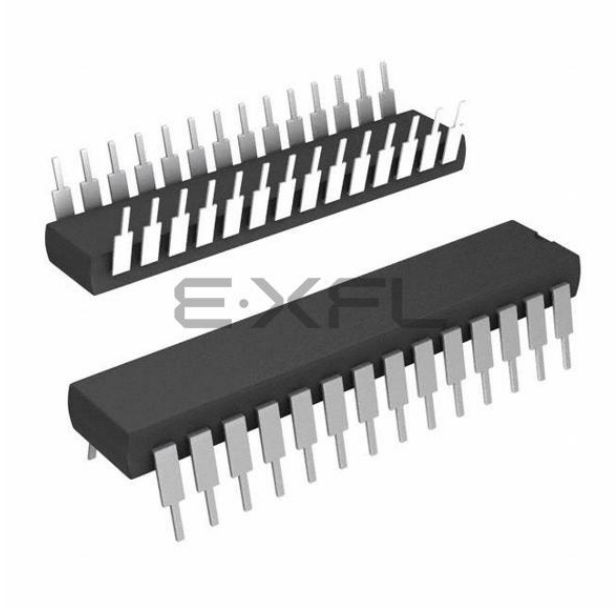


Welcome to E-XFL.COM

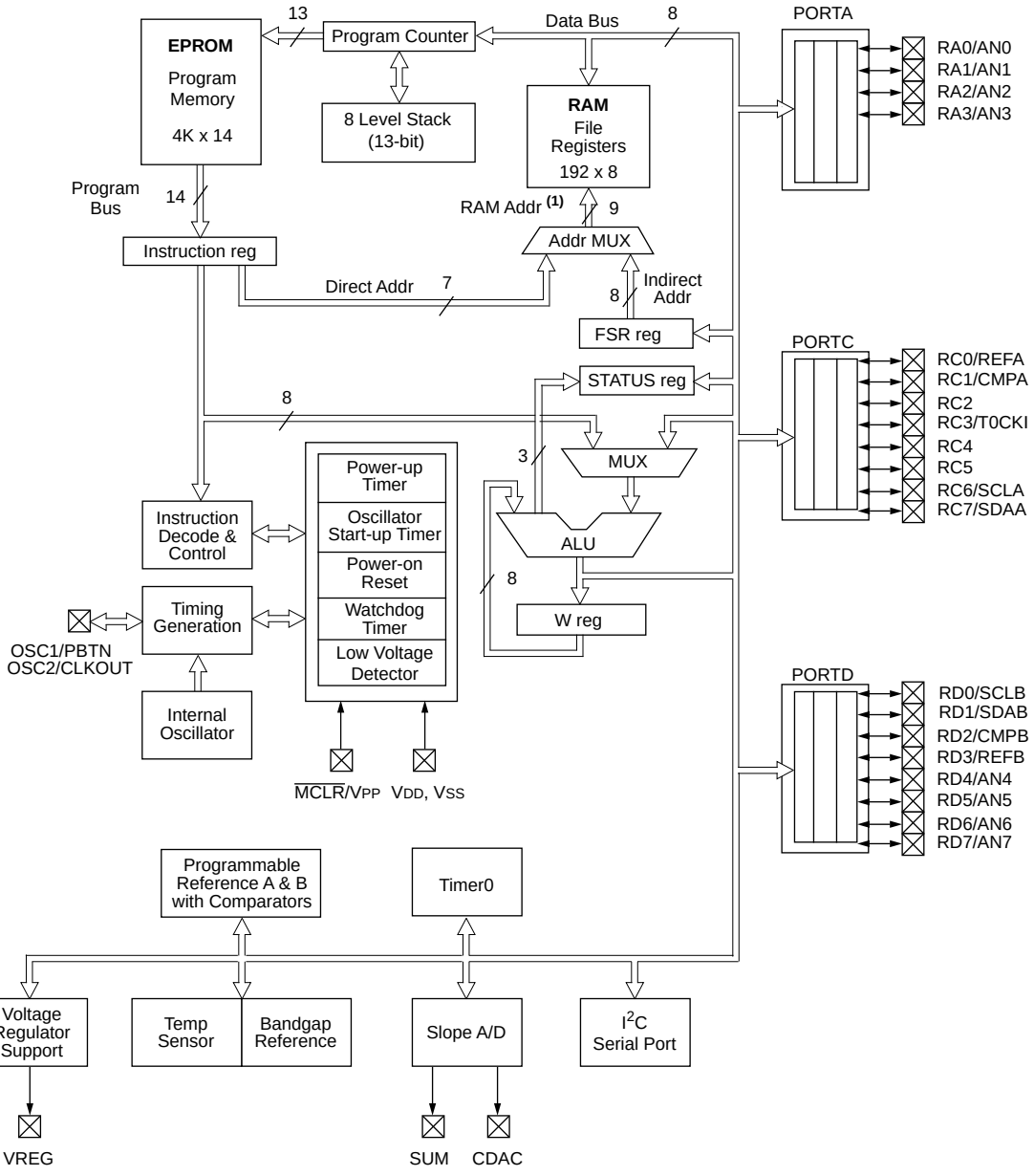
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C
Peripherals	POR, Temp Sensor, WDT
Number of I/O	20
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	Slope A/D
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic14000-20i-sp

