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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k42-e-ss

2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 7.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-3. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

For additional information and design guidance on oscillator circuits, refer to these Microchip application notes, available at the corporate website (www.microchip.com):

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices*"
- AN849, *"Basic PICmicro® Oscillator Design"*
- AN943, *"Practical PICmicro® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to Vss on unused pins and drive the output to logic low.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT

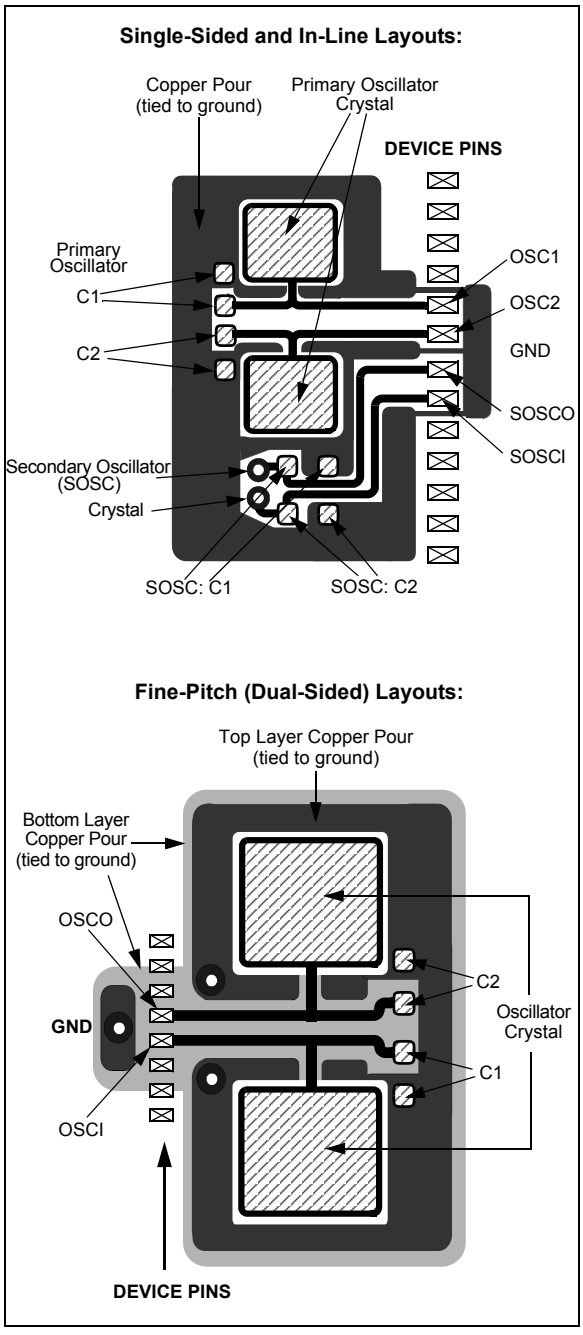


TABLE 4-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES BANK 64

40FFh	—	40DFh	—	40BFh	—	409Fh	—	407Fh	—	405Fh	—	403Fh	—	401Fh	—
40FEh	—	40DEh	—	40BEh	—	409Eh	—	407Eh	—	405Eh	—	403Eh	—	401Eh	—
40FDh	—	40DDh	T6PR_M2	40BDh	ADRESH_M2	409Dh	—	407Dh	—	405Dh	—	403Dh	—	401Dh	—
40FCh	—	40DCh	PWM5DCH_M2	40BCh	ADRESL_M2	409Ch	—	407Ch	—	405Ch	—	403Ch	—	401Ch	—
40FBh	TMR5H_M1	40DBh	PWM5DCL_M2	40BBh	ADPCH_M2	409Bh	—	407Bh	—	405Bh	—	403Bh	—	401Bh	—
40FAh	TMR5L_M1	40DAh	T6PR_M1	40BAh	ADCLK_M1	409Ah	—	407Ah	—	405Ah	—	403Ah	—	401Ah	—
40F9h	TMR3H_M1	40D9h	CCPR1H_M2	40B9h	ADACT_M1	4099h	—	4079h	—	4059h	—	4039h	—	4019h	—
40F8h	TMR3L_M1	40D8h	CCPR1L_M2	40B8h	ADREF_M1	4098h	—	4078h	—	4058h	—	4038h	—	4018h	—
40F7h	TMR1H_M1	40D7h	T4PR_M4	40B7h	ADCON3_M1	4097h	—	4077h	—	4057h	—	4037h	—	4017h	—
40F6h	TMR1L_M1	40D6h	PWM8DCH_M1	40B6h	ADCON2_M1	4096h	ADRESH_M1	4076h	—	4056h	—	4036h	—	4016h	—
40F5h	—	40D5h	PWM8DCL_M1	40B5h	ADCON1_M1	4095h	ADRESL_M1	4075h	—	4055h	—	4035h	—	4015h	—
40F4h	—	40D4h	T4PR_M3	40B4h	ADCON0_M1	4094h	ADPCH_M1	4074h	—	4054h	—	4034h	—	4014h	—
