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)	2V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf1220-i-so

TABLE 1-2: **PIC18F1220/1320 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP/ SOIC	SSOP	QFN			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/AN4/INT0	8	9	9			
RB0				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
INT0				I	ST	External interrupt 0.
RB1/AN5/TX/CK/INT1	9	10	10			
RB1				I/O	TTL	Digital I/O.
AN5				I	Analog	Analog input 5.
TX				O	—	EUSART asynchronous transmit.
CK				I/O	ST	EUSART synchronous clock (see related RX/DT).
INT1				I	ST	External interrupt 1.
RB2/P1B/INT2	17	19	23			
RB2				I/O	TTL	Digital I/O.
P1B				O	—	Enhanced CCP1/PWM output.
INT2				I	ST	External interrupt 2.
RB3/CCP1/P1A	18	20	24			
RB3				I/O	TTL	Digital I/O.
CCP1				I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
P1A				O	—	Enhanced CCP1/PWM output.
RB4/AN6/RX/DT/KBI0	10	11	12			
RB4				I/O	TTL	Digital I/O.
AN6				I	Analog	Analog input 6.
RX				I	ST	EUSART asynchronous receive.
DT				I/O	ST	EUSART synchronous data (see related TX/CK).
KBI0				I	TTL	Interrupt-on-change pin.
RB5/PGM/KBI1	11	12	13			
RB5				I/O	TTL	Digital I/O.
PGM				I/O	ST	Low-Voltage ICSP™ Programming enable pin.
KBI1				I	TTL	Interrupt-on-change pin.
RB6/PGC/T1OSO/ T13CKI/P1C/KBI2	12	13	15			
RB6				I/O	TTL	Digital I/O.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
T1OSO				O	—	Timer1 oscillator output.
T13CKI				I	ST	Timer1/Timer3 external clock output.
P1C				O	—	Enhanced CCP1/PWM output.
KBI2				I	TTL	Interrupt-on-change pin.
RB7/PGD/T1OSI/ P1D/KBI3	13	14	16			
RB7				I/O	TTL	Digital I/O.
PGD				I/O	ST	In-Circuit Debugger and ICSP programming data pin.
T1OSI				I	CMOS	Timer1 oscillator input.
P1D				O	—	Enhanced CCP1/PWM output.
KBI3				I	TTL	Interrupt-on-change pin.
Vss	5	5, 6	3, 5	P	—	Ground reference for logic and I/O pins.
VDD	14	15, 16	17, 19	P	—	Positive supply for logic and I/O pins.
NC	—	—	18	—	—	No connect.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 O = Output
 OD = Open-drain (no P diode to VDD)

 CMOS = CMOS compatible input or output
 I = Input
 P = Power

TABLE 2-2: **CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

Osc Type	Crystal Freq.	Typical Capacitor Values Tested:	
		C1	C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	1 MHz	33 pF	33 pF
	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Capacitor values are for design guidance only.

These capacitors were tested with the crystals listed below for basic start-up and operation. **These values are not optimized.**

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected V$_{DD}$ and temperature range for the application.

See the notes following this table for additional information.

Crystals Used:	
32 kHz	4 MHz
200 kHz	8 MHz
1 MHz	20 MHz

Note 1: Higher capacitance increases the stability of oscillator, but also increases the start-up time.

2: When operating below 3V V$_{DD}$, or when using certain ceramic resonators at any voltage, it may be necessary to use the HS mode or switch to a crystal oscillator.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

4: R$_S$ may be required to avoid overdriving crystals with low drive level specification.

5: Always verify oscillator performance over the V$_{DD}$ and temperature range that is expected for the application.

An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 2-2.

**FIGURE 2-2:** **EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)**
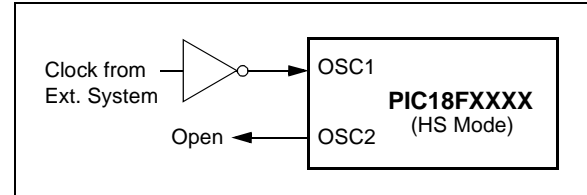
2.3 HSPLL

A Phase Locked Loop (PLL) circuit is provided as an option for users who wish to use a lower frequency crystal oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals.

