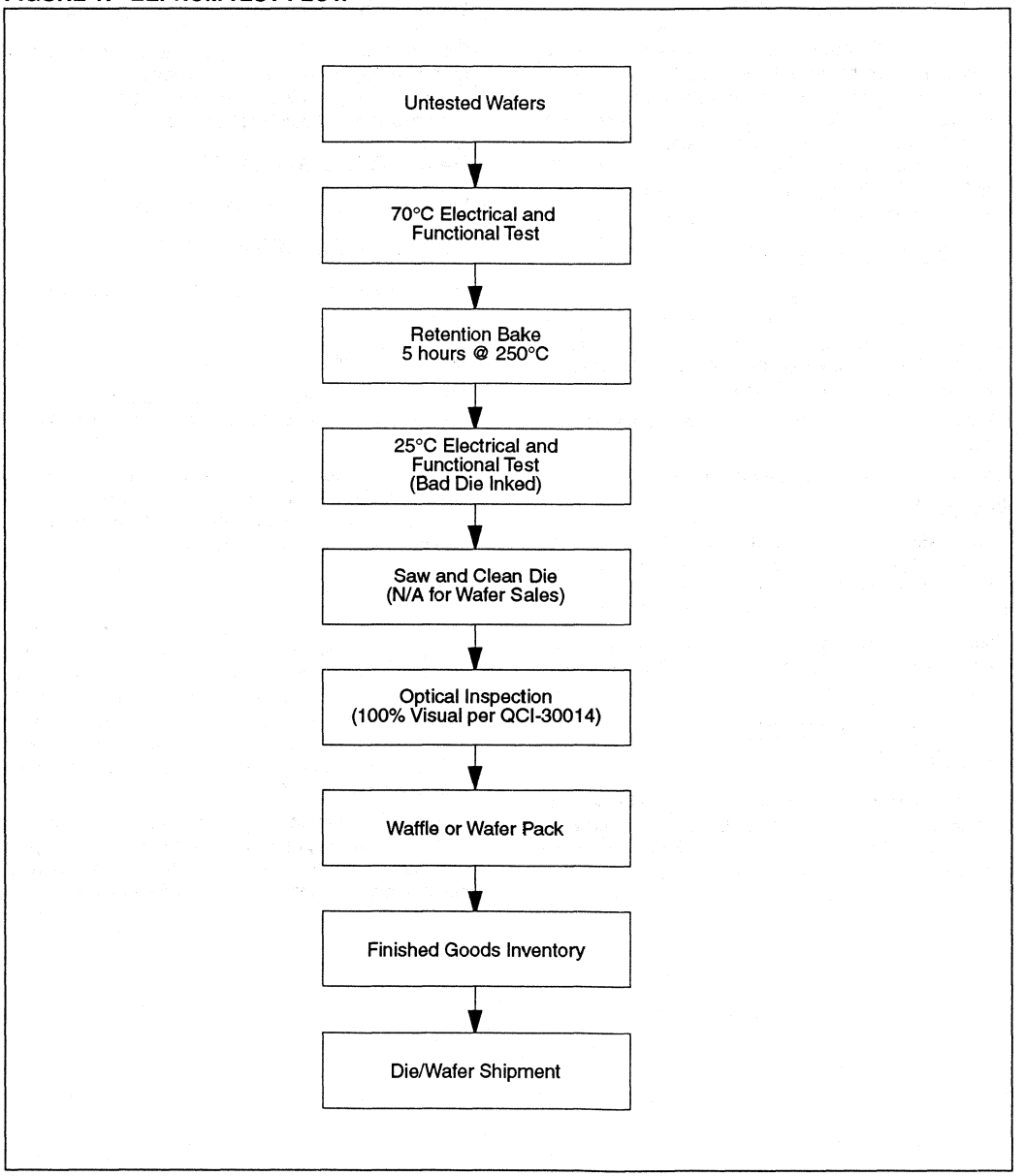
DIE/WAFER PRODUCT OFFERING

All Microchip Technology Inc.'s serial EEPROMs, parallel EEPROMs and EPROMS are available in die or wafer form in the commercial temperature range of 0°C to 70°C. Part number suffixes of /s and /w are used to designate devices in die and wafer form, respectively.

TEST FLOW

All Microchip Technology Inc.'s die products are subjected to functional and parametric testing at the wafer level. The typical EEPROM test flow is shown in Figure 1.

FIGURE 1: EEPROM TEST FLOW



© 1994 Microchip Technology Inc.

SECTION 9
OFFICE LOCATIONS

© 1995 Microchip Technology Inc.

© 1995 Microchip Technology Inc.

MICROCHIP

Factory Representatives

AFRICA

AMDS
P.O. Box 58208
Newville 2114
Republic of South Africa
Tel: 27/11-6732141
Fax: 27/11-6736024

ASIA/PACIFIC

Hong Kong

Microchip Technology Inc.
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200
Fax: 852 2 401 3431

Japan

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166
Fax: 81 45 471 6122

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200
Fax: 82 2 558 5934

Singapore

Microchip Technology Inc.
200 Middle Road
#10-03 Prime Centre
Singapore 0718
Tel (mobile): 65 634 2305

Taiwan

Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175
Fax: 886 2 545 0139

CANADA

Alberta

Enerlec Sales Ltd.
#103 155 Glendeer Circle S.E.
Calgary, Alberta T2H 2S8 Canada
Tel: 403-777-1550
Fax: 403-777-1553

British Columbia

Enerlec Sales Ltd.
#7 3671 Viking Way
Richmond, B.C. V6V 1W1 Canada
Tel: 604-273-0882
Fax: 604-273-0884

Manitoba

Enerlec Sales Ltd.
#7 3671 Viking Way
Richmond, B.C. V6V 1W1 Canada
Tel: 604-273-0882
Fax: 604-273-0884

New Brunswick

Dynasty Components, Inc.
1140 Morrison Drive, Unit 110
Ottawa, Ontario K2H 8S9 Canada
Tel: 613-596-9800
Fax: 613-596-9886

Newfoundland

Dynasty Components, Inc.
1140 Morrison Drive, Unit 110
Ottawa, Ontario K2H 8S9 Canada
Tel: 613-596-9800
Fax: 613-596-9886

Ontario

Dynasty Components, Inc.
1140 Morrison Drive, Unit 110
Ottawa, Ontario K2H 8S9 Canada
Tel: 613-596-9800
Fax: 613-596-9886

Prince Edward Island

Dynasty Components, Inc.
1140 Morrison Drive, Unit 110
Ottawa, Ontario K2H 8S9 Canada
Tel: 613-596-9800
Fax: 613-596-9886

Quebec

Dynasty Components, Inc.
1140 Morrison Drive, Unit 110
Ottawa, Ontario K2H 8S9 Canada
Tel: 613-596-9800
Fax: 613-596-9886

Saskatchewan

Enerlec Sales Ltd.
#7 3671 Viking Way
Richmond, B.C. V6V 1W1 Canada
Tel: 604-273-0882
Fax: 604-273-0884

© 1995 Microchip Technology Inc.

