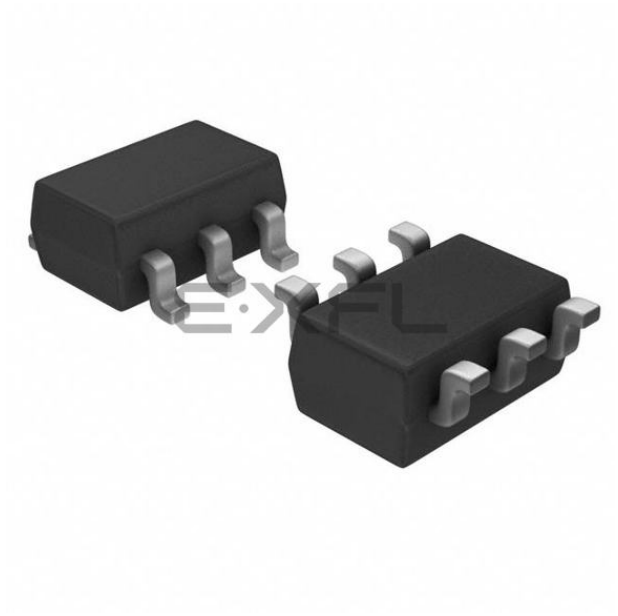


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	4
Program Memory Size	384B (256 x 12)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	SOT-23-6
Supplier Device Package	SOT-23-6
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic10f220-e-ot

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

¤ 2005-2013 Microchip Technology Inc.

3.1 Clocking Scheme/Instruction Cycle

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

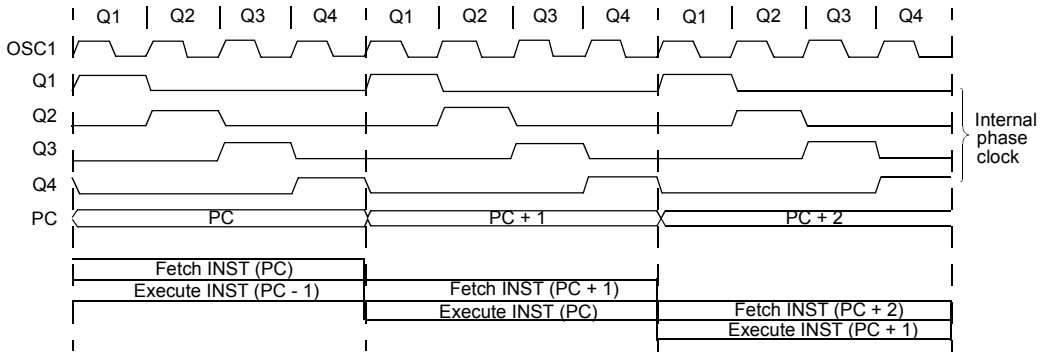
3.2 Instruction Flow/Pipelining

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

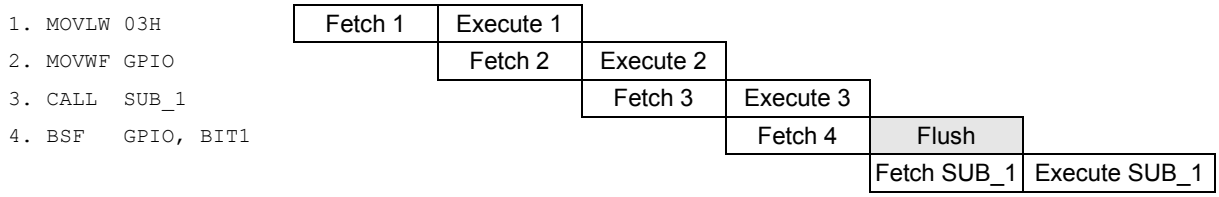
A fetch cycle begins with the PC incrementing in Q1.

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

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



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

REGISTER 4-1: STATUS REGISTER (ADDRESS: 03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **GPWUF:** GPIO Reset bit
1 = Reset due to wake-up from Sleep on pin change
0 = After power-up or other Reset

bit 6 **Reserved:** Do not use. Use of this bit may affect upward compatibility with future products.

bit 5 **Reserved:** Do not use. Use of this bit may affect upward compatibility with future products.

bit 4 **$\overline{TO}$:** Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit 2 **Z:** Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit carry/$\overline{borrow}$ bit (for ADDWF and SUBWF instructions)
ADDWF:
1 = A carry to the 4th low-order bit of the result occurred
0 = A carry to the 4th low-order bit of the result did not occur
SUBWF:
1 = A borrow from the 4th low-order bit of the result did not occur
0 = A borrow from the 4th low-order bit of the result occurred

bit 0 **C:** Carry/$\overline{borrow}$ bit (for ADDWF, SUBWF and RRF, RLF instructions)

ADDWF:	SUBWF:	RRF or RLF:
1 = A carry occurred	1 = A borrow did not occur	Load bit with LSb or MSb, respectively
0 = A carry did not occur	0 = A borrow occurred	

¤ 2005-2013 Microchip Technology Inc.

