PIC14000

28-Pin Programmable Mixed Signal Controller

High-Performance RISC CPU:

- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
- 4096 x 14 on-chip EPROM program memory
- 192 x 8 general purpose registers (SRAM)
- 6 internal and 5 external interrupt sources
- 38 special function hardware registers
- Eight-level hardware stack

Analog Peripherals Features:

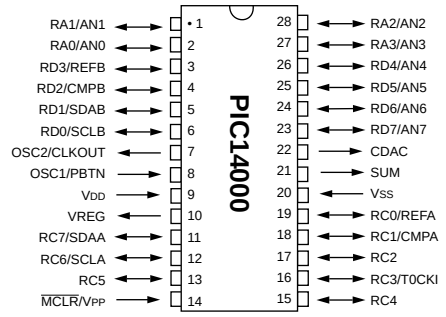
- Slope Analog-to-Digital (A/D) converter
  - Eight external input channels including two channels with selectable level shift inputs
  - Six internal input channels
  - 16-bit programmable timer with capture register
  - 16 ms maximum conversion time at maximum (16-bit) resolution and 4 MHz clock
  - 4-bit programmable current source
- Internal bandgap voltage reference
- Factory calibrated with calibration constants stored in EPROM
- On-chip temperature sensor
- Voltage regulator control output
- Two comparators with programmable references
- On-chip low voltage detector

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
- Multi-segment programmable code-protection
- Selectable oscillator options
  - Internal 4 MHz oscillator
  - External crystal oscillator
- Serial in-system programming (via two pins)

Pin Diagram

PDIP, SOIC, SSOP, Windowed CERDIP



Digital Peripherals Features:

- 22 I/O pins with individual direction control
- High current sink/source for direct LED drive
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler
- 16-bit A/D timer: can be used as a general purpose timer
- I^2C^{TM} serial port compatible with System Management Bus

CMOS Technology:

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range (2.7V to 6.0V)
- Commercial and Industrial Temperature Range
- Low power dissipation (typical)
  - < 3 mA @5V, 4 MHz operating mode
  - < 300 µA @3V (Sleep mode: clocks stopped with analog circuits active)
  - < 5 µA @3V (Hibernate mode: clocks stopped, analog inactive, and WDT disabled)

Applications:

- Battery Chargers
- Battery Capacity Monitoring
- Uninterruptable Power Supply Controllers
- Power Management Controllers
- HVAC Controllers
- Sensing and Data Acquisition

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.

1.0 GENERAL DESCRIPTION

The PIC14000 features include medium to high resolution A/D conversion (10 to 16 bits), temperature sensing, closed loop charge control, serial communication, and low power operation.

The PIC14000 uses a RISC Harvard architecture CPU with separate 14-bit instruction and 8-bit data buses. A 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 are available. Additionally, a large register set is included.

PIC16/17 microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers.

Features:

The PIC14000 is a 28-pin device with these features:

- 4K of EPROM
- 192 bytes of RAM
- 22 I/O pins

The analog peripherals include:

- 8 external analog input channels, two with level shift inputs
- 6 internal analog input channels
- 2 comparators with programmable references
- A bandgap reference
- An internal temperature sensor
- A programmable current source

In addition, the I^2C serial port through a multiplexer supports two separate I^2C channels.

A special oscillator option allows either an internal 4 MHz oscillator or an external crystal oscillator. Using the internal 4 MHz oscillator requires no external components.

The PIC14000 contains three timers, the Watchdog Timer (WDT), Timer0 (TMR0), and A/D Timer (ADTMR). The Watchdog Timer includes its own on-chip RC oscillator providing protection against software lock-up. TMR0 is a general purpose 8-bit timer/counter with an 8-bit prescaler. It may be clocked externally using the RC3/T0CKI pin. The ADTMR is intended for use with the slope A/D converter, but can also be used as a general purpose timer. It has an associated capture register which can be used to measure the time between events.

An internal low-voltage detect circuit allows for tracking of voltage levels. Upon detecting the low voltage condition, the PIC14000 can be instructed to save its operating state then enter an idle state.

The internal band-gap reference is used for calibrating the measurements of the analog peripherals. The calibration factors are stored in EPROM and can be used to achieve high measurement accuracy.

Power savings modes are available for portable applications. The SLEEP and HIBERNATE modes offer different levels of power savings. The PIC14000 can wake up from these modes through interrupts or reset.

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.

The PIC14000 fits perfectly in applications for battery charging, capacity monitoring, and data logging. The EPROM technology makes customization of application programs (battery characteristics, feature sets, etc.) extremely fast and convenient. The small footprint packages make this microcontroller based mixed signal device perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC14000 very versatile in other applications such as temperature monitors/controllers.

1.1 Family and Upward Compatibility

Code written for PIC16C6X/7X can be easily ported to the PIC14000 (see Appendix A).

1.2 Development Support

The PIC14000 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.

NOTES:

2.0 DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. The PIC14000 Product Selection System section at the end of this data sheet provides the devices options to be selected for your specific application and production requirements. When placing orders, please use the “PIC14000 Product Identification System” at the back 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.

Note:	Please note that erasing the device will also erase the pre-programmed calibration factors. Please refer to AN621 for more information.

Microchip's PICSTART®, PICSTART-PLUS and PRO MATE™ programmers all support programming of the PIC14000. Third party programmers also are available; refer to the *Microchip 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 or 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.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The PIC14000 addresses 4K x 14 program memory. All program memory is internal. The PIC14000 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 PIC14000 has an orthogonal 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 PIC14000 simple yet efficient. In addition, the learning curve is reduced significantly.

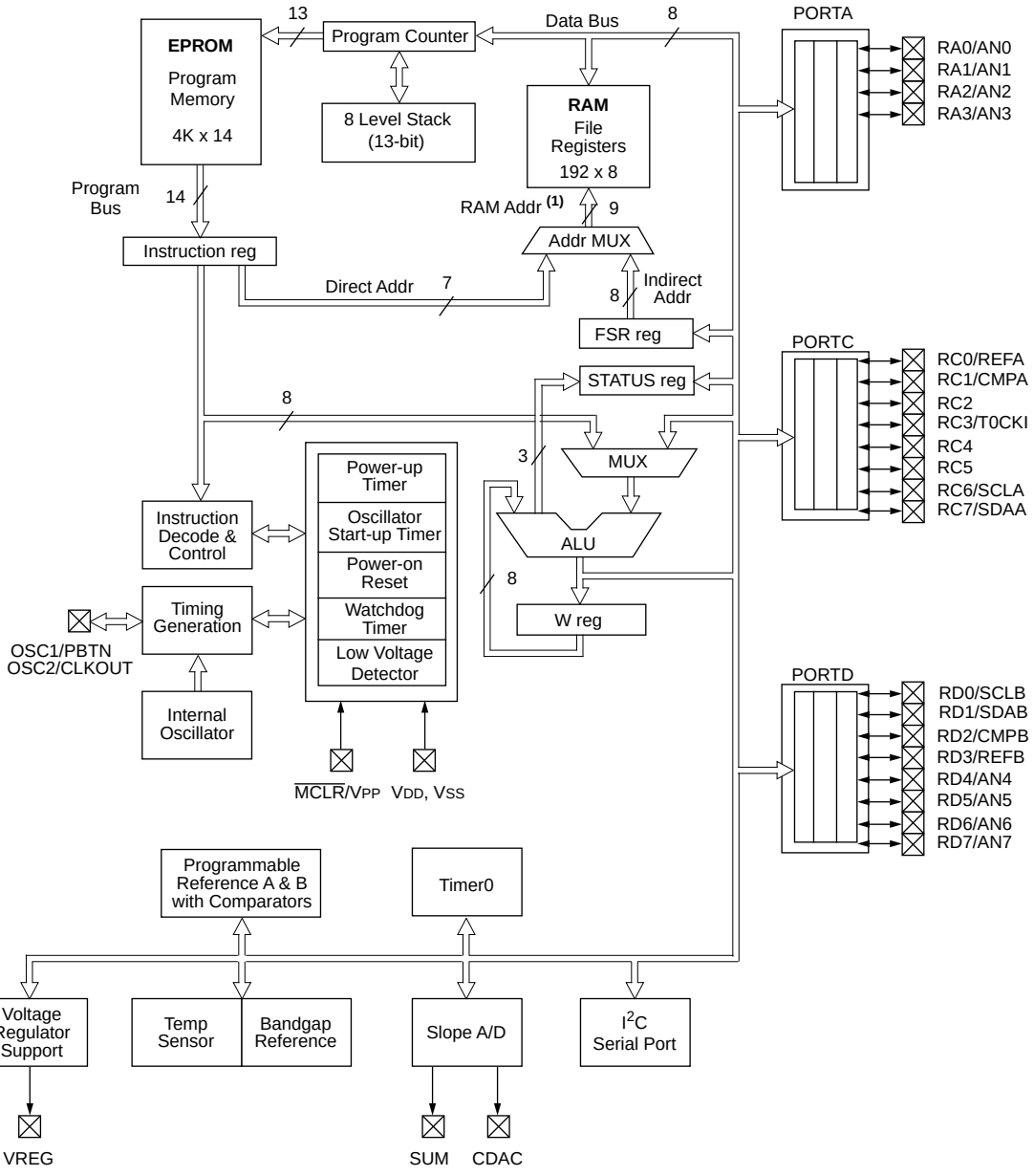
The PIC14000 contains an 8-bit ALU and working register. The ALU performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is capable of addition, subtraction, shift, and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement. 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.

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 $\overline{\text{borrow}}$ bit and a $\overline{\text{digit}}$ $\overline{\text{borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram for the PIC14000 is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

FIGURE 3-1: PIC14000 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

TABLE 3-1: PIN DESCRIPTIONS

Pin Name	Pin No.	I/O	Pin Type Input	Pin Type Output	Description
CDAC	22	O	—	AN	A/D ramp current source output. Normally connected to external capacitor to generate a linear voltage ramp.
RA0/AN0	2	I/O	AN/ST	CMOS	Analog input channel 0. This pin can also serve as a general-purpose I/O.
RA1/AN1	1	I/O	AN/ST	CMOS	Analog input channel 1. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. This pin can also serve as a general-purpose I/O.
RA2/AN2	28	I/O	AN/ST	CMOS	Analog input channel 2. This pin can also serve as a general purpose digital I/O.
RA3/AN3	27	I/O	AN/ST	CMOS	Analog input channel 3. This pin can also serve as a general purpose digital I/O.
SUM	21	O	—	AN	AN1 summing junction output. This pin can be connected to an external capacitor for averaging small duration pulses.
RC0/REFA	19	I/O-PU	ST	CMOS	LED direct-drive output or programmable reference A output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to V_{DD}.
RC1/CMPA	18	I/O-PU	ST	CMOS	LED direct-drive output or comparator A output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to V_{DD}.
RC2	17	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to V_{DD}
RC3/T0CKI	16	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO, or an external clock input for Timer0. If enabled, this pin has a weak internal pull-up to V_{DD}.
RC4	15	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to V_{DD}.
RC5	13	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to V_{DD}.
RC6/SCLA	12	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial clock for I^2C interface. Also is the serial programming clock. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RC7/SDAA	11	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial data I/O for I^2C interface. Also is the serial programming data line. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD0/SCLB	6	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial clock for I^2C interface. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD1/SDAB	5	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial data I/O for I^2C interface. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD2/CMPB	4	I/O-PU	AN/ST	CMOS	General purpose I/O or comparator B output.

TABLE 3-1: PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin No.	I/O	Pin Type Input	Pin Type Output	Description
RD3/REFB	3	I/O-PU	AN/ST	CMOS	General purpose I/O or programmable reference B output.
RD4/AN4	26	I/O	AN/ST	CMOS	Analog input channel 4. This pin can also serve as a GPIO.
RD5/AN5	25	I/O	AN/ST	CMOS	Analog input channel 5. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. This pin can also serve as a GPIO.
RD6/AN6	24	I/O	AN/ST	CMOS	Analog input channel 6. This pin can also serve as a GPIO.
RD7/AN7	23	I/O	AN/ST	CMOS	Analog input channel 7. This pin can also serve as a GPIO.
VREG	10	O	—	AN	This pin is an output to control the gate of an external N-FET for voltage regulation.
OSC1/PBTN	8	I-PU	ST	—	IN Mode: Input with weak pull-up resistor, can be used to generate an interrupt. HS Mode: External oscillator input.
OSC2/ CLKOUT	7	O	—	CMOS	IN Mode: General purpose output. HS Mode: External oscillator/clock output.
$\overline{\text{MCLR}}$/VPP	14	I/PWR	ST		Master clear (reset) input / programming voltage input. This pin is an active low reset to the device.
V_{DD}	9	PWR			Positive supply connection
V_{SS}	20	GND			Return supply connection

Legend:

Type:	Definition:
TTL	TTL-compatible input
CMOS	CMOS-compatible input or output
ST	Schmitt Trigger input, with CMOS levels
SM	SMBus compatible input
OD	Open-drain output. An external pull-up resistor is required if this pin is used as an output.
NPU	N-channel pull-up. This pin will pull-up to approximately V_{DD} - 1.0V when outputting a logical ‘1’.
PU	Weak internal pull-up (10K-50K ohms)
No-P diode	No P-diode to V_{DD}. This pin may be pulled above the supply rail (to 6.0V maximum).
AN	Analog input or output

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1 or the internal oscillator) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. 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 are shown in Figure 3-2.

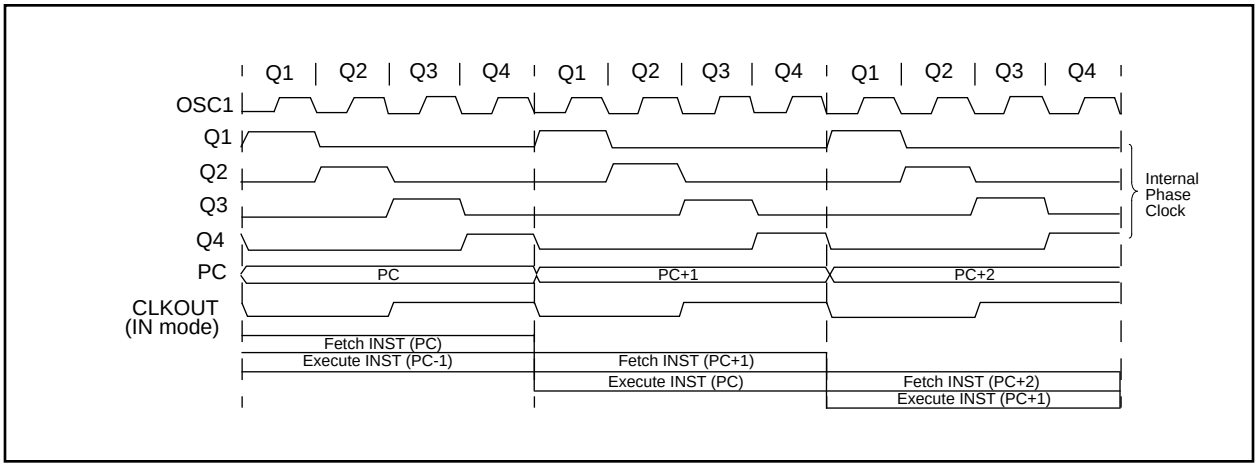
3.2 Instruction Flow/Pipelining

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 (Example 3-1).

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

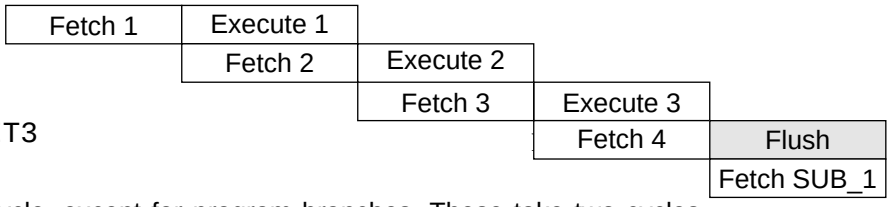
In the execution cycle, the fetched instruction is latched into the “Instruction Register (IR)” in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

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 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.

NOTES:

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

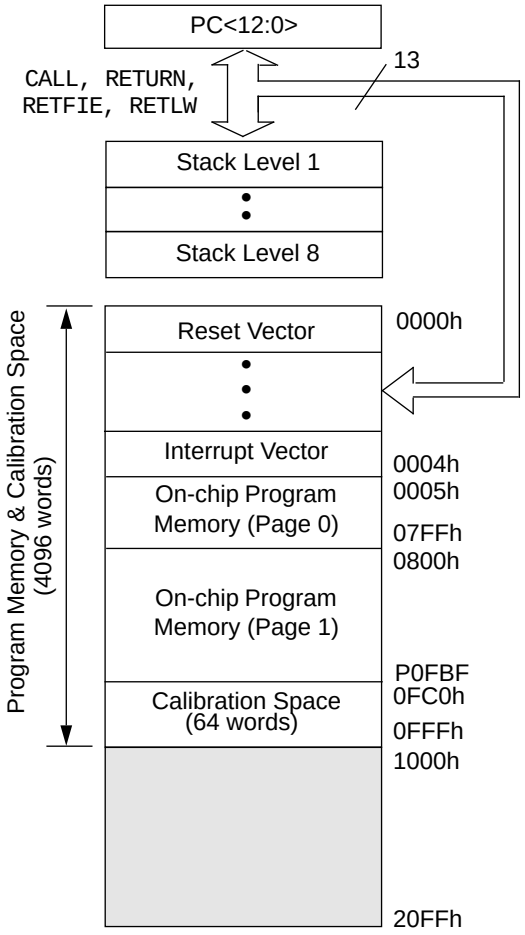
The PIC14000 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 4K x 14 (0000-0FFFh) are physically implemented. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1).

The 4096 words of Program Memory space are divided into:

- Address Vectors (addr 0000h-0004h)
- Program Memory Page 0 (addr 0005h-07FFH)
- Program Memory Page 1 (addr 0800h-0FBFh)
- Calibration Space (64 words, addr 0FC0h-0FFFh)

Program code may reside in Page 0 and Page 1.

FIGURE 4-1: PIC14000 PROGRAM MEMORY MAP AND STACK



P

TABLE 4-2: CALIBRATION CONSTANT ADDRESSES

Address	Data
0FC0h	K_{REF} , exponent
0FC1h	K_{REF} , mantissa high byte
0FC2h	K_{REF} , mantissa middle byte
0FC3h	K_{REF} , mantissa low byte
0FC4h	K_{BG} , exponent
0FC5h	K_{BG} , mantissa high byte
0FC6h	K_{BG} , mantissa middle byte
0FC7h	K_{BG} , mantissa low byte
0FC8h	V_{THERM} , exponent
0FC9h	V_{THERM} , mantissa high byte
0FCAh	V_{THERM} , mantissa middle byte
0FCBh	V_{THERM} , mantissa low byte
0FCCh	K_{TC} , exponent
0FCDh	K_{TC} , mantissa high byte
0FCEh	K_{TC} , mantissa middle byte
0FCFh	K_{TC} , mantissa low byte
0FD0h	F_{OSC}, unsigned byte
0FD1h	reserved
0FD2h	T_{WDT}, unsigned byte
0FD3h - 0FF8h	reserved
0FF9h-Fh	calibration space checksums

4.2 Data Memory Organization

The data memory (Figure 4-2) 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 first 32 locations of each bank are reserved for the Special Function Registers. Several Special Function Registers are mapped in both Bank 0 and Bank 1. The general purpose registers, implemented as static RAM, are located from address 20h through 7Fh, and A0 through FF.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly, or indirectly through the file select register FSR (Section 4.4).

FIGURE 4-2: REGISTER FILE MAP

File Address			
00h	Indirect add.(*)	Indirect addr.(*)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	RESERVED	RESERVED	86h
07h	PORTC	TRISC	87h
08h	PORTD	TRISD	88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh	ADTMRL	PCON	8Eh
0Fh	ADTMRH	SLPCON	8Fh
10h			90h
11h			91h
12h			92h
13h	I^2CBUF	I^2CADD	93h
14h	I^2CCON	I^2CSTAT	94h
15h	ADCAPL		95h
16h	ADCAPH		96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh		PREFA	9Bh
1Ch		PREFB	9Ch
1Dh		CMCON	9Dh
1Eh		MISC	9Eh
1Fh	ADCON0	ADCON1	9Fh
20h	General Purpose Register (96 Bytes)	General Purpose Register (96 Bytes)	A0h
7F			FF

* Not a physical register.
Shaded areas are unimplemented memory locations, read as ‘0’s.

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 (Table 4-3). These registers are static RAM.

The special registers are classified into two sets. Special registers associated with the “core” functions are described in this section. Those registers related to the operation of the peripheral features are described in the section specific to that peripheral.

TABLE 4-3: SPECIAL FUNCTION REGISTERS FOR THE PIC14000

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank0									
00h*	INDF (Indirect Address)	Addressing this location uses contents of the FSR to address data memory (not a physical register).							
01h	TMR0	Timer0 data							
02h*	PCL	Program Counter’s (PC’s) least significant byte							
03h*	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04h*	FSR	Indirect data memory address pointer							
05h	PORTA	PORTA data latch.							
06h	Reserved	Reserved for emulation.							
07h	PORTC	PORTC data latch							
08h	PORTD	PORTD data latch							
09h	Reserved								
0Ah*	PCLATH	Buffered register for the upper 5 bits of the Program Counter (PC)							
0Bh*	INTCON	GIE	PEIE	T0IE	r	r	T0IF	r	r
0Ch	PIR1	CMIF	—	—	PBIF	I^2CIF	RCIF	ADCIF	OVFIF
0Dh	Reserved								
0Eh	ADTMRL	A/D capture timer data least significant byte							
0Fh	ADTMRH	A/D capture timer data most significant byte							
10h	Reserved								
11h	Reserved								
12h	Reserved								
13h	I^2CBUF	I^2C Serial Port Receive Buffer/Transmit Register							
14h	I^2CCON	WCOL	I^2COV	I^2CEN	CKP	I^2CM3	I^2CM2	I^2CM1	I^2CM0
15h	ADCAPL	A/D capture latch least significant byte							
16h	ADCAPH	A/D capture latch most significant byte							
17h	Reserved								
18h	Reserved								
19h	Reserved								
1Ah	Reserved								
1Bh	Reserved								
1Ch	Reserved								
1Dh	Reserved								
1Eh	Reserved								
1Fh	ADCON0	ADCS3	ADCS2	ADCS1	ADCS0	—	AMUXOE	ADRST	ADZERO

Legend
— = unimplemented bits, read as ‘0’ but cannot be overwritten
r = reserved bits, default is POR value and **should not be overwritten with any value**
Reserved indicates reserved register and **should not be overwritten with any value**
* indicates registers that can be addressed from either bank

TABLE 4-3: SPECIAL FUNCTION REGISTERS FOR THE PIC14000 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank1									
80h*	INDF (Indirect Address)	Addressing this location uses contents of FSR to address data memory (not a physical register).							
81h	OPTION	$\overline{RCPU}$	r	TOCS	TOSE	PSA	PS2	PS1	PS0
82h*	PCL	Program Counter's (PC's) least significant byte							
83h*	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
84h*	FSR	Indirect data memory address pointer							
85h	TRISA	PORTA Data Direction Register							
86h	Reserved	Reserved for emulation							
87h	TRISC	PORTC Data Direction Register							
88h	TRISD	PORTD Data Direction Register							
89h	Reserved								
8Ah*	PCLATH	Buffered register for the upper 5 bits of the Program Counter (PC)							
8Bh*	INTCON	GIE	PEIE	T0IE	r	r	T0IF	r	r
8Ch	PIE1	CMIE	—	—	PBIE	I^2CIE	RCIE	ADCIE	OVFIE
8Dh	Reserved								
8Eh	PCON	r	—	—	—	—	—	$\overline{POR}$	$\overline{LVD}$
8Fh	SLPCON	HIBEN	—	REFOFF	LSOFF	OSCOFF	CMOFF	TEMPOFF	ADOFF
90h	Reserved								
91h	Reserved								
92h	Reserved								
93h	I^2CADD	I^2C Synchronous Serial Port Address Register							
94h	I^2CSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
95h	Reserved								
96h	Reserved								
97h	Reserved								
98h	Reserved								
99h	Reserved								
9Ah	Reserved								
9Bh	PREFA	PRA7	PRA6	PRA5	PRA4	PRA3	PRA2	PRA1	PRA0
9Ch	PREFB	PRB7	PRB6	PRB5	PRB4	PRB3	PRB2	PRB1	PRB0
9Dh	CMCON	—	CMBOUT	CMBOE	CPOLB	—	CMAOUT	CMAOE	CPOLA
9Eh	MISC	SMHOG	SPGNDB	SPGNDA	I^2CSEL	SMBUS	INCLKEN	OSC2	OSC1
9Fh	ADCON1	ADDAC3	ADDAC2	ADDAC1	ADDAC0	PCFG3	PCFG2	PCFG1	PCFG0

Legend
— = unimplemented bits, read as '0' but cannot be overwritten
r = reserved bits, default is POR value and **should not be overwritten with any value**
Reserved indicates reserved register and **should not be overwritten with any value**
* indicates registers that can be addressed from either bank

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 4-3, 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, as with 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 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).

It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. 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 PIC14000 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 $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

FIGURE 4-3: STATUS REGISTER

83h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
Read/Write	R/W	R/W	R/W	R	R	R/W	R/W	R/W
POR value FFh	0	0	0	1	1	X	X	X

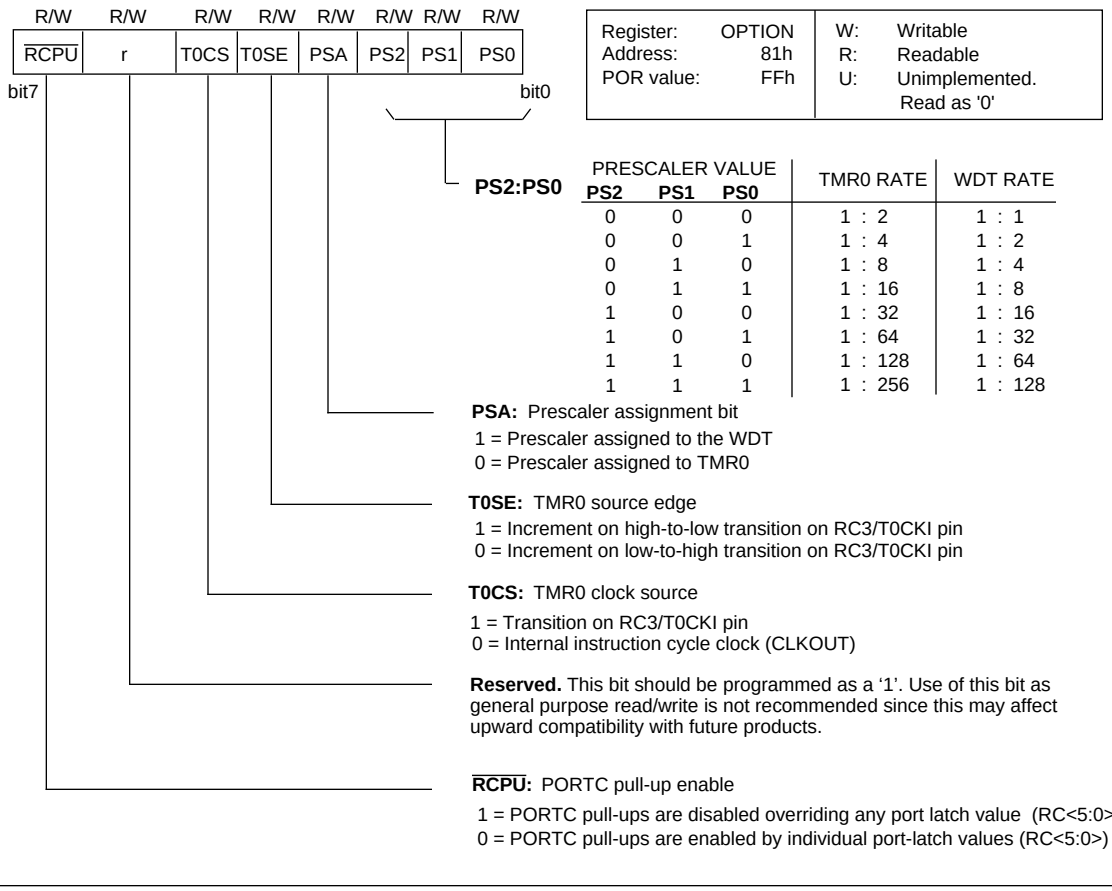
Bit	Name	Function
B7	IRP	Not used. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.
B6	RP1	Not used. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.
B5	RP0	Register page select for direct addressing. 1 = Bank1 (80h - FFh) 0 = Bank0 (00h - 7Fh) Each page is 128 bytes. Only the RP0 bit is used.
B4	$\overline{TO}$	Time-out bit. 1 = After power-up and by the CLRWDT and SLEEP instruction. 0 = A watchdog timer time-out has occurred.
B3	$\overline{PD}$	Power down bit. 1 = After power-up or by a CLRWDT instruction. 0 = By execution of the SLEEP instruction.
B2	Z	Zero bit. 1 = The result of an arithmetic or logic operation is zero. 0 = The result of an arithmetic or logical operation is not zero.
B1	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. 0 = No carry-out from the 4th low order bit of the result. Note: For Borrow, the polarity is reversed.
B0	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 (RRF, RLF) 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.

4.2.2.2 OPTION REGISTER

The OPTION register (Address 81h) is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, TMR0, and the weak pull-ups on PORTC<5:0>. Bit 6 is reserved.

Note: To achieve a 1:1 prescaler assignment, assign the prescaler to the WDT (PSA=1)

FIGURE 4-4: OPTION REGISTER

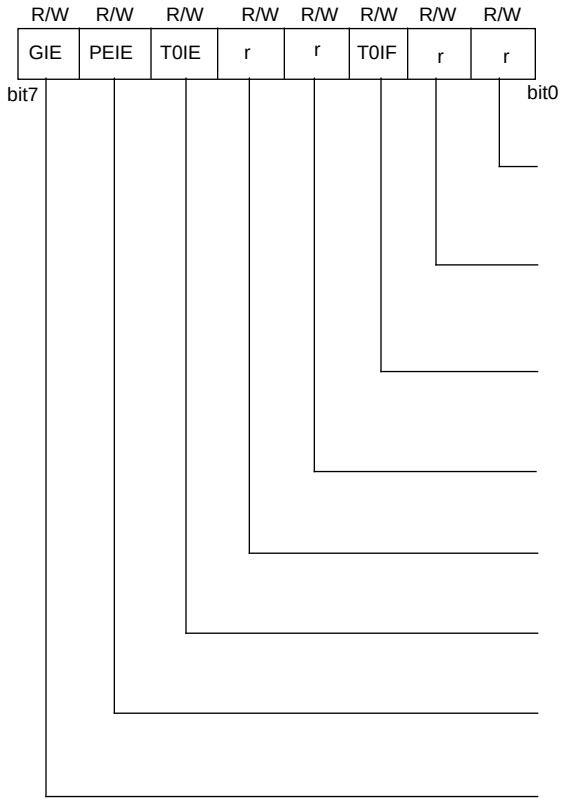


4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains the various enable and flag bits for the Timer0 overflow and peripheral interrupts. Figure 4-5 shows the bits for the INTCON register.

Note:	The T0IF 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 interrupt, clear the interrupt flag, to ensure that the program does not immediately branch to the peripheral interrupt service routine

FIGURE 4-5: INTCON REGISTER



Register:	INTCON	W:	Writable
Address:	0Bh or 8Bh	R:	Readable
POR value:	0000 000xb	U:	Unimplemented, read as '0'

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

T0IF: TMR0 overflow interrupt flag

1 = The TMR0 has overflowed
Must be cleared by software
0 = TMR0 did not overflow

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

T0IE: TMR0 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

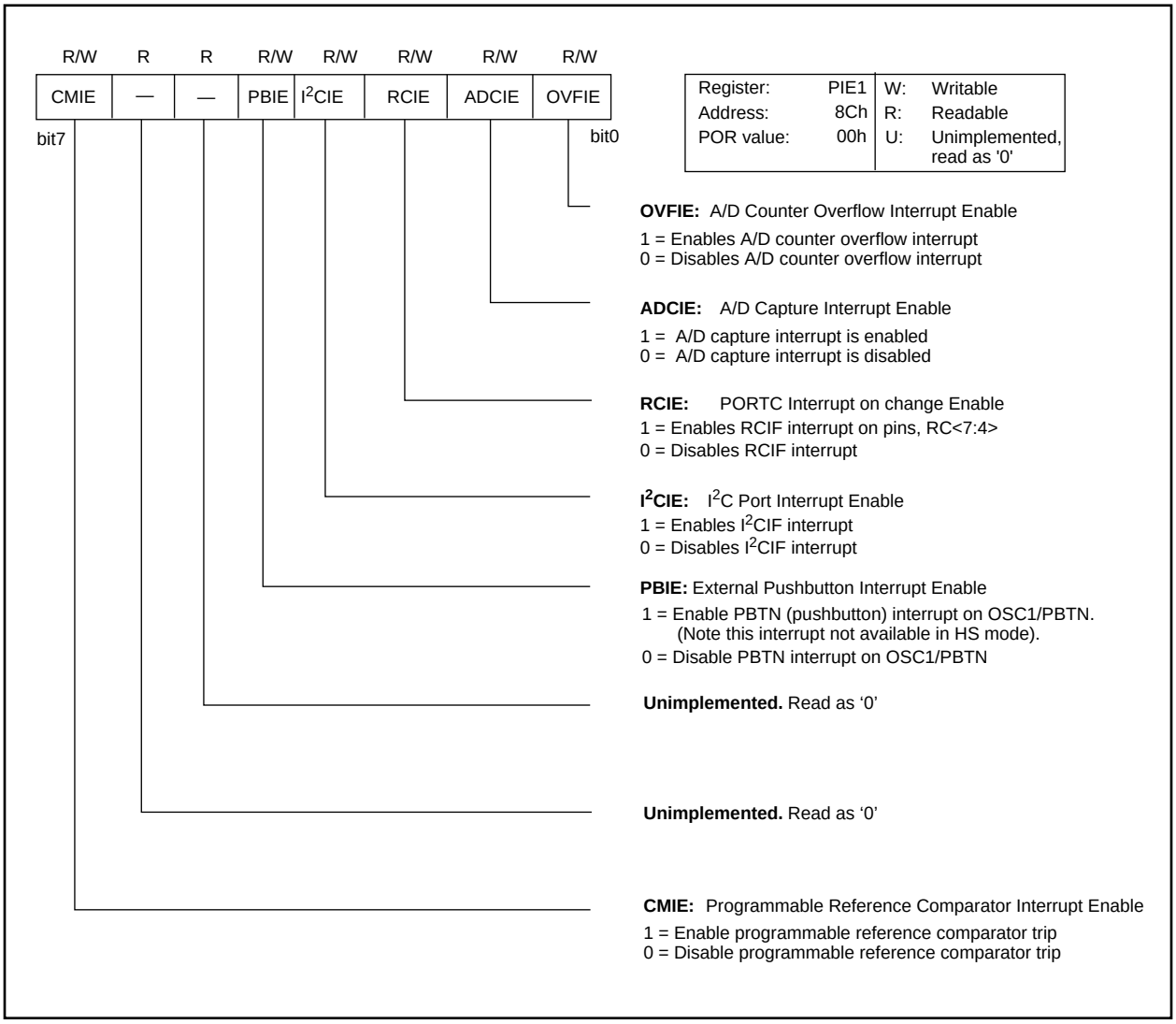
1 = Enables all un-masked interrupts
0 = Disables all interrupts

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the Peripheral interrupts including A/D capture event, I^2C serial port, PORTC change and A/D capture timer overflow, and external push button.

Note: INTCON<6> must be enabled to enable any interrupt in PIE1.

FIGURE 4-6: PIE1 REGISTER

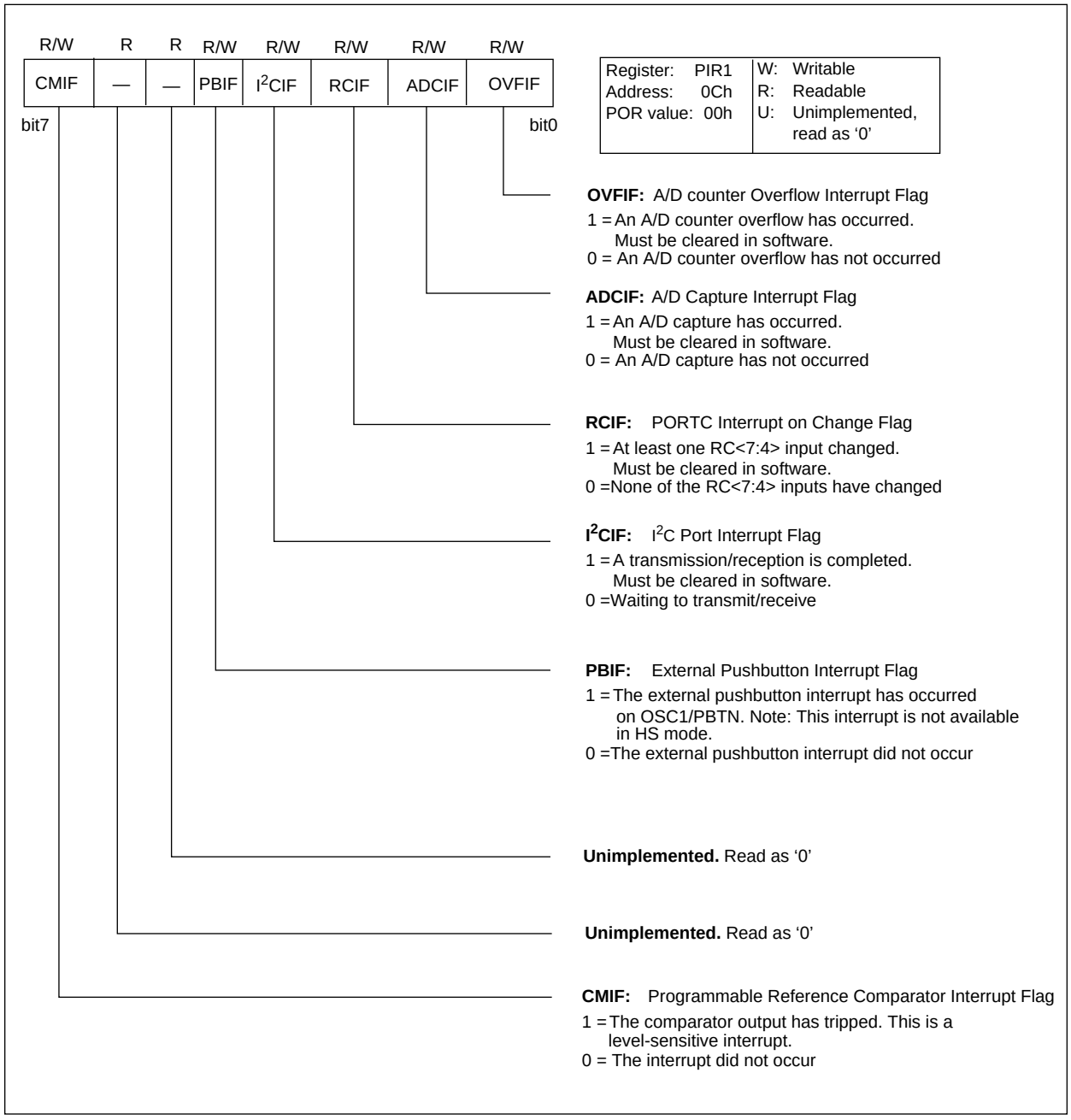


4.2.2.5 PIR1 REGISTER

This register contains the individual flag bits for the Peripheral interrupts (Figure 4-7).