Factory Representatives

EUROPE

Finland

Memec Finland
Asemankello
Vernissankatu 6
Fin 01300 Vantaa
Finland
Tel: 358-07001-9830
Fax: 358-07001-9839

Germany

Active Rep GmbH
Beningastr. 24
D-26721 Emden
Germany
Tel: 49/4921-979071
Fax: 49/4921-979072

Active Rep GmbH
Blumental 1-3
D-42653 Solingen
Germany
Tel: 49/212-25823-0
Fax: 49/212--282323

Active Rep GmbH
Kennedy-StraBe 5
D-75438 Knittlingen
Germany
Tel: 49/7043-33276
Fax: 49/7043-33492

Active Rep GmbH
Reichenaustr. 18
D-81234 Muenchen
Germany
Tel: 49/89-89689181
Fax: 49/89-89689183

Ireland

Eltech Agencies, Ltd.
27 Maccurtain Street
Cork Ireland
Tel: 353-21-509366
Fax: 353-21-509344

Israel

Elina Electronics Ltd.
14 Raoul Wallenberg St.
P.O. Box 13190
Tel Aviv 61131 Israel
Tel: 972 3 49 85 43
Fax: 972 3 49 87 45

Netherlands

Sonetech
P.O. Box 258
5670 AG Nuenen
Netherlands
Tel: 31-40-837075
Fax: 31-40-832300

Scotland

Juniper Solutions
Unit NR8, Newhouse Business Park
NewHouse Road
Grangemouth FK3 8LL
Scotland
Tel: 44-1324-666990
Fax: 44-1324-666980

Sweden

Memec Scandinavia AB
Kvarnholmsvagen 52
131 31 Nacka
Sweden
Tel: 46-8643-4190
Fax: 46-8643-1195

Turkey

Inter Muehendislik Danismanlik
Ve Ticaret A.S. 1
Hasircbasi Caddesi No. 55
81310 Kadikoy
Istanbul
Turkey
Tel: 90 216 349 94 00
Fax: 90 216 349 94 31

MEXICO

CompTech de Mexico
Av. Morelos Sur 882
Col. Centro
Cuernavaca, Morelos CP62000
Tel: 52 73122733
Fax: 52 73126368

© 1995 Microchip Technology Inc.

SOUTH AMERICA

Argentina

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Bolivia

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Brazil

Aplicacoes Eletronicas
Artimar Ltda.
Rua Marques de Itu N°. 70-10And.
CEP 01223
Sao Paulo, Brazil
Tel: 55-11-231-0277
Fax: 55-11-255-0511

Caribbean

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Chile

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Columbia

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Ecuador

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Paraguay

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Peru

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Uruguay

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Venezuela

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

© 1995 Microchip Technology Inc.

Factory Representatives

UNITED STATES

Alaska

Trinity Technologies
1261 Oakmead Parkway
Sunnyvale, CA 94086
Tel: 408-733-9000
Fax: 408-733-9970

Alabama

Concord Component Reps Inc.
190 Lime Quarry Road, Suite 102
Madison, AL 35758
Tel: 205-772-8883
Fax: 205-772-8262

Arizona

Western High Tech Marketing, Inc.
9414 E. San Salvador, Suite 206
Scottsdale, AZ 85258
Tel: 602-860-2702
Fax: 602-860-2712

Arkansas

CompTech Sales, Inc.
2401 Gateway Drive
Suite 114
Irving, TX 75063
Tel: 214-751-1181
Fax: 214-550-8113

California

Costa Mesa

Competitive Technology, Inc.
200 Baker Street, Suite 101
Costa Mesa, CA 92626
Tel: 714-540-5501
Fax: 714-540-5171

Escondido

Eagle Technical Sales Assoc. Inc.
1900 Sunset Drive, Suite A
Escondido, CA 92025
Tel: 619-743-6550
TeL: 619-743-6585

Sunnyvale

Trinity Technologies
1261 Oakmead Parkway
Sunnyvale, CA 94086
Tel: 408-733-9000
Fax: 408-733-9970

Colorado

Western Region Marketing
9176 Marshall Place
Westminster, CO 80030
Tel: 303-428-8088
Fax: 303-426-8585

Connecticut

Gilford

S-J Associates, Inc.
10 Copper Ridge Circle
Gilford, CT 06437
Tel: 203-458-7558
Fax: 203-458-1181

Naugatuck

S-J Associates, Inc.
15 Coventry Lane
Naugatuck, CT 06770
Tel: 203-723-4707
Fax: 202-723-1629

Delaware

S-J Mid-Atlantic, Inc.
131-D Gaither Drive
Mt. Laurel, NJ 08054
Tel: 609-866-1234
Fax: 609-866-8627

District of Columbia

S-J Chesapeake
900 S. Washington Street
Suite 307
Falls Church, VA 22046
Tel: 703-533-2233
Fax: 703-533-2236

Florida

Altamonte Springs

Electramark Florida, Inc.
401 Whooping Loop, Suite 1565
Altamonte Springs, FL 32701
Tel: 407-830-0844
Fax: 407-830-0847

Boca Raton

Electramark Florida, Inc.
621 NW 53rd Street, Suite 240
Boca Raton, FL 33487
Tel: 407-998-8820
Fax: 407-998-8821

Tampa

Electramark Florida, Inc.
2910 W. Waters Ave.
Tampa, FL 33614
Tel: 813-915-1177
Fax: 813-915-1188

Georgia

Concord Component Reps Inc.
6825 Jimmy Carter Boulevard
Norcross, GA 30071
Tel: 404-416-9597
Fax: 404-441-0790

Hawaii

Trinity Technologies
1261 Oakmead Parkway
Sunnyvale, CA 94086
Tel: 408-733-9000
Fax: 408-733-9970

Idaho

Micro Sales
2122 - 112th Ave. N.E.
Bellevue, WA 98004
Tel: 206-451-0568
Fax: 206-453-0092

Illinois

Janus Incorporated
650 E. Devon Avenue
Itasca, IL 60143
Tel: 708-250-9650
Fax: 708-250-8761

Indiana

Fort Wayne

Electro Reps, Inc.
125 Airport North Office Park
Fort Wayne, IN 46825
Tel: 219-489-8205
Fax: 219-489-8408

Indianapolis

Electro Reps, Inc.
7240 Shadeland Station, Suite 275
Indianapolis, IN 46256
Tel: 317-842-7202
Fax: 317-841-0230

Iowa

Spectrum Sales
1364 Elmhurst Drive NE
Cedar Rapids, IA 52402
Tel: 319-366-0576
Fax: 319-366-0635

Kansas

Spectrum Sales
5382 W. 95th Street
Prairie Village, KS 66207
Tel: 913-648-6811
Fax: 913-648-6823

© 1995 Microchip Technology Inc.