The HSPLL mode makes use of the HS mode oscillator for frequencies up to 10 MHz. A PLL then multiplies the oscillator output frequency by 4 to produce an internal clock frequency up to 40 MHz.

The PLL is enabled only when the oscillator Configuration bits are programmed for HSPLL mode. If programmed for any other mode, the PLL is not enabled.

FIGURE 2-3: **PLL BLOCK DIAGRAM**
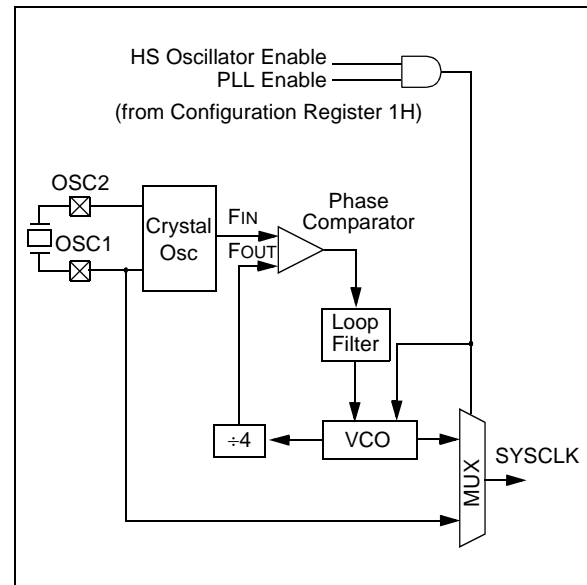
3.6.1 EXAMPLE – EUSART

An adjustment may be indicated when the EUSART begins to generate framing errors, or receives data with errors while in Asynchronous mode. Framing errors indicate that the system clock frequency is too high – try decrementing the value in the OSCTUNE register to reduce the system clock frequency. Errors in data may suggest that the system clock speed is too low – increment OSCTUNE.

3.6.2 EXAMPLE – TIMERS

This technique compares system clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator.

Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is greater than expected, then the internal oscillator block is running too fast – decrement OSCTUNE.

3.6.3 EXAMPLE – CCP IN CAPTURE MODE

A CCP module can use free running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast – decrement OSCTUNE. If the measured time is much less than the calculated time, the internal oscillator block is running too slow–increment OSCTUNE.
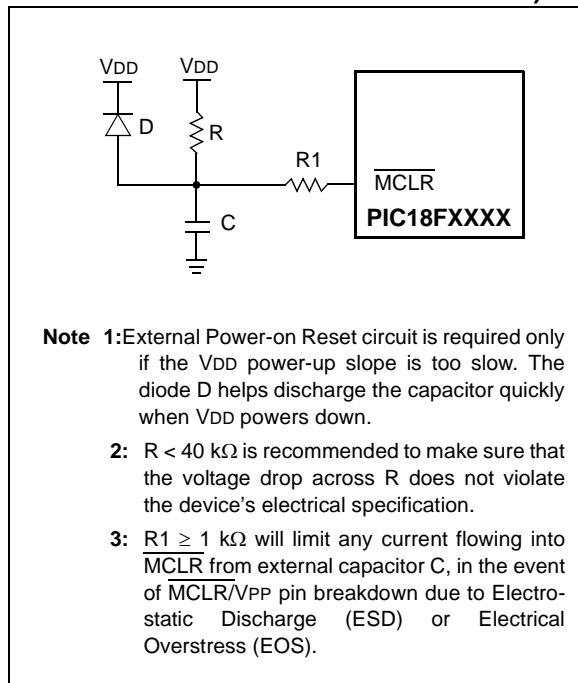
4.1 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when V_{DD} rise is detected. To take advantage of the POR circuitry, just tie the $\overline{MCLR}$ pin through a resistor (1k to 10 kΩ) to V_{DD}. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for V_{DD} is specified (parameter D004). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

FIGURE 4-2: **EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW V_{DD} POWER-UP)**

Note 1: External Power-on Reset circuit is required only if the V_{DD} power-up slope is too slow. The diode D helps discharge the capacitor quickly when V_{DD} powers down.

