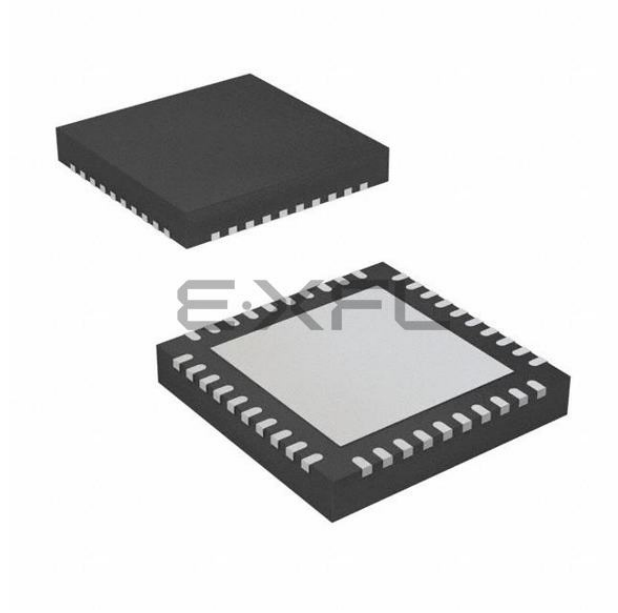


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 35x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k40-i-mv

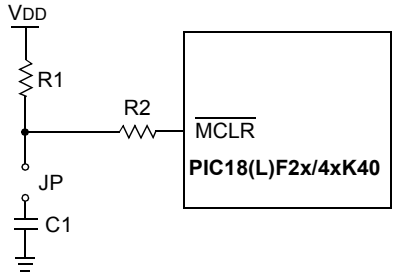
2.3 Master Clear (MCLR) Pin

The MCLR pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the MCLR pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the MCLR pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the MCLR pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the MCLR pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into MCLR from the external capacitor, C1, in the event of MCLR pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the MCLR pin VIH and VIL specifications are met.

2.4 ICSP™ Pins

The PGC and PGD pins are used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the PGC and PGD pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin input voltage high (VIH) and input low (VIL) requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., PGCx/PGDx pins), programmed into the device, matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 36.0 "Development Support"**.

REGISTER 3-9: Configuration Word 5L (30 0008h): Code Protection

U-1	U-1	U-1	U-1	U-1	U-1	R/W-1	R/W-1
—	—	—	—	—	—	$\overline{CPD}$	$\overline{CP}$
bit 7							bit 0

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

bit 7-2 **Unimplemented:** Read as '1'

bit 1 **$\overline{CPD}$:** Data NVM Memory Code Protection bit
- 1 = Data NVM code protection disabled
- 0 = Data NVM code protection enabled

bit 0 **$\overline{CP}$:** User NVM Program Memory Code Protection bit
- 1 = User NVM code protection disabled
- 0 = User NVM code protection enabled

REGISTER 3-10: Configuration Word 6L (30 000Ah): Memory Read Protection

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
EBTR7	EBTR6	EBTR5	EBTR4	EBTR3	EBTR2	EBTR1	EBTR0
bit 7							bit 0

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

bit 7-0 **EBTR<7:0>:** Table Read Protection bits(1)
- 1 = Corresponding Memory Block NOT protected from table reads executed in other blocks
- 0 = Corresponding Memory Block protected from table reads executed in other blocks

Note 1: Refer to Table 10-2 for details on implementation of the individual EBTR bits.

REGISTER 3-11: Configuration Word 6H (30 000Bh): Memory Read Protection

U-1	U-1	U-1	U-1	U-1	U-1	R/W-1	U-1
—	—	—	—	—	—	EBTRB	—
bit 7							bit 0

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

bit 7-2 **Unimplemented:** Read as '1'

bit 1 **EBTRB:** Table Read Protection bit
- 1 = Memory Boot Block NOT protected from table reads executed in other blocks
- 0 = Memory Boot Block protected from table reads executed in other blocks

bit 0 **Unimplemented:** Read as '1'

