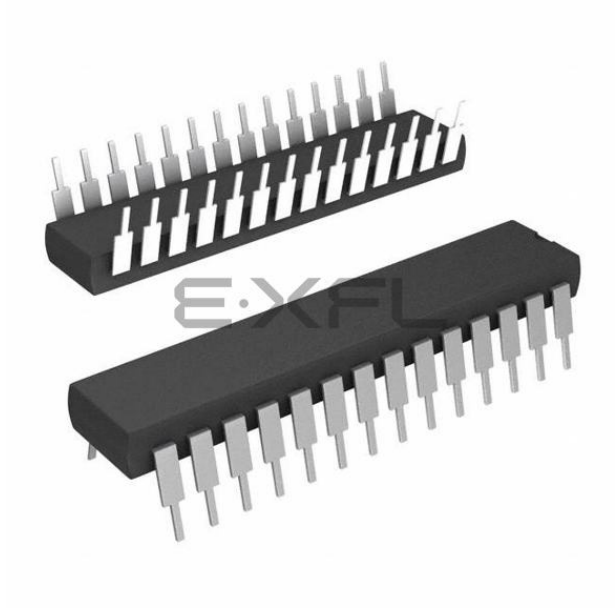


Welcome to E-XFL.COM

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"

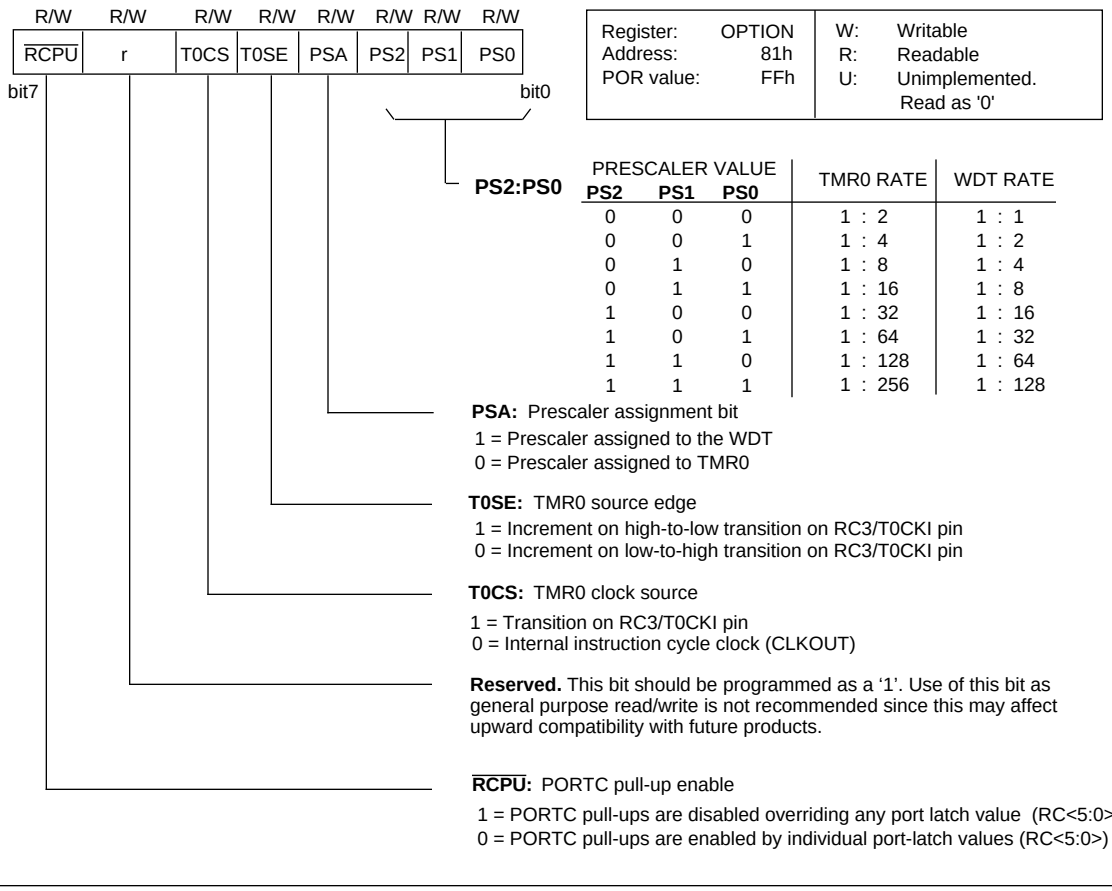
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I²C
Peripherals	POR, Temp Sensor, WDT
Number of I/O	20