2: R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 ≥ 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C, in the event of $\overline{MCLR}$/V_{PP} pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

4.2 Power-up Timer (PWRT)

The Power-up Timer (PWRT) of the PIC18F1220/1320 is an 11-bit counter, which uses the INTRC source as the clock input. This yields a count of 2048 x 32 μs = 65.6 ms. While the PWRT is counting, the device is held in Reset.

The power-up time delay will vary from chip-to-chip due to V_{DD}, temperature and process variation. See DC parameter 33 for details.

The PWRT is enabled by clearing Configuration bit, $\overline{PWRTEN}$.

4.3 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (parameter 33). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes and only on Power-on Reset, or on exit from most low-power modes.

4.4 PLL Lock Time-out

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (T_{PLL}) is typically 2 ms and follows the Oscillator Start-up Time-out.

4.5 Brown-out Reset (BOR)

A Configuration bit, BOR, can disable (if clear/programmed), or enable (if set) the Brown-out Reset circuitry. If V_{DD} falls below V_{BOR} (parameter D005) for greater than T_{BOR} (parameter 35), the brown-out situation will reset the chip. A Reset may not occur if V_{DD} falls below V_{BOR} for less than T_{BOR}. The chip will remain in Brown-out Reset until V_{DD} rises above V_{BOR}. If the Power-up Timer is enabled, it will be invoked after V_{DD} rises above V_{BOR}; it then will keep the chip in Reset for an additional time delay, T_{PWRT} (parameter 33). If V_{DD} drops below V_{BOR} while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once V_{DD} rises above V_{BOR}, the Power-up Timer will execute the additional time delay. Enabling BOR Reset does not automatically enable the PWRT.

4.6 Time-out Sequence

On power-up, the time-out sequence is as follows: First, after the POR pulse has cleared, PWRT time-out is invoked (if enabled). Then, the OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 4-3, Figure 4-4, Figure 4-5, Figure 4-6 and Figure 4-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, all time-outs will expire. Bringing $\overline{MCLR}$ high will begin execution immediately (Figure 4-5). This is useful for testing purposes or to synchronize more than one PIC18FXXXX device operating in parallel.

Table 4-2 shows the Reset conditions for some Special Function Registers, while Table 4-3 shows the Reset conditions for all the registers.

TABLE 4-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
BSR	1220	1320	---- 0000	---- 0000	---- uuuu
INDF2	1220	1320	N/A	N/A	N/A
POSTINC2	1220	1320	N/A	N/A	N/A
POSTDEC2	1220	1320	N/A	N/A	N/A
PREINC2	1220	1320	N/A	N/A	N/A
PLUSW2	1220	1320	N/A	N/A	N/A
FSR2H	1220	1320	---- 0000	---- 0000	---- uuuu
FSR2L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
STATUS	1220	1320	---x xxxx	---u uuuu	---u uuuu
TMR0H	1220	1320	0000 0000	0000 0000	uuuu uuuu
TMR0L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T0CON	1220	1320	1111 1111	1111 1111	uuuu uuuu
OSCCON	1220	1320	0000 q000	0000 q000	uuuu qquu
LVDCON	1220	1320	--00 0101	--00 0101	--uu uuuu
WDTCON	1220	1320	---- ---0	---- ---0	---- ---u
RCON[4]	1220	1320	0--1 11q0	0--q qquu	u--u qquu
TMR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	1220	1320	0000 0000	u0uu uuuu	uuuu uuuu
TMR2	1220	1320	0000 0000	0000 0000	uuuu uuuu
PR2	1220	1320	1111 1111	1111 1111	1111 1111
T2CON	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADRESH	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1220	1320	00-0 0000	00-0 0000	uu-u uuuu
ADCON1	1220	1320	-000 0000	-000 0000	-uuu uuuu
ADCON2	1220	1320	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	1220	1320	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
PWM1CON	1220	1320	0000 0000	0000 0000	uuuu uuuu
ECCPAS	1220	1320	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

    2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

    3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

    4: See Table 4-2 for Reset value for specific condition.

    5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the Oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

    6: Bit 5 of PORTA is enabled if MCLR is disabled.

REGISTER 5-1:   **STKPTR: STACK POINTER REGISTER**

R/C-0/0	R/C-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
STKFUL[1]	STKUNF[1]	—	SP<4:0>				
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	C = Clearable only bit

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   **SP<4:0>:** 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 STVR bit in Configuration Register 4L. When the STVR bit is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. When the STVR 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 Power-on Reset.

