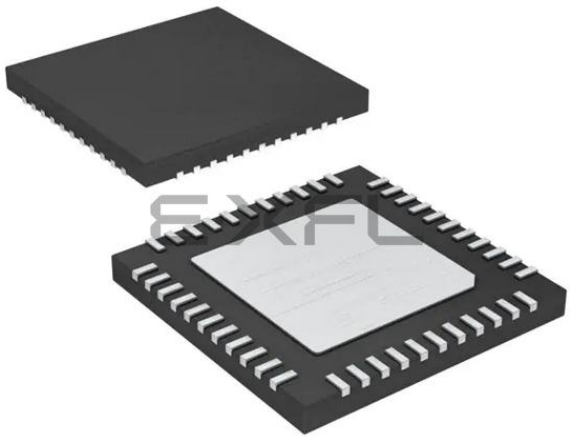


Welcome to E-XFL.COM

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, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf47k42-i-ml

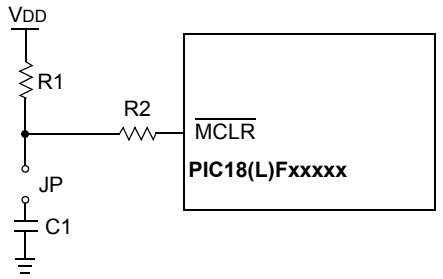
2.3 Master Clear (MCLR) Pin

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 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 ICSPCLK and ICSPDAT 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 ICSPCLK and ICSPDAT 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., ICSPCLK/ICSPDAT 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 43.0 "Development Support"**.

6.4 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-out Reset (LPBOR) provides an additional BOR circuit for low power operation. Refer to Figure 6-2 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset.

6.4.1 ENABLING LPBOR

The LPBOR is controlled by the LPBOREN bit of Configuration Word 2L. When the device is erased, the LPBOR module defaults to disabled.

6.4.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic BOR signal, which goes to the PCON0 register and to the power control block.

6.5 MCLR

The MCLR is an optional external input that can reset the device. The MCLR function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 6-2). The RMCLR bit in the PCON0 register will be set to '0' if a MCLR Reset has occurred.

TABLE 6-2: MCLR CONFIGURATION

MCLRE	LVP	MCLR
x	1	Enabled
1	0	Enabled
0	0	Disabled

6.5.1 MCLR ENABLED

When MCLR is enabled and the pin is held low, the device is held in Reset. The MCLR pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the MCLR Reset path. The filter will detect and ignore small pulses.

Note:	An internal Reset event (RESET instruction, BOR, WWDT, POR stack), does not drive the MCLR pin low.

6.5.2 MCLR DISABLED

When MCLR is disabled, the MCLR pin becomes input-only and pin functions such as internal weak pull-ups are under software control. See **Section 16.1 "I/O Priorities"** for more information.

6.6 Windowed Watchdog Timer (WWDT) Reset

The Windowed Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period or window set. The TO and PD bits in the STATUS register and the RWDT bit in the PCON0 register are changed to indicate a WWDT Reset. The WDTWV bit in the PCON0 register indicates if the WDT Reset has occurred due to a time out or a window violation. See **Section 11.0 "Windowed Watchdog Timer (WWDT)"** for more information.

6.7 RESET Instruction

A RESET instruction will cause a device Reset. The RI bit in the PCON0 register will be set to '0'. See Table 6-3 for default conditions after a RESET instruction has occurred.

6.8 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON0 register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 4.2.5 "Return Address Stack"** for more information.

6.9 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR occurred.

6.10 Power-up Timer (PWRT)

The Power-up Timer provides a selected time-out duration on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is selected by setting the PWRTS<1:0> Configuration bits, appropriately.

The Power-up Timer starts after the release of the POR and BOR/LPBOR if enabled, as shown in Figure 6-1.

REGISTER 9-14: PIE0: PERIPHERAL INTERRUPT ENABLE 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
IOCIE	CRCIE	SCANIE	NVMIE	CSWIE	OSFIE	HLVDIE	SWIE
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 **IOCIE:** Interrupt-on-Change Enable bit
1 = Enabled
0 = Disabled

bit 6 **CRCIE:** CRC Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **SCANIE:** Memory Scanner Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **NVMIE:** NVM Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **CSWIE:** Clock Switch Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **HLVDIE:** HLVD Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **SWIE:** Software Interrupt Enable bit
1 = Enabled
0 = Disabled

REGISTER 9-32: IPR7: PERIPHERAL INTERRUPT PRIORITY REGISTER 7

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
—	—	INT2IP	CLC2IP	CWG2IP	—	CCP2IP	TMR4IP
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 7-6 **Unimplemented:** Read as '0'

bit 5 **INT2IP:** External Interrupt 2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **CLC2IP:** CLC2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CWG2IP:** CWG2 Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 1 **CCP2IP:** CRC Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR4IP:** TMR4 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 9-33: IPR8: PERIPHERAL INTERRUPT PRIORITY REGISTER 8