Kentucky

Northern

Technology Marketing Corporation
9200 Montgomery Road
Suite 11A
Cincinnati, OH 45242
Tel: 513-984-6720
Fax: 513-984-6874

Southern

Electro Reps, Inc.
7240 Shadeland Station, Suite 275
Indianapolis, IN 46256
Tel: 317-842-7202
Fax: 317-841-0230

Louisiana

CompTech Sales, Inc.
10550 Richmond Avenue
Suite 105
Houston, TX 77042
Tel: 713-781-7420
Fax: 713-781-5865

Maine

S-J New England, Inc.
44 Mall Road
Burlington, MA 01803
Tel: 617-272-5552
Fax: 617-272-5515

Maryland

S-J Chesapeake
900 S. Washington Street
Suite 307
Falls Church, VA 22046
Tel: 703-533-2233
Fax: 703-533-2236

Massachusetts

S-J New England, Inc.
44 Mall Road
Burlington, MA 01803
Tel: 617-272-5552
Fax: 617-272-5515

Michigan

Farmington Hills

J. L. Montgomery Associates, Inc.
34405 West 12 Mile Road, Suite 149
P.O. Box 2726
Farmington Hills, MI 48333-2726
Tel: 810-489-0099
Fax: 810-489-0189

Grand Rapids

J. L. Montgomery Associates, Inc.
2215 Oak Industrial Drive NE
Grand Rapids, MI 49505
Tel: 616-458-5490
Fax: 616-458-5709

Minnesota

Comprehensive Technical Sales Inc.
6513 City West Parkway
Eden Prairie, MN 55344
Tel: 612-941-7181
Fax: 612-941-4322

Mississippi

Concord Component Reps Inc.
190 Lime Quarry Road, Suite 102
Madison, AL 35758
Tel: 205-772-8883
Fax: 205-772-8262

Missouri

Spectrum Sales
5494 Brown Road, Suite 124
St. Louis, MO 63042
Tel: 314-731-4477
Fax: 314-731-1332

Montana

Western Region Marketing
9176 Marshall Place
Westminster, CO 80030
Tel: 303-428-8088
Fax: 303-426-8585

Nebraska

Spectrum Sales
5382 W. 95th Street
Prairie Village, KS 66207
Tel: 913-648-6811
Fax: 913-648-6823

Nevada

Northern

Trinity Technologies
1261 Oakmead Parkway
Sunnyvale, CA 94086
Tel: 408-733-9000
Fax: 408-733-9970

Southern

Western High Tech Marketing, Inc.
9414 E. San Salvador, Suite 206
Scottsdale, AZ 85258
Tel: 602-860-2702
Fax: 602-860-2712

New Hampshire

S-J New England, Inc.
44 Mall Road
Burlington, MA 01803
Tel: 617-272-5552
Fax: 617-272-5515

New Jersey

Northern

Parallax
734 Walt Whitman Road
Melville, NY 11747
Tel: 516-351-1000
Fax: 516-351-1606

Southern

S-J Mid-Atlantic, Inc.
131-D Gaither Drive
Mt. Laurel, NJ 08054
Tel: 609-866-1234
Fax: 609-866-8627

New Mexico

Western High Tech Marketing, Inc.
9414 E. San Salvador, Suite 206
Scottsdale, AZ 85258
Tel: 602-860-2702
Fax: 602-860-2712

New York

Melville

Parallax
734 Walt Whitman Road
Melville, NY 11747
Tel: 516-351-1000
Fax: 516-351-1606

Rochester

Apex Associates, Inc.
1210 Jefferson Road
Rochester, NY 14623
Tel: 716-272-7040
Fax: 716-272-7756

North Carolina

Zucker Associates
4070 Barrett Drive
Raleigh, NC 27609
Tel: 919-782-8433
Fax: 919-782-8476

North Dakota

Comprehensive Technical Sales Inc.
6513 City West Parkway
Eden Prairie, MN 55344
Tel: 612-941-7181
Fax: 612-941-4322

© 1995 Microchip Technology Inc.

Factory Representatives

Ohio

Cincinnati

Technology Marketing Corporation
9200 Montgomery Road, Suite 11A
Cincinnati, OH 45242
Tel: 513-984-6720
Fax: 513-984-6874

Middleburg Heights

Technology Marketing Corporation
7017 Pearl Road
Middleburg Heights, OH 44130
Tel: 216-885-5544
Fax: 216-885-5011

Oklahoma

CompTech Sales, Inc.
18700 Woodbriar Lane
Catoosa, OK 74015
Tel: 918-266-1966
Fax: 918-266-1808

Oregon

Micro Sales
1865 N.W. 169th Pl. #210
Beaverton, OR 97006
Tel: 503-645-2841
Fax: 503-645-3754

Pennsylvania

Eastern

S-J Mid-Atlantic, Inc.
131-D Gaither Drive
Mt. Laurel, NJ 08054
Tel: 609-866-1234
Fax: 609-866-8627

Western

Technology Marketing Corporation
9200 Montgomery Road, Suite 11A
Cincinnati, OH 45242
Tel: 513-984-6720
Fax: 513-984-6874

Rhode Island

S-J New England, Inc.
44 Mall Road
Burlington, MA 01803
Tel: 617-272-5552
Fax: 617-272-5515

South Carolina

Zucker Associates
4070 Barrett Drive
Raleigh, NC 27609
Tel: 919-782-8433
Fax: 919-782-8476

South Dakota

Comprehensive Technical Sales Inc.
6513 City West Parkway
Eden Prairie, MN 55344
Tel: 612-941-7181
Fax: 612-941-4322

Tennessee

Eastern

Zucker Associates
4070 Barrett Drive
Raleigh, NC 27609
Tel: 919-782-8433
Fax: 919-782-8476

Western

Concord Component Reps Inc.
6825 Jimmy Carter Boulevard
Norcross, GA 30071
Tel: 404-416-9597
Fax: 404-441-0790

Texas

Austin

CompTech Sales, Inc.
11130 Jollyville Road, Suite 200
Austin, TX 78759
Tel: 512-343-0300
Fax: 512-345-2530

Brownsville

CompTech Sales, Inc.
2390 Central Blvd., Suite P
Brownsville, TX 78520
Tel: 210-504-9693
Fax: 210-504-9982

El Paso

CompTech Sales, Inc.
3120 Wheeling Rd.
El Paso, TX 79930
Tel: 915-566-1022
Fax: 915-566-1030

Houston

CompTech Sales, Inc.
10550 Richmond Avenue
Suite 105
Houston, TX 77042
Tel: 713-781-7420
Fax: 713-781-5865

Irving

CompTech Sales, Inc.
2401 Gateway Drive
Suite 114
Irving, TX 75063
Tel: 214-751-1181
Fax: 214-550-8113

Utah

Western Region Marketing
3539 South Main, Suite 210
Salt Lake City, UT 84115
Tel: 801-268-9768
Fax: 801-268-9796

Vermont

S-J New England, Inc.
44 Mall Road
Burlington, MA 01803
Tel: 617-272-5552
Fax: 617-272-5515

Virginia

S-J Chesapeake
900 S. Washington Street
Suite 307
Falls Church, VA 22046
Tel: 703-533-2233
Fax: 703-533-2236

Washington

Micro Sales
2122 - 112th Ave. N.E.
Bellevue, WA 98004
Tel: 206-451-0568
Fax: 206-453-0092

West Virginia

S-J Chesapeake
900 S. Washington Street
Suite 307
Falls Church, VA 22046
Tel: 703-533-2233
Fax: 703-533-2236

Wisconsin

Janus Incorporated
375 Williamstowne
Delafield, WI 53018
Tel: 414-646-5420
Fax: 414-646-2421

Wyoming

Western Region Marketing
9176 Marshall Place
Westminster, CO 80030
Tel: 303-428-8088
Fax: 303-426-8585

© 1995 Microchip Technology Inc.