6.4 Erasing Flash Program Memory

The minimum erase block size is 32 words or 64 bytes under firmware control. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be bulk erased. Word erase in Flash memory is not supported.

When initiating an erase sequence from the micro-controller itself, a block of 64 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. TBLPTR<5:0> are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The CFGS bit must be clear to access program Flash and data EEPROM memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. The WR bit is set as part of the required instruction sequence (as shown in Example 6-2) and starts the actual erase operation. It is not necessary to load the TABLAT register with any data as it is ignored.

For protection, the write initiate sequence using EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

6.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer with address of row being erased.
2. Set the EECON1 register for the erase operation:
   •set EEPGD bit to point to program memory;
   •clear the CFGS bit to access program memory;
   •set WREN bit to enable writes;
   •set FREE bit to enable the erase.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write AAh to EECON2.
6. Set the WR bit. This will begin the row erase cycle.
7. The CPU will stall for duration of the erase (about 2 ms using internal timer).
8. Execute a NOP.
9. Re-enable interrupts.

EXAMPLE 6-2: ERASING A FLASH PROGRAM MEMORY ROW

```
                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
                MOVWF   TBLPTRL
        ERASE_ROW
                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
                MOVWF   EECON2               ; write 55H
    Required    MOVLW   AAh
    Sequence    MOVWF   EECON2               ; write AAH
                BSF     EECON1, WR           ; start erase (CPU stall)
                NOP
                BSF     INTCON, GIE          ; re-enable interrupts
```