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	Slope A/D
Oscillator 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/pic14000-04i-sp

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)

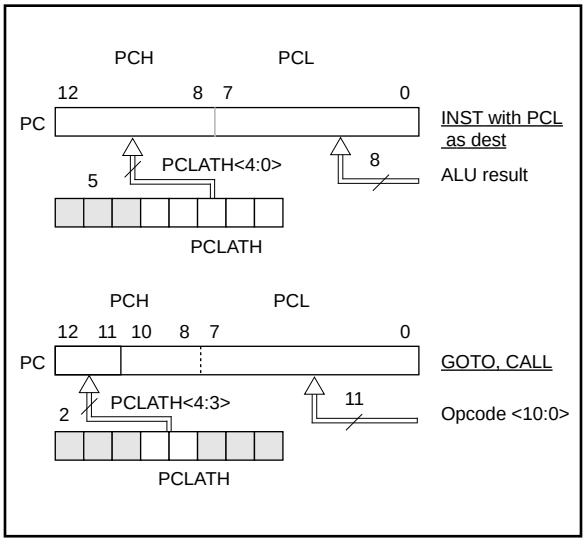
FIGURE 4-4: OPTION REGISTER



4.3 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)
```

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

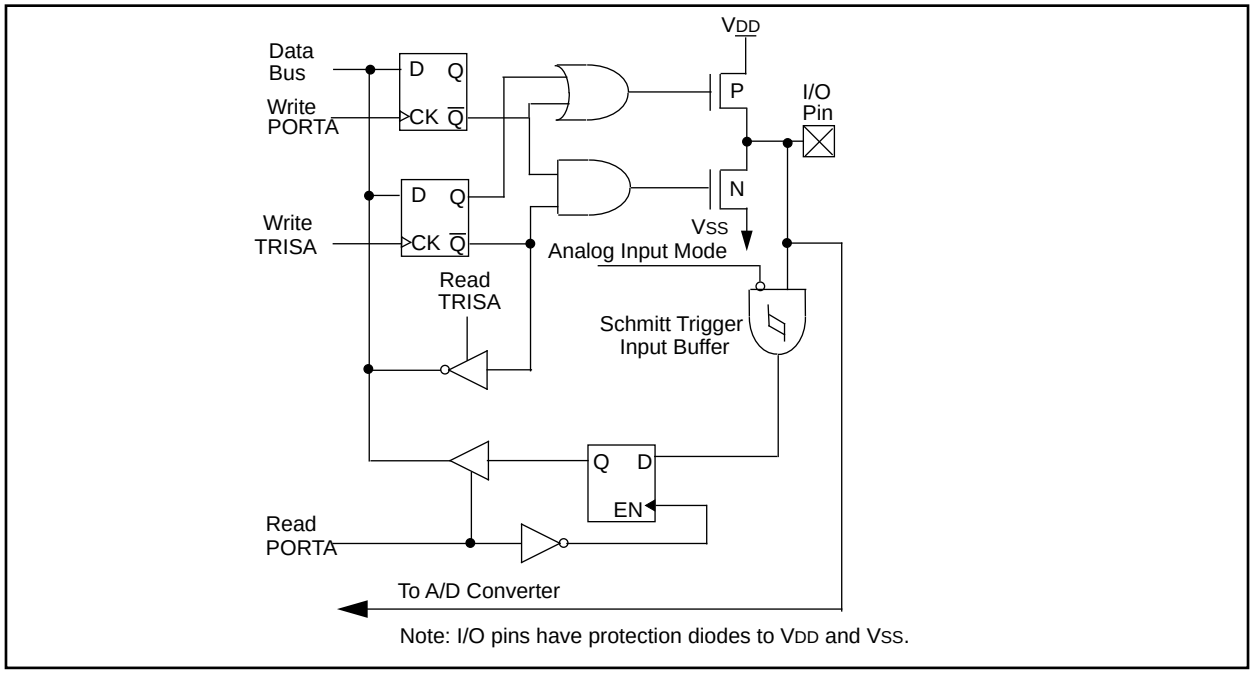


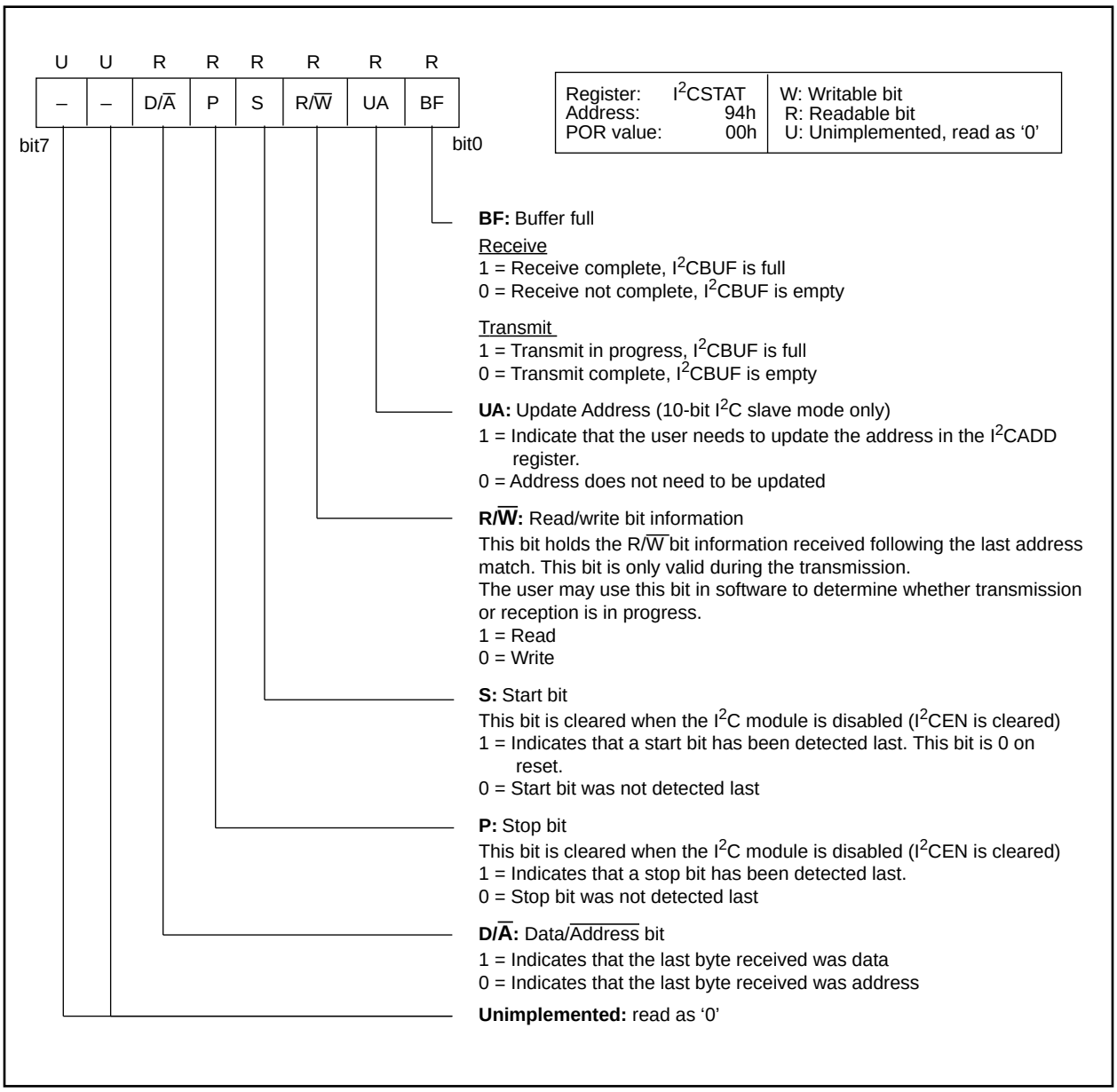
FIGURE 5-6: PORTC DATA REGISTER

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.

