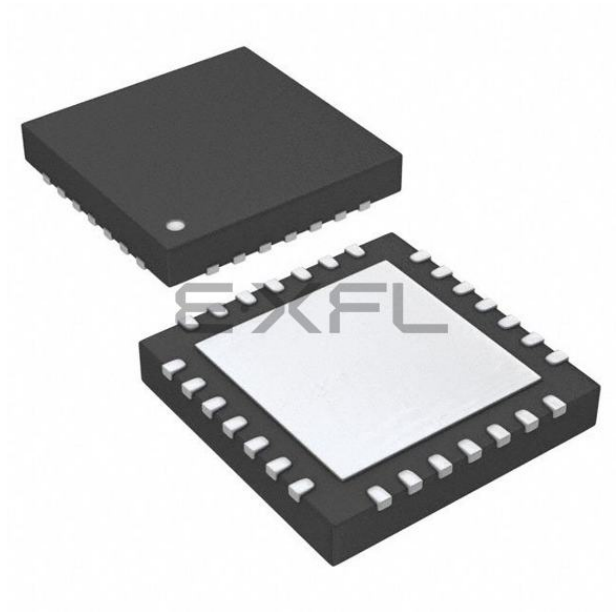


Welcome to E-XFL.COM

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	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k42t-i-ml

TABLE 2: 40/44-PIN ALLOCATION TABLE FOR PIC18(L)F4XK42

I/O	40-Pin PDIP	44-Pin TQFP	40-Pin UQFN	44-Pin QFN	ADC	Voltage Reference	DAC	Comparators	Zero Cross Detect	I^2C	SPI	UART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC3	18	37	33	37	ANC3	—	—	—	—	SCL1(3,4)	SCK1(1)	—	—	T2IN(1)	—	—	—	—	—	IOCC3	—
RC4	23	42	38	42	ANC4	—	—	—	—	SDA1(3,4)	SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	24	43	39	43	ANC5	—	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	25	44	40	44	ANC6	—	—	—	—	—	—	CTS1(1)	—	—	—	—	—	—	—	IOCC6	—
RC7	26	1	1	1	ANC7	—	—	—	—	—	—	RX1(1)	—	—	—	—	—	—	—	IOCC7	—
RD0	19	38	34	38	AND0	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD1	20	39	35	39	AND1	—	—	—	—	—(4)	—	—	—	—	—	—	—	—	—	—	—
RD2	21	40	36	40	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD3	22	41	37	41	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD4	27	2	2	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD5	28	3	3	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD6	29	4	4	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD7	30	5	5	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE0	8	25	23	25	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE1	9	26	24	26	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE2	10	27	25	27	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE3	1	18	16	18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	$\overline{MCLR}$ V_{PP}
V_{DD}	11, 32	7, 28	7, 26	7, 28	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
V_{SS}	12, 31	6, 29	6, 27	6, 30	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

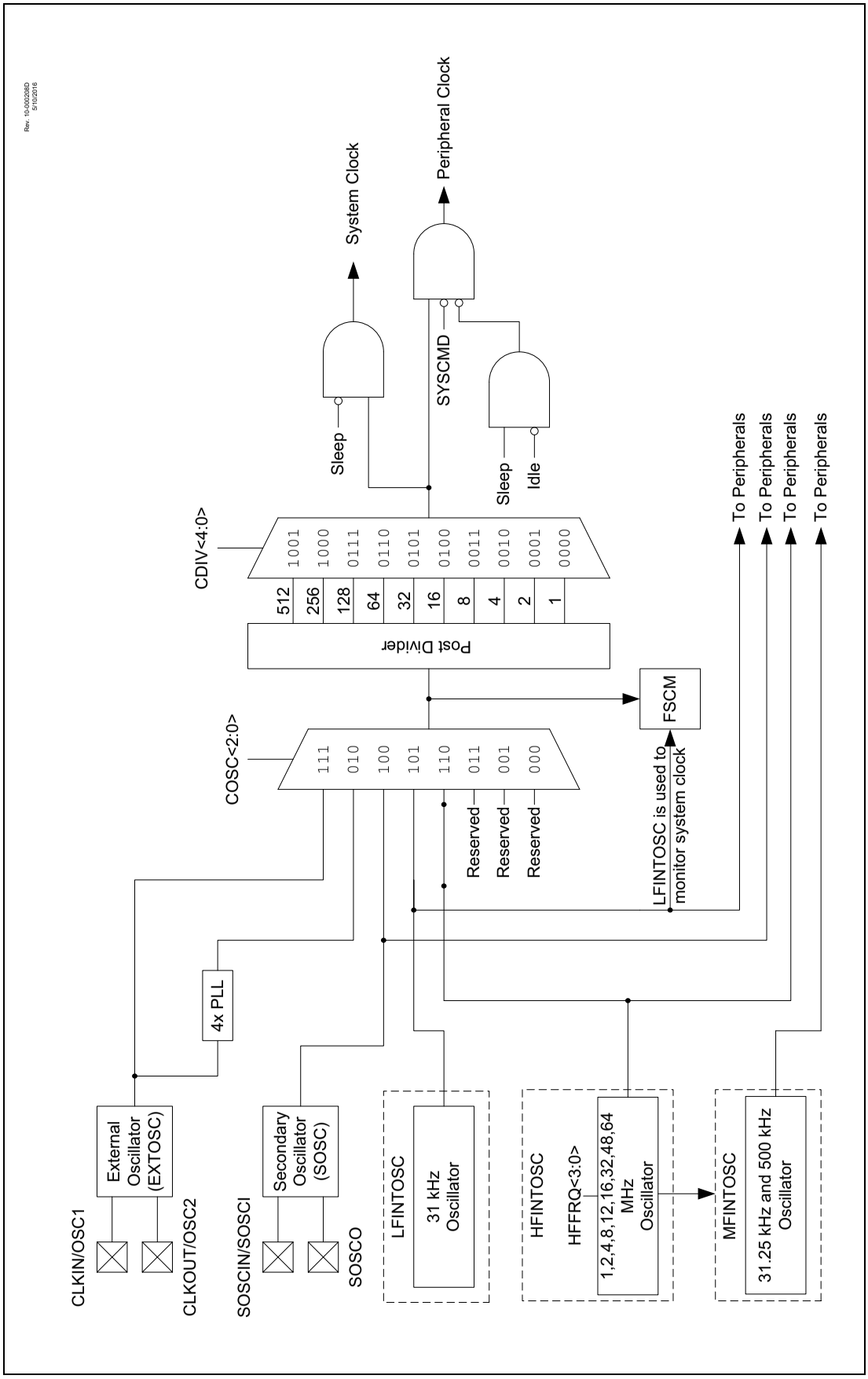
Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.