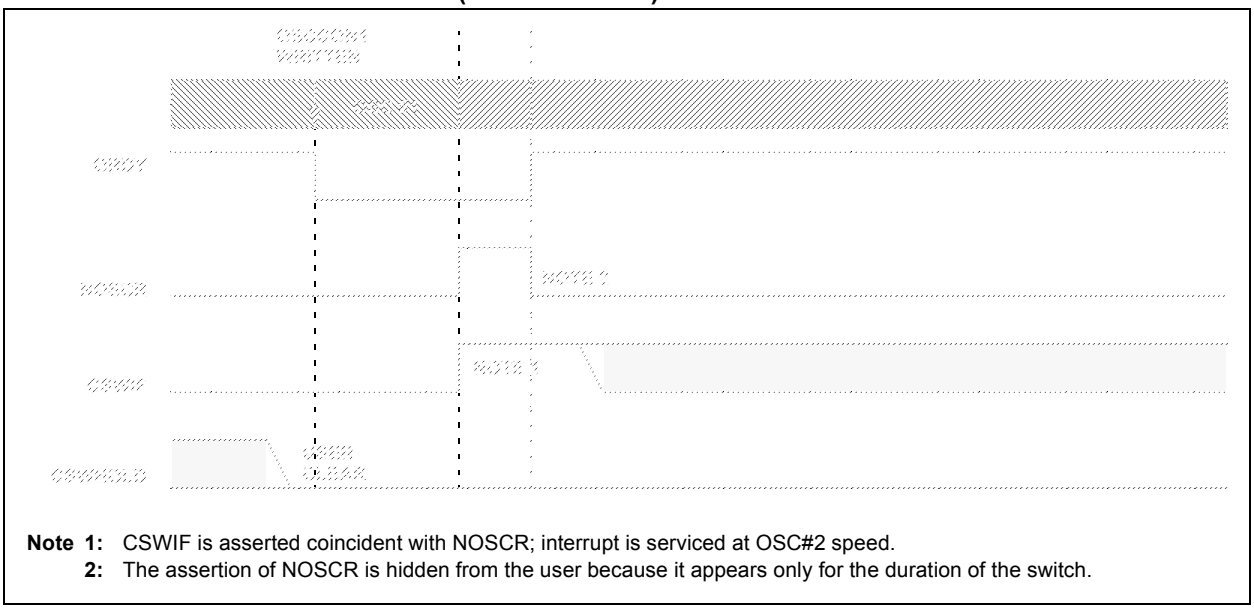
4.4.2 CLOCK SWITCH AND SLEEP

If OSCCON1 is written with a new value and the device is put to Sleep before the switch completes, the switch will not take place and the device will enter Sleep mode.

When the device wakes from Sleep and the CSWHOLD bit is clear, the device will wake with the 'new' clock active, and the clock switch interrupt flag bit (CSWIF) will be set.

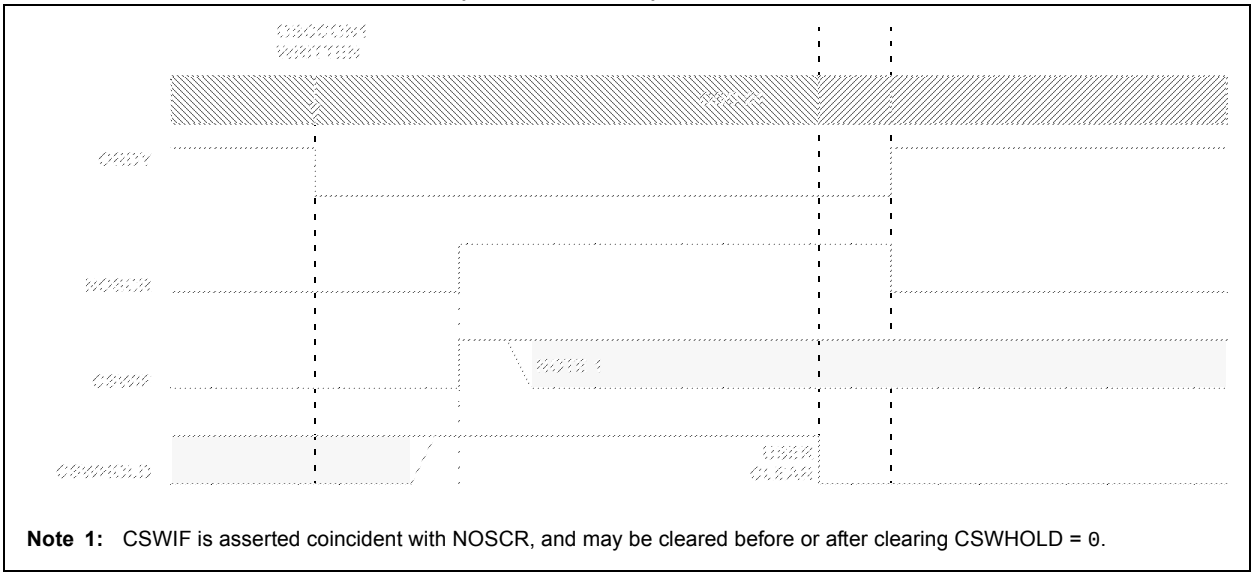
When the device wakes from Sleep and the CSWHOLD bit is set, the device will wake with the 'old' clock active and the new clock will be requested again.

FIGURE 4-6: CLOCK SWITCH (CSWHOLD = 0)



Note 1: CSWIF is asserted coincident with NOSCR; interrupt is serviced at OSC#2 speed.
2: The assertion of NOSCR is hidden from the user because it appears only for the duration of the switch.