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 4-7: PIR1 REGISTER



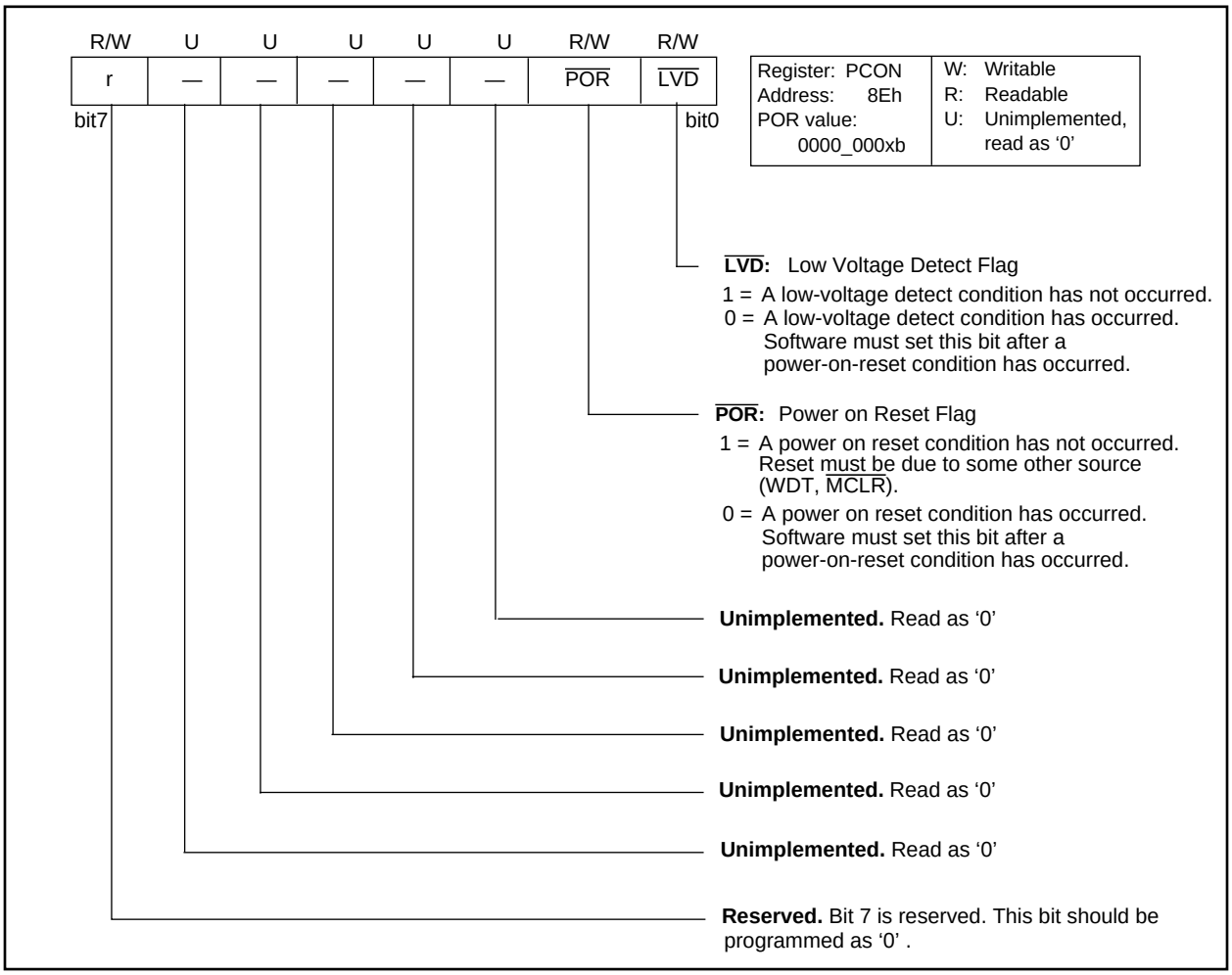
4.2.2.6 PCON REGISTER

The Power Control (PCON) register status contains 2 flag bits to allow differentiation between a Power-on Reset, an external $\overline{MCLR}$ reset, WDT reset, or low-voltage condition (Figure 4-8).

These bits are cleared on POR. The user must set these bits following POR. On a subsequent reset if POR is cleared, this is an indication that the reset was due to a power-on reset condition.

Note: $\overline{LVD}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{LVD}$ is cleared, indicating a low voltage condition has occurred.

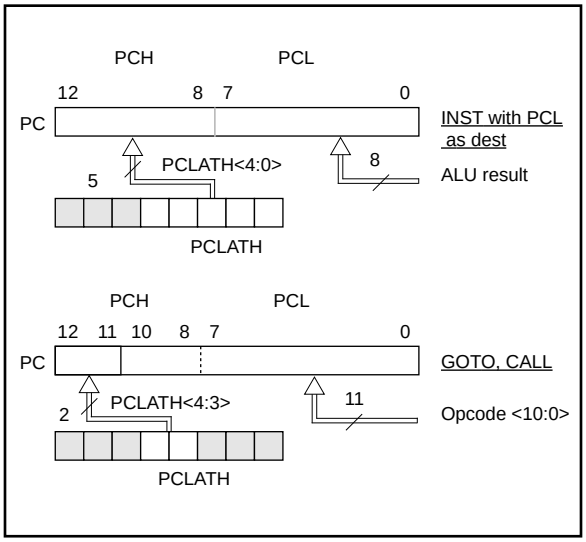
FIGURE 4-8: PCON REGISTER



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. PCLATH is a holding register for PC<12:8> where contents are transferred to the upper byte of the program counter. When 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 4-9.

FIGURE 4-9: 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 Read Using the PIC16CXX”(AN556).

4.3.2 STACK

The PIC14000 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 PC is PUSHed in 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. 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 overflow or stack underflow conditions.

Note 2: There are no instruction mnemonics called PUSH nor POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, or RETFIE instructions, or the vectoring to an interrupt address

4.3.3 PROGRAM MEMORY PAGING

The PIC14000 has 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 (Figure 4-9). 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 PIC14000 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.

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).

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.

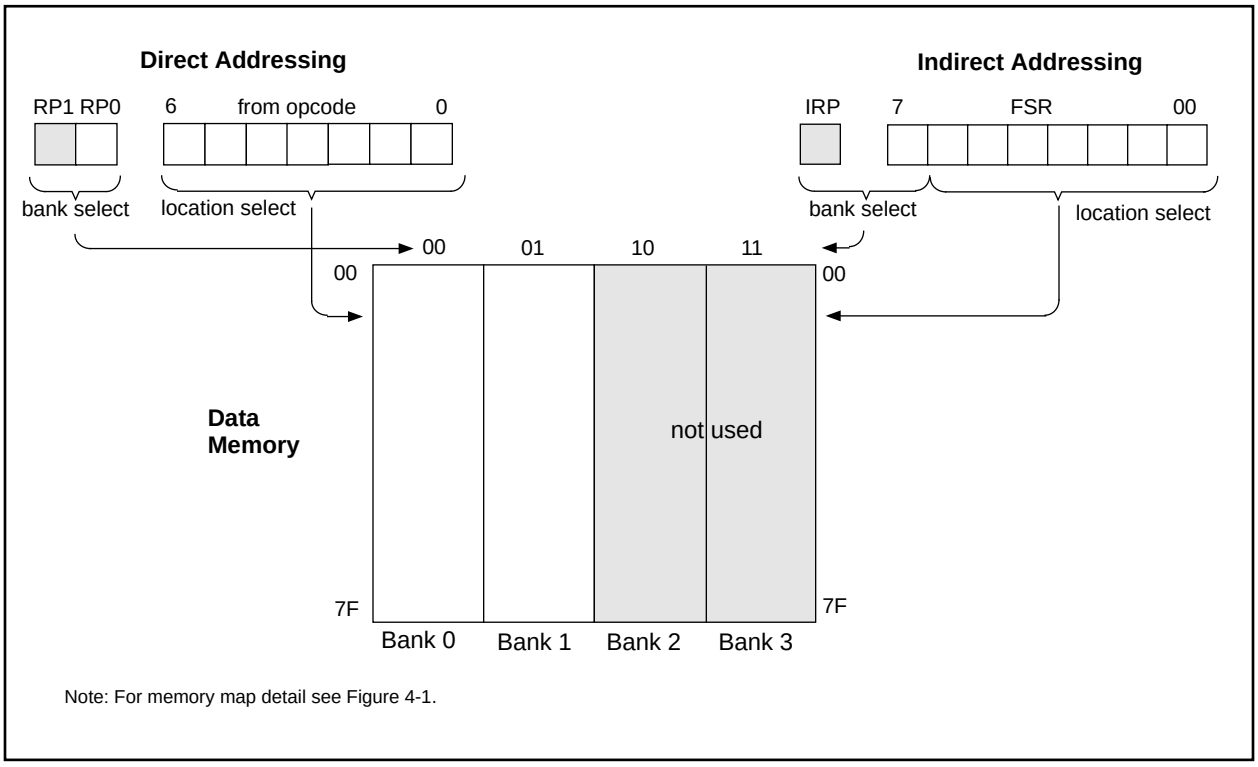
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status 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 4-10. However, IRP is not used in the PIC14000.

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
        incf    FSR     ;inc pointer
        btfss   FSR,4   ;all done?
        goto    NEXT    ;no clear next
                        ;yes continue
CONTINUE:
```

FIGURE 4-10: INDIRECT/INDIRECT ADDRESSING



Note: For memory map detail see Figure 4-1.

5.0 I/O PORTS

The PIC14000 has three ports, PORTA, PORTC and PORTD, described in the following paragraphs. Generally, PORTA is used as the analog input port. PORTC is used for general purpose I/O and for host communication. PORTD provides additional I/O lines. Four lines of PORTD may function as analog inputs.

5.1 PORTA and TRISA

PORTA is a 4-bit wide port with data register located at location 05h and corresponding data direction register (TRISA) at 85h. PORTA can operate as either analog inputs for the internal A/D converter or as general purpose digital I/O ports. These inputs are Schmitt Triggers when used as digital inputs, and have CMOS drivers as outputs.

PORTA pins are multiplexed with analog inputs. ADCON1<1:0> bits control whether these pins are analog or digital as shown in Section 8.7. When configured to the digital mode, reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. When selected as an analog input, these pins will read as '0's.

Note:	On Reset, PORTA is configured as analog inputs

The TRISA register controls the direction of the PORTA 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. A '1' in each location configures the corresponding port pin as an input. This register resets to all '1's, meaning all PORTA pins are initially inputs. The data register should be initialized prior to configuring the port as outputs. See Figure 5-2 and Figure 5-3.

PORTA inputs go through a Schmitt Trigger AND gate that is disabled when the input is in analog mode. Refer to Figure 5-1.

Note that bits RA<7:4> are unimplemented and always read as '0'. Unused inputs should not be left floating to avoid leakage currents. All pins have input 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  0x0F         ;Value used to initialize
                    ;data direction
MOVWF  TRISA        ;Set RA<3:0> as inputs
```

FIGURE 5-1: PORTA BLOCK DIAGRAM

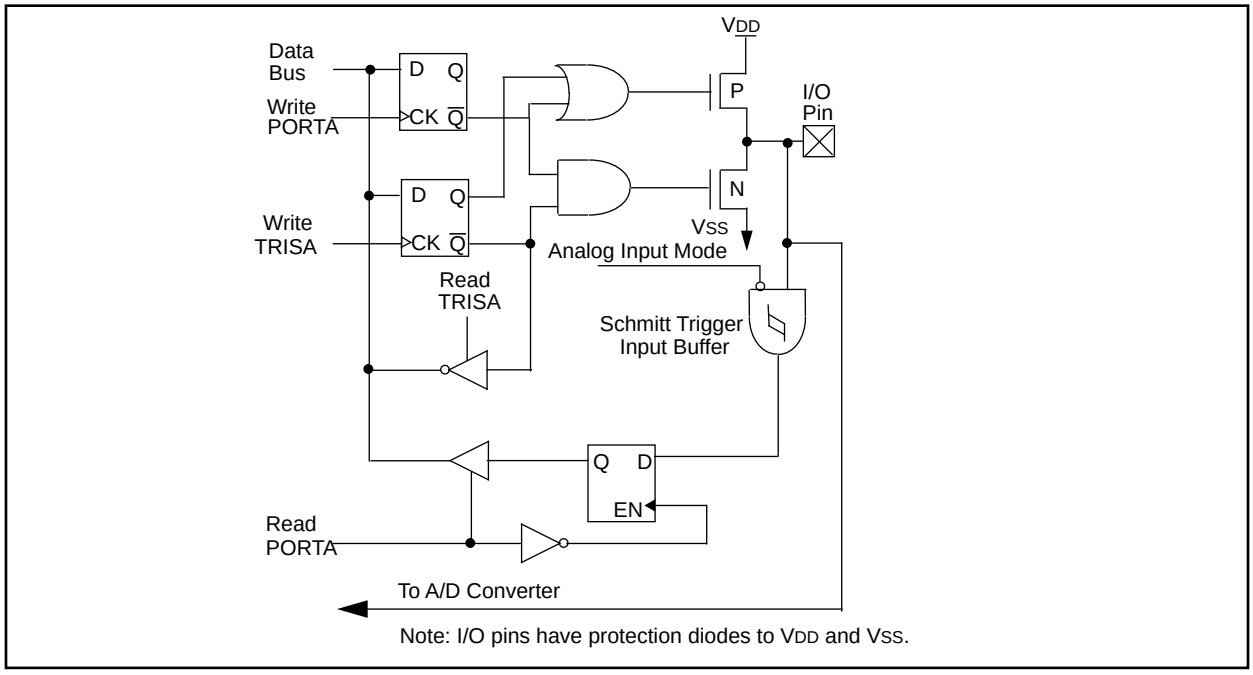


FIGURE 5-2: PORTA DATA REGISTER

05h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	—	—	—	—	RA3/AN3	RA2/AN2	RA1/AN1	RA0/AN0
Read/Write	U	U	U	U	R/W	R/W	R/W	R/W
POR value 0xh	0	0	0	0	X	X	X	X

Bit	Name	Function
B7-B4	—	Unimplemented. Reads as‘0’.
B3	RA3/AN3	GPIO or analog input. Returns value on pin RA3/AN3 when used as a digital input. When configured as an analog input, reads as ‘0’.
B2	RA2/AN2	GPIO or analog input. Returns value on pin RA2/AN2 when used as a digital input. When configured as an analog input, reads as ‘0’.
B1	RA1/AN1	GPIO or analog input. Returns value on RA1/AN1 when used as a digital input. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. When configured as an analog input, reads as ‘0’.
B0	RA0/AN0	GPIO or analog input. Returns value on pin RA0/AN0 when used as a digital input. When configured as an analog input, reads as ‘0’.

5.2 PORTC and TRISC

PORTC is a 8-bit wide bidirectional port, with Schmitt Trigger inputs, that serves the following functions depending on programming:

- Direct LED drive (PORTC<7:0>).
- I^2C communication lines (PORTC<7:6>), refer to Section 7.0 I^2C Serial Port.
- Interrupt on change function (PORTC<7:4>), discussed below and in Section 10.3 Interrupts.
- Programmable reference and comparator outputs.
- Timer0 clock source on RC3

The PORTC data register is located at location 07h and its data direction register (TRISC) is at 87h.

PORTC<5:0> have weak internal pull-ups (~100 uA typical). A single control bit can turn on all the pull-ups. This is done by clearing bit $\overline{RCPU}$ (OPTION<7>). 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 and in hibernate mode.

When using PORTC<0> as an analog output (CMCON<1> bit is set), the TRISC<0> bit should be cleared to disable the weak pull-up on this pin. Refer to Table 5-1.

Four of the PORTC pins, RC<7:4> have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur. In other words, any pin RC<7:4> configured as an output is excluded from the interrupt on change comparison. The input pins of RC<7:4> are compared with the old value latched on the last read of PORTC. The “mismatch” outputs of RC<7:4> are OR’ed together to assert the RCIF flag (PIR1 register<2>) and cause a CPU interrupt, if enabled.

Note:	If the I^2C function is enabled, (I^2CCON<5>, address 14h), RC<7:6> are automatically excluded from the interrupt-on-change comparison.

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:

- Disable the interrupt by clearing the RCIE (PIE1<2>) bit
- Read PORTC. This will end mismatch condition. Then, clear the RCIF (PIR1<2>) bit.

A mismatch condition will continue to set the RCIF bit. Reading PORTC will end the mismatch condition, and allow the RCIF bit to be cleared.

If bit CMAOE (CMCON<1>) is set, the RC0/REFA pin becomes the programmable reference A and analog output. Pin RC1/CMPA becomes the comparator A output.

Note:	Setting CMAOE changes the definition of RC0/REFA and RC1/CMPA, bypassing the PORTC data and TRISC register settings.

PORTC<7:6> also serves multiple functions. These pins act as the I^2C data and clock lines when the I^2C module is enabled. They also serve as the serial programming interface data and clock line for in-circuit programming of the EPROM.

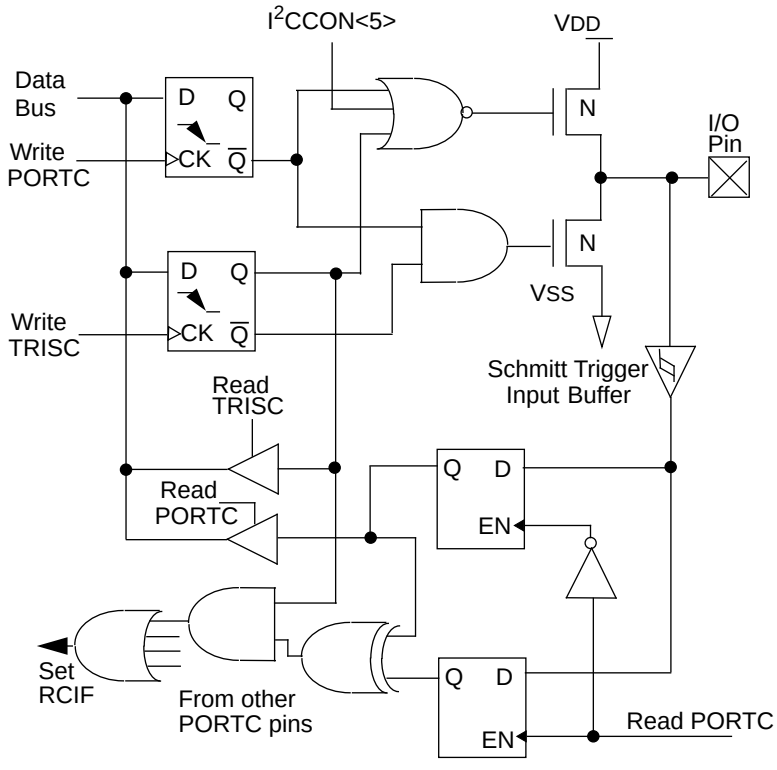
The TRISC register controls the direction of the PORTC pin. A '1' in each location configures the corresponding port pin as an input. Upon reset, this register sets to FFh, meaning all PORTC pins are initially inputs. The data register should be initialized prior to configuring the port as outputs.

Unused inputs should not be left floating to avoid leakage currents. All pins have input protection diodes to VDD and VSS.

EXAMPLE 5-2: 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 5-3: BLOCK DIAGRAM OF PORTC<7:6> PINS

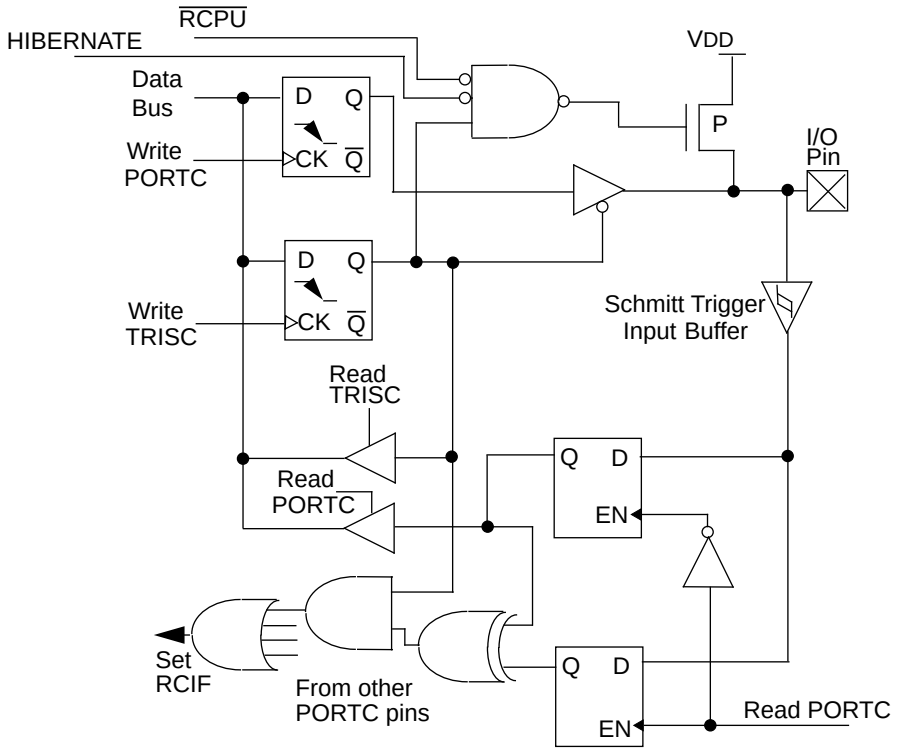


Note: I/O pins have protection diodes to VDD and VSS. These pins do not have a P-channel pull-up.

TABLE 5-1: PORT RC0 PIN CONFIGURATION SUMMARY

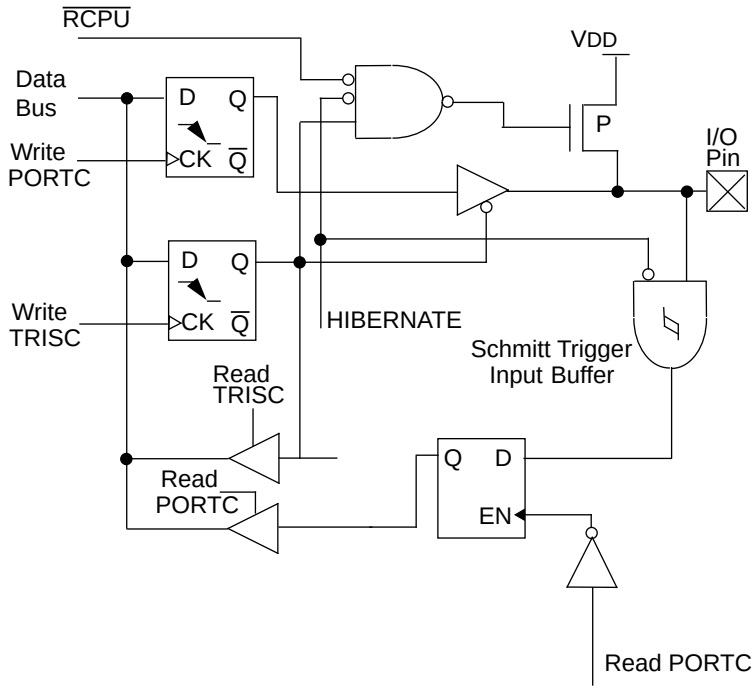
RC0 Pin Configuration	TRISC<0>	$\overline{\text{RCPU}}$ OPTION<7>	CMAOE CMCON<1>	Comment
Digital Input (weak pull-up)	1	0	0	
Digital Input (no pull-up)	1	1	0	
Digital Output	0	X	0	
Analog Output	0	X	1	Must clear TRISC<0> to disable pull-up when used as an analog output.

FIGURE 5-4: BLOCK DIAGRAM OF PORTC<5:4> PINS



1. I/O pins have protection diodes to VDD and VSS.
2. Port Latch = ‘1’ and TRISC = ‘1’ enables weak pull-up if $\overline{\text{RCPU}}$ = ‘0’ in OPTION register.

FIGURE 5-5: BLOCK DIAGRAM OF PORTC<3:0> PINS



1. I/O pins have protection diodes to V_{DD} and V_{SS}.
2. Port Latch ='1' and TRISC ='1' enables weak pull-up if $\overline{RCPU}$ ='0' in OPTION register.
3. If the CMAOE bit (CMCON<1>) is set to'1', RC0 becomes REFA, RC1 becomes CMPA, ignoring the PORTC<1:0> data and TRISC<1:0> register settings.

FIGURE 5-6: PORTC DATA REGISTER

07h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTC	RC7/SDAA	RC6/SCLA	RC5	RC4	RC3/T0CKI	RC2	RC1/CMPA	RC0/REFA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value xxh	x	x	x	x	x	x	x	x

Bit	Name	Function
B7	RC7/SDAA	Synchronous serial data I/O for I^2C interface. Also is the serial programming data line. This pin can also serve as a general purpose I/O. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up to VDD which is disabled in I^2C mode.
B6	RC6/SCLA	Synchronous serial clock for I^2C interface. Also is the serial programming clock. This pin can also serve as a general purpose I/O. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up to VDD which is disabled in I^2C mode.
B5	RC5	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
B4	RC4	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
B3	RC3/T0CKI	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD. T0CKI is enabled as TMR0 clock via the OPTION register.
B2	RC2	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD.
B1	RC1/CMPA	LED direct-drive output. This pin can also serve as a GPIO, or comparator A output. If enabled, this pin has a weak internal pull-up to VDD.
B0	RC0/REFA	LED direct-drive output. This pin can also serve as a GPIO, or programmable reference A output. If enabled, this pin has a weak internal pull-up to VDD.

U= unimplemented, X = unknown.

5.2.1 TRISC PORTC DATA DIRECTION REGISTER

This register defines each pin of PORTC as either an input or output under software control. A '1' in each location configures the corresponding port pin as an input. This register resets to all '1's, meaning all PORTC pins are initially inputs. The data register should be initialized prior to configuring the port as outputs.

FIGURE 5-7: TRISC REGISTER

87h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value FFh	1	1	1	1	1	1	1	1

Bit	Name	Function
B7	TRISC7	Control direction on pin RC7/SDAA (has no effect if I^2C is enabled): 0 = pin is an output 1 = pin is an input
B6	TRISC6	Control direction on pin RC6/SCLA (has no effect if I^2C is enabled): 0 = pin is an output 1 = pin is an input
B5	TRISC5	Control direction on pin RC5: 0 = pin is an output 1 = pin is an input
B4	TRISC4	Control direction on pin RC4: 0 = pin is an output 1 = pin is an input
B3	TRISC3	Control direction on pin RC3: 0 = pin is an output 1 = pin is an input
B2	TRISC2	Control direction on pin RC2: 0 = pin is an output 1 = pin is an input
B1	TRISC1	Control direction on pin RC1/CMPA (has no effect if the CMAOE bit is set): 0 = pin is an output 1 = pin is an input
B0	TRISC0	Control direction on pin RC0/REFA (has no effect if the CMAOE bit is set): 0 = pin is an output 1 = pin is an input

U= unimplemented, X = unknown.

5.3 PORTD and TRISD

PORTD is an 8-bit port that may be used for general purpose I/O. Four pins can be configured as analog inputs.

FIGURE 5-8: BLOCK DIAGRAM OF PORTD<7:4> PINS

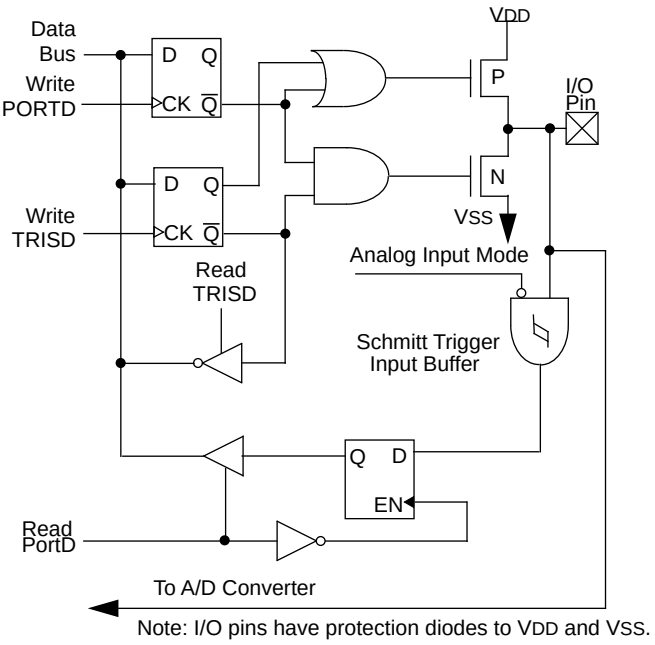
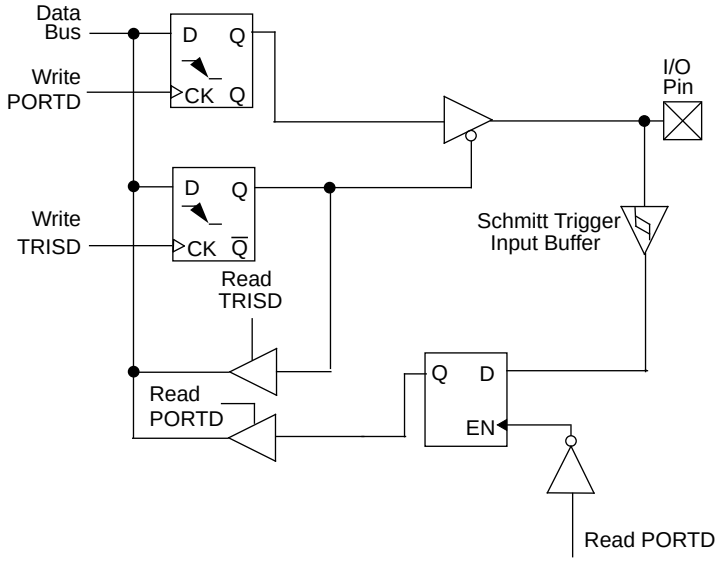


FIGURE 5-9: BLOCK DIAGRAM OF PORTD<3:2> PINS



1. I/O pins have protection diodes to VDD and VSS.
2. If CMBOE (CMCON<5>) is set to '1', RD2 becomes CMPB, RD3 becomes REFB, ignoring the PORTD<3:2> data and TRISD<3:2> register settings.

FIGURE 5-10: BLOCK DIAGRAM OF PORTD<1:0> PINS

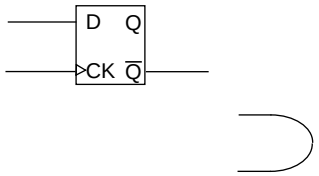


FIGURE 5-11: PORTD DATA REGISTER

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTD	RD7/AN7	RD6/AN6	RD5/AN5	RD4/AN4	RD3/REFB	RD2/CMPB	RD1/SDAB	RD0/SCLB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value xxh	X	X	X	X	X	X	X	X

Bit	Name	Function
B7	RD7/AN7	GPIO or analog input. Returns value on pin RD7/AN7 when used as a digital input. When configured as an analog input, reads as '0'.
B6	RD6/AN6	GPIO or analog input. Returns value on pin RD6/AN6 when used as a digital input. When configured as an analog input, reads as '0'.
B5	RD5/AN5	GPIO or analog input. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. When configured as an analog input, reads as '0'.
B4	RD4/AN4	GPIO or analog input. Returns value on pin RD4/AN4 when used as a digital input. When configured as an analog input, reads as '0'.
B3	RD3/REFB	This pin can serve as a GPIO, or programmable reference B output.
B2	RD2/CMPB	This pin can serve as a GPIO, or comparator B output.
B1	RD1/SDAB	Alternate synchronous serial data I/O for I^2C interface enabled by setting the I^2CSEL bit in the MISC register. This pin can also serve as a general purpose I/O. This pin has an N-channel pull-up to VDD which is disabled in I^2C mode.
B0	RD0/SCLB	Alternate synchronous serial clock for I^2C interface, enabled by setting the I^2CSEL bit in the MISC register. This pin can also serve as a general purpose I/O. This pin has an N-Channel pull-up to VDD which is disabled in I^2C mode.

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

FIGURE 5-12: TRISD REGISTER

88h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value FFh	1	1	1	1	1	1	1	1

Bit	Name	Function
B7	TRISD7	Control direction on pin RD7/AN7: 0 = pin is an output 1 = pin is an input
B6	TRISD6	Control direction on pin RD6/AN6: 0 = pin is an output 1 = pin is an input
B5	TRISD5	Control direction on pin RD5/AN5: 0 = pin is an output 1 = pin is an input
B4	TRISD4	Control direction on pin RD4/AN4: 0 = pin is an output 1 = pin is an input
B3	TRISD3	Control direction on pin RD3/REFB (has no effect if the CMBOE bit is set): 0 = pin is an output 1 = pin is an input
B2	TRISD2	Control direction on pin RD2/CMPB (has no effect if the CMBOE bit is set): 0 = pin is an output 1 = pin is an input
B1	TRISD1	Control direction on pin RD1/SDAB: 0 = pin is an output 1 = pin is an input
B0	TRISD0	Control direction on pin RD0/SCLB: 0 = pin is an output 1 = pin is an input

If the CMBOE bit (CMCON<5>) is set, the RD3/REFB pin becomes the programmable reference B output and pin RD2/CMPB becomes the comparator B output.

Note:	Setting CMBOE changes the definition of RD3/REFB and RD2/CMPB, bypassing the PORTD data and TRISD register settings.

PORTD<1:0> also serve multiple functions. These pins act as the I^2C data and clock lines when the I^2C module is enabled.

The TRISD register controls the direction of the Port D pins. A '1' in each location configures the corresponding port pin as an input. Upon reset, this register sets to FFh, meaning all PORTD pins are initially inputs. The data register should be initialized prior to configuring the port as outputs.

Unused inputs should not be left floating to avoid leakage currents. All pins have input protection diodes to V_{DD} and V_{SS}.

EXAMPLE 5-3: INITIALIZING PORTD

```
CLRF   PORTD        ; Initialize PORTD data
                    ;    latches before setting
                    ;    the data direction
                    ;    register
BSF    STATUS, RP0  ; Select Bank1
MOVLW  0xFF         ; Value used to initialize
                    ; data direction
MOVWF  TRISD        ; Set RD<7:0> as inputs
```

5.4 I/O Programming Considerations

5.4.1 BI-DIRECTIONAL I/O PORTS

Reading the port register reads the values of the port pins. Writing to the port register writes the value to the port latch. Some instructions operate internally as read-modify-write. 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 bit5 of PORTC will cause all eight bits of PORTC to be read into the CPU. Then the BSF operation takes place on bit5 and PORTC is written to the output latches. If another bit of PORTC 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 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.

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 ("wire-or", "wire-and"). The resulting high output currents may damage the chip.

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:   PORTC<7:4> Inputs
;
;                          PORTC<3:0> Outputs
; PORTC<7:6> have external pull-up and are not
; connected to other circuitry
;
;                          PORT  latch  PORT pins
;                          ----------   ----------