2: All output signals shown in this row are PPS remappable.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins can be configured for I^2C and SMB™ 3.0/2.0 logic levels; The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4/RD0/RD1 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

FIGURE 7-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



7.2 Clock Source Types

Clock sources can be classified as external or internal.

External clock sources rely on external circuitry for the clock source to function. Examples are: oscillator modules (ECH, ECM, ECL mode), quartz crystal resonators or ceramic resonators (LP, XT and HS modes).

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators that are used to generate internal system clock sources. The High-Frequency Internal Oscillator (HFINTOSC) can produce 1, 2, 4, 8, 12, 16, 32, 48 and 64 MHz clock. The frequency can be controlled through the OSCFRQ register (Register 7-5). The Low-Frequency Internal Oscillator (LFINTOSC) generates a fixed 31 kHz frequency.

A 4x PLL is provided that can be used with an external clock. When used with the HFINTOSC the 4x PLL has input frequency limitations.See **Section 7.2.1.4 "4x PLL"** for more details.

The system clock can be selected between external or internal clock sources via the NOSC bits in the OSCCON1 register. See **Section 7.3 "Clock Switching"** for additional information. The system clock can be made available on the OSC2/CLKOUT pin for any of the modes that do not use the OSC2 pin. The clock out functionality is governed by the $\overline{\text{CLKOUTEN}}$ bit in the CONFIG1H register (Register 5-2). If enabled, the clock out signal is always at a frequency of FOSC/4.

7.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the RSTOSC<2:0> and FEXTOSC<2:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> and NDIV<3:0> bits in the OSCCON1 register to switch the system clock source.

See **Section 7.3 "Clock Switching"** for more information.

7.2.1.1 EC Mode

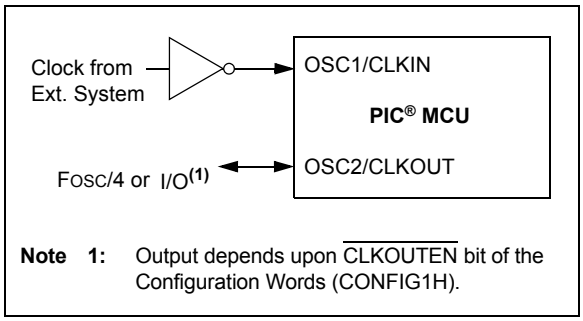
The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 7-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through Configuration Words:

- ECH – High power
- ECM – Medium power
- ECL – Low power

Refer to Table 44-9 for External Clock/Oscillator Timing Requirements. The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC® MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.

FIGURE 7-2: EXTERNAL CLOCK (EC) MODE OPERATION



Note 1: Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the Configuration Words (CONFIG1H).

7.2.1.2 LP, XT, HS Modes

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 7-3). The three modes select a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

XT Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification (above 100 kHz - 8 MHz).

HS Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting (above 8 MHz).

Figure 7-3 and Figure 7-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

REGISTER 7-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f	R-f/f
—	COSC<2:0>			CDIV<3:0>			
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 **Unimplemented:** Read as '0'

bit 6-4 **COSC<2:0>:** Current Oscillator Source Select bits (read-only)(1)
Indicates the current source oscillator and PLL combination per Table 7-1.

bit 3-0 **CDIV<3:0>:** Current Divider Select bits (read-only)(1)
Indicates the current postscaler division ratio per Table 7-1.

Note 1: The POR value is the value present when user code execution begins.

REGISTER 7-3: OSCCON3: OSCILLATOR CONTROL REGISTER 3

R/W/HC-0/0	R/W-0/0	U-0	R-0/0	R-0/0	U-0	U-0	U-0
CSWHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—
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	HC = Bit is cleared by hardware

bit 7 **CSWHOLD:** Clock Switch Hold bit
1 = Clock switch will hold (with interrupt) when the oscillator selected by NOSC is ready
0 = Clock switch may proceed when the oscillator selected by NOSC is ready; NOSCR becomes '1', the switch will occur

bit 6 **SOSCPWR:** Secondary Oscillator Power Mode Select bit
1 = Secondary oscillator operating in High-Power mode
0 = Secondary oscillator operating in Low-Power mode

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

bit 4 **ORDY:** Oscillator Ready bit (read-only)
1 = OSCCON1 = OSCCON2; the current system clock is the clock specified by NOSC
0 = A clock switch is in progress

bit 3 **NOSCR:** New Oscillator is Ready bit (read-only)(1)
1 = A clock switch is in progress and the oscillator selected by NOSC indicates a "ready" condition
0 = A clock switch is not in progress, or the NOSC-selected oscillator is not yet ready

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

Note 1: If CSWHOLD = 0, the user may not see this bit set because, when the oscillator becomes ready there may be a delay of one instruction clock before this bit is set. The clock switch occurs in the next instruction cycle and this bit is cleared.

EXAMPLE 9-4: SETTING UP VECTORED INTERRUPTS USING XC8

```
// NOTE 1: If IVTBASE is changed from its default value of 0x000008, then the
// "base(...)" argument must be provided in the ISR. Otherwise the vector
// table will be placed at 0x0008 by default regardless of the IVTBASE value.

// NOTE 2: When MVECEN=0 and IPEN=1, a separate argument as "high_priority"
// or "low_priority" can be used to distinguish between the two ISRs.
// If the argument is not provided, the ISR is considered high priority
// by default.

// NOTE 3: Multiple interrupts can be handled by the same ISR if they are
// specified in the "irq(...)" argument. Ex: irq(IRQ_TMR0, IRQ_CCP1)

void __interrupt(irq(IRQ_TMR0), base(0x4008)) TMR0_ISR(void)
{
        PIR3bits.TMR0IF = 0;                 // Clear the interrupt flag
        LATCbits.LC0 ^= 1;                   // ISR code goes here
}

void __interrupt(irq(default), base(0x4008)) DEFAULT_ISR(void)
{
        // Unhandled interrupts go here
}

void INTERRUPT_Initialize (void)
{
        INTCON0bits.GIEH = 1;                // Enable high priority interrupts
        INTCON0bits.GIEL = 1;                // Enable low priority interrupts
        INTCON0bits.IPEN = 1;                // Enable interrupt priority

        PIE3bits.TMR0IE = 1;                 // Enable TMR0 interrupt
        PIE4bits.TMR1IE = 1;                 // Enable TMR1 interrupt

        IPR3bits.TMR0IP = 0;                 // Make TMR0 interrupt low priority

        // Change IVTBASE if required
        IVTBASEU = 0x00;                     // Optional
        IVTBASEH = 0x40;                     // Default is 0x0008
        IVTBASEL = 0x08;
}
```