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



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

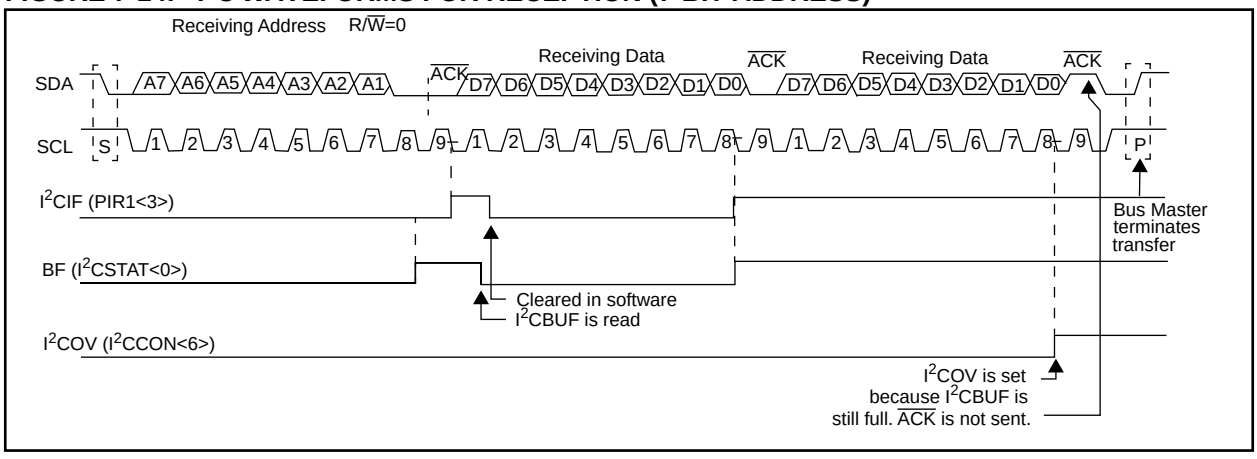
When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the I^2CSTAT register is cleared. The received address is loaded into the I^2CBUF.

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)



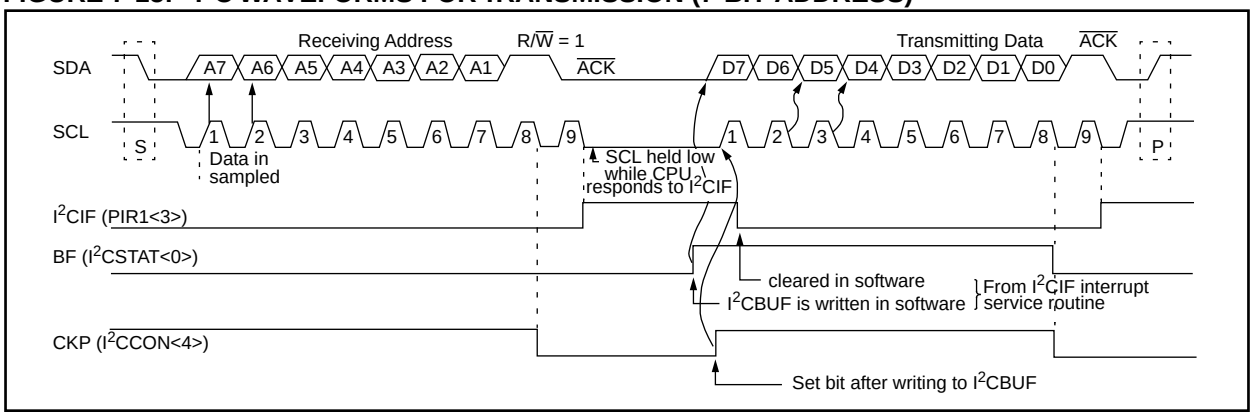
7.5.1.3 TRANSMISSION

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.

