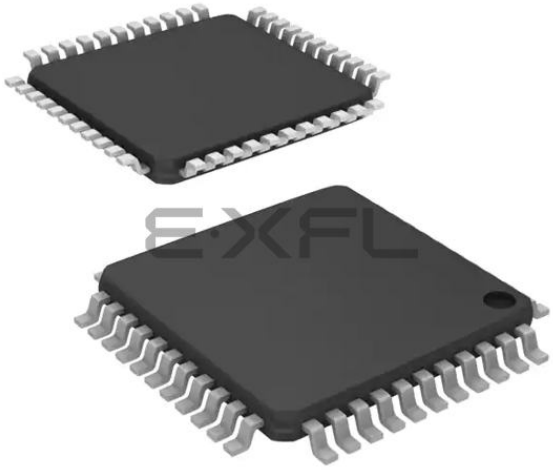


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4220t-i-pt

MICROCHIP PIC18F2220/2320/4220/4320

28/40/44-Pin High-Performance, Enhanced Flash MCUs with 10-Bit A/D and nanoWatt Technology

Low-Power Features:

- Power-Managed modes:
  - Run: CPU on, peripherals on
  - Idle: CPU off, peripherals on
  - Sleep: CPU off, peripherals off
- Power Consumption modes:
  - PRI_RUN: 150 μA, 1 MHz, 2V
  - PRI_IDLE: 37 μA, 1 MHz, 2V
  - SEC_RUN: 14 μA, 32 kHz, 2V
  - SEC_IDLE: 5.8 μA, 32 kHz, 2V
  - RC_RUN: 110 μA, 1 MHz, 2V
  - RC_IDLE: 52 μA, 1 MHz, 2V
  - Sleep: 0.1 μA, 1 MHz, 2V
- Timer1 Oscillator: 1.1 μA, 32 kHz, 2V
- Watchdog Timer: 2.1 μA
- Two-Speed Oscillator Start-up

Oscillators:

- Four Crystal modes:
  - LP, XT, HS: up to 25 MHz
  - HSPLL: 4-10 MHz (16-40 MHz internal)
- Two External RC modes, Up to 4 MHz
- Two External Clock modes, Up to 40 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz
  - 125 kHz-8 MHz calibrated to 1%
  - Two modes select one or two I/O pins
  - OSCTUNE – Allows user to shift frequency
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
  - Allows for safe shutdown if peripheral clock stops

Peripheral Highlights:

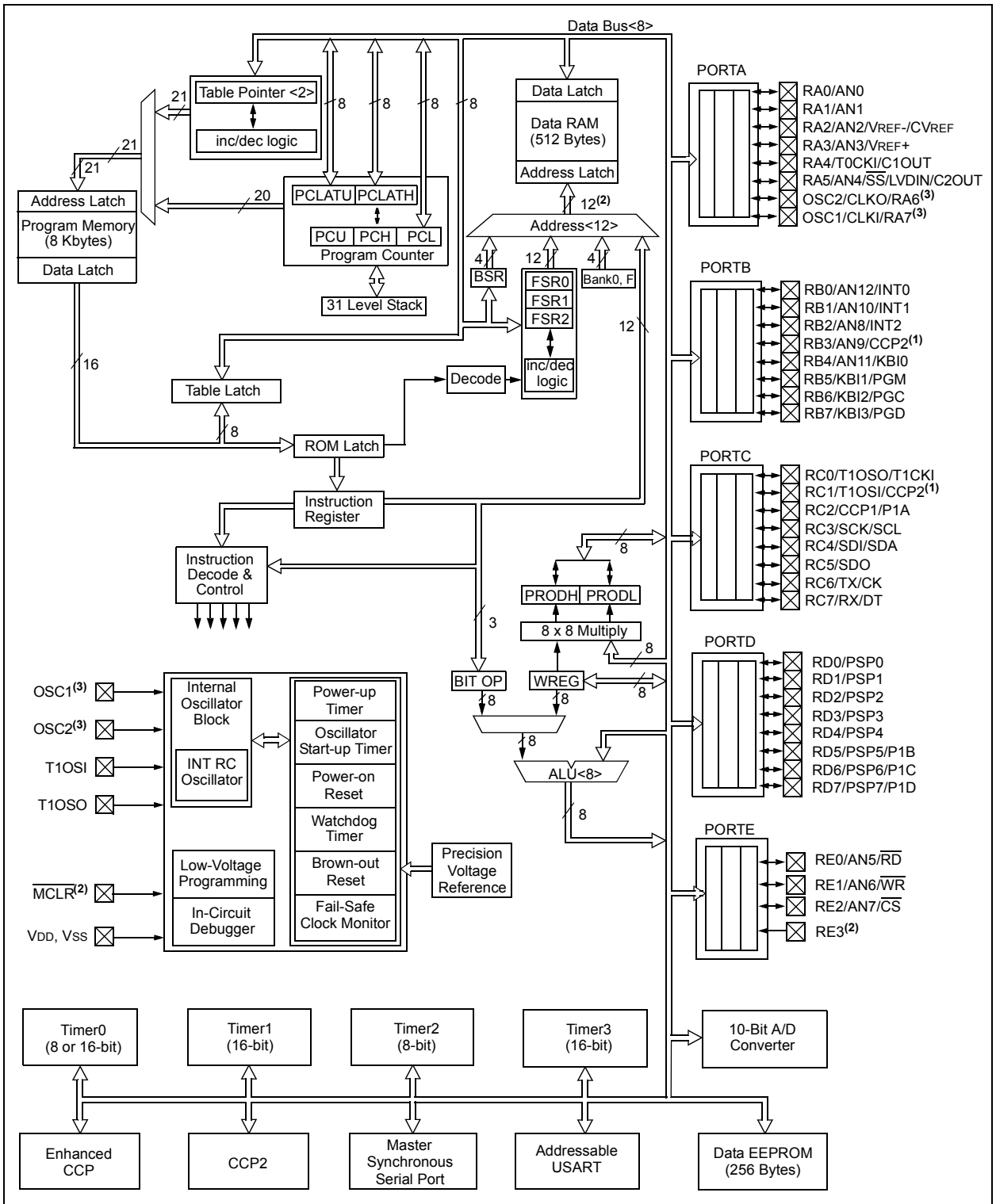
- High-Current Sink/Source 25 mA/25 mA
- Three External Interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution is 6.25 ns (TCY/16)
  - Compare is 16-bit, max. resolution is 100 ns (TCY)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - One, two or four PWM outputs
  - Selectable polarity
  - Programmable dead time
  - Auto-Shutdown and Auto-Restart
- Compatible 10-Bit, Up to 13-Channel Analog-to-Digital Converter (A/D) module with Programmable Acquisition Time
- Dual Analog Comparators
- Addressable USART module:
  - RS-232 operation using internal oscillator block (no external crystal required)