REGISTER 9-39: IVTADU: INTERRUPT VECTOR TABLE ADDRESS UPPER REGISTER

U-0	U-0	U-0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
—	—	—	AD<20:16>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **AD<20:16>**: Interrupt Vector Table Address bits

REGISTER 9-40: IVTADH: INTERRUPT VECTOR TABLE ADDRESS HIGH REGISTER

R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
AD<15:8>							
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 7-0 **AD<15:8>**: Interrupt Vector Table Address bits

REGISTER 9-41: IVTADL: INTERRUPT VECTOR TABLE ADDRESS LOW REGISTER

R-0/0	R-0/0	R-0/0	R-0/0	R-1/1	R-0/0	R-0/0	R-0/0
AD<7:0>							
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	

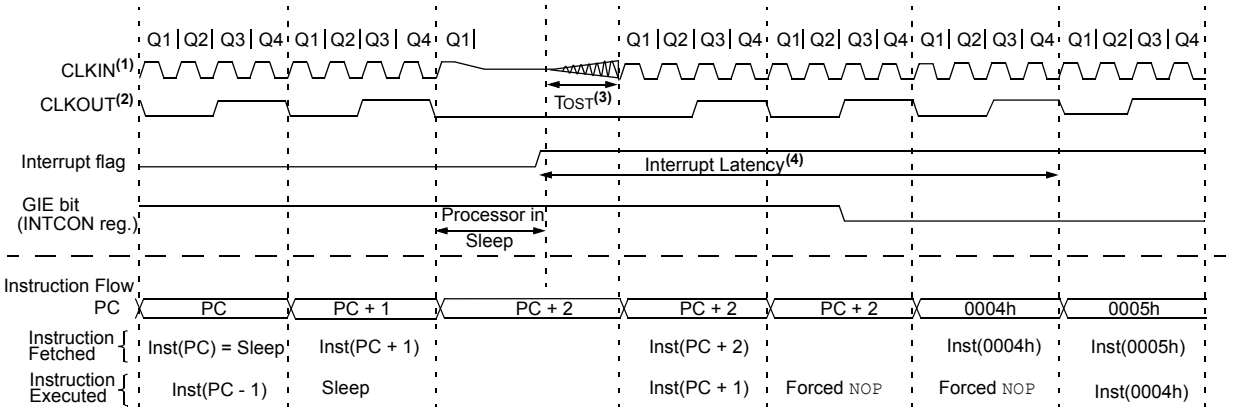
bit 7-0 **AD<7:0>**: Interrupt Vector Table Address bits

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON0	GIE/GIEH	GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	135
INTCON1	STAT<1:0>		—	—	—	—	—	—	136
PIE0	IOCIE	CRCIE	SCANIE	NVMIE	CSWIE	OSFIE	HLVDIE	SWIE	147
PIE1	SMT1PWAIE	SMT1PRAIE	SMT1IE	C1IE	ADTIE	ADIE	ZCDIE	INT0IE	148
PIE2	I2C1RXIE	SPI1IE	SPI1TXIE	SPI1RXIE	DMA1AIE	DMA1ORIE	DMA1DCNTIE	DMA1SCNTIE	149
PIE3	TMR0IE	U1IE	U1EIE	U1TXIE	U1RXIE	I2C1EIE	I2C1IE	I2C1TXIE	150
PIE4	CLC1IE	CWG1IE	NCO1IE	—	CCP1IE	TMR2IE	TMR1GIE	TMR1IE	151
PIE5	I2C2TXIE	I2C2RXIE	DMA2AIE	DMA2ORIE	DMA2DCNTIE	DMA2SCNTIE	C2IE	INT1IE	152
PIE6	TMR3GIE	TMR3IE	U2IE	U2EIE	U2TXIE	U2RXIE	I2C2EIE	I2C2IE	153
PIE7	—	—	INT2IE	CLC2IE	CWG2IE	—	CCP2IE	TMR4IE	154
PIE8	TMR5GIE	TMR5IE	—	—	—	—	—	—	155
PIE9	—	—	—	—	CLC3IE	CWG3IE	CCP3IE	TMR6IE	155
PIE10	—	—	—	—	—	—	CLC4IE	CCP4IE	156
PIR0	IOCIF	CRCIF	SCANIF	NVMIF	CSWIF	OSFIF	HLVDIF	SWIF	137
PIR1	SMT1PWAIF	SMT1PRAIF	SMT1IF	C1IF	ADTIF	ADIF	ZCDIF	INT0IF	138
PIR2	I2C1RXIF	SPI1IF	SPI1TXIF	SPI1RXIF	DMA1AIF	DMA1ORIF	DMA1DCNTIF	DMA1SCNTIF	139
PIR3	TMR0IF	U1IF	U1EIF	U1TXIF	U1RXIF	I2C1EIF	I2C1IF	I2C1TXIF	140
PIR4	CLC1IF	CWG1IF	NCO1IF	—	CCP1IF	TMR2IF	TMR1GIF	TMR1IF	141
PIR5	I2C2TXF	I2C2RXF	DMA2AIF	DMA2ORIF	DMA2DCNTIF	DMA2SCNTIF	C2IF	INT1IF	142
PIR6	TMR3GIF	TMR3IF	U2IF	U2EIF	U2TXIF	U2RXIF	I2C2EIF	I2C2IF	143
PIR7	—	—	INT2IF	CLC2IF	CWG2IF	—	CCP2IF	TMR4IF	144
PIR8	TMR5GIF	TMR5IF	—	—	—	—	—	—	145
PIR9	—	—	—	—	CLC3IF	CWG3IF	CCP3IF	TMR6IF	145
PIR10	—	—	—	—	—	—	CLC4IF	CCP4IF	146
IPR0	IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP	157
IPR1	SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP	158
IPR2	I2C1RIP	SPI1IP	SPI1TIP	SPI1RIP	DMA1AIP	DMA1ORIP	DMA1DCNTIP	DMA1SCNTIP	159
IPR3	TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP	160
IPR4	CLC1IP	CWG1IP	NCO1IP	—	CCP1IP	TMR2IP	TMR1GIP	TMR1IP	161
IPR5	I2C2TXP	I2C2RXP	DMA2AIP	DMA2ORIP	DMA2DCNTIP	DMA2SCNTIP	C2IP	INT1IP	162
IPR6	TMR3GIP	TMR3IP	U2IP	U2EIP	U2TXIP	U2RXIP	I2C2EIP	I2C2IP	163
IPR7	—	—	INT2IP	CLC2IP	CWG2IP	—	CCP2IP	TMR4IP	164
IPR8	TMR5GIP	TMR5IP	—	—	—	—	—	—	164
IPR9	—	—	—	—	CLC3IP	CWG3IP	CCP3IP	TMR6IP	165
IPR10	—	—	—	—	—	—	CLC4IP	CCP4IP	165
IVTBASEU	—	—	—	BASE<20:16>					166
IVTBASEH	BASE<15:8>								166
IVTBASEL	BASE<7:0>								166
IVTADU				AD<20:16>					167
IVTADH	AD<15:8>								167
IVTADL	AD<7:0>								167
IVTLOCK	—	—	—	—	—	—	—	IVTLOCKED	168