4.7 Program Counter

As a program instruction is executed, the Program Counter (PC) will contain the address of the next program instruction to be executed. The PC value is increased by one every instruction cycle, unless an instruction changes the PC.

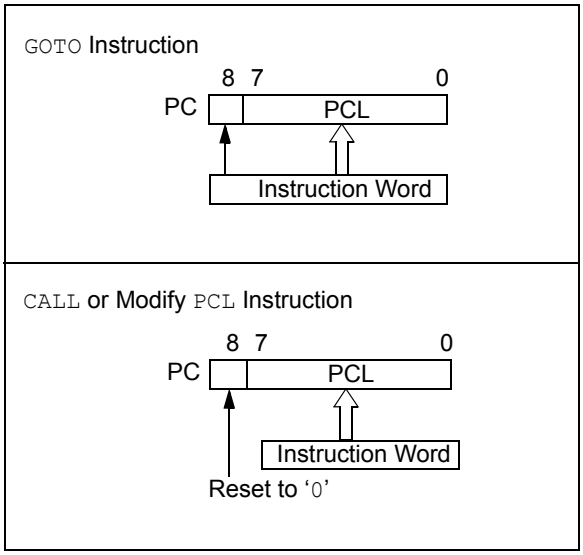
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. The PC Latch (PCL) is mapped to PC<7:0>.

For a CALL instruction or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 4-5).

Instructions where the PCL is the destination or Modify PCL instructions, include MOVWF PC, ADDWF PC and BSF PC, 5.

Note:	Because PC<8> is cleared in the CALL instruction or any Modify PCL instruction, all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

FIGURE 4-5: LOADING OF PC BRANCH INSTRUCTIONS



4.7.1 EFFECTS OF RESET

The PC is set upon a Reset, which means that the PC addresses the last location in program memory (i.e., the oscillator calibration instruction). After executing MOVLW XX, the PC will roll over to location 0000h and begin executing user code.

4.8 Stack

The PIC10F220 device has a 2-deep, 8-bit wide hardware PUSH/POP stack.

The PIC10F222 device has a 2-deep, 9-bit wide hardware PUSH/POP stack.

A CALL instruction will PUSH the current value of stack 1 into stack 2 and then PUSH the current PC value, incremented by one, into stack level 1. If more than two sequential CALL's are executed, only the most recent two return addresses are stored.

A RETLW instruction will POP the contents of stack level 1 into the PC and then copy stack level 2 contents into level 1. If more than two sequential RETLW's are executed, the stack will be filled with the address previously stored in level 2.

Note 1:	The W register will be loaded with the literal value specified in the instruction. This is particularly useful for the implementation of data look-up tables within the program memory.
2:	There are no Status bits to indicate stack overflows or stack underflow conditions.
3:	There are no instructions mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL and RETLW instructions.

4.9 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). This is indirect addressing.

4.9.1 INDIRECT ADDRESSING

- Register file 09 contains the value 10h
- Register file 0A contains the value 0Ah
- Load the value 09 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 0A)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using Indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

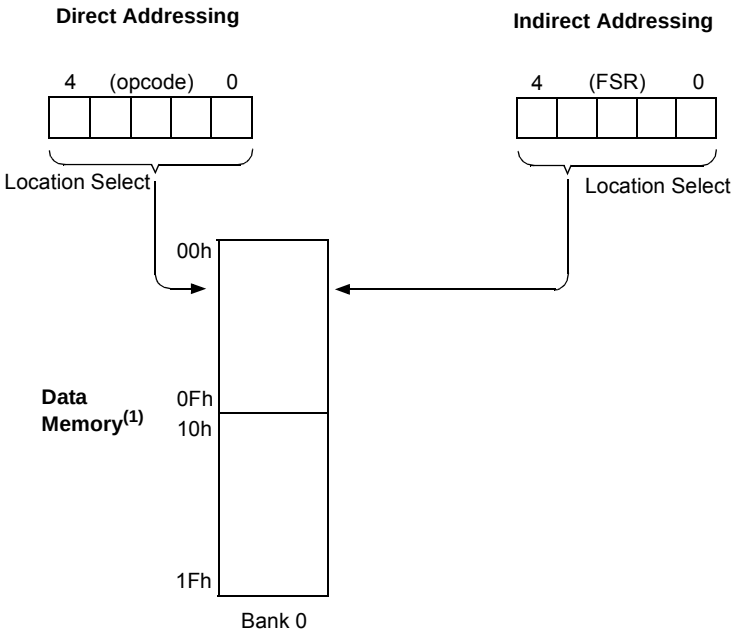
```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
            :           ;YES, continue
            :
```

The FSR is a 5-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

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

FIGURE 4-6: DIRECT/INDIRECT ADDRESSING



Note 1: For register map detail, see **Section 4.3 "Data Memory Organization"**.