MICROCHIP

Distributors

AFRICA

South Africa

Pace Electronic Components Ltd.
Cnr. Vanacht & Gewel Streets
P.O. Box 701
Isando 1600, Transvaal
Republic of South Africa
Tel: 27/11-9741211
Fax: 27/11-9741271

ASIA/PACIFIC

Australia

Har-Tec Australia
205 Middleborough Road
Box Hill, Victoria Australia 3128
Tel: 61-3-890-0970
Fax: 61-3-899-5191

China

Wuhan Liyuan
107-2 Luo Yu Road
Wuhan 430070
Peoples Republic of China
Tel: 86-27-7802986
Fax: 86-27-7802985

Fuzhou Dingxu
3/F, No3 Building
Chang Ting Industrial Area
Xianjin Rd
Fuzhou
Peoples Republic of China
Tel: 86-591-372-9174
Fax: 86-591-371-7704

Goldenchip Electronics
7/F, 275 Wushi Road
Fuzhou
Peoples Republic of China
Tel: 86-591-784-4159
Fax: 86-591-784-4160

Hong Kong

Goldenchip Research
11/F, 1116-7 Kwai on Fty Bldg
103 Tai Lin Pai Road
Kwai Chung, N.T. Hong Kong
Tel: 852-2426-3968
Fax: 852-2481-7403

Infinitron
Rm 802, 8/F Kinox Centre
9 Hung To Rd, Kwun Tong
Kowloon
Hong Kong
Tel: 852-2341-6611
Fax: 852-2950-0987

Excelpoint Systems
Rm. 2108, Fortress Tower
250 King's Road
North Point
Hong Kong
Tel: 852-2503-2212
Fax: 852-2503-1558

India

Excelpoint Systems
11/1 &111/2 Dickenson Road
Bangalore 560042 India
Tel: 91-80-5586719
Fax: 91-80-5586606

Japan

Dianichi Contronics Inc.
Dainichi Bldg. 1-7 Karaku-Chrome,
Bunkyo-Ku
Tokyo 112, Japan
Tel: 81-3-818-8081
Fax: 81-3-3818-8088

Marubeni Hytech Co., Ltd.
Marubeni Hytech Building
4-20-22, Koishikawa
Bunkyo-Ku
Tokyo 112 Japan
Tel: 81-3-3817-4921
Fax: 81-3-3817-4880

Japan (continued)

Nippon Precision Device Corp.
Nichibei Time 24 Bldg.
35 Tansu-Cho, Shinjuku-Ku
Tokyo 162 Japan
Tel: 81-3-3260-1411
Fax: 81-3-3260-7100

Ryoden Trading Co.
3-15-15, Higashi Ikebukuro
Toshima-Ku
Tokyo 170 Japan
Tel: 81-3-5396-6197
Fax: 81-3-5396-6443

Unidux
5-1-21, Kyonan-cho
Musashino-shi
Tokyo 180 Japan
Tel: 81-422-32-4111
Fax: 81-422-31-2050

Korea

ProCHIPS Inc.
779-12, Daelim 3-Dong
Youngdeungpo-Ku,
Seoul, Korea
Tel: 822-849-8567
Fax: 822-849-8659

New Zealand

Har-Tec NZ Limited
50 O'Rourke Road
P.O. Box 12055
Penrose, Aukland New Zealand
Tel: 64-9-525-1096
Fax: 64-9-525-1097

9

© 1995 Microchip Technology Inc.

Distributors

ASIA/PACIFIC (continued)

Singapore

Gates Engineering
5611 North Bridge Road
#06-01 Eng Cheong Tower
Singapore 0719
Tel: 65-299-9937
Fax: 65-299-7636

Excelpoint Systems
12 Tannery Lane #06-01/02
Beam Building
Singapore 1334
Tel: 65-741-8980
Fax: 65-741-8980

Taiwan, R.O.C.

Solomon Technology Corp.
5th Floor, No. 293, Sec. 5
Chung Hsiao E. Rd.
Taipei, Taiwan, R.O.C.
Tel: 886-02-788-8989
Fax: 866-02-788-8029

Pinnacle Technologies Co. Ltd.
4F, No. 270, Sec. 3
Nan-Kang Road
Taipei, Taiwan, R.O.C.
Tel: 02 788 4800
Fax: 02 788 5969

CANADA

Vancouver, British Columbia

Farnell Electronic Services
8525 Baxter Place, Unit 101
Production Court
Burnaby, B.C. V5A 4V7
Canada
Tel: 604-421-6222
Fax: 604-421-0582

Semad Electronics
3700 Gilmore Way, #212
Burnaby, B.C. V5G 4M1
Canada
Tel: 604-451-3444
Fax: 604-451-3445

Pioneer/Zentronics
4455 No. 6 Road
Richmond, B.C. V6V 1P8
Canada
Tel: 604-273-5575
Fax: 604-273-2413

Future Electronics
1695 Boundary Road
Vancouver, B.C. V5K 4X7
Canada
Tel: 604-294-1166
Fax: 604-294-1206

Calgary, Alberta

Farnell Electronic Services
3015 - 5th Ave. NE
Suite 210
Calgary, Alberta T2A 6T8
Canada
Tel: 403-273-2780
Fax: 403-273-7458

Future Electronics
2015 32nd Ave. N.E., Unit 1
Calgary, Alberta T2E 6Z3
Canada
Tel: 403-250-5550
Fax: 403-291-7054

Pioneer/Zentronics
#560, 1212-31st Avenue N.E.
Calgary, Alberta T2E 7S8
Canada
Tel: 403-291-1988
Fax: 403-291-0740

Semad Electronics
6815 8th St. N.E.
Suite 175
Calgary, Alberta T2E 7H7
Canada
Tel: 403-252-5664
Fax: 800-565-9779

© 1995 Microchip Technology Inc.

Edmonton, Alberta

Future Electronics
4606-97th Street
Edmonton, Alberta T6E 5NG
Canada
Tel: 403-438-2858
Fax: 403-434-0812

Pioneer/Zentronics
Plaza 124 #708
10216-124 Street
Edmonton, Alberta T5N 4A3
Canada
Tel: 403-482-3038
Fax: 403-482-1336

London, Ontario

Pioneer/Zentronics
148 York Street, Suite 209
London N6A 1A9
Canada
Tel: 519-672-4666
Fax: 519-672-3528

Toronto, Ontario

Farnell Electronic Services
300 N. Rivermede Road
Concord, Ontario L4K 2Z4
Canada
Tel: 416-798-4884
Fax: 416-798-4889

Semad Electronics
85 Spy Court
Markham, Ontario L3R 4Z4
Canada
Tel: 416-475-3922
Fax: 416-475-4158

Future Electronics
5935 Airport Road, Suite 200
Mississauga, Ontario L4V 1W5
Canada
Tel: 905-612-9200
Fax: 905-612-9185

Pioneer/Zentronics
3415 American Drive
Mississauga, Ontario L4V 1T4
Canada
Tel: 905-405-8300
Fax: 905-405-6423

Ottawa, Ontario

Farnell Electronic Services
39 Robertson Road, Suite 506
Bell Mews
Nepean, Ontario K2H 8R2
Canada
Tel: 613-596-6980
Fax: 613-596-6987

Pioneer/Zentronics
223 Colonnade Road, Suite 112
Nepean, Ontario K2E 7K3
Canada
Tel: 613-226-8840
Fax: 613-226-6352