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
```

REGISTER 7-1: **EECON1: EEPROM CONTROL 1 REGISTER**

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR[1]	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

bit 7   **EEPGD:** Flash Program or Data EEPROM Memory Select bit

         1 = Access program Flash memory
         0 = Access data EEPROM memory

bit 6   **CFGS:** Flash Program/Data EEPROM or Configuration Select bit

         1 = Accesses Configuration, User ID and Device ID Registers
         0 = Accesses Flash Program or data EEPROM Memory

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

bit 4   **FREE:** Flash Row Erase Enable bit

         1 = Erase the program memory row addressed by TBLPTR on the next WR command
             (cleared by completion of erase operation – TBLPTR<5:0> are ignored)
         0 = Perform write only

bit 3   **WRERR:** EEPROM Error Flag bit[1]

         1 = A write operation was prematurely terminated (any Reset during self-timed programming)
         0 = The write operation completed normally

bit 2   **WREN:** Program/Erase Enable bit

         1 = Allows program/erase cycles
         0 = Inhibits programming/erasing of program Flash and data EEPROM

bit 1   **WR:** Write Control bit

         1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle.
             (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR
             bit can only be set (not cleared) in software.)
         0 = Write cycle completed

bit 0   **RD:** Read Control bit

         1 = Initiates a memory read
             (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in
             software. RD bit cannot be set when EEPGD = 1.)
         0 = Read completed

Note 1: When a WRERR occurs, the EEPGD and CFGS bits are not cleared. This allows
               tracing of the error condition.

REGISTER 9-7: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0
OSCFIE	—	—	EEIE	—	LVDIE	TMR3IE	—
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   **OSCFIE:** Oscillator Fail Interrupt Enable bit
                      1 = Enabled
                      0 = Disabled

bit 6-5   **Unimplemented:** Read as '0'

bit 4   **EEIE:** Data EEPROM/Flash Write Operation Interrupt Enable bit
                      1 = Enabled
                      0 = Disabled

bit 3   **Unimplemented:** Read as '0'

bit 2   **LVDIE:** Low-Voltage Detect Interrupt Enable bit
                      1 = Enabled
                      0 = Disabled

bit 1   **TMR3IE:** TMR3 Overflow Interrupt Enable bit
                      1 = Enabled
                      0 = Disabled

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

9.4 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are two Peripheral Interrupt Priority registers (IPR1, IPR2). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

REGISTER 9-8:   IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

U-0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP
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   **ADIP:** A/D Converter Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

bit 5   **RCIP:** EUSART Receive Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

bit 4   **TXIP:** EUSART Transmit Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

bit 3   **Unimplemented:** Read as '0'

bit 2   **CCP1IP:** CCP1 Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

bit 1   **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

bit 0   **TMR1IP:** TMR1 Overflow Interrupt Priority bit

　　　　1 = High priority
　　　　0 = Low priority

9.6 INTn Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1 and RB2/INT2 pins are edge-triggered: either rising if the corresponding INTEDGx bit is set in the INTCON2 register, or falling if the INTEDGx bit is clear. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit, INTxF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxE. Flag bit, INTxF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt. All external interrupts (INT0, INT1 and INT2) can wake-up the processor from low-power modes if bit INTxE was set prior to going into low-power modes. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up.

Interrupt priority for INT1 and INT2 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>) and INT2IP (INTCON3<7>). There is no priority bit associated with INT0. It is always a high priority interrupt source.

9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow (FFh $\rightarrow$ 00h) in the TMR0 register will set flag bit, TMR0IF. In 16-bit mode, an overflow (FFFFh $\rightarrow$ 0000h) in the TMR0H:TMR0L registers will set flag bit, TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 11.0 "Timer0 Module"** for further details on the Timer0 module.

9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit, RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).
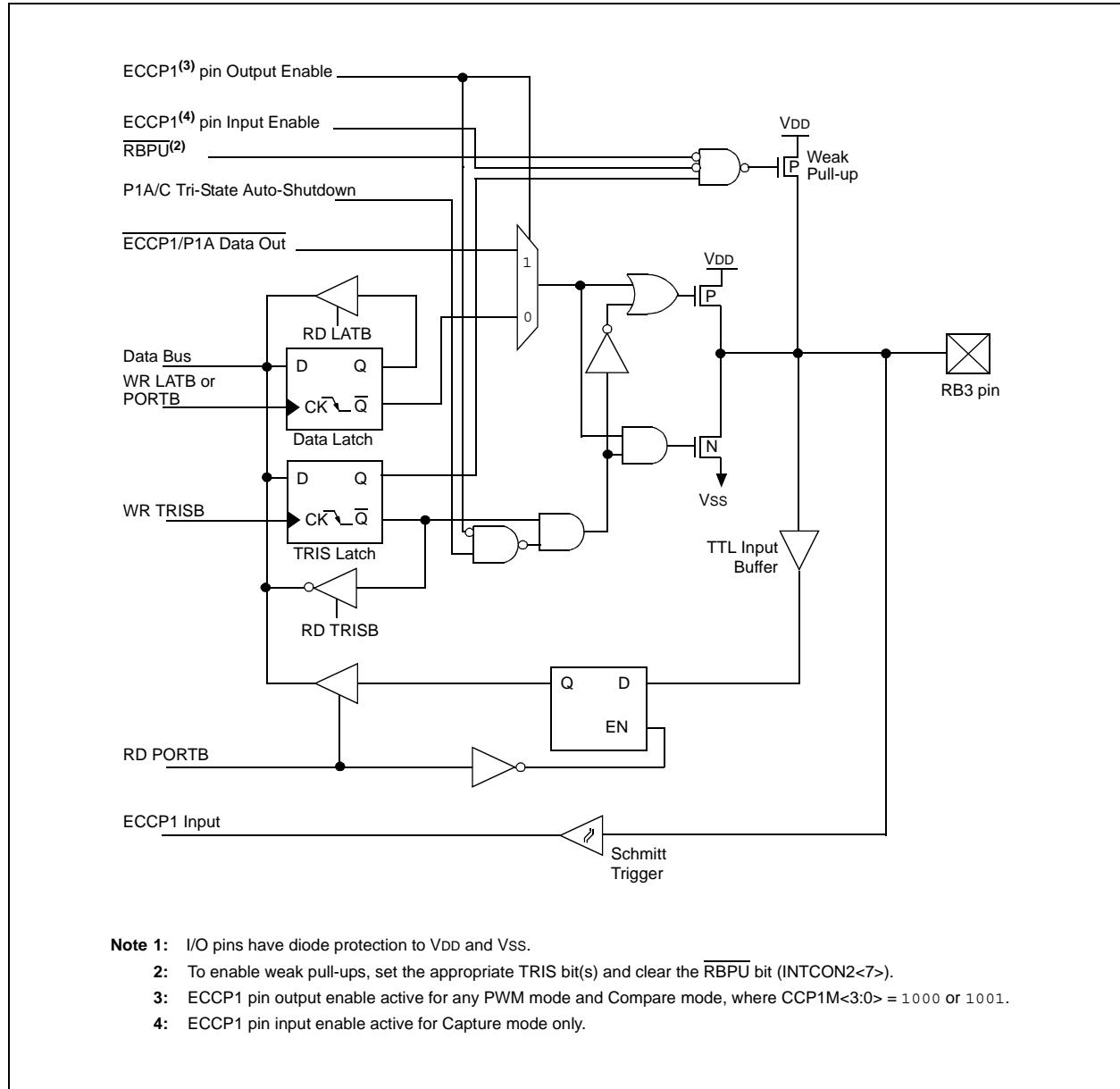
9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, Status and BSR registers are saved on the fast return stack. If a fast return from interrupt is not used (see **Section 5.3 "Fast Register Stack"**), the user may need to save the WREG, Status and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, Status and BSR registers during an Interrupt Service Routine.

EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF   W_TEMP                  ; W_TEMP is in virtual bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

FIGURE 10-10: BLOCK DIAGRAM OF RB3/CCP1/P1A PIN

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

2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (INTCON2<7>).

3: ECCP1 pin output enable active for any PWM mode and Compare mode, where CCP1M<3:0> = `1000` or `1001`.

4: ECCP1 pin input enable active for Capture mode only.

15.3.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON       ; Turn CCP module off
MOVLW   NEW_CAPT_PS   ; Load WREG with the
                      ; new prescaler mode
                      ; value and CCP ON
MOVWF   CCP1CON       ; Load CCP1CON with
                      ; this value
```