¤ 2005-2013 Microchip Technology Inc.

TABLE 5-3: SUMMARY OF PORT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
N/A	TRISGPIO	—	—	—	—	I/O Control Registers				---- 1111	---- 1111
N/A	OPTION	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03h	STATUS	GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	q00q quuu(1)
06h	GPIO	—	—	—	—	GP3	GP2	GP1	GP0	---- xxxx	---- uuuu

Legend: Shaded cells not used by PORT registers, read as '0', – = unimplemented, read as '0', x = unknown, u = unchanged, q = depends on condition.

Note 1: If Reset was due to wake-up on pin change, then bit 7 = 1. All other Resets will cause bit 7 = 0.

5.4 I/O Programming Considerations

5.4.1 BIDIRECTIONAL I/O PORTS

Some instructions operate internally as read followed by write operations. The BCF and BSF instructions, for example, read the entire port into the CPU, execute the bit operation and re-write the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a BSF operation on bit 2 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit 2 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bidirectional I/O pin (say bit 0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Example 5-1 shows the effect of two sequential Read-Modify-Write instructions (e.g., BCF, BSF, etc.) on an I/O port.

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

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

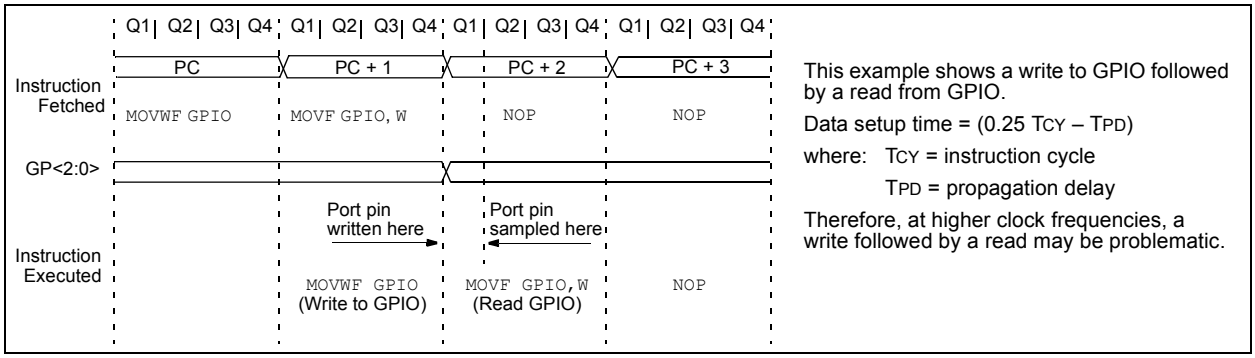
```
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;                 GPIO latch   GPIO pins
;                 ----------   ----------
 BCF    GPIO,  1 ;---- pp01    ---- pp11
 BCF    GPIO,  0 ;---- pp10    ---- pp11
 MOVLW  007h;
 TRIS   GPIO     ;---- pp10    ---- pp11
;
```

Note: The user may have expected the pin values to be ---- pp00. The second BCF caused GP1 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 (Figure 5-5). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. 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 5-5: SUCCESSIVE I/O OPERATION



This example shows a write to GPIO followed by a read from GPIO.

Data setup time = (0.25 TCY – TPD)