FIGURE 3-1: PIC14000 BLOCK DIAGRAM



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

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.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

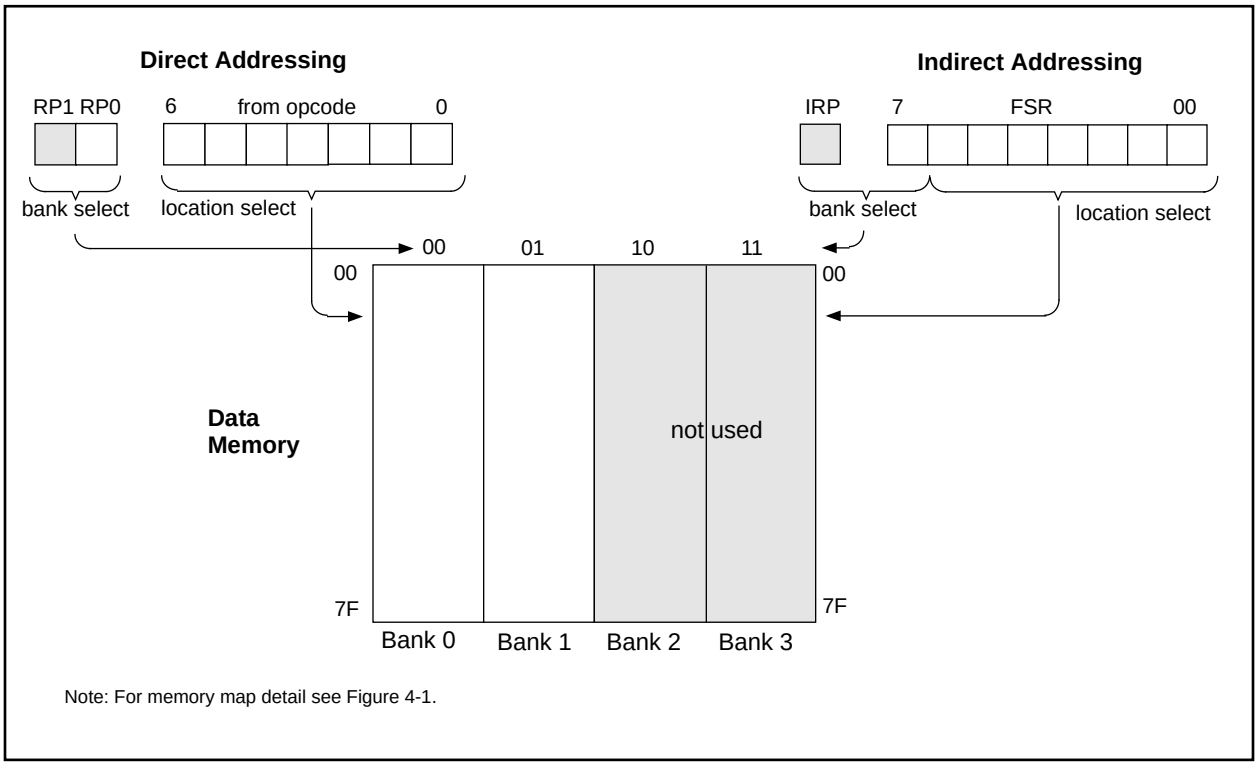
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status 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.

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.