FIGURE 4-7: CLOCK SWITCH (CSWHOLD = 1)



Note 1: CSWIF is asserted coincident with NOSCR, and may be cleared before or after clearing CSWHOLD = 0.

7.5 Register Definitions: Peripheral Module Disable

REGISTER 7-1: PMD0: PMD CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SYSCMD	FVRMD	HLVDMD	CRCMD	SCANMD	NVMMD	CLKRMD	IOCMD
7							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	q = Value depends on condition

bit 7 **SYSCMD:** Disable Peripheral System Clock Network bit(1)
See description in **Section 7.4 "System Clock Disable"**.
1 = System clock network disabled (FOSC)
0 = System clock network enabled

bit 6 **FVRMD:** Disable Fixed Voltage Reference bit
1 = FVR module disabled
0 = FVR module enabled

bit 5 **HLVDMD:** Disable Low-Voltage Detect bit
1 = HLVD module disabled
0 = HLVD module enabled

bit 4 **CRCMD:** Disable CRC Engine bit
1 = CRC module disabled
0 = CRC module enabled

bit 3 **SCANMD:** Disable NVM Memory Scanner bit(2)
1 = NVM Memory Scan module disabled
0 = NVM Memory Scan module enabled

bit 2 **NVMMD:** NVM Module Disable bit(3)
1 = All Memory reading and writing is disabled; NVMCON registers cannot be written
0 = NVM module enabled

bit 1 **CLKRMD:** Disable Clock Reference bit
1 = CLKR module disabled
0 = CLKR module enabled

bit 0 **IOCMD:** Disable Interrupt-on-Change bit, All Ports
1 = IOC module(s) disabled
0 = IOC module(s) enabled

Note 1: Clearing the SYSCMD bit disables the system clock (FOSC) to peripherals, however peripherals clocked by FOSC/4 are not affected.
2: Subject to SCANE bit in CONFIG4H.
3: When enabling NVM, a delay of up to 1 µs may be required before accessing data.

8.14 Power Control (PCON0) Register

The Power Control (PCON0) register contains flag bits to differentiate between a:

- Brown-out Reset ($\overline{BOR}$)
- Power-on Reset ($\overline{POR}$)
- Reset Instruction Reset ($\overline{RI}$)
- $\overline{MCLR}$ Reset ($\overline{RMCLR}$)
- Watchdog Timer Reset ($\overline{RWDT}$)
- Watchdog Window Violation ($\overline{WDTWV}$)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

The PCON0 register bits are shown in Register 8-2.

Hardware will change the corresponding register bit during the Reset process; if the Reset was not caused by the condition, the bit remains unchanged (Table 8-3).

Software should reset the bit to the inactive state after restart (hardware will not reset the bit).

Software may also set any PCON0 bit to the active state, so that user code may be tested, but no Reset action will be generated.

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	75
PCON0	STKOVF	STKUNF	$\overline{WDTWV}$	$\overline{RWDT}$	$\overline{RMCLR}$	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	76
STATUS	—	$\overline{TO}$	$\overline{PD}$	N	OV	Z	DC	C	118
WDTCON0	—	—	WDTPS<4:0>					SEN	85
WDTCON1	—	WDTCS<2:0>			—	WINDOW<2:0>			86

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Resets.