where: TCY = instruction cycle

TPD = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

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

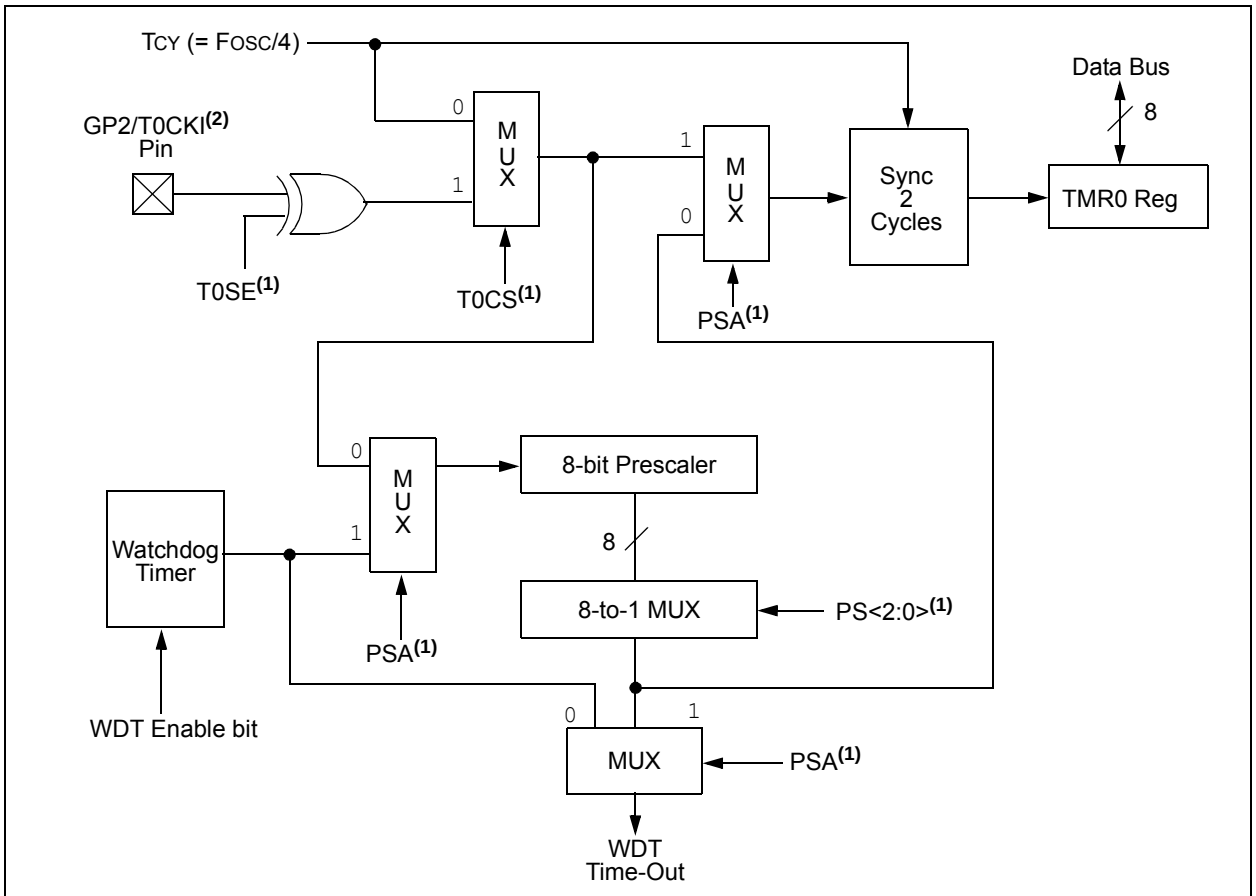
```
CLRWDT              ;Clear WDT
CLRF   TMR0         ;Clear TMR0 & Prescaler
MOVLW  '00xx1111'b  ;These 3 lines (5, 6, 7)
OPTION              ;are required only if
                    ;desired
CLRWDT              ;PS<2:0> are 000 or 001
MOVLW  '00xx1xxx'b  ;Set Postscaler to
OPTION              ;desired WDT rate
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled. A CLRWDT instruction should be executed before switching the prescaler.

EXAMPLE 6-2: CHANGING PRESCALER (WDT o TIMER0)

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

FIGURE 6-5: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



Note 1: T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.

2: T0CKI is shared with pin GP2 on the PIC10F220/222.

¤ 2005-2013 Microchip Technology Inc.

7.0 ANALOG-TO-DIGITAL (A/D) CONVERTER

The A/D converter allows conversion of an analog signal into an 8-bit digital signal.

7.1 Clock Divisors

The A/D Converter has a single clock source setting, INTOSC/4. The A/D Converter requires 13 TAD periods to complete a conversion. The divisor values do not affect the number of TAD periods required to perform a conversion. The divisor values determine the length of the TAD period.

Note:	Due to the fixed clock divisor, a conversion will complete in 13 CPU instruction cycles.

7.2 Voltage Reference

Due to the nature of the design, there is no external voltage reference allowed for the A/D Converter. The A/D Converter reference voltage will always be VDD.

7.3 Analog Mode Selection

The ANS<1:0> bits are used to configure pins for analog input. Upon any Reset ANS<1:0> defaults to 11. This configures pins AN0 and AN1 as analog inputs. Pins configured as analog inputs are not available for digital output. Users should not change the ANS bits while a conversion is in process. ANS bits are active regardless of the condition of ADON.

7.4 A/D Converter Channel Selection

The CHS bits are used to select the analog channel to be sampled by the A/D Converter. The CHS bits should not be changed during a conversion. To acquire an analog signal, the CHS selection must match one of the pin(s) selected by the ANS bits. The Internal Absolute Voltage Reference can be selected regardless of the condition of the ANS bits. All channel selection information will be lost when the device enters Sleep.

Note:	The A/D Converter module consumes power when the ADON bit is set even when no channels are selected as analog inputs. For low-power applications, it is recommended that the ADON bit be cleared when the A/D Converter is not in use.

7.5 The GO/$\overline{\text{DONE}}$ bit

The GO/$\overline{\text{DONE}}$ bit is used to determine the status of a conversion, to start a conversion and to manually halt a conversion in process. Setting the GO/$\overline{\text{DONE}}$ bit starts a conversion. When the conversion is complete, the A/D Converter module clears the GO/$\overline{\text{DONE}}$ bit. A conversion can be terminated by manually clearing the GO/$\overline{\text{DONE}}$ bit while a conversion is in process. Manual termination of a conversion may result in a partially converted result in ADRES.

The GO/$\overline{\text{DONE}}$ bit is cleared when the device enters Sleep, stopping the current conversion. The A/D Converter does not have a dedicated oscillator, it runs off of the system clock.

The GO/$\overline{\text{DONE}}$ bit cannot be set when ADON is clear.

7.6 Sleep

This A/D Converter does not have a dedicated A/D Converter clock and therefore no conversion in Sleep is possible. If a conversion is underway and a Sleep command is executed, the GO/$\overline{\text{DONE}}$ and ADON bit will be cleared. This will stop any conversion in process and power-down the A/D Converter module to conserve power. Due to the nature of the conversion process, the ADRES may contain a partial conversion. At least 1 bit must have been converted prior to Sleep to have partial conversion data in ADRES. The CHS bits are reset to their default condition and CHS<1:0> = 11.