FIGURE 15-1: CAPTURE MODE OPERATION BLOCK DIAGRAM
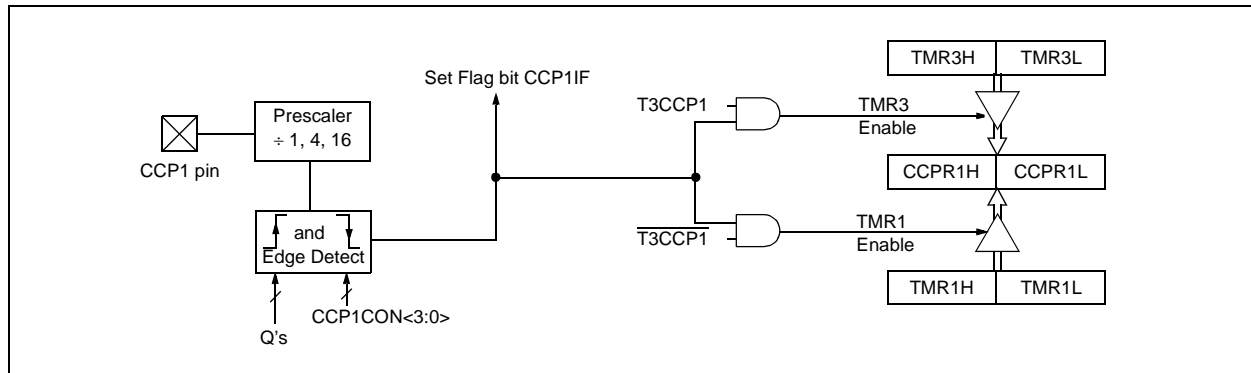
15.4 Compare Mode

In Compare mode, the 16-bit CCPR1 register value is constantly compared against either the TMR1 register pair value, or the TMR3 register pair value. When a match occurs, the RB3/CCP1/P1A 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. At the same time, interrupt flag bit, CCP1IF, is set.

15.4.1 CCP PIN CONFIGURATION

The user must configure the RB3/CCP1/P1A pin as an output by clearing the TRISB<3> bit.