Special Microcontroller Features:

- 100,000 Erase/Write Cycle Enhanced Flash Program Memory Typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory Typical
- Flash/Data EEPROM Retention: > 40 Years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
  - 2% stability over VDD and Temperature
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) via Two Pins
- Wide Operating Voltage Range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		USART	Comparators	Timers 8/16-bit
	Flash (bytes)	# Single Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I²C™			
PIC18F2220	4096	2048	512	256	25	10	2/0	Y	Y	Y	2	2/3
PIC18F2320	8192	4096	512	256	25	10	2/0	Y	Y	Y	2	2/3
PIC18F4220	4096	2048	512	256	36	13	1/1	Y	Y	Y	2	2/3
PIC18F4320	8192	4096	512	256	36	13	1/1	Y	Y	Y	2	2/3

FIGURE 1-2: PIC18F4220/4320 BLOCK DIAGRAM



Note 1: Optional multiplexing of CCP2 input/output with RB3 is enabled by selection of the CCP2MX Configuration bit.

2: RE3 is available only when the MCLR Resets are disabled.

3: OSC1, OSC2, CLKI and CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 2.0 "Oscillator Configurations"** for additional information.

REGISTER 2-3: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R[(1)]	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
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 **IDLEN:** Idle Enable bit
- 1 = Idle mode enabled; CPU core is not clocked in power-managed modes
- 0 = Run mode enabled; CPU core is clocked in power-managed modes

bit 6-4 **IRCF2:IRCF0:** Internal Oscillator Frequency Select bits
- 111 = 8 MHz (8 MHz source drives clock directly)
- 110 = 4 MHz
- 101 = 2 MHz
- 100 = 1 MHz
- 011 = 500 kHz
- 010 = 250 kHz
- 001 = 125 kHz
- 000 = 31 kHz (INTRC source drives clock directly)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit[(1)]
- 1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running
- 0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready

bit 2 **IOFS:** INTOSC Frequency Stable bit
- 1 = INTOSC frequency is stable
- 0 = INTOSC frequency is not stable

bit 1-0 **SCS1:SCS0:** System Clock Select bits
- 1x = Internal oscillator block (RC modes)
- 01 = Timer1 oscillator (Secondary modes)[(2)]
- 00 = Primary oscillator (Sleep and PRI_IDLE modes)

Note 1: Depends on state of IESO bit in Configuration Register 1H.

2: SCS0 may not be set while T1OSCEN (T1CON<3>) is clear.

3.4 Run Modes

If the IDLEN bit is clear when a SLEEP instruction is executed, the CPU and peripherals are both clocked from the source selected using the SCS1:SCS0 bits. While these operating modes may not afford the power conservation of Idle or Sleep modes, they do allow the device to continue executing instructions by using a lower frequency clock source. RC_RUN mode also offers the possibility of executing code at a frequency greater than the primary clock.

Wake-up from a power-managed Run mode can be triggered by an interrupt, or any Reset, to return to full-power operation. As the CPU is executing code in Run modes, several additional exits from Run modes are possible. They include exit to Sleep mode, exit to a corresponding Idle mode, and exit by executing a RESET instruction. While the device is in any of the power-managed Run modes, a WDT time-out will result in a WDT Reset.

3.4.1 PRI_RUN MODE

The PRI_RUN mode is the normal full-power execution mode. If the SLEEP instruction is never executed, the microcontroller operates in this mode (a SLEEP instruction is executed to enter all other power-managed modes). All other power-managed modes exit to PRI_RUN mode when an interrupt or WDT time-out occur.

There is no entry to PRI_RUN mode. The OSTS bit is set. The IOFS bit may be set if the internal oscillator block is the primary clock source (see **Section 2.7.1 "Oscillator Control Register"**).

3.4.2 SEC_RUN MODE

The SEC_RUN mode is the compatible mode to the "clock switching" feature offered in other PIC18 devices. In this mode, the CPU and peripherals are clocked from the Timer1 oscillator. This gives users the option of lower power consumption while still using a high-accuracy clock source.