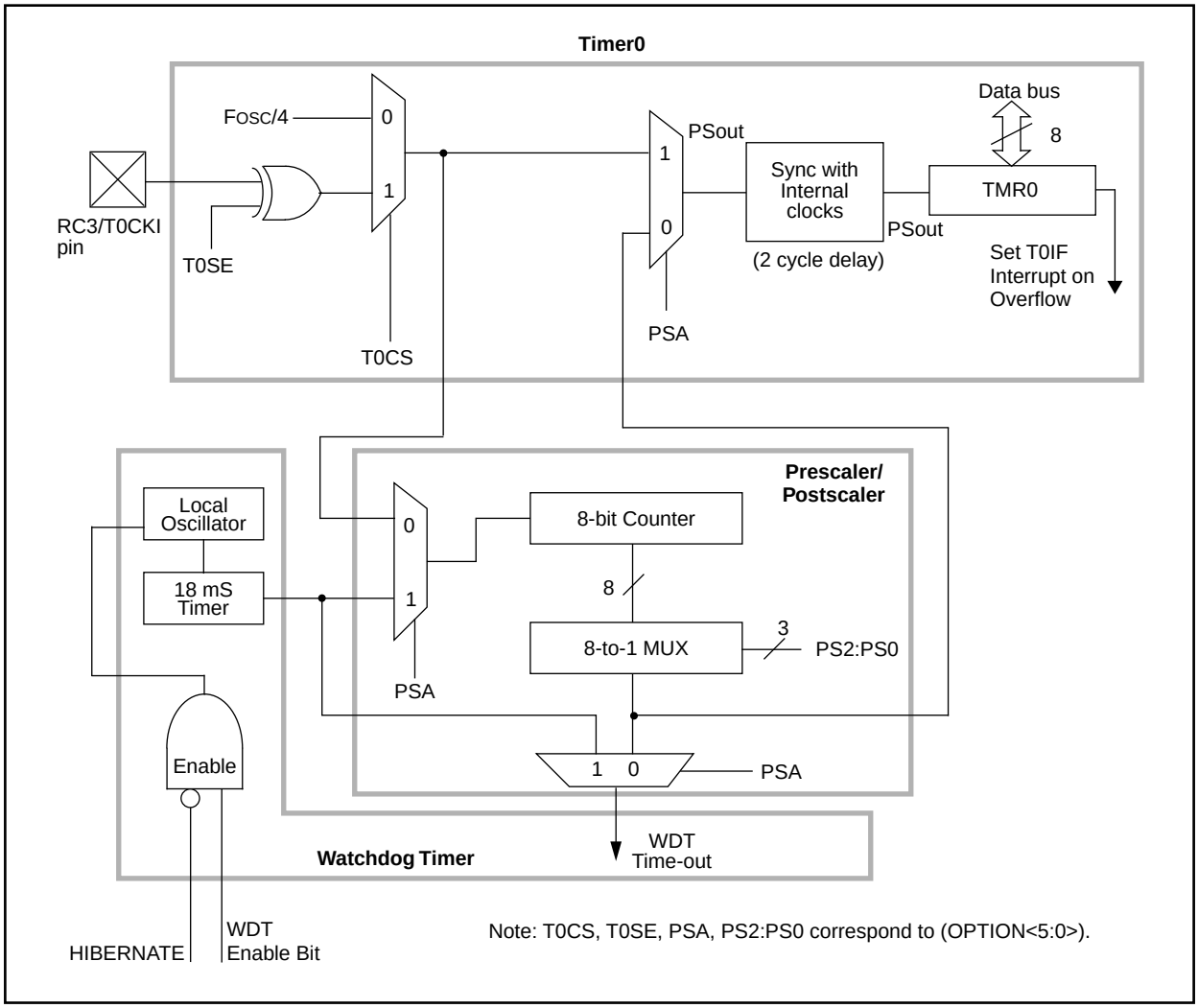
The Timer0 module has the following features:

- 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



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)

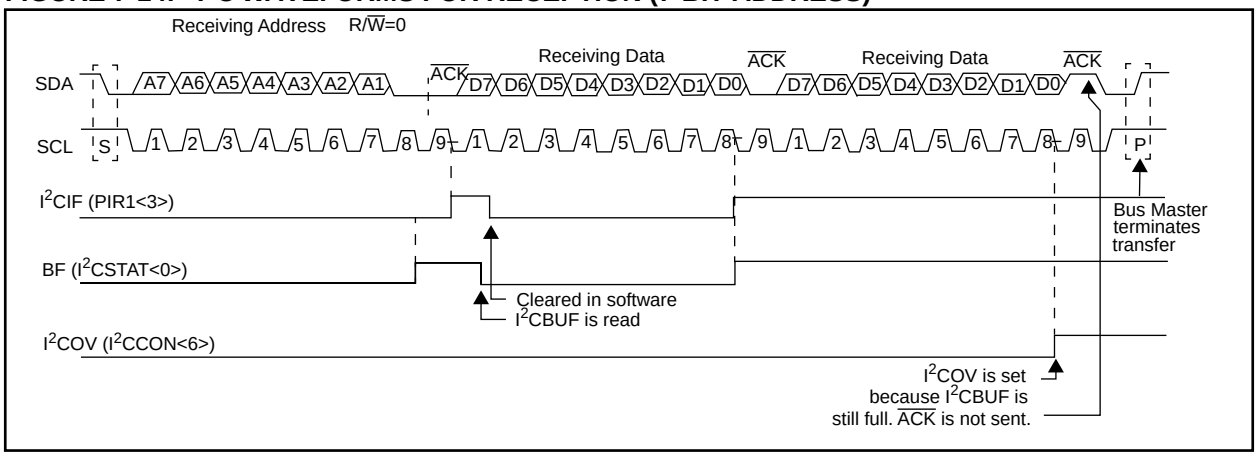


FIGURE 7-17: OPERATION OF THE I^2C IN IDLE_MODE, RCV_MODE OR XMIT_MODE

```
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;
        }
```


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:

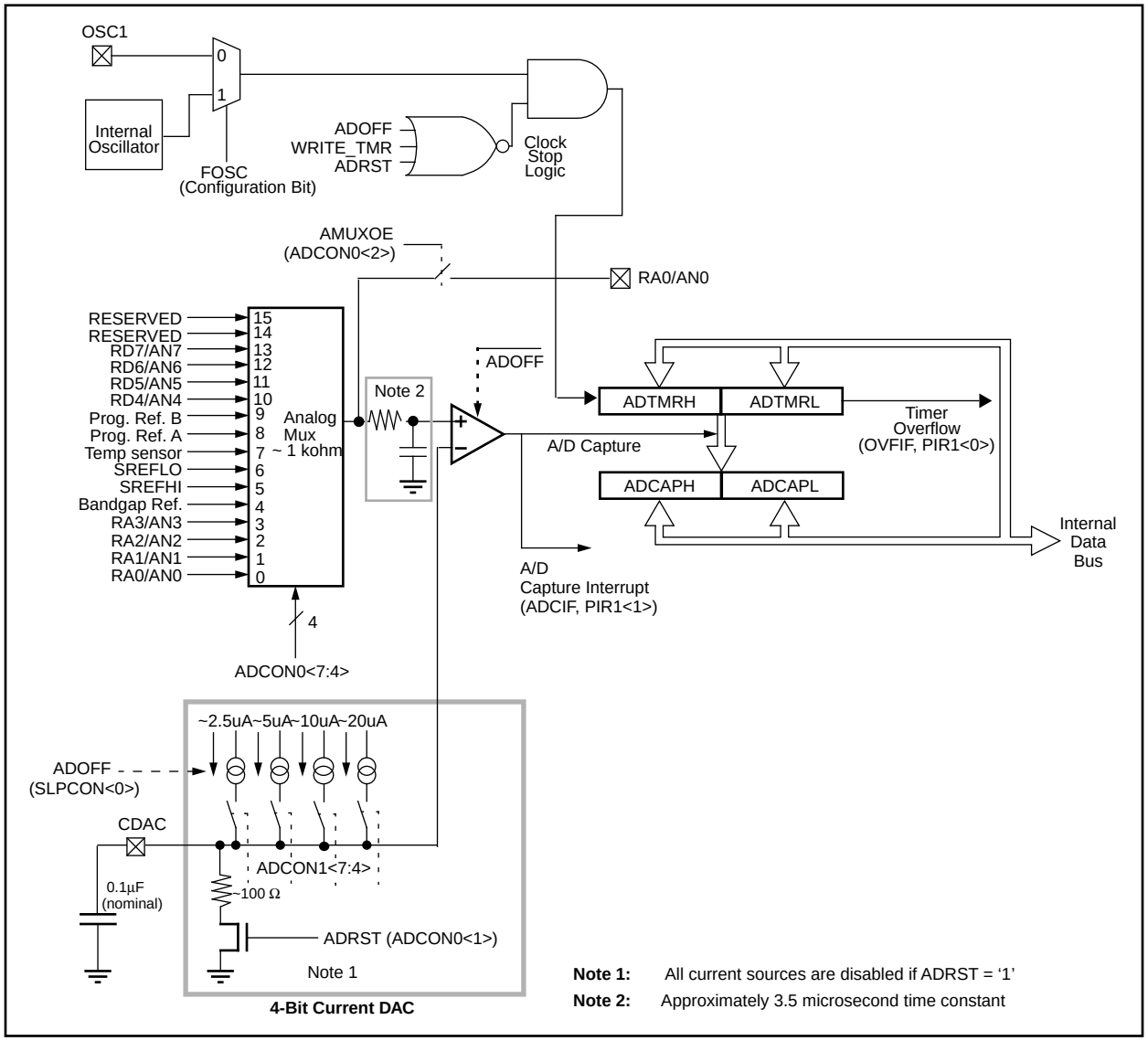
1. capture event
2. timer overflow

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

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 teflon, 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.