   BCF  PORTC, 7          ; 01pp  pppp   11pp pppp
   BCF  PORTC, 6          ; 10pp  pppp   11pp pppp
   BSF  STATUS,RP0        ;
   BCF  TRISC, 7          ; 10pp  pppp   11pp pppp
   BCF  TRISC, 6          ; 10pp  pppp   10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RC7 to be latched as the pin value (High).
```

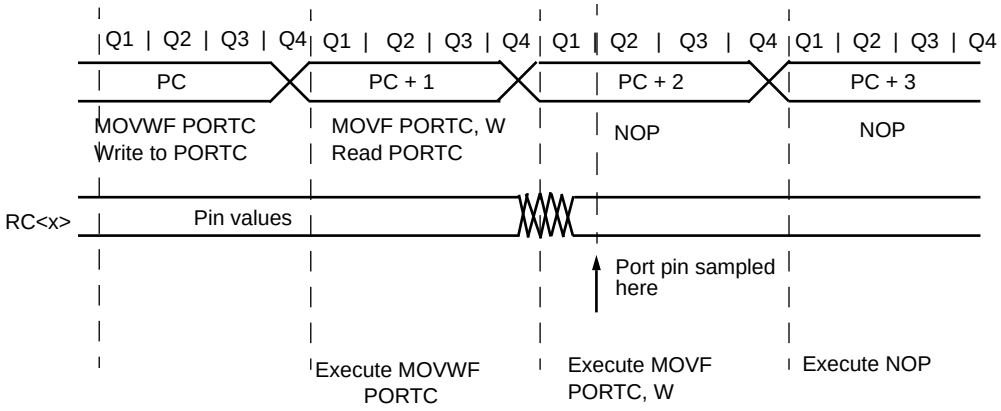
5.4.2 SUCCESSIVE OPERATIONS ON I/O PORTS

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. Therefore, care must be exercised if a write operation is followed by a read operation on the same I/O port.

The sequence of instructions should be such to allow the pin voltage to stabilize before the next instruction which causes that port 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 5-13: SUCCESSIVE I/O OPERATION

Example showing write to PORTC followed by immediate read. Some delays in settling may cause “old” Port data to be read, especially at higher clock frequencies. Data setup time = (0.25 Tcyc- Tpd), where Tcyc = instruction cycle time.



6.0 TIMER MODULES

The PIC14000 contains two general purpose timer modules, Timer0 (TMR0) and the Watchdog Timer (WDT). The ADTMR is described in the A/D section.

The Timer0 module is identical to the Timer0 module of the PIC16C7X enhanced core products. It is an 8-bit overflow counter.

The Timer0 module has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer (WDT). PSA (OPTION<3>) assigns the prescaler, and PS2:PS0 (OPTION<2:0>) determines the prescaler value. Timer0 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.

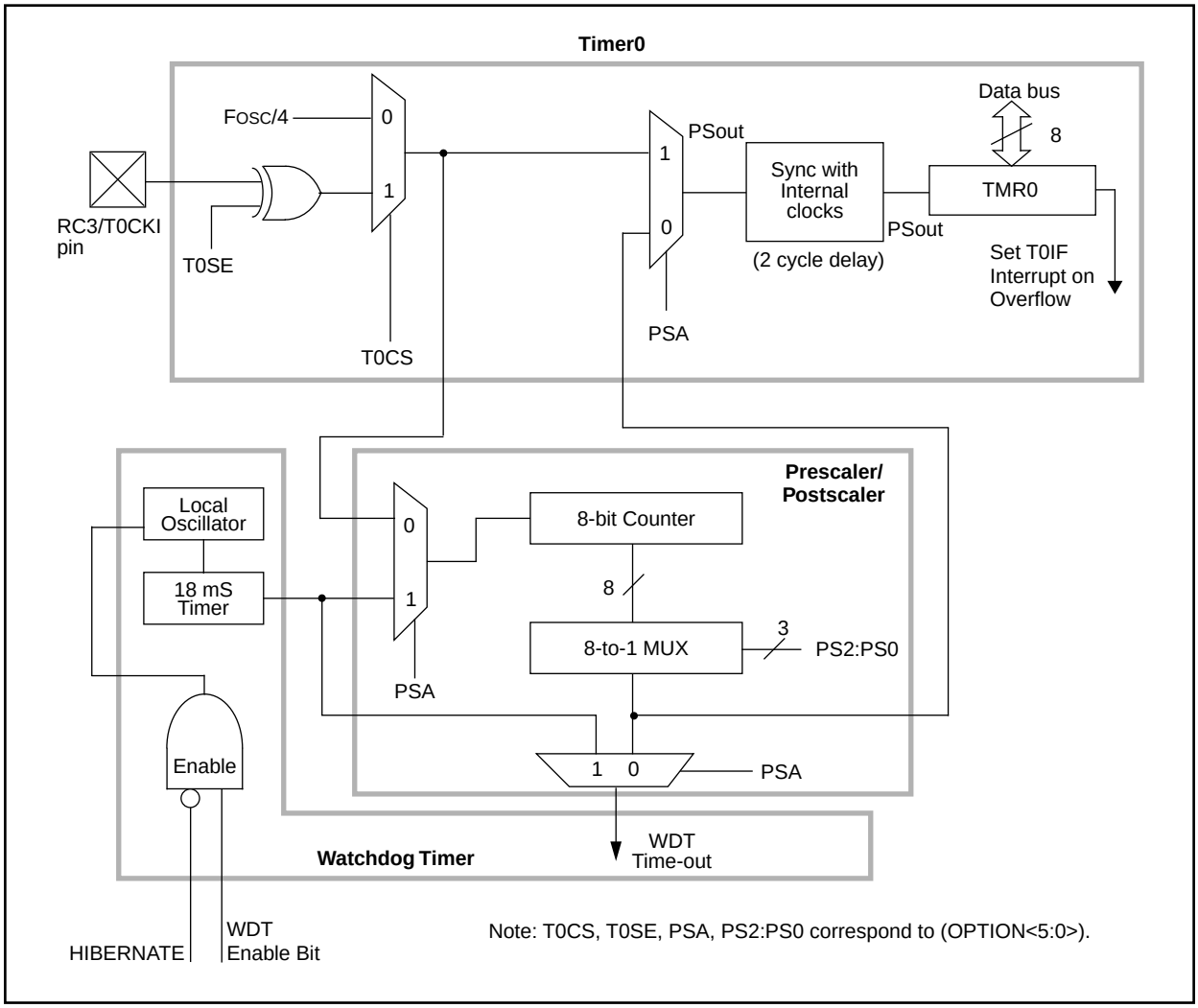
The Timer0 module has the following features:

- 8-bit timer
- Readable and writable (file address 01h)
- 8-bit software programmable prescaler
- Interrupt on overflow from FFh to 00h

Figure 6-1 is a simplified block diagram of the Timer0 module.

The Timer0 module will increment every instruction cycle (without prescaler). If TMR0 is written, increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can compensate by writing an adjusted value to TMR0.

FIGURE 6-1: TIMER0 AND WATCHDOG TIMER BLOCK DIAGRAM



6.1 Timer0 Interrupt

The TMR0 interrupt is generated when the Timer0 overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing bit T0IE (INTCON<5>). Flag bit T0IF (INTCON<2>) must be cleared in software by the TMR0 module interrupt service routine before re-enabling this interrupt. The Timer0 module interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. The timing of the Timer0 interrupt is shown in Figure 6-4.

FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE

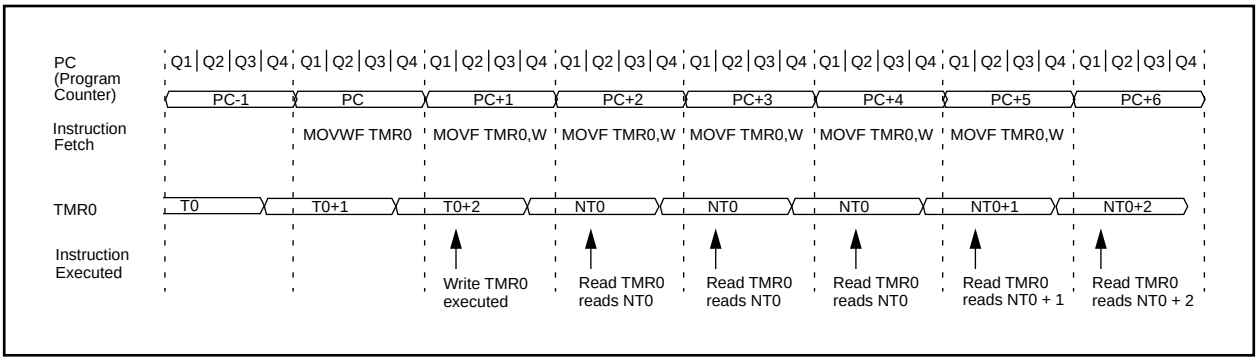


FIGURE 6-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

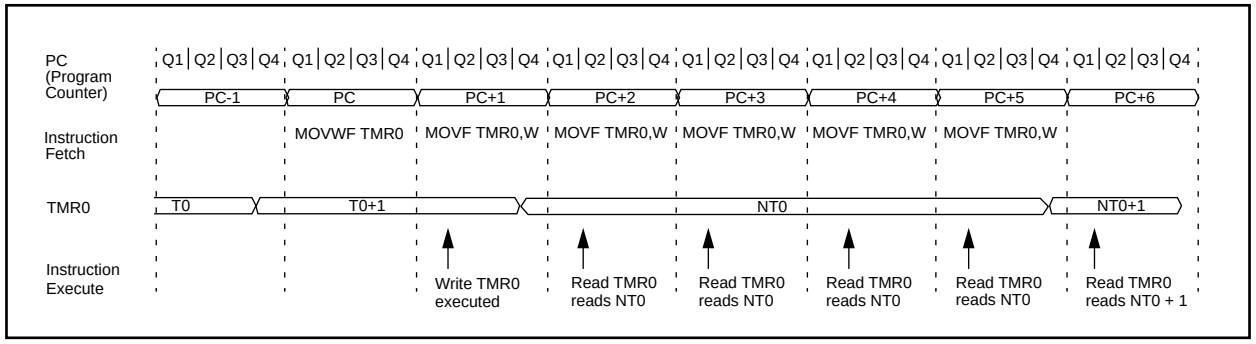
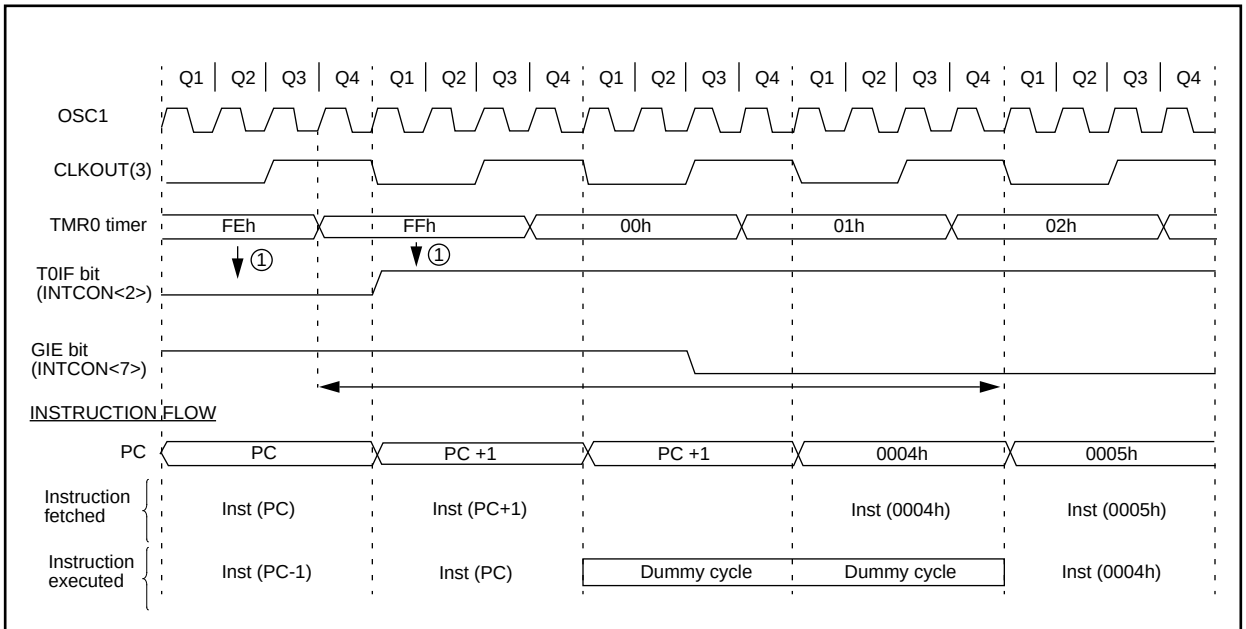


FIGURE 6-4: TIMER0 INTERRUPT TIMING



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 HS oscillator mode.

6.2 Using Timer0 with External Clock

When the external clock input (pin RC3/T0CKI) is used for Timer0, 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 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns).

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 4Tosc (and a small RC delay of 40 ns) 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.

6.2.2 TIMER0 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 Timer0 module is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

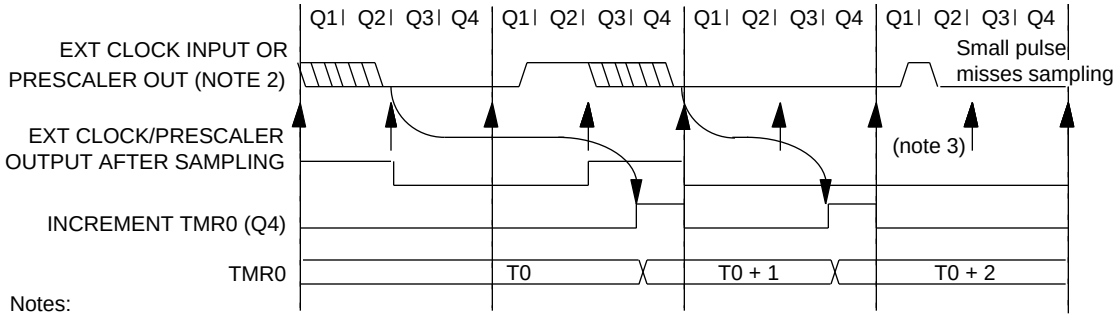
6.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a post-scaler for the Watchdog Timer (Figure 6-1). 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 Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

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

When assigned to the Timer0 module, all instructions writing to the Timer0 module (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x`) 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.

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.

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 Timer0 to WDT.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```
 1.BCF   STATUS,RP0  ;Skip if already in
                     ; Bank 0
 2.CLRWDT            ;Clear WDT
 3.CLRF  TMR0        ;Clear TMR0 & Prescaler
 4.BSF   STATUS, RP0 ;Bank 1
 5.MOVLW '00101111'b;These 3 lines (5, 6, 7)
 6.MOVWF OPTION      ; are required only
                     ; if desired PS<2:0>
 7.CLRWDT            ; are 000 or 001
 8.MOVLW '00101xxx'b ;Set Postscaler to
 9.MOVWF OPTION      ; desired WDT rate
10.BCF   STATUS, RP0 ;Return to Bank 0
```

To change prescaler from the WDT to the Timer0 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→TIMER0)

```
CLRWDT                  ;Clear WDT and
                        ;prescaler
BSF     STATUS, RP0
MOVLW   B'xxxx0xxx'     ;Select TMR0, new
                        ;prescale value and
                        ;clock source
MOVWF   OPTION
BCF     STATUS, RP0
```

TABLE 6-1: SUMMARY OF TIMER0 REGISTERS

Register Name	Function	Address	Power-on Reset Value
TMR0	Timer/counter register	01h	xxxx xxxx
OPTION	Configuration and prescaler assignment bits for TMR0.	81h	1111 1111
INTCON	TMR0 overflow interrupt flag and mask bits.	0Bh	0000 000x

Legend: x = unknown,
Note 1: For reset values of registers in other reset situations refer to Table 10-4.

TABLE 6-2: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	TMR0	TIMER0 TIMER/COUNTER							
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	r	r	T0IF	r	r
81h	OPTION	$\overline{\text{RCPU}}$	r	T0CS	T0SE	PSA	PS2	PS1	PS0
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0

Legend: r = Reserved locations
Shaded boxes are not used by Timer0 module

7.0 INTER-INTEGRATED CIRCUIT SERIAL PORT (I^2C™)

The I^2C 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 I^2C module is compatible with the following interface specifications:

- Inter-Integrated Circuit (I^2C)
- System Management Bus (SMBus)

Note: The I^2C module on PIC14000 only supports I^2C mode. This is different from the standard module used on the PIC16C7X family, which supports both I^2C and SPI modes. Caution should be exercised to avoid enabling SPI mode on the PIC14000.

This section provides an overview of the Inter-IC(I^2C) bus. 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 I^2C module’s hardware, except general call support, while portions of the master protocol will need to be addressed in the PIC14000 software. Table 7-1 defines some of the I^2C bus terminology. For additional information on the I^2C interface specification, please refer to the Philips Corporation document *“The I^2C-bus and How to Use It”.*

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 from or write to the slave device. The master and slave are always in opposite modes (transmitter/receiver) of operation during a data transfer. They may operate in either of these two states:

- 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.

7.1 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 determine the start and stop of data transmission. The START is defined as a high to low transition of SDA when SCL is high. The STOP is defined as a low to high transition of SDA when SCL is high. Figure 7-1 shows the START and STOP. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP, when data is being transmitted the SDA line can only change state when the SCL line is low.

FIGURE 7-1: I^2C START AND STOP CONDITIONS

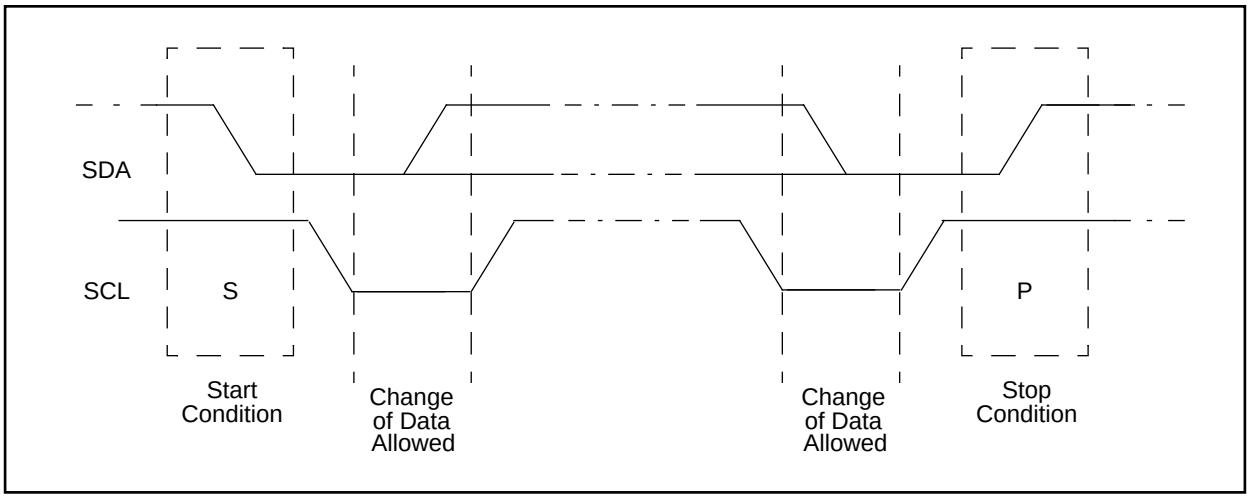


FIGURE 7-2: I^2CSTAT: I^2C PORT STATUS REGISTER

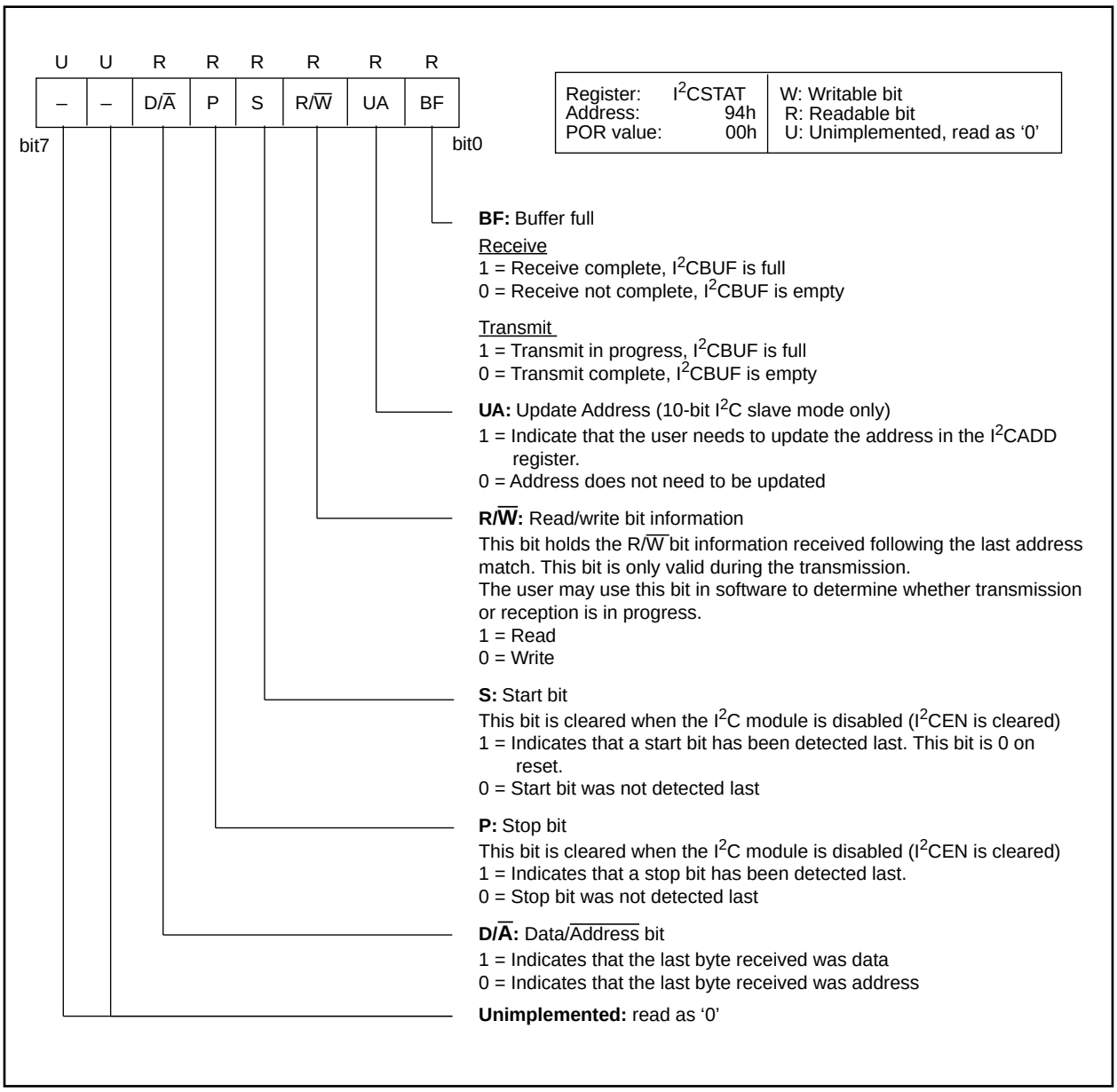


FIGURE 7-3: I^2CCON: I^2C PORT CONTROL REGISTER

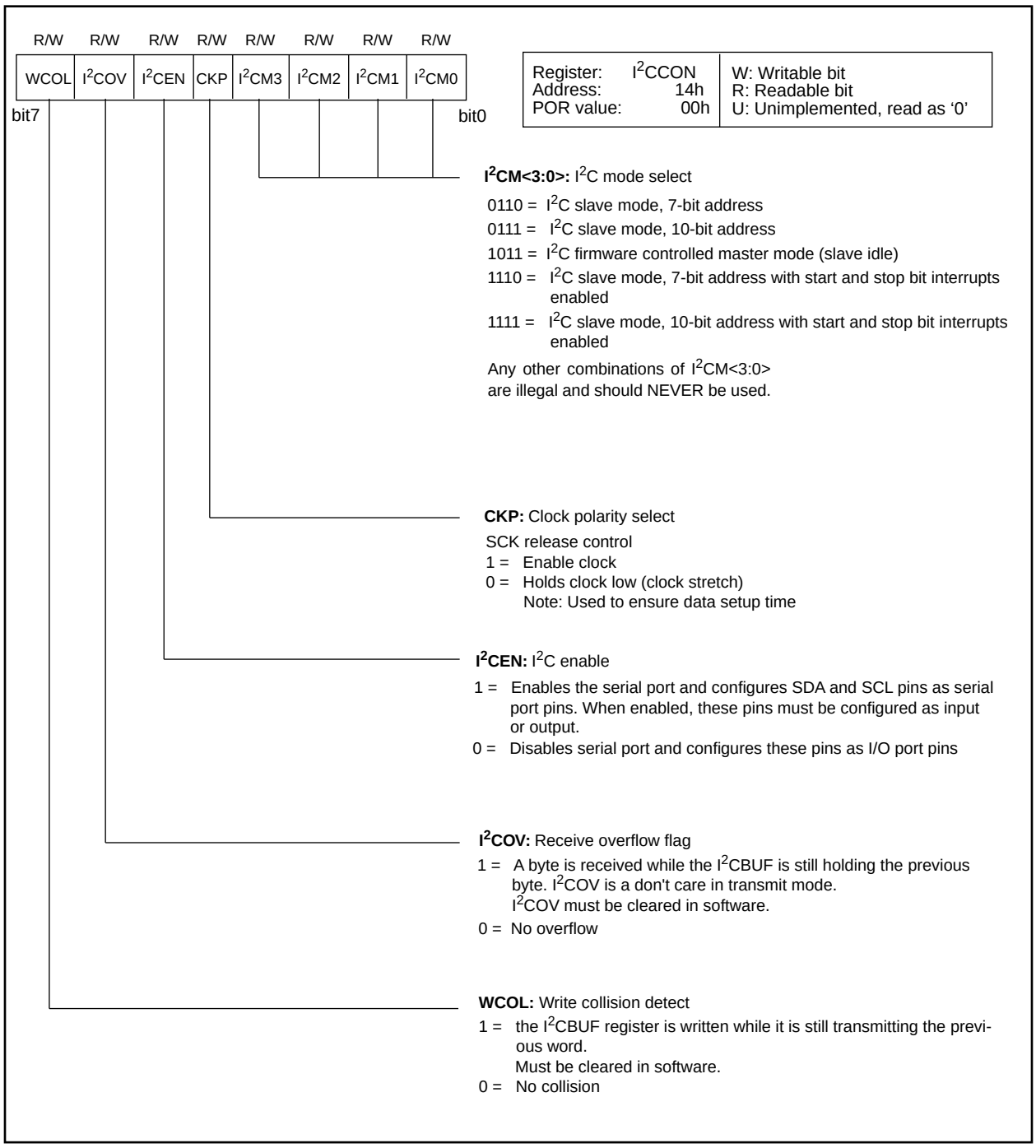


TABLE 7-1: I^2C BUS TERMINOLOGY

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 ensures that the transfer data does not get corrupted.
Synchronization	Procedure where the clock signals of two or more devices are synchronized.

FIGURE 7-4: I^2C 7-BIT ADDRESS FORMAT

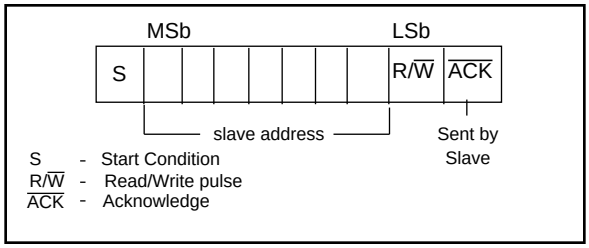
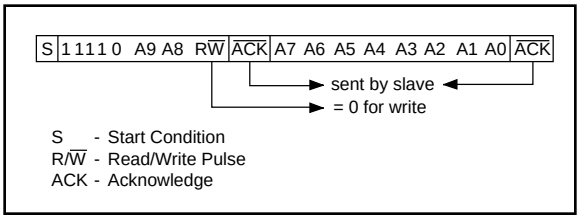


FIGURE 7-5: I^2C 10-BIT ADDRESS FORMAT



7.2 Addressing I^2C Devices

There are two address formats. The simplest is the 7-bit address format with a R/$\overline{W}$ bit (Figure 7-4). The address is the most significant seven bits of the byte. For example when loading the I^2CADD register, the least significant bit is a “don’t care”. The more complex is the 10-bit address with a R/$\overline{W}$ bit (Figure 7-5). For 10-bit address format, two bytes must be transmitted with the first five bits specifying this to be a 10-bit address.

7.3 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 7-6. 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 (Figure 7-1).

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. The slave then releases the SDA line so the master can generate the STOP. The master can also generate the STOP 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 can be accomplished by setting SMHOG (MISC<7>) high. Clearing MISC<7> will resume the data transfer. Figure 7-7 shows a data transfer waveform.

Figure 7-8 and Figure 7-9 show master-transmitter and master-receiver data transfer sequences.

FIGURE 7-6: I^2C SLAVE-RECEIVER ACKNOWLEDGE

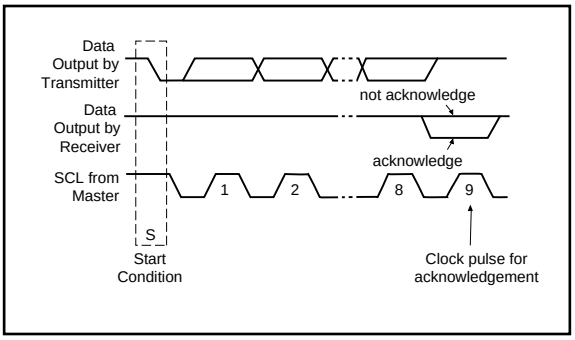
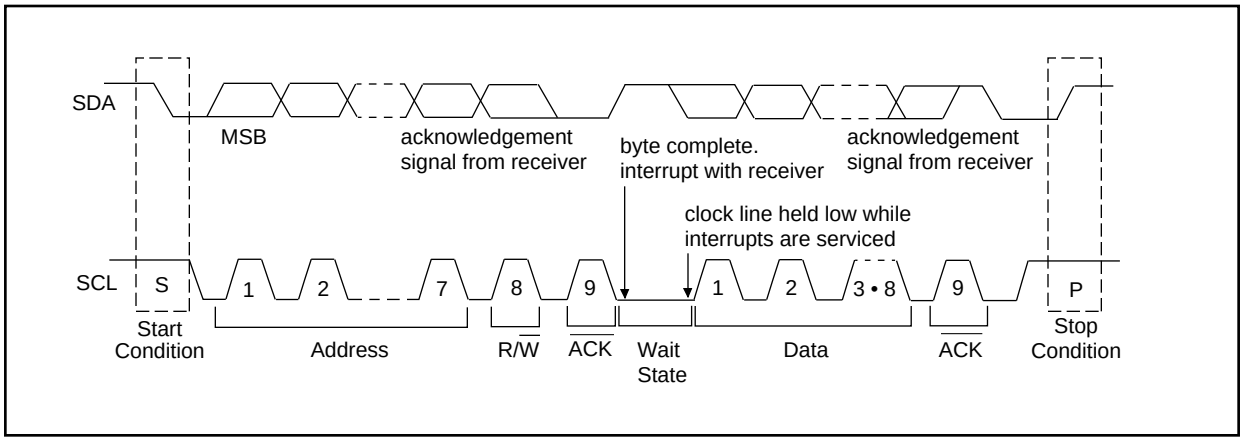


FIGURE 7-7: SAMPLE I^2C DATA TRANSFER



When a master does not wish to relinquish the bus (by generating a STOP condition), a repeated START (Sr) must be generated. This condition is identical to the START (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 7-10.

FIGURE 7-8: MASTER - TRANSMITTER SEQUENCE

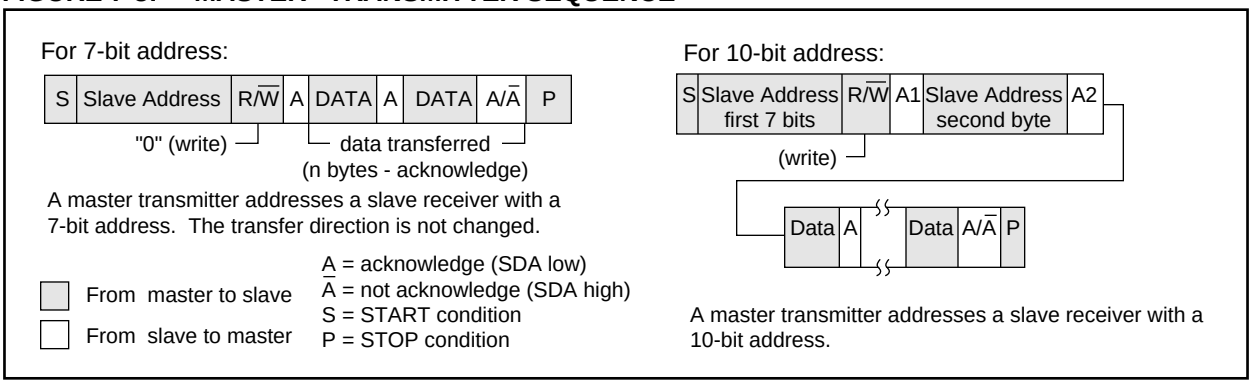


FIGURE 7-9: MASTER - RECEIVER SEQUENCE

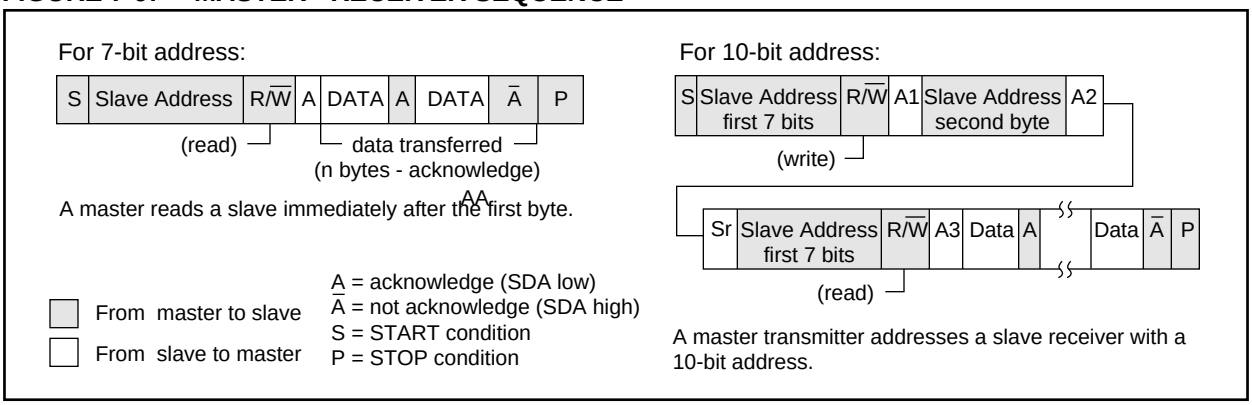
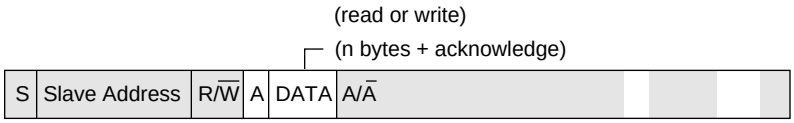


FIGURE 7-10: COMBINED FORMAT



Transfer direction of data and acknowledgement bits depends on R/W bits.

Combined Format:

7.4 Multi-Master Operation

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.

7.4.1 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 (Figure 7-11) and turns off its data output stage. A master which lost arbitrating 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.

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
- A STOP and a data bit
- A repeated START and a STOP

Care needs to be taken to ensure that these conditions do not occur.

7.4.2 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 SCA line high time is determined by the device with the shortest high period. This is shown in the Figure 7-12.

FIGURE 7-11: MULTI-MASTER ARBITRATION (2 MASTERS)

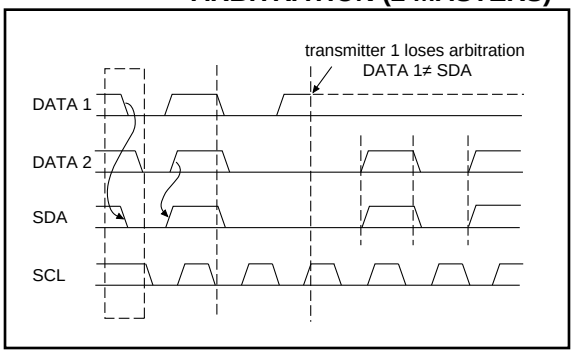


FIGURE 7-12: I^2C CLOCK SYNCHRONIZATION

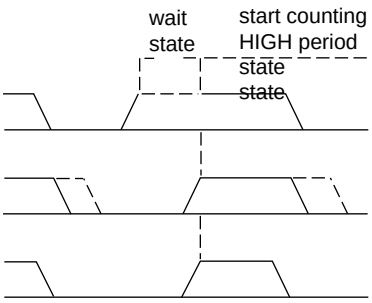
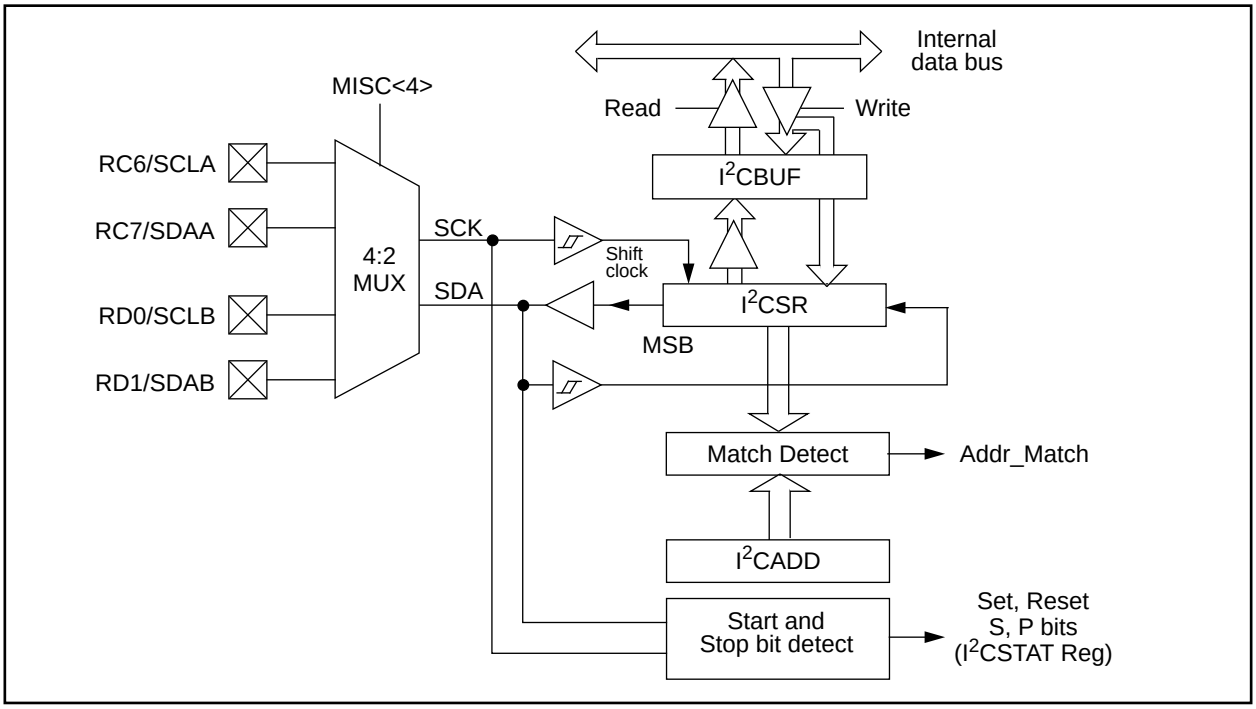


FIGURE 7-13: I^2C BLOCK DIAGRAM



7.5 I^2C Operation

The I^2C module in I^2C mode fully implements all slave functions, and provides support in hardware to facilitate software implementations of the master functions. The I^2C 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 RC6/SCLA pin, which is the I^2C clock, and the RC7/SDAA pin which acts as the I^2C data. The I^2C module can also be accessed via the RD0/SCLB and RD1/SDAB pins by setting I^2CSEL (MISC<4>).The user must configure these pins as inputs or outputs through the TRISC<7:6> or TRISD<1:0> bits. A block diagram of the I^2C module in I^2C mode is shown in Figure 7-13. The I^2C module functions are enabled by setting the I^2CCON<5> bit.

The I^2C module has five registers for I^2C operation. These are the:

- I^2C Control Register (I^2CCON)
- I^2C Status Register (I^2CSTAT)
- Serial Receive/Transmit Buffer (I^2CBUF)
- I^2C Shift Register (I^2CSR) - Not directly accessible
- Address Register (I^2CADD)

The I^2CCON register (14h) allows control of the I^2C operation. Four mode selection bits (I^2CCON<3:0>) allow one of the following I^2C modes to be selected:

- I^2C Slave mode (7-bit address)
- I^2C Slave mode (10-bit address)
- I^2C Slave mode (7-bit address), with start and stop bit interrupts enabled
- I^2C Slave mode (10-bit address), with start and stop bit interrupts enabled
- I^2C Firmware Controlled Master mode, slave is idle

Selection of any I^2C mode with the I^2CEN bit set, forces the SCL and SDA pins to be open collector, provided these pins are set to inputs through the TRISC bits.

The I^2CSTAT 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 I^2CSTAT register is read only.

The I^2CBUF is the register to which transfer data is written to or read from. The I^2CSR register shifts the data in or out of the device. In receive operations, the I^2CBUF and I^2CSR create a double 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 I^2CBUF and PIR1<3> is set. If another complete byte is received before the I^2CBUF is read, a receiver overflow has occurred and the I^2CCON<6> is set.

The I^2CADD 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).

7.5.1 SLAVE MODE

In slave mode, the SCLx and SDAx pins must be configured as inputs (TRISC<7:6> or TRISD<1:0> are set). The I^2C 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 I^2CBUF with the received value in the I^2CSR.

There are two conditions that will cause the I^2C module not to give this $\overline{ACK}$ pulse. These are if either (or both) occur:

- the Buffer Full (BF), $I^2CSTAT<0>$, bit was set before the transfer was received, or
- the Overflow (I^2COV), $I^2CCON<6>$ bit was set before the transfer was received.

In this case, the I^2CSR value is not loaded into the I^2CBUF, but the I^2CIF bit is set. Table 7-2 shows what happens when a data transfer byte is received, given the status of the BF and I^2COV bits. The shaded boxes show the conditions where user software did not properly clear the overflow condition. The BF flag is cleared by reading the I^2CBUF register while the I^2COV 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 I^2C module is shown in the AC timing specifications.

TABLE 7-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		I^2CSR-> I^2CBUF	Generate $\overline{ACK}$ Pulse	Set I^2CIF bit (I^2C interrupt if enabled)
BF	I^2COV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

7.5.1.1 ADDRESSING

Once the I^2C module has been enabled, the I^2C waits for a START to occur. Following the START, the 8-bits are shifted into the I^2CSR. All incoming bits are sampled with the rising edge of the clock (SCL) line. The $I^2CSR<7:1>$ is compared to the I^2CADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and I^2COV bits are clear, the following things happen:

- I^2CSR loaded into I^2CBUF
- Buffer Full (BF) bit is set
- $\overline{ACK}$ pulse is generated
- I^2C Interrupt Flag (I^2CIF) is set (interrupt is generated if enabled (I^2CIE set) on falling edge of ninth SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave (Figure 7-5). 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 received 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 (I^2CIF, BF and UA are set).
2. Update I^2CADD with second (low) byte of address (clears UA and releases SCL line).
3. Read I^2CBUF (clears BF) and clear I^2CIF.
4. Receive second (low) byte of address (I^2CIF, BF and UA are set).
5. Update I^2CADD with first (high) byte of address (clears UA, if match releases SCL line).
6. Read I^2CBUF (clears BF) and clear I^2CIF
7. Receive Repeated START.
8. Receive first (high) byte of address (I^2CIF and BF are set).
9. Read I^2CBUF (clears BF) and clear I^2CIF.

7.5.1.2 RECEPTION

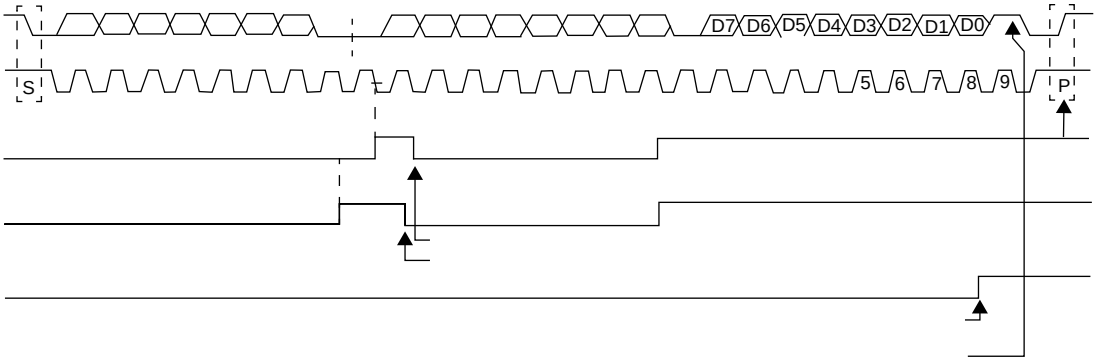
When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the I^2CSTAT register is cleared. The received address is loaded into the I^2CBUF.

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 ($I^2CSTAT<0>$) is set or the I^2COV bit ($I^2CCON<6>$) is set (Figure 7-14).

An I^2CIF interrupt is generated for each data transfer byte. The I^2CIF bit must be cleared in software, and the I^2CSTAT register is used to determine the status of the byte. In master mode with slave enabled, three interrupt sources are possible. Reading BF, P and S will indicate the source of the interrupt.

Caution: BF is set after receipt of eight bits and automatically cleared after the I^2CBUF is read. However, the flag is not actually cleared until receipt of the acknowledge pulse. Otherwise extra reads appear to be valid.

FIGURE 7-14: I^2C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)



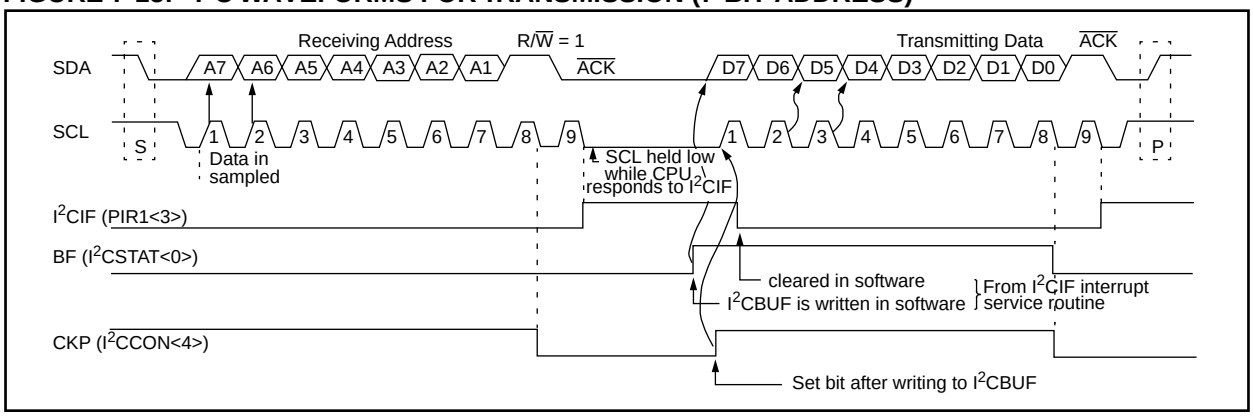
7.5.1.3 TRANSMISSION

When the R/$\overline{W}$ bit of the address byte is set and an address match occurs, the R/$\overline{W}$ bit of the I^2CSTAT register is set. The received address is loaded into the I^2CBUF 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 I^2CBUF register, which also loads the I^2CSR register. Then the SCL pin should be enabled by setting the CKP bit (I^2CCON<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 (Figure 7-15).

A I^2CIF interrupt is generated for each data transfer byte. The I^2CIF bit must be cleared in software, and the I^2CSTAT register is used to determine the status of the byte. The I^2CIF 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 I^2CBUF register, which also loads the I^2CSR register. Then the SCL pin should be enabled by setting the CKP bit (I^2CCON<4>).

FIGURE 7-15: I^2C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



7.5.2 MASTER MODE

Master mode operation is supported by interrupt generation on the detection of the START and STOP. The STOP(P) and START(S) bits are cleared from a reset or when the I^2C 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<7:6> or TRISD<1:0> bits to an output (cleared). The output level is always low, regardless of the value(s) in PORTC<7:6> or PORTD<1:0>. So when transmitting data, a “1” data bit must have the TRISC<7> or TRISD<1> bit set (input) and a “0” data bit must have the TRISC<7> or TRISD<1> bit cleared (output). The same scenario is true for the SCL line with the TRISC<6> or TRISD<0> bit.

The following events will cause the I^2C interrupt Flag (I^2CIF) to be set (I^2C interrupt if enabled):

- START
- STOP
- Data transfer byte transmitted/received

Master mode of operation can be done with either the slave mode idle ($I^2CM3...I^2CM0$ = 1011b) or with the slave active. When both master and slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

7.5.3 MULTI-MASTER MODE

In multi-master mode, the interrupt generation on the detection of the START and STOP allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a reset or when the I^2C 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 I^2C interrupt will generate the interrupt when the STOP 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 low level is present, the device needs to release the SDA and SCL lines (set TRISC<7:6>). 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 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 7-3: 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	r	r	T0IF	r	r
0Ch	PIR1	CMIF	—	—	PBIF	I^2CIF	RCIF	ADCIF	OVFIF
8Ch	PIE1	CMIE	—	—	PBIE	I^2CIE	RCIE	ADCIE	OVFIE
13h	I^2CBUF	I^2C Serial Port Receive Buffer/Transmit Register							
93h	I^2CADD	I^2C mode Synchronous Serial Port (I^2C mode) Address Register							
14h	I^2CCON	WCOL	I^2CON	I^2CEN	CKP	I^2CM3	I^2CM2	I^2CM1	I^2CM0
94h	I^2CSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
9Eh	MISC	SMHOG	SPGNDB	SPGNDA	I^2CSEL	SMBUS	INCLKEN	OSC2	OSC1
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
88h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0

Legend: — = Unimplemented location, read as ‘0’
r = reserved locations, default is POR value and **should not be overwritten with any value**

Note: Shaded boxes are not used by the I^2C module.

FIGURE 7-16: MISC REGISTER

9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MISC	SMHOG	SPGNDB	SPGNDA	I^2CSEL	SMBUS	INCLKEN	OSC2	OSC1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
POR value 00h	0	0	0	0	0	0	0	X

Bit	Name	Function
B7	SMHOG	SMHOG enable 1 = Stretch I^2C CLK signal (hold low) when receive data buffer is full (refer to Section 7.5.4). For pausing I^2C transfers while preventing interruptions of A/D conversions. 0 = Disable I^2C CLK stretch.
B6	SPGNDB	Serial Port Ground Select 1 = PORTD<1:0> ground reference is the RD5/AN5 pin. 0 = PORTD<1:0> ground reference is Vss.
B5	SPGNDA	Serial Port Ground Select 1 = PORTC<7:6> ground reference is the RA1/AN1 pin. 0 = PORTC<7:6> ground reference is Vss.
B4	I^2CSEL	I^2C Port select Bit. 1 = PORTD<1:0> are used as the I^2C clock and data lines. 0 = PORTC<7:6> are used as the I^2C clock and data lines.
B3	SMBus	SMBus-Compatibility Select 1 = SMBus compatibility mode is enabled. PORTC<7:6> and PORTD<1:0> have SMBus-compatible input thresholds. 0 = SMBus-compatibility is disabled. PORTC<7:6> and PORTD<1:0> have Schmitt Trigger input thresholds.
B2	INCLKEN	Oscillator Output Select (available in IN mode only). 1 = Output IN oscillator signal divided by four on OSC2 pin. 0 = Disconnect IN oscillator signal from OSC2 pin.
B1	OSC2	OSC2 output port bit (available in IN mode only). Writes to this location affect the OSC2 pin in IN mode. Reads return the value of the output latch.
B0	OSC1	OSC1 input port bit (available in IN mode only). Reads from this location return the status of the OSC1 pin in IN mode. Writes have no effect.

FIGURE 7-17: OPERATION OF THE I^2C 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 ((I2CBUF=Full) OR (I2COV = 1))
        {       Set I2COV;
                Do not acknowledge;
        }
else    {       transfer I2CSR → I2CBUF;
                send ACK = 0;
        }
Receive 8-bits in I2CSR;
Set interrupt;
```

```
XMIT_MODE:
While ((I2CBUF = 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;
```

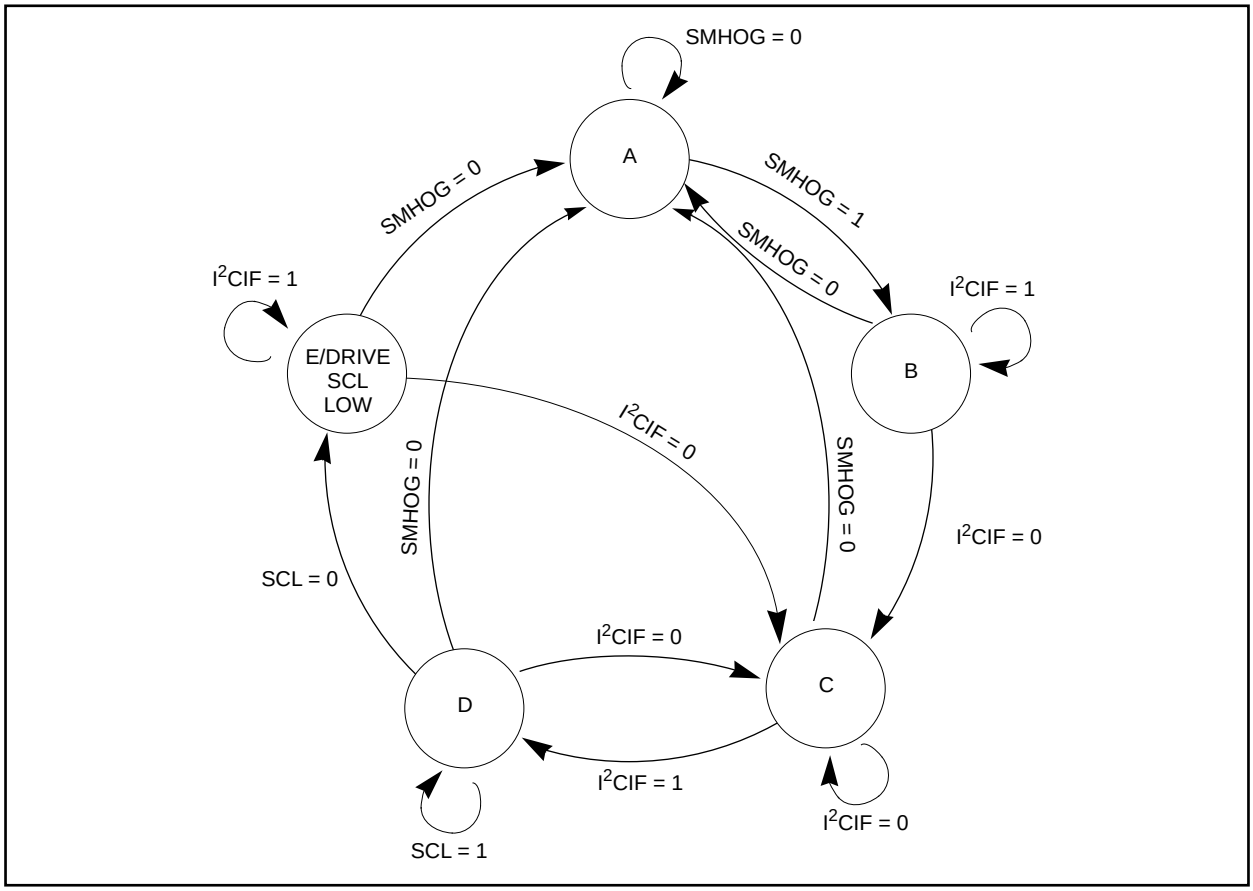
```
IDLE_MODE (10-Bit):
If (High_byte_addr_match AND (R/W = 0))
        {       PRIOR_ADDR_MATCH = FALSE;
                Set interrupt;
                if ((I2CBUF = Full) OR ((I2COV = 1))
                        {       Set I2COV;
                                Do not acknowledge;
                        }
                else    {       Set UA = 1;
                                Send ACK = 0;
                                While (I2CADD 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 (I2CADD 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;
        }
```


7.5.4 SMBus™ AND ACCESS.bus™ CONSIDERATIONS

PIC14000 is compliant with the SMBus specification published by Intel. Some key points to note regarding the bus specifications and how it pertains to the PIC14000 hardware are listed below:

- SMBus has fixed input voltage thresholds. PIC14000 I/O buffers have programmable levels that can be selected to be compatible with both SMBus threshold levels via the SMBus and SPGND bits in the MISC register.
- A mechanism to stretch the I^2C clock time has been implemented to support SMBus slave transactions. The SMHOG bit (MISC<7>) allows hardware to automatically force and hold the I^2C clock line low when a data byte has been received. This prevents the SMBus master from overflowing the receive buffer in instances where the microcontroller may be to busy servicing higher priority tasks to respond to a I^2C module interrupt. Or, if the microcontroller is in SLEEP mode and needs time to wake-up and respond to the I^2C interrupt.

FIGURE 7-18: SMHOG STATE MACHINE



NOTES:

8.0 ANALOG MODULES FOR A/D CONVERSION

8.1 Overview

The PIC14000 includes analog components to create a slope A/D converter including:

- Comparator
- 4-bit programmable current source
- 16-channel analog mux
- 16-bit timer with capture register

Each channel is converted independently by means of a slope conversion method using a single precision comparator. The programmable current source feeds an external 0.1 µF (nominal) capacitor to generate the ramp voltage used in the conversion.

8.2 Conversion Process

These are the steps to perform data conversion:

- Clear REFOFF (SLPCON<5>) and ADOFF (SLPCON<0>) bits to enable the A/D module.
- Initialize ADCON1<7:4> to initialize the programmable current source.
- Set ADRST (ADCON0<1>), for a minimum of 200 µs to stop the timer and fully discharge the ramp capacitor to ground.
- The A/D timer (ADTMR) increments from 0000h to FFFFh and must be initialized before each conversion.
- To start a conversion, clear ADRST through software, it will allow the timer to begin counting and the ramp capacitor to begin charging.
- When the ramp voltage exceeds the analog input, the comparator output changes from high to low.
- This transition causes a capture event and copies the current A/D timer value into the 16-bit capture register.
- An interrupt is generated to the CPU if enabled.

Note: The A/D timer continues to run following a capture event.

The maximum A/D timer count is 65,536. It can be clocked by the on-chip or external oscillator. At a 4 MHz oscillation frequency, the maximum conversion time is 16.38 ms for a full count. A typical conversion should complete before full-count is reached. A timer overflow flag is set once the timer rolls over (FFFFh to 0000h), and an interrupt is sent to the CPU, if enabled.

End-user calibration is simplified or eliminated by making use of the on-chip EPROM. Internal component values are measured at factory final test and stored in the memory for use by the application firmware.

Periodic conversion cycles should be performed on the bandgap and slope references (described in Section 9.0) to compensate for A/D component drift. Measurements for the reference voltage count are equated to the voltage value stored into EPROM during calibration. All other channel measurements are compensated for by ratioing the actual count with the bandgap count and multiplying by the bandgap voltage value stored in EPROM. Since all measurements are relative to the reference, offset voltages inherent in the comparator are cancelled out. See AN624, “PIC14000 A/D Theory and Implementation” for further details of A/D operation.

The analog components used in the conversion and the A/D timer can be disabled during idle periods for maximum power savings. Power-saving can be achieved via software and/or hardware control (Section 10.8).

8.3 A/D Timer (ADTMR) Module

The A/D timer (ADTMR) is comprised of a 16-bit up timer, which is incremented every oscillator cycle. ADTMR is reset to 0000h by a power-up reset; otherwise the software must initialize it after each conversion. A separate 16-bit capture register (ADCAP) is used to capture the ADTMR count if an A/D capture event occurs (see below). Both the A/D timer and capture register are readable and writable. The low byte of the A/D timer (ADTMRL) is accessed at location 0Eh while the high byte (ADTMRH) is accessed at location 0Fh. Similarly, the low byte of the A/D capture register (ADCAP) is accessed at location 15h, and the high byte is located at 16h.

Caution: Reading or writing the ADTMR register during an A/D conversion cycle can produce unpredictable results and is not recommended.

Note: The correct sequence for writing the ADTMR register is HI byte followed by LO byte. Reversing this order will prevent the A/D timer from running.

During conversion one or both of the following events will occur:

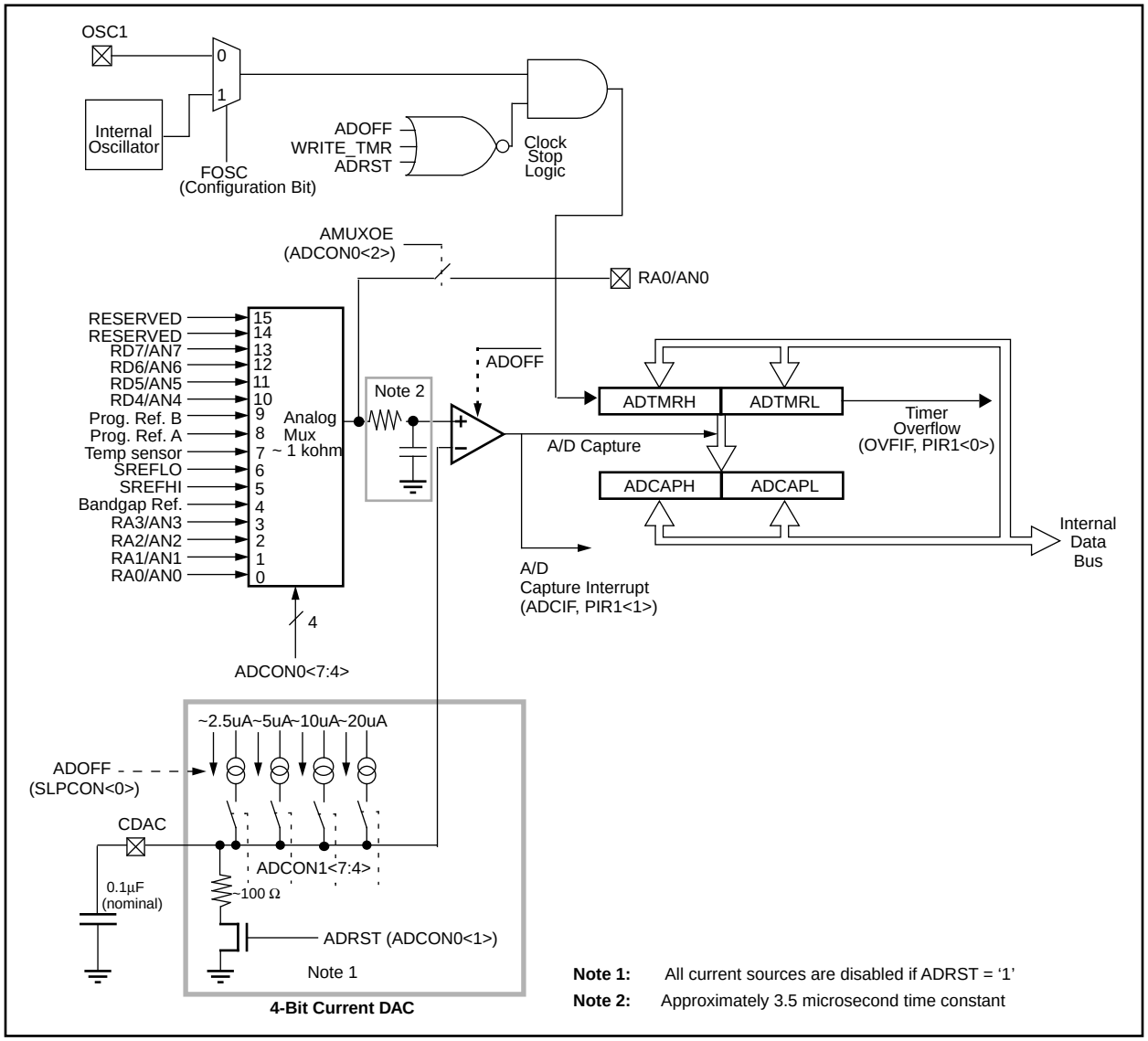
1. capture event
2. timer overflow

In a capture event, the comparator trips when the slope voltage on the CDAC output exceeds the input voltage, causing the comparator output to transition from high to low. This causes a transfer of the current timer count to the capture register and sets the ADCIF flag (PIR1<1>).

A CPU interrupt will be generated if bit ADCIE (PIE1<1>) is set to '1' (interrupt enabled). In addition, the Global Interrupt Enable and Peripheral Interrupt Enables (INTCON<7,6>) must also be set. Software is responsible for clearing the ADCIF flag prior to the next conversion cycle. Note that this interrupt can only occur once per conversion cycle.

In a timer overflow condition, the timer rolls over from FFFFh to 0000h, and a capture overflow flag (OVFIF) is asserted (PIR1<0>). The timer continues to increment following a timer overflow. A CPU interrupt can be generated if bit OVFIE (PIE1<0>) is set (interrupt enabled). In addition, the Global Interrupt Enable and Peripheral Interrupt Enables (INTCON<7,6>) must also be set. Software is responsible for clearing the OVFIF flag prior to the next conversion cycle.

FIGURE 8-1: A/D BLOCK DIAGRAM



Note 1: All current sources are disabled if ADRST = '1'

Note 2: Approximately 3.5 microsecond time constant

FIGURE 8-2: EXAMPLE A/D CONVERSION CYCLE

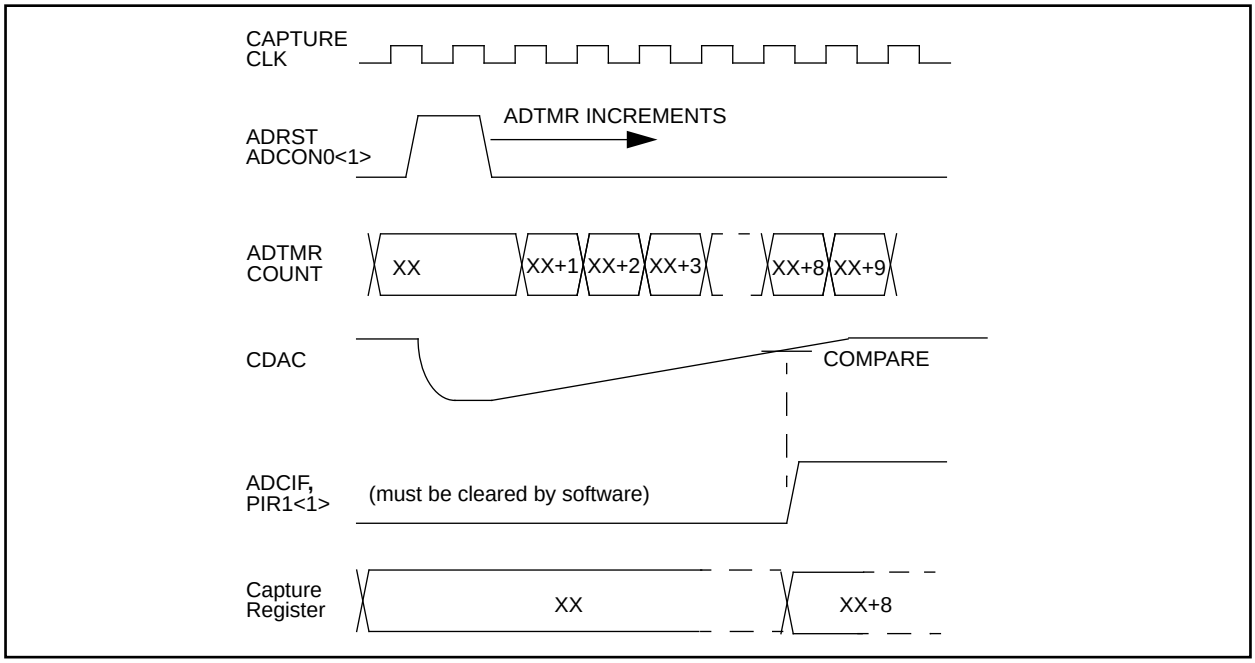


FIGURE 8-3: A/D CAPTURE TIMER (LOW BYTE)

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTMRL	b7	b6	b5	b4	b3	b2	b1	b0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

FIGURE 8-4: A/D CAPTURE TIMER (HIGH BYTE)

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTMRH	b15	b14	b13	b12	b11	b10	b9	b8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

FIGURE 8-5: A/D CAPTURE REGISTER (LOW BYTE)

15h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCAPL	b7	b6	b5	b4	b3	b2	b1	b0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

FIGURE 8-6: A/D CAPTURE REGISTER (HIGH BYTE)

16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCAPH	b15	b14	b13	b12	b11	b10	b9	b8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

Legend: U= unimplemented, X = unknown.

8.4 A/D Comparator

The PIC14000 includes a high gain comparator for A/D conversions. The positive input terminal of the A/D comparator is connected to the output of an analog mux through an RC low-pass filter. The nominal time-constant for the RC filter is 3.5 μs. The negative input terminal is connected to the external 0.1 μF (nominal) ramp capacitor.

8.5 Analog Mux

A total of 16 channels are internally multiplexed to the single A/D comparator positive input. Four configuration bits (ADCON0<7:4>) select the channel to be converted. Refer to Table 8-1 for channel assignments.

TABLE 8-1: A/D CHANNEL ASSIGNMENT

ADCON0(7:4)				A/D Channel
0	0	0	0	RA0/AN0 pin
0	0	0	1	RA1/AN1 pin
0	0	1	0	RA2/AN2 pin
0	0	1	1	RA3/AN3 pin
0	1	0	0	Bandgap reference voltage
0	1	0	1	Slope reference SREFHI
0	1	1	0	Slope reference SREFLO
0	1	1	1	Internal temperature sensor
1	0	0	0	Programmable reference A output
1	0	0	1	Programmable reference B output
1	0	1	0	RD4/AN4 pin
1	0	1	1	RD5/AN5 pin
1	1	0	0	RD6/AN6 pin
1	1	0	1	RD7/AN7 pin
1	1	1	0	Reserved
1	1	1	1	Reserved

8.6 Programmable Current Source

Four configuration bits (ADCON1<7:4>) are used to control a programmable current source for generating the ramp voltage to the A/D comparator. It allows compensation for full-scale input voltage, clock frequency and CDAC capacitor tolerance variations. The current values range from 0 to 33.75 μA (nominal) in 2.25 μA increments. The intermediate values of the current source are as follows:

TABLE 8-2: PROGRAMMABLE CURRENT SOURCE SELECTION

ADCON1<7:4>				Current Source Output
0	0	0	0	OFF - all current sources disabled
0	0	0	1	2.25 μA
0	0	1	0	4.5 μA
0	0	1	1	6.75 μA
0	1	0	0	9 μA
0	1	0	1	11.25 μA
0	1	1	0	13.5 μA
0	1	1	1	15.75 μA
1	0	0	0	18 μA
1	0	0	1	20.25 μA
1	0	1	0	22.5 μA
1	0	1	1	24.75 μA
1	1	0	0	27 μA
1	1	0	1	29.25 μA
1	1	1	0	31.5 μA
1	1	1	1	33.75 μA

The programmable current source output is tied to the CDAC pin and is used to charge an external capacitor to generate the ramp voltage for the A/D comparator. (Refer to Figure 8-1.) This capacitor should have a low voltage-coefficient as found in PTFE, polypropylene, or polystyrene capacitors, for optimum results. The capacitor must be discharged at the beginning of each conversion cycle by asserting ADRST (ADCON0<1>) for at least 200 μs to allow a complete discharge. Asserting ADRST disables the current sources internally. Current flow begins when ADRST is cleared.

8.7 A/D Control Registers

Two A/D control registers are provided on the PIC14000 to control the conversion process. These are ADCON0 (1Fh) and ADCON1 (9Fh). Both registers are readable and writable.

TABLE 8-3: A/D CONTROL AND STATUS REGISTER 0

1Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON0	ADCS3	ADCS2	ADCS1	ADCS0	—	AMUXOE	ADRST	ADZERO
Read/Write	R/W	R/W	R/W	R/W	U	R/W	R/W	R/W
POR value 02h	0	0	0	0	0	0	1	0

Bit	Name	Function
B7-B4	ADCS3 ADCS2 ADCS1 ADCS0	A/D Channel Selects. Refer to Table 8-1.
B3	—	Unimplemented. Read as ‘0’.
B2	AMUXOE	Analog Mux Output Enable 1 = Connect AMUX Output to RA0/AN0 pin (overrides TRISA<0> setting) 0 = RA0/AN0 pin normal
B1	ADRST	A/D Reset Control Bit 1 = Stop the A/D Timer, discharge CDAC capacitor 0 = Normal operation (A/D running)
B0	ADZERO	A/D Zero Select Control. (Refer to Section 9.2) 1 = Enable zeroing operation on RA1/AN1 and RD5/AN5 0 = Normal operation (sample RA1/AN1 and RD5/AN5 pins)

TABLE 8-4: A/D CONTROL AND STATUS REGISTER 1

9Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCON1	ADDAC3	ADDAC2	ADDAC1	ADDAC0	PCFG3	PCFG2	PCFG1	PCFG0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

Bit	Name	Function
B7-B4	ADDAC3 ADDAC2 ADDAC1 ADDAC0	A/D Current Source Selects. Refer to Table 8-2.
B3-B2	PCFG3 PCFG2	PORTD Configuration Selects (See Table 8-5)
B1-B0	PCFG1 PCFG0	PORTA Configuration Selects (See Table 8-5)

TABLE 8-5: PORTA AND PORTD CONFIGURATION

ADCON1<1:0>	RA0/AN0	RA1/AN1	RA2/AN2	RA3/AN3
ADCON1<3:2>	**RD4/AN4**	**RD5/AN5**	**RD6/AN6**	**RD7/AN7**
0 0	A	A	A	A
0 1	A	A	A	D
1 0	A	A	D	D
1 1	D	D	D	D

Legend: A = Analog input, D = Digital I/O

8.8 A/D Speed, Resolution and Capacitor Selection

The conversion time for the A/D converter on the PIC14000 can be calculated using the equation:

Conversion Time = (1/Fosc) x 2^N

Where Fosc is the oscillator frequency and N is the number of bits of resolution desired.

Therefore at 4MHz, the conversion time for 16 bits is 16.384 msec. Conversely, it is 256 µsec for 10 bits.

Choosing the correct ramp capacitor for the CDAC pin is required to achieve the desired resolution, conversion time and full scale input voltage. The equation for selecting the ramp capacitor value is:

Capacitor = (conversion time in seconds) X
(current source output in amps) / (full scale in volts)

Table 8-6 provides example capacitor values for the desired A/D resolution, conversion time, and full scale voltage measurement.

TABLE 8-6: CDAC CAPACITOR SELECTION (EXAMPLES FOR FULL SCALE OF 3.5V AND 1.5V)

A/D Resolution (Bits)	**Conversion Time** (Seconds)	**Full Scale** (Volts)	**A/D Current Source Output (µamps)**	**Calculated CDAC Capacitor** (Farads)	**CDAC Capacitor Nearest Standard Value**
16	0.016384	3.5	24.75	1.17E-07	.1uF
14	0.004096	3.5	24.75	2.93E-08	.022uF
12	0.001024	3.5	24.75	7.31E-09	6800pF
16	0.016384	1.5	24.75	2.73E-07	0.22µF
14	0.004096	1.5	24.75	6.83E-08	68nF
12	0.001024	1.5	24.75	1.71E-08	15nF

Note: Assumes FOSC of 4 MHz.

9.0 OTHER ANALOG MODULES

The PIC14000 has additional analog modules for mixed signal applications. These include:

- bandgap voltage reference
- comparators with programmable references
- internal temperature sensor
- voltage regulator control

9.1 Bandgap Voltage Reference

The bandgap reference circuit is used to generate a 1.2V nominal stable voltage reference for the A/D and the low-voltage detector. The bandgap reference is channel 4 of the analog mux. The bandgap reference voltage is stored in the calibration space EPROM (See Table 4-2). To enable the bandgap reference REFOFF (SLPCON<5>) must be cleared.

9.2 Level-Shift Networks

The RA1/AN1 and RA5/AN5 pins have an internal level-shift network. A current source and resistor are used to bias the pin voltage by about +0.5V into a range usable by the A/D converter. The nominal value of bias current source is 5 μA and the resistor is 100 kohms.

The level-shift function can be turned on by clearing the LSOFF bit (SLPCON<4>) to '0'.

Note:	The minimum voltage permissible at the RA1/AN1 and RA5/AN5 pins is -0.3V. The input protection diodes will begin to turn on beyond -0.3V, introducing significant errors in the A/D readings. Under no conditions should the pin voltage fall below -0.5V.

9.2.1 ZEROING/FILTERING SWITCHES

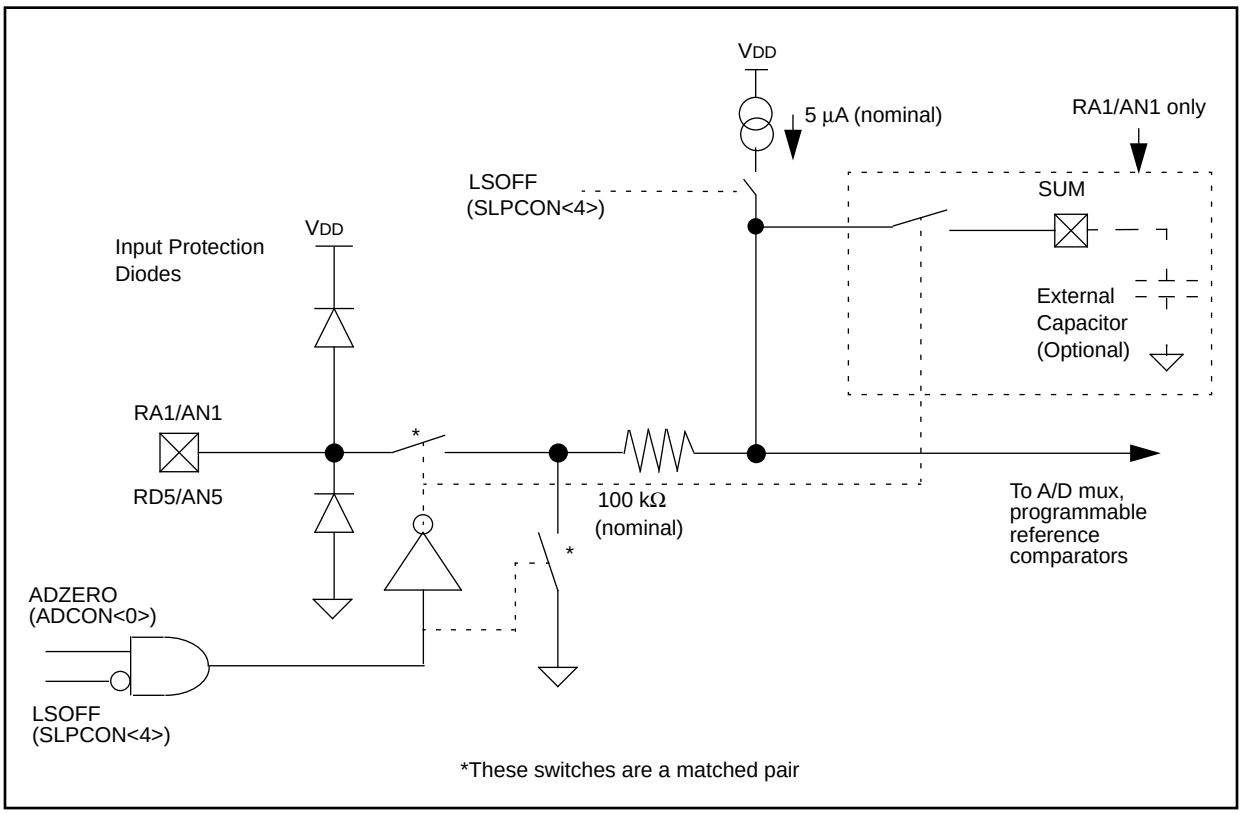
The RA1/AN1 and RA5/AN5 inputs also have a matched pair of pass gates useful for current-measurement applications. One gate is connected between the pin and the level-shift network. The second pass gate is connected to ground as shown in Figure 9-1. By setting the ADZERO bit (ADCON0<0>), a zero-current condition is simulated. Subsequent A/D readings are calculated relative to this zero count from the A/D. This zeroing of the current provides very high accuracies at low current values where it is most needed.

For additional noise filtering or for capturing short duration periodic pulses, an optional filter capacitor may be connected from the SUM pin to ground (this feature is available for RA1/AN1 only). This forms an RC network with the internal 100 kohm (nominal) bias resistor to act as a low pass filter. The capacitor size can be adjusted for the desired time constant.

A switch is included between the output from the RA1/AN1 level-shift network and the SUM pin. This switch is closed during A/D sampling periods and is automatically opened during a zeroing operation (if ADZERO = '1'). If not required in the system, this pin should be left floating (not connected).

Setting the LSOFF bit (SLPCON<4>) disables the level-shift networks, so the RA1/AN1 and RA5/AN5 pins can continue to be used as general-purpose analog inputs.

FIGURE 9-1: LEVEL-SHIFT NETWORKS



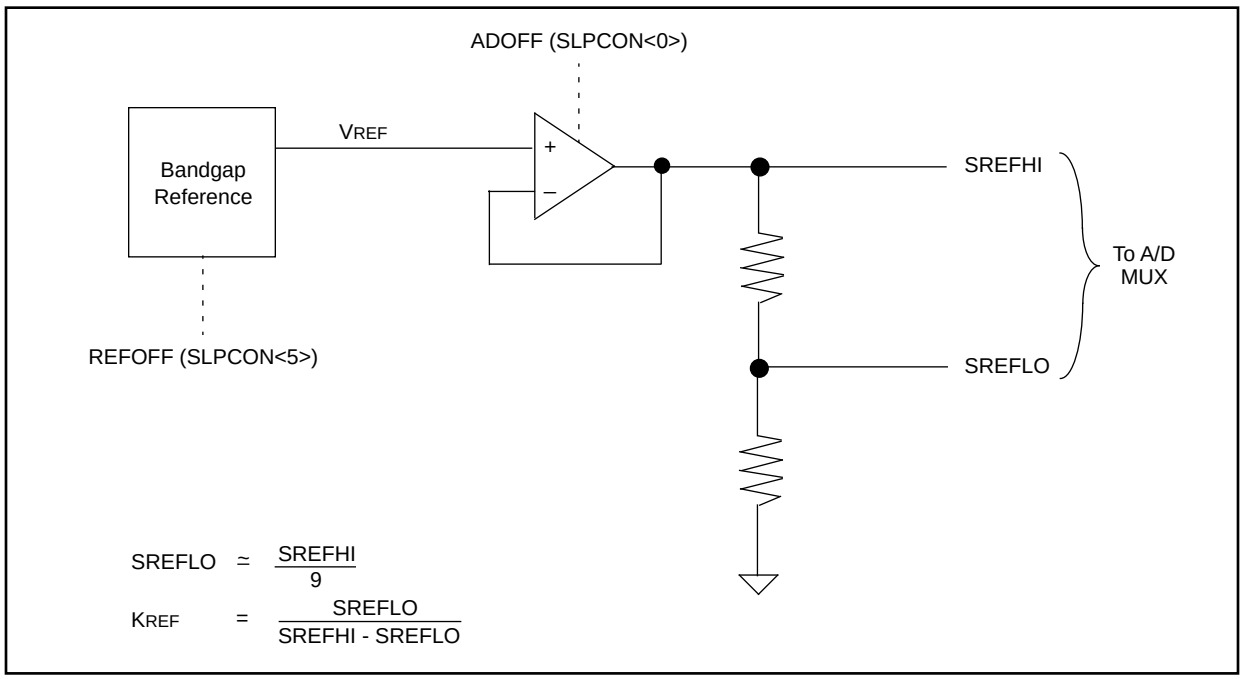
9.3 Slope Reference Voltage Divider

The slope reference voltage divider circuit, consisting of a buffer amplifier and resistor divider, is connected to the internal bandgap reference producing two other voltage references called SREFHI and SREFLO (see Figure 9-2). SREFHI is nominally the same as the bandgap voltage, 1.2V, and SREFLO is nominally 0.13V. These reference voltages are available on two of the analog multiplexer channels. The A/D module and firmware can measure the SREFHI and SREFLO voltages, and in conjunction with the KREF and KBG calibration data correct for the A/D's offset and slope errors. See AN624 for further details.

9.4 Internal Temperature Sensor

The internal temperature sensor is connected to the channel 7 input of the A/D converter. The sensor voltage is 1.05V nominal at 25°C and its temperature coefficient is approximately 3.7mV/°C. The sensor voltage at 25°C and the temperature coefficient values are stored in the calibration space EPROM (See Table 4-2). To enable the temperature sensor, the TEMPOFF bit (SLPCON<1>) must be cleared.

FIGURE 9-2: SLOPE REFERENCE DIVIDER



9.5 Comparator and Programmable Reference Modules

9.5.1 COMPARATORS

The PIC14000 includes two independent low-power comparators for comparing the programmable reference outputs to either the RA1/AN1 or RA5/AN5 pins. The negative input of each comparator is tied to one of the reference outputs as shown in Figure 9-3. The comparator positive inputs are connected to the output of the RA1/AN1 and RA5/AN5 level-shift networks.

At reset, the RA1/AN1 level-shift output is connected to the positive inputs of both comparators. This allows a window comparison of the RA1/AN1 voltage using the two programmable references and comparators. Setting CMBOE (CMCON<5>) changes the configuration so that RA1/AN1 and RA5/AN5 may be independently monitored.

The comparator outputs can be read by the CMAOUT (CMCON<2>) and CMBOUT (CMCON<6>) bits. These are read-only bits and writes to these locations have no effect.

Either a rising or falling comparator output can generate an interrupt to the CPU as controlled by the polarity bits CPOLA (CMCON<0>) and CPOLB (CMCON<4>). The CMIF bit (PIR1<7>) interrupt flag is set whenever the exclusive-OR of the comparator output CMxOUT and the CPOLx bits equal a logic one. As with other peripheral interrupts, the corresponding enable bit CMIE (PIE1<7>) must also be set to enable the comparator interrupt. In addition, the global interrupt enable and peripheral interrupt enable bits INTCON<7:6> must also be set. This comparator interrupt is level sensitive.

The comparator outputs are visible at either RC1/CMPA or RD2/CMPB pins by setting the CMAOE (CMCON<1>) or CMBOE (CMCON<5>) bits. Setting CMxOE does not affect the comparator operation. It only enables the pin function regardless of the port TRIS register setting.

Both the references and the comparators are enabled by clearing the CMOFF (SLPCON<2>) bit.

9.5.2 PROGRAMMABLE REFERENCES

The PIC14000 includes two independent, programmable voltage references. Each reference is built using two resistor ladders, bandgap-referenced current source, and analog multiplexers. The first ladder contains 32 taps, and is divided into three ranges (upper, middle, and lower) to provide a coarse voltage adjustment. The coarse ladder includes 1k and 10k resistors yielding a step size of either 5 or 50 mV (nominal) depending on the selected range. Figure 9-8 shows the comparator and reference architecture.

A second ladder contains eight taps, and is connected across the selected coarse ladder resistor to increase resolution. This subdivides the coarse ladder step by approximately 1/8. Thus, resolutions approaching 5/8 mV are obtainable.

Two registers PREFA (9Bh) and PREFB (9Ch) are used to select the reference output voltages. The PREFx<7:3> bits select the output from the coarse ladder, while PREFx<2:0> bits are for the fine-tune adjustment. Table 9-1 and Table 9-2 show the reference decoding.

These voltages are visible at either RC0/REFA or RD3/REFB pins by setting the CMAOE (CMCON<1>) or CMBOE (CMCON<5>) bits. Setting CMxOE does not affect the reference voltages. It only enables the pin function regardless of the port TRIS register setting. These outputs are not buffered, so they cannot directly drive any DC loads.

The reference outputs are also connected to two independent comparators, COMPA and COMPB. Thus, the references can be used to set the comparator trippoints. The A/D converter can also monitor the reference outputs via A/D channels 8 and 9. Refer to Section 8 for the description of the A/D operation.

The programmable reference output is designed to track the output from the level shift network. However, there will always be some mismatch due to component drift. For best accuracy, the A/D should be used to periodically calibrate the references to the desired set-point.

FIGURE 9-3: COMPARATOR AND PROGRAMMABLE REFERENCE BLOCK DIAGRAM (ONE OF TWO SHOWN)

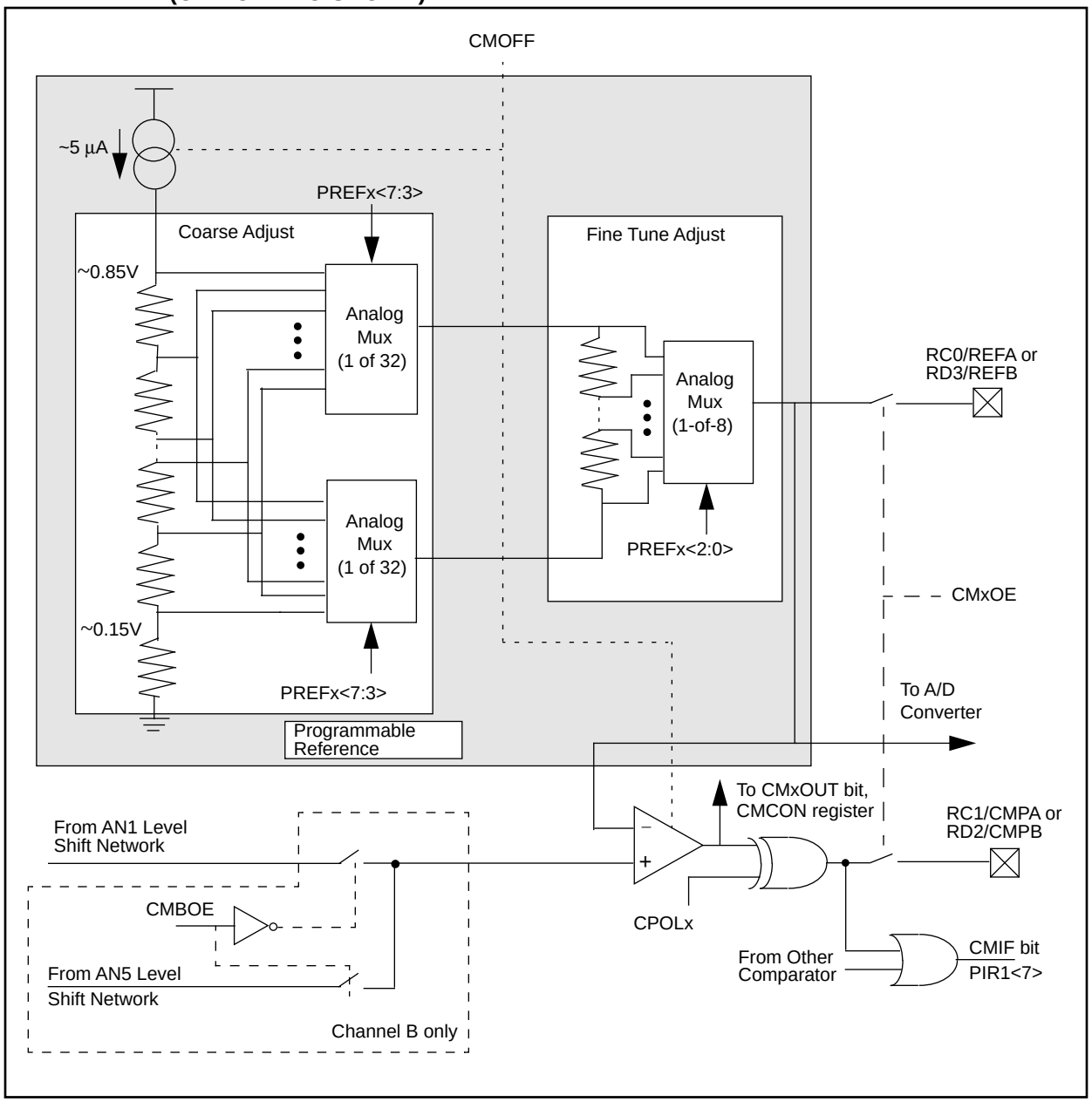


TABLE 9-1: PROGRAMMABLE REFERENCE COARSE RANGE SELECTION

	PREFx<7:3>					Nominal Output Voltage Range (V)
Upper	0	1	1	1	1	0.8000 - 0.8500
	0	1	1	1	0	0.7500 - 0.8000
	0	1	1	0	1	0.7000 - 0.7500
	0	1	1	0	0	0.6500 - 0.7000
	0	1	0	1	1	0.6000 - 0.6500
	0	1	0	1	0	0.5500 - 0.6000
Middle	0	1	0	0	1	0.5450 - 0.5500
	0	1	0	0	0	0.5400 - 0.5450
	0	0	1	1	1	0.5350 - 0.5400
	0	0	1	1	0	0.5300 - 0.5350
	0	0	1	0	1	0.5250 - 0.5300
	0	0	1	0	0	0.5200 - 0.5250
	0	0	0	1	1	0.5150 - 0.5200
	0	0	0	1	0	0.5100 - 0.5150
	0	0	0	0	1	0.5050 - 0.5100
	0	0	0	0	0	0.5000 - 0.5050
	1	0	0	0	0	0.4950 - 0.5000
	1	0	0	0	1	0.4900 - 0.4950
	1	0	0	1	0	0.4850 - 0.4900
	1	0	0	1	1	0.4800 - 0.4850
	1	0	1	0	0	0.4750 - 0.4800
	1	0	1	0	1	0.4700 - 0.4750
	1	0	1	1	0	0.4650 - 0.4700
	1	0	1	1	1	0.4600 - 0.4650
	1	1	0	0	0	0.4550 - 0.4600
	1	1	0	0	1	0.4500 - 0.4550
Lower	1	1	0	1	0	0.4000 - 0.4500
	1	1	0	1	1	0.3500 - 0.4000
	1	1	1	0	0	0.3000 - 0.3500
	1	1	1	0	1	0.2500 - 0.3000
	1	1	1	1	0	0.2000 - 0.2500
	1	1	1	1	1	0.1500 - 0.2000

TABLE 9-2: PROGRAMMABLE REFERENCE FINE RANGE SELECTION

PREFx<2:0>			Fractional Value Of The Coarse Range
0	0	0	1/8
0	0	1	1/4
0	1	0	3/8
0	1	1	1/2
1	0	0	5/8
1	0	1	3/4
1	1	0	7/8
1	1	1	1

FIGURE 9-4: PROGRAMMABLE REFERENCE TRANSFER FUNCTION

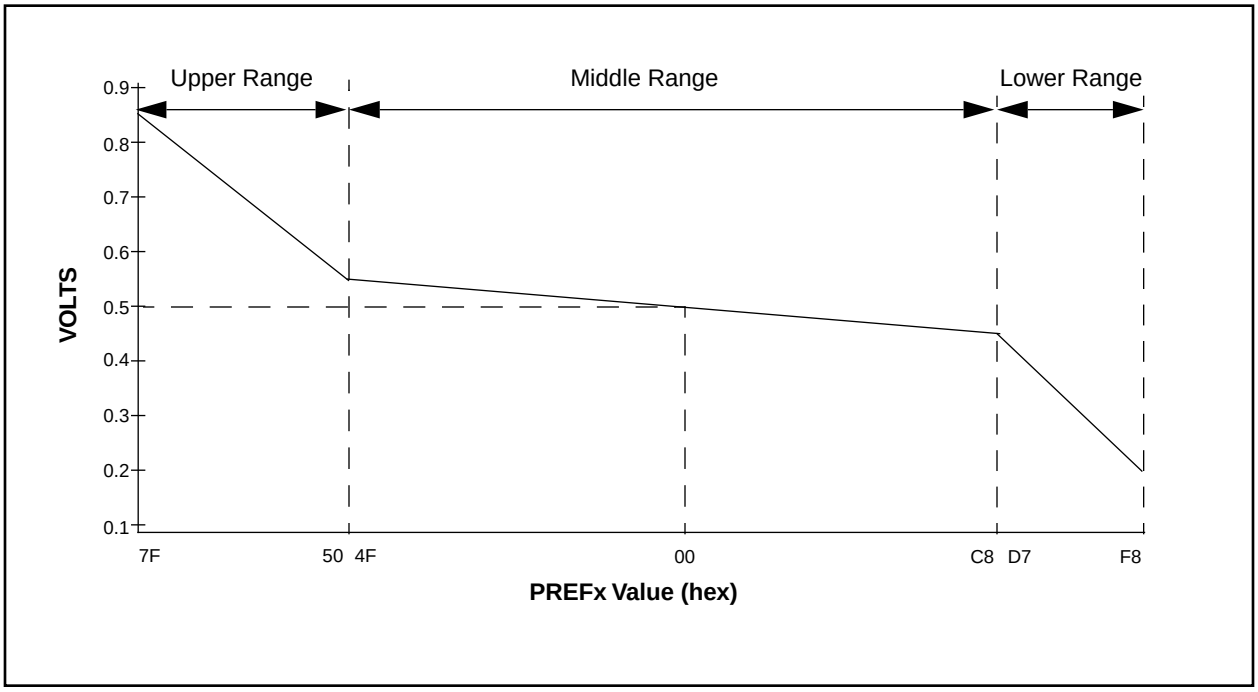


FIGURE 9-5: COMPARATOR CONTROL REGISTER

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMCON	U	CMBOUT	CMBOE	CPOLB	U	CMAOUT	CMAOE	CPOLA
Read/Write	—	R	R/W	R/W	—	R	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

Bit	Name	Function
B7	—	Unimplemented. Read as '0'.
B6	CMBOUT	Comparator B Output Reading this bit returns the status of the comparator B output. Writes to this bit have no effect.
B5	CMBOE	Comparator B Output Enable 1 = Comparator B output is available on RD2/CMPB pin and Reference B output is available on RD3/REFB pin. 0 = RD2/CMPB and RD3/REFB assume normal PORTD function.
B4	CPOLB	Comparator B Polarity Bit 1 = Invert the output of comparator B. 0 = Do not invert the output of comparator B.
B3	—	Unimplemented. Read as '0'.
B2	CMAOUT	Comparator A Output Reading this bit returns the status of the comparator A output. Writes to this bit have no effect.
B1	CMAOE	Comparator A Output Enable 1 = Comparator A output is available on RC1/CMPA pin and Reference A output is available on RC0/REFA pin. 0 = RC0/REFA and RC1/CMPA assume normal PORTC function.
B0	CPOLA	Comparator A Polarity Bit 1 = Invert the output of comparator A. 0 = Do not invert the output of comparator A.

FIGURE 9-6: PREFA REGISTER

9Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PREFA	PRA7	PRA6	PRA5	PRA4	PRA3	PRA2	PRA1	PRA0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

Bit	Name	Function
B7-B0	PRA7 PRA6 PRA5 PRA4 PRA3 PRA2 PRA1 PRA0	Programmable Reference A Voltage Select Bits. See Table 9-1 and Table 9-2 for decoding.

FIGURE 9-7: PREFB REGISTER

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PREFB	PRB7	PRB6	PRB5	PRB4	PRB3	PRB2	PRB1	PRB0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value 00h	0	0	0	0	0	0	0	0

Bit	Name	Function
B7-B0	PRB7 PRB6 PRB5 PRB4 PRB3 PRB2 PRB1 PRB0	Programmable Reference B Voltage Select Bits. See Table 9-1 and Table 9-2 for decoding.

9.6 V

NOTES:

10.0 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 PIC14000 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 (oscillator) selection
  - Crystal/resonator
  - Internal oscillator
- Reset options
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP and HIBERNATE modes
- Code protection
- In-circuit serial programming

These features will be described in the following sections.

10.1 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 10-1: CONFIGURATION WORD

2007h	Bit 13-8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BITS	r	CPC	r	CPP1	CPP0	$\overline{\text{PWRTE}}$	WDTE	r	FOSC
Read/Write	R/W	R/W	Reserved	R/W	R/W	R/W	R/W	Reserved	R/W
Erased value	1	1	1	1	1	1	1	1	1

Bit	Name	Function
B13-B8	r	Reserved
B7	CPC	Calibration Space Code Protection Bit 1 = Calibration space is readable and programmable 0 = Calibration space is write protected
B6	r	Reserved
B5	CPP1	Program Space Code Protection Bit 1 = Program space is readable and programmable 0 = Program space is read/write protected
B4	CPP0	Program Space Code Protection Bit 1 = Program space is readable and programmable 0 = Program space is read/write protected
B3	$\overline{\text{PWRTE}}$	Power-up Timer Enable Bit 1 = Power-up timer is disabled 0 = Power-up timer is enabled
B2	WDTE	Watchdog Timer Enable Bit 1 = WDT is enabled 0 = WDT is disabled
B1	r	Reserved
B0	FOSC	Oscillator Selection Bit 1 = IN oscillator (internal) 0 = HS oscillator (crystal/resonator)

10.2 Oscillator Configurations

The PIC14000 can be operated with two different oscillator options. The user can program a configuration word (CONFIG<0>) to select one of these:

- HS High Speed Crystal/Ceramic Resonator (CONFIG<0> =‘0’)
- IN Internal oscillator (CONFIG<0> =‘1’) (Default)

10.2.1 INTERNAL OSCILLATOR CIRCUIT

The PIC14000 includes an internal oscillator option that offers additional cost and board-space savings. No external components are required. The nominal operating frequency is 4 MHz. The frequency is measured and stored into the calibration space in EPROM.

By selecting IN mode OSC1/PBTN becomes a digital input (with weak internal pull-up resistor) and can be read via bit MISC<0>. Writes to this location have no effect. The OSC1/PBTN input is capable of generating an interrupt to the CPU if enabled (Section 10.6). Also, the OSC2 pin becomes a digital output for general purpose use and is accessed via MISC<1>. Writes to this bit directly affect the OSC2 pin. Reading this bit returns the contents of the output latch. The MISC register format is shown in Figure 10-2.

The OSC2 pin can also output the IN oscillator frequency, divided-by-four, by setting INCLKEN (MISC<2>).

Note: The OSC2 output buffer provides less drive than standard I/O.

FIGURE 10-2: MISC REGISTER

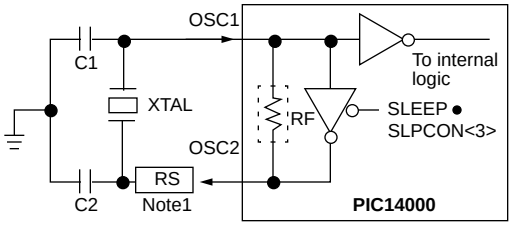
9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MISC	SMHOG	SPGNDB	SPGNDA	I^2CSEL	SMBUS	INCLKEN	OSC2	OSC1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
POR value 00h	0	0	0	0	0	0	0	X

Bit	Name	Function
B7	SMHOG	SMHOG enable 1 = Stretch I^2C CLK signal (hold low) when receive data buffer is full (refer to Section 7.5.4). For pausing I^2C transfers while preventing interruptions of A/D conversions. 0 = Disable I^2C CLK stretch.
B6	SPGNDB	Serial Port Ground Select 1 = PORTD<1:0> ground reference is the RD5/AN5 pin. 0 = PORTD<1:0> ground reference is Vss.
B5	SPGNDA	Serial Port Ground Select 1 = PORTC<7:6> ground reference is the RA1/AN1 pin. 0 = PORTC<7:6> ground reference is Vss.
B4	I^2CSEL	I^2C Port select Bit. 1 = PORTD<1:0> are used as the I^2C clock and data lines. 0 = PORTC<7:6> are used as the I^2C clock and data lines.
B3	SMBus	SMBus-Compatibility Select 1 = SMBus compatibility mode is enabled. PORTC<7:6> and PORTD<1:0> have SMBus-compatible input thresholds. 0 = SMBus-compatibility is disabled. PORTC<7:6> and PORTD<1:0> have Schmitt Trigger input thresholds.
B2	INCLKEN	Oscillator Output Select (available in IN mode only). 1 = Output IN oscillator signal divided by four on OSC2 pin. 0 = Disconnect IN oscillator signal from OSC2 pin.
B1	OSC2	OSC2 output port bit (available in IN mode only). Writes to this location affect the OSC2 pin in IN mode. Reads return the value of the output latch.
B0	OSC1	OSC1 input port bit (available in IN mode only). Reads from this location return the status of the OSC1 pin in IN mode. Writes have no effect.

10.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATOR

In HS mode, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. A parallel cut crystal is required. Use of a series cut crystal may give a frequency outside of the crystal manufacturer's specifications. When in HS mode, the device can have an external clock source to drive the OSC1 pin.

FIGURE 10-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS OSC CONFIGURATION)



See Table 10-1 and Table 10-2 for recommended values of C1 and C2.

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

FIGURE 10-4: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)

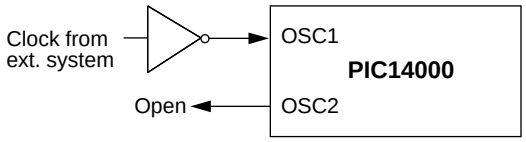


TABLE 10-1: CERAMIC RESONATORS

Mode	Freq	C1	C2
HS	4 MHz	15 - 68 pF	15 - 68 pF
	8 MHz	10 - 68 pF	10 - 68 pF
	16 MHz	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:		
4 MHz	Murata Erie CSA4.00MG	+/-.5%
8 MHz	Murata Erie CSA8.00MT	+/-.5%
16 MHz	Murata Erie CSA16.00MX	+/-.5%

All resonators used did not have built-in capacitors.

TABLE 10-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Mode	Freq	C1	C2
HS	4 MHz	15 - 33 pF	15 - 33 pF
	8 MHz	15 - 47 pF	15 - 47 pF
	20 MHz	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 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 ≈ 30pf is recommended.

10.2.3 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 10-5 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 10-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

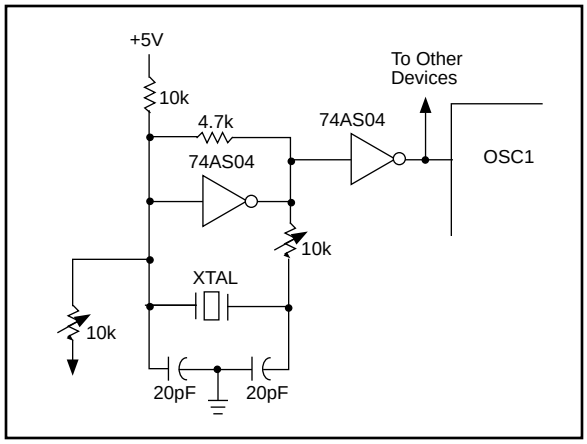
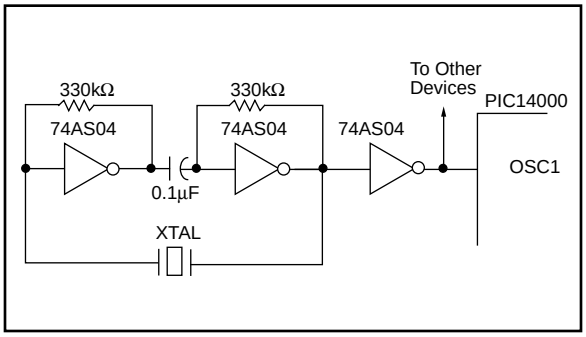


Figure 10-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 10-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



10.3 Reset

The PIC14000 differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{MCLR}$ Reset during normal operation
- $\overline{MCLR}$ Reset during SLEEP
- WDT Reset (normal operation)

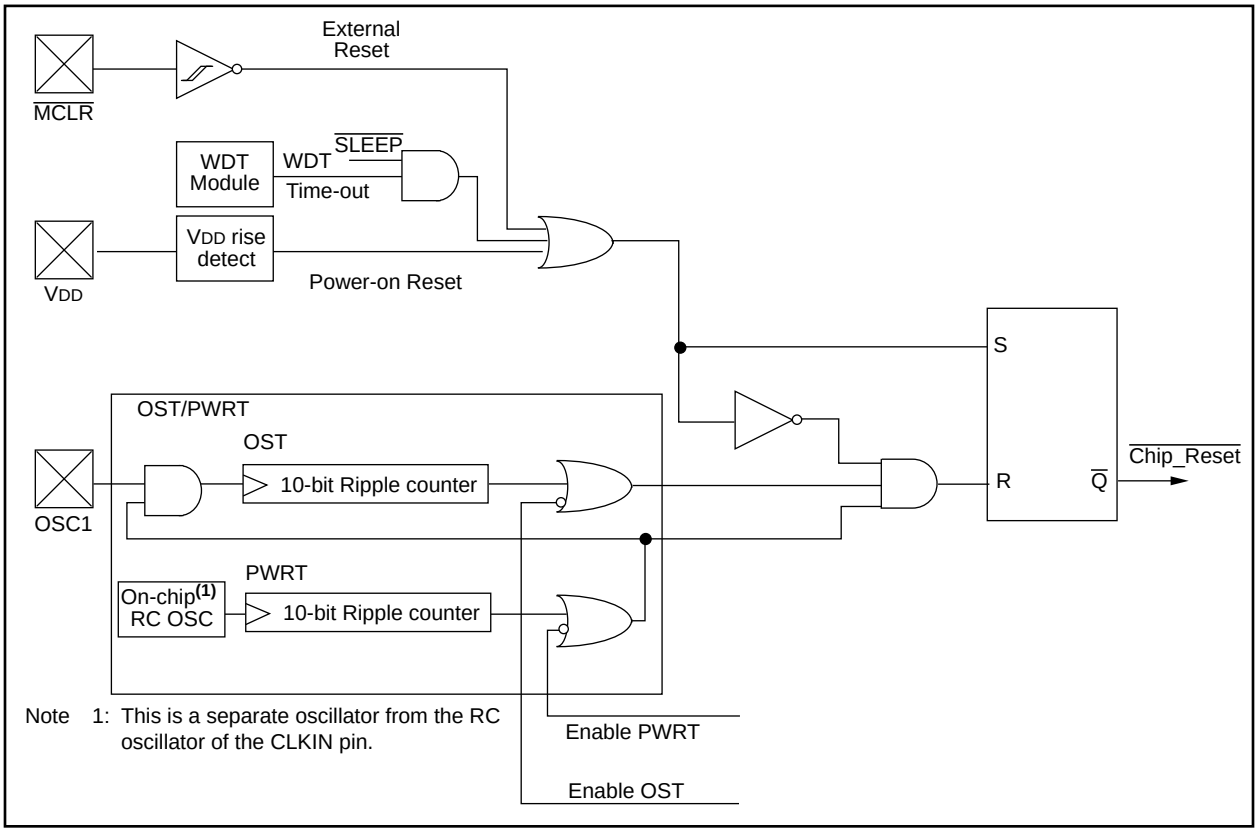
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), on the $\overline{MCLR}$ and WDT Reset, and on $\overline{MCLR}$ Reset during SLEEP. They are not affected by a WDT Wake-up, which 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 10-3. These bits are used in software to determine the nature of the reset. See Table 10-5 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 10-7.

The devices all have a $\overline{MCLR}$ noise filter in the $\overline{MCLR}$ reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{MCLR}$ pin low.

FIGURE 10-7: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.

TABLE 10-3: STATUS BITS AND THEIR SIGNIFICANCE

$\overline{POR}$	$\overline{TO}$	$\overline{PD}$	Meaning
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 time-out wakeup from sleep
1	1	1	$\overline{MCLR}$ reset during normal operation
1	1	0	$\overline{MCLR}$ reset during SLEEP or HIBERNATE, or interrupt wake-up from SLEEP or HIBERNATE.

10.4 Low-Voltage Detector

The PIC14000 contains an integrated low-voltage detector. The supply voltage is divided and compared to the bandgap reference output. If the supply voltage (V_{DD}) falls below V_{TRIP}-, then the low-voltage detector will cause $\overline{LVD}$ (PCON<0>) to be reset. This bit can be read by software to determine if a low voltage condition occurred. This bit must be set by software.

The nominal values of the low-voltage detector trip points are as follows:

- V_{TRIP}- = 2.55V
- V_{TRIP}+ = 2.60V
- Hysteresis (V_{TRIP}+ – V_{TRIP}-) = 55 mV

10.5 Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

10.5.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when V_{DD} rise is detected (in the range of 1.5V - 2.1V). To take advantage of the POR, just tie the $\overline{MCLR}$ pin directly (or through a resistor) to V_{DD}. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for V_{DD} is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, ...) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting*."

10.5.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) time-out on power-up only, from $\overline{POR}$. The power-up timer operates from a local internal oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the V_{DD} to rise to an acceptable level. A configuration bit, $\overline{PWRTE}$, can disable (if set, or unprogrammed) or enable (if cleared, or programmed) the power-up timer.

The power-up timer delay will vary from chip to chip and due to V_{DD} and temperature.

10.5.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over. This guarantees that the crystal oscillator or resonator has started and stabilized.

10.5.4 IN OSCILLATOR START-UP

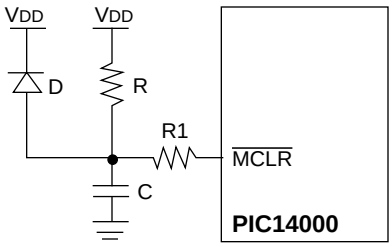
There is an 8-cycle delay in IN mode to ensure stability only after a Power-on Reset (POR) or wake-up from SLEEP.

10.5.5 TIMEOUT SEQUENCE

On power-up the time-out sequence is as follows: First the PWRT time-out is invoked after $\overline{POR}$ has expired. The OST is activated only in HS (crystal oscillator) mode. The total time-out will vary based on the oscillator configuration and $\overline{PWRTE}$ status. For example, in IN mode, with $\overline{PWRTE}$ unprogrammed (PWRT disabled), there will be no time-out delay at all. Figure 13-4 depicts the power-on reset time-out sequences.

Table 10-4 shows the reset conditions for some special registers, while Table 10-5 shows the reset conditions for all registers.

FIGURE 10-8: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



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 < 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 $\overline{MCLR}$ pin.
3. R1 = 100 Ω 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 ESD or EOS.

TABLE 10-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	PCL Addr: 02h	STATUS Addr: 03h	PCON Addr: 8Eh
Power-on Reset	000h	0001 1xxx	0--- --0x
$\overline{MCLR}$ reset during normal operation	000h	0001 1uuu	u--- --ux
$\overline{MCLR}$ reset during SLEEP	000h	0001 0uuu	u--- --ux
WDT reset during normal operation	000h	0000 1uuu	u--- --ux
WDT during SLEEP	PC + 1	uuu0 0uuu	u--- --ux
Interrupt wake-up from SLEEP	PC + 1(1)	uuu1 0uuu	u--- --ux

Legend: u = unchanged
x = unknown
- = unimplemented, 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 10-5: RESET CONDITIONS FOR REGISTERS

Register	Address	Power-on Reset	MCLR reset during - normal operation - SLEEP WDT time-out during normal operation	Wake-up from SLEEP through interrupt Wake up from SLEEP through WDT time-out
W	-	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h	-	-	-
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h	0000h	0000h	PC + 1(2)
STATUS	03h/83h	0001 1xxx	000? ?uuu(3)	uuu? ?uuu(3)
FSR	04h/84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---- xxxx	---- uuuu	---- uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah/8Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh	0000 000x	0000 000u	uuuu uuuu(1)
PIR1	0Ch	0000 0000	0000 0000	uuuu uuuu(1)
ADTMRL	0Eh	0000 0000	0000 0000	uuuu uuuu
ADTMRH	0Fh	0000 0000	0000 0000	uuuu uuuu
I2CBUF	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
I2CCON	14h	0000 0000	0000 0000	uuuu uuuu
ADCAPL	15h	0000 0000	0000 0000	uuuu uuuu
ADCAPH	16h	0000 0000	0000 0000	uuuu uuuu
ADCON0	1Fh	0000 0010	0000 0010	uuuu uuuu
OPTION	81h			

10.6 Interrupts

The PIC14000 has several sources of interrupt:

- External interrupt from OSC1/PBTN pin
- I^2C port interrupt
- PORTC interrupt on change (pins RC<7:4> only)
- Timer0 overflow
- A/D timer overflow
- A/D converter capture event
- Programmable reference comparator interrupt

This section addresses the external and Timer0 interrupts only. Refer to the appropriate sections for description of the serial port, programmable reference and A/D interrupts.

INTCON records individual interrupt requests in flag bits. It also has individual and global enable bits. The peripheral interrupt flags reside in the PIR1 register. Peripheral interrupt enable interrupts are contained in the PIE1 register.

Global interrupt masking is controlled by GIE (INTCON<7>). Individual interrupts can be disabled through their corresponding mask bit in the INTCON register. GIE is cleared on reset to mask interrupts.

When an interrupt is serviced, the GIE is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h, the interrupt vector. For external interrupt events, such as the I^2C interrupt, the interrupt latency will be 3 or 4 instruction cycles. The exact latency depends when the interrupt event occurs. The latency is the same for 1 or 2 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 to allow polling.

The return from interrupt instruction, RETFIE, exits the interrupt routine as well as sets the GIE bit to re-enable interrupts.

Note 1: The individual interrupt flags will be set by the specified condition even though the corresponding interrupt enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled).

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:

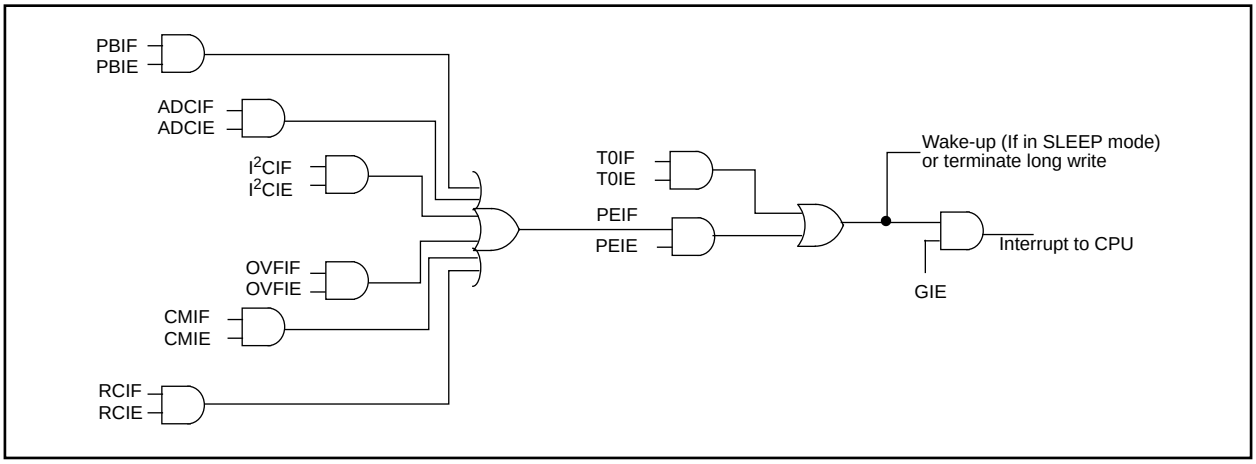
1. An instruction clears the GIE bit while an interrupt is acknowledged.
2. The program branches to the interrupt vector and executes the Interrupt Service Routine.
3. The interrupt service routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (enables interrupts), 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
```

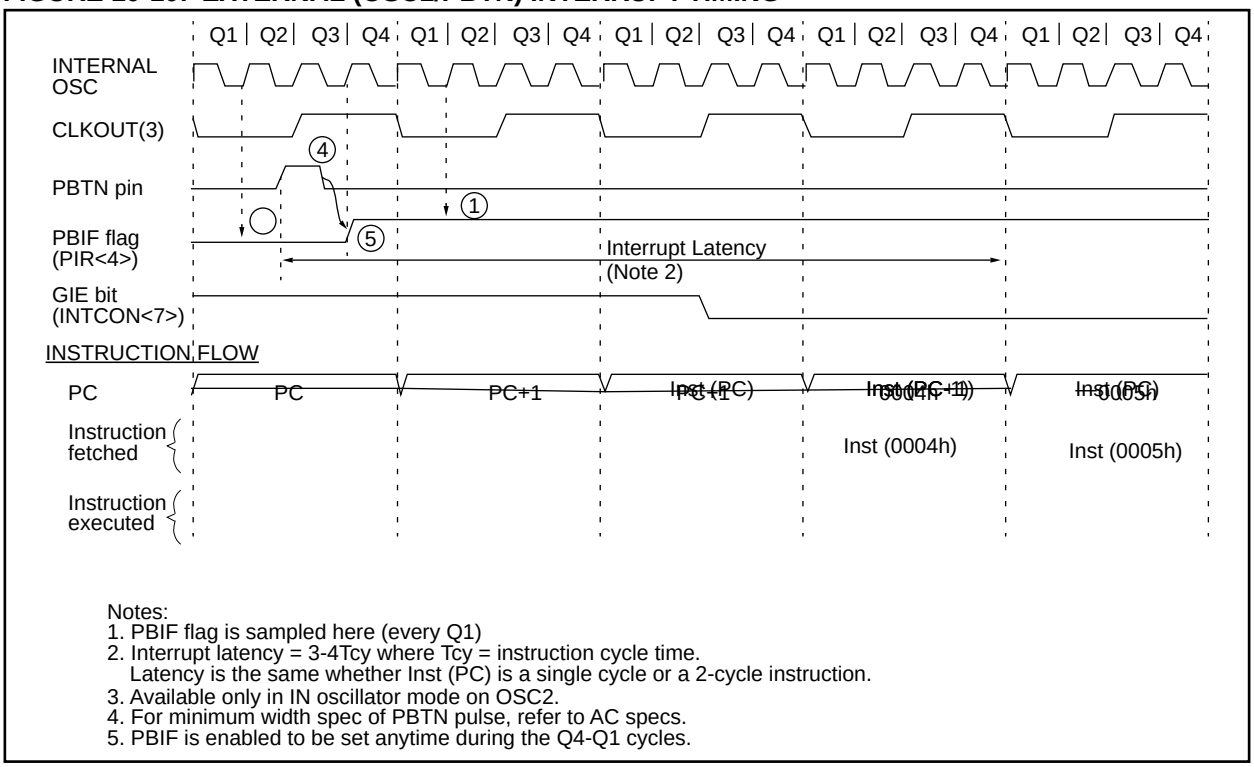
FIGURE 10-9: INTERRUPT LOGIC SCHEMATIC



10.6.1 EXTERNAL INTERRUPT

An external interrupt can be generated via the OSC1/PBTN pin if IN (internal oscillator) mode is enabled. This interrupt is falling edge triggered. When a valid edge appears on OSC1/PBTN pin, PBIF (PIR1<4>) is set. This interrupt can be disabled by clearing PBIE (PIE1<4>). PBIF must be cleared in software in the interrupt service routine before re-enabling the interrupt. This interrupt can wake up the processor from SLEEP if PBIE bit is set (interrupt enabled) prior to going into SLEEP mode. The status of the GIE bit determines whether or not the processor branches to the interrupt vector following wake-up. The timing of the external interrupt is shown in Figure 10-10.

FIGURE 10-10: EXTERNAL (OSC1/PBTN) INTERRUPT TIMING



Notes:
1. PBIF 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. Available only in IN oscillator mode on OSC2.
4. For minimum width spec of PBTN pulse, refer to AC specs.
5. PBIF is enabled to be set anytime during the Q4-Q1 cycles.

10.6.2 TIMER0 INTERRUPT

An overflow (FFh → 00h) in Timer0 will set the T0IF (INTCON<2>) flag. Setting T0IE (INTCON<5>) enables the interrupt.

10.6.3 PORTC INTERRUPT ON CHANGE

An input change on PORTC<7:4> sets RCIF (PIR1<2>). Setting RCIE (PIE1<2>) enables the interrupt. For operation of PORTC, refer to 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 RCIF interrupt flag may not be set.

10.6.4 CONTEXT SWITCHING 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, for example, W register and Status register. Example 10-1 is an example that shows saving registers in RAM.

EXAMPLE 10-1: SAVING STATUS AND W REGISTERS IN RAM

```
MOVWF    W_TEMP           ;Copy W to TEMP register, could be any bank
SWAPF    STATUS,W         ;Swap status to be saved into W
BCF      STATUS,RP1       ;Change to bank zero, regardless of current bank
BCF      STATUS,RP0       ;
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
```


10.7 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 IN oscillator used to generate the CPU and A/D clocks. That means that the WDT will run even if the clock has been stopped, for example, by execution of a SLEEP instruction. Refer to Section 10.8.1 for more information.

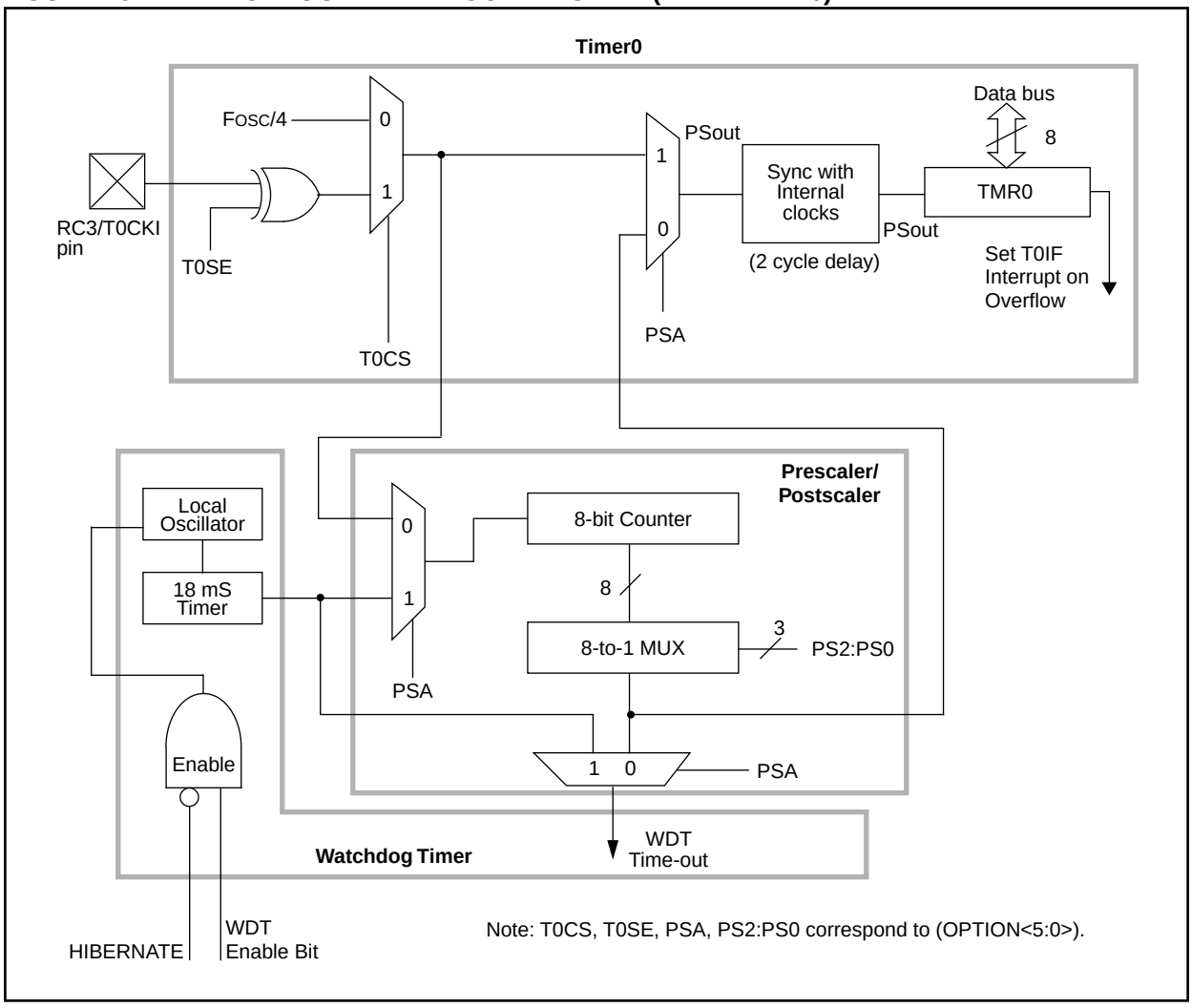
During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation.

The WDT can be permanently disabled by programming the configuration bit WDTE as a '0'. Its oscillator can be shut down to conserve battery power by entering HIBERNATE Mode. Refer to Section 10.8.3 for more information on HIBERNATE mode.

CAUTION: Beware of disabling WDT if software routines require exiting based on WDT reset. For example, the MCU will not exit HIBERNATE mode based on WDT reset.

A block diagram of the watchdog timer is shown in Figure 10-11. It should be noted that a RESET generated by the WDT time-out does not drive $\overline{MCLR}$ low.

FIGURE 10-11: WATCHDOG TIMER BLOCK DIAGRAM (WITH TIMER0)



10.7.1 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 (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 registers. Thus, time-out periods up to 2.3 seconds can be realized. The CLRWDT and SLEEP instructions clear the WDT and the prescaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

The $\overline{TO}$ bit in the status register will be cleared upon a watchdog timer time-out. The WDT time-out period (no prescaler) is measured and stored in calibration space at location 0FD2h.

10.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst-case conditions (minimum VDD, maximum temperature, maximum WDT prescaler) it may take several seconds before a WDT time-out occurs. Refer to Section 6.3 for prescaler switching considerations.

10.8 Power Management Options

The PIC14000 has several power management options to prolong battery lifetime. The SLEEP instruction halts the CPU and can turn off the on-chip oscillators. The CPU can be in SLEEP mode, yet the A/D converter can continue to run. Several bits are included in the SLPCON register (8Fh) to control power to analog modules.

TABLE 10-6: SUMMARY OF POWER MANAGEMENT OPTIONS

Function	Summary
CPU Clock	OFF during SLEEP/HIBERNATE mode, ON otherwise
Main Oscillator	ON if NOT in SLEEP mode. In SLEEP mode, controlled by OSCOFF bit, SLPCON<3>.
Watchdog Timer	Controlled by WDTE, 2007h<2> and HIBEN, SLPCON<7>
Temperature Sensor	Controlled by TEMPOFF, SLPCON<1>
Low-voltage Detector	Controlled by REFOFF, SLPCON<5>
Comparator and Programmable References	Controlled by CMOFF, SLPCON<2>
A/D Comparator	Controlled by ADOFF, SLPCON<0>
Programmable Current Source	Controlled by ADOFF, SLPCON<0> and ADCON1<7:4>
Slope Reference Voltage Divider	Controlled by ADOFF, SLPCON<0>
Level Shift Networks	Controlled by LSOFF, SLPCON<4>
Bandgap Reference	Controlled by REFOFF, SLPCON<5>
Voltage Regulator Control	Always ON. Does not consume power if unconnected.
Power On Reset	Always ON, except in SLEEP/HIBERNATE mode

Note: Refer to analog specs for individual peripheral operating currents.

10.8.1 SLEEP MODE

The SLEEP mode is entered by executing a SLEEP instruction.

If SLEEP mode is enabled, the WDT will be cleared but keep running. The $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set, and on-chip oscillators are shut off, except the WDT RC oscillator, which continues to run. The I/O ports maintain the status they had before the SLEEP command was executed (driving high, low, or high-impedance).

It is an option while in SLEEP mode to leave the on-chip oscillator running. This option allows an A/D conversion to continue while the CPU is in SLEEP mode. The CPU clocks are stopped in this condition to preserve power. The operation of the on-chip oscillator during SLEEP is controlled by OSCOFF (SLPCON<3>). Clearing this bit to '0' allows the oscillator to continue to run. This bit is only active in SLEEP mode.

For lowest power consumption in this mode, all I/O pins should be either at V_{DD} or V_{SS} with no external circuitry drawing current from the I/O pin. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid leakage currents caused by floating inputs. The $\overline{MCLR}$ pin must be at a logic high level (V_{IH}). The contribution from any on-chip pull-up resistors should be considered.

10.8.2 WAKE-UP FROM SLEEP

The PIC14000 can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin
2. Watchdog Timer time-out (if WDT is enabled)
3. Interrupt from OSC1/PBTN pin
4. RC<7:4> port change
5. I^2C (serial port) start/stop bit detect interrupt.
6. Wake-up on programmable reference comparator interrupt.
7. A/D conversion complete (comparator trip) interrupt.
8. A/D timer overflow interrupt.

An external reset on $\overline{MCLR}$ pin causes a device reset. The other wake-up 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. 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 a 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. Wake-up occurs regardless of the state of bit GIE. If bit GIE is clear, the device continues execution at the instruction after the SLEEP instruction. If bit GIE is set, 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.

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

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.

10.8.3 HIBERNATE MODE

HIBERNATE mode is an extension of SLEEP mode with the following additions.

- WDT is forced off
- Weak pull-ups on RC<5:0> are disabled
- Some input buffers are gated-off (refer to Section 5.0)

The HIBERNATE mode is entered by executing a SLEEP instruction with HIBEN (SLPCON<7>) bit set.

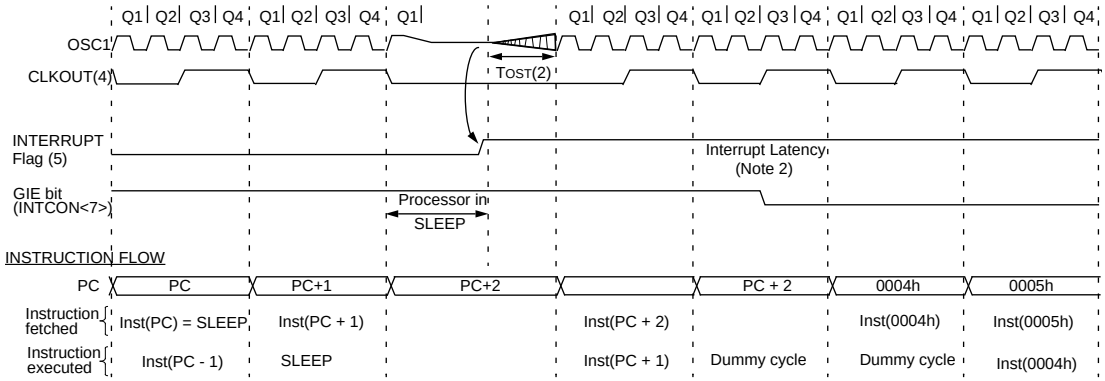
The PIC14000 wakes up from HIBERNATE mode via all the same mechanisms as SLEEP mode, except for WDT time-out. HIBERNATE mode allows power consumption to be reduced to a minimum.

FIGURE 10-12: SLPCON REGISTER

8Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SLPCON	HIBEN	—	REFOFF	LSOFF	OSCOFF	CMOFF	TEMPOFF	ADOFF
Read/Write	R/W	U	R/W	R/W	R/W	R/W	R/W	R/W
POR value 3Fh	0	0	1	1	1	1	1	1

Bit	Name	Function
B7	HIBEN	Hibernate Mode Select 1 = Hibernate mode enable 0 = Normal operating mode
B6	–	Unimplemented. Read as '0'
B5	REFOFF	References Power Control (bandgap reference, low voltage detector, bias generator) 1 = The references are off 0 = The references are on
B4	LSOFF	Level Shift Network Power Control 1 = The level shift network is off. The RA1/AN1, RD5/AN5 inputs can continue to function as either analog or digital. 0 = The level shift network is on. The signals at the RA1/AN1, RD5/AN5 inputs are level shifted by approximately 0.5V.
B3	OSCOFF	Main Oscillator Power Control 1 = The main oscillator is disabled during SLEEP mode 0 = The main oscillator is running during SLEEP mode for A/D conversions to continue
B2	CMOFF	Programmable Reference and Comparator Power Control 1 = The programmable reference and comparator circuits are off 0 = The programmable reference and comparator circuits are on
B1	TEMPOFF	On-chip Temperature Sensor Power Control 1 = The temperature sensor is off 0 = The temperature sensor is on
B0	ADOFF	A/D Module Power Control (comparator, programmable current source, slope reference voltage divider) 1 = The A/D module power is off 0 = The A/D module power is on

FIGURE 10-13: WAKE-UP FROM SLEEP AND HIBERNATE THROUGH INTERRUPT



Note 1: HS oscillator mode assumed.
2: TOST = 1024 TOSC (drawing not to scale). This delay will be 8 TOSC for IN 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.
5: Refer to Section 10.8 for sources.

10.9 Code Protection

The code in the program memory can be protected by programming the code protect bits. When code protected, the contents of the program memory cannot be read out. In code-protected mode, the configuration word (2007h) will not be scrambled, allowing reading of all configuration bits.

10.10 In-Circuit Serial Programming

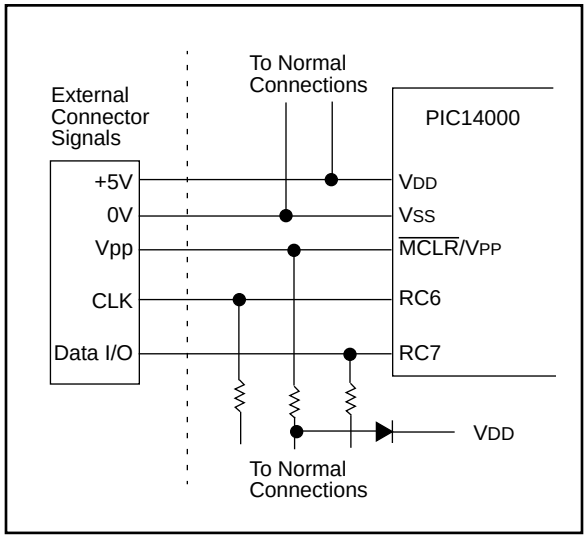
PIC14000 can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RC6/SCL and RC7/SDA pins low while raising the $\overline{MCLR}$ (V_{PP}) pin from V_{IL} to V_{IH}. RC6 then becomes the programming clock and RC7 becomes the programmed data. Both RC6 and RC7 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. For complete details about serial programming, please refer to the *PIC16C6X/7X Programming Specifications* (Literature #DS30228).

A typical in-system serial programming connection is shown in Figure 10-14.

FIGURE 10-14: TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION



NOTES:

11.0 INSTRUCTION SET SUMMARY

The PIC14000's instruction set is the same as PIC16CXX. Each 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 instruction set summary in Table 11-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 11-1 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 used 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 11-1: 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
$\rightarrow$	Assigned to
< >	Register bit field
$\in$	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 µs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 11-2 lists the instructions recognized by the MPASM assembler.

Figure 11-1 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 11-1: GENERAL FORMAT FOR INSTRUCTIONS

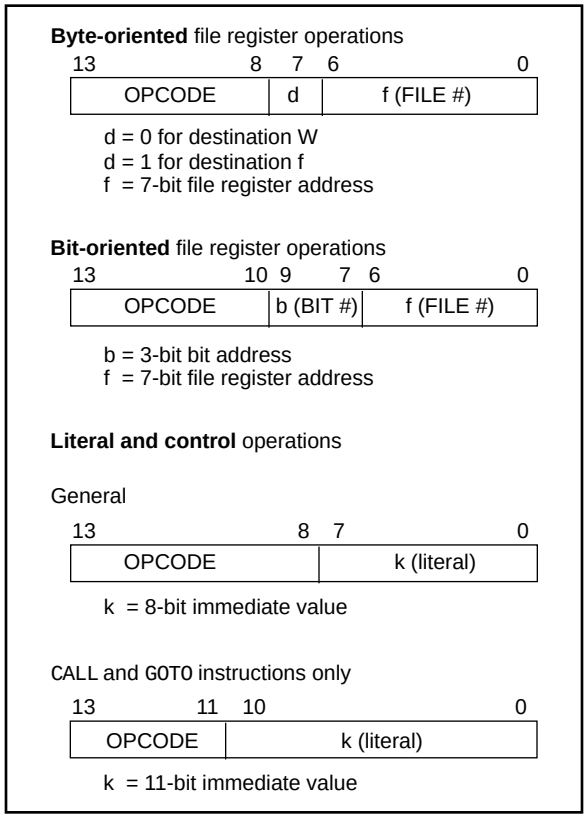


TABLE 11-2: PIC14000 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W with 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 with 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 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		1,2
XORWF	**f, d**	Exclusive OR W with 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 and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal with 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 with 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**	Exclusive OR literal with 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'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

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.

11.1 Instruction Descriptions

ADDLW **Add Literal and 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

ANDLW **And Literal with 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

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 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
```
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: 00 0101 dfff ffff

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W

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

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 **Call Subroutine**

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: Call Subroutine. 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**

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

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 prescaler = 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

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: $0 \le k \le 2047$

Operation: $k \rightarrow PC<10:0>$
$PCLATH<4:3> \rightarrow PC<12:11>$

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 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 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 is 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
Z = 1

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:

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
Z = 1

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 a destination dependant upon the status of 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
Z = 1

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

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: Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). 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 with Literal in 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 k8

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

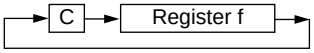
Operation: See description below

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    =   1110 0110
C       =   0
```

After Instruction

```
REG1    =   1110 0110
W       =   0111 0011
C       =   0
```

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f,d

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

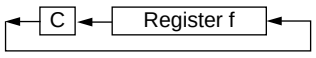
Operation: See description below

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

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 10.8 for more details.

Words: 1

Cycles: 1

Example: SLEEP

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 = 0xFF
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 method) 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 = 0xFF
W    = 2
C    = 0; result is negative
```

SWAPF **Swap Nibbles in 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:

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

```
SWAPF  REG,   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:

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

12.0 DEVELOPMENT SUPPORT

12.1 Development Tools

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

- PICMASTER/PICMASTER CE Real-Time In-Circuit Emulator
- ICEPIC Low-Cost PIC16C5X and PIC16CXX In-Circuit Emulator
- PRO MATE™ II Universal Programmer
- PICSTART® Plus Entry-Level Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB-SIM Software Simulator
- MPLAB-C (C Compiler)
- Fuzzy logic development system (fuzzyTECH®–MP)

12.2 PICMASTER: High Performance Universal In-Circuit Emulator with MPLAB IDE

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 PIC12C5XX, PIC14000, PIC16C5X, PIC16CXX and PIC17CXX families. PICMASTER is supplied with the MPLAB™ Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

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 Microchip microcontrollers.

The PICMASTER Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC compatible 386 (and higher) machine platform and Microsoft Windows® 3.x environment were chosen to best make these features available to you, the end user.

A CE compliant version of PICMASTER is available for European Union (EU) countries.

12.3 ICEPIC: Low-cost PIC16CXX In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC16C5X and PIC16CXX families of 8-bit OTP microcontrollers.

ICEPIC is designed to operate on PC-compatible machines ranging from 286-AT® through Pentium™ based machines under Windows 3.x environment. ICEPIC features real time, non-intrusive emulation.

12.4 PRO MATE II: Universal Programmer

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

The PRO MATE II has programmable V_{DD} and V_{PP} supplies which allows it to verify programmed memory at V_{DD} min and V_{DD} 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 II can read, verify or program PIC16C5X, PIC16CXX, PIC17CXX and PIC14000 devices. It can also set configuration and code-protect bits in this mode.

12.5 PICSTART Plus Entry Level Development System

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. PICSTART Plus is not recommended for production programming.

PICSTART Plus supports all PIC12C5XX, PIC14000, PIC16C5X, PIC16CXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923 and PIC16C924 may be supported with an adapter socket.

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

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 rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

12.11 Software Simulator (MPLAB-SIM)

The MPLAB-SIM 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.

MPLAB-SIM fully supports symbolic debugging using MPLAB-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.

12.12 C Compiler (MPLAB-C)

The MPLAB-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 MPLAB IDE memory display (PICMASTER emulator software versions 1.13 and later).

12.13 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.

12.14 MP-DriveWay™ – Application Code Generator

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

12.15 SEEVAL® Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product 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.

12.16 TrueGauge® Intelligent Battery Management

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

12.17 KEELOQ® Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

TABLE 12-1: DEVELOPMENT TOOLS FROM MICROCHIP

Product	** MPLAB™ Integrated Development Environment	MPLAB™ C Compiler	MP-DriveWay Applications Code Generator	fuzzyTECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool	*** PICMASTER®/ PICMASTER-CE In-Circuit Emulator	ICEPIC Low-Cost In-Circuit Emulator	****PRO MATE™ II Universal Microchip Programmer	PICSTART® Lite Ultra Low-Cost Dev. Kit	PICSTART® Plus Low-Cost Universal Dev. Kit
PIC12C508, 509	SW007002	SW006005	—	—	EM167015/ EM167101	—	DV007003	—	DV003001
PIC14000	SW007002	SW006005	—	—	EM147001/ EM147101	—	DV007003	—	DV003001
PIC16C52, 54, 54A, 55, 56, 57, 58A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167015/ EM167101	EM167201	DV007003	DV162003	DV003001
PIC16C554, 556, 558	SW007002	SW006005	—	DV005001/ DV005002	EM167033/ EM167113	—-	DV007003	—	DV003001
PIC16C61	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167021/ N/A	EM167205	DV007003	DV162003	DV003001
PIC16C62, 62A, 64, 64A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167203	DV007003	DV162002	DV003001
PIC16C620, 621, 622	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167023/ EM167109	EM167202	DV007003	DV162003	DV003001
PIC16C63, 65, 65A, 73, 73A, 74, 74A	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167025/ EM167103	EM167204	DV007003	DV162002	DV003001
PIC16C642, 662*	SW007002	SW006005	—	—	EM167035/ EM167105	—-	DV007003	DV162002	DV003001
PIC16C71	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	EM167205	DV007003	DV162003	DV003001
PIC16C710, 711	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167027/ EM167105	—	DV007003	DV162003	DV003001
PIC16C72	SW007002	SW006005	SW006006	—	EM167025/ EM167103	—	DV007003	DV162002	DV003001
PIC16F83	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	EM167206	DV007003	DV162003	DV003001
PIC16F84	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167029/ EM167107	—	DV007003	DV162003	DV003001
PIC16C923, 924*	SW007002	SW006005	SW006006	DV005001/ DV005002	EM167031/ EM167111	—	DV007003	—	DV003001
PIC17C42, 42A, 43, 44	SW007002	SW006005	SW006006	DV005001/ DV005002	EM177007/ EM177107	—	DV007003	—	DV003001

*Contact Microchip Technology for availability date
**MPLAB Integrated Development Environment includes MPLAB-SIM Simulator and MPASM Assembler
***All PICMASTER and PICMASTER-CE ordering part numbers above include PRO MATE II programmer
****PRO MATE socket modules are ordered separately. See development systems ordering guide for specific ordering part numbers

Product	TRUEGAUGE® Development Kit	SEEVAL® Designers Kit	Hopping Code Security Programmer Kit	Hopping Code Security Eval/Demo Kit
All 2 wire and 3 wire Serial EEPROM's	N/A	DV243001	N/A	N/A
MTA11200B	DV114001	N/A	N/A	N/A
HCS200, 300, 301 *	N/A	N/A	PG306001	DM303001

13.0 ELECTRICAL CHARACTERISTICS FOR PIC14000 ABSOLUTE MAXIMUM RATINGS †

Ambient temperature under bias ..... -55°C to+ 125°C

Storage Temperature ..... -65°C to +150°C

Voltage on any pin with respect to V_{SS} (except V_{DD} and $\overline{MCLR}$) ..... -0.5V to V_{DD} +0.6V

Voltage on V_{DD} with respect to V_{SS} ..... 0 to +6.0 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_O <0 or V_O>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, PORTC, and PORTD(combined) 200mA

Maximum Current sourced by PORTA, PORTC, 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: $Pdis = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 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 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.

13.1 DC Characteristics: PIC14000

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 = 2.7V to 6.0V				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Supply Voltage	VDD	2.7	—	6.0	V	IN or HS at Fosc ≤ 4 MHz
		4.5	—	5.5	V	HS at Fosc > 4 MHz
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
Operating Current in SLEEP Mode (Note 2)						
During A/D conversion: all analog on and internal oscillator active	IPD1	—	TBD	900	µA	VDD = 3.0V
	IPD1	—	TBD	1250	µA	VDD = 4.0V
Comparator interrupt enabled: level-shift, programmable reference, and comparator active	IPD2	—	75	100	µA	VDD = 3.0V, CMOFF = 0, LSOFF = 0, REFOFF = 0
	IPD2	—	95	125	µA	VDD = 4.0V, CMOFF = 0, LSOFF = 0, REFOFF = 0
All analog off, WDT on (Note 5)	IPD3	—	7.5	20	µA	VDD = 3.0V
	IPD3	—	10.5	28	µA	VDD = 4.0V
All analog off, WDT off (Hibernate mode) (Note 5)	IPD4	—	0.9	12	µA	VDD = 3.0V
	IPD4	—	1.5	16	µA	VDD = 4.0V
Operating Supply Current (Note 2, 4)						
Internal oscillator mode	I_{DD}	—	2.2	TBD	mA	Fosc = 4 MHz, VDD = 5.5V
		—	1.1	TBD	mA	Fosc = 4 MHz, VDD = 3.0V
HS oscillator mode		—	2.4	TBD	mA	Fosc = 4 MHz, VDD = 5.5V
		—	1.2	TBD	mA	Fosc = 4 MHz, VDD = 3.0V
		—	10	TBD	mA	Fosc = 20 MHz, VDD = 5.5V

* 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: Measured with all inputs at rails, no DC loads. IPD1 measured with internal oscillator active.

4: IDD values of individual analog module cannot be tested independently but are characterized.

5: Worst-case IPD conditions with all configuration bits unprogrammed. Programming configuration bits may reduce IPD.

13.2 DC Characteristics: PIC14000

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 13.1.				
Characteristic	**Sym**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
Input Low Voltage						
I/O ports	VIL					
Schmitt Trigger mode		VSS	—	0.2VDD	V	
SMBus mode (RC7, RC6, RD0, RD1)		VSS	—	0.6	V	SMBus bit, MISC<3> = 1
$\overline{MCLR}$, OSC1 (in IN mode)		Vss	—	0.2VDD	V	
OSC1 (in HS mode)		Vss	—	0.3VDD	V	
Input High Voltage						
I/O ports	VIH		—			
Schmitt Trigger mode		0.85 VDD	—	VDD	V	
SMBus mode (RC7, RC6, RD0, RD1)		1.4V	—	VDD	V	SMBus bit, MISC<3> = 1
PORTC<5:0> weak pull-up current	IPURC	50	200	†400	µA	VDD = 5V, VPIN = VSS
Input Leakage Current (Notes 1,2)						
I/O ports, CDAC	IIL			±1	µA	Vss ≤ VPIN ≤ VDD, Pin at hi-impedance
$\overline{MCLR}$				±5	µA	Vss ≤ VPIN ≤ VDD
OSC1				±5	µA	Vss ≤ VPIN ≤ VDD
Output Low Voltage						
I/O ports	VOL	—	—	0.6	V	IOL = 8.5mA, VDD-4.5V, -40°C to +85°C
OSC2		—	—	0.6	V	IOL = 1.6mA, VDD-4.5V, -40°C to +85°C
Output High Voltage						
I/O ports (Note 2)	VOH	VDD-0.7	—	—	V	IOH = -3.0mA, VDD=4.5V, -40°C to +85°C
RC6, RC7, RD0, RD1 (except I^2C mode)		2.4	—	—	V	IOH = -2.0mA, VDD=4.5V, -40°C to +85°C
OSC2		VDD-0.7	—	—	V	IOH = -1.3mA, VDD=4.5V, -40°C to +85°C
Capacitive Loading Specs on Output Pins						
OSC2 pin	COSC2			15	pF	
All I/O pins except OSC2 (in IN 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: 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.

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

13.3 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			
ck	CLKOUT	osc	OSC1
di	SDI	t0	T0CKI
io	I/O port		
mc	$\overline{MCLR}$		

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		

13.4 Timing Diagrams and Specifications

FIGURE 13-1: EXTERNAL CLOCK TIMING

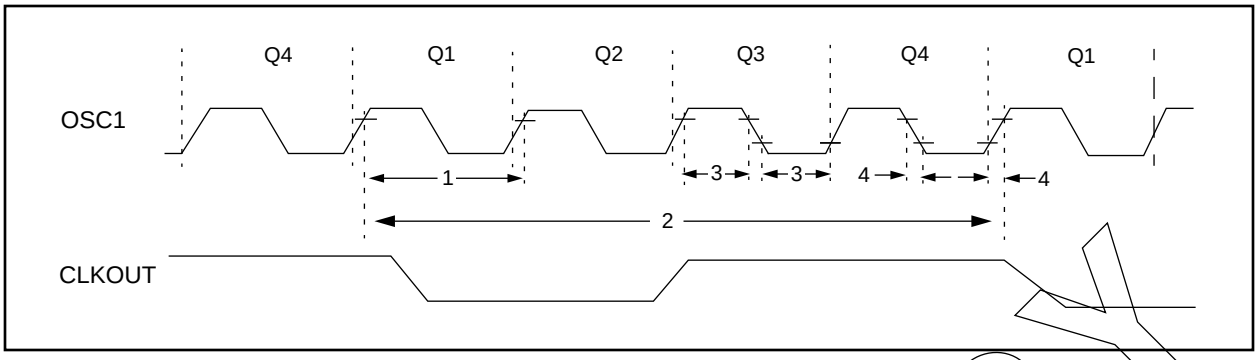


TABLE 13-1: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	FOSC	External CLKIN Frequency (Note 1)	DC DC	— —	4 20	MHz MHz	HS osc mode (PIC14000-04) HS osc mode (PIC14000-20)
		Oscillator Frequency (Note 1)	4 4	— —	4 20	MHz MHz	HS osc mode (PIC14000-04) HS osc mode (PIC14000-20)
1	TOSC	External CLKIN Period (Note 1)	250 50	— —	— —	ns ns	HS osc mode (PIC14000-04) HS osc mode (PIC14000-20)
		Oscillator Period (Note 1)	250 50	— —	250 250	ns ns	HS osc mode (PIC14000-04) HS osc mode (PIC14000-20)
2	TCY	Instruction Cycle Time (Note 1)	200	—	DC	ns	TCY = 4/FOSC
3	TOSL, TOSH	Clock in (OSC1) High or Low Time	10	—	—	ns	HS oscillator
4	TOSR, TOSF	Clock in (OSC1) Rise or Fall Time	—	—	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.

FIGURE 13-2: LOAD CONDITIONS

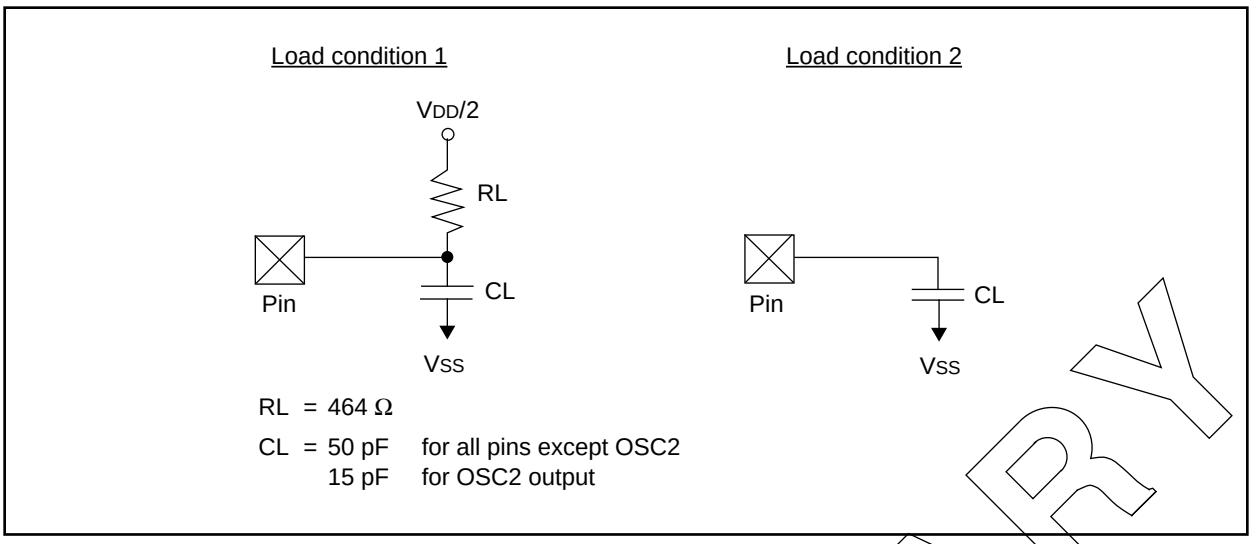
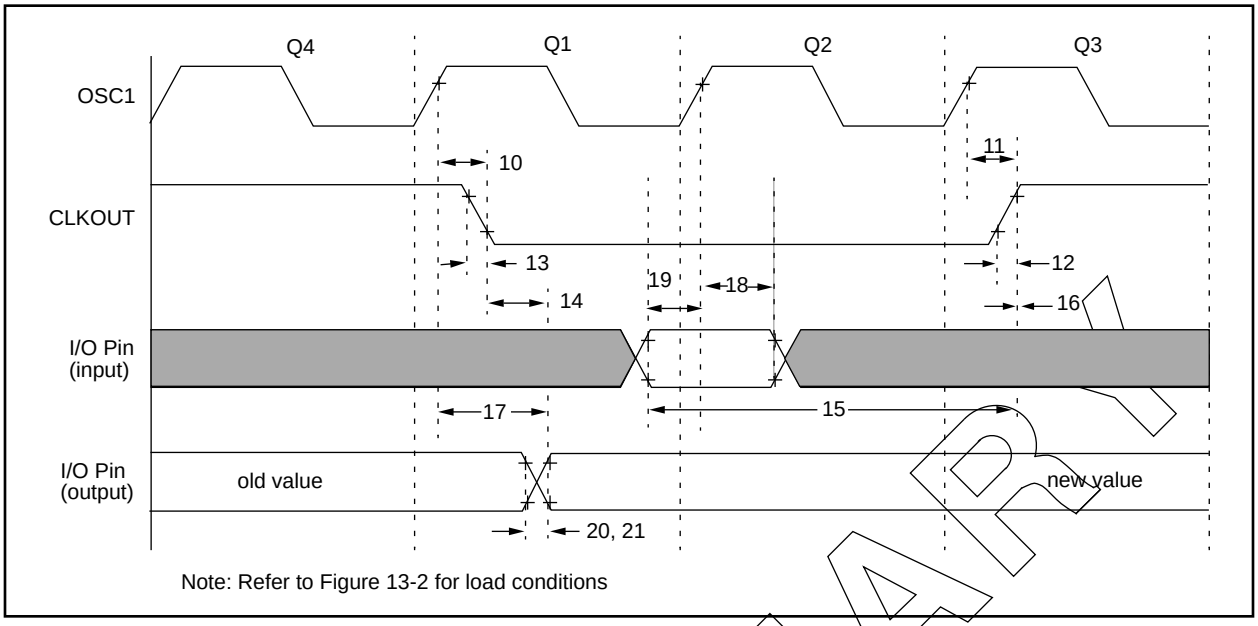


FIGURE 13-3: CLKOUT AND I/O TIMING



Note: Refer to Figure 13-2 for load conditions

TABLE 13-2: 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)	100	—	—	ns	
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns	
20	TioR	Port output rise time	—	10	25	ns	
21	TioF	Port output fall time	—	10	25	ns	
22††	Tinp	PBTN pin high or low time	20	—	—	ns	IN mode only
23††	Trbp	RC<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 IN Mode where CLKOUT output is 4 x TOSC

FIGURE 13-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER (HS MODE) AND POWER-UP TIMER TIMING

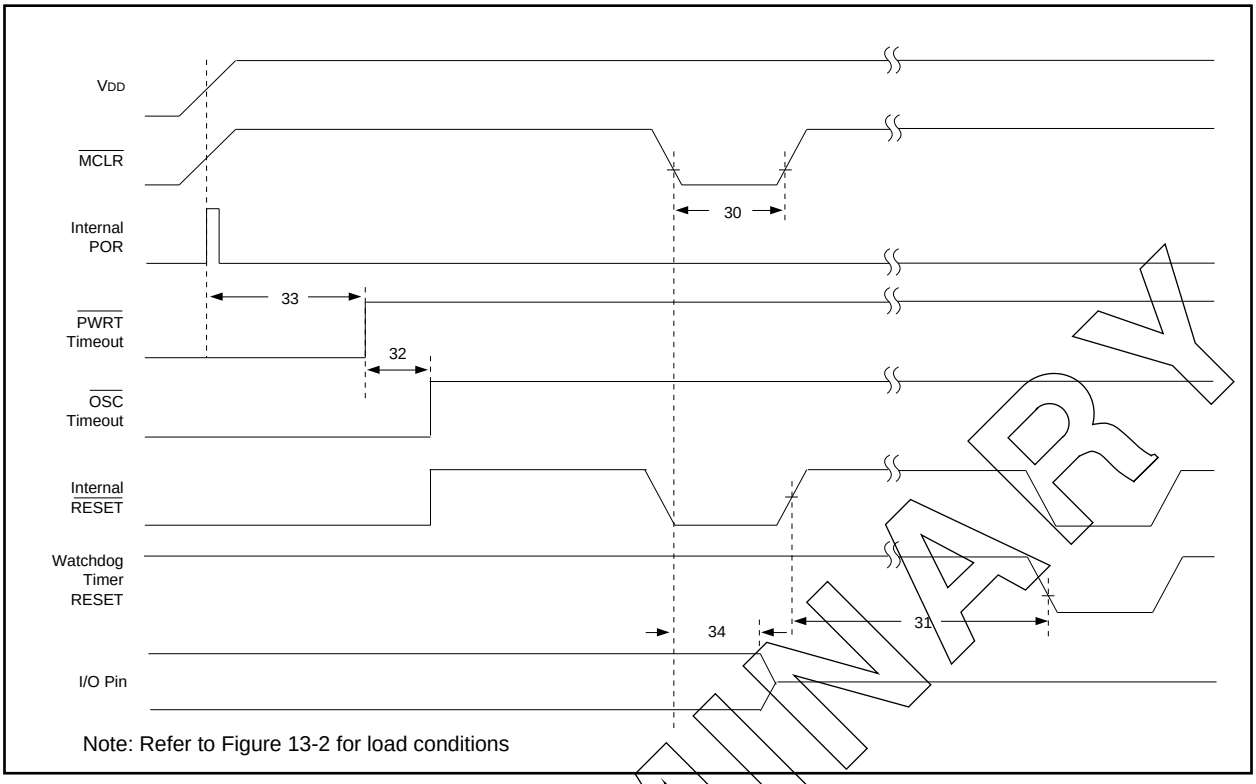


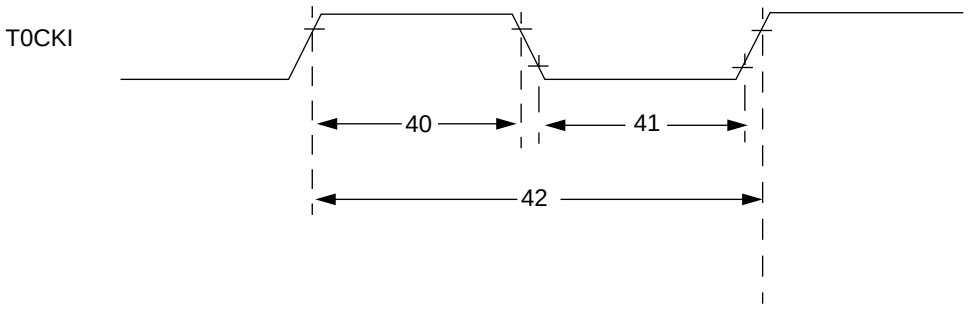
TABLE 13-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	MCLR Pulse Width (low)	100	—	—	ns	VDD = 5V, -40°C to +85°C
31	TWDT	Watchdog Timer Timeout Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +85°C
	ss(WDT)	Supply Sensitivity	—	-12.6	—	%/V	TA = 25°C
	tc(WDT)	Temperature Coefficient	—	0.5	—	%/°C	VDD = 5V
32	TOST	Oscillation Start-up Timer Period		1024 TOSC 8 TOSC		ms ms	TOSC = OSC1 period IN osc mode
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	

* 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 13-5: TIMER0 CLOCK TIMINGS



Note: Refer to Figure 13-2 for load conditions.

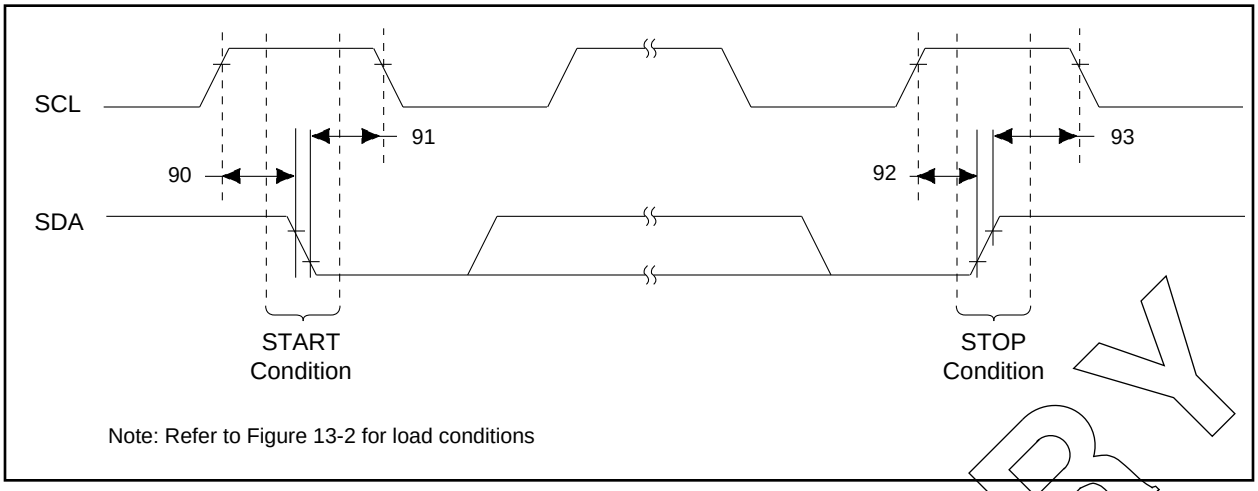
TABLE 13-4: TIMER0 CLOCK REQUIREMENTS

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.