Future Electronics
1050 Baxter Road
Ottawa, Ontario K2C 3P2
Canada
Tel: 613-820-8313
Fax: 613-820-3271

Semad Electronics
2781 Lancaster
Suite 302
Ottawa, Ontario K1B 1A7
Canada
Tel: 613-526-4866
Fax: 613-523-4372

Montreal, Quebec

Farnell Electronic Services
6600 Trans Canada Highway
Suite 620
Pointe Clare, Quebec H9R 4S2
Canada
Tel: 514-697-8149
Fax: 514-697-1210

Future Electronics/Branch
237 Hymus Boulevard
Pointe Claire, Quebec H9R 5C7
Canada
Tel: 514-694-7710
Fax: 514-695-3707

Semad Electronics
243 Place Frontenac
Pointe Clare, Quebec H9R 427
Canada
Tel: 514-694-0860
Fax: 514-694-0965

Pioneer/Zentronics
520 McCaffrey Street
Ville St. Laurent, Quebec H4T 1N1
Canada
Tel: 514-737-9700
Fax: 514-737-5212

Quebec City, Quebec

Future Electronics
1000 Ave. St. Jean Baptiste
Suite 100
Quebec G2E 5G5
Canada
Tel: 418-877-6666
Fax: 418-877-6671

Pioneer/Zentronics
2954 Blvd. Laurier
Suite 100
Ste-Foy, Quebec G1V 4T2
Canada
Tel: 418-654-1077
Fax: 418-654-2958

Winnipeg, Manitoba

Pioneer/Zentronics
540 Marjorie Street
Winnipeg, Manitoba R3H 0S9
Canada
Tel: 204-989-1957
Fax: 204-633-9255

Future Electronics
106 King Edward
Winnipeg, Manitoba R3H 0N8
Canada
Tel: 204-944-1446
Fax: 204-783-8133

Farnell Electronic Services
Unit 250, 1625 Dublin Ave.
Winnipeg, Manitoba R3H 0W3
Canada
Tel: 204-786-2589
Fax: 204-786-2637

© 1995 Microchip Technology Inc.

Distributors

EUROPE

Austria

Elbatex GmbH
Eitnergasse 6
A-1230 Wien
Austria
Tel: 43/1-86642-0
Fax: 43/1-86642-201

Belgium, Netherlands

Sonetech
P.O. Box 258
5670 AG Nuenen
Netherlands
Tel: 31-40-837075
Fax: 31-40-832300

Memec Benelux
Rue 1200 Bruxelles
Saint Lambert 135
Belgium
Tel: 010-32-2778-9850
Fax: 010-32-2778-9858

Denmark

Arrow Exatec
Mileparken 20E
2740 Skovlund
Denmark
Tel: 45-44-927000
Fax: 45-44-926020

England

Polar Electronics PLC
Cherrycourt Way
Leighton Buzzard
Bedfordshire LU7 8YY England
Tel: 44-1525-377093
Fax: 44-1525-378367

Farnell Multi
346 Edinburgh Avenue
Slough
Berkshire SL1 4TU England
Tel: 44-1753-84212
Fax: 44-1753-824160

Future Electronics Ltd.
Future House
Poyle Road, Colnbrook
Berkshire SL3 0EZ England
Tel: 44-1753-763000
Fax: 44-1753-689100

Eiger Technologies
14 Howard Court
Manor Park Avenue
Manor Park Runcorn
Cheshire WA7 1SJ England
Tel: 44-1928-579009
Fax: 44-1928-579123

Hawke Components Ltd.
26 Campbell Court
Bramley NR.Bassingstoke
Hantshire RG26 5EG England
Tel: 44-1258-880800
Fax: 44-1256-880325

H.B. Electronics Ltd.
Lever Street
Bolton
Lancshire BL3 6BJ England
Tel: 44-1204-525544
Fax: 44-1204-384911

Finland

Memec Finland
Asemankello
Vernissankatu 6
Fin 01300 Vantaa
Finland
Tel: 358-07001-9830
Fax: 358-07001-9839

France

Mecodis
Parc d'Activites
3 Allee des Erables
94042 CRETEIL Cedex
France
Tel: 33 1 43 99 44 00
Fax: 33 1 43 99 98 28

Farnell Electronic Services
BP 69 Saint Aubin
91192 GIF Sur Yvette Cedex
France
Tel: 33 1 69 85 83 00
Fax: 33 1 69 85 83 99

Scaib SA
6 rue Ambroise Croizat
ZI des Glaises - BP 58
91122 Palaiseau Cedex
France
Tel: 33 1 69 19 89 00
Fax: 33 1 69 19 89 20

© 1995 Microchip Technology Inc.

Germany

Rutronik RSC-Halbleiter GmbH
Industriestr. 2
D-75228 Ispringen
Germany
Tel: 49/7231-801-0
Fax: 49/7231-82282

Future Electronics Deutschland GmbH
Muenchner Str. 18
D-85774 Unterfoehring
Germany
Tel: 49/89-95727-0
Fax: 49/89-95727-140

Semitron W. Roeck GmbH
Im Gut 1
D-79790 Kuessaberg
Germany
Tel: 49/7742-8001-0
Fax: 49/7742-6901

Avnet E2000 GmbH
Stahlgruberring 12
D-81829 Muenchen
Germany
Tel: 49/89-4 5110-01
Fax: 49/89-4 5110-210

Metronik GmbH
Leonhardsweg 2
D-82008 Unterhaching
Germany
Tel: 49/89-61108-0
Fax: 49/89-61108-155

Greece

P. Caritato & Associates S.A.
Ilia Iliou 31
Athens 11743
Greece
Tel: 30 1 902 01 15
Fax: 30 1 901 70 24

Hungary

Humansoft Ltd.
Angol u. 24/b
H-1149 Budapest
Hungary
Tel: 36/1162-2879
Fax: 36/1251-3673

Israel

Elina Electronics Ltd.
14 Raoul Wallenberg St.
P.O. Box 13190
Tel Aviv 61131 Israel
Tel: 972 3 49 85 43
Fax: 972 3 49 87 45

Italy

Eurelettronica Srl
Viale E. Fermi 8
20090 Assago Milano
Italy
Tel: 39-2-457841
Fax: 39-2-4880275

Farnell Spa
Viale Milanofiori E/5
20090 Assago Milano
Italy
Tel: 39-2-824701
Fax: 39-2-82470278

Kevin
Via del Gradenigo, 3
20148 Milano
Italy
Tel: 39-2-48706300
Fax: 39-2-48706500

Norway

Berendsen
P.O. Box 9376 Gronlund
N0135
Oslo Norway
Tel: 47-22-677290
Fax: 47-22-677380

Portugal

Digicontrole
Dpt Comercial
Aven. Eng Arantes E. Oliveira 52d
1900 LISBOA
Portugal
Tel: 35 11 80 57 30
Fax: 35 11 849 03 73

Russia

Gamma Ltd.
Grazdanski PR-T
111 KOM 427, 429
Elektronmash Building
R-195265 St. Petersburg
Russia
Tel: 7/812-5311-180
Fax: 7/812-5311-402