FIGURE 9-1: WINDOWED WATCHDOG TIMER BLOCK DIAGRAM

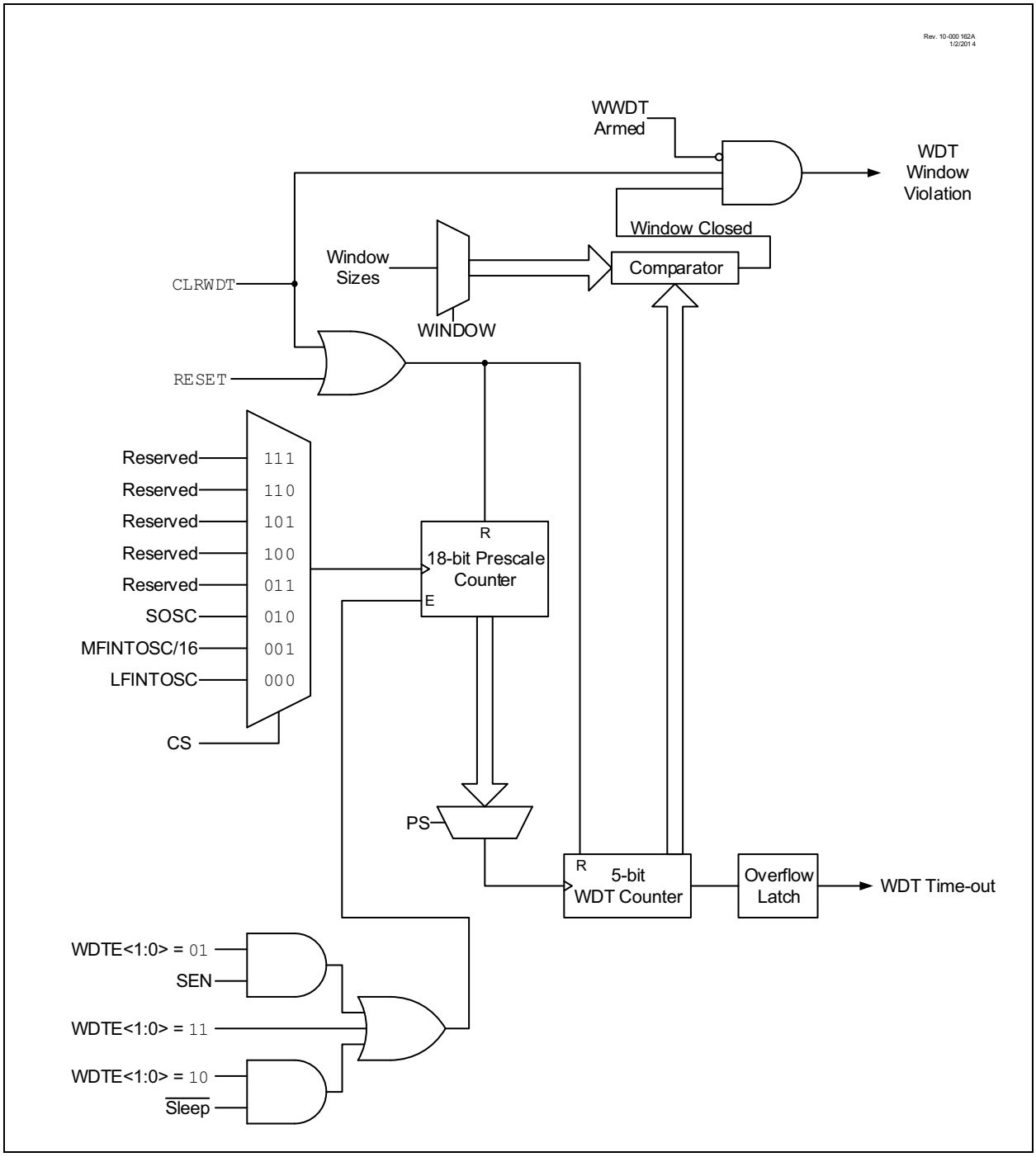


TABLE 10-5: REGISTER FILE SUMMARY FOR PIC18(L)F26/45/46K40 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
FFFh	TOSU	—	—	—	Top of Stack Upper byte (TOS<20:16>)					---xxxxx
FFEh	TOSH	Top of Stack High byte (TOS<15:8>)								xxxxxxxx
FFDh	TOSL	Top of Stack Low byte (TOS<7:0>)								xxxxxxxx
FFCh	STKPTR	—	—	—	STKPTR<4:0>					--000000
FFBh	PCLATU	—	—	—	Holding Register for PC<20:16>					---00000
FFAh	PCLATH	Holding Register for PC<15:8>								00000000
FF9h	PCL	PC Low byte (PC<7:0>)								00000000
FF8h	TBLPTRU	—	—	Program Memory Table Pointer (TBLPTR<21:16>)						--000000
FF7h	TBLPTRH	Program Memory Table Pointer (TBLPTR<15:8>)								00000000
FF6h	TBLPTRL	Program Memory Table Pointer (TBLPTR<7:0>)								00000000
FF5h	TABLAT	TABLAT								00000000
FF4h	PRODH	Product Register High byte								xxxxxxxx
FF3h	PRODL	Product Register Low byte								xxxxxxxx
FF2h	INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	000--111
FF1h	—	Unimplemented								—
FF0h	—	Unimplemented								—
FEFh	INDF0	Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)								--------
FEEh	POSTINC0	Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)								--------
FEDh	POSTDEC0	Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)								--------
FECh	PREINC0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)								--------
FEBh	PLUSW0	Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W								--------
FEAh	FSR0H	—	—	—	—	Indirect Data Memory Address Pointer 0 High				----xxxx
FE9h	FSR0L	Indirect Data Memory Address Pointer 0 Low								xxxxxxxx
FE8h	WREG	Working Register								xxxxxxxx
FE7h	INDF1	Uses contents of FSR0 to address data memory – value of FSR1 not changed (not a physical register)								--------
FE6h	POSTINC1	Uses contents of FSR0 to address data memory – value of FSR1 post-incremented (not a physical register)								--------
FE5h	POSTDEC1	Uses contents of FSR0 to address data memory – value of FSR1 post-decremented (not a physical register)								--------
FE4h	PREINC1	Uses contents of FSR0 to address data memory – value of FSR1 pre-incremented (not a physical register)								--------
FE3h	PLUSW1	Uses contents of FSR0 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR0 offset by W								--------

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Not available on LF devices.

2: Not available on PIC18(L)F26K40 (28-pin variants).