R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0	U-0	U-0
TMR5GIP	TMR5IP	—	—	—	—	—	—
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 **TMR5GIP:** TMR5 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **TMR5IP:** TMR5 Interrupt Priority bit
1 = High priority
0 = Low priority

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

13.1.3 READING THE PROGRAM FLASH MEMORY

The TBLRD instruction retrieves data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

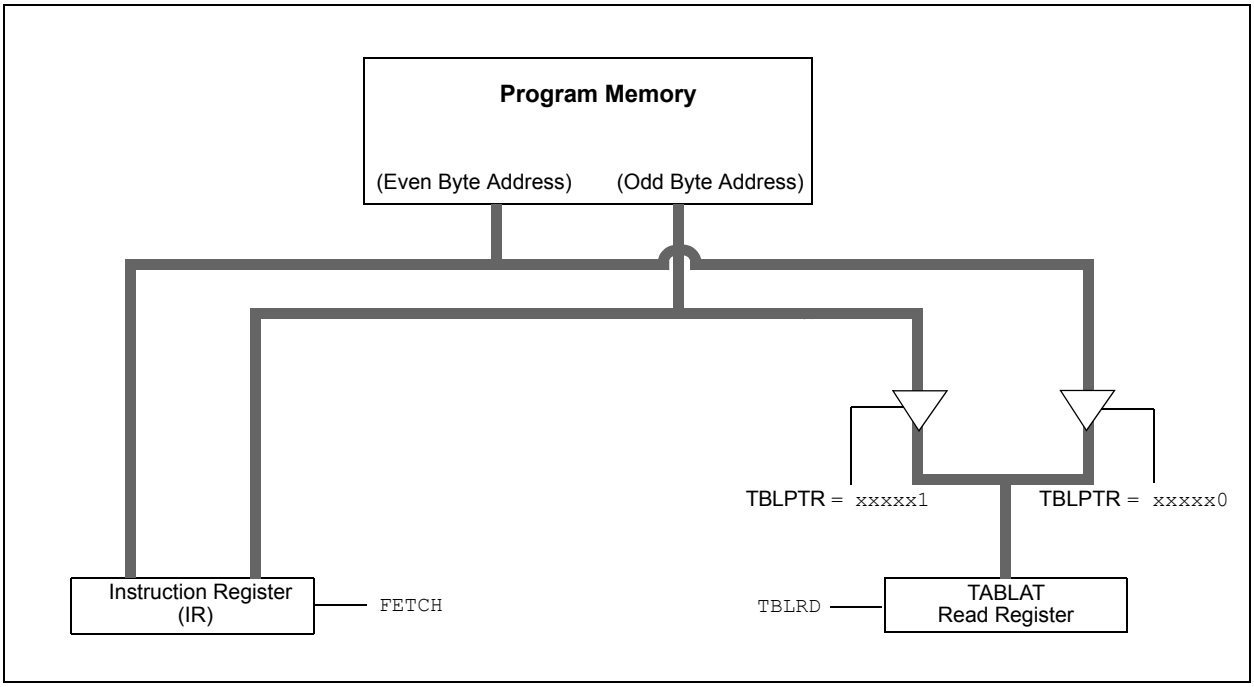
TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation.

The CPU operation is suspended during the read, and it resumes immediately after. From the user point of view, TABLAT is valid in the next instruction cycle.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word.

Figure 13-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 13-4: READS FROM PROGRAM FLASH MEMORY



EXAMPLE 13-1: READING A PROGRAM FLASH MEMORY WORD

```
            BCF       NVMCON1, REG0          ; point to Program Flash Memory
            BSF       NVMCON1, REG1          ; access Program Flash Memory
            MOVLW     CODE_ADDR_UPPER        ; Load TBLPTR with the base
            MOVWF     TBLPTRU                ; address of the word
            MOVLW     CODE_ADDR_HIGH
            MOVWF     TBLPTRH
            MOVLW     CODE_ADDR_LOW
            MOVWF     TBLPTRL
READ_WORD
            TBLRD*+                          ; read into TABLAT and increment
            MOVF      TABLAT, W              ; get data
            MOVWF     WORD_EVEN
            TBLRD*+                          ; read into TABLAT and increment
            MOVFW     TABLAT, W              ; get data
            MOVF      WORD_ODD
```

TABLE 17-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	283
INT0PPS	—	—	—	INT0PPS<4:0>					277
INT1PPS	—	—	—	INT1PPS<4:0>					277
INT2PPS	—	—	—	INT2PPS<4:0>					277
T0CKIPPS	—	—	—	T0CKIPPS<4:0>					277
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					277
T1GPPS	—	—	—	T1GPPS<4:0>					277
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					277
T3GPPS	—	—	—	T3GPPS<4:0>					277
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					277
T5GPPS	—	—	—	T5GPPS<4:0>					277
T2INPPS	—	—	—	T2INPPS<4:0>					277
T4INPPS	—	—	—	T4INPPS<4:0>					277
T6INPPS	—	—	—	T6INPPS<4:0>					277
CCP1PPS	—	—	—	CCP1PPS<4:0>					277
CCP2PPS	—	—	—	CCP2PPS<4:0>					277
CCP3PPS	—	—	—	CCP3PPS<4:0>					277
CCP4PPS	—	—	—	CCP4PPS<4:0>					277
SMT1WINPPS	—	—	—	SMT1WINPPS<4:0>					277
SMT1SIGPPS	—	—	—	SMT1SIGPPS<4:0>					277
CWG1PPS	—	—	—	CWG1PPS<4:0>					277
CWG2PPS	—	—	—	CWG2PPS<4:0>					277
CWG3PPS	—	—	—	CWG3PPS<4:0>					277
MD1CARLPPS	—	—	—	MDCARLPPS<4:0>					277
MD1CARHPPS	—	—	—	MDCARHPPS<4:0>					277
MD1SRCPPS	—	—	—	MDSRCPPS<4:0>					277
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					277
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					277
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					277
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					277
ADACTPPS	—	—	—	ADACTPPS<4:0>					277
SPI1SCKPPS	—	—	—	SPI1SCKPPS<4:0>					277
SPI1SDIPPS	—	—	—	SPI1SDIPPS<4:0>					277
SPI1SSPPS	—	—	—	SPI1SSPPS<4:0>					277
I2C1SCLPPS	—	—	—	I2C1SCLPPS<4:0>					277
I2C1SDAPPS	—	—	—	I2C1SDAPPS<4:0>					277
I2C2SCLPPS	—	—	—	I2C2SCLPPS<4:0>					277
I2C2SDAPPS	—	—	—	I2C2SDAPPS<4:0>					277
U1RXPPS	—	—	—	U1RXPPS<4:0>					277
U1CTSPPS	—	—	—	U1CTSPPS<4:0>					277
U2RXPPS	—	—	—	U2RXPPS<4:0>					277
U2CTSPPS	—	—	—	U2CTPPS<4:0>					277
RxyPPS	—	—	—	RxyPPS<4:0>					280

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.

REGISTER 19-5: PMD4: PMD CONTROL REGISTER 4

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	U-0
CWG3MD	CWG2MD	CWG1MD	—	—	—	—	—
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	q = Value depends on condition

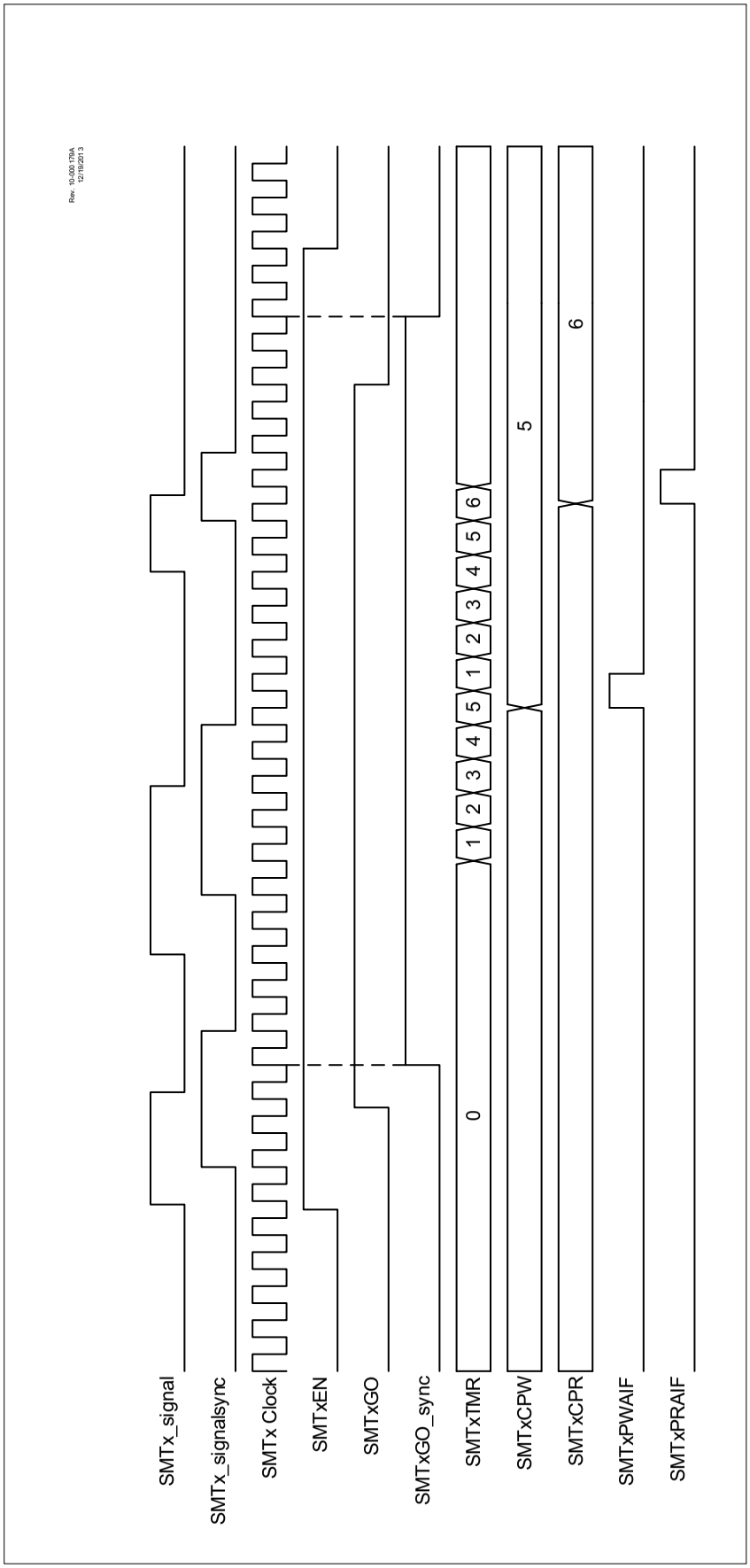
bit 7 **CWG3MD:** Disable CWG3 Module bit
1 = CWG3 module disabled
0 = CWG3 module enabled

bit 6 **CWG2MD:** Disable CWG2 Module bit
1 = CWG2 module disabled
0 = CWG2 module enabled

bit 5 **CWG1MD:** Disable CWG1 Module bit
1 = CWG1 module disabled
0 = CWG1 module enabled

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

FIGURE 25-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM



27.2 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR5 register will be set when either edge detector is triggered and its associated enable bit is set. The INTP enables rising edge interrupts and the INTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- CLCxIE bit of the respective PIE register
- INTP bit of the CLCxCON register (for a rising edge detection)
- INTN bit of the CLCxCON register (for a falling edge detection)
- GIE bits of the INTCON0 register

The CLCxIF bit of the respective PIR register, must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

27.3 Output Mirror Copies

Mirror copies of all CON output bits are contained in the CLCxDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the OUT bits in the individual CLCxCON registers.

27.4 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

27.5 Operation During Sleep

The CLC module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

The HFINTOSC remains active during Sleep when the CLC module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLC input source, when the CLC is enabled, the CPU will go idle during Sleep, but the CLC will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

27.6 CLCx Setup Steps

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the EN bit.
- Select desired inputs using CLCxSEL0 through CLCxSEL3 registers (See Table 27-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the GyPOL bits of the CLCxPOL register.
- Select the desired logic function with the MODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the POL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device pin, set the desired pin PPS control register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the INTP bit in the CLCxCON register for rising event.
  - Set the INTN bit in the CLCxCON register for falling event.
  - Set the CLCxIE bit of the respective PIE register.
  - Set the GIE bits of the INTCON0 register.
- Enable the CLCx by setting the EN bit of the CLCxCON register.

31.8 Polarity

Receive and transmit polarity is user selectable and affects all modes of operation.

The idle level is programmable with the polarity control bits in the UxCON2 register. The control bits default to '0', which select a high idle level. The low level Idle state is selected by setting the control bit to '1'. TXPOL controls the TX idle level. RXPOL controls the RX idle level.

REGISTER 31-3: UxCON2: UART CONTROL REGISTER 2

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
RUNOVF	RXPOL	STP<1:0>		C0EN	TXPOL	FLO<1:0>	
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	

bit 7 **RUNOVF:** Run During Overflow Control bit
1 = RX input shifter continues to synchronize with Start bits after overflow condition
0 = RX input shifter stops all activity on receiver overflow condition

bit 6 **RXPOL**: Receive Polarity Control bit
1 = Invert RX polarity, Idle state is low
0 = RX polarity is not inverted, Idle state is high

bit 5-4 **STP<1:0>:** Stop Bit Mode Control bits(1)
11 = Transmit 2 Stop bits, receiver verifies first Stop bit
10 = Transmit 2 Stop bits, receiver verifies first and second Stop bits
01 = Transmit 1.5 Stop bits, receiver verifies first Stop bit
00 = Transmit 1 Stop bit, receiver verifies first Stop bit

bit 3 **C0EN**: Checksum Mode Select bit(2)
LIN mode:
1 = Checksum Mode 1, enhanced LIN checksum includes PID in sum
0 = Checksum Mode 0, legacy LIN checksum does not include PID in sum
Other modes:
1 = Add all TX and RX characters
0 = Checksums disabled

bit 2 **TXPOL:** Transmit Polarity Control bit
1 = Output data is inverted, TX output is low in Idle state
0 = Output data is not inverted, TX output is high in Idle state

bit 1-0 **FLO<1:0>:** Handshake Flow Control bits
11 = Reserved
10 = RTS/CTS and TXDE Hardware flow control
01 = XON/XOFF Software flow control
00 = Flow control is off

Note 1: All modes transmit selected number of Stop bits. Only DMX and DALI receivers verify selected number of Stop bits and all others verify only the first Stop bit.
2: UART1 only.

32.7 SPI Operation in Sleep Mode

SPI master mode will operate in Sleep, provided the clock source selected by SPIxCLK is active in Sleep mode. FIFOs will operate as they would when the part is awake. When TXR = 1, the TXFIFO will need to contain data in order for transfers to take place in Sleep. All interrupts will still set the interrupt flags in Sleep but only enabled interrupts will wake the device from Sleep.

SPI Slave mode will operate in Sleep, because the clock is provided by an external master device. FIFOs will still operate and interrupts will set interrupt flags, and enabled interrupts will wake the device from Sleep.

32.8 SPI Interrupts

There are three top level SPI interrupts in the PIRx register:

- SPI Transmit
- SPI Receive
- SPI Module status

The status interrupts are enabled at the module level in the SPIxINTE register. Only enabled status interrupts will cause the single top level SPIxIF flag to be set.

32.8.1 SPI RECEIVER DATA INTERRUPT

The SPI Receiver Data Interrupt is set when RXFIFO contains data, and is cleared when the RXFIFO is empty. The interrupt flag SPI1RXIF is located in PIRx and the interrupt enable SPI1RXIE is located in PIEx. This interrupt flag is read-only.

32.8.2 SPI TRANSMITTER DATA INTERRUPT

The SPI Transmitter Data Interrupt is set when TXFIFO is not full, and is cleared when the TXFIFO is full. The interrupt flag SPI1TXIF is located in PIRx and the interrupt enable SPI1TXIE is located in PIEx. The interrupt flag is read-only.

32.8.3 SPI MODULE STATUS INTERRUPTS

The SPIxIF flag in the respective PIR register is set when any of the individual status flags in SPIxINTF and their respective SPIxINTE bits are set. In order for the setting of any specific interrupt flag to interrupt normal program flow both the SPIxIE bit as well as the specific bit in SPIxINTE associated with that interrupt must be set.

The Status Interrupts are:

- Shift Register Empty Interrupt
- Transfer Counter is Zero Interrupt
- Start of Slave Select Interrupt
- End of Slave Select Interrupt
- Receiver Overflow Interrupt
- Transmitter Underflow Interrupt

33.4.3.3 Slave operation in 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '11110A9A80'. A9 and A8 are the two MSb of the 10-bit address. The first byte is compared with the value in I2CxADR1 and I2CxADR3 registers. After the high byte is acknowledged, the low address byte is clocked in and all eight bits are compared to the low address value in the I2CxADR0 and I2CxADR2 registers. A high and low address match as a write request is required at the start of all 10-bit addressing communication. To initiate a read, the Master needs to issue a Restart once the slave is addressed and clock in the high address with the R/W bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. The SMA (slave active) bit is set only when both the high and low address bytes match.

Note:	All seven bits of the received high address are compared to the values in the I2CxADR1 and I2CxADR3 registers. The five-bit '11110' high address format is not enforced by module hardware. It is up to the user to configure these bits correctly.

33.4.3.4 Slave Reception (10-bit Addressing Mode)

This section describes the sequence of events for the I^2C module configured as an I^2C slave in 10-bit Addressing mode and is receiving data. Figure 33-11 is used as a visual reference for this description.

1. Master asserts Start condition (can also be a restart) on the bus. Start condition Interrupt Flag (SCIF) in I2CxPIR register is set. If Start condition interrupt is enabled (SCIE bit is set), generic interrupt I2CxIF is set.
2. Master transmits high address byte with R/W = 0.
3. The received high address is compared with the values in I2CxADR1 and I2CxADR3 registers.
4. If high address matches; R/W is copied to R/W bit, D/A bit is cleared, high address data is copied to I2CxADB1. If the address does not match; module becomes idle.
5. If Address hold interrupt is enabled (ADRIE = 1), CSTR is set. I2CxIF is set.
6. Slave software can read high address from I2CxADB1 and set/clear ACKDT before releasing SCL.
7. ACKDT value is copied out to SDA for ACK pulse. SCL line is released by clearing CSTR.
8. Master sends ninth SCL pulse for ACK
9. Slave can force a NACK at this point due to previous error not being cleared. E.g. Receive buffer overflow or transmit buffer underflow errors. In these cases the Slave hardware forces a NACK and the module becomes idle.
10. Master transmits low address data byte
11. If the low address matches; SMA is set, ADRIF is set, R/W is copied to R/W bit, D/A bit is cleared, low address data is copied to I2CxADB0, and ACTDT is copied to SDA. If the address does not match; module becomes idle.
12. If address hold interrupt is enabled, the CSTR bit is set as mentioned in step 6. Slave software can read low address byte from I2CxADB0 register and change ACKDT value before releasing SCL.
13. Master sends ninth SCL pulse for ACK.
14. If the Acknowledge interrupt and hold is enabled (ACKTIE = 1), CSTR is set, I2CxIF is set.
15. Slave software can read address from I2CxADB0 and I2CxADB1 registers and change the value of ACKDT before releasing SCL by clearing CSTR.
16. Master sends first seven SCL pulses of the data byte or a Stop condition (in the case of NACK).
17. If Stop condition; PCIF in I2CxPIR register is set, module becomes idle.
18. If the receive buffer is full from the previous transaction i.e. RXBF = 1 (I2CxRXIF = 1), CSTR is set. Slave software must read data out of I2CxRXB to resume communication.
19. Master sends eighth SCL pulse of the data byte. D/A bit is set, WRIF is set. I2CxRXB is loaded with new data, RXBF bit is set.
20. If Data write interrupt and hold is enabled (WRIE = 1), CSTR is set, I2CxIF is set. Slave software can read data from I2CxRXB and set/clear ACKDT before releasing SCL by clearing CSTR.
21. If I2CxCNT = 0, the ACKCNT value is output to the SDA; else, the ACKDT value is used and the value of I2CxCNT is decremented.
22. Master sends SCL pulse for ACK.
23. If I2CxCNT = 0, CNTIF is set.
24. If the response was a NACK; NACKIF is set, module becomes idle.
25. If ACKTIE = 1, CSTR is set, I2CxIF is set. Slave software can read data from I2CxRXB clearing RXBF; before releasing SCL by clearing CSTR
26. Go to step 16.

33.5.11 MASTER TRANSMISSION IN 10-BIT ADDRESSING MODE

This section describes the sequence of events for the I^2C module configured as an I^2C master in 10-bit Addressing mode and is transmitting data. Figure 33-21 is used as a visual reference for this description

1. If ABD = 0; i.e., Address buffers are enabled

Master software loads number of bytes to be transmitted in one sequence in I2CxCNT, high address byte of slave address in I2CxADB1 with R/W = 0, low address byte in I2CxADB0 and the first byte of data in I2CxTXB. Master software has to set the Start (S) bit to initiate communication.

If ABD = 1; i.e., Address buffers are disabled

Master software loads the number of bytes to be transmitted in one sequence in I2CxCNT and the high address byte of the slave address with R/W = 0 into the I2CxTXB register. Writing to the I2CxTXB will assert the start condition on the bus and sets the S bit. Software writes to the S bit are ignored in this case.

2. Master hardware waits for BFRE bit to be set; then shifts out the start and high address and waits for acknowledge.
3. If NACK, master hardware sends Stop.
4. If ABD = 0; i.e., Address buffer are enabled

If ACK, master hardware sends the low address byte from I2CxADB0.

If ABD = 1; i.e., Address buffer are disabled

If ACK, master hardware sets TXIF and MDR bits and the software has to write the low address byte into I2CxTXB. Writing to I2CxTXB sends the low address on the bus.

5. If TXBE = 1 and I2CxCNT! = 0, I2CxTXIF and MDR bits are set. Clock is stretched on 8th falling SCL edge until master software writes next data byte to I2CxTXB.
6. Master hardware sends ninth SCL pulse for ACK from slave and loads the shift register from I2CxTXB. I2CxCNT is decremented.
7. If slave sends a NACK, master hardware sends Stop and ends transmission.
8. If slave sends an ACK, master hardware outputs data in the shift register on SDA. I2CxCNT value is checked on the 8th falling SCL edge. If I2CxCNT = 0; master hardware sends 9th SCL pulse for ACK and CNTIF is set.
9. If I2CxCNT! = 0; go to step 5.

38.7 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 44-17 and Table 44-19 for more details.

38.8 Analog Input Connection Considerations

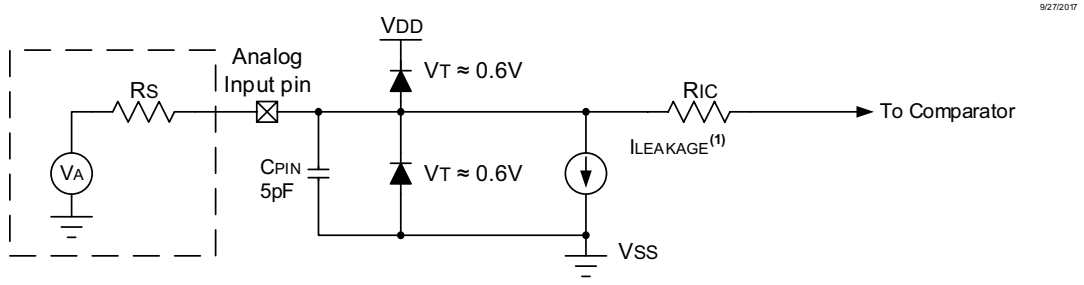
A simplified circuit for an analog input is shown in Figure 38-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

The maximum source impedance for analog sources is mentioned in Parameter AD08 in Table 44-15. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a '0'. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 38-3: ANALOG INPUT MODEL



Legend: C_{PIN} = Input Capacitance
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_S = Source Impedance
V_A = Analog Voltage
V_T = Threshold Voltage

Note 1: See **Section 44.0 "Electrical Specifications"**.

MOVLW **Move literal to W**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

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 5Ah`

After Instruction

W = 5Ah

MOVWF **Move W to f**

Syntax: MOVWF f {,a}

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

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 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 41.2.3 "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 register 'f'

Example: `MOVWF REG, 0`

Before Instruction

W = 4Fh
REG = FFh

After Instruction

W = 4Fh
REG = 4Fh

RETURN **Return from Subroutine**

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
if s = 1
(WS) → W,
(STATUSS) → Status,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	0000	0001	001s

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers, WS, STATUSS and BSRS, are loaded into their corresponding registers, W, Status and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: RETURN

After Instruction:
PC = TOS

RLCF **Rotate Left f through Carry**

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

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

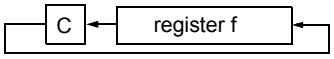
Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

Encoding:

0011	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the CARRY flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored 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 ≤ 95 (5Fh). See **Section 41.2.3 "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: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
39DAh	OSCCON2	—	COSC			CDIV				105
39D9h	OSCCON1	—	NOSC			NDIV				104
39D8h	CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE			177
39D7h - 39D2h	—	Unimplemented								
39D1h	VREGCON(1)	—	—	—	—	—	—	VREGPM	—	176
39D0h	BORCON	SBOREN	—	—	—	—	—	—	BORRDY	85
39CFh - 39C8h	—	Unimplemented								
39C7h	PMD7	—	—	—	—	—	—	DMA2MD	DMA1MD	297
39C6h	PMD6	—	—	SMT1MD	CLC4MD	CLC3MD	CLC2MD	CLC1MD	DSMMD	296
39C5h	PMD5	—	—	U2MD	U1MD	—	SPI1MD	I2C2MD	I2C1MD	295
39C4h	PMD4	CWG3MD	CWG2MD	CWG1MD	—	—	—	—	—	294
39C3h	PMD3	PWM8MD	PWM7MD	PWM6MD	PWM5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	293
39C2h	PMD2	—	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	ZCDMD	292
39C1h	PMD1	NCO1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	291
39C0h	PMD0	SYSCMD	FVRMD	HLVDMD	CRCMD	SCANMD	NVMMD	CLKRMD	IOCMD	290
39BFh - 39ABh	—	Unimplemented								
39AAh	PIR10	—	—	—	—	—	—	CLC4IF	CCP4IF	146
39A9h	PIR9	—	—	—	—	CLC3IF	CWG3IF	CCP3IF	TMR6IF	145
39A8h	PIR8	TMR5GIF	TMR5IF	—	—	—	—	—	—	145
39A7h	PIR7	—	—	INT2IF	CLC2IF	CWG2IF	—	CCP2IF	TMR4IF	144
39A6h	PIR6	TMR3GIF	TMR3IF	U2IF	U2EIF	U2TXIF	U2RXIF	I2C2EIF	I2C2IF	143
39A5h	PIR5	I2C2TXIF	I2C2RXIF	DMA2AIF	DMA2ORIF	DMA2DCN-TIF	DMA2SCN-TIF	C2IF	INT1IF	142
39A4h	PIR4	CLC1IF	CWG1IF	NCO1IF	—	CCP1IF	TMR2IF	TMR1GIF	TMR1IF	141
39A3h	PIR3	TMR0IF	U1IF	U1EIF	U1TXIF	U1RXIF	I2C1EIF	I2C1IF	I2C1TXIF	140
39A2h	PIR2	I2C1RXIF	SPI1IF	SPI1TXIF	SPI1RXIF	DMA1AIF	DMA1ORIF	DMA1DCN-TIF	DMA1SCNTIF	138
39A1h	PIR1	SMT1PWAIF	SMT1PRAIF	SMT1IF	C1IF	ADTIF	ADIF	ZCDIF	INT0IF	138
39A0h	PIR0	IOCIF	CRCIF	SCANIF	NVMIF	CSWIF	OSFIF	HLVDIF	SWIF	137
399Fh - 399Bh	—	Unimplemented								
399Ah	PIE10	—	—	—	—	—	—	CLC4IE	CCP4IE	156
3999h	PIE9	—	—	—	—	CLC3IE	CWG3IE	CCP3IE	TMR6IE	155
3998h	PIE8	TMR5GIE	TMR5IE	—	—	—	—	—	—	155
3997h	PIE7	—	—	INT2IE	CLC2IE	CWG2IE	—	CCP2IE	TMR4IE	154
3996h	PIE6	TMR3GIE	TMR3IE	U2IE	U2EIE	U2TXIE	U2RXIE	I2C2EIE	I2C2IE	153
3995h	PIE5	I2C2TXIE	I2C2RXIE	DMA2AIE	DMA2ORIE	DMA2DCN-TIE	DMA2SCN-TIE	C2IE	INT1IE	152
3994h	PIE4	CLC1IE	CWG1IE	NCO1IE	—	CCP1IE	TMR2IE	TMR1GIE	TMR1IE	151
3993h	PIE3	TMR0IE	U1IE	U1EIE	U1TXIE	U1RXIE	I2C1EIE	I2C1IE	I2C1TXIE	150
3992h	PIE2	I2C1RXIE	SPI1IE	SPI1TXIE	SPI1RXIE	DMA1AIE	DMA1ORIE	DMA1DCN-TIE	DMA1SCNTIE	149
3991h	PIE1	SMT1PWAIE	SMT1PRAIE	SMT1IE	C1IE	ADTIE	ADIE	ZCDIE	INT0IE	148
3990h	PIE0	IOCIE	CRCIE	SCANIE	NVMIE	CSWIE	OSFIE	HLVDIE	SWIE	147
398Fh - 398Bh	—	Unimplemented								
398Ah	IPR10	—	—	—	—	—	—	CLC4IP	CCP4IP	165

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

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

TABLE 44-4: POWER-DOWN CURRENT (I_{PD})(1,2)

PIC18LF26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated)					
PIC18F26/45/46/55/56K42				Standard Operating Conditions (unless otherwise stated) VREGPM = 1					
Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								V_{DD}	Note
D200	I_{PD}	I_{PD} Base	—	0.07	2	6	µA	3.0V	
D200 D200A	I_{PD}	I_{PD} Base	—	0.4	4	8	µA	3.0V	
			—	20	38	42	µA	3.0V	VREGPM = 0
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	0.9	3.2	7	µA	3.0V	
D201	I_{PD_WDT}	Low-Frequency Internal Oscillator/ WDT	—	1.1	3.2	9	µA	3.0V	
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	0.75	5	9	µA	3.0V	LP mode
D202	I_{PD_SOSC}	Secondary Oscillator (S_{OSC})	—	1.0	6.5	10	µA	3.0V	LP mode
D203	I_{PD_FVR}	FVR	—	45	74	75	µA	3.0V	FVRCON = 0x81 or 0x84
D203	I_{PD_FVR}	FVR	—	40	70	76	µA	3.0V	FVRCON = 0x81 or 0x84
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	14	18	µA	3.0V	
D204	I_{PD_BOR}	Brown-out Reset (BOR)	—	9.4	15	18	µA	3.0V	
D205	I_{PD_LPBOR}	Low-Power Brown-out Reset (LPBOR)	—	0.2	3	6	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.5	14.8	18	µA	3.0V	
D206	I_{PD_HLVD}	High/Low Voltage Detect (HLVD)	—	9.7	14.2	17	µA	3.0V	
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	2	6	µA	3.0V	ADC not converting (4)
D207	I_{PD_ADCA}	ADC - Non-Converting	—	0.1	4	8	µA	3.0V	ADC not converting (4)
D208	I_{PD_CMP}	Comparator	—	33	49	50	µA	3.0V	
D208	I_{PD_CMP}	Comparator	—	30	49	50	µA	3.0V	

† 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: The peripheral current is the sum of the base I_{DD} and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base I_{DD} or I_{PD} current from this limit. Max. values should be used when calculating total current consumption.

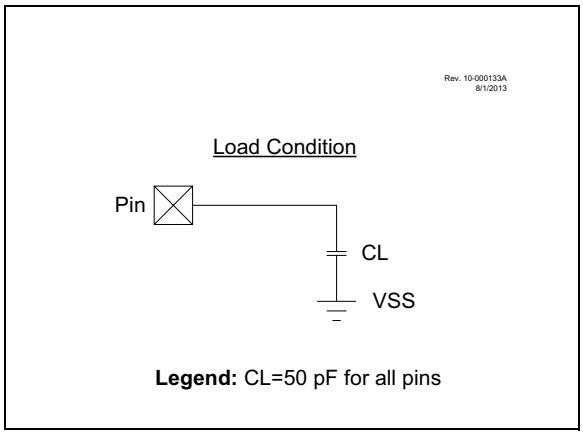
2: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to V_{SS}.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is FRC.

44.4 AC Characteristics

FIGURE 44-4: LOAD CONDITIONS



Legend: CL=50 pF for all pins