Spain

Sagitron
Corazon de Maria 80/82
28002 Madrid
Spain
Tel: +34 1 416 92 61
Fax: +34 1 415 86 52

Sweden

Memec Scandinavia AB
Kvarnholmsvagen 52
131 31 Nacka
Sweden
Tel: 011-46-8643-4190
Fax: 011-46-8643-1195

Switzerland

Elbatex Gruppe AG
Hardstr. 72
CH-5430 Wettingen
Switzerland
Tel: 41/56-275100
Fax: 41/56-275454

Semitron W:Roeck & Co.
Promenadenstr.6
CH-8047 Zuerich
Switzerland
Tel: 41/56-493383
Fax: 41/56-493569

Avnet E2000 AG
Boehnirainstr. 11
CH-8801 Thalwil
Switzerland
Tel: 41/1-7221330
Fax: 41/1-7221340)

Turkey

Inter Muehendislik Danismanlik
Ve Ticaret A.S. 1
Hasircbasi Caddesi No. 55
81310 Kadikoy
Istanbul
Turkey
Tel: 90 216 349 94 00
Fax: 90 216 349 94 31

© 1995 Microchip Technology Inc.

Distributors

MEXICO

Mexico, D.F.

Semiconductores Profesionales
Madrid No. 55
Col. Del Carmen Coyoacan
04100 Mexico, D. F.
Tel: 525-658-60-11
Fax: 525-658-60-44

Edo de Mexico

Electronica Seta
Galeana No. 114-20 Piso
La Loma Tlalnepantla
54060 Edo de Mexico
Tel: 525-390-77-13
Fax: 525-390-94-68

Guadalajara

Future Electronics de Mexico
S.A. de C.V.
Prol. Americas 1600
2 do Pisco
Guadalajara, Jalisco
Mexico 44610
Tel: 523-678-9281
Fax: 523-678-9200

SOUTH AMERICA

Argentina

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Bolivia

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Brazil

Aplicacoes Eletronicas
Artimar Ltda.
Rua Marques de Itu N°. 70-10And.
CEP 01223
Sao Paulo, Brazil
Tel: 55-11-231-0277
Fax: 55-11-255-0511

Caribbean

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Chile

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Columbia

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Ecuador

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Paraguay

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Peru

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Uruguay

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

Venezula

Ibars Electronics Corporation
10020 N.W. 6th Ct.
Pembroke Pines, Florida 33024
USA
Tel: 305-430-3740
Fax: 305-430-3763

© 1995 Microchip Technology Inc.

UNITED STATES

Alabama

Future Electronics
4825 University Square, Suite 12
Huntsville, AL 35816
Tel: 205-830-2322
Fax: 205-830-6664

Pioneer Technologies
4835 University Square, Suite 5
Huntsville, AL 35816
Tel: 205-837-9300
Fax: 205-837-9358

Arizona

Bell Industries
10611 North Hayden Road
Building D, Suite 103
Scottsdale, AZ 85260
Tel: 602-905-2355
Fax: 602-905-2356

Future Electronics
4636 E. University Dr., Suite 245
Phoenix, AZ 85034
Tel: 602-968-7140
Fax: 602-968-0334

Pioneer Standard
1438 W. Broadwa, Suite B140
Tempe, AZ 85282
Tel: 602-350-9335
Fax: 602-350-9376

California

San Jose Area

Bell Industries
1161 N. Fairoaks Ave.
Sunnyvale, CA 94089
Tel: 408-734-8570
Fax: 408-734-8875

Future Electronics
1024 Marilyn Dr.
Mountain View, CA 94040
Tel: 408-232-1998
Fax: 408-433-0822

Future Electronics
2220 O'Toole Ave.
San Jose, CA 95131
Tel: 408-434-1122
Fax: 408-433-0822

Pioneer Technical Products
333 River Oaks Parkway
San Jose, CA 95134
Tel: 408-954-9100
Fax: 408-954-9113

Roseville

Bell Industries
300 Douglas Blvd., Suite 205
Roseville, CA 95661
Tel: 916-781-8070
Fax: 916-781-2954

Future Electronics
755 Sunrise Avenue, Suite 150
Roseville, CA 95661
Tel: 916-783-7877
Fax: 916-783-7988

Agoura Hills

Bell Industries
30101 Agoura Court, Suite 118
Agoura Hills, CA 91301
Tel: 818-865-7900
Fax: 818-991-7695

Future Electronics
27489 West Agoura Road, Suite 300
Agoura Hills, CA 91301
Tel: 818-865-0040
Fax: 818-865-1340

Pioneer Standard
5126 Clareton Dr., Suite 160
Agoura Hills, CA 91301
Tel: 818-865-5800
Fax: 818-865-5814

Los Angeles

Bell Industries
11812 San Vicente Blvd., Suite 300
Los Angeles, CA 90049
Tel: 310-826-2355
Fax: 310-447-3265

Irvine

Bell Industries
220 Technology Dr. #100
Irvine, CA 92718
Tel: 714-727-4500
Fax: 714-453-4610

Future Electronics
258 Technology, Suite 200
Irvine, CA 92718
Tel: 714-453-1515
Fax: 714-453-1226

Pioneer Standard
217 Technology Drive #110
Irvine, CA 92718
Tel: 800-753-5090
Fax: 714-753-5074

San Diego

Aegis Electronic Group, Inc.
1015 Chestnut Ave., Suite G2
Carlsbad, CA 92008
Tel: 619-729-2026
Fax: 619-729-9295

Bell Industries
5520 Ruffin Rd., Suite 209
San Diego, CA 92123
Tel: 619-576-3294
Fax: 800-444-0139

Future Electronics
5151 Shoreham Place, Suite 220
San Diego, CA 92122
Tel: 619-625-2800
Fax: 619-625-2810

Pioneer Standard
9449 Balboa Ave., Suite 114
San Diego, CA 92123
Tel: 619-514-7700
Fax: 619-514-7799

© 1995 Microchip Technology Inc.

Distributors

Colorado

Bell Industries
1873 S. Bellaire St.
Denver, CO 80222
Tel: 303-691-9270
Fax: 303-790-4991

Future Electronics
12600 W. Colfax Avenue
Suite B110
Lakewood, CO 80215
Tel: 303-232-2008
Fax: 303-232-2009

Pioneer Technologies
5600 Greenwood Plaza Blvd.
Suite 201
Englewood, CO 80111
Tel: 303-773-8090
Fax: 303-773-8194

Connecticut

Bell Industries
1064 East Main Street
Meriden, CT 06450
Tel: 203-639-6000
Fax: 203-639-6005

Future Electronics
700 W. Johnson Ave.
Westgate Office Center
Cheshire, CT 06410
Tel: 203-250-0083
Fax: 203-250-0081

Pioneer Standard
Two Trap Falls #101
Shelton, CT 06484
Tel: 203-929-5600
Fax: 203-929-9791

Florida

Altamonte Springs

Bell Industries
650 S. North Lake Blvd. #400
Altamonte Springs, FL 32701
Tel: 407-339-0078
Fax: 407-339-0139

Future Electronics
237 S. Westmonte Drive, Suite 307
Altamonte Springs, FL 32714
Tel: 407-865-7900
Fax: 407-865-7660

Pioneer Technologies
337 South-North Lake, Suite 1000
Altamonte Springs, FL 32701
Tel: 407-834-9090
Fax: 407-834-0865