For accurate conversions, TAD must meet the following:

- 500 ns < TAD < 50 µs
- TAD = 1/(FOSC/divisor)

TABLE 7-1: EFFECTS OF SLEEP AND WAKE ON ADCON0

	ANS1	ANS0	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
Prior to Sleep	x	x	x	x	0	0
Prior to Sleep	x	x	x	x	1	1
Entering Sleep	Unchanged	Unchanged	1	1	0	0
Wake	1	1	1	1	0	0

7.7 Analog Conversion Result Register

The ADRES register contains the results of the last conversion. These results are present during the sampling period of the next analog conversion process. After the sampling period is over, ADRES is cleared (= 0). A 'leading one' is then right shifted into the ADRES to serve as an internal conversion complete bit. As each bit weight, starting with the MSb, is converted, the leading one is shifted right and the converted bit is stuffed into ADRES. After a total of 9 right shifts of the 'leading one' have taken place, the conversion is complete; the 'leading one' has been shifted out and the GO/$\overline{\text{DONE}}$ bit is cleared.

If the GO/$\overline{\text{DONE}}$ bit is cleared in software during a conversion, the conversion stops. The data in ADRES is the partial conversion result. This data is valid for the bit weights that have been converted. The position of the 'leading one' determines the number of bits that have been converted. The bits that were not converted before the GO/$\overline{\text{DONE}}$ was cleared are unrecoverable.

7.8 Internal Absolute Voltage Reference

The function of the Internal Absolute Voltage Reference is to provide a constant voltage for conversion across the devices V_{DD} supply range. The A/D Converter is ratiometric with the conversion reference voltage being V_{DD}. Converting a constant voltage of 0.6V (typical) will result in a result based on the voltage applied to V_{DD} of the device. The result of conversion of this reference across the V_{DD} range can be approximated by: Conversion Result = 0.6V/(V_{DD}/256)

Note:	The actual value of the Absolute Voltage Reference varies with temperature and part-to-part variation. The conversion is also susceptible to analog noise on the V_{DD} pin and noise generated by the sinking or sourcing of current on the I/O pins.

REGISTER 7-1: ADCON0: A/D CONVERTER 0 REGISTER

R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1	R/W-0	R/W-0
ANS1	ANS0	—	—	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ANS1:** ADC Analog Input Pin Select bit
1 = GP1/AN1 configured for analog input
0 = GP1/AN1 configured as digital I/O

bit 6 **ANS0:** ADC Analog Input Pin Select bit(1), (2)
1 = GP0/AN0 configured as an analog input
0 = GP0/AN0 configured as digital I/O

bit 5-4 **Unimplemented:** Read as '0'

bit 3-2 **CHS<1:0>:** ADC Channel Select bits(3)
00 = Channel 00 (GP0/AN0)
01 = Channel 01 (GP1/AN1)
1X = 0.6V absolute Voltage reference

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit(4)
1 = ADC conversion in progress. Setting this bit starts an ADC conversion cycle. This bit is automatically cleared by hardware when the ADC is done converting.
0 = ADC conversion completed/not in progress. Manually clearing this bit while a conversion is in process terminates the current conversion.

bit 0 **ADON:** ADC Enable bit
1 = ADC module is operating
0 = ADC module is shut-off and consumes no power

Note 1: When the ANS bits are set, the channel(s) selected are automatically forced into analog mode regardless of the pin function previously defined.
2: The ANS<1:0> bits are active regardless of the condition of ADON
3: CHS<1:0> bits default to 11 after any Reset.
4: If the ADON bit is clear, the GO/$\overline{\text{DONE}}$ bit cannot be set.

¤ 2005-2013 Microchip Technology Inc.

8.5 Device Reset Timer (DRT)

On the PIC10F220/222 devices, the DRT runs any time the device is powered up.

The DRT operates on an internal oscillator. The processor is kept in Reset as long as the DRT is active. The DRT delay allows VDD to rise above VDD min. and for the oscillator to stabilize.