40F3h	—	40D3h	PWM7DCH_M1	40B3h	ADCAP_M2	4093h	ADCAP_M1	4073h	—	4053h	—	4033h	—	4013h	—
40F2h	—	40D2h	PWM7DCL_M1	40B2h	ADACQH_M2	4092h	ADACQH_M1	4072h	—	4052h	—	4032h	—	4012h	—
40F1h	—	40D1h	T4PR_M2	40B1h	ADACQL_M2	4091h	ADACQL_M1	4071h	—	4051h	—	4031h	—	4011h	—
40F0h	—	40D0h	CCPR4H_M1	40B0h	ADPREVH_M2	4090h	ADPREVH_M1	4070h	—	4050h	—	4030h	—	4010h	—
40EFh	PWM8DCH_M2	40CFh	CCPR4L_M1	40AFh	ADPREVL_M2	408Fh	ADPREVL_M1	406Fh	—	404Fh	—	402Fh	—	400Fh	—
40EEh	PWM8DCL_M2	40CEh	T4PR_M1	40AEh	ADRPT_M2	408Eh	ADRPT_M1	406Eh	—	404Eh	—	402Eh	—	400Eh	—
40EDh	PWM7DCH_M2	40CDh	CCPR3H_M1	40ADh	ADCNT_M2	408Dh	ADCNT_M1	406Dh	—	404Dh	—	402Dh	—	400Dh	—
40ECh	PWM7DCL_M2	40CCh	CCPR3L_M1	40ACh	ADACCU_M2	408Ch	ADACCU_M1	406Ch	—	404Ch	—	402Ch	—	400Ch	—
40EBh	PWM6DCH_M2	40CBh	T2PR_M3	40ABh	ADACCH_M2	408Bh	ADACCH_M1	406Bh	—	404Bh	—	402Bh	—	400Bh	—
40EAh	PWM6DCL_M2	40CAh	PWM6DCH_M1	40AAh	ADACCL_M2	408Ah	ADACCL_M1	406Ah	—	404Ah	—	402Ah	—	400Ah	—
40E9h	PWM5DCH_M3	40C9h	PWM6DCL_M1	40A9h	ADFLTRH_M2	4089h	ADFLTRH_M1	4069h	—	4049h	—	4029h	—	4009h	—
40E8h	PWM5DCL_M3	40C8h	T2PR_M2	40A8h	ADFLTRL_M2	4088h	ADFLTRL_M1	4068h	—	4048h	—	4028h	—	4008h	—
40E7h	CCPR4H_M2	40C7h	PWM5DCH_M1	40A7h	ADSTPTH_M2	4087h	ADSTPTH_M1	4067h	—	4047h	—	4027h	—	4007h	—
40E6h	CCPR4L_M2	40C6h	PWM5DCL_M1	40A6h	ADSTPTL_M2	4086h	ADSTPTL_M1	4066h	—	4046h	—	4026h	—	4006h	—
40E5h	CCPR3H_M2	40C5h	T2PR_M2	40A5h	ADERRH_M2	4085h	ADERRH_M1	4065h	—	4045h	—	4025h	—	4005h	—
40E4h	CCPR3L_M2	40C4h	CCPR2H_M1	40A4h	ADERRL_M2	4084h	ADERRL_M1	4064h	—	4044h	—	4024h	—	4004h	—
40E3h	CCPR2H_M2	40C3h	CCPR2L_M1	40A3h	ADUTHH_M2	4083h	ADUTHH_M1	4063h	IOCEF_M1	4043h	—	4023h	—	4003h	—
40E2h	CCPR2L_M2	40C2h	T2PR_M1	40A2h	ADUTHL_M2	4082h	ADUTHL_M1	4062h	IOCCF_M1	4042h	—	4022h	—	4002h	—
40E1h	CCPR1H_M3	40C1h	CCPR1H_M1	40A1h	ADLTHH_M2	4081h	ADLTHH_M1	4061h	IOCBF_M1	4041h	—	4021h	—	4001h	—
40E0h	CCPR1L_M3	40C0h	CCPR1L_M1	40A0h	ADLTHL_M2	4080h	ADLTHL_M1	4060h	IOCAF_M1	4040h	—	4020h	—	4000h	—

Note 1: Addresses in Bank 64 are accessible ONLY through DMA Source and Destination Address Registers. CPU does not have access to registers in Bank 64.

7.5 Register Definitions: Oscillator Control

REGISTER 7-1: OSCCON1: OSCILLATOR CONTROL REGISTER 1

U-0	R/W-f/f	R/W-f/f	R/W-f/f	R/W-q/q	R/W-q/q	R/W-q/q	R/W-q/q
—	NOSC<2:0>			NDIV<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	f = determined by Configuration bit setting
		q = Reset value is determined by hardware

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **NOSC<2:0>:** New Oscillator Source Request bits(1,2,3)
The setting requests a source oscillator and PLL combination per Table 7-1.
POR value = RSTOSC (Register 5-1).

bit 3-0 **NDIV<3:0>:** New Divider Selection Request bits(2,3)
The setting determines the new postscaler division ratio per Table 7-1.

Note 1: The default value (f/f) is determined by the RSTOSC Configuration bits. See Table 7-2 below.
2: If NOSC is written with a reserved value (Table 7-1), the operation is ignored and neither NOSC nor NDIV is written.
3: When CSWEN = 0, this register is read-only and cannot be changed from the POR value.

TABLE 7-2: DEFAULT OSCILLATOR SETTINGS

RSTOSC	SFR Reset Values			Initial Fosc Frequency
	NOSC/COSC	CDIV	OSCFRQ	
111	111	1:1	4 MHz	EXTOSC per FEXTOSC
110	110	4:1		FOSC = 1 MHz (4 MHz/4)
101	101	1:1		LFINTOSC
100	100	1:1		SOSC
011	Reserved			
010	010	1:1	4 MHz	EXTOSC + 4xPLL(1)
001	Reserved			
000	110	1:1	64 MHz	FOSC = 64 MHz

Note 1: EXTOSC must meet the PLL specifications (Table 44-11).

9.6 Returning from Interrupt Service Routine (ISR)

The “Return from Interrupt” instruction (`RETFIE`) is used to mark the end of an ISR.

When `RETFIE 1` instruction is executed, the PC is loaded with the saved PC value from the top of the PC stack. Saved context is also restored with the execution of this instruction. Thus, execution returns to the previous state of operation that existed before the interrupt occurred.

When `RETFIE 0` instruction is executed, the saved context is not restored back to the registers.

9.7 Interrupt Latency

By assigning each interrupt with a vector address/ number (MVECEN = `1`), scanning of all interrupts is not necessary to determine the source of the interrupt.

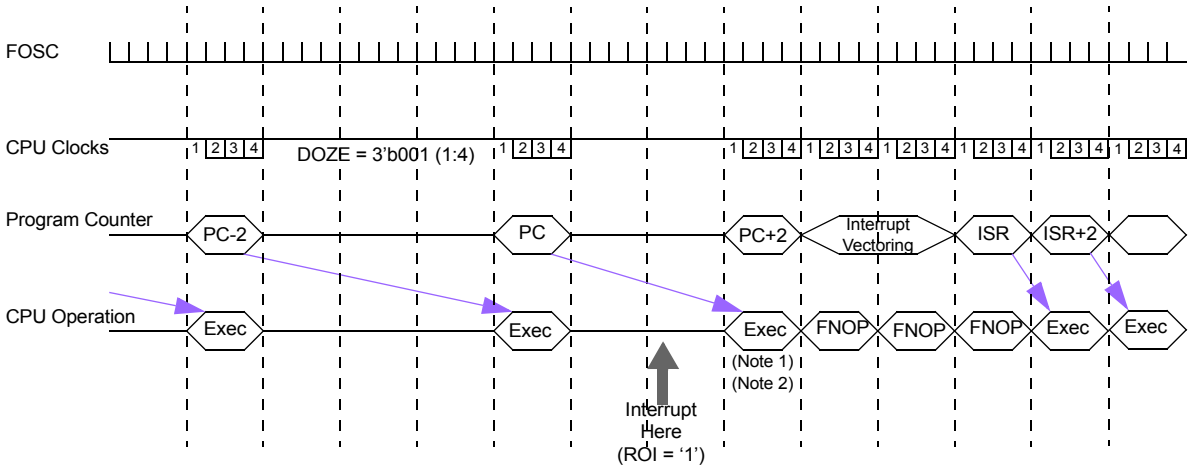
When MVECEN = `1`, Vectored interrupt controller requires three clock cycles to vector to the ISR from main routine, thereby removing dependency of interrupt timing on compiled code.

There is a fixed latency of three instruction cycles between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine. Figure 9-7, Figure 9-8 and Figure 9-9 illustrate the sequence of events when a peripheral interrupt is asserted when the last executed instruction is one-cycle, two-cycle and three-cycle respectively, when MVECEN = `1`.

After the Interrupt Flag Status bit is set, the current instruction completes executing. In the first latency cycle, the contents of the PC, STATUS, WREG, BSR, FSR0/1/2, PRODL/H and PCLATH/U registers are context saved and the IVTBASE+ Vector number is calculated. In the second latency cycle, the PC is loaded with the calculated vector table address for the interrupt source and the starting address of the ISR is fetched. In the third latency cycle, the PC is loaded with the ISR address. All the latency cycles are executed as a `FNOP` instruction.

When MVECEN = `0`, Vectored interrupt controller requires two clock cycles to vector to the ISR from main routine. There is a latency of two instruction cycles plus the software latency between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine.

FIGURE 10-1: DOZE MODE OPERATION EXAMPLE (DOZE<2:0> = 001, 1:4)



Note 1: Multicycle instructions are executed to completion before fetching the interrupt vector.
2: If the prefetched instruction clears the interrupt enable or GIEx, ISR vectoring will not occur, but DOZEN is cleared and the CPU will resume execution at full speed.

10.2 Sleep Mode

Sleep mode is entered by executing the SLEEP instruction, while the Idle Enable (IDLEN) bit of the CPUDOZE register is clear (IDLEN = 0).

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running if enabled for operation during Sleep
2. The $\overline{PD}$ bit of the STATUS register is cleared (Register 4-2)
3. The $\overline{TO}$ bit of the STATUS register is set (Register 4-2)
4. The CPU clock is disabled
5. LFINTOSC, SOSC, HFINTOSC and ADCRC are unaffected and peripherals using them may continue operation in Sleep.
6. I/O ports maintain the status they had before Sleep was executed (driving high, low, or high-impedance)
7. Resets other than WDT are not affected by Sleep mode

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using any oscillator

I/O pins that are high-impedance inputs should be pulled to VDD or VSS externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include modules such as the DAC and FVR modules. See **Section 37.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 34.0 "Fixed Voltage Reference (FVR)"** for more information on these modules.