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

9.6 Voltage Regulator Output

For systems with a main supply voltage above 6V, an inexpensive, low quiescent current voltage regulator can be formed by connecting the VREG pin to an external resistor and FET as shown in Figure 9-8. This circuit will provide a VDD of about 5V, after the voltage drop across the FET.

FIGURE 9-8: VOLTAGE REGULATOR CIRCUIT

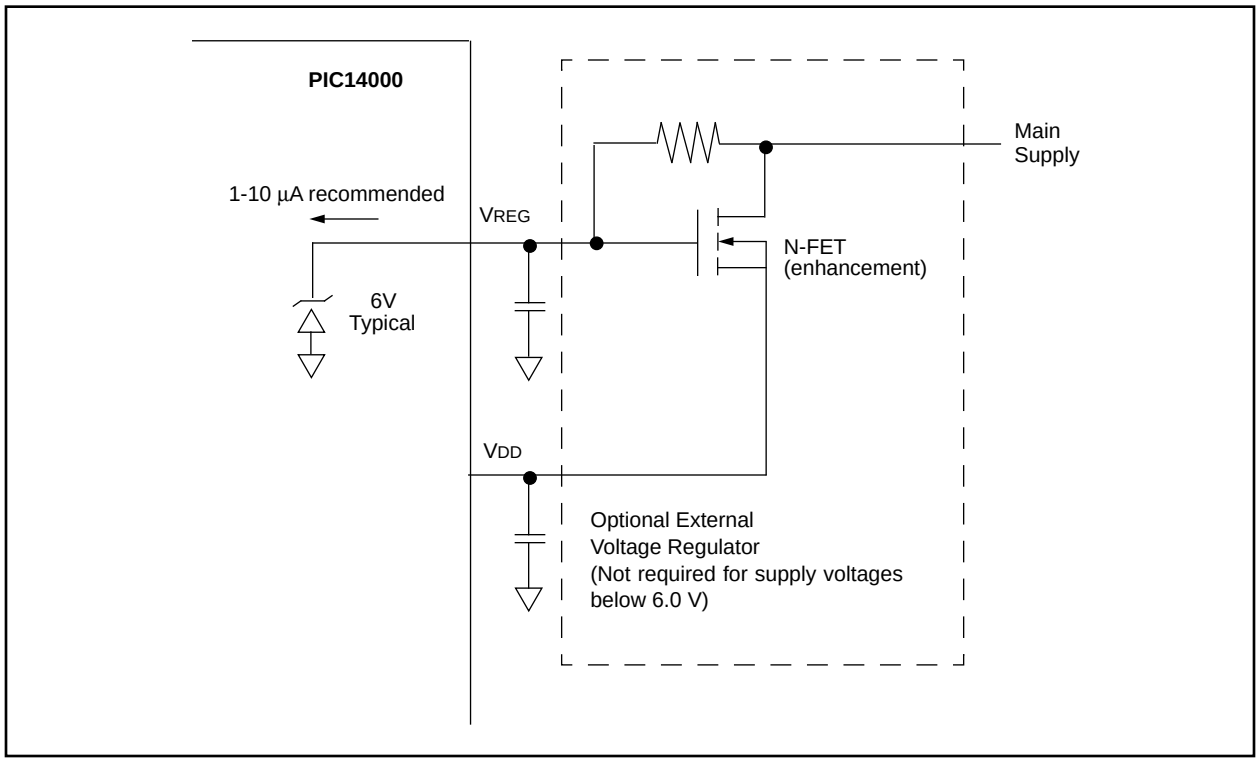