The on-chip DRT keeps the devices in a Reset condition for approximately 1.125 ms after $\overline{MCLR}$ has reached a logic high (VIH $\overline{MCLR}$) level. Programming GP3/$\overline{MCLR}$/VPP as $\overline{MCLR}$ and using an external RC network connected to the $\overline{MCLR}$ input is not required in most cases. This allows savings in cost-sensitive and/or space restricted applications, as well as allowing the use of the GP3/$\overline{MCLR}$/VPP pin as a general purpose input.

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

Reset sources are POR, $\overline{MCLR}$, WDT time-out and wake-up on pin change. See **Section 8.9.2 "Wake-up from Sleep", Notes 1, 2 and 3**.

TABLE 8-3: DRT (DEVICE RESET TIMER PERIOD)

POR Reset	Subsequent Resets
1.125 ms (typical)	10 µs (typical)

8.6 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is a free running on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the internal 4/8 MHz oscillator. This means that the WDT will run even if the main processor clock has been stopped, for example, by execution of a `SLEEP` instruction. During normal operation or Sleep, a WDT Reset or wake-up Reset, generates a device Reset.

The $\overline{TO}$ bit (STATUS<4>) will be cleared upon a Watchdog Timer Reset.

The WDT can be permanently disabled by programming the configuration WDTE as a '0' (see **Section 8.1 "Configuration Bits"**). Refer to the PIC10F220/222 Programming Specification to determine how to access the Configuration Word.

8.6.1 WDT PERIOD

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

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

8.6.2 WDT PROGRAMMING CONSIDERATIONS

The `CLRWDT` instruction clears the WDT and the postscaler, if assigned to the WDT, and prevents it from timing out and generating a device Reset.

The `SLEEP` instruction resets the WDT and the postscaler, if assigned to the WDT. This gives the maximum Sleep time before a WDT wake-up Reset.

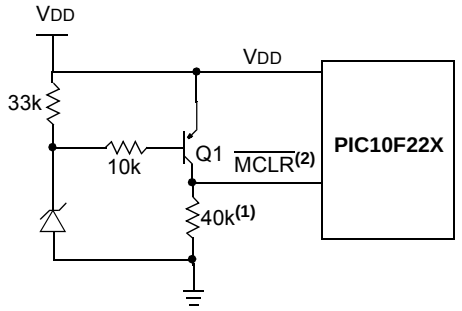
¤ 2005-2013 Microchip Technology Inc.

8.8 Reset on Brown-out

A Brown-out is a condition where device power (VDD) dips below its minimum value, but not to zero, and then recovers. The device should be reset in the event of a Brown-out.

To reset PIC10F220/222 devices when a Brown-out occurs, external Brown-out protection circuits may be built, as shown in Figure 8-7 and Figure 8-8.

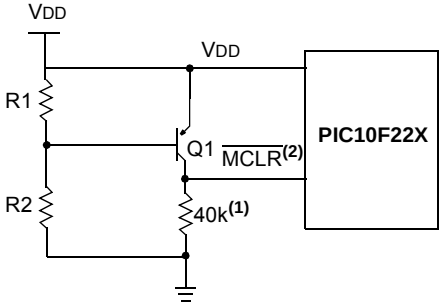
FIGURE 8-7: BROWN-OUT PROTECTION CIRCUIT 1



Note 1: This circuit will activate Reset when VDD goes below Vz + 0.7V (where Vz = Zener voltage).

2: Pin must be configured as $\overline{MCLR}$.

FIGURE 8-8: BROWN-OUT PROTECTION CIRCUIT 2

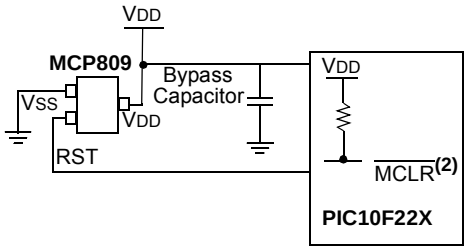


Note 1: This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

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

2: Pin must be configured as $\overline{MCLR}$.

FIGURE 8-9: BROWN-OUT PROTECTION CIRCUIT 3



Note 1: This Brown-out Protection circuit employs Microchip Technology's MCP809 micro-controller supervisor. There are 7 different trip point selections to accommodate 5V to 3V systems.

2: Pin must be configured as $\overline{MCLR}$.

8.9 Power-down Mode (Sleep)

A device may be powered down (Sleep) and later powered up (wake-up from Sleep).

8.9.1 SLEEP

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

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

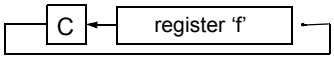
Note: A Reset generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low.

For lowest current consumption while powered down, the T0CKI input should be at VDD or VSS and the GP3/$\overline{MCLR}$/VPP pin must be at a logic high level if $\overline{MCLR}$ is enabled.

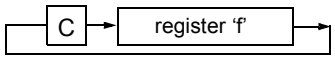
¤ 2005-2013 Microchip Technology Inc.

RETLW	Return with Literal in W
Syntax:	[label] RETLW k
Operands:	0 dk d255
Operation:	k o (W); TOS o PC
Status Affected:	None
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.

RLF	Rotate Left f through Carry
Syntax:	[label] RLF f,d
Operands:	0 df d31 d • [0,1]
Operation:	See description below
Status Affected:	C
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’.



RRF	Rotate Right f through Carry
Syntax:	[label] RRF f,d
Operands:	0 df d31 d • [0,1]
Operation:	See description below
Status Affected:	C
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’.



SLEEP	Enter SLEEP Mode
Syntax:	[label] SLEEP
Operands:	None
Operation:	00h o WDT; 0 o WDT prescaler; 1 o $\overline{TO}$; 0 o $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$, RBWUF
Description:	Time-out Status bit ($\overline{TO}$) is set. The Power-down Status bit ($\overline{PD}$) is cleared. RBWUF is unaffected. The WDT and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped. See section on Sleep for more details.

SUBWF	Subtract W from f
Syntax:	[label] SUBWF f,d
Operands:	0 df d31 d • [0,1]
Operation:	(f) – (W) o dest)
Status Affected:	C, DC, Z
Description:	Subtract (2’s complement method) the W register from register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

SWAPF	Swap Nibbles in f
Syntax:	[label] SWAPF f,d
Operands:	0 df d31 d • [0,1]
Operation:	(f<3:0>) o (dest<7:4>); (f<7:4>) o (dest<3:0>)
Status Affected:	None
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’.

10.2 DC Characteristics: PIC10F220/222 (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40×C £ TA £ +125×C (extended)				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0		5.5	V	See **Figure 10-1**
D002	VDR	**RAM Data Retention Voltage(2)**	1.5*		—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	
D010	IDD	**Supply Current(3)**					
			— — — —	175 0.625 250 0.800	275 1.1 400 1.5	PA mA PA mA	VDD = 2.0V, Fosc = 4 MHz VDD = 5.0V, Fosc = 4 MHz VDD = 2.0V, Fosc = 8 MHz VDD = 5.0V, Fosc = 8 MHz
D020	IPD	**Power-down Current(4)**					
			— —	0.1 1	9 15	PA PA	VDD = 2.0V VDD = 5.0V
D022	IWDT	**WDT Current(4)**					
			— —	1.0 7	18 22	PA PA	VDD = 2.0V VDD = 5.0V

* These parameters are characterized but not tested.

Note 1: Data in the Typical ("Typ") column is based on characterization results at 25 ¢C. This data is for design guidance only and is not tested.

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

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

a) The test conditions for all IDD measurements in active operation mode are:
All I/O pins tri-stated, pulled to VSS, T0CKI = VDD, $\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

b) For standby current measurements, the conditions are the same, except that the device is in Sleep mode.

4: Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS. The peripheral current is the sum of the base IPD and the additional current consumed when the peripheral is enabled.

¤ 2005-2013 Microchip Technology Inc.

10.3 DC Characteristics: PIC10F220/222 (Industrial, Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating temperature -40°C dTA d+85°C (industrial) -40°C dTA d+125°C (extended) Operating voltage VDD range as described in DC specification				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports:					
D030		with TTL buffer	Vss	—	0.8	V	For all 4.5 dVDD d5.5V
D030A			Vss	—	0.15 VDD	V	Otherwise
D031		with Schmitt Trigger buffer	Vss	—	0.2 VDD	V	
D032		$\overline{MCLR}$, T0CKI	Vss	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:		—			
D040		with TTL buffer	2.0	—	VDD	V	4.5 dVDD d5.5V
D040A			0.25 VDD + 0.8	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.8VDD	—	VDD	V	For entire VDD range
D042		$\overline{MCLR}$, T0CKI	0.8VDD	—	VDD	V	
D070	IPUR	**GPIO weak pull-up current**	50	250	400	PA	VDD = 5V, VPIN = VSS
	IIL	**Input Leakage Current(1)**					
D060		I/O ports	—	±0.1	± 1	PA	Vss dVPIN dVDD, Pin at high-impedance
D061		GP3/$\overline{MCLR}$(2)	—	±0.7	± 5	PA	Vss dVPIN dVDD
		Output Low Voltage					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40 C to +85 C
D080A			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, -40 C to +125 C
		Output High Voltage					
D090		I/O ports(2)	VDD – 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40 C to +85 C
D090A			VDD – 0.7	—	—	V	IOH = -2.5 mA, VDD = 4.5V, -40 C to +125 C
		Capacitive Loading Specs on Output Pins					
D101		All I/O pins	—	—	50*	pF	

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

* These parameters are for design guidance only and are not tested.

Note 1: Negative current is defined as coming out of the pin.

2: This specification applies when GP3/$\overline{MCLR}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{MCLR}$ circuit is higher than the standard I/O logic.

NOTES:

¤ 2005-2013 Microchip Technology Inc.

12.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

12.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

12.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

12.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

¤ 2005-2013 Microchip Technology Inc.

12.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

12.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

12.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

12.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

12.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

TABLE 13-1: 8-LEAD 2x3 DFN (MC) TOP MARKING

Part Number	Marking
PIC10F220-I/MC	BJ0
PIC10F220-E/MC	BK0
PIC10F222-I/MC	BL0
PIC10F222-E/MC	BM0

TABLE 13-2: 6-LEAD SOT-23 (OT) PACKAGE TOP MARKING

Part Number	Marking
PIC10F220-I/OT	20NN
PIC10F220-E/OT	A0NN
PIC10F222-I/OT	22NN
PIC10F222-E/OT	A2NN

Note: NN represents the alphanumeric traceability code.

¤ 2005-2013 Microchip Technology Inc.

INDEX

A

A/D
- Specifications ... 60

ADC
- Internal Sampling Switch (RSS) IMPEDANCE ... 32
- Source Impedance ... 32

ALU ... 9

Assembler
- MPASM Assembler ... 62

B

Block Diagram
- On-Chip Reset Circuit ... 36
- Timer0 ... 25
- TMR0/WDT Prescaler ... 28
- Watchdog Timer ... 39

Block Diagrams
- Analog Input Model ... 32

Brown-Out Protection Circuit ... 40

C

C Compilers
- MPLAB C18 ... 62
- MPLAB C30 ... 62

Carry ... 9

Clocking Scheme ... 11

Code Protection ... 33, 41

Configuration Bits ... 33

Customer Change Notification Service ... 75

Customer Notification Service ... 75

Customer Support ... 75

D

DC and AC Characteristics ... 65

Development Support ... 61

Digit Carry ... 9

E

Errata ... 3

F

Family of Devices
- PIC10F22X ... 5

FSR ... 20

G

GPIO ... 21

I

I/O Interfacing ... 21

I/O Ports ... 21

I/O Programming Considerations ... 23

ID Locations ... 33, 41

INDF ... 20

Indirect Data Addressing ... 20

Instruction Cycle ... 11

Instruction Flow/Pipelining ... 11

Instruction Set Summary ... 44

Internal Sampling Switch (RSS) IMPEDANCE ... 32

Internet Address ... 75

L

Loading of PC ... 19

M

Memory Organization ... 13
- Data Memory ... 14
- Program Memory (PIC10F220) ... 13
- Program Memory (PIC10F222) ... 13

Microchip Internet Web Site ... 75

MPLAB ASM30 Assembler, Linker, Librarian ... 62

MPLAB ICD 2 In-Circuit Debugger ... 63

MPLAB ICE 2000 High-Performance Universal
- In-Circuit Emulator ... 63

MPLAB Integrated Development Environment Software ... 61

MPLAB PM3 Device Programmer ... 63

MPLAB REAL ICE In-Circuit Emulator System ... 63

MPLINK Object Linker/MPLIB Object Librarian ... 62

O

OPTION Register ... 17

OSCCAL Register ... 18

Oscillator Configurations ... 34

Oscillator Types
- HS ... 34
- LP ... 34

P

PIC10F220/222 Device Varieties ... 7

PICSTART Plus Development Programmer ... 64

POR
- Device Reset Timer (DRT) ... 33, 38
- $\overline{PD}$ ... 39
- Power-on Reset (POR) ... 33
- $\overline{TO}$ ... 39

Power-down Mode ... 40

Prescaler ... 27

Program Counter ... 19

Q

Q cycles ... 11

R

Reader Response ... 76

Read-Modify-Write ... 23

Register File Map
- PIC10F220 ... 14
- PIC10F222 ... 14

Registers
- Special Function ... 14

Reset ... 33

Reset on Brown-Out ... 40

S

Sleep ... 33, 40

Software Simulator (MPLAB SIM) ... 62

Special Features of the CPU ... 33

Special Function Registers ... 14

Stack ... 19

STATUS Register ... 9, 15, 29

T

Timer0
- Timer0 ... 25
- Timer0 (TMR0) Module ... 25
- TMR0 with External Clock ... 26

Timing Parameter Symbology and Load Conditions ... 57

TRIS Registers ... 21