3: Not available on PIC18(L)F45K40 devices.

REGISTER 14-11: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
OSCFIE	CSWIE	—	—	—	—	ADTIE	ADIE
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 **OSCFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **CSWIE:** Clock-Switch Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5-2 **Unimplemented:** Read as '0'

bit 1 **ADTIE:** ADC Threshold Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **ADIE:** ADC Interrupt Enable bit
1 = Enabled
0 = Disabled

TABLE 20-1: TIMER2 OPERATING MODES

Mode	MODE<4:0>		Output Operation	Operation	Timer Control		
	<4:3>	<2:0>			Start	Reset	Stop
Free Running Period	00	000	Period Pulse	Software gate (Figure 20-4)	ON = 1	—	ON = 0
		001		Hardware gate, active-high (Figure 20-5)	ON = 1 and TMRx_ers = 1	—	ON = 0 or TMRx_ers = 0
		010		Hardware gate, active-low	ON = 1 and TMRx_ers = 0	—	ON = 0 or TMRx_ers = 1
		011	Period Pulse with Hardware Reset	Rising or falling edge Reset	ON = 1	TMRx_ers ↕	ON = 0
		100		Rising edge Reset (Figure 20-6)		TMRx_ers ↑	
		101		Falling edge Reset		TMRx_ers ↓	
		110		Low level Reset		TMRx_ers = 0	ON = 0 or TMRx_ers = 0
		111		High level Reset (Figure 20-7)		TMRx_ers = 1	ON = 0 or TMRx_ers = 1
One-shot	01	000	One-shot	Software start (Figure 20-8)	ON = 1	—	ON = 0 or Next clock after TMRx = PRx (**Note 2**)
		001	Edge triggered start (**Note 1**)	Rising edge start (Figure 20-9)	ON = 1 and TMRx_ers ↑	—	
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
		100	Edge triggered start and hardware Reset (**Note 1**)	Rising edge start and Rising edge Reset (Figure 20-10)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑	
		101		Falling edge start and Falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers ↓	
		110		Rising edge start and Low level Reset (Figure 20-11)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0	
		111		Falling edge start and High level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1	
Mono-stable	10	000	Reserved				
		001	Edge triggered start (**Note 1**)	Rising edge start (Figure 20-12)	ON = 1 and TMRx_ers ↑	—	ON = 0 or Next clock after TMRx = PRx (**Note 3**)
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
Reserved		100	Reserved				
Reserved		101	Reserved				
One-shot		110	Level triggered start and hardware Reset	High level start and Low level Reset (Figure 20-13)	ON = 1 and TMRx_ers = 1	TMRx_ers = 0	ON = 0 or Held in Reset (**Note 2**)
		111		Low level start & High level Reset	ON = 1 and TMRx_ers = 0	TMRx_ers = 1	
Reserved	11	xxx	Reserved				

Note 1: If ON = 0 then an edge is required to restart the timer after ON = 1.

2: When TMRx = PRx then the next clock clears ON and stops TMRx at 00h.

3: When TMRx = PRx then the next clock stops TMRx at 00h but does not clear ON.

REGISTER 21-3: CCPxCAP: CAPTURE INPUT SELECTION MULTIPLEXER REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x
—	—	—	—	—	—	CTS<1:0>	
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 7-2 **Unimplemented:** Read as '0'

bit 1-0 **CTS<1:0>:** Capture Trigger Input Selection bits

CTS<1:0>	**Connection**	
	CCP1	**CCP2**
11	IOC_Interrupt	
10	CMP2_output	
01	CMP1_output	
00	Pin selected by CCP1PPS	Pin selected by CCP2PPS

REGISTER 21-4: CCPRxL: CCPx REGISTER LOW BYTE

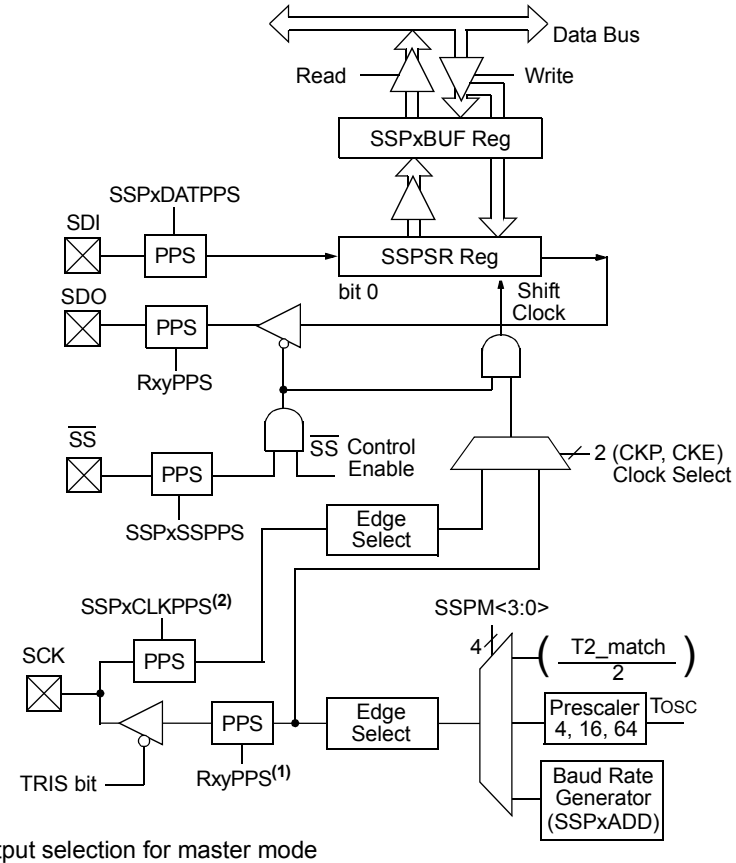
R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRx<7:0>							
bit 7							bit 0