FIGURE 13-6: I²C BUS START/STOP BITS TIMING

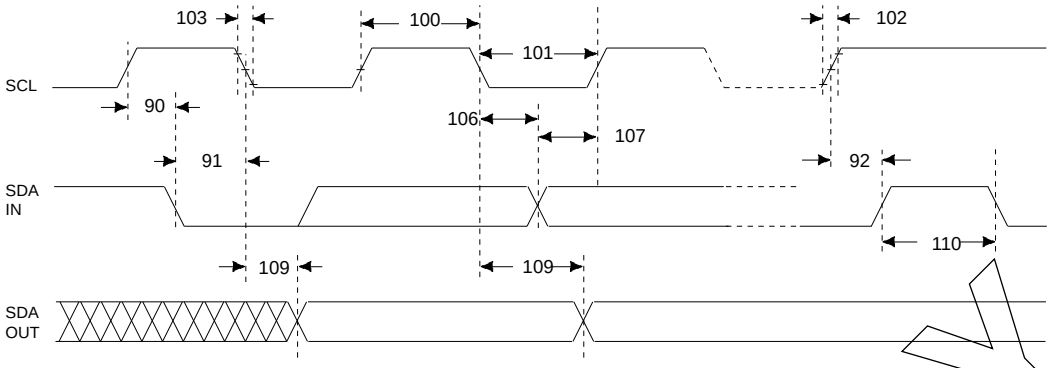


Note: Refer to Figure 13-2 for load conditions

TABLE 13-5: I²C BUS START/STOP BITS REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ	Max	Units	Conditions
90	TSU:STA	START condition	100 KHZ mode	4700	—	—	ns	Only relevant for repeated START condition
		Setup time	400 KHz mode	600	—	—		
91	THD:STA	START condition	100 KHz mode	4000	—	—	ns	After this period the first clock pulse is generated
		Hold time	400 KHz mode	600	—	—		
92	TSU:STO	STOP condition	100 KHZ mode	4700	—	—	ns	
		Setup time	400 KHz mode	600	—	—		
93	THD:STO	STOP condition	100 KHz mode	4000	—	—	ns	
		Hold time	400 KHz mode	600	—	—		

FIGURE 13-7: I^2C BUS DATA TIMING



Note: Refer to Figure 13-2 for load conditions

TABLE 13-8: I^2C BUS DATA REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock high time	100 kHz mode	4.0	—	µs	PIC14000 must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	PIC14000 must operate at a minimum of 10 MHz
			I^2C Module	1.5 TCY	—		
101	TLOW	Clock low time	100 kHz mode	4.7	—	µs	PIC14000 must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	PIC14000 must operate at a minimum of 10 MHz
			I^2C Module	1.5 TCY	—		
102	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20+0.1 C_b	300	ns	C_b 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 C_b	300	ns	C_b 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	
	C_b	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 STARTs or STOPs.

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=1250 ns (according to the standard-mode I^2C bus specification) before the SCL line is released.

13.5 DC and AC Characteristics Graphs and Tables for PIC14000

FIGURE 13-9: TYPICAL I_{PD4} vs V_{DD} AT 25°C

TO BE DETERMINED.

FIGURE 13-10: TYPICAL I_{PD3} vs V_{DD} AT 25°C

TO BE DETERMINED.

FIGURE 13-11: V_{TH} (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN HS MODE) vs V_{DD}

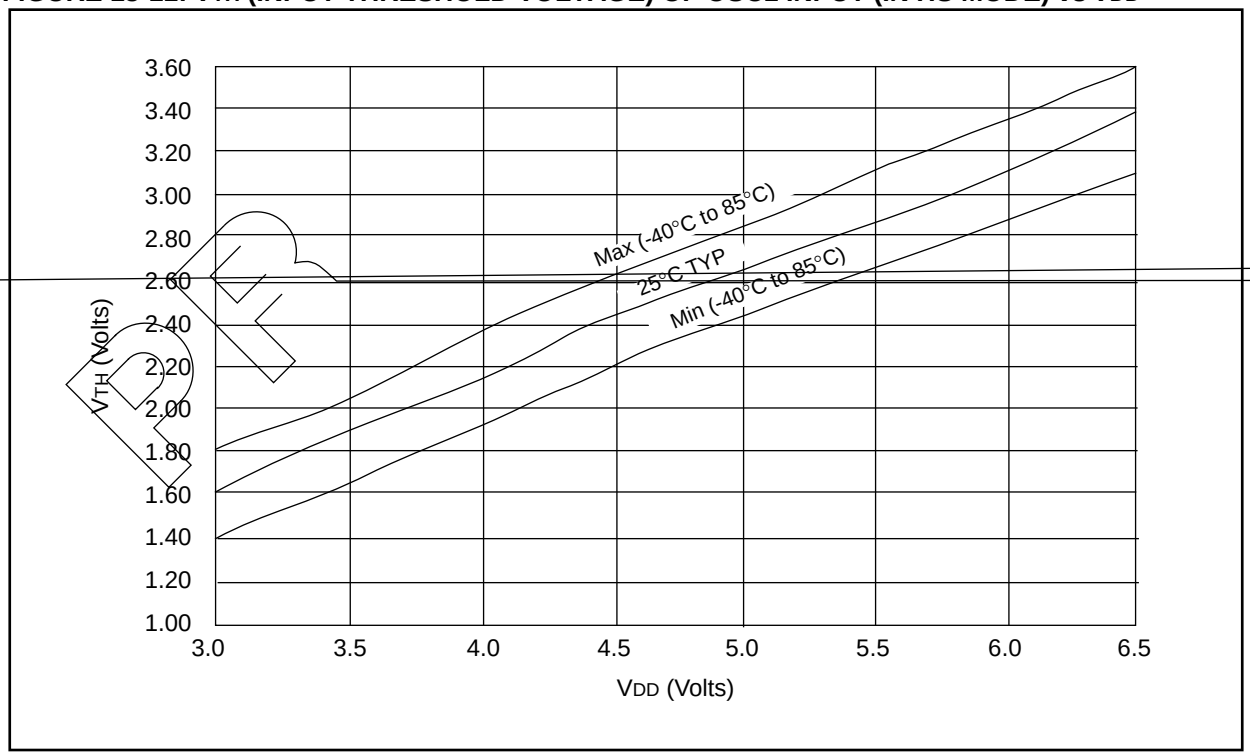


FIGURE 13-12: TYPICAL OPERATING SUPPLY CURRENT vs FREQ (EXT CLOCK, 25°C)

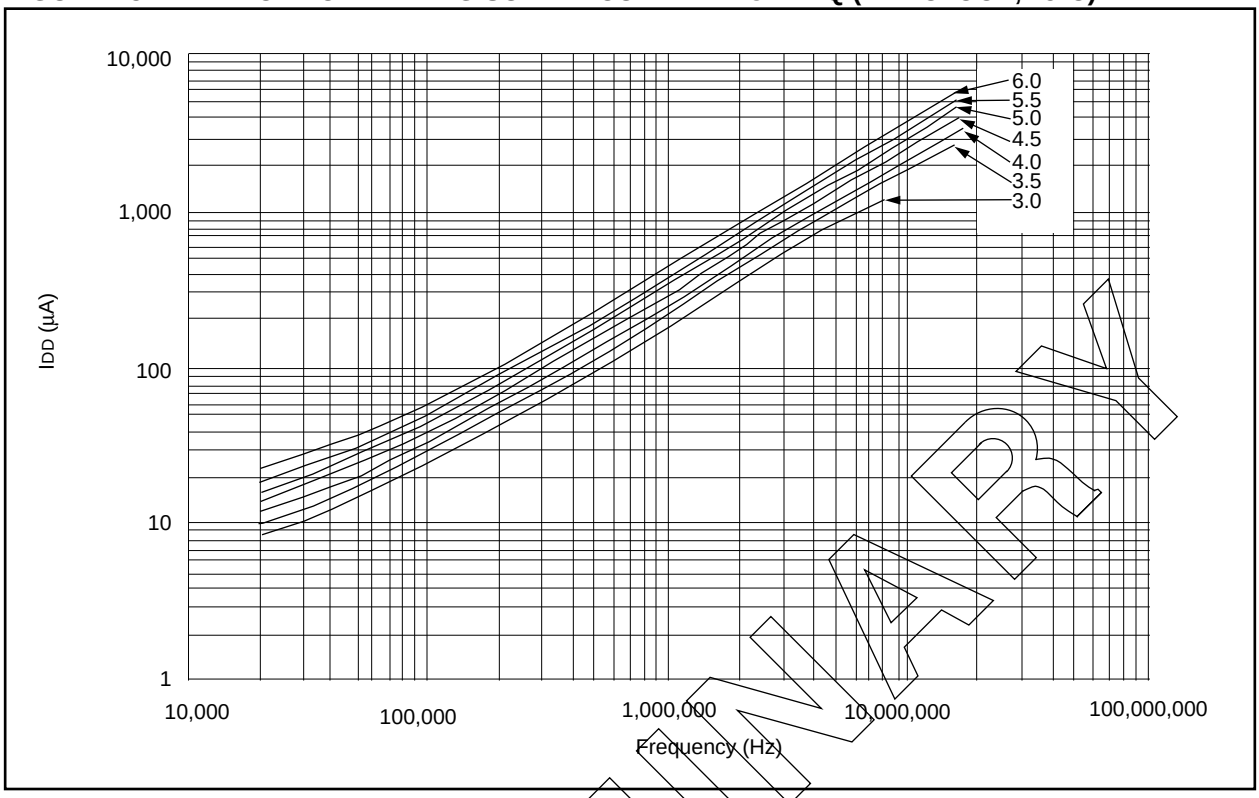


FIGURE 13-13: MAXIMUM OPERATING SUPPLY CURRENT vs FREQ (EXT CLOCK, -40° TO +85°C)

TO BE DETERMINED.

FIGURE 13-14: MAXIMUM I_{PD1} VS FREQ (EXT CLOCK, -40° TO +85°C)

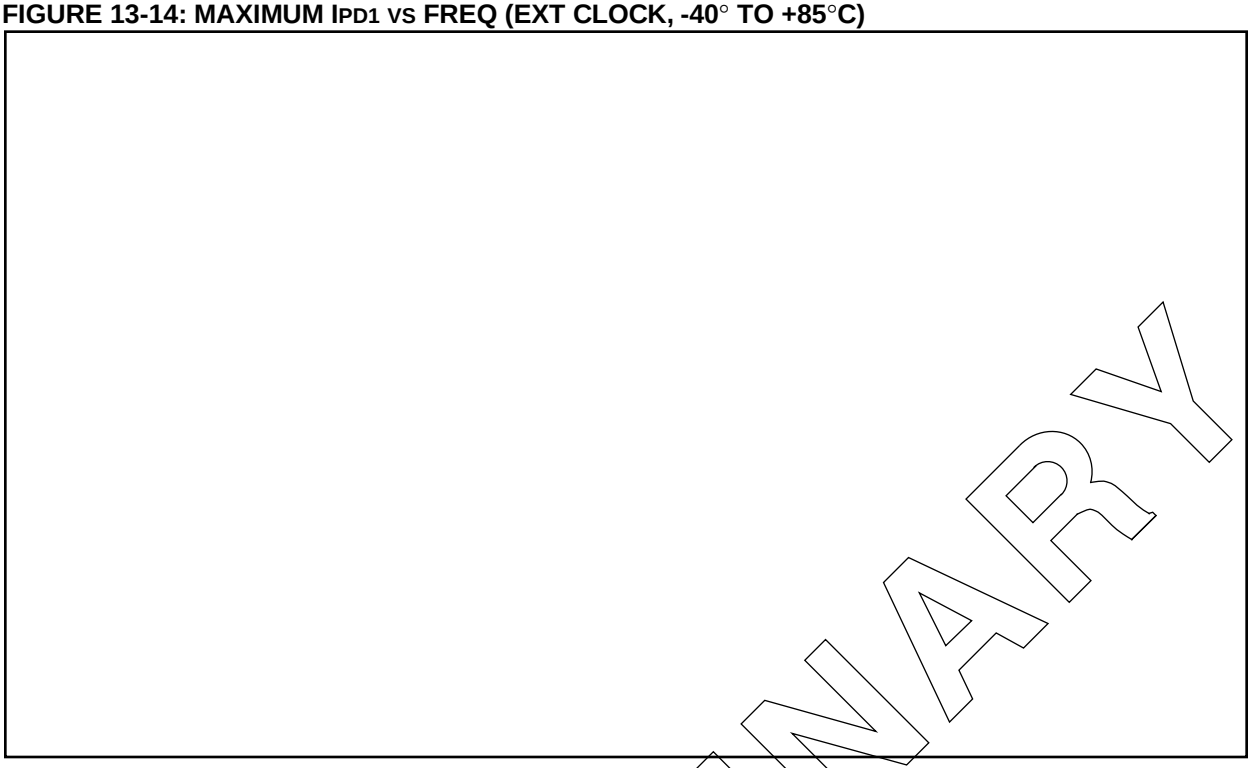


FIGURE 13-15: WATCHDOG TIMER TIME-OUT PERIOD (T_{WDT}) VS. TEMPERATURE (TYPICAL)

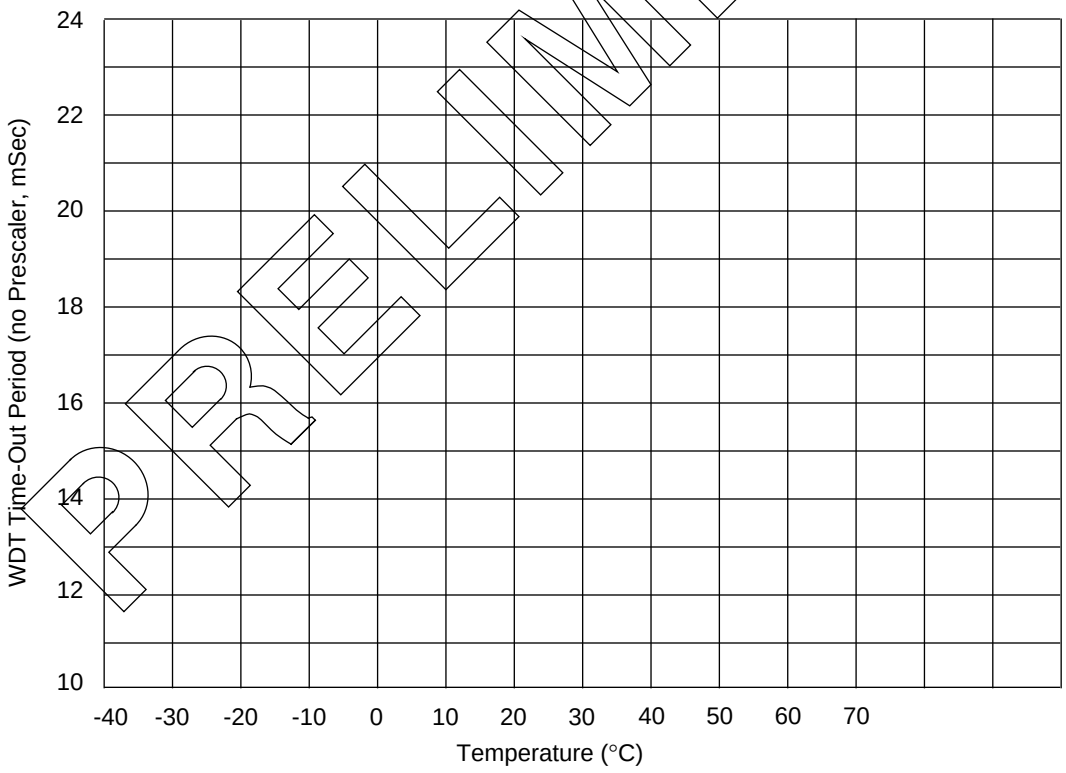


FIGURE 13-16: WDT TIMER TIME-OUT PERIOD VS VDD

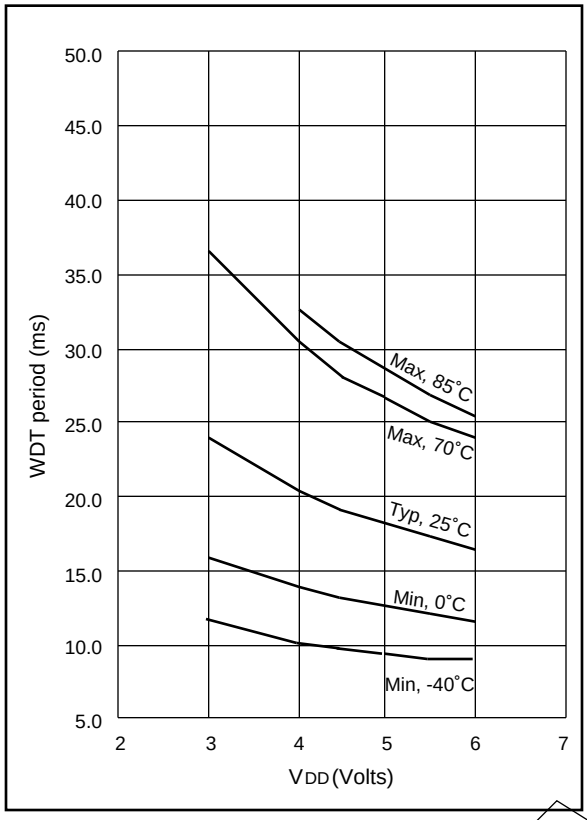


FIGURE 13-17: TRANSCONDUCTANCE (GM) OF HS OSCILLATOR VS VDD

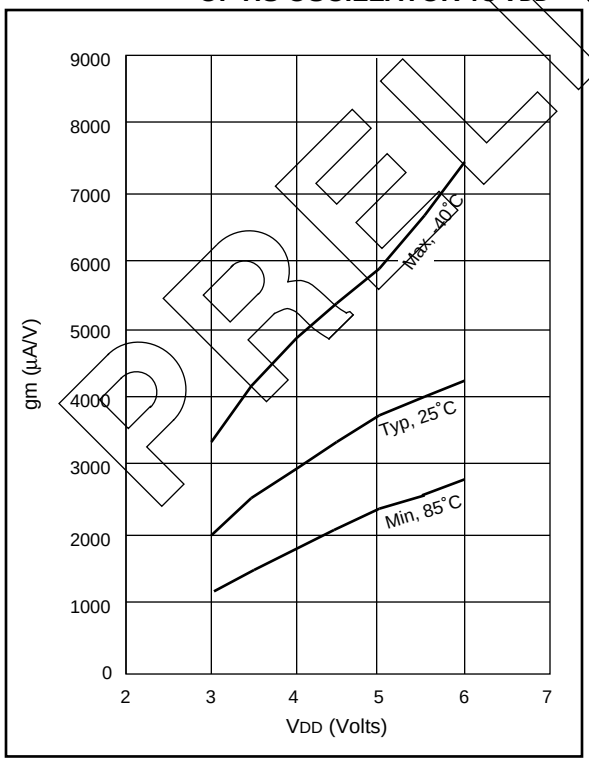


FIGURE 13-18: IOH VS VOH, VDD = 3V*

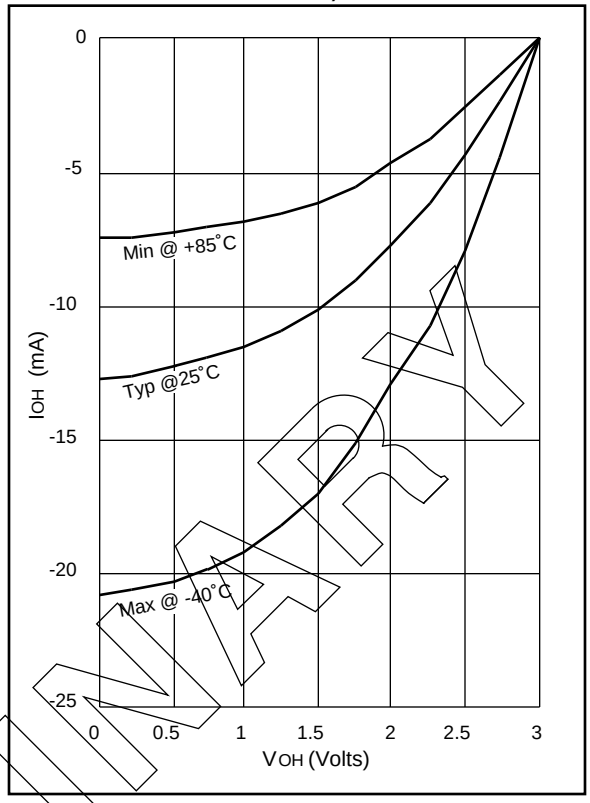
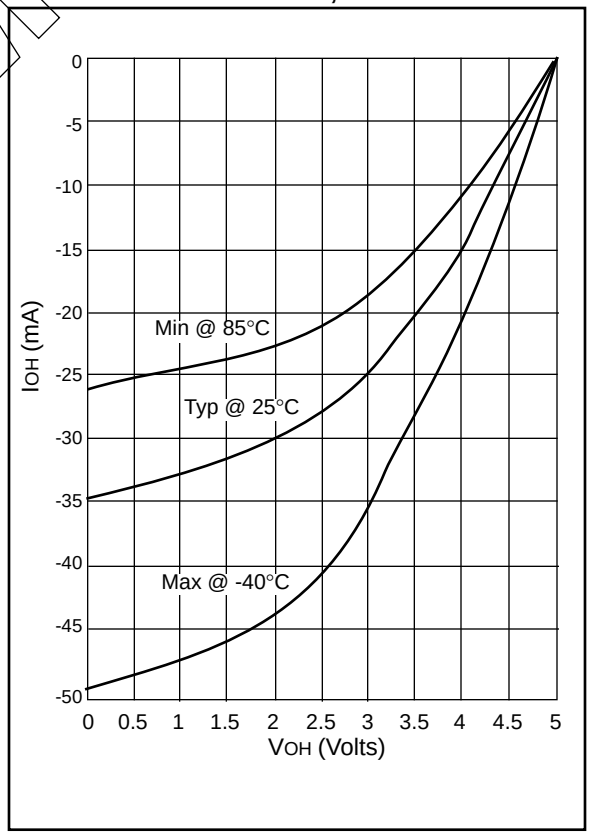


FIGURE 13-19: IOH VS VOH, VDD = 5V*



***NOTE:** All pins except RC6, RC7, RD0, RD1,OSC2

FIGURE 13-20: IOL VS VOL, VDD = 3V*

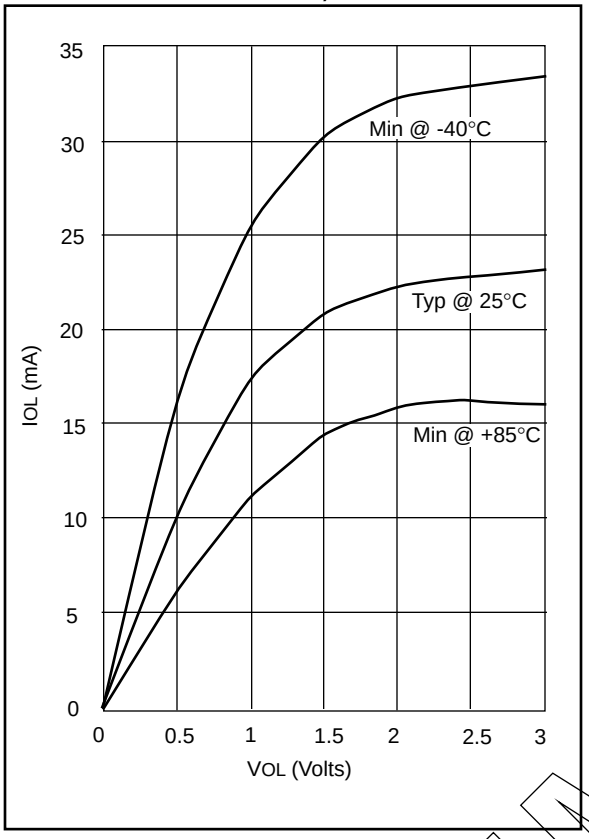
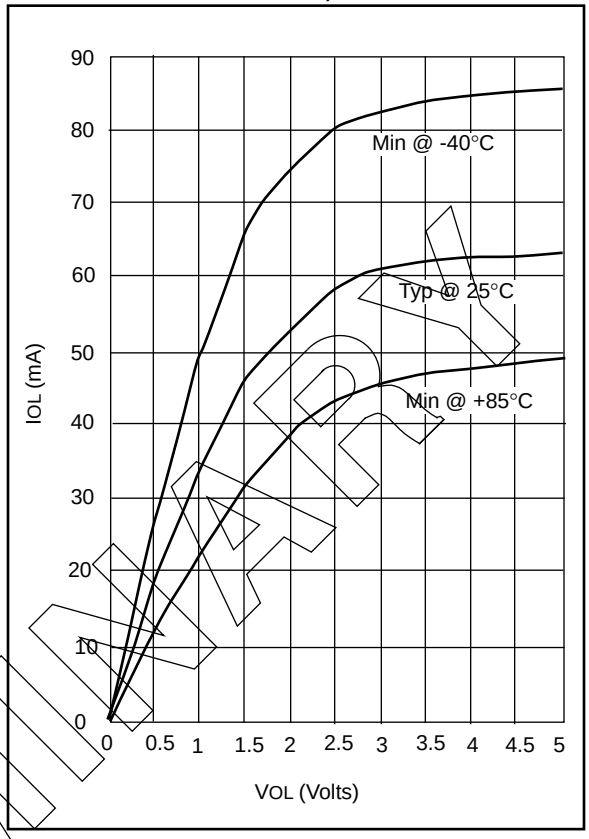


FIGURE 13-21: IOL VS VOL, VDD = 5V*



***NOTE:** All pins except OSC2

14.0 ANALOG SPECIFICATIONS: PIC14000-04 (COMMERCIAL, INDUSTRIAL)

Standard Operating Conditions (unless otherwise stated)

Operating Temperature: -40°C ≤ TA ≤ +85°C for industrial
0°C ≤ TA ≤ +70°C for commercial

VDD range: 2.7V (min) to 6.0V (max) unless otherwise stated.

Characteristic	Sym.	Min.	Typ.	Max.	Units	Conditions	Notes
Bandgap Voltage Reference							
Output Voltage	vo(vref)	1.14	1.19	1.24	V		
Turn-on Settling Time to < 0.1%	ton(vref)	—	1	10	ms	REFOFF bit in SLPCON register → 0	1
Temperature Coefficient	tc(vref)	—	±50	—	ppm/°C	Measured from 25°C to -40°C, +85°C	1
Temperature Coefficient	tc(vref)	—	±20	—	ppm/°C	Measured from 25°C to 0°C, +70°C	1
Supply Sensitivity	ss(vref)	—	0.04	—	%/V	From VDDmin to VDDmax	1
Operating Current (on)	idd(vref)	—	20	30	µA	REFOFF = 0	2
Operating Current (off)	idd(vref)	—	0	—	µA	REFOFF = 1	2
Programmable Current Source							
Output Current	io(cdac)					CDAC pin = 0V	3
		18.75	33.75	48.75	µA	ADCON1<7:4> = 1111b (full-scale)	
		1.25	2.25	3.25	µA	ADCON1<7:4> = 0001b (1 LSB)	
		-0.5	0	0.5	µA	ADCON1<7:4> = 0000b (zero-scale)	
Resolution	res(cdac)	1.25	2.25	3.25	µA	1 LSB	
Relative accuracy (linearity error)	racc(cdac)	-1/2		+1/2	lsb	CDAC = 0V	
Turn-on Settling Time to < 0.1% (reference start-up)	ton(cdac)	—	1	10	ms	Bias generator (reference) turn-on time (REFOFF 1 → 0)	1
Turn-on Settling Time to < 0.1% (reference already on and stable)	ton(cdac)	—	1	10	µs	REFOFF = 0 (constant), ADCON1<7:4> 0000b → 1111b	1
Temperature Coefficient	tc(cdac)	—	±0.1	—	%/°C	Measured from 25°C to Tmin, Tmax	1
Supply Sensitivity	ss(cdac)	—	0.2	—	%/V	From VDDmin to VDDmax	1
Output Voltage Sensitivity	vs(cdac)	-0.1	-0.01	–	%/V	CDAC pin voltage = 0V to VDD - 1.4V	
Output Voltage Range	vo(cdac)	0	—	VDD-1.4	V		
Operating Current (A/D on)	idd(cdac)	—	50	70	µA	ADCON1<7:4> = 1111b	2
Operating Current (A/D off)	idd(cdac)	—	0	—	µA	REFOFF = 1, ADOFF = 1	2
Temperature Sensor							
Output Voltage	vo(temp)	0.92	1.05	1.18	V	TA = 25°C	
Supply Sensitivity	ss(temp)	—	0.2	—	%/V	From VDDmin to VDDmax	1
Temperature Coefficient	KTC	3.2	3.65	4.1	mV/°C	Measured from 25°C to Tmax. Includes ± 2°C temperature calibration tolerance	

Standard Operating Conditions (unless otherwise stated)

Operating Temperature: -40°C ≤ TA ≤ +85°C for industrial
0°C ≤ TA ≤ +70°C for commercial

VDD range: 2.7V (min) to 6.0V (max) unless otherwise stated.

Characteristic	Sym.	Min.	Typ.	Max.	Units	Conditions	Notes
Temperature Sensor (continued)							
Output Linearity	lin(temp)	—	TBD	—			1
Operating Current (sensor on)	idd(temp)	—	150	250	µA	TEMPOFF = 0	2
Operating Current (sensor off)	idd(temp)	—	0	—	µA	TEMPOFF = 1	2
Slope Reference Voltage Divider							
Output Voltage (SREFHI)	voh(sref)	1.14	1.19	1.24	V		
Output Voltage (SREFLO)	vol(sref)	0.10	0.13	0.16	V		
Slope Reference Calibration Factor	KREF	0.09	0.126	0.16		TA = 25°C, VDD = 5V	
KREF Supply Sensitivity	ss(KREF)	—	0.02	—	%/V	From VDDmin to VDDmax	1
KREF Temperature Coefficient	tc(KREF)	—	20	—	ppm/°C	From Tmin to Tmax	1
Operating Current (A/D on)	idd(sref)	—	55	85	µA	ADOFF = 0	2
Operating Current (A/D off)	idd(sref)	—	0	—	µA	ADOFF = 1	2
A/D Comparator							
Input Offset Voltage	ioff(adc)	-10	2	10	mV	Measured over common-mode range	
Input Common Mode Voltage Range	cmr(adc)	0	—	VDD-1.4	V		
Differential Voltage Gain	gain(adc)	—	100	—	dB		1
Common Mode Rejection Ratio	cmrr(adc)	—	80	—	dB	VDD = 5V, TA = 25°C, over common-mode range	1
Power Supply Rejection Ratio	psrr(adc)	—	70	—	dB	TA = 25°C, VDDmin to VDDmax	1
Operating Current (A/D on)	idd(adc)	—	40	65	µA	ADOFF = 0	2
Operating Current (A/D off)		—	0	—	µA	ADOFF = 1	2
Programmable Reference(s)							
Upper Range						TA = 25°C	
Output Voltage	vo(pref)	0.627	0.792	0.957	V	PREFx<7:0> = 7Fh (127 decimal), max	
		0.418	0.528	0.638	V	PREFx<7:0> = 50h (80 decimal), min	
Coarse Resolution	resc(pref)	38.0	48.0	58.0	mV	PREFx<2:0> = constant	
Fine Resolution	resf(pref)	4.0	5.0	6.0	mV	PREFx<7:3> = constant	
Middle Range						TA = 25°C	
Output Voltage	vo(pref)	0.414	0.523	0.632	V	PREFx<7:0> = 4F (79 decimal), max	
		0.380	0.480	0.580	V	PREFx<7:0> = 00h (default), mid-point	
		0.342	0.432	0.522	V	PREFx<7:0> = C8h (200 decimal), min	
Coarse Resolution	resc(pref)	3.8	4.8	5.8	mV	PREFx<2:0> = constant	
Fine Resolution	resf(pref)	0.38	0.46	0.54	mV	PREFx<7:3> = constant	1

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature: -40°C ≤ TA ≤ +85°C for industrial 0°C ≤ TA ≤ +70°C for commercial VDD range: 2.7V (min) to 6.0V (max) unless otherwise stated.							
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**	**Notes**
Programmable Reference(s) (continued)							
Lower Range Output Voltage	vo(pref)	0.304 0.114	0.384 0.144	0.464 0.174	V V	TA = 25°C PREFx<7:0> = D7h (215 decimal), max PREFx<7:0> = F8h (248 decimal), min	
Coarse Resolution	resc(pref)	38.0	48.0	58.0	mV	PREFx<2:0> = constant	
Fine Resolution	resf(pref)	4.0	5.0	6.0	mV	PREFx<7:3> = constant	
Relative accuracy (linearity error)	racc(pref)	–1/2	—	+1/2	lsb	lsb = resolution within selected range	
Settling Time to < ±1/2 LSB	ts(pref)	—	1	10	µS	PREFx<7:0> transition from 7Fh to FFh	1
Temperature Coefficient	tc(pref)	—	0.39	—	%/°C	From Tmin to Tmax	1
Supply Sensitivity	ss(pref)	—	0.2	—	%/V	From VDDmin to VDDmax	1
Operating Current (on)	idd(pref)	—	5	10	µA	CMOFF = 0	2
Operating Current (off)	idd(pref)	—	0	—	µA	CMOFF = 1	2
Low-Voltage Detector							
Detect Voltage	v-(lvd)	2.43	2.55	2.67	V	Decreasing VDD	
Release Voltage	v+(lvd)	2.48	2.60	2.72	V	Increasing VDD	
Hysteresis	vhys(lvd)	35	55	75	mV	Between detect and release trip points	
Operating Current (on)	idd(lvd)	—	15	25	µA	REFOFF = 0	2
Operating Current (off)	idd(lvd)	—	0	—	µA	REFOFF = 1	2
Internal Oscillator							
Frequency Range	fosc(in)	3.0	4.0	5.0	MHz		
Temperature Coefficient	tc(in)	—	-0.04	—	%/°C	From Tmin to Tmax	1
Supply Sensitivity	ss(in)	—	0.8	—	%/V	From VDDmin to VDDmax	1
Jitter	jit(in)	—	100	—	ppm	±3 sigma from mean	1
Start-up Time	tsu(in)	—	8	—	Tcycs	At Power-On Reset and exit from SLEEP	4
Operating Current (oscillator on)	idd(in)	—	300	500	µA		2
Operating Current (oscillator off)	idd(in)	—	0	—	µA	SLEEP mode, OSCOFF = 1	2
Voltage Regulator Control Output							
Regulation Voltage	vo(reg)	5.2	5.9	6.6	V	Measured with Ivreg = 10µA at TA = 25°C	
Temperature Coefficient	tc(vreg)	—	-0.2	—	%/°C	From Tmin to Tmax	1
Operating Current (Recommended)	idd(vreg)	1	–	10	µA	Determined by external components	
Operating Current		—	0	—		If VREG pin is open	

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature: -40°C ≤ TA ≤ +85°C for industrial 0°C ≤ TA ≤ +70°C for commercial VDD range: 2.7V (min) to 6.0V (max) unless otherwise stated.							
Characteristic	**Sym.**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**	**Notes**
Programmable Reference Comparator(s)							
Input Offset Voltage	ioff(comp)	-10	3	10	mV	Tested at 0.5V common-mode voltage	
Input Common Mode Voltage Range	cmr(comp)	0	—	VDD-1.4	V		1
Differential Voltage Gain	gain(comp)	—	80	—	dB		1
Common Mode Rejection Ratio	cmrr(comp)	—	60	—	dB	VDD = 5V, TA = 25°C, over common-mode range	1
Power Supply Rejection Ratio	psrr(comp)	—	55	—	dB	TA = 25°C, VDDmin to VDDmax	1
Operating Current (on)	idd(comp)	—	10	20	µA	CMOFF = 0	2
Operating Current (off)	idd(comp)	—	0	—	µA	CMOFF = 1	2
Level-Shift Network(s)							
Input Current (RA1/RD5 pin)	iin(lvs)	-3.4	-4.8	-6.2	µA	TA = 25°C, RA1/RD5 = 0V (SUM pin is open)	
Output Voltage	vo(lvs)	0.37	0.46	0.55	V	TA = 25°C, RA1/RD5 = 0V, (SUM pin is open)	
Zeroing Mismatch Error	zm(lvs)	—	0.02	—	%		1
Output Voltage Temperature Coefficient	tc(lvs)	—	0.39	—	%/°C	From Tmin to Tmax	1
Output Voltage Supply Sensitivity	ss(lvs)	—	0.2	—	%/V	From VDDmin to VDDmax	1
Operating Current (network on)	idd(lvs)	—	5	15	µA	LSOFF = 0	2
Operating Current (network off)	idd(lvs)	—	0	—	µA	LSOFF = 1	2

Standard Operating Conditions (unless otherwise stated)

Operating Temperature: -40°C ≤ TA ≤ +85°C for industrial
0°C ≤ TA ≤ +70°C for commercial

VDD range: 2.7V (min) to 6.0V (max) unless otherwise stated.

Characteristic	Sym.	Min.	Typ.	Max.	Units	Conditions	Notes
Calibration Accuracy						All parameters calibrated at VDD = 5V, TA = 25°C unless noted.	3, 5
				Accuracy			
Parameter	**Sym.**	**Resolution**	**Units**	**Typ**	**Max**	**Conditions**	**Notes**
Slope Reference Ratio	KREF	0.015%	—	.02%	—		
Bandgap Reference Voltage	KBG	10	µV	.01%	—		
Temperature Sensor Output Voltage	VTHERM	20	µV	.02%	—		
Temperature Sensor Slope Coefficient	KTC	0.33	µV/°C	6.7%	—	Calibrated at 25°C and Tmax	
Internal Oscillator Frequency	FOSC	10.0	kHz	0.14%	—		
Watchdog Timer Time-out Period	TWDT	1	ms	0.5 ms	—		

Notes for the analog specifications:

Note 1: This parameter is characterized but not tested.

Note 2: IDD values of individual analog module cannot be tested independently but are characterized.

Note 3: Calibration temp accuracy is ± 1°C typical, ± 2°C max.

Note 4: Guaranteed by design.

Note 5: Refer to AN621 for further information on calibration parameters and accuracy.