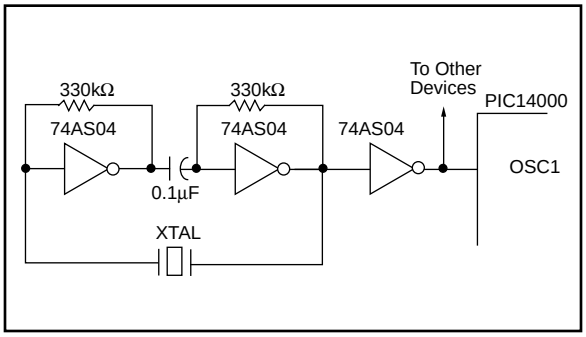


Figure 10-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their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

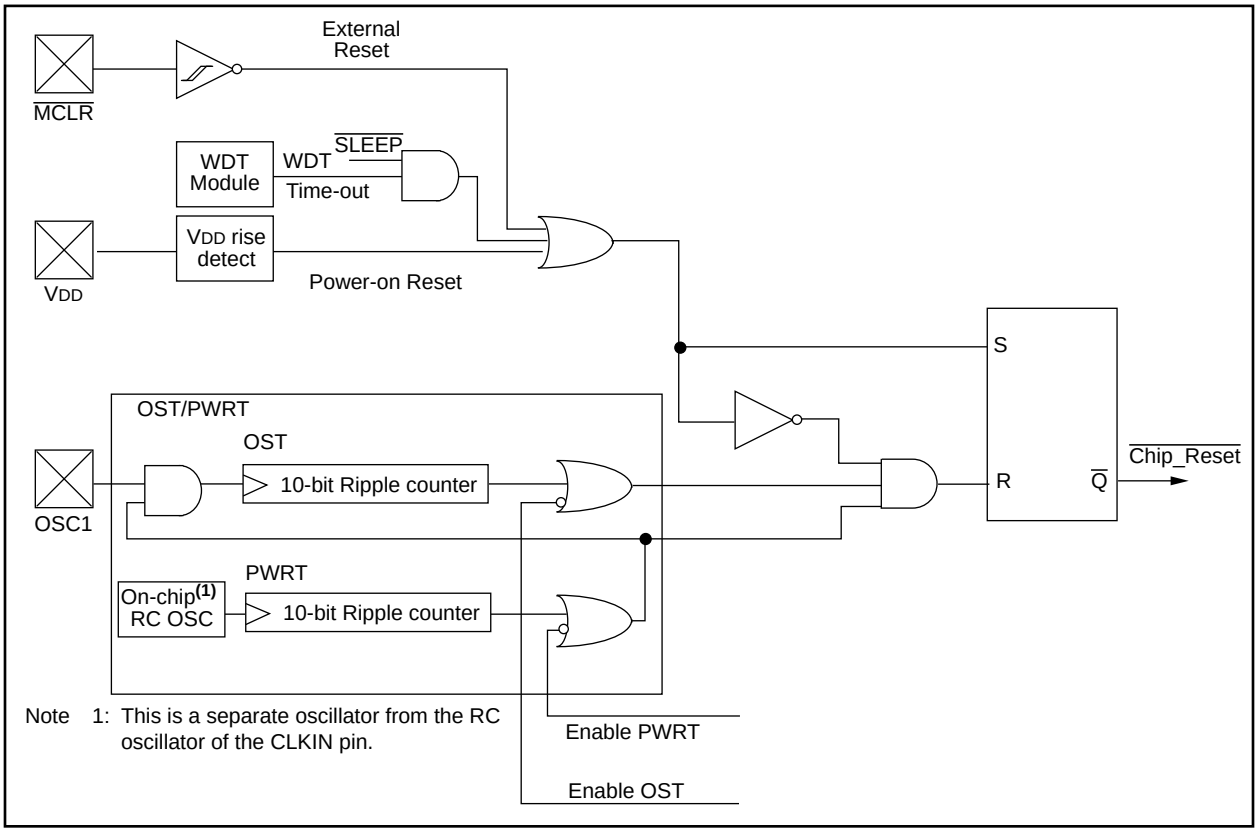
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), on the $\overline{MCLR}$ and WDT Reset, and on $\overline{MCLR}$ Reset during SLEEP. They are not affected 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.