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 7-0

MODE = Capture Mode:
CCPRxL<7:0>: LSB of captured TMR1 value
MODE = Compare Mode:
CCPRxL<7:0>: LSB compared to TMR1 value
MODE = PWM Mode && FMT = 0:
CCPRxL<7:0>: CCPW<7:0> – Pulse-Width LS 8 bits
MODE = PWM Mode && FMT = 1:
CCPRxL<7:6>: CCPW<1:0> – Pulse-Width LS 2 bits
CCPRxL<5:0>: Not used

FIGURE 26-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note 1: Output selection for master mode

2: Input selection for slave mode

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 26-2 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

REGISTER 26-7: SSPxCON1: MSSPx CONTROL REGISTER 1 (I²C MASTER MODE)

R/W-0	R/W/HC-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	HC = Bit is cleared by hardware	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **WCOL:** Write Collision Detect bit

In Master Transmit mode:
1 = A write to the SSPxBUF register was attempted while the I²C conditions were not valid for a transmission to be started (must be cleared in software)
0 = No collision

In Slave Transmit mode:
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

In Receive mode (Master or Slave modes):
This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)
0 = No overflow

In Transmit mode:
This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCKx Release Control bit

In Slave mode:
1 = Releases clock
0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:
Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)

1111 = I²C Slave mode: 10-bit address with Start and Stop bit interrupts enabled
1110 = I²C Slave mode: 7-bit address with Start and Stop bit interrupts enabled
1011 = I²C Firmware Controlled Master mode (slave Idle)
1000 = I²C Master mode: Clock = FOSC/(4 * (SSPxADD + 1))
0111 = I²C Slave mode: 10-bit address(3,4)
0110 = I²C Slave mode: 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.
2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

TABLE 27-2: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDxCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	395
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	170
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	182
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	174
IPR3	RC2IP	TX2IP	RC1IP	TX1IP	BCL2IP	SSP2IP	BCL1IP	SSP1IP	190
RCxREG	EUSARTx Receive Register								399*
RCxSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	394
RxyPPS	—	—	—	RxyPPS<4:0>					218
RXxPPS	—	—	—	RXPPS<4:0>					216
SPxBRGH	EUSARTx Baud Rate Generator, High Byte								404*
SPxBRGL	EUSARTx Baud Rate Generator, Low Byte								404*
TXxSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	393

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous reception.
* Page provides register information.

TABLE 31-1: ADC CLOCK PERIOD (T_{AD}) VS. DEVICE OPERATING FREQUENCIES(1,4)

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})						
ADC Clock Source	ADCS<5:0>	64 MHz	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
F_{OSC}/2	000000	31.25 ns(2)	62.5 ns(2)	100 ns(2)	125 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
F_{OSC}/4	000001	62.5 ns(2)	125 ns(2)	200 ns(2)	250 ns(2)	500 ns(2)	1.0 μs	4.0 μs
F_{OSC}/6	000010	125 ns(2)	187.5 ns(2)	300 ns(2)	375 ns(2)	750 ns(2)	1.5 μs	6.0 μs
F_{OSC}/8	000011	187.5 ns(2)	250 ns(2)	400 ns(2)	500 ns(2)	1.0 μs	2.0 μs	8.0 μs(3)
...	...	...	...	...	...	...	...	...
F_{OSC}/16	000100	250 ns(2)	500 ns(2)	800 ns(2)	1.0 μs	2.0 μs	4.0 μs	16.0 μs(3)
...	...	...	...	...	...	...	...	...
F_{OSC}/128	111111	2.0 μs	4.0 μs	6.4 μs	8.0 μs	16.0 μs(3)	32.0 μs(2)	128.0 μs(2)
FRC	ADCS(ADCON0<4>) = 1	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs

Legend: Shaded cells are outside of recommended range.