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for interrupts.

FIGURE 10-2: WAKE-UP FROM SLEEP THROUGH INTERRUPT



Note 1: External clock. High, Medium, Low mode assumed.
2: CLKOUT is shown here for timing reference.
3: TOST = 1024 TOSC. This delay does not apply to EC and INTOSC Oscillator modes.
4: GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

10.2.3 LOW-POWER SLEEP MODE

The PIC18F26/27/45/46/47/55/56/57K42 device family contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

The PIC18F26/27/45/46/47/55/56/57K42 devices allow the user to optimize the operating current in Sleep, depending on the application requirements.

Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register.

10.2.3.1 Sleep Current vs. Wake-up Time

In the default operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking-up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The Normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

13.1.2 CONTROL REGISTERS

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the following registers:

- NVMCON1 register
- NVMCON2 register
- TABLAT register
- TBLPTR registers

13.1.2.1 NVMCON1 and NVMCON2 Registers

The NVMCON1 register (Register 13-1) is the control register for memory accesses. The NVMCON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading NVMCON2 will read all '0's.

The REG<1:0> control bits determine if the access will be to Data EEPROM Memory locations. PFM locations or User IDs, Configuration bits, Rev ID and Device ID. When REG<1:0> = 00, any subsequent operations will operate on the Data EEPROM Memory. When REG<1:0> = 10, any subsequent operations will operate on the program memory. When REG<1:0> = x1, any subsequent operations will operate on the Configuration bits, User IDs, Rev ID and Device ID.

The FREE bit allows the program memory erase operation. When the FREE bit is set, an erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled. This bit is applicable only to the PFM and not to data EEPROM.

When set, the WREN bit will allow a program/erase operation. The WREN bit is cleared on power-up.

The WRERR bit is set by hardware when the WR bit is set and is cleared when the internal programming timer expires and the write operation is successfully complete.

The WR control bit initiates erase/write cycle operation when the REG<1:0> bits point to the Data EEPROM Memory location, and it initiates a write operation when the REG<1:0> bits point to the PFM location. The WR bit cannot be cleared by firmware; it can only be set by firmware. Then the WR bit is cleared by hardware at the completion of the write operation.

The NVMIF Interrupt Flag bit is set when the write is complete. The NVMIF flag stays set until cleared by firmware.

13.1.2.2 TABLAT – Table Latch Register

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

13.1.2.3 TBLPTR – Table Pointer Register

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the Device ID, the User ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations on the TBLPTR affect only the low-order 21 bits.

13.1.2.4 Table Pointer Boundaries

TBLPTR is used in reads, writes and erases of the Program Flash Memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory directly into the TABLAT register.

When a TBLWT is executed the byte in the TABLAT register is written, not to memory but, to a holding register in preparation for a program memory write. The holding registers constitute a write block which varies depending on the device (see Table 5-4).The 3, 4, or 5 LSbs of the TBLPTRL register determine which specific address within the holding register block is written to. The MSBs of the Table Pointer have no effect during TBLWT operations.

When a program memory write is executed the entire holding register block is written to the memory at the address determined by the MSbs of the TBLPTR. The 3, 4, or 5 LSBs are ignored during memory writes. For more detail, see **Section 13.1.6 "Writing to Program Flash Memory"**.

Figure 13-3 describes the relevant boundaries of TBLPTR based on Program Flash Memory operations.

13.1.4 NVM UNLOCK SEQUENCE

The unlock sequence is a mechanism that protects the NVM from unintended self-write programming or erasing. The sequence must be executed and completed without interruption to successfully complete any of the following operations:

- PFM Row Erase
- Write of PFM write latches to PFM memory
- Write of PFM write latches to User IDs
- Write to Data EEPROM Memory
- Write to Configuration Words

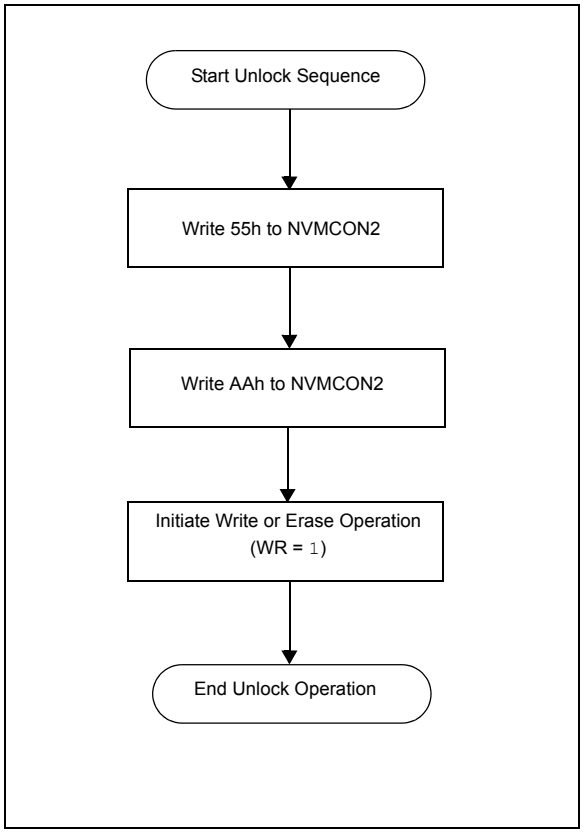
The unlock sequence consists of the following steps and must be completed in order:

- Write 55h to NVMCON2
- Write AAh to NMVCON2
- Set the WR bit of NVMCON1

Once the WR bit is set, the processor will stall internal operations until the operation is complete and then resume with the next instruction.

Since the unlock sequence must not be interrupted, global interrupts should be disabled prior to the unlock sequence and re-enabled after the unlock sequence is completed.

FIGURE 13-6: NVM UNLOCK SEQUENCE FLOWCHART



EXAMPLE 13-2: NVM UNLOCK SEQUENCE

```
BCF                 INTCON0,GIE             ; Recommended so sequence is not interrupted
BANKSEL             NVMCON1
BSF                 NVMCON1,WREN            ; Enable write/erase
MOVLW               55h                     ; Load 55h

MOVWF               NVMCON2                 ; Step 1: Load 55h into NVMCON2
MOVLW               AAh                     ; Step 2: Load W with AAh
MOVWF               NVMCON2                 ; Step 3: Load AAh into NVMCON2
BSF                 INTCON1,WR              ; Step 4: Set WR bit to begin write/erase

BSF                 INTCON0,GIE             ; Re-enable interrupts
```

Note 1: Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown. If the timing of the steps 1 to 4 is corrupted by an interrupt or a debugger Halt, the action will not take place.

2: Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

15.5.3.1 Clearing the SIRQEN bit

Clearing the SIRQEN bit (DMAxCON1 register) stops the sampling of external start interrupt triggers, hence preventing further DMA Message transfers.

An example would be a communications peripheral with a level-triggered interrupt. The peripheral will continue to request data (because its buffer is empty) even though there is no more data to be moved. Disabling the SIRQEN bit prevents the DMA from processing these requests.

15.5.3.2 Source/Destination Stop

The SSTP and DSTP bits (DMAxCON0 register) determine whether or not to disable the hardware triggers (SIRQEN = 0) once a DMA message has completed.

When the SSTP bit is set and the DMAxSCNT = 0, then the SIRQEN bit will be cleared. Similarly, when the DSTP bit is set and the DMAxDCNT = 0, the SIRQEN bit will be cleared.

Note:	The SSTP and DSTP bits are independent functions and do not depend on each other. It is possible for a message to be stopped by either counter at message end or both counters at message end.

15.6 Types of Hardware Triggers

The DMA has two different trigger inputs namely the Source trigger and the abort trigger. Each of these trigger sources is user configurable using the DMAxSIRQ and DMAxAIRQ registers.

Based on the source selected for each trigger, there are two types of requests that can be sent to the DMA.

- Edge triggers
- Level triggers

15.6.1 EDGE TRIGGER REQUESTS

An Edge request occurs only once when a given module interrupt requirements are true.

15.6.2 LEVEL TRIGGER REQUESTS

A level request is asserted as long as the condition that causes the interrupt is true.

15.7 Types of Data Transfers

Based on the memory access capabilities of the DMA (See Table 15-1), the following sections discuss the different types of data movement between the Source and Destination Memory regions.

- N: 1

This type of transfer is common when sending predefined data packets (such as strings) through a single interface point (such as communications modules transmit registers).

- N: N

This type of transfer is useful for moving information out of the Program Flash or Data EEPROM to SRAM for manipulation by the CPU or other peripherals.

- 1: N

This type of transfer is common when bridging two different modules data streams together (communications bridge).

- 1: N

This type of transfer is useful for moving information from a single data source into a memory buffer (communications receive registers).

15.8 DMA Interrupts

Each DMA has its own set of four interrupt flags, used to indicate a range of conditions during data transfers. The interrupt flag bits can be accessed using the corresponding PIR registers (Refer to the Interrupt Section).

15.8.1 DMA SOURCE COUNT INTERRUPT

The DMAxSCNTIF source count interrupt flag is set every time the DMAxSCNT<11:0> reaches zero and is reloaded to its starting value.

15.8.2 DMA DESTINATION COUNT INTERRUPT

The DMAxDCNTIF destination count interrupt flag is set every time the DMAxDCNT<11:0> reaches zero and is reloaded to its starting value.

The DMA Source Count zero and Destination Count zero interrupts are used in conjunction to determine when to signal the CPU when the DMA Messages are completed.

15.8.3 ABORT INTERRUPT

The DMAxAIF abort interrupt flag is used to signal that the DMA has halted activity due to an abort signal from one of the abort sources. This is used to indicate that the transaction has been halted for some reason.

FIGURE 25-11: WINDOWED MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM

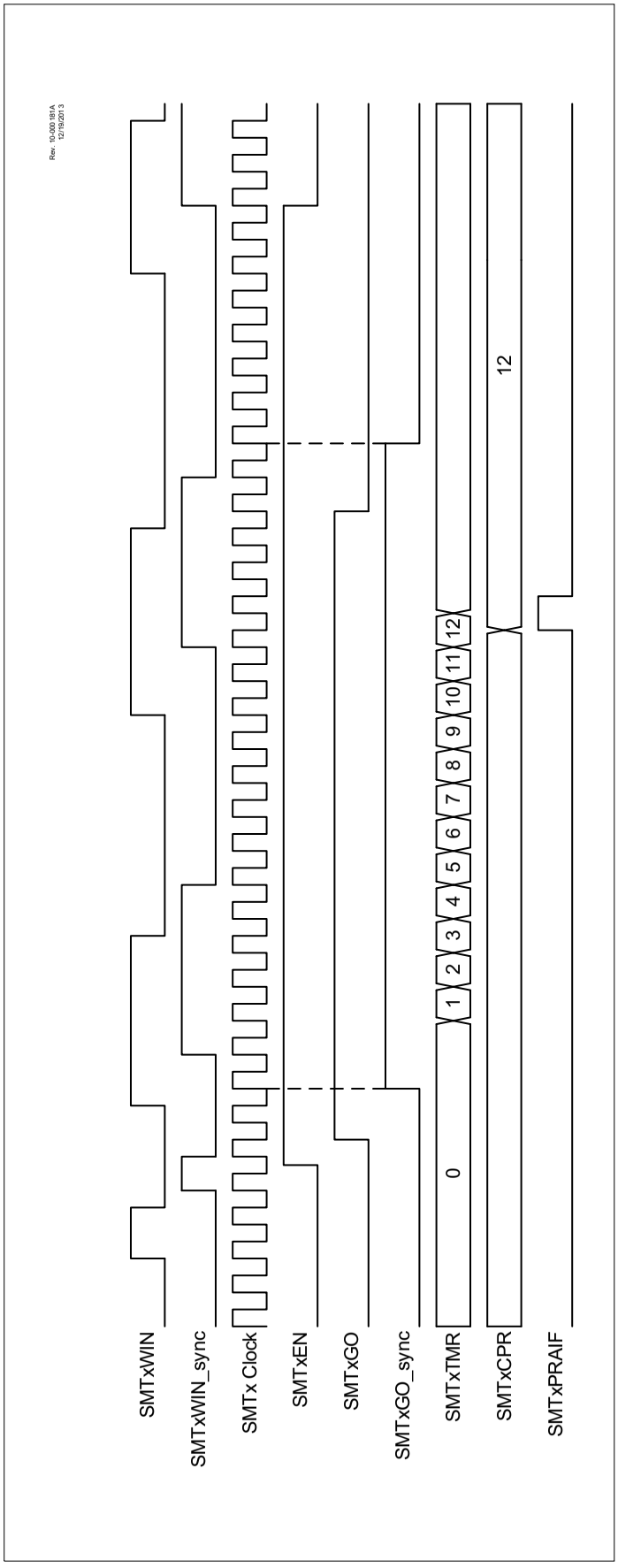


TABLE 26-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWGxCON0	EN	LD	—	—	—	MODE<2:0>			424
CWGxCON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
CWGxCLK	—	—	—	—	—	—	—	CS	426
CWGxISM	—	—	—	ISM<4:0>					427
CWGxSTR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
CWGxAS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	429
CWGxAS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
CWGxDBR	—	—	DBR<5:0>						431
CWGxDBF	—	—	DBF<5:0>						431

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

REGISTER 33-14: I2CxADR2: I²C ADDRESS 2 REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
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	HS = Hardware set HC = Hardware clear

bit 7-0 **ADR<7-0>:** Address 2 bits

MODE<2:0> = 000 | 110 - 7-bit Slave/Multi-Master Modes
ADR<7:1>:7-bit Slave Address

MODE<2:0> = 001 | 111 - 7-bit Slave/Multi-Master Modes with Masking
ADR<7:1>:7-bit Slave Address

MODE<2:0> = 010 - 10-Bit Slave Mode
ADR<7:0>: Eight Least Significant bits of second 10-bit address

MODE<2:0> = 011 - 10-Bit Slave Mode with Masking
MSK0<7-0>: The received address byte is masked, then compared to I2CxADR0

36.6.5 BURST AVERAGE MODE

The Burst Average mode (ADMD = 011) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the CNT value is greater than or equal to RPT, even if Continuous Sampling mode (see **Section 36.6.8 "Continuous Sampling mode"**) is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

36.6.6 LOW-PASS FILTER MODE

The Low-pass Filter mode (ADMD = 100) acts similarly to the Average mode in how it handles samples (accumulates samples until CNT value greater than or equal to RPT, then triggers threshold comparison), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. (see Table 36-2 for a more detailed description of the mathematical operation). In this mode, the ADCRS bits determine the cut-off frequency of the low-pass filter (as demonstrated by Table 36-3).

36.6.7 THRESHOLD COMPARISON

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error is calculated based on a difference calculation which is selected by the ADCALC<2:0> bits in the ADCON3 register. The value can be one of the following calculations (see Register 36-4 for more details):
  - The first derivative of single measurements
  - The CVD result in CVD mode
  - The current result vs. a setpoint
  - The current result vs. the filtered/average result
  - The first derivative of the filtered/average value
  - Filtered/average value vs. a setpoint
- The result of the calculation (ERR) is compared to the upper and lower thresholds, UTH<ADUTHH:ADUTHL> and LTH<ADLTHH:ADLTHL> registers, to set the ADUTHR and ADLTHR flag bits. The threshold logic is selected by ADTMD<2:0> bits in the ADCON3 register. The threshold trigger option can be one of the following:
  - Never interrupt
  - Error is less than lower threshold
  - Error is greater than or equal to lower threshold
  - Error is between thresholds (inclusive)
  - Error is outside of thresholds
  - Error is less than or equal to upper threshold
  - Error is greater than upper threshold
  - Always interrupt regardless of threshold test results
  - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

Note 1: The threshold tests are signed operations.

2: If ADAOV is set, a threshold interrupt is signaled.

36.6.8 CONTINUOUS SAMPLING MODE

Setting the CONT bit in the ADCON0 register automatically retriggers a new conversion cycle after updating the ADACC register. The GO bit remains set and re-triggering occurs automatically.

If ADSOI = 1, a threshold interrupt condition will clear GO and the conversions will stop.

36.6.9 DOUBLE SAMPLE CONVERSION

Double sampling is enabled by setting the ADDSEN bit of the ADCON1 register. When this bit is set, two conversions are required before the module will calculate threshold error (each conversion must still be triggered separately). The first conversion will set the ADMATH bit of the ADSTAT register and update ADACC, but will not calculate ERR or trigger ADTIF. When the second conversion completes, the first value is transferred to PREV (depending on the setting of ADPSIS) and the value of the second conversion is placed into ADRES. Only upon the completion of the second conversion is ERR calculated and ADTIF triggered (depending on the value of ADCALC).

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

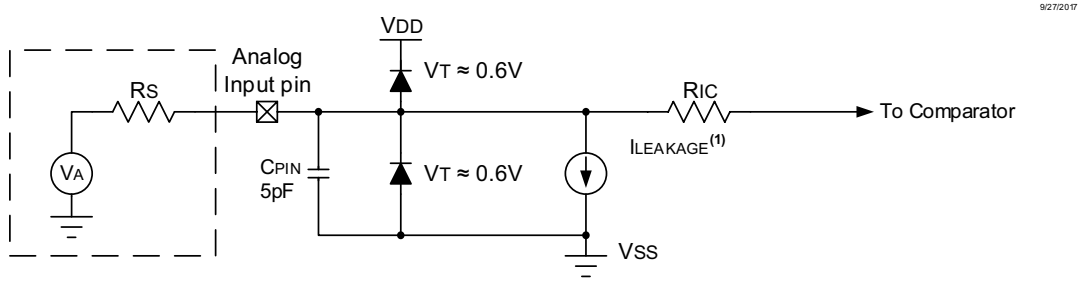
A simplified circuit for an analog input is shown in Figure 38-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to V_{DD} and V_{SS}. The analog input, therefore, must be between V_{SS} and V_{DD}. If the input voltage deviates 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"**.

41.1.1 STANDARD INSTRUCTION SET

ADDFSR **Add Literal to FSR**

Syntax: ADDFSR f, k

Operands: $0 \le k \le 63$
$f \in [0, 1, 2]$

Operation: FSR(f) + k → FSR(f)

Status Affected: None

Encoding:

1110	1000	ffkk	kkkk

Description: The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: `ADDFSR 2, 23h`

Before Instruction
FSR2 = 03FFh
After Instruction
FSR2 = 0422h

ADDLW **ADD literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → W

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

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal 'k' and 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: `ADDLW 15h`

Before Instruction
W = 10h
After Instruction
W = 25h

ADDWF **ADD W to f**

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

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

Operation: (W) + (f) → dest

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

Encoding:

0010	01da	ffff	ffff

Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
If 'a' is '0', the Access Bank 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 to destination

Example: `ADDWF REG, 0, 0`

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

TABLE 44-13: RESET, WDT, OSCILLATOR START-UP TIMER, POWER-UP TIMER, BROWN-OUT RESET AND LOW-POWER BROWN-OUT RESET SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
RST01*	TMCLR	$\overline{MCLR}$ Pulse Width Low to ensure Reset	2	—	—	µs	
RST02*	TIOZ	I/O high-impedance from Reset detection	—	—	2	µs	
RST03	TWDT	Watchdog Timer Time-out Period	—	16	—	ms	1:512 Prescaler
RST04*	TPWRT	Power-up Timer Period	— — —	1 16 64	—	ms ms ms	PWRTS = 00 PWRTS = 01 PWRTS = 10
RST05	TOST	Oscillator Start-up Timer Period(1,2)	—	1024	—	TOSC	
RST06	VBOR	Brown-out Reset Voltage(4)	2.7 2.55 2.3 2.3 1.8	2.85 2.7 2.45 2.45 1.9	3.0 2.85 2.6 2.6 2.05	V V V V V	BORV = 00 BORV = 01 BORV = 10 BORV = 11 (PIC18Fxxx) BORV = 11 (PIC18LFxxx)
RST07	VBORHYS	Brown-out Reset Hysteresis	—	40	—	mV	
RST08	TBORDC	Brown-out Reset Response Time	—	3	—	µs	
RST09	VLPBOR	Low-Power Brown-out Reset Voltage	1.8	1.9	2.5	V	PIC18LFXXX only

* 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.

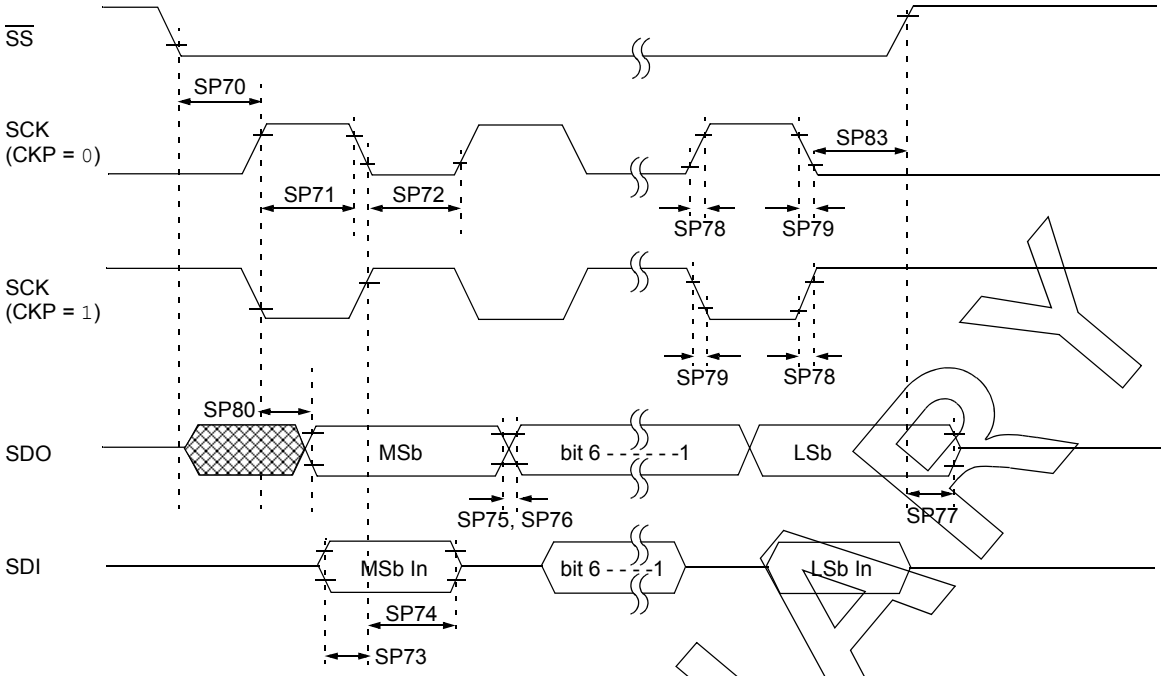
Note 1: By design, the Oscillator Start-up Timer (OST) counts the first 1024 cycles, independent of frequency.

2: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

TABLE 44-14: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS

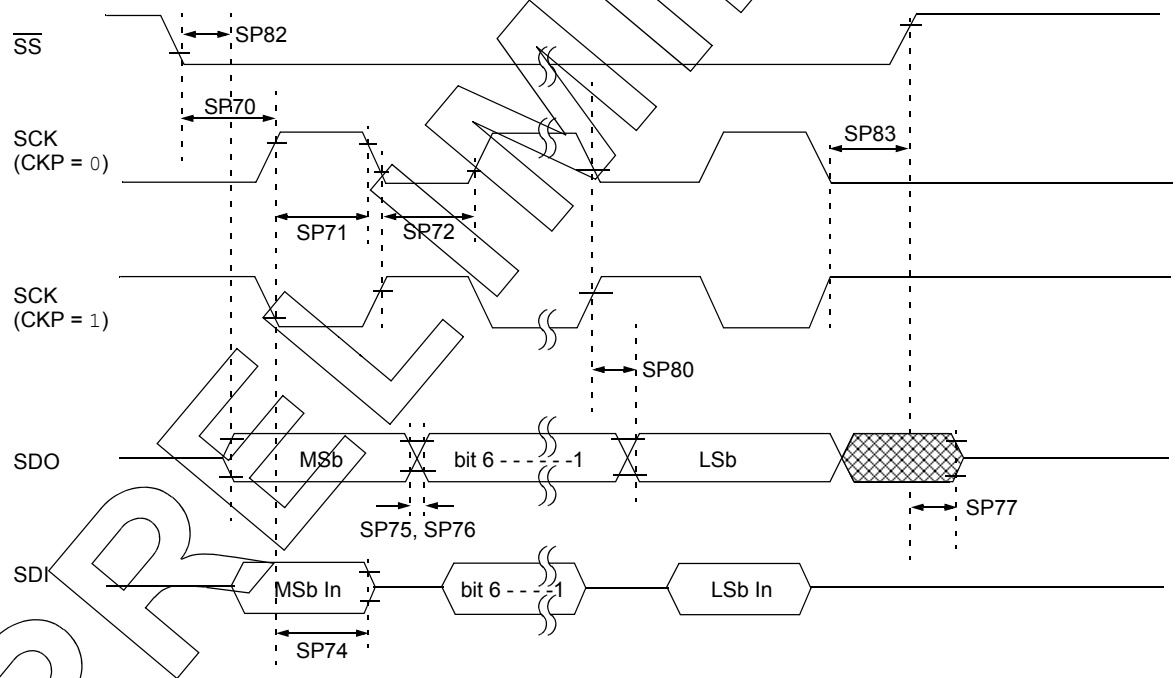
Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Characteristic	Min.	Typ†	Max.	Units	Conditions
HLVD01	V_{DET}	Voltage Detection	—	1.85	—	V	HLVDSEL<3:0>=0000
			—	2.06	—	V	HLVDSEL<3:0>=0001
			—	2.26	—	V	HLVDSEL<3:0>=0010
			—	2.47	—	V	HLVDSEL<3:0>=0011
			—	2.57	—	V	HLVDSEL<3:0>=0100
			—	2.78	—	V	HLVDSEL<3:0>=0101
			—	2.88	—	V	HLVDSEL<3:0>=0110
			—	3.09	—	V	HLVDSEL<3:0>=0111
			—	3.40	—	V	HLVDSEL<3:0>=1000
			—	3.60	—	V	HLVDSEL<3:0>=1001
			—	3.71	—	V	HLVDSEL<3:0>=1010
			—	3.91	—	V	HLVDSEL<3:0>=1011
			—	4.12	—	V	HLVDSEL<3:0>=1100
			—	4.32	—	V	HLVDSEL<3:0>=1101
			—	4.63	—	V	HLVDSEL<3:0>=1110

FIGURE 44-16: SPI SLAVE MODE TIMING (CKE = 0)



Note: Refer to Figure 44-4 for load conditions.

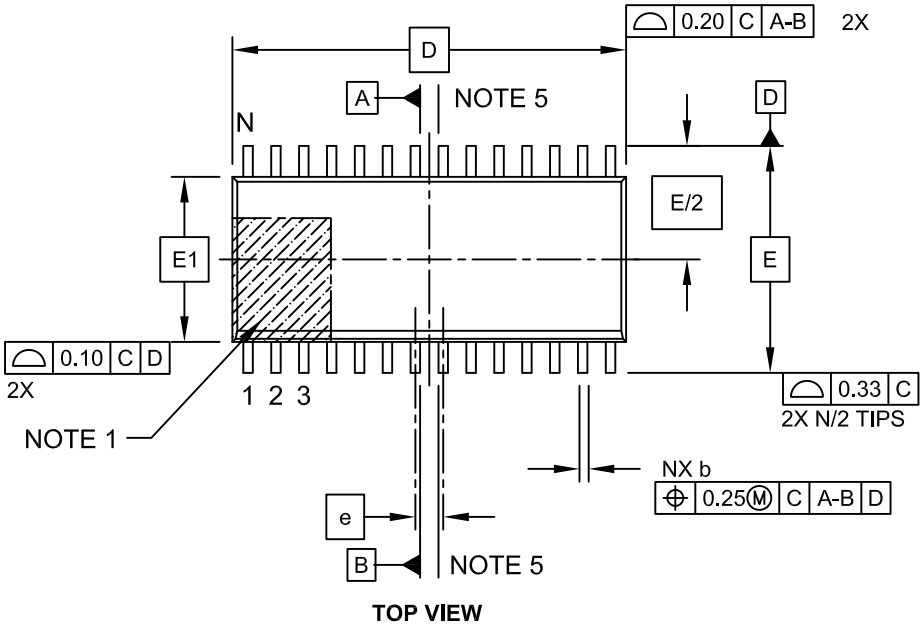
FIGURE 44-17: SPI SLAVE MODE TIMING (CKE = 1)



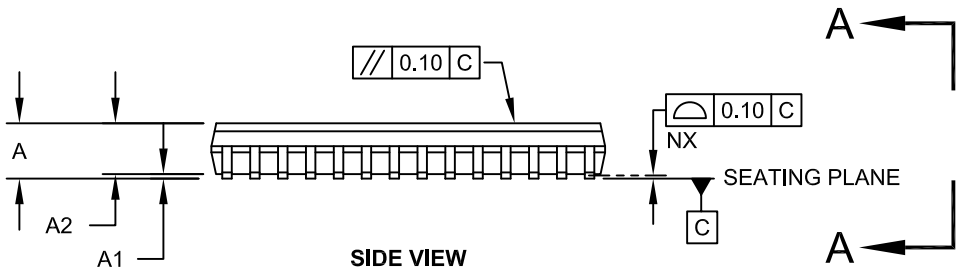
Note: Refer to Figure 44-4 for load conditions.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

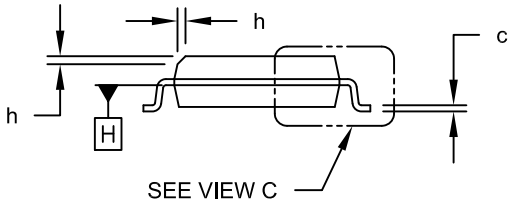
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2