Deerfield Beach

Future Electronics
1400 E. Newport Center Drive
Suite 200
Deerfield Beach, FL 33442
Tel: 305-426-4043
Fax: 305-426-3939

Pioneer Technologies
674 S. Military Trail
Deerfield Beach, FL 33442
Tel: 305-428-8877
Fax: 305-481-2950

Largo

Future Electronics
2200 Tall Pines Drive, Suite 108
Largo, FL 34641
Tel: 813-530-1222
Fax: 813-538-9598

Georgia

Pioneer Technologies
4250 C Rivergreen Pkwy.
Duluth, GA 30136
Tel: 404-623-1003
Fax: 404-623-0665

Bell Industries
3020 Business Park Drive, Suite A
Norcross, GA 30071
Tel: 404-446-7167
Fax: 404-446-7264

Future Electronics
3150 Holcomb Bridge Rd.
Suite 130
Norcross, GA 30071
Tel: 404-441-7676
Fax: 404-441-7580

Illinois

Bell Industries
870 Cambridge Drive
Elk Grove Village, IL 60007
Tel: 708-640-1910
Fax: 708-640-1942

Future Electronics
3150 W. Higgins Rd., Suite 160
Hoffman Estates, IL 60195
Tel: 708-882-1255
Fax: 708-490-9290

Pioneer Standard
2171 Executive Drive #200
Addison, IL 60101
Tel: 708-495-9680
Fax: 708-495-9831

© 1995 Microchip Technology Inc.

Indiana

Fort Wayne

Bell Industries
525 Airport North Office Park
Fort Wayne, IN 46825
Tel: 219-490-2100
Fax: 219-490-2104

Pioneer Standard
237 Airport N. Office Park
Fort Wayne, IN 46825
Tel: 219-489-0283
Fax: 219-489-6262

Indianapolis

Bell Industries
5230 W. 79th St.
P.O. Box 6885
Indianapolis, IN 46268
Tel: 317-875-8200
Fax: 317-875-8219

Future Electronics
8425 Woodfield Crossing
Suite 175
Indianapolis, IN 46240
Tel: 317-469-0447
Fax: 317-469-0448

Pioneer Standard
9350 N. Priority Way W. Dr.
Indianapolis, IN 46240
Tel: 317-573-0880
Fax: 317-573-0979

Kansas

Future Electronics
8826 Sante Fe Drive, Suite 150
Overland Park, KS 66212
Tel: 913-649-1531
Fax: 913-649-1786

Maryland

Bell Industries
8945 Guilford Rd., Suite 130
Columbia, MD 21046
Tel: 800-274-6953
Fax: 410-290-8006

Future Electronics
6716 Alexander Bell Drive
Suite 101
Columbia, MD 21046
Tel: 410-290-0600
Fax: 410-290-0328

Seymour Electronics
Columbia Business Center
6440 Dobbin Road, Suite B
Columbia, MD 21045
Tel: 410-992-7474
Fax: 410-992-7410

Pioneer Technologies
9100 Gaither Road
Gaithersburg, MD 20877
Tel: 301-921-3953
Fax: 301-921-4255

Pioneer Technologies
15810 Gaither Drive
Gaithersburg, MD 20877
Tel: 301-921-3822
Fax: 301-921-3858

Massachusetts

Bell Industries
100 Burtt Road #106
Andover, MA 01810
Tel: 508-474-8880
Fax: 508-474-8902

Future Electronics
41 Main Street
Bolton, MA 01740
Tel: 508-779-3000
Fax: 508-779-3050

Pioneer Standard
44 Hartwell Ave.
Lexington, MA 02173
Tel: 617-861-9200
Fax: 617-863-1547

Michigan

Grand Rapids

Future Electronics
4505 Broadmoor S.E.
Grand Rapids, MI 49512
Tel: 616-698-6800
Fax: 616-698-6821

Pioneer Standard
4467 Byron Ctr Rd SW
Grand Rapids, MI 49509
Tel: 616-534-3145
Fax: 616-534-3922

Detroit Area

Future Electronics
35200 Schoolcraft Road, Suite 106
Livonia, MI 48150
Tel: 313-261-5270
Fax: 313-261-8125

Pioneer Standard
44190 Plymouth Oak Blvd.
Plymouth, MI 48170
Tel: 313-525-1800
Fax: 313-427-3720

© 1995 Microchip Technology Inc.

Distributors

Minnesota

Bloomington

Bell Industries
9400 James Ave. So. #142
Bloomington, MN 55431
Tel: 612-888-7247
Fax: 612-888-7757

Eden Prairie

Future Electronics
10025 Valley View Road, Suite 196
Eden Prairie, MN 55344
Tel: 612-944-2200
Fax: 612-944-2520

Pioneer Standard
7625 Golden Triangle
Eden Prairie, MN 56344
Tel: 612-944-3355
Fax: 612-944-3794

Thief River Falls

Digi-Key Corporation
701 Brooks Ave. So.
P.O. Box 677
Thief River Falls, MN 55344
Tel: 218-681-6674
Fax: 218-681-3380

Missouri

Future Electronics
12125 Woodcrest Executive Dr.
Suite 220
St. Louis, MO 63141
Tel: 314-469-6805
Fax: 314-469-7226

New Jersey

Northern

Bell Industries
271 Route 46 West
Suites F202-203
Fairfield, NJ 07004
Tel: 201-227-6060
Fax: 201-227-2626

Pioneer Standard
14A Madison Road
Fairfield NJ 07006
Tel: 201-575-3510
Fax: 201-575-3454

Future Electronics
1259 Route 46 East
Parsippany, NJ 07054
Tel: 201-299-0400
Fax: 201-299-1377

Phase 1 Technology
295 Molnar Drive
Elmwood Park, NJ 07407
Tel: 201-791-2990
Fax: 201-791-2552

Seymour Electronics Corporation
357 Crossways Park Drive
Woodbury, NY 11797-2042
Tel: 516-496-7474
Fax: 516-496-0857

Southern

Bell Industries
158 Gaither Drive, Suite 110
Mt. Laurel, NJ 08054
Tel: 609-439-9009
Fax: 609-439-0570

Seymour Electronics
520 Fellowship Road, Suite A104
Mt. Laurel, NJ 08054
Tel: 609-235-7474
Fax: 609-235-4992

Future Electronics
12 East Stow Road, Suite 200
Marlton, NJ 08053
Tel: 609-596-4080
Fax: 609-596-4266

New Mexico

Bell Industries
11728 Linn N.E.
Albuquerque, NM 87123
Tel: 505-292-2700
Fax: 505-275-2819

New York

Binghamton

Pioneer Standard
1249 Front Street, Suite 201
Binghamton, NY 13901
Tel: 607-722-9300
Fax: 607-722-9562

Long Island

Future Electronics
801 Motor Parkway
Hauppauge, NY 11788
Tel: 516-234-4000
Fax: 516-234-6183

Phase 1 Technology
46 Jefryne Blvd.
Deer Park, NY 11729
Tel: 516-254-2600
Fax: 516-254-2693

Pioneer Standard
60 Crossways Park West
Woodbury, NY 11797
Tel: 516-921-8700
Fax: 516-921-2143

Seymour Electronics
357 Crossways Park Drive
Woodbury, NY 11797-2042
Tel: 516-496-7474
Fax: 516-496-0857

Fairport

Pioneer Standard
840 Fairport Park
Fairport, NY 14450
Tel: 716-381-7070
Fax: 716-381-5955

Rochester

Future Electronics
300 Linden Oaks
Rochester, NY 14625
Tel: 716-387-9550
Fax: 716-387-9563

Syracuse

Future Electronics
200 Salina Meadows Parkway
Suite 130
Syracuse, NY 13212
Tel: 315-451-2371
Fax: 315-451-7258

© 1995 Microchip Technology Inc.