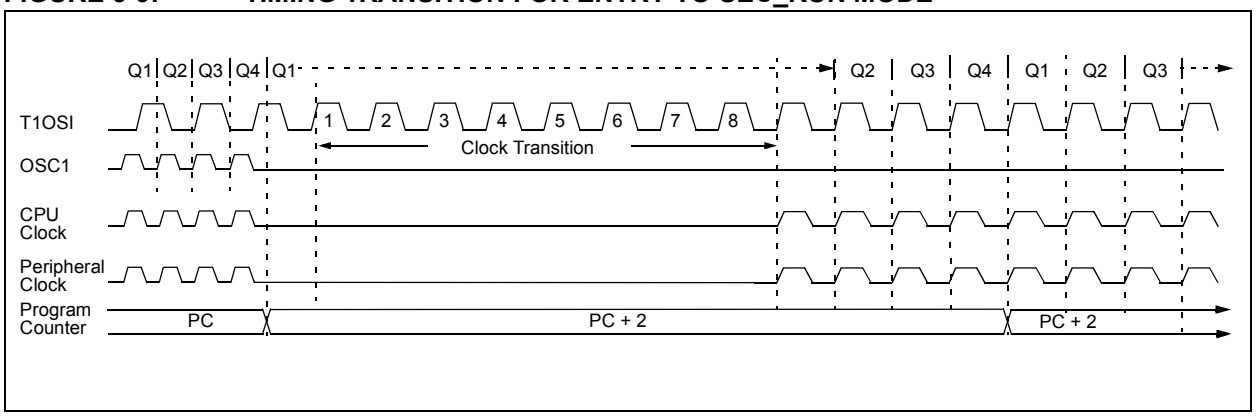
SEC_RUN mode is entered by clearing the IDLEN bit, setting SCS1:SCS0 = 01 and executing a SLEEP instruction. The system clock source is switched to the Timer1 oscillator (see Figure 3-9), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.

Note:	The Timer1 oscillator should already be running prior to entering SEC_RUN mode. **If the T1OSCEN bit is not set when trying to set the SCS0 bit, the write to SCS0 will not occur.** If the Timer1 oscillator is enabled, but not yet running, system clocks will be delayed until the oscillator has started; in such situations, initial oscillator operation is far from stable and unpredictable operation may result.

When a wake-up event occurs, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-6). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the system clock. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run.

Firmware can force an exit from SEC_RUN mode. By clearing the T1OSCEN bit (T1CON<3>), an exit from SEC_RUN back to normal full-power operation is triggered. The Timer1 oscillator will continue to run and provide the system clock even though the T1OSCEN bit is cleared. The primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-6). When the clock switch is complete, the Timer1 oscillator is disabled, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the system clock. The IDLEN and SCS bits are not affected by the wake-up.

FIGURE 3-9: TIMING TRANSITION FOR ENTRY TO SEC_RUN MODE



REGISTER 5-1: STKPTR: STACK POINTER REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	SP4	SP3	SP2	SP1	SP0
bit 7							bit 0

Legend:	C = Clearable bit		
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 **STKFUL:** Stack Full Flag bit(1)
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit(1)
1 = Stack underflow occurred
0 = Stack underflow did not occur

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

bit 4-0 **SP4:SP0:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

5.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable option. To push the current PC value onto the stack, a PUSH instruction can be executed. This will increment the Stack Pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place data or a return address on the stack.

The ability to pull the TOS value off of the stack and replace it with the value that was previously pushed onto the stack, without disturbing normal execution, is achieved by using the POP instruction. The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

5.2.4 STACK FULL/UNDERFLOW RESETS

These Resets are enabled by programming the STVREN bit in Configuration Register 4L. When the STVREN bit is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit but not cause a device Reset. When the STVREN bit is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a POR Reset.

5.12 Indirect Addressing, INDF and FSR Registers

Indirect Addressing is a mode of addressing data memory, where the data memory address in the instruction is not fixed. An FSR register is used as a pointer to the data memory location that is to be read or written. Since this pointer is in RAM, the contents can be modified by the program. This can be useful for data tables in the data memory and for software stacks. Figure 5-8 shows how the fetched instruction is modified prior to being executed.

Indirect Addressing is possible by using one of the INDF registers. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = 0), will read 00h. Writing to the INDF register indirectly, results in a no operation. The FSR register contains a 12-bit address which is shown in Figure 5-9.

The INDFn register is not a physical register. Addressing INDFn actually addresses the register whose address is contained in the FSRn register (FSRn is a pointer); this is indirect addressing.

Example 5-5 shows a simple use of indirect addressing to clear the RAM in Bank 1 (locations 100h-1FFh) in a minimum number of instructions.

EXAMPLE 5-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
          LFSR   FSR0,0x100 ;
NEXT      CLRF   POSTINC0   ; Clear INDF
                            ; register then
                            ; inc pointer
          BTFSS  FSR0H, 1   ; All done with
                            ; Bank1?
          GOTO   NEXT       ; NO, clear next
CONTINUE                    ; YES, continue
```

There are three Indirect Addressing registers. To address the entire data memory space (4096 bytes), these registers are 12 bits wide. To store the 12 bits of addressing information, two 8-bit registers are required:

1. FSR0: composed of FSR0H:FSR0L
2. FSR1: composed of FSR1H:FSR1L
3. FSR2: composed of FSR2H:FSR2L

In addition, there are registers INDF0, INDF1 and INDF2, which are not physically implemented. Reading or writing to these registers activates Indirect Addressing with the value in the corresponding FSR register being the address of the data. If an instruction writes a value to INDF0, the value will be written to the address pointed to by FSR0H:FSR0L. A read from INDF1 reads the data from the address pointed to by FSR1H:FSR1L. INDFn can be used in code anywhere an operand can be used.

If INDF0, INDF1 or INDF2 are read indirectly via an FSR, all '0's are read (zero bit is set). Similarly, if INDF0, INDF1 or INDF2 are written to indirectly, the operation will be equivalent to a NOP instruction and the Status bits are not affected.

5.12.1 INDIRECT ADDRESSING OPERATION

Each FSR register has an INDF register associated with it, plus four additional register addresses. Performing an operation using one of these five registers determines how the FSR will be modified during indirect addressing.

When data access is performed using one of the five INDFn locations, the address selected will configure the FSRn register to:

- Do nothing to FSRn after an indirect access (no change) – INDFn
- Auto-decrement FSRn after an indirect access (post-decrement) – POSTDECn
- Auto-increment FSRn after an indirect access (post-increment) – POSTINCn
- Auto-increment FSRn before an indirect access (pre-increment) – PREINCn
- Use the value in the WREG register as an offset to FSRn. Do not modify the value of the WREG or the FSRn register after an indirect access (no change) – PLUSWn

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the STATUS register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

Auto-incrementing or auto-decrementing an FSR affects all 12 bits. That is, when FSRnL overflows from an increment, FSRnH will be incremented automatically.

Adding these features allows the FSRn to be used as a Stack Pointer, in addition to its use for table operations in data memory.

Each FSR has an address associated with it that performs an indexed indirect access. When a data access to this INDFn location (PLUSWn) occurs, the FSRn is configured to add the signed value in the WREG register and the value in FSR to form the address before an indirect access. The FSR value is not changed. The WREG offset range is -128 to +127.

If an FSR register contains a value that points to one of the INDFn, an indirect read will read 00h (zero bit is set) while an indirect write will be equivalent to a NOP (Status bits are not affected).

If an Indirect Addressing write is performed when the target address is an FSRnH or FSRnL register, the data is written to the FSR register but no pre- or post-increment/decrement is performed.

10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to five ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

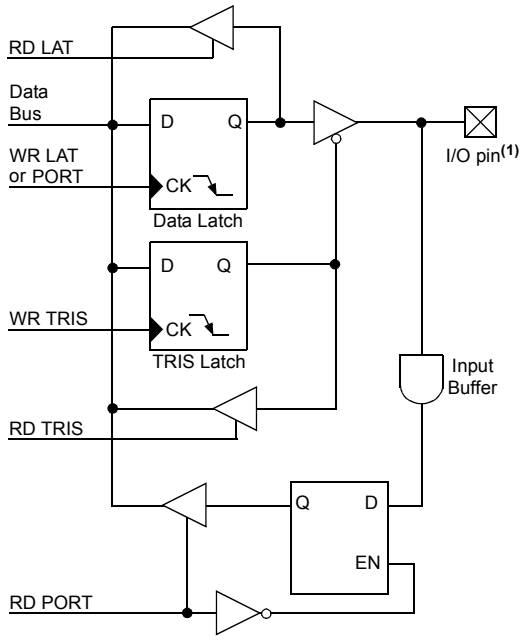
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Data Latch)

The Data Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified model of a generic I/O port without the interfaces to other peripherals is shown in Figure 10-1.

FIGURE 10-1: GENERIC I/O PORT OPERATION



Note 1: I/O pins have diode protection to VDD and VSS.

10.1 PORTA, TRISA and LATA Registers

PORTA is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it, will write to the port latch.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA4 pin is multiplexed with the Timer0 module clock input and one of the comparator outputs to become the RA4/T0CKI/C1OUT pin. Pins RA6 and RA7 are multiplexed with the main oscillator pins; they are enabled as oscillator or I/O pins by the selection of the main oscillator in Configuration Register 1H (see **Section 23.1 "Configuration Bits"** for details). When they are not used as port pins, RA6 and RA7 and their associated TRIS and LAT bits are read as '0'.

The other PORTA pins are multiplexed with analog inputs, the analog VREF+ and VREF- inputs and the comparator voltage reference output. The operation of pins, RA3:RA0 and RA5, as A/D converter inputs is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1). Pins RA0 through RA5 may also be used as comparator inputs or outputs by setting the appropriate bits in the CMCON register.

Note:	On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as '0'. RA4 is configured as a digital input.

The RA4/T0CKI/C1OUT pin is a Schmitt Trigger input and an open-drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

The TRISA register controls the direction of the RA pins even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA   ;   Initialize PORTA by
                ;   clearing output
                ;   data latches
CLRF    LATA    ;   Alternate method
                ;   to clear output
                ;   data latches
MOVLW   0x0F    ;   Set all A/D pins as
MOVWF   ADCON1  ;   digital I/O pins
MOVLW   0xCF    ;   Value used to
                ;   initialize data
                ;   direction
MOVWF   TRISA   ;   Set RA<3:0> as inputs
                ;   RA<5:4> as outputs
```

15.4 Compare Mode

In Compare mode, the 16-bit CCPR1 (CCPR2) register value is constantly compared against either the TMR1 register pair value, or the TMR3 register pair value. When a match occurs, the RC2/CCP1/P1A (RC1/T1OSI/CCP2) pin:

- Is driven high
- Is driven low
- Toggles output (high-to-low or low-to-high)
- Remains unchanged (interrupt only)

The action on the pin is based on the value of control bits, CCP1M3:CCP1M0 (CCP2M3:CCP2M0). At the same time, interrupt flag bit, CCP1IF (CCP2IF), is set.

15.4.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRISC bit.

Note:	Clearing the CCP1CON register will force the RC2/CCP1/P1A compare output latch to the default low level. This is not the PORTC I/O data latch.

15.4.2 TIMER1/TIMER3 MODE SELECTION

Timer1 and/or Timer3 must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

15.4.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

15.4.4 SPECIAL EVENT TRIGGER

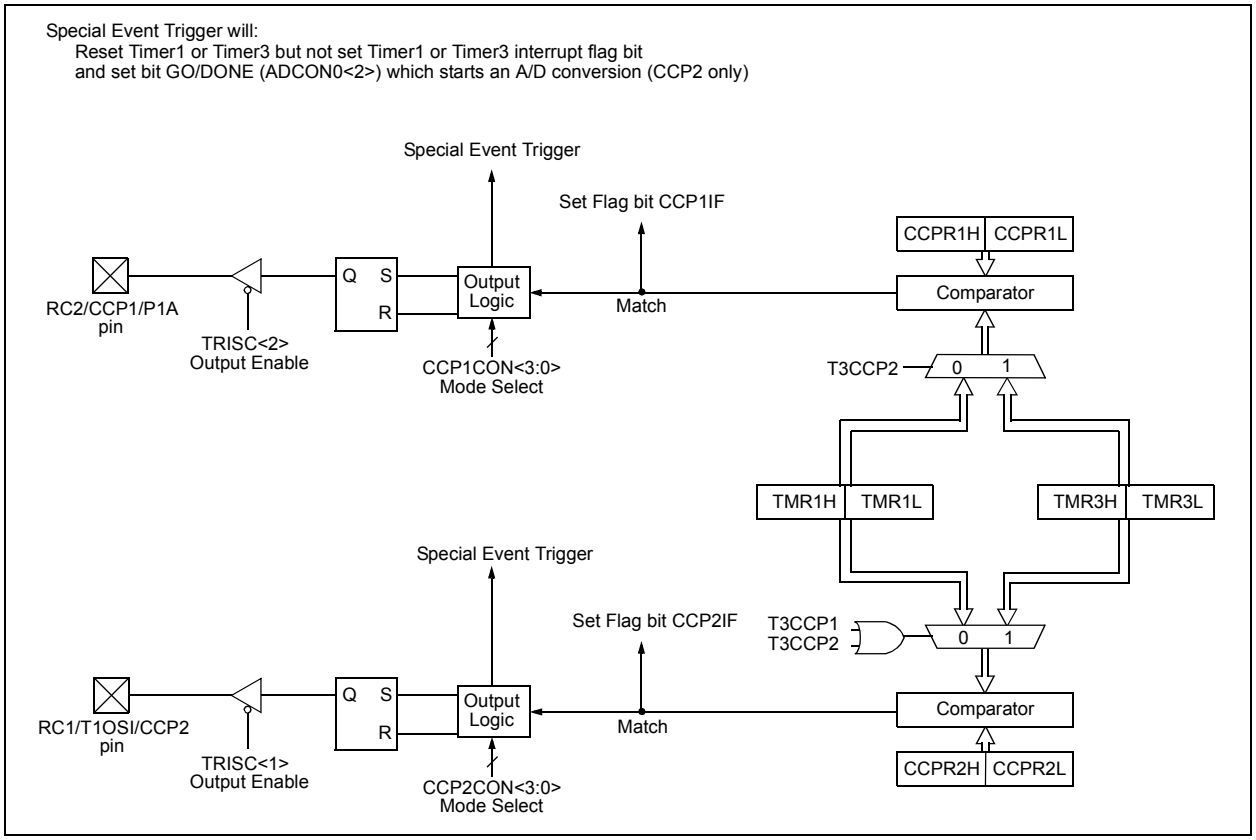
In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The Special Event Trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable Period register for Timer1.

The special trigger output of CCP2 resets either the TMR1 or TMR3 register pair. Additionally, the CCP2 Special Event Trigger will start an A/D conversion if the A/D module is enabled.

Note:	The Special Event Trigger from the CCP2 module will not set the Timer1 or Timer3 interrupt flag bits.

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM



17.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON1 and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 17-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
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 **SMP:** Sample bit
SPI Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Edge Select bit
When CKP = 0:
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK
When CKP = 1:
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit
Used in I^2C mode only.

bit 4 **P:** Stop bit
Used in I^2C mode only.

bit 3 **S:** Start bit
Used in I^2C mode only.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit
Used in I^2C mode only.

bit 1 **UA:** Update Address bit
Used in I^2C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

19.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the Charge Holding Capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 19-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 19-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Example 19-1 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	120 pF
RS	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	5V → Rss = 7 kΩ
Temperature	=	50°C (system max.)
VHOLD	=	0V @ time = 0

19.2 A/D VREF+ and VREF- References

If external voltage references are used instead of the internal AVDD and AVSS sources, the source impedance of the VREF+ and VREF- voltage sources must be considered. During acquisition, currents supplied by these sources are insignificant. However, during conversion, the A/D module sinks and sources current through the reference sources.

In order to maintain the A/D accuracy, the voltage reference source impedances should be kept low to reduce voltage changes. These voltage changes occur as reference currents flow through the reference source impedance. **The maximum recommended impedance of the VREF+ and VREF- external reference voltage sources is 75Ω.**

Note:	When using external references, the source impedance of the external voltage references must be less than 75Ω in order to achieve the specified ADC resolution. A higher reference source impedance will increase the ADC offset and gain errors. Resistive voltage dividers will not provide a low enough source impedance. To ensure the best possible ADC performance, external VREF inputs should be buffered with an op amp or other low-impedance circuit.

EQUATION 19-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 19-2: MINIMUM A/D HOLDING CAPACITOR

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{(-Tc/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

or

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2048)$$

EXAMPLE 19-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$

$T_{AMP} = 5\ \mu s$

$T_{COFF} = (\text{Temp} - 25°C)(0.05\ \mu s/°C)$
$(50°C - 25°C)(0.05\ \mu s/°C)$
$1.25\ \mu s$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, $T_{COFF} = 0\ \mu s$.

$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S) \ln(1/2047)\ \mu s$
$-(120\ pF)(1\ k\Omega + 7\ k\Omega + 2.5\ k\Omega) \ln(0.0004883)\ \mu s$
$9.61\ \mu s$

$T_{ACQ} = 5\ \mu s + 1.25\ \mu s + 9.61\ \mu s$
$12.86\ \mu s$

21.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of CVREF values and has a power-down function to conserve power when the reference is not being used. The CVRCON register controls the operation of the reference as shown in Register 21-1. The block diagram is given in Figure 21-1.

The comparator reference supply voltage comes from VDD and VSS.

21.1 Configuring the Comparator Voltage Reference

The comparator voltage reference can output 16 distinct voltage levels for each range. The equations used to calculate the output of the comparator voltage reference are as follows:

EQUATION 21-1:

$$\text{If CVRR} = 1: \quad CV_{REF} = (CVR\langle 3:0\rangle) \bullet \frac{V_{DD}}{24}$$

$$\text{If CVRR} = 0: \quad CV_{REF} = (CVR\langle 3:0\rangle + 8) \bullet \frac{V_{DD}}{32}$$

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 26-2 in **Section 26.0 "Electrical Characteristics"**).

REGISTER 21-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE(1)	CVRR	—	CVR3	CVR2	CVR1	CVR0
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 **CVREN**: Comparator Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down

bit 6 **CVROE**: Comparator VREF Output Enable bit(1)
1 = CVREF voltage level is also output on the RA2/AN2/VREF-/CVREF pin
0 = CVREF voltage is disconnected from the RA2/AN2/VREF-/CVREF pin

bit 5 **CVRR**: Comparator VREF Range Selection bit
1 = 0.00 VDD to 0.667 VDD, with VDD/24 step size
0 = 0.25 VDD to 0.75 VDD, with VDD/32 step size

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

bit 3-0 **CVR3:CVR0:** Comparator VREF Value Selection 0 ≤ VR3:VR0 ≤ 15 bits

When CVRR = 1:
$CV_{REF} = (CVR\langle 3:0\rangle) \bullet \frac{V_{DD}}{24}$

When CVRR = 0:
$CV_{REF} = 1/4 \bullet (CV_{RSRC}) + (CVR\langle 3:0\rangle + 8) \bullet \frac{V_{DD}}{32}$

Note 1: CVROE overrides the TRISA<2> bit setting.

REGISTER 23-14: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN(1)
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 7-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit(1)
1 = Watchdog Timer is on
0 = Watchdog Timer is off

Note 1: This bit has no effect if the Configuration bit, WDTEN (CONFIG2H<0>), is enabled.

TABLE 23-2: SUMMARY OF WATCHDOG TIMER REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG2H	—	—	—	WDTPS3	WDTPS2	WDTPS2	WDTPS0	WDTEN
RCON	IPEN	—	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
WDTCON	—	—	—	—	—	—	—	SWDTEN

Legend: Shaded cells are not used by the Watchdog Timer.

23.4.3 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.

23.4.4 POR OR WAKE FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary system clock is EC, RC or INTRC modes, monitoring can begin immediately following these events.

For oscillator modes involving a crystal or resonator (HS, HSPLL, LP or XT), the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FCSM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the system clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

Note:	The same logic that prevents false oscillator failure interrupts on POR or wake from Sleep will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 23.3.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary system clock to become stable. When the new powered managed mode is selected, the primary clock is disabled.

23.5 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 five blocks. One of these is a boot block of 512 bytes. The remainder of the memory is divided into four blocks on binary boundaries.

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

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

Figure 23-5 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 23-3.

FIGURE 23-5: CODE-PROTECTED PROGRAM MEMORY FOR PIC18F2X20/4X20

MEMORY SIZE/DEVICE			
4 Kbytes (PIC18F2220/4220)	8 Kbytes (PIC18F2320/4320)	Address Range	Block Code Protection Controlled By:
Boot Block	Boot Block	000000h 0001FFh	CPB, WRTB, EBTRB
Block 0	Block 0	000200h 0007FFh	CP0, WRT0, EBTR0
Block 1	Block 1	000800h 000FFFh	CP1, WRT1, EBTR1
Unimplemented Read '0's	Block 2	001000h 0017FFh	CP2, WRT2, EBTR2
Unimplemented Read '0's	Block 3	001800h 001FFFh	CP3, WRT3, EBTR3
Unimplemented Read '0's	Unimplemented Read '0's	002000h 1FFFFFh	(Unimplemented Memory Space)

TABLE 23-3: 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	—	—	—	—	CP3	CP2	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded cells are unimplemented.

NOTES:

BZ **Branch if Zero**

Syntax: [*label*] BZ n

Operands: -128 ≤ n ≤ 127

Operation: if Zero bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn

Description: If the Zero bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BZ Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If Zero = 1;
PC = address `(Jump)`
If Zero = 0;
PC = address `(HERE + 2)`

CALL **Subroutine Call**

Syntax: [*label*] CALL k [,s]

Operands: 0 ≤ k ≤ 1048575
s ∈ [0,1]

Operation: (PC) + 4 → TOS,
k → PC<20:1>;
if s = 1,
(W) → WS,
(STATUS) → STATUSS,
(BSR) → BSRS

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	110s	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: Subroutine call of entire 2 Mbyte memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>. CALL is a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: `HERE CALL THERE,FAST`

Before Instruction
PC = address `(HERE)`
After Instruction
PC = address `(THERE)`
TOS = address `(HERE + 4)`
WS = W
BSRS = BSR
STATUSS= STATUS

SUBWFB **Subtract W from f with Borrow**

Syntax: [*label*] SUBWFB f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) – (W) – ($\overline{C}$) → dest

Status Affected: N, OV, C, DC, Z

Encoding:

0101	10da	ffff	ffff

Description: Subtract W and the Carry flag (borrow) 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: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG  =  0x19    (0001 1001)
    W    =  0x0D    (0000 1101)
    C    =  0x01
After Instruction
    REG  =  0x0C    (0000 1011)
    W    =  0x0D    (0000 1101)
    C    =  0x01
    Z    =  0x00
    N    =  0x00    ; result is positive
```

Example 2: `SUBWFB REG, 0, 0`

```
Before Instruction
    REG  =  0x1B    (0001 1011)
    W    =  0x1A    (0001 1010)
    C    =  0x00
After Instruction
    REG  =  0x1B    (0001 1011)
    W    =  0x00
    C    =  0x01
    Z    =  0x01    ; result is zero
    N    =  0x00
```

Example 3: `SUBWFB REG, 1, 0`

```
Before Instruction
    REG  =  0x03    (0000 0011)
    W    =  0x0E    (0000 1101)
    C    =  0x01
After Instruction
    REG  =  0xF5    (1111 0100)
                    ; [2's comp]
    W    =  0x0E    (0000 1101)
    C    =  0x00
    Z    =  0x00
    N    =  0x01    ; result is negative
```

TABLE 26-4: LOW-VOLTAGE DETECT CHARACTERISTICS (CONTINUED)

PIC18LF2220/2320/4220/4320 (Industrial)				**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC18F2220/2320/4220/4320 (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.	**Symbol**	**Characteristic**		**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VLVD	LVD Voltage on VDD Transition High-to-Low — Date codes above 0417xxx						
D420G		PIC18F2X20/4X20		Industrial (-10°C to +85°C)				
			LVDL<3:0> = 1101	3.93	4.28	4.62	V	
			LVDL<3:0> = 1110	4.23	4.60	4.96	V	
D420H		PIC18F2X20/4X20		Industrial (-40°C to -10°C)				
			LVDL<3:0> = 1101	3.76	4.28	4.79	V	Reserved
			LVDL<3:0> = 1110	4.04	4.60	5.15	V	
D420J		PIC18F2X20/4X20		Extended (-10°C to +85°C)				
			LVDL<3:0> = 1101	3.94	4.28	4.62	V	
			LVDL<3:0> = 1110	4.23	4.60	4.96	V	
D420K		PIC18F2X20/4X20		Extended (-40°C to -10°C, +85°C to +125°C)				
			LVDL<3:0> = 1101	3.77	4.28	4.79	V	Reserved
			LVDL<3:0> = 1110	4.05	4.60	5.15	V	

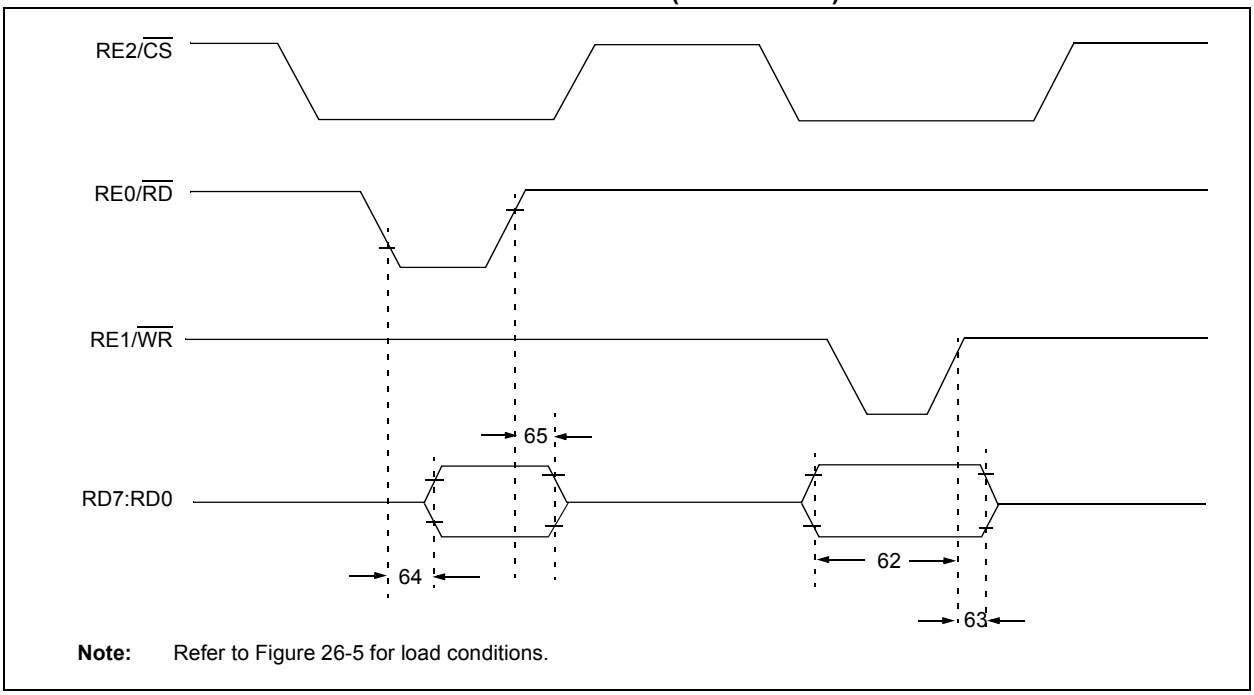
Legend: Shading of rows is to assist in readability of the table.

† Production tested at TAMB = 25°C. Specifications over temperature limits ensured by characterization.

TABLE 26-12: CAPTURE/COMPARE/PWM REQUIREMENTS (ALL CCP MODULES)

Param No.	Symbol	Characteristic			Min	Max	Units	Conditions
50	TccL	CCPx Input Low Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18**F**XX20	10	—	ns	
				PIC18**LF**XX20	20	—	ns	
51	TccH	CCPx Input High Time	No prescaler		0.5 TCY + 20	—	ns	
			With prescaler	PIC18**F**XX20	10	—	ns	
				PIC18**LF**XX20	20	—	ns	
52	TccP	CCPx Input Period			$\frac{3\ T_{CY} + 40}{N}$	—	ns	N = prescale value (1,4 or 16)
53	TccR	CCPx Output Fall Time		PIC18**F**XX20	—	25	ns	
				PIC18**LF**XX20	—	45	ns	
54	TccF	CCPx Output Fall Time		PIC18**F**XX20	—	25	ns	
				PIC18**LF**XX20	—	45	ns	

FIGURE 26-12: PARALLEL SLAVE PORT TIMING (PIC18F4X20)



Note: Refer to Figure 26-5 for load conditions.

FIGURE 27-25: V_{OH} vs. I_{OH} OVER TEMPERATURE (-40°C TO +125°C), V_{DD} = 5.0V

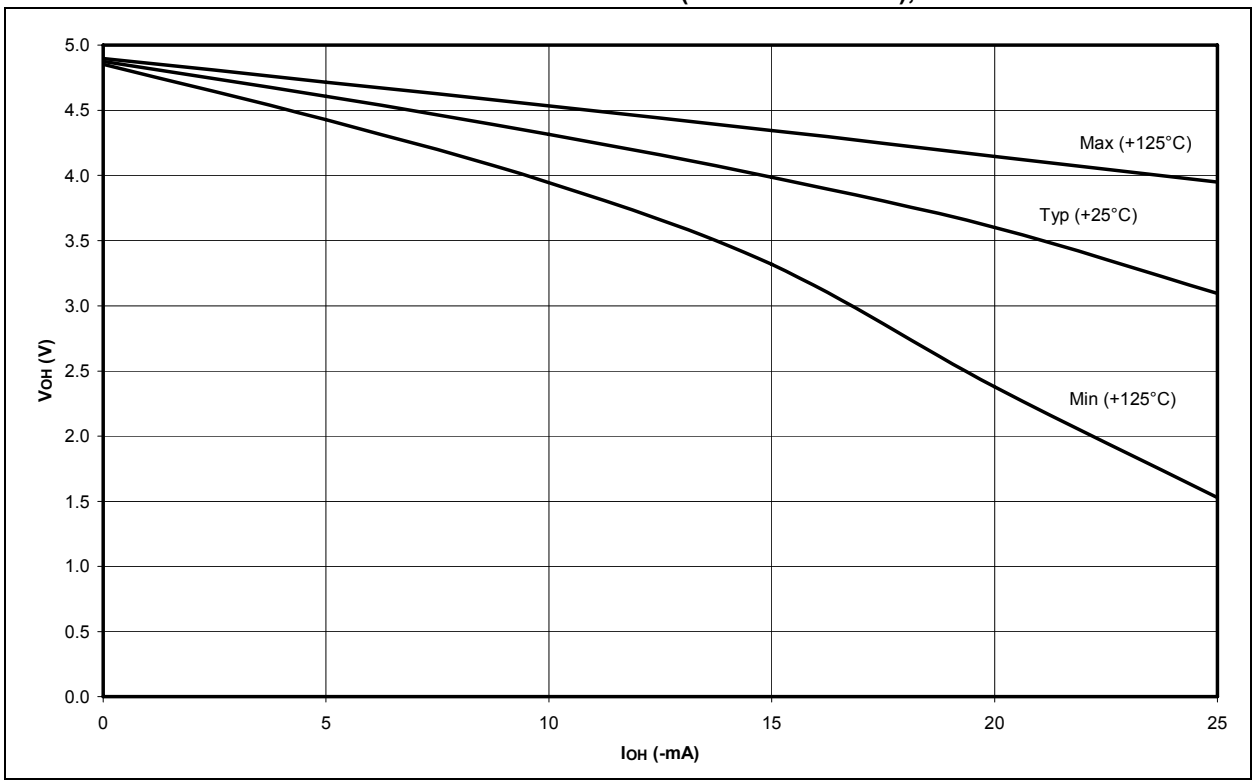


FIGURE 27-26: V_{OL} vs. I_{OL} OVER TEMPERATURE (-40°C TO +125°C), V_{DD} = 3.0V