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)

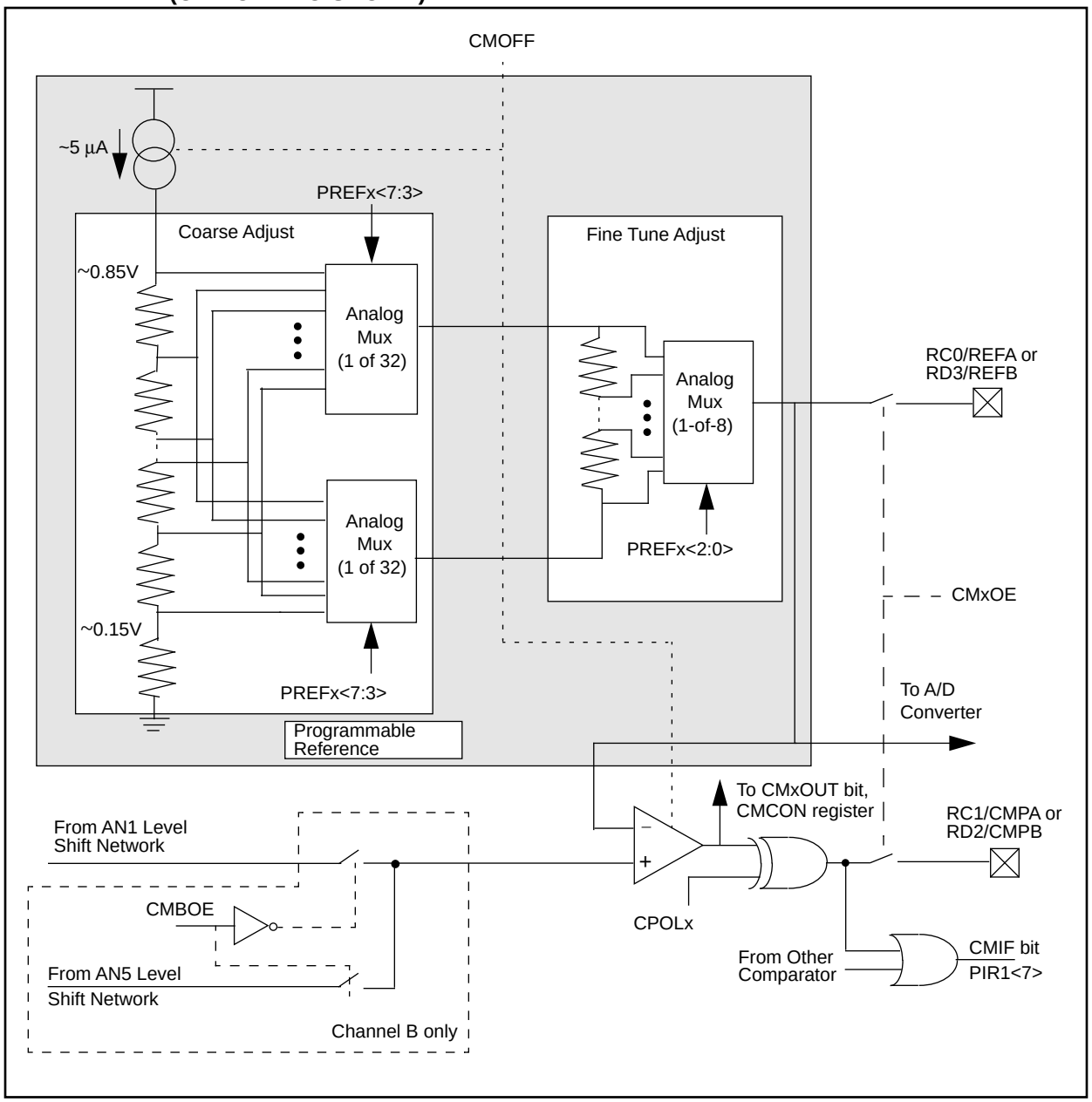


FIGURE 9-5: COMPARATOR CONTROL REGISTER

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.

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.

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.

10.7.1 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.

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

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

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

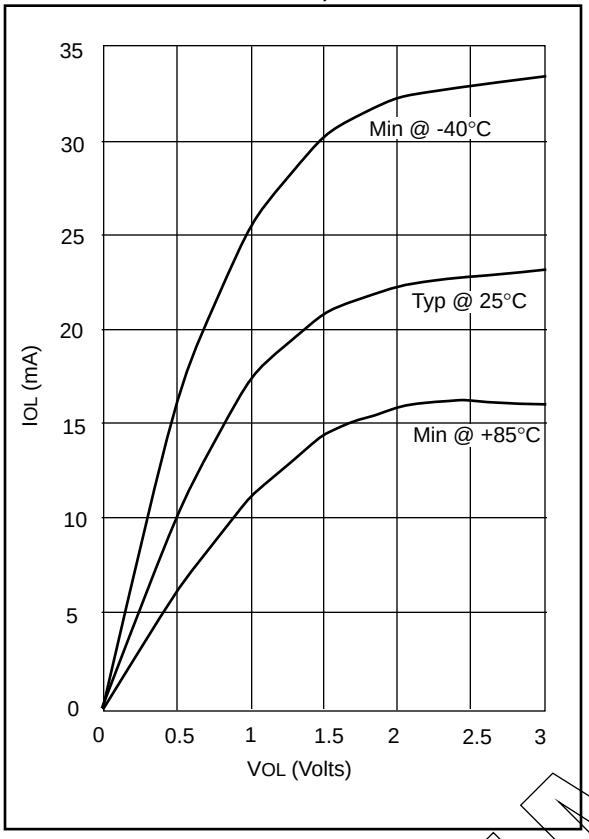
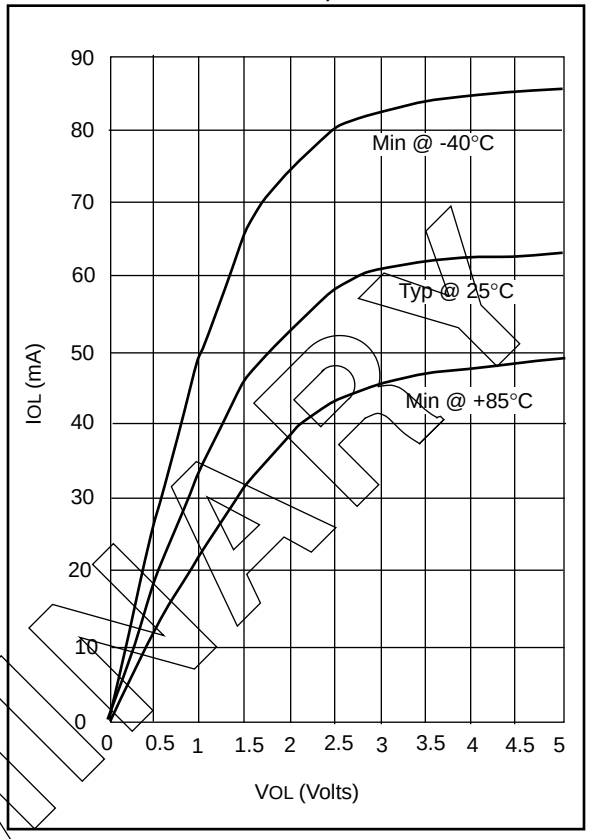