Note:	Clearing the CCP1CON register will force the RB3/CCP1/P1A compare output latch to the default low level. This is not the PORTB 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 RB3/CCP1/P1A pin is not affected. CCP1IF is set and an interrupt is generated (if enabled).
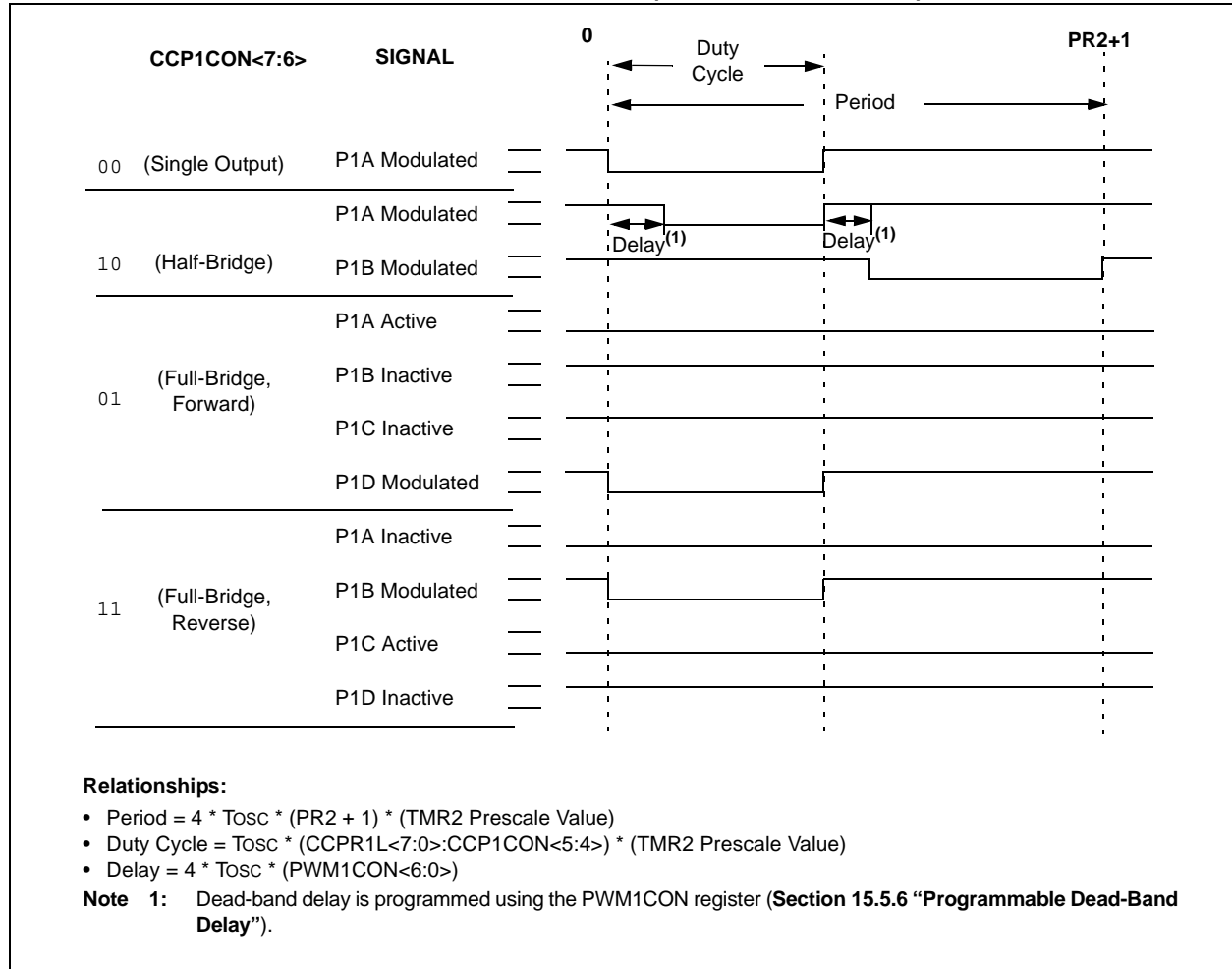
15.4.4 SPECIAL EVENT TRIGGER

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 event trigger also sets the GO/DONE bit (ADCON0<1>). This starts a conversion of the currently selected A/D channel if the A/D is on.

FIGURE 15-5: **PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)**

Relationships:

- Period = 4 * T$_{OSC}$ * (PR2 + 1) * (TMR2 Prescale Value)
- Duty Cycle = T$_{OSC}$ * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * T$_{OSC}$ * (PWM1CON<6:0>)

Note 1: Dead-band delay is programmed using the PWM1CON register (**Section 15.5.6 "Programmable Dead-Band Delay"**).