Calculations:

Temperature coefficients are calculated as:

tc = (value @TMAX - value @TMIN) / ((TMAX-TMIN) * Average(value @TMAX,value @TMIN))

Temperature coefficient for the internal temperature sensor is calculated as:

tc sensor = (sensor voltage @ TMAX - sensor voltage @ 25°C) / (TMAX - 25°C)

Temperature coefficients for the bandgap reference and programmable current source are calculated as

the larger TC from 25°C to either TMIN or TMAX

Supply sensitivities are calculated as:

ss = (value@VDDMAX - value@VDDMIN)/((VDDMAX - VDDMIN)*
Average(value@VDDMAX, value@VDDMIN))

Programmable current source output sensitivity is calculated as:

vs = (value@(VDD - 1.4V) - value @ 0V)/(VDD - 1.4V) *
Average(value@(VDD - 1.4V), value @ 0V)

FIGURE 14-1: BANDGAP REFERENCE OUTPUT VOLTAGE vs. TEMPERATURE (TYPICAL DEVICES SHOWN)

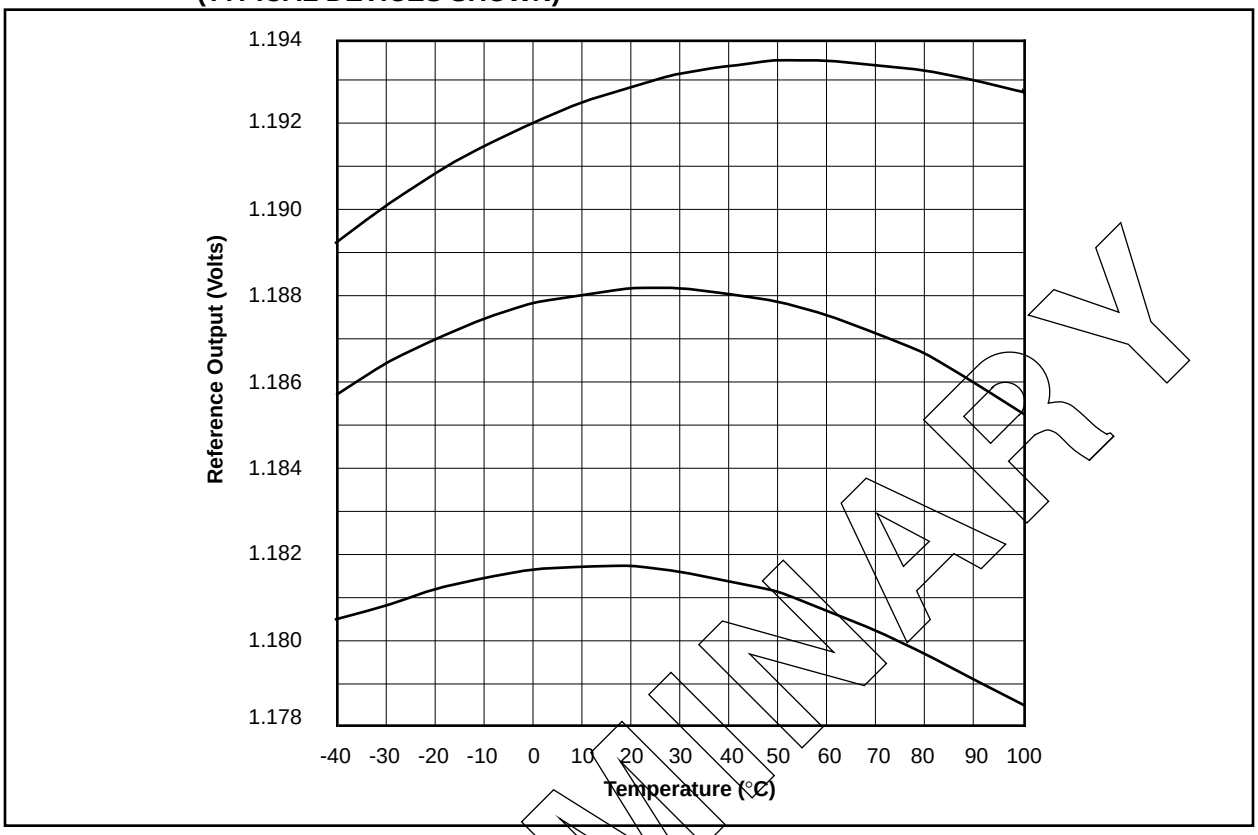


FIGURE 14-2: PROGRAMMABLE CURRENT SOURCE vs. TEMPERATURE (TYPICAL DEVICES SHOWN)

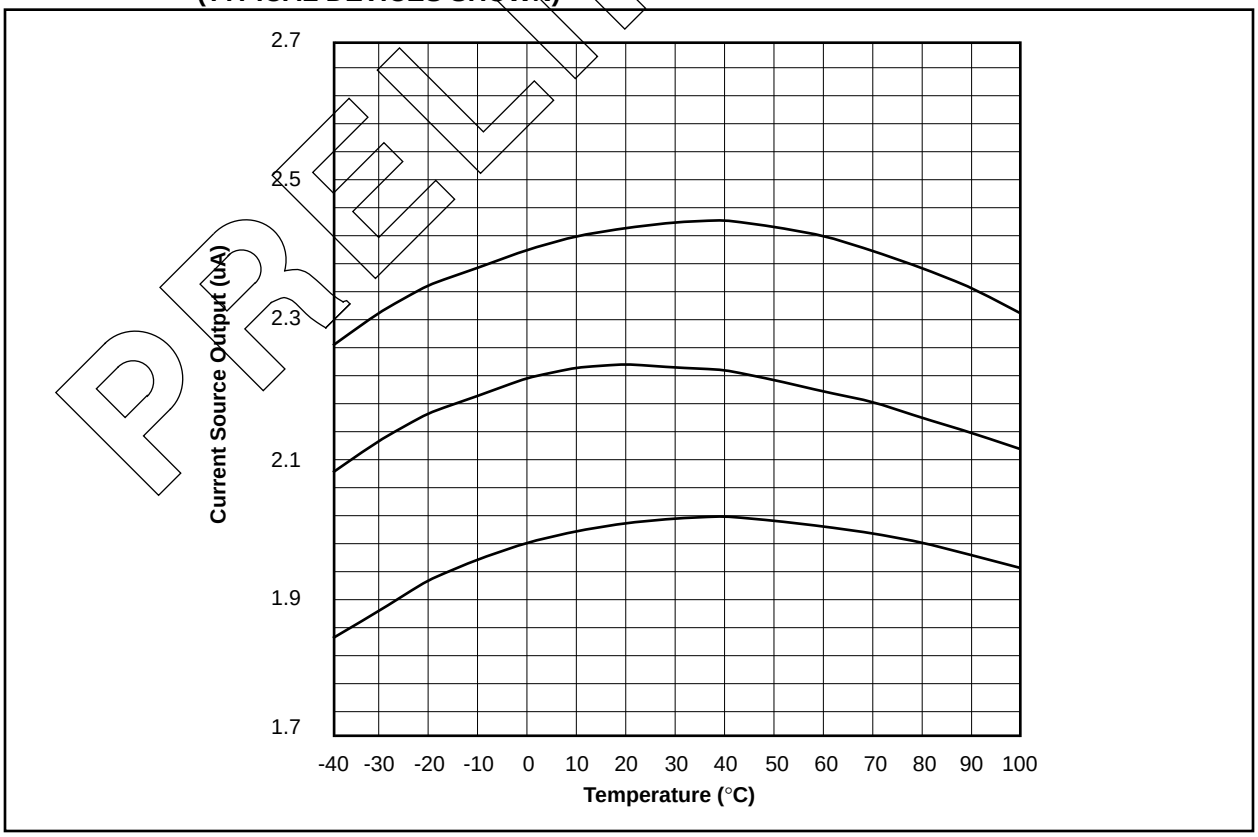


FIGURE 14-3: TEMPERATURE SENSOR OUTPUT VOLTAGE vs. TEMPERATURE (TYPICAL DEVICES SHOWN)

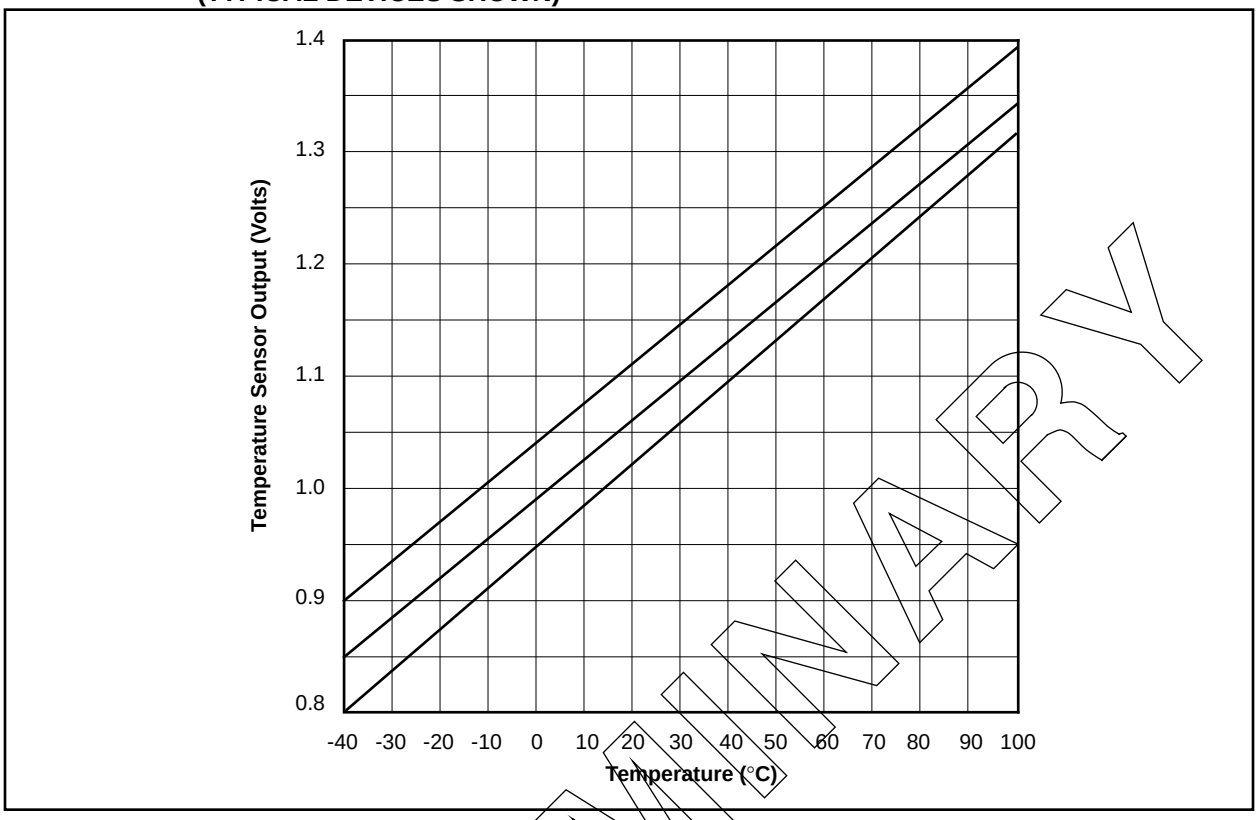


FIGURE 14-4: SLOPE REFERENCE RATIO (K_{REF}) vs. SUPPLY VOLTAGE (TYPICAL DEVICES SHOWN)

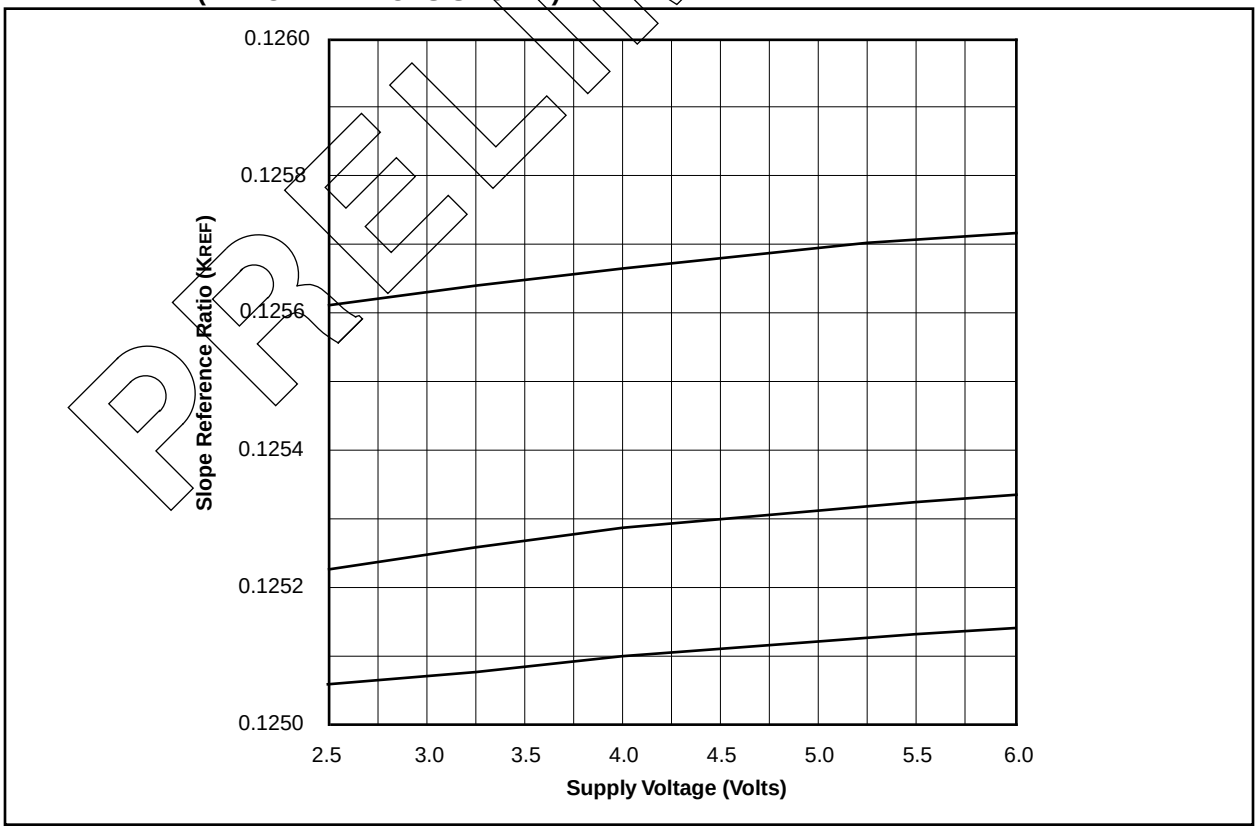


FIGURE 14-5: SLOPE REFERENCE RATIO (KREF) vs. TEMPERATURE (TYPICAL DEVICES SHOWN)

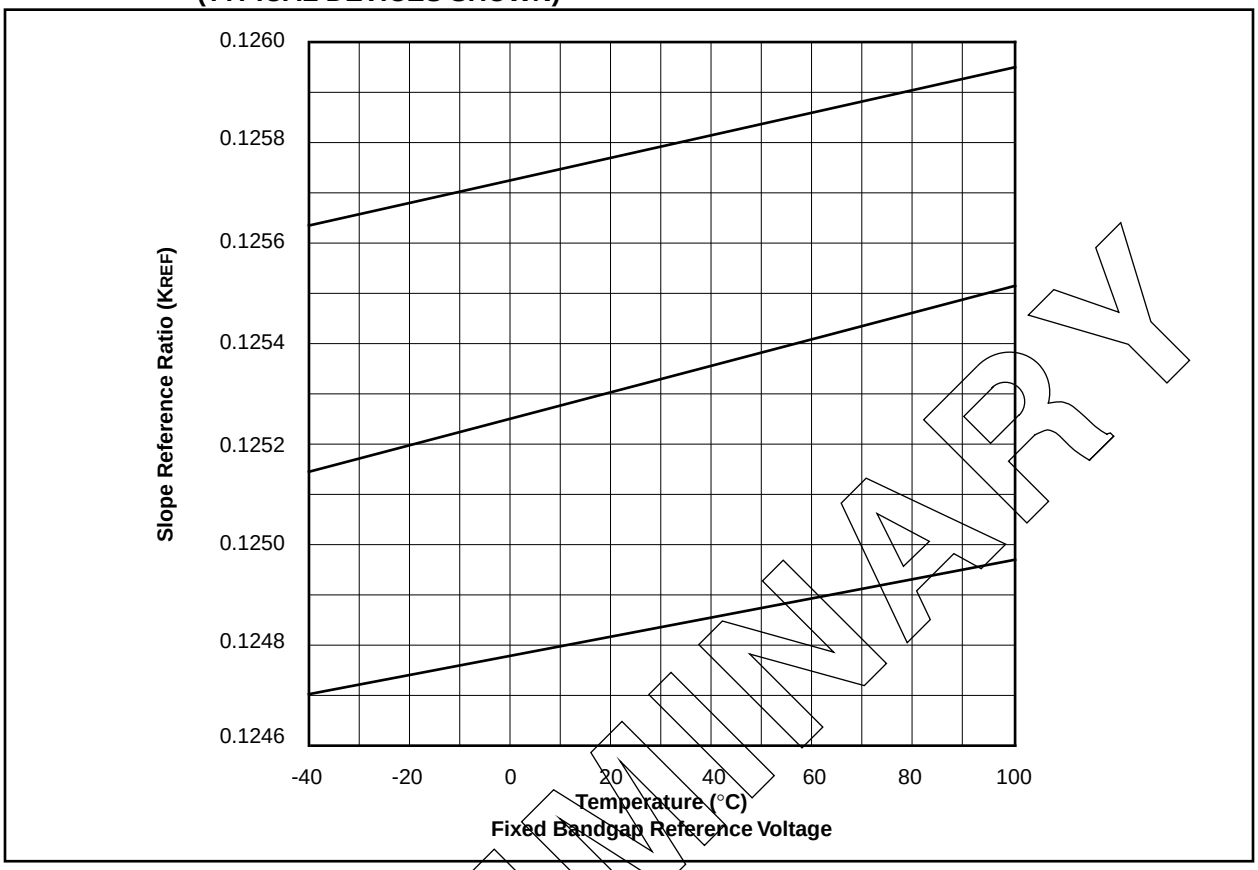


FIGURE 14-6: PROGRAMMABLE REFERENCE OUTPUT vs. TEMPERATURE (TYPICAL)

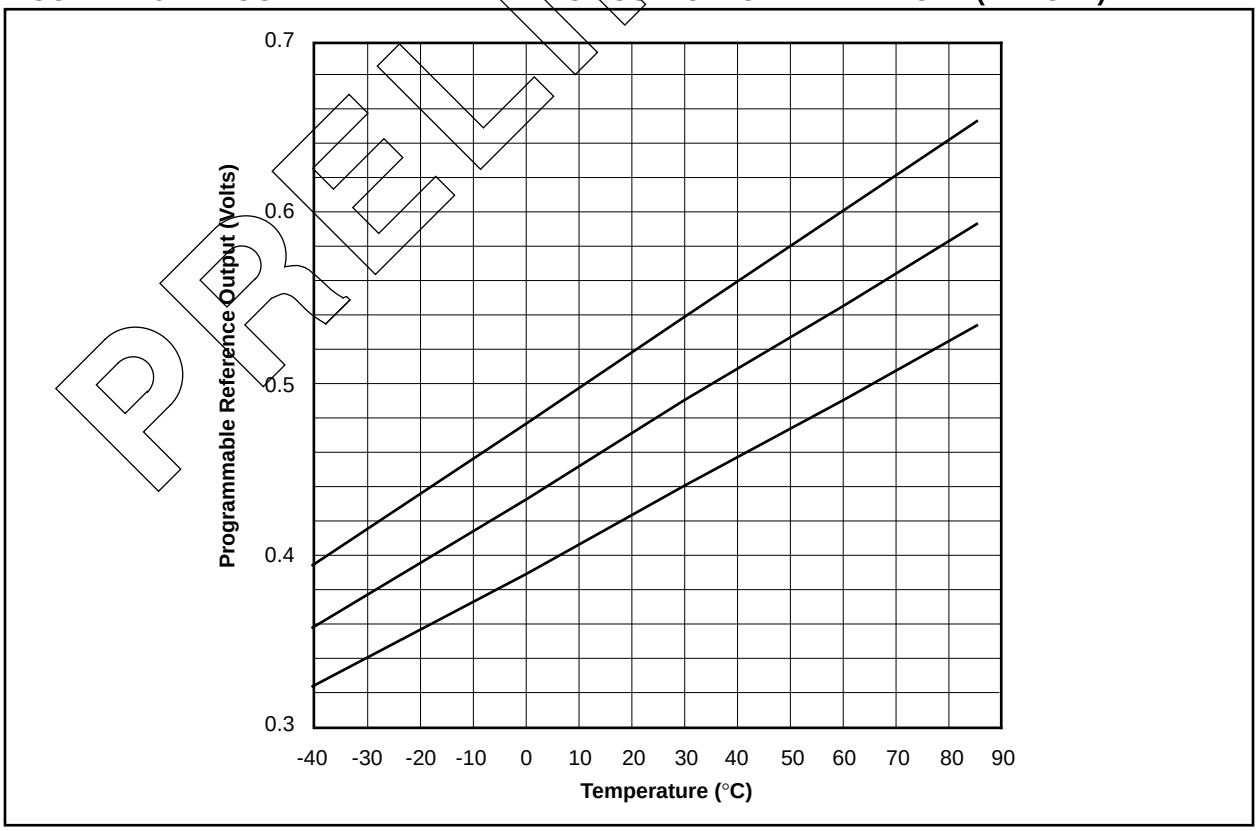


FIGURE 14-7: INTERNAL RC OSCILLATOR FREQUENCY vs. SUPPLY VOLTAGE (TYPICAL DEVICES SHOWN)

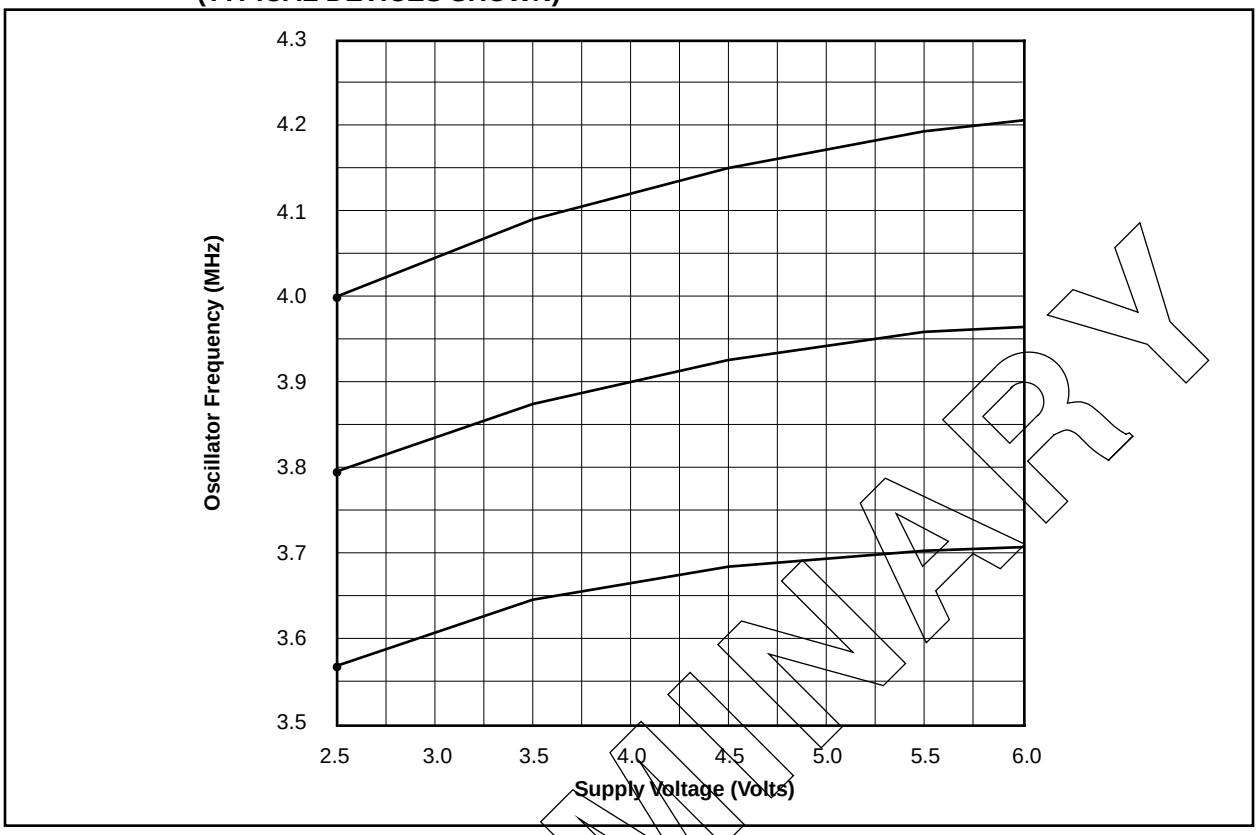
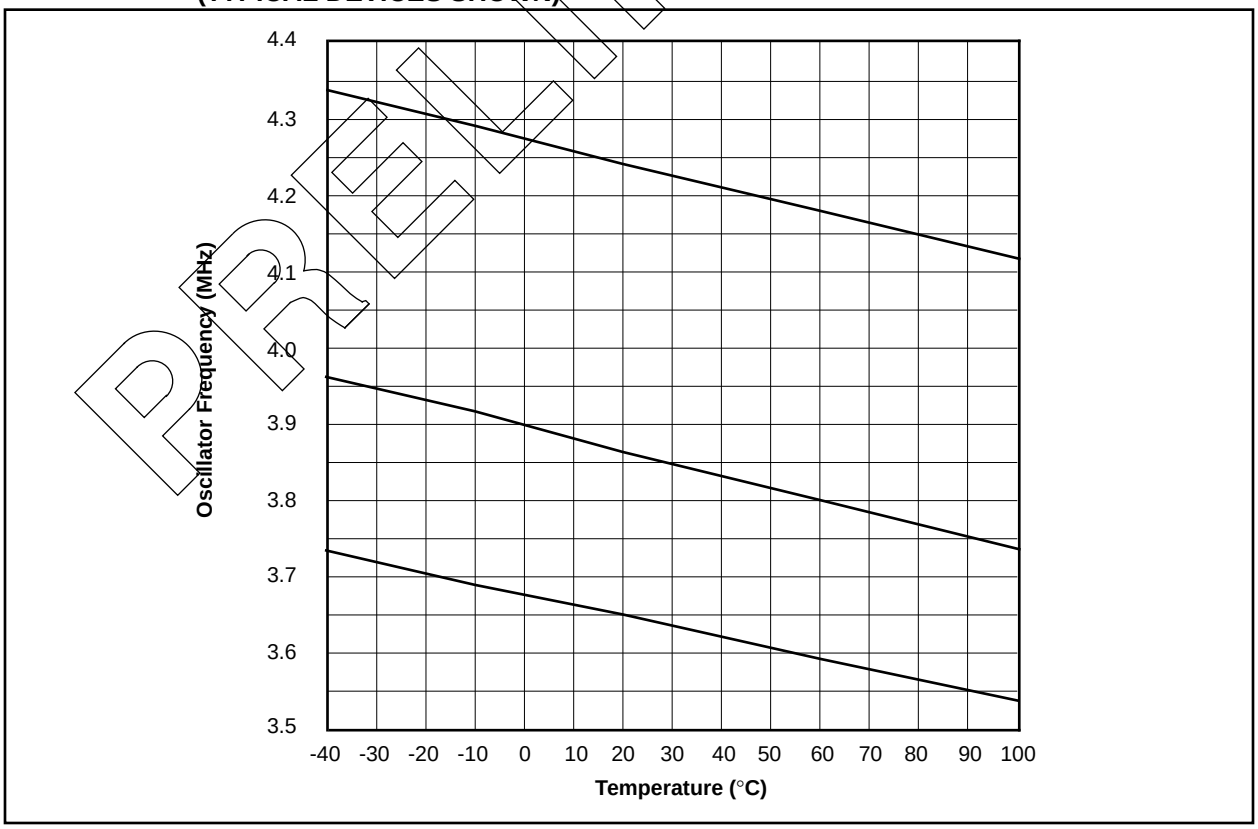


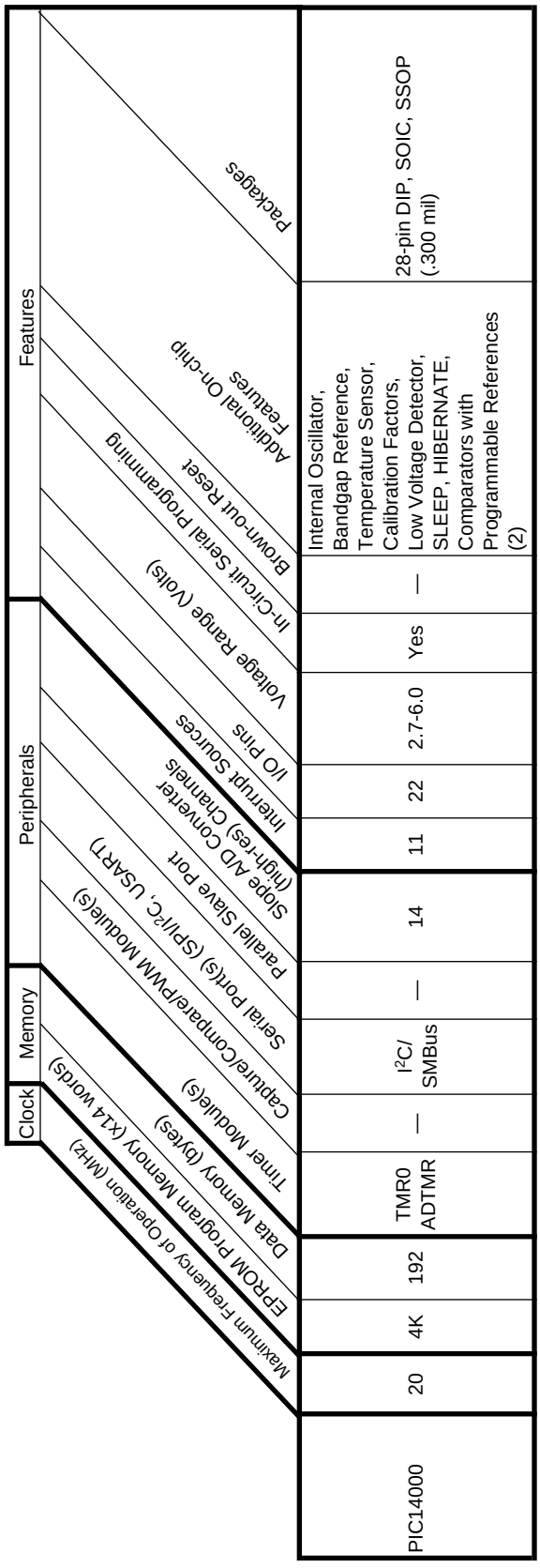
FIGURE 14-8: INTERNAL RC OSCILLATOR FREQUENCY vs. TEMPERATURE (TYPICAL DEVICES SHOWN)



NOTES:

APPENDIX A: PIC16/17 MICROCONTROLLERS

A.1 PIC14000 Devices



A.2 PIC16C5X Family of Devices

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory (x12 words)	EPROM	ROM	RAM Data Memory (bytes)	Timer Module(s)	I/O Pins	Voltage Range (Volts)	Number of Instructions	Packages
PIC16C52	4	384	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC	
PIC16C54	20	512	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	
PIC16C54A	20	512	—	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	
PIC16CR54A	20	—	512	25	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	
PIC16C55	20	512	—	24	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP	
PIC16C56	20	1K	—	25	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	
PIC16C57	20	2K	—	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP	
PIC16CR57B	20	—	2K	72	TMR0	20	2.5-6.25	33	28-pin DIP, SOIC, SSOP	
PIC16C58A	20	2K	—	73	TMR0	12	2.0-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	
PIC16CR58A	20	—	2K	73	TMR0	12	2.5-6.25	33	18-pin DIP, SOIC; 20-pin SSOP	

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

A.3 PIC16CXXX Family of Devices

	Clock	Memory			Peripherals		Features			
	Maximum Frequency of Operation (MHz)	EPROM	Program Memory (x14 words)	Data Memory (bytes)	Timer Module(s)	Comparator(s)	Internal Reference Voltage	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Brown-out Reset	Packages
PIC16C554	20	512	80		TMR0	—	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C556	20	1K	80		TMR0	—	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C558	20	2K	128		TMR0	—	—	3	13	2.5-6.0	—	18-pin DIP, SOIC; 20-pin SSOP
PIC16C620	20	512	80		TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C621	20	1K	80		TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C622	20	2K	128		TMR0	2	Yes	4	13	2.5-6.0	Yes	18-pin DIP, SOIC; 20-pin SSOP

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16C6XXX Family devices use serial programming with clock pin RB6 and data pin RB7.

A.4 PIC16C6X Family of Devices

	Clock	Memory				Peripherals						Features			
	Maximum Frequency of Operation (MHz)	Program Memory (X14 words)	EPROM	ROM	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C62	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	—		28-pin SDIP, SOIC, SSOP
PIC16C62A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes		28-pin SDIP, SOIC, SSOP
PIC16CR62(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	7	22	2.5-6.0	Yes	Yes		28-pin SDIP, SOIC, SSOP
PIC16C63	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes		28-pin SDIP, SOIC
PIC16CR63(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	10	22	2.5-6.0	Yes	Yes		28-pin SDIP, SOIC
PIC16C64	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	2.5-6.0	Yes	—		40-pin DIP; 44-pin PLCC, MQFP
PIC16C64A(1)	20	2K	—	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	2.5-6.0	Yes	Yes		40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR64(1)	20	—	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	Yes	8	33	2.5-6.0	Yes	Yes		40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16C65	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	—		40-pin DIP; 44-pin PLCC, MQFP
PIC16C65A(1)	20	4K	—	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes		40-pin DIP; 44-pin PLCC, MQFP, TQFP
PIC16CR65(1)	20	—	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	11	33	2.5-6.0	Yes	Yes		40-pin DIP; 44-pin PLCC, MQFP, TQFP

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16C6X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

A.5 PIC16C7X Family of Devices

	Clock		Memory			Peripherals						Features			
	Maximum Frequency of Operation (MHz)	EPROM Program Memory (x14 words)	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	A/D Converter (8-bit) Channels	Interrupt Sources	I/O Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C710	20	512	36	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C71	20	1K	36	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	—	18-pin DIP, SOIC
PIC16C711	20	1K	68	TMR0	—	—	—	4	4	13	2.5-6.0	Yes	Yes	18-pin DIP, SOIC; 20-pin SSOP
PIC16C72	20	2K	128	TMR0, TMR1, TMR2	1	SPI/I²C	—	5	8	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC, SSOP
PIC16C73	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	5	11	22	2.5-6.0	Yes	—	28-pin SDIP, SOIC
PIC16C73A(1)	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	—	5	11	22	2.5-6.0	Yes	Yes	28-pin SDIP, SOIC
PIC16C74	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	8	12	33	2.5-6.0	Yes	—	40-pin DIP; 44-pin PLCC, MQFP
PIC16C74A(1)	20	4K	192	TMR0, TMR1, TMR2	2	SPI/I²C, USART	Yes	8	12	33	2.5-6.0	Yes	Yes	40-pin DIP; 44-pin PLCC, MQFP, TQFP

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

All PIC16C7X Family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

A.6 PIC16C8X Family of Devices

	Clock	Memory				Peripherals		Features	
	Maximum Frequency of Operation (MHz)	Program Memory	Flash	EEPROM	ROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Module(s)	Interrupt Sources	I/O Pins	Voltage Range (Volts)	Packages
PIC16C84	10	—	1K	—	36	64	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F84(1)	10	1K	—	—	68	64	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR84(1)	10	—	—	1K	68	64	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16F83(1)	10	512	—	—	36	64	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC
PIC16CR83(1)	10	—	—	512	36	64	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.
All PIC16C8X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

A.7 PIC16C9XX Family Of Devices

	Clock	Memory			Peripherals							Features			
	Maximum Frequency of Operation (MHz)	EPROM Program Memory	Data Memory (bytes)	Timer Module(s)	Capture/Compare/PWM Module(s)	Serial Port(s) (SPI/I²C, USART)	Parallel Slave Port	A/D Converter (8-bit) Channels	LCD Module	Interrupt Sources	I/O Pins	Input Pins	Voltage Range (Volts)	In-Circuit Serial Programming	Brown-out Reset	Packages
PIC16C923	8	4K	176	TMR0, TMR1, TMR2	1	SPI/I²C	—	—	4 Com 32 Seg	8	25	27	3.0-6.0	Yes	—	64-pin SDIP(1), TQFP, 68-pin PLCC, DIE
PIC16C924	8	4K	176	TMR0, TMR1, TMR2	1	SPI/I²C	—	5	4 Com 32 Seg	9	25	27	3.0-6.0	Yes	—	64-pin SDIP(1), TQFP, 68-pin PLCC, DIE

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC16CXX Family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local Microchip representative for availability of this package.

A.8 PIC17CXX Family of Devices

	Clock	Memory					Peripherals						Features			
	Maximum Frequency of Operation (MHz)	EPROM	ROM	Program Memory (Words)	RAM Data Memory (bytes)	Timer Module(s)	Captures	PWMs	Serial Port(s) (USART)	Hardware Multiply	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, MQFP
PIC17C42A	25	2K	—	232		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-5.5	58	40-pin DIP; 44-pin PLCC, MQFP
PIC17CR42	25	—	2K	232		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-5.5	58	40-pin DIP; 44-pin PLCC, MQFP
PIC17C43	25	4K	—	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17CR43	25	—	4K	454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP
PIC17C44	25	8K		454		TMR0,TMR1, TMR2,TMR3	2	2	Yes	Yes	Yes	11	33	2.5-6.0	58	40-pin DIP; 44-pin PLCC, TQFP, MQFP

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

A.9 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 A-1: PIN COMPATIBLE DEVICES

Pin Compatible Devices	Package
PIC12C508, PIC12C509	8-pin
PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622, PIC16C710, PIC16C71, PIC16C711, PIC16C83, PIC16CR83, PIC16C84, PIC16C84A, PIC16CR84	18-pin 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16C62, PIC16CR62, PIC16C62A, PIC16C63, PIC16C72, PIC16C73, PIC16C73A	28-pin
PIC16C64, PIC16CR64, PIC16C64A, PIC16C65, PIC16C65A, PIC16C74, PIC16C74A	40-pin
PIC17C42, PIC17C43, PIC17C44	40-pin

NOTES:

INDEX

LIST OF EXAMPLES

LIST OF FIGURES

LIST OF TABLES

ON-LINE SUPPORT

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp.mchip.com/biz/mchip

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products

Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

Internet:

You can telnet or ftp to the Microchip BBS at the address:

mchipbbs.microchip.com

CompuServe Communications Network:

When using the BBS via the Compuserve 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 Microchip BBS. 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 various baud rates depending on the local point of access.

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 the <Enter> key and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress the <Enter> key and “Host Name:” will appear.
5. Type MCHIPBBS, depress the <Enter> key and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe 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 the <Enter> key and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

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. 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.

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.

960513

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: **PIC14000** Literature Number: **DS40122B**

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 usefulness? _____

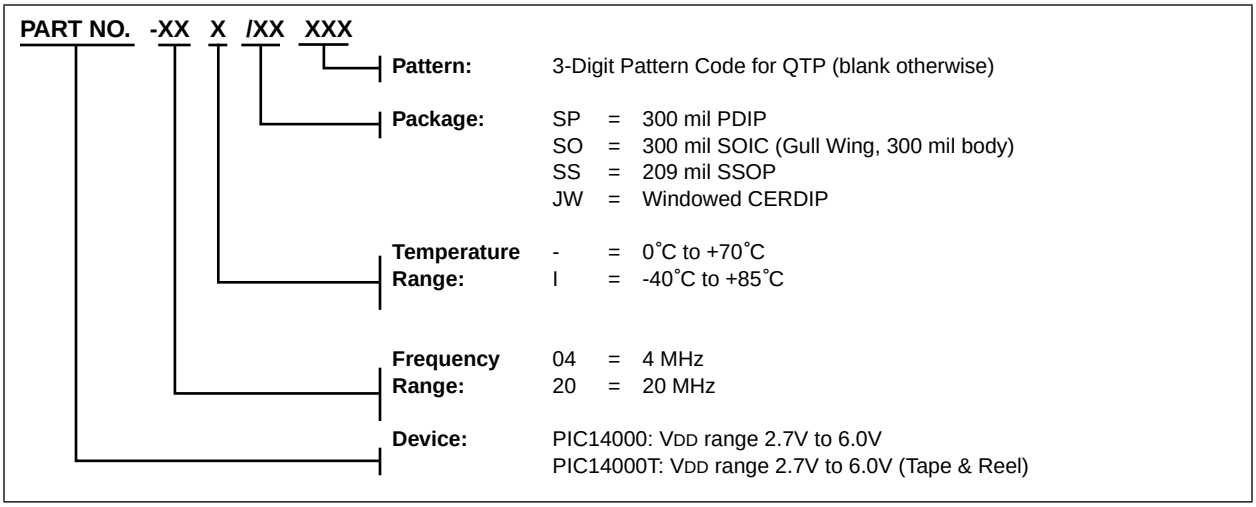
6. Is there any incorrect or misleading information (what and where)? _____

7. How would you improve this document? _____

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

PIC14000 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.



NOTES:

NOTES:

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.microchip.com

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770 640-0034 Fax: 770 640-0307

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

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.
Suite 150
Two Prestige Place
Miamisburg, OH 45342
Tel: 513 291-1654 Fax: 513 291-9175

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

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

Toronto
Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905 405-6279 Fax: 905 405-6253

ASIA/PACIFIC

China
Microchip Technology
Unit 406 of Shanghai Golden Bridge Mansion
2077 Yan'an Road West, Hongiao District
Shanghai, Peoples Republic of China
Tel: 86 21 6275 6060 x 406

Hong Kong
Microchip Technology
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

India
Microchip Technology
No. 6, Legacy, Convent Road
Bangalore 560 025 India
Tel: 91 80 526 3148 Fax: 91 80 558 6606

Korea
Microchip Technology
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
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
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 1628 850303 Fax: 44 1628 850178

France
Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
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 Microchip Technology 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 Colleone Pas Taurus 1
Viale Colleoni 1
20041 Agrate Brianza
Milan Italy
Tel: 39 39 6899939 Fax: 39 39 689 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

7/30/96