FIGURE 13-21: IOL VS VOL, VDD = 5V*



***NOTE:** All pins except OSC2

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

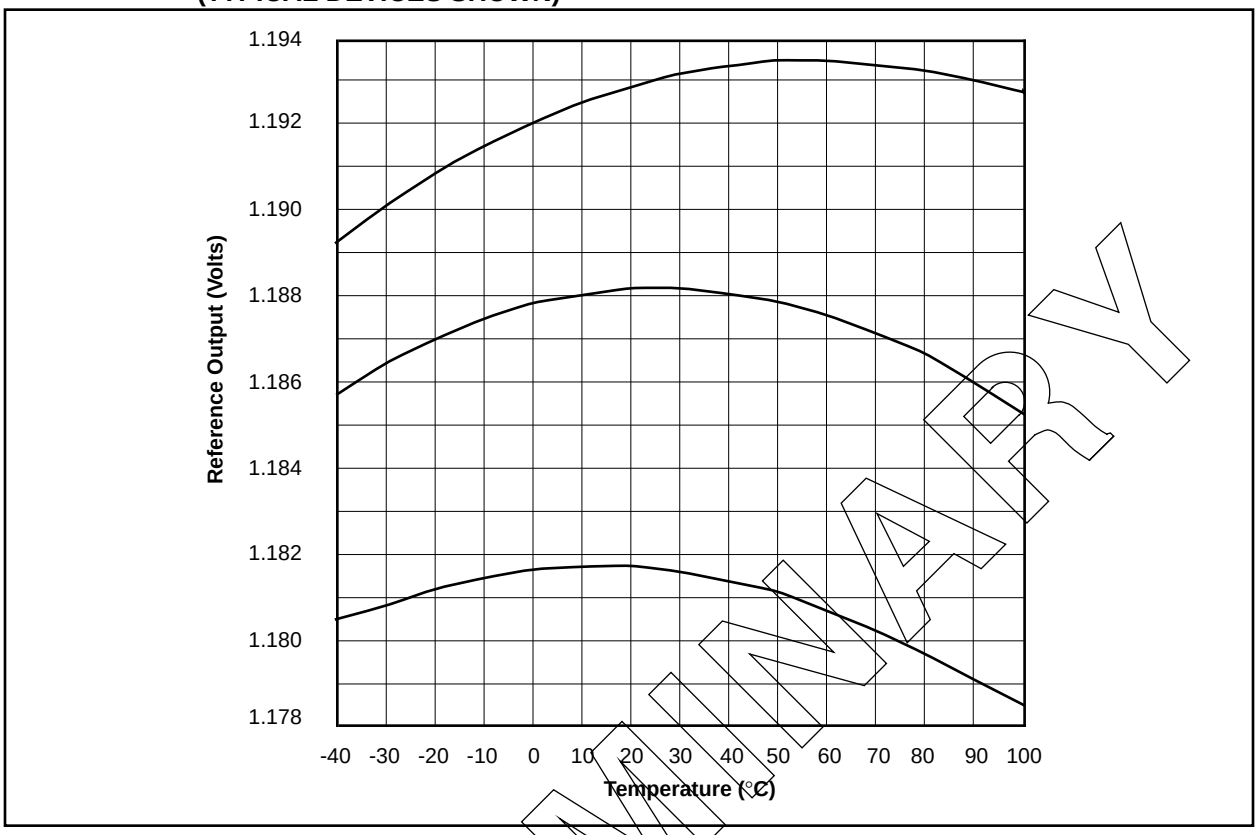
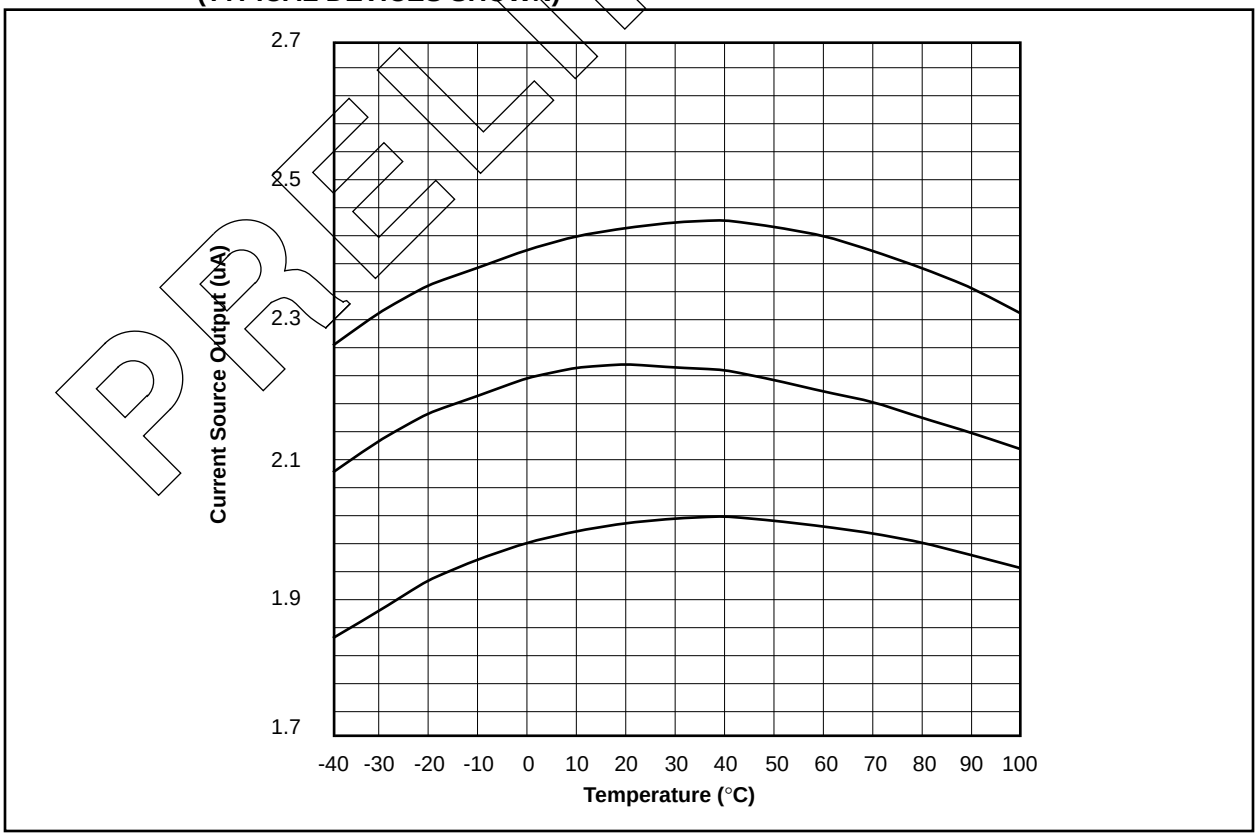
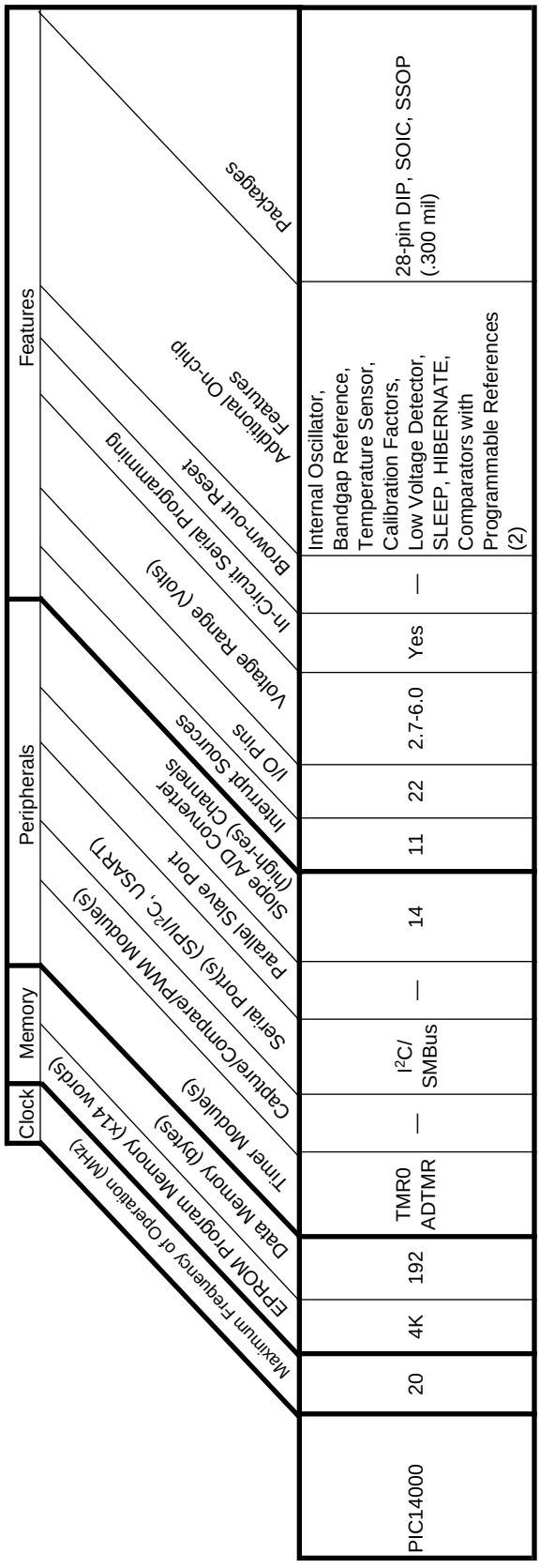


FIGURE 14-2: PROGRAMMABLE CURRENT SOURCE vs. TEMPERATURE (TYPICAL DEVICES SHOWN)



APPENDIX A: PIC16/17 MICROCONTROLLERS

A.1 PIC14000 Devices



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

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — ISO 9001 / QS-9000 REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 certified.