IORLW	**Inclusive OR literal with W**
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k $\to$ W
Status Affected:	N, Z
Encoding:	

0000	1001	kkkk	kkkk

Description: The contents of W are OR'ed with 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

<u>Example</u>:   IORLW   0x35

Before Instruction

   W   =   0x9A

After Instruction

   W   =   0xBF

IORWF	**Inclusive OR W with f**
Syntax:	[*label*] IORWF f [,d [,a]]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(W) .OR. (f) $\to$ dest
Status Affected:	N, Z
Encoding:	

0001	00da	ffff	ffff

Description: Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 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

<u>Example</u>:   IORWF RESULT, W

Before Instruction

   RESULT =   0x13
   W   =   0x91

After Instruction

   RESULT =   0x13
   W   =   0x93

MOVLW	**Move literal to W**
Syntax:	[*label*] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	$k \rightarrow W$
Status Affected:	None
Encoding:	

0000	1110	kkkk	kkkk

Description:	The 8-bit literal 'k' is loaded into W.
Words:	1
Cycles:	1

Q Cycle Activity:

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

Example: MOVLW   0x5A

After Instruction

W   =   0x5A

MOVWF	**Move W to f**
Syntax:	[*label*] MOVWF f [,a]
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$
Operation:	$(W) \rightarrow f$
Status Affected:	None
Encoding:	

0110	111a	ffff	ffff

Description:	Move data from W to register 'f'. Location 'f' can be anywhere in the 256-byte bank. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 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 register 'f'

Example: MOVWF   REG

Before Instruction

W   =   0x4F
REG   =   0xFF

After Instruction

W   =   0x4F
REG   =   0x4F

SLEEP	Enter Sleep mode

Syntax:	[*label*] SLEEP
Operands:	None
Operation:	00h → WDT, 0 → WDT postscaler, 1 → $\overline{TO}$, 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0011

Description:	The Power-down Status bit ($\overline{PD}$) is cleared. The Time-out status bit ($\overline{TO}$) is set. The Watchdog Timer and its postscaler are cleared. The processor is put into Sleep mode with the oscillator stopped.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to Sleep

Example: SLEEP

Before Instruction
$\overline{TO}$ =   ?
$\overline{PD}$ =   ?

After Instruction
$\overline{TO}$ =   1 †
$\overline{PD}$ =   0

†   If WDT causes wake-up, this bit is cleared.

SUBFWB	Subtract f from W with borrow

Syntax:	[*label*] SUBFWB f [,d [,a]]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	$(W) - (f) - (\overline{C}) \to$ dest
Status Affected:	N, OV, C, DC, Z

Encoding:

0101	01da	ffff	ffff

Description:	Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored 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:   SUBFWB REG

Before Instruction
REG   =   0x03
W   =   0x02
C   =   0x01

After Instruction
REG   =   0xFF
W   =   0x02
C   =   0x00
Z   =   0x00
N   =   0x01   ; result is negative

Example 2:   SUBFWB   REG, 0, 0

Before Instruction
REG   =   2
W   =   5
C   =   1

After Instruction
REG   =   2
W   =   3
C   =   1
Z   =   0
N   =   0   ; result is positive

Example 3:   SUBFWB   REG, 1, 0

Before Instruction
REG   =   1
W   =   2
C   =   0

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

XORWF	Exclusive OR W with f

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

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

Operation:	(W) .XOR. (f) $\rightarrow$ dest

Status Affected:	N, Z

Encoding:	0001	10da	ffff	ffff

Description:	Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the 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: XORWF   REG

Before Instruction

REG	=	0xAF
W	=	0xB5

After Instruction

REG	=	0x1A
W	=	0xB5

PIC18F1220/1320

APPENDIX E: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442"*. The changes discussed, while device specific, are generally applicable to all mid-range to enhanced device migrations.

This Application Note is available as Literature Number DS00716.

APPENDIX F: MIGRATION FROM HIGH-END TO ENHANCED DEVICES

A detailed discussion of the migration pathway and differences between the high-end MCU devices (i.e., PIC17CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in *AN726, "PIC17CXXX to PIC18CXXX Migration"*.

This Application Note is available as Literature Number DS00726.