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

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

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

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

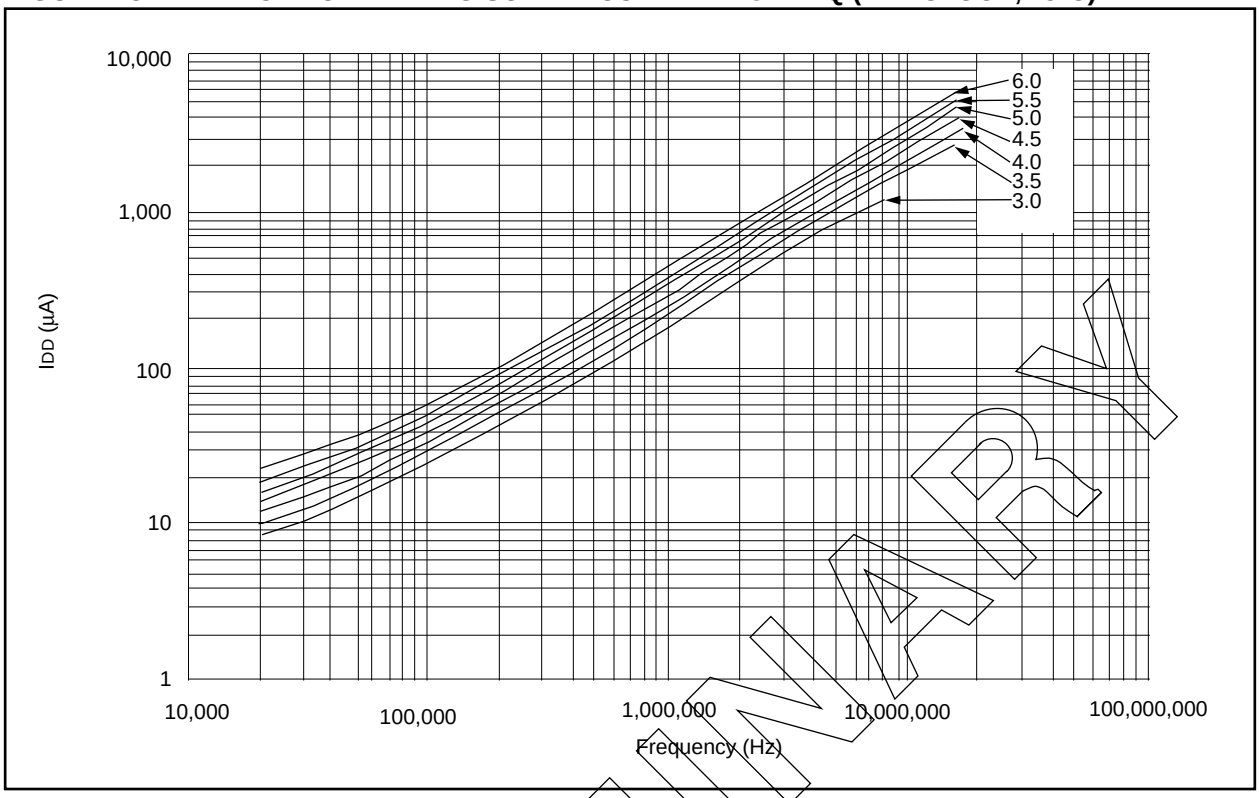


FIGURE 13-13: MAXIMUM OPERATING SUPPLY CURRENT vs FREQ (EXT CLOCK, -40° TO +85°C)

TO BE DETERMINED.

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.

LIST OF TABLES

NOTES: