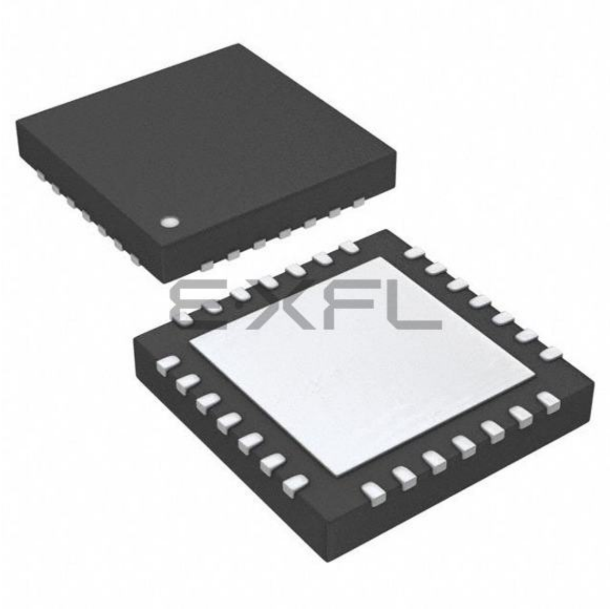
Microchip Technology - PIC18F1220T-I/ML Datasheet



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	40MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f1220t-i-ml

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

3.0 POWER MANAGED MODES

The PIC18F1220/1320 devices offer a total of six operating modes for more efficient power management (see Table 3-1). These provide a variety of options for selective power conservation in applications where resources may be limited (i.e., battery powered devices).

There are three categories of power managed modes:

- Sleep mode
- Idle modes
- Run modes

These categories define which portions of the device are clocked and sometimes, what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or INTOSC multiplexer); the Sleep mode does not use a clock source.

The clock switching feature offered in other PIC18 devices (i.e., using the Timer1 oscillator in place of the primary oscillator) and the Sleep mode offered by all PIC® devices (where all system clocks are stopped) are both offered in the PIC18F1220/1320 devices (SEC_RUN and Sleep modes, respectively). However, additional power managed modes are available that allow the user greater flexibility in determining what portions of the device are operating. The power managed modes are event driven; that is, some specific event must occur for the device to enter or (more particularly) exit these operating modes.

For PIC18F1220/1320 devices, the power managed modes are invoked by using the existing SLEEP instruction. All modes exit to PRI_RUN mode when triggered by an interrupt, a Reset or a WDT time-out (PRI_RUN mode is the normal full-power execution mode; the CPU and peripherals are clocked by the primary oscillator source). In addition, power managed Run modes may also exit to Sleep mode, or their corresponding Idle mode.

3.1 Selecting Power Managed Modes

Selecting a power managed mode requires deciding if the CPU is to be clocked or not and selecting a clock source. The IDLEN bit controls CPU clocking, while the SCS1:SCS0 bits select a clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

3.1.1 CLOCK SOURCES

The clock source is selected by setting the SCS bits of the OSCCON register (Register 2-2). Three clock sources are available for use in power managed Idle modes: the primary clock (as configured in Configuration Register 1H), the secondary clock (Timer1 oscillator) and the internal oscillator block. The secondary and internal oscillator block sources are available for the power managed modes (PRI_RUN mode is the normal full-power execution mode; the CPU and peripherals are clocked by the primary oscillator source).

TABLE 3-1: POWER MANAGED MODES

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN <7>	SCS1:SCS0 <1:0>	CPU	Peripherals	
Sleep	0	00	Off	Off	None – All clocks are disabled
PRI_RUN	0	00	Clocked	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC, INTRC[(1)] This is the normal full-power execution mode.
SEC_RUN	0	01	Clocked	Clocked	Secondary – Timer1 Oscillator
RC_RUN	0	1x	Clocked	Clocked	Internal Oscillator Block[(1)]
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, HSPLL, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – Timer1 Oscillator
RC_IDLE	1	1x	Off	Clocked	Internal Oscillator Block[(1)]

Note 1: Includes INTOSC and INTOSC postscaler, as well as the INTRC source.

4.0 RESET

The PIC18F1220/1320 devices differentiate between various kinds of Reset:

a) Power-on Reset (POR)

b) $\overline{\text{MCLR}}$ Reset during normal operation

c) $\overline{\text{MCLR}}$ Reset during Sleep

d) Watchdog Timer (WDT) Reset (during execution)

e) Programmable Brown-out Reset (BOR)

f) `RESET` Instruction

g) Stack Full Reset

h) Stack Underflow Reset

Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a “Reset state”, depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register (Register 5-2), $\overline{\text{RI}}$, $\overline{\text{TO}}$, $\overline{\text{PD}}$, $\overline{\text{POR}}$ and $\overline{\text{BOR}}$, are set or cleared differently in different Reset situations, as indicated in Table 4-2. These bits are used in software to determine the nature of the Reset. See Table 4-3 for a full description of the Reset states of all registers.

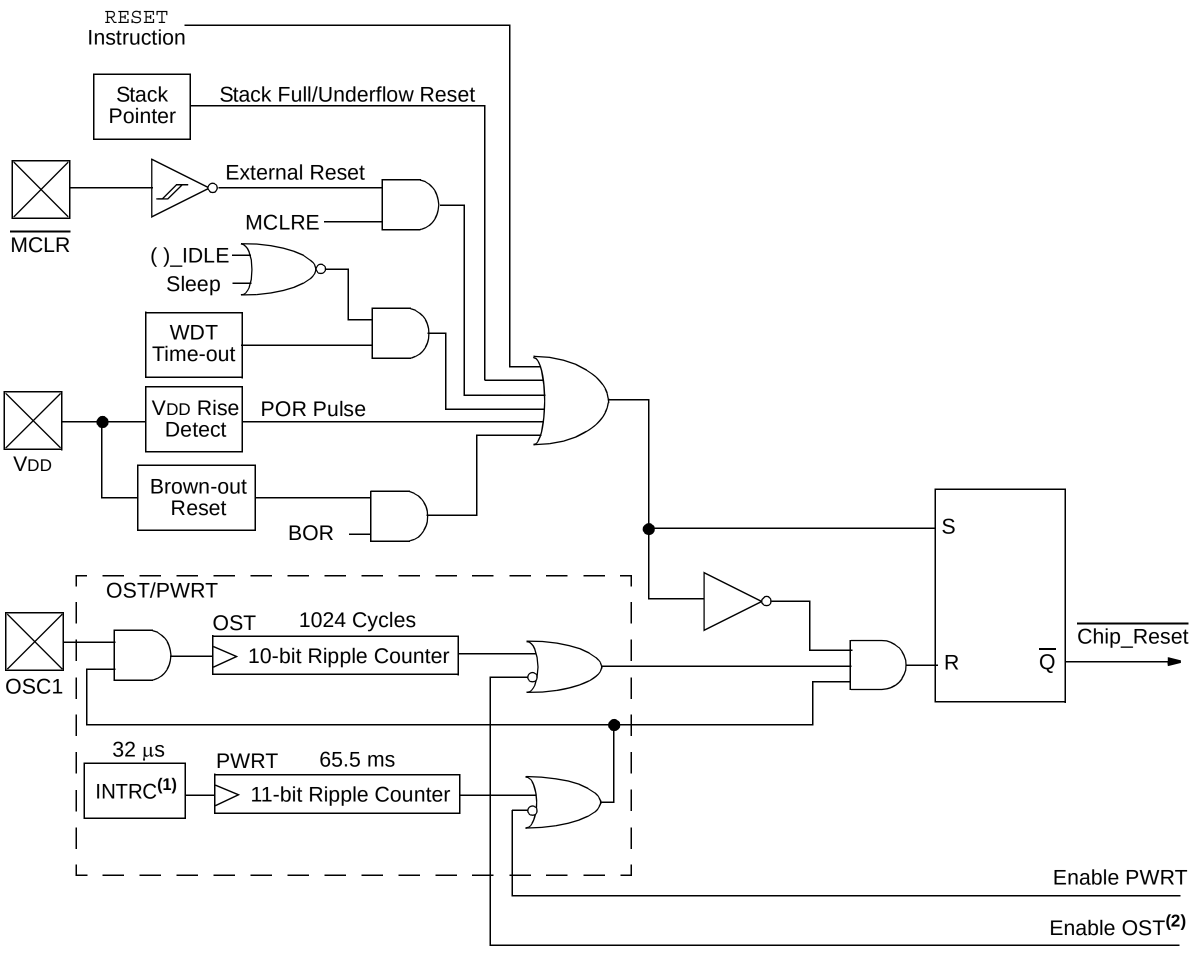
A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 4-1.

The Enhanced MCU devices have a $\overline{\text{MCLR}}$ noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

The $\overline{\text{MCLR}}$ pin is not driven low by any internal Resets, including the WDT.

The $\overline{\text{MCLR}}$ input provided by the $\overline{\text{MCLR}}$ pin can be disabled with the MCLRE bit in Configuration Register 3H (CONFIG3H<7>).

FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



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

2: See Table 4-1 for time-out situations.

5.8 Look-up Tables

Look-up tables are implemented two ways:

- Computed GOTO
- Table Reads

5.8.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (see Example 5-4).

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW 0xnn instructions. WREG is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW 0xnn instructions, that returns the value 0xnn to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSB = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 5-4: COMPUTED GOTO USING AN OFFSET VALUE

```
        MOVFW   OFFSET
        CALL    TABLE
ORG     0xnn00
TABLE   ADDWF   PCL
        RETLW   0xnn
        RETLW   0xnn
        RETLW   0xnn
        .
        .
        .
```

5.8.2 TABLE READS/TABLE WRITES

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to/from program memory, one byte at a time.

The table read/table write operation is discussed further in **Section 6.1 "Table Reads and Table Writes"**.

5.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. Figure 5-6 shows the data memory organization for the PIC18F1220/1320 devices.

The data memory map is divided into as many as 16 banks that contain 256 bytes each. The lower four bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper four bits for the BSR are not implemented.

The data memory contains Special Function Registers (SFR) and General Purpose Registers (GPR). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratch pad operations in the user's application. The SFRs start at the last location of Bank 15 (FFFh) and extend towards F80h. Any remaining space beyond the SFRs in the Bank may be implemented as GPRs. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as '0's.

The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of a File Select Register (FSRn) and a corresponding Indirect File Operand (INDFn). Each FSR holds a 12-bit address value that can be used to access any location in the Data Memory map without banking. See **Section 5.12 "Indirect Addressing, INDF and FSR Registers"** for indirect addressing details.

The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction that moves a value from one register to another.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM. **Section 5.10 "Access Bank"** provides a detailed description of the Access RAM.

5.9.1 GENERAL PURPOSE REGISTER FILE

Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

Data RAM is available for use as GPR registers by all instructions. The second half of Bank 15 (F80h to FFFh) contains SFRs. All other banks of data memory contain GPRs, starting with Bank 0.

FIGURE 5-8: INDIRECT ADDRESSING OPERATION

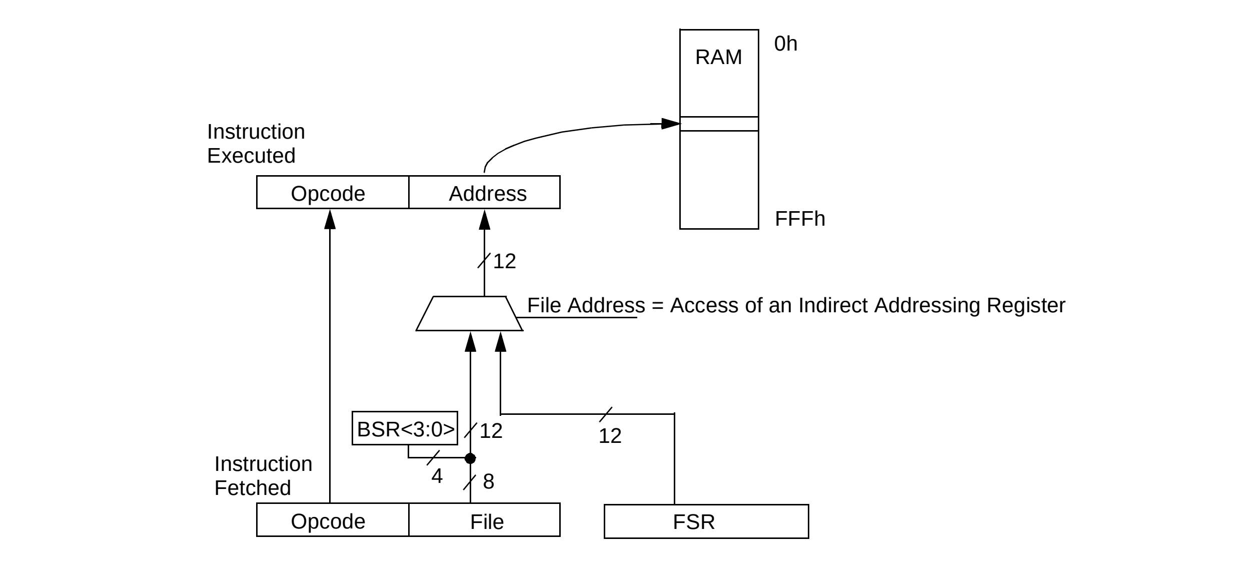
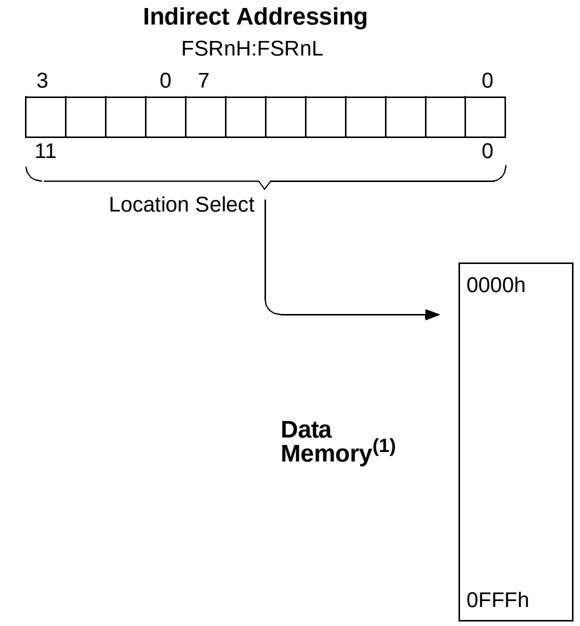


FIGURE 5-9: INDIRECT ADDRESSING



Note 1: For register file map detail, see Table 5-1.

EXAMPLE 6-3: WRITING TO FLASH PROGRAM MEMORY

```
           MOVLW   D'64                      ; number of bytes in erase block
           MOVWF   COUNTER
           MOVLW   BUFFER_ADDR_HIGH          ; point to buffer
           MOVWF   FSR0H
           MOVLW   BUFFER_ADDR_LOW
           MOVWF   FSR0L
           MOVLW   CODE_ADDR_UPPER           ; Load TBLPTR with the base
           MOVWF   TBLPTRU                   ; address of the memory block
           MOVLW   CODE_ADDR_HIGH
           MOVWF   TBLPTRH
           MOVLW   CODE_ADDR_LOW             ; 6 LSB = 0
           MOVWF   TBLPTRL
READ_BLOCK
           TBLRD*+                           ; read into TABLAT, and inc
           MOVF    TABLAT, W                 ; get data
           MOVWF   POSTINC0                  ; store data and increment FSR0
           DECFSZ  COUNTER                   ; done?
           GOTO    READ_BLOCK                ; repeat
MODIFY_WORD
           MOVLW   DATA_ADDR_HIGH            ; point to buffer
           MOVWF   FSR0H
           MOVLW   DATA_ADDR_LOW
           MOVWF   FSR0L
           MOVLW   NEW_DATA_LOW              ; update buffer word and increment FSR0
           MOVWF   POSTINC0
           MOVLW   NEW_DATA_HIGH             ; update buffer word
           MOVWF   INDF0
ERASE_BLOCK
           MOVLW   CODE_ADDR_UPPER           ; load TBLPTR with the base
           MOVWF   TBLPTRU                   ; address of the memory block
           MOVLW   CODE_ADDR_HIGH
           MOVWF   TBLPTRH
           MOVLW   CODE_ADDR_LOW             ; 6 LSB = 0
           MOVWF   TBLPTRL
           BCF     EECON1, CFGS              ; point to PROG/EEPROM memory
           BSF     EECON1, EEPGD             ; point to FLASH program memory
           BSF     EECON1, WREN              ; enable write to memory
           BSF     EECON1, FREE              ; enable Row Erase operation
           BCF     INTCON, GIE               ; disable interrupts
           MOVLW   55h                       ; Required sequence
           MOVWF   EECON2                    ; write 55H
           MOVLW   AAh
           MOVWF   EECON2                    ; write AAH
           BSF     EECON1, WR                ; start erase (CPU stall)
           NOP
           BSF     INTCON, GIE               ; re-enable interrupts
WRITE_BUFFER_BACK
           MOVLW   8                         ; number of write buffer groups of 8 bytes
           MOVWF   COUNTER_HI
           MOVLW   BUFFER_ADDR_HIGH          ; point to buffer
           MOVWF   FSR0H
           MOVLW   BUFFER_ADDR_LOW
           MOVWF   FSR0L
PROGRAM_LOOP
           MOVLW   8                         ; number of bytes in holding register
           MOVWF   COUNTER
```

7.7 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if either of these mechanisms are enabled.

The microcontroller itself can both read and write to the internal data EEPROM, regardless of the state of the code-protect Configuration bit. Refer to **Section 19.0 "Special Features of the CPU"** for additional information.

7.8 Using the Data EEPROM

The data EEPROM is a high endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). Frequently changing values will typically be updated more often than specification D124. If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

Note: If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification D124.

EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR               ; Start at address 0
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   AAh                 ;
        MOVWF   EECON2              ; Write AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     Loop                ; Not zero, do it again

        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

TABLE 7-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
EEADR	EEPROM Address Register								0000 0000	0000 0000
EEDATA	EEPROM Data Register								0000 0000	0000 0000
EECON2	EEPROM Control Register 2 (not a physical register)								—	—
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	xx-0 x000	uu-0 u000
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	TMR3IP	—	1--1 -11-	1--1 -11-
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	TMR3IF	—	0--0 -00-	0--0 -00-
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	TMR3IE	—	0--0 -00-	0--0 -00-

Legend: x = unknown, u = unchanged, – = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

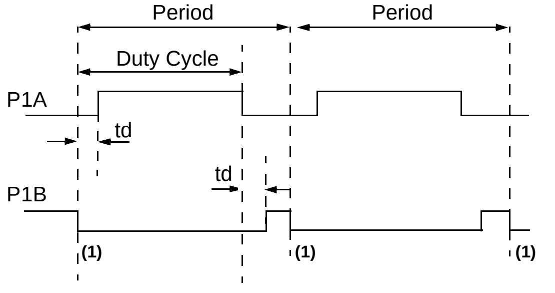
15.5.4 HALF-BRIDGE MODE

In the Half-Bridge Output mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the RB3/CCP1/P1A pin, while the complementary PWM output signal is output on the RB2/P1B/INT2 pin (Figure 15-6). This mode can be used for half-bridge applications, as shown in Figure 15-7, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge Output mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of bits, PDC6:PDC0 (PWM1CON<6:0>), sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 15.5.6 "Programmable Dead-Band Delay"** for more details of the dead-band delay operations.

The TRISB<3> and TRISB<2> bits must be cleared to configure P1A and P1B as outputs.

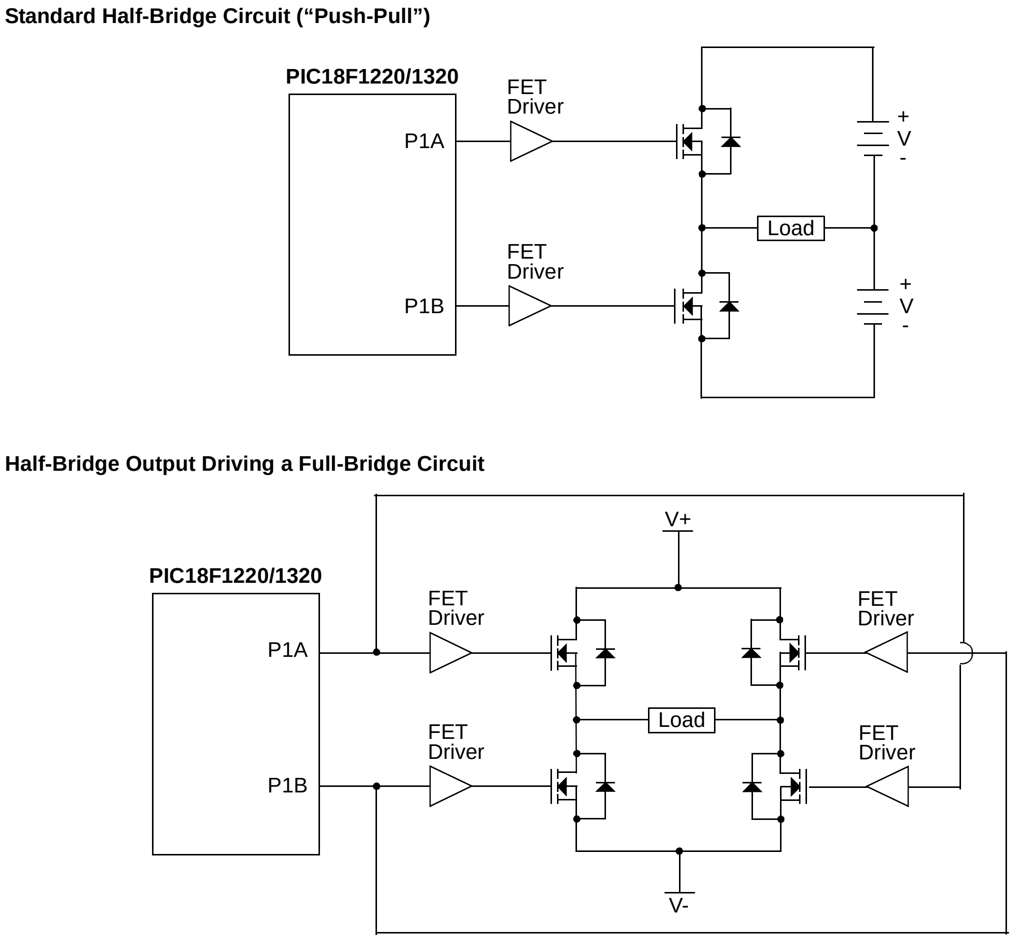
FIGURE 15-6: HALF-BRIDGE PWM OUTPUT (ACTIVE-HIGH)



td = Dead-Band Delay

Note 1: At this time, the TMR2 register is equal to the PR2 register.

FIGURE 15-7: EXAMPLES OF HALF-BRIDGE OUTPUT MODE APPLICATIONS



16.0 ENHANCED ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Addressable Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module can be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced Addressable USART module implements additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network (LIN) bus systems.

The EUSART can be configured in the following modes:

- Asynchronous (full duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The RB1/AN5/TX/CK/INT1 and RB4/AN6/RX/DT/KBI0 pins must be configured as follows for use with the Universal Synchronous Asynchronous Receiver Transmitter:

- SPEN (RCSTA<7>) bit must be set (= 1),
- PCFG6:PCFG5 (ADCON1<5:6>) must be set (= 1),
- TRISB<4> bit must be set (= 1) and
- TRISB<1> bit must be set (= 1).

Note: The EUSART control will automatically reconfigure the pin from input to output as needed.

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)

These are detailed in on the following pages in Register 16-1, Register 16-2 and Register 16-3, respectively.

16.1 Asynchronous Operation in Power Managed Modes

The EUSART may operate in Asynchronous mode while the peripheral clocks are being provided by the internal oscillator block. This makes it possible to remove the crystal or resonator that is commonly connected as the primary clock on the OSC1 and OSC2 pins.

The factory calibrates the internal oscillator block output (INTOSC) for 8 MHz (see Table 22-6). However, this frequency may drift as VDD or temperature changes and this directly affects the asynchronous baud rate. Two methods may be used to adjust the baud rate clock, but both require a reference clock source of some kind.

The first (preferred) method uses the OSCTUNE register to adjust the INTOSC output back to 8 MHz. Adjusting the value in the OSCTUNE register allows for fine resolution changes to the system clock source (see **Section 3.6 “INTOSC Frequency Drift”** for more information).

The other method adjusts the value in the Baud Rate Generator (BRG). There may not be fine enough resolution when adjusting the Baud Rate Generator to compensate for a gradual change in the peripheral clock frequency.

REGISTER 16-3: BAUDCTL: BAUD RATE CONTROL REGISTER

U-0	R-1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
—	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **RCIDL**: Receive Operation Idle Status bit

1 = Receiver is Idle
0 = Receiver is busy

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **SCKP**: Synchronous Clock Polarity Select bit

Asynchronous mode:
Unused in this mode.

Synchronous mode:
1 = Idle state for clock (CK) is a high level
0 = Idle state for clock (CK) is a low level

bit 3 **BRG16:** 16-bit Baud Rate Register Enable bit

1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG
0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **WUE:** Wake-up Enable bit

Asynchronous mode:
1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge
0 = RX pin not monitored or rising edge detected

Synchronous mode:
Unused in this mode.

bit 0 **ABDEN**: Auto-Baud Detect Enable bit

Asynchronous mode:
1 = Enable baud rate measurement on the next character – requires reception of a Sync byte (55h); cleared in hardware upon completion
0 = Baud rate measurement disabled or completed

Synchronous mode:
Unused in this mode.

19.4.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset, or by entering a power managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INTOSC multiplexer provides the system clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock system source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power managed mode is entered.

Entering a power managed mode by loading the OSCCON register and executing a SLEEP instruction will clear the Fail-Safe condition. When the Fail-Safe condition is cleared, the clock monitor will resume monitoring the peripheral clock.

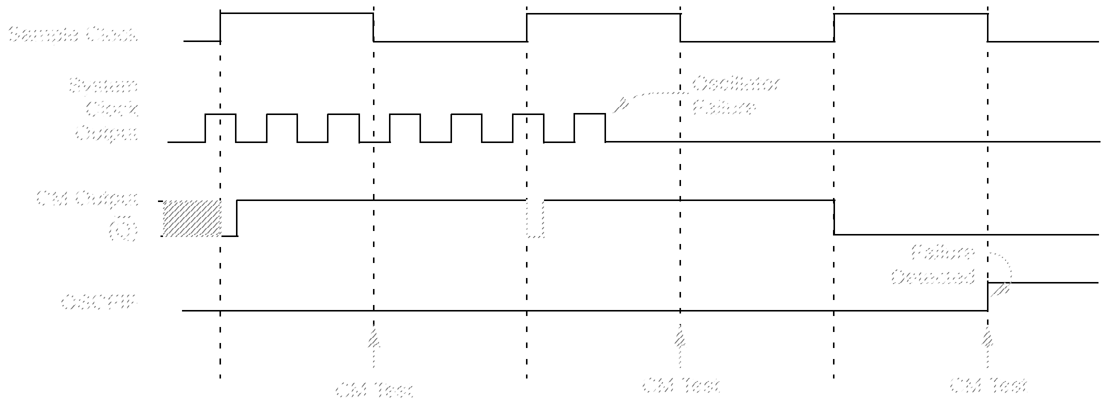
19.4.3 FSCM INTERRUPTS IN POWER MANAGED MODES

As previously mentioned, entering a power managed mode clears the Fail-Safe condition. By entering a power managed mode, the clock multiplexer selects the clock source selected by the OSCCON register. Fail-Safe monitoring of the power managed clock source resumes in the power managed mode.

If an oscillator failure occurs during power managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTOSC multiplexer. An automatic transition back to the failed clock source will not occur.

If the interrupt is disabled, the device will not exit the power managed mode on oscillator failure. Instead, the device will continue to operate as before, but clocked by the INTOSC multiplexer. While in Idle mode, subsequent interrupts will cause the CPU to begin executing instructions while being clocked by the INTOSC multiplexer. The device will not transition to a different clock source until the Fail-Safe condition is cleared.

FIGURE 19-4: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

19.5 Program Verification and Code Protection

The overall structure of the code protection on the PIC18 Flash devices differs significantly from other PIC devices.

The user program memory is divided into three blocks. One of these is a boot block of 512 bytes. The remainder of the memory is divided into two blocks on binary boundaries.

Each of the three blocks has three protection bits associated with them. They are:

- Code-Protect bit (CPn)
- Write-Protect bit (WRTn)
- External Block Table Read bit (EBTRn)

Figure 19-5 shows the program memory organization for 4 and 8-Kbyte devices and the specific code protection bit associated with each block. The actual locations of the bits are summarized in Table 19-3.

FIGURE 19-5: CODE-PROTECTED PROGRAM MEMORY FOR PIC18F1220/1320

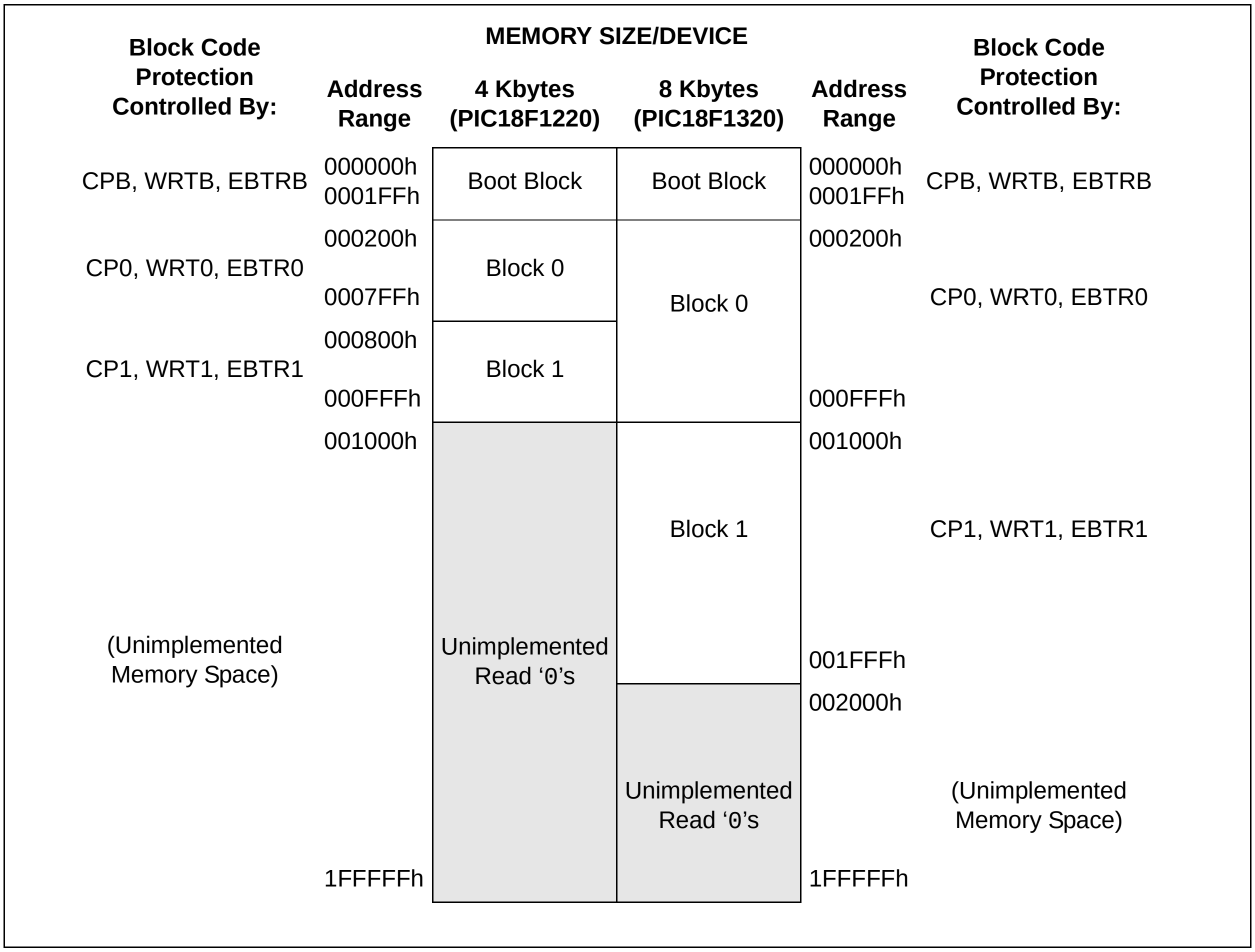


TABLE 19-3: SUMMARY OF CODE PROTECTION REGISTERS

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	—	—	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	—	—	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	—	—	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded cells are unimplemented.

POP	Pop Top of Return Stack
Syntax:	[*label*] POP
Operands:	None
Operation:	(TOS) → bit bucket
Status Affected:	None
Encoding:	0000 0000 0000 0110
Description:	The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Pop TOS value	No operation

Example:
```
POP
GOTO      NEW
```

Before Instruction

TOS	=	0x0031A2
Stack (1 level down)	=	0x014332

After Instruction

TOS	=	0x014332
PC	=	NEW

PUSH	Push Top of Return Stack
Syntax:	[*label*] PUSH
Operands:	None
Operation:	(PC + 2) → TOS
Status Affected:	None
Encoding:	0000 0000 0000 0101
Description:	The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Push PC + 2 onto return stack	No operation	No operation

Example:
```
PUSH
```

Before Instruction

TOS	=	0x00345A
PC	=	0x000124

After Instruction

PC	=	0x000126
TOS	=	0x000126
Stack (1 level down)	=	0x00345A

RETFIE Return from Interrupt

Syntax: [*label*] RETFIE [s]

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
1 → GIE/GIEH or PEIE/GIEL,
if s = `1`
(WS) → W,
(STATUSS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged.

Status Affected: GIE/GIEH, PEIE/GIEL.

Encoding:

`0000`	`0000`	`0001`	`000s`

Description: Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low priority Global Interrupt Enable bit. If ‘s’ = `1`, the contents of the shadow registers, WS, STATUSS and BSRS, are loaded into their corresponding registers, W, Status and BSR. If ‘s’ = `0`, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	Pop PC from stack Set GIEH or GIEL
No operation	No operation	No operation	No operation

Example: `RETFIE 1`

After Interrupt

PC	=	TOS
W	=	WS
BSR	=	BSRS
Status	=	STATUSS
GIE/GIEH, PEIE/GIEL	=	1

RETLW Return Literal to W

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W,
(TOS) → PC,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

`0000`	`1100`	`kkkk`	`kkkk`

Description: W is loaded with the 8-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Pop PC from stack, Write to W
No operation	No operation	No operation	No operation

Example:

```
    CALL TABLE ; W contains table
               ; offset value
               ; W now has
               ; table value
   :
TABLE
    ADDWF PCL  ; W = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
   :
   :
    RETLW kn   ; End of table
```

Before Instruction

W	=	0x07

After Instruction

W	=	value of kn

SUBLW	Subtract W from literal
Syntax:	[*label*] SUBLW k
Operands:	0 ≤ k ≤ 255
Operation:	k – (W) → W
Status Affected:	N, OV, C, DC, Z
Encoding:	`0000` `1000` `kkkk` `kkkk`
Description:	W is subtracted from the 8-bit literal ‘k’. The result is placed in W.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example 1: `SUBLW 0x02`

Before Instruction

W	=	1	
C	=	?	

After Instruction

W	=	1	
C	=	1	; result is positive
Z	=	0	
N	=	0	

Example 2: `SUBLW 0x02`

Before Instruction

W	=	2	
C	=	?	

After Instruction

W	=	0	
C	=	1	; result is zero
Z	=	1	
N	=	0	

Example 3: `SUBLW 0x02`

Before Instruction

W	=	3	
C	=	?	

After Instruction

W	=	FF	; (2’s complement)
C	=	0	; result is negative
Z	=	0	
N	=	1	

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f [,d [,a]]
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	(f) – (W) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	`0101` `11da` `ffff` `ffff`
Description:	Subtract W from register ‘f’ (2’s complement method). If ‘d’ is ‘`0`’, the result is stored in W. If ‘d’ is ‘`1`’, the result is stored back in register ‘f’ (default). If ‘a’ is ‘`0`’, the Access Bank will be selected, overriding the BSR value. If ‘a’ is ‘`1`’, then the bank will be selected as per the BSR value (default).
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example 1: `SUBWF REG`

Before Instruction

REG	=	3	
W	=	2	
C	=	?	

After Instruction

REG	=	1	
W	=	2	
C	=	1	; result is positive
Z	=	0	
N	=	0	

Example 2: `SUBWF REG, W`

Before Instruction

REG	=	2	
W	=	2	
C	=	?	

After Instruction

REG	=	2	
W	=	0	
C	=	1	; result is zero
Z	=	1	
N	=	0	

Example 3: `SUBWF REG`

Before Instruction

REG	=	0x01	
W	=	0x02	
C	=	?	

After Instruction

REG	=	0xFFh	;(2’s complement)
W	=	0x02	
C	=	0x00	;result is negative
Z	=	0x00	
N	=	0x01	

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial)

PIC18LF1220/1320 (Industrial)					**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial		
PIC18F1220/1320 (Industrial, Extended)					**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended		
Param No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Power-Down Current (IPD)[1]						
	PIC18LF1220/1320	0.1	0.5	μA	-40°C	VDD = 2.0V, (**Sleep** mode)	
		0.1	0.5	μA	+25°C		
		0.2	1.9	μA	+85°C		
	PIC18LF1220/1320	0.1	0.5	μA	-40°C	VDD = 3.0V, (**Sleep** mode)	
		0.1	0.5	μA	+ 25°C		
		0.3	1.9	μA	+85°C		
	All devices	0.1	2.0	μA	-40°C	VDD = 5.0V, (**Sleep** mode)	
		0.1	2.0	μA	+25°C		
		0.4	6.5	μA	+85°C		
	Extended devices	11.2	50	μA	+125°C		
	Supply Current (IDD)[2,3]						
	PIC18LF1220/1320	8	40	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (**RC_RUN** mode, Internal oscillator source)
		9	40	μA	+25°C		
		11	40	μA	+85°C		
	PIC18LF1220/1320	25	68	μA	-40°C	VDD = 3.0V	
		25	68	μA	+25°C		
		20	68	μA	+85°C		
	All devices	55	80	μA	-40°C	VDD = 5.0V	
		55	80	μA	+25°C		
		50	80	μA	+85°C		
	Extended devices	50	80	μA	+125°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

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

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

22.2 DC Characteristics: Power-Down and Supply Current PIC18F1220/1320 (Industrial) PIC18LF1220/1320 (Industrial) (Continued)

PIC18LF1220/1320 (Industrial)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial					
PIC18F1220/1320 (Industrial, Extended)		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for industrial -40°C ≤ T_A ≤ +125°C for extended					
Param No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (I_{DD})[2,3]						
	All devices	3.2	4.1	mA	-40°C	V_{DD} = 4.2 V	F_{OSC} = 40 MHz (**PRI_IDLE** mode, EC oscillator)
		3.2	4.1	mA	+25°C		
		3.3	4.1	mA	+85°C		
	All devices	4.0	5.1	mA	-40°C	V_{DD} = 5.0V	
		4.1	5.1	mA	+25°C		
		4.1	5.1	mA	+85°C		
	PIC18LF1220/1320	5.1	9	μA	-10°C	V_{DD} = 2.0V	F_{OSC} = 32 kHz[4] (**SEC_RUN** mode, Timer1 as clock)
		5.8	9	μA	+25°C		
		7.9	11	μA	+70°C		
	PIC18LF1220/1320	7.9	12	μA	-10°C	V_{DD} = 3.0V	
		8.9	12	μA	+25°C		
		10.5	14	μA	+70°C		
	All devices	12.5	20	μA	-10°C	V_{DD} = 5.0V	
		16.3	20	μA	+25°C		
		18.4	25	μA	+70°C		

Legend: Shading of rows is to assist in readability of the table.

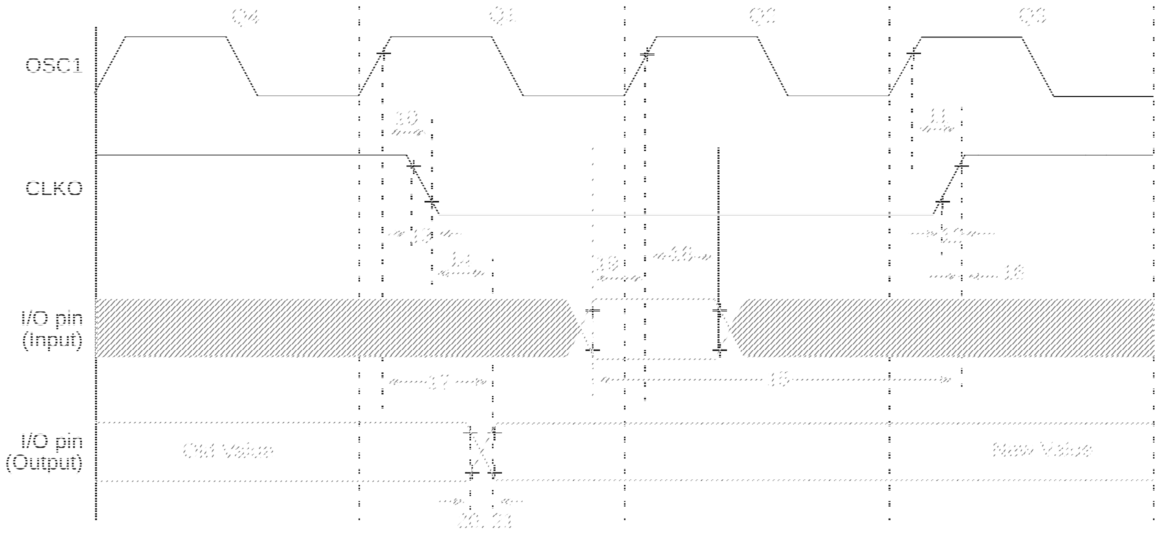
Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to V_{DD} or V_{SS} and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all I_{DD} measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD};
$\overline{\text{MCLR}}$ = V_{DD}; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be estimated by the formula Ir = $V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

FIGURE 22-7: CLKO AND I/O TIMING



Note: Refer to Figure 22-5 for load conditions.

TABLE 22-7: CLKO AND I/O TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Typ.	Max.	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKO↓		—	75	200	ns	**(Note 1)**
11	TosH2ckH	OSC1↑ to CLKO↑		—	75	200	ns	**(Note 1)**
12	TckR	CLKO Rise Time		—	35	100	ns	**(Note 1)**
13	TckF	CLKO Fall Time		—	35	100	ns	**(Note 1)**
14	TckL2ioV	CLKO↓ to Port Out Valid		—	—	0.5 TCY + 20	ns	**(Note 1)**
15	TioV2ckH	Port In Valid before CLKO↑		0.25 TCY + 25	—	—	ns	**(Note 1)**
16	TckH2ioI	Port In Hold after CLKO↑		0	—	—	ns	**(Note 1)**
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port Out Valid		—	50	150	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	PIC18F1X20	100	—	—	ns	
18A			PIC18LF1X20	200	—	—	ns	
19	TioV2osH	Port Input Valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20	TioR	Port Output Rise Time	PIC18F1X20	—	10	25	ns	
20A			PIC18LF1X20	—	—	60	ns	
21	TioF	Port Output Fall Time	PIC18F1X20	—	10	25	ns	
21A			PIC18LF1X20	—	—	60	ns	

Note 1: Measurements are taken in RC mode, where CLKO output is 4 x TOSC.

FIGURE 23-21: IPD SEC_RUN MODE, -10°C TO +70°C, 32.768 kHz XTAL, 2 x 22 pF, ALL PERIPHERALS DISABLED

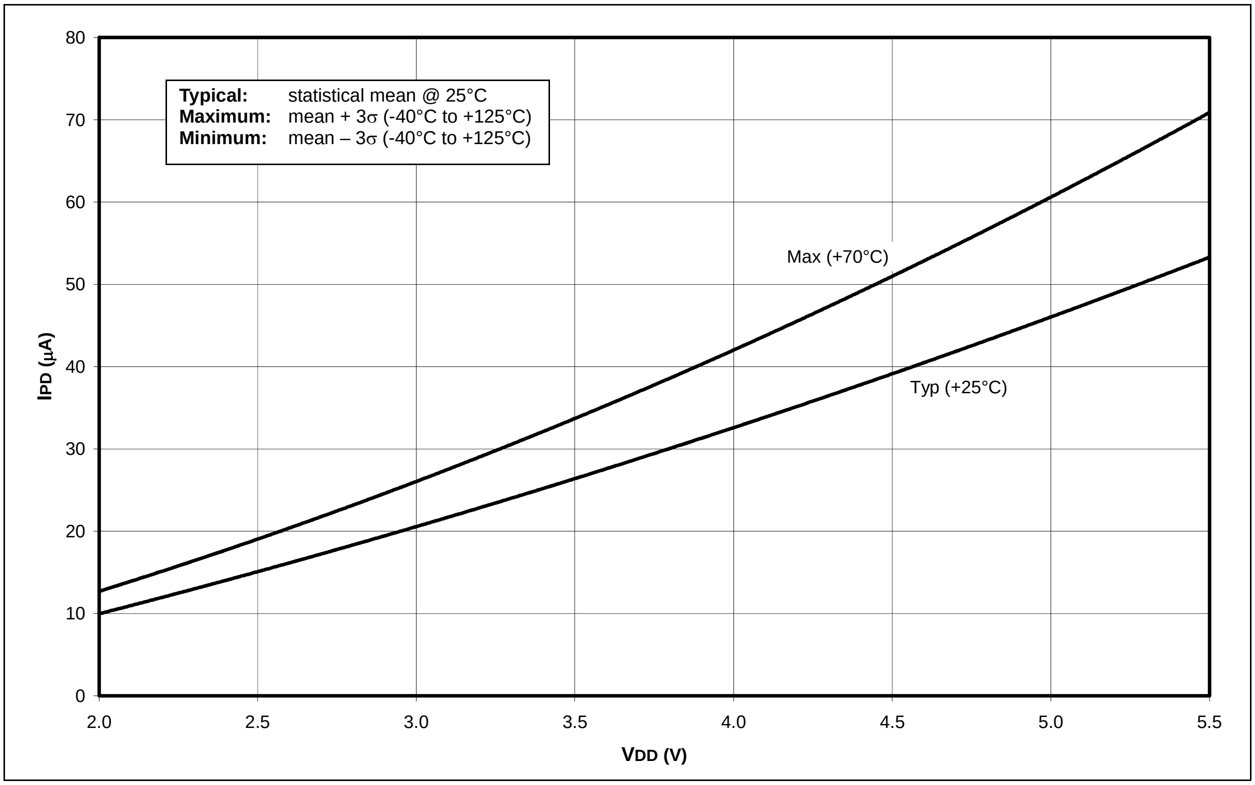
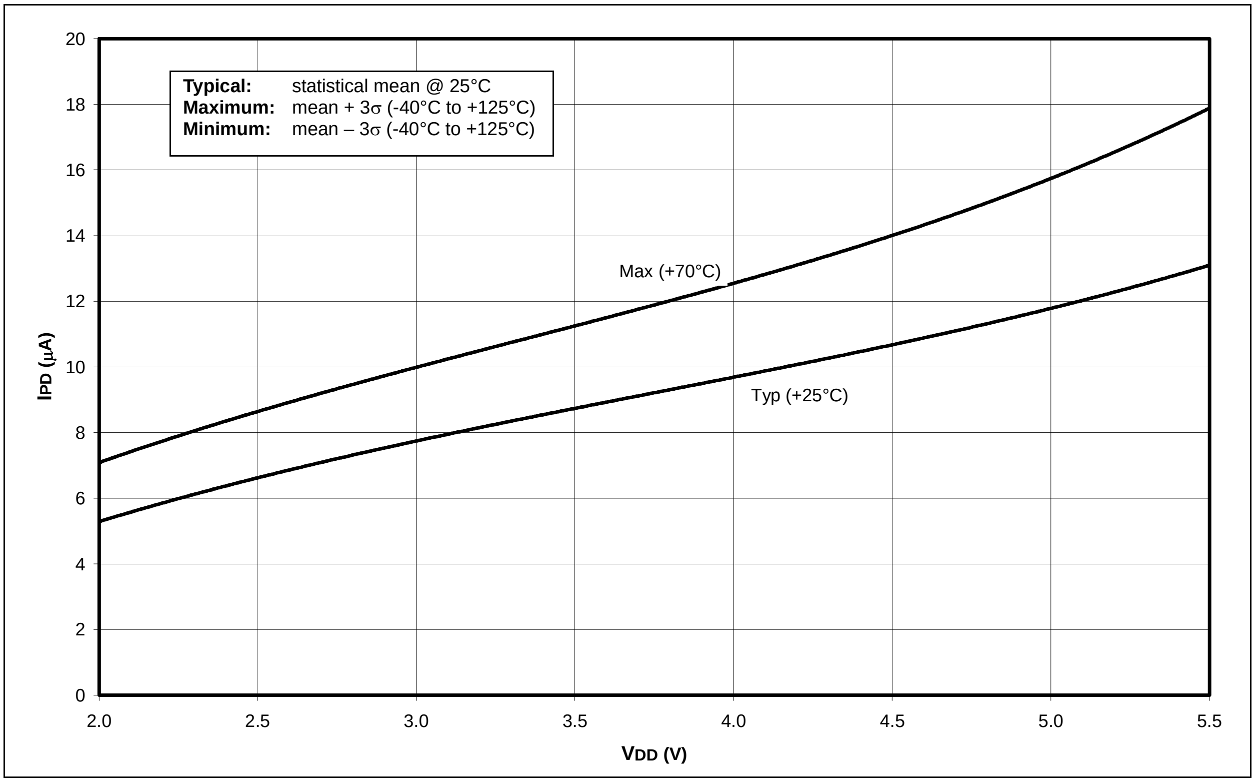
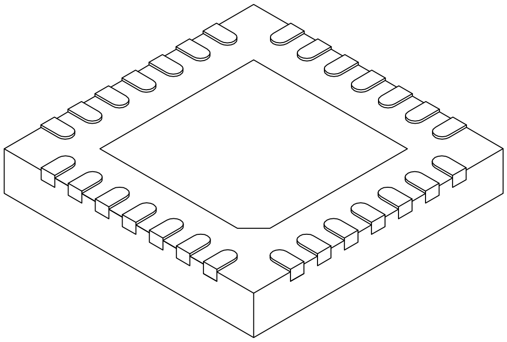


FIGURE 23-22: IPD SEC_IDLE MODE, -10°C TO +70°C, 32.768 kHz, 2 x 22 pF, ALL PERIPHERALS DISABLED



28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105C Sheet 2 of 2