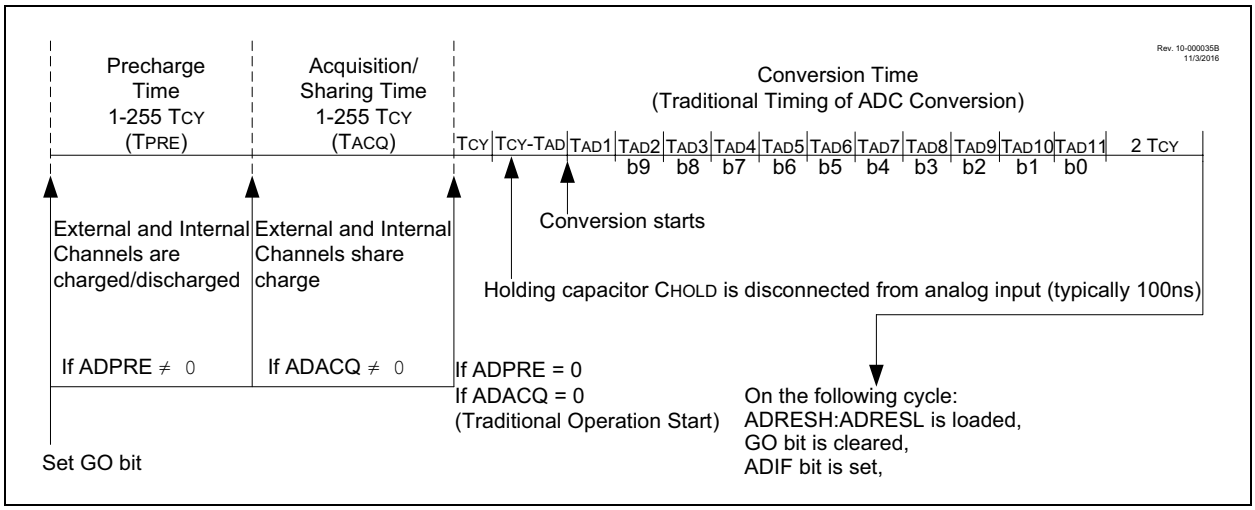
Note 1: See T_{AD} parameter for FRC source typical T_{AD} value.

2: These values violate the required T_{AD} time.

3: Outside the recommended T_{AD} time.

4: The ADC clock period (T_{AD}) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock F_{OSC}. However, the FRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 31-2: ANALOG-TO-DIGITAL CONVERSION T_{AD} CYCLES



31.4.2 PRECHARGE CONTROL

The Precharge stage is an optional period of time that brings the external channel and internal sample and hold capacitor to known voltage levels. Precharge is enabled by writing a non-zero value to the ADPRE register. This stage is initiated when an ADC conversion begins, either from setting the ADGO bit, a special event trigger, or a conversion restart from the computation functionality. If the ADPRE register is cleared when an ADC conversion begins, this stage is skipped.

During the precharge time, CHOLD is disconnected from the outer portion of the sample path that leads to the external capacitive sensor and is connected to either VDD or VSS, depending on the value of the ADPPOL bit of ADCON1. At the same time, the port pin logic of the selected analog channel is overridden to drive a digital high or low out, in order to precharge the outer portion of the ADC's sample path, which includes the external sensor. The output polarity of this override is also determined by the ADPPOL bit of ADCON1. The amount of time that this charging needs is controlled by the ADPRE register.

Note:	The external charging overrides the TRIS setting of the respective I/O pin. If there is a device attached to this pin, Precharge should not be used.

31.4.3 ACQUISITION CONTROL

The Acquisition stage is an optional time for the voltage on the internal sample and hold capacitor to charge or discharge from the selected analog channel.This acquisition time is controlled by the ADACQ register. If ADPRE = 0, acquisition starts at the beginning of conversion. When ADPRE = 1, the acquisition stage begins when precharge ends.

At the start of the acquisition stage, the port pin logic of the selected analog channel is overridden to turn off the digital high/low output drivers so they do not affect the final result of the charge averaging. Also, the selected ADC channel is connected to CHOLD. This allows charge averaging to proceed between the precharged channel and the CHOLD capacitor.

Note:	When ADPRE! = 0, acquisition time cannot be '0'. In this case, setting ADACQ to '0' will set a maximum acquisition time (256 ADC clock cycles). When precharge is disabled, setting ADACQ to '0' will disable hardware acquisition time control.

31.4.4 GUARD RING OUTPUTS

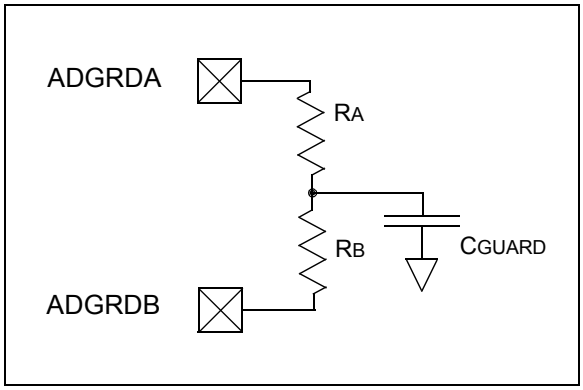
Figure 31-8 shows a typical guard ring circuit. CGUARD represents the capacitance of the guard ring trace placed on the PCB board. The user selects values for RA and RB that will create a voltage profile on CGUARD, which will match the selected acquisition channel.