North Carolina

Future Electronics
8401 University Executive PK.
Suite 108
Charlotte, NC 28262
Tel: 704-547-1107
Fax: 704-547-9650

Future Electronics
5225 Capital Blvd.
1 North Commerce Center
Raleigh, NC 27604
Tel: 919-790-7111
Fax: 919-790-9022

Pioneer Technologies
2200 Gateway Centre Blvd.
Suite 215
Morrisville, NC 27560
Tel: 919-460-1530
Fax: 919-460-1540

Ohio

Cleveland

Pioneer Standard
4800 East 131st St.
Cleveland, OH 44105
Tel: 216-498-6305
Fax: 216-587-3906

Future Electronics
6009-E Landerhaven Dr.
Mayfield Heights, OH 44124
Tel: 216-449-6996
Fax: 216-449-8987

Bell Industries
31200 Solon Road Unit 11
Solon, OH 44139
Tel: 216-498-2002
Fax: 216-498-2006

Pioneer Standard
29125 Solon Road
Solon, OH 44139
Tel: 216-248-8710
Fax: 216-248-9166

Dayton

Bell Industries
446 Windsor Park Drive
Dayton, OH 45459
Tel: 513-434-8231
Fax: 513-434-8103

Pioneer Standard
4433 Interpoint Blvd.
Dayton, OH 45424
Tel: 513-236-9900
Fax: 513-236-8133

Future Electronics
1430 Oak Ct., Suite 203
Beavercreek, OH 45430
Tel: 513-426-0090
Fax: 513-426-8490

Worthington

Pioneer Standard
100 Old Wilson Bridge, Suite 105
Worthington, OH 43085
Tel: 614-848-4854
Fax: 614-848-4889

Oklahoma

Pioneer Standard
9717 E. 42nd St., Suite 105
Tulsa, OK 74146
Tel: 918-665-7840
Fax: 918-665-1891

Oregon

Bell Industries
9275 S.W. Nimbus
Beaverton, OR 97005
Tel: 503-644-3444
Fax: 503-520-1948

Future Electronics
15236 N.W. Greenbrier Pkwy.
Beaverton, OR 97006
Tel: 503-645-9454
Fax: 503-645-1559

Pioneer Technologies
8905 SW Nimbus, Suite 160
Beaverton, OR 97008
Tel: 503-626-7300
Fax: 503-626-5300

Pennsylvania

Horsham

Pioneer Technologies
500 Enterprise Road
Keith Valley Business Center
Horsham, PA 19044
Tel: 215-674-4000
Fax: 215-674-3107

Pittsburgh

Pioneer Standard
259 Kappa Drive
Pittsburgh, PA 15238
Tel: 412-782-2300
Fax: 412-963-8255

© 1995 Microchip Technology Inc.

Distributors

Texas

Austin

Future Electronics
9020 II Capital of Texas Hwy N
Suite 610
Austin, TX 78759
Tel: 512-502-0991
Fax: 512-502-0740

Dallas

Pioneer Standard
13765 Beta Road
Dallas, TX 75244
Tel: 214-386-7300
Fax: 214-490-6419

Richardson

Bell Industries
1701 Greenville Ave. #306
Richardson, TX 75081
Tel: 214-690-9096
Fax: 214-690-0467

Future Electronics
800 E. Campbell Rd., Suite 130
Richardson, TX 75081
Tel: 214-437-2437
Fax: 214-669-2347

Houston

Future Electronics
10333 Richmond Ave., Suite 970
Houston, TX 77042
Tel: 713-785-1155
Fax: 713-785-4558

Pioneer Standard
10530 Rockley Road, Ste 100
Houston, TX 77099
Tel: 713-495-4700
Fax: 713-495-5642

Utah

Bell Industries
6912 S. 185 West, Suite B
Midvale, UT 84047
Tel: 801-561-9691
Fax: 801-255-2477

Future Electronics
3540 S. Highland Drive #301
Salt Lake City, UT 84106
Tel: 801-467-4448
Fax: 801-467-3604

Washington

Bell Industries
1715 114th Ave SE #208
Bellevue, WA 98004
Tel: 206-646-8750
Fax: 206-646-8559

Pioneer Technologies
2800 156th Ave., SE, Suite 100
Bellevue, WA 98007
Tel: 206-644-7500
Fax: 206-644-7300

Future Electronics
19102 N. Creek Parkway South
Suite 118
Bothell, WA 98011
Tel: 206-489-3400
Fax: 206-489-3411

Wisconsin

Future Electronics
250 N. Patrick Boulevard, Suite 170
Brookfield, WI 53045
Tel: 414-879-0244
Fax: 414-879-0250

Pioneer Standard
120 Bishop's Way, Suite 163
Brookfield, WI 53005
Tel: 414-784-3480
Fax: 414-784-8207

Bell Industries
W 226 N 900 Eastmound
Waukesha, WI 53186
Tel: 414-547-8879
Fax: 414-547-6547

Authorized Distributor for Obsolete Products
Rochester Electronics, Inc.
10 Malcolm Hoyt Drive
Newburyport, MA 01950
Tel: 508-462-9332
Fax: 508-462-9512

© 1995 Microchip Technology Inc.

MICROCHIP

Factory Sales

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200
Fax: 602 786-7277

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 404 640-0034
Fax: 404 640-0307

Boston
Microchip Technology Inc.
Five The Mountain Road, Suite 120
Framingham, MA 01701
Tel: 508 820-3334
Fax: 508 820-4326

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071
Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177
Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543
Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888
Fax: 714 263-1338

AMERICAS (continued)

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305
Fax: 516 273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950
Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology Inc.
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200
Fax: 852 2 401 3431

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200
Fax: 82 2 558 5934

Singapore
Microchip Technology Inc.
200 Middle Road
#10-03 Prime Centre
Singapore 0718
Tel (mobile): 65 634 2305

Taiwan
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175
Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire
SL8 5AJ
Tel: 44 (0) 1628 851077
Fax: 44 (0) 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20
Fax: 33 1 69 30 90 79

Germany
Arizona MicrochipTechnology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0
Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939
Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166
Fax: 81 45 471 6122

© 1995 Microchip Technology Inc.

Factory Sales

MICROCHIP

Printed in the USA, 5/95
© 1995, Microchip Technology Inc.

Microchip Technology Inc. • 2355 W. Chandler Blvd. • Chandler, Arizona USA • 602-786-7200 • FAX: 602-899-9210

© 1995 Microchip Technology Inc.

Microchip Worldwide Sales and Distribution

MICROCHIP
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602.786.7200 Fax: 602.899.9210

©1995 Microchip Technology Inc. Printed in the U.S.A. 5/95 DS00158A