The purpose of the guard ring is to generate a signal in phase with the CVD sensing signal to minimize the effects of the parasitic capacitance on sensing electrodes. It also can be used as a mutual drive for mutual capacitive sensing. For more information about active guard and mutual drive, see Application Note AN1478, "*mTouch™ Sensing Solution Acquisition Methods Capacitive Voltage Divider*" (DS01478).

The ADC has two guard ring drive outputs, ADGRDA and ADGRDB. These outputs can be routed through PPS controls to I/O pins (see **Section 17.0 "Peripheral Pin Select (PPS) Module"** for details) and the polarity of these outputs are controlled by the ADGPOL and ADIPEN bits of ADCON1.

At the start of the first precharge stage, both outputs are set to match the ADGPOL bit of ADCON1. Once the acquisition stage begins, ADGRDA changes polarity, while ADGRDB remains unchanged. When performing a double sample conversion, setting the ADIPEN bit of ADCON1 causes both guard ring outputs to transition to the opposite polarity of ADGPOL at the start of the second precharge stage, and ADGRDA toggles again for the second acquisition. For more information on the timing of the guard ring output, refer to Figure 31-8 and Figure 31-9.

FIGURE 31-8: GUARD RING CIRCUIT



CPFSGT **Compare f with W, skip if f > W**

Syntax: CPFSGT f {,a}

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

Operation: (f) – (W),
skip if (f) > (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	010a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction.
If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 35.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE      CPFSGT REG, 0
NGREATER  :
GREATER   :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG > W;
PC = Address (GREATER)
If REG ≤ W;
PC = Address (NGREATER)

CPFSLT **Compare f with W, skip if f < W**

Syntax: CPFSLT f {,a}

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

Operation: (f) – (W),
skip if (f) < (W)
(unsigned comparison)

Status Affected: None

Encoding:

0110	000a	ffff	ffff

Description: Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.
If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    CPFSLT REG, 1
NLESS   :
LESS    :
```
Before Instruction
PC = Address (HERE)
W = ?
After Instruction
If REG < W;
PC = Address (LESS)
If REG ≥ W;
PC = Address (NLESS)

IORLW **Inclusive OR literal with W**

Syntax: IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. k → W

Status Affected: N, Z

Encoding:

0000	1001	kkkk	kkkk

Description: The contents of W are ORed 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

Example: IORLW 35h

Before Instruction
W = 9Ah
After Instruction
W = BFh

IORWF **Inclusive OR W with f**

Syntax: IORWF f {,d {,a}}

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

Operation: (W) .OR. (f) → 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 is selected.
If 'a' is '1', the BSR is used to select the GPR bank.
If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 35.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: IORWF RESULT, 0, 1

Before Instruction
RESULT = 13h
W = 91h
After Instruction
RESULT = 13h
W = 93h

MOVFF **Move f to f**

Syntax: MOVFF f_s,f_d

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)	1100	ffff	ffff	$ffff_s$
2nd word (destin.)	1111	ffff	ffff	$ffff_d$

Description: The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh.
Either source or destination can be W (a useful special situation).
MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).
The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVFF REG1, REG2

Before Instruction
REG1 = 33h
REG2 = 11h
After Instruction
REG1 = 33h
REG2 = 33h

MOVLB **Move literal to low nibble in BSR**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow BSR$

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk

Description: The 8-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of k_7:k_4.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

TABLE 37-13: ANALOG-TO-DIGITAL CONVERTER (ADC) ACCURACY SPECIFICATIONS(1,2):

Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C, TAD = 1μs							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF-= 0V
AD03	EDL	Differential Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF-= 0V
AD04	EOFF	Offset Error	—	0.5	±2.0	LSb	ADCREF+ = 3.0V, ADCREF-= 0V
AD05	EGN	Gain Error	—	±0.2	±1.0	LSb	ADCREF+ = 3.0V, ADCREF-= 0V
AD06	VADREF	ADC Reference Voltage (ADREF+ - ADREF-)	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	ADREF-	—	ADREF+	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	10	—	kΩ	
AD09	RVREF	ADC Voltage Reference Ladder Impedance	—	50	—	kΩ	**Note 3**

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Total Absolute Error is the sum of the offset, gain and integral non-linearity (INL) errors.

2: The ADC conversion result never decreases with an increase in the input and has no missing codes.

3: This is the impedance seen by the VREF pads when the external reference pads are selected.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-1639-5