TABLE 15-6: EXAMPLE DMA USE CASE TABLE

Source Module	Source Register(s)	Destination Module	Destination Register(s)	DCHxSIRQ	Comment
Signal Measurement Timer (SMT)	SMTxCPW[U:H:L]	GPR	GPR[x,y,z]	SMTxPWAIF	Store Captured Pulse-width values
	SMTxCPR[U:H:L]			SMTxPRAIF	Store Captured Period values
GPR/SFR/Program Flash/Data EEPROM	MEMORY[x,y]	TMR0	TMR0[H:L]	TMR0IF	Use as a Timer0 reload for custom 16-bit value
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	TMR0	PR0	ANY	Update TMR0 frequency based on a specific trigger
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y]	TMR1	TMR1[H:L]	TMR1IF	Use as a Timer1 reload for custom 16-bit value
TMR1	TMR1[H:L]	GPR	GPR[x,y]	TMR1GIF	Use TMR1 Gate interrupt flag to read data out of TMR1 register
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	TMR2	PR2	TMR2IF	
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y,z]	TMR2 CCP or PWM	PR2 CCPR[H:L] or PWMDC[H:L]	ANY	Frequency generator with 50% duty cycle look up table
CCP	CCPR[H:L]	GPR	GPR[x,y]	CCPxIF	Move data from CCP 16b Capture
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y]	CCP	CCPR[H:L]	ANY	Load Compare value or PWM values into the CCP
GPR/SFR/ Program Flash/Data EEPROM	MEMORY [x,y,z,u,v,w]	CCPx CCPy CCPz	CCPxR[H:L] CCPyR[H:L] CCPzR[H:L]	ANY	Update multiple PWM values at the same time e.g. 3-phase motor control
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x,y,z]	NCO	NCOxINC[U:H:L]	ANY	Frequency Generator look-up table
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	DAC	DACxCON0	ANY	Update DAC values
GPR/SFR/ Program Flash/Data EEPROM	MEMORY[x]	OSCTUNE	OSCTUNE	ANY	Automated Frequency dithering

REGISTER 16-4: ANSELx: ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSELx7	ANSELx6	ANSELx5	ANSELx4	ANSELx3	ANSELx2	ANSELx1	ANSELx0
bit 7							bit 0

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

bit 7-0 **ANSELx<7:0>**: Analog Select on Pins Rx<7:0>

1 = Digital Input buffers are disabled.
0 = ST and TTL input devices are enabled

TABLE 16-5: ANALOG SELECT PORT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELA	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	ANSELA1	ANSELA0
ANSELB	ANSELB7	ANSELB6	ANSELB5	ANSELB4	ANSELB3	ANSELB2	ANSELB1	ANSELB0
ANSELC	ANSELC7	ANSELC6	ANSELC5	ANSELC4	ANSELC3	ANSELC2	ANSELC1	ANSELC0
ANSELD(1)	ANSELD7	ANSELD6	ANSELD5	ANSELD4	ANSELD3	ANSELD2	ANSELD1	ANSELD0
ANSELE(1)	—	—	—	—	—	ANSELE2	ANSELE1	ANSELE0
ANSELF(2)	ANSELF7	ANSELF6	ANSELF5	ANSELF4	ANSELF3	ANSELF2	ANSELF1	ANSELF0

Note 1: Unimplemented in PIC18(L)F26/27K42.
2: Unimplemented in PIC18(L)F26/45/46/47K42.

25.8 Register Definitions: SMT Control

Long bit name prefixes for the Signal Measurement Timer peripherals are shown in **Section 1.3 "Register and Bit naming conventions"**.

TABLE 25-2: LONG BIT NAMES PREFIXES FOR SMT PERIPHERALS

Peripheral	Bit Name Prefix
SMT1	SMT1

REGISTER 25-1: SMT1CON0: SMT CONTROL REGISTER 0

R/W-0/0	U-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
EN(1)	—	STP	WPOL	SPOL	CPOL	PS<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 **EN:** SMT Enable bit(1)
1 = SMT is enabled
0 = SMT is disabled; internal states are reset, clock requests are disabled

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

bit 5 **STP:** SMT Counter Halt Enable bit
When SMT1TMR = SMT1PR:
1 = Counter remains SMT1PR; period match interrupt occurs when clocked
0 = Counter resets to 24'h000000; period match interrupt occurs when clocked

bit 4 **WPOL:** SMT1WIN Input Polarity Control bit
1 = SMT1WIN signal is active-low/falling edge enabled
0 = SMT1WIN signal is active-high/rising edge enabled

bit 3 **SPOL:** SMT1SIG Input Polarity Control bit
1 = SMT1_signal is active-low/falling edge enabled
0 = SMT1_signal is active-high/rising edge enabled

bit 2 **CPOL:** SMT Clock Input Polarity Control bit
1 = SMT1TMR increments on the falling edge of the selected clock signal
0 = SMT1TMR increments on the rising edge of the selected clock signal

bit 1-0 **PS<1:0>:** SMT Prescale Select bits
11 = Prescaler = 1:8
10 = Prescaler = 1:4
01 = Prescaler = 1:2
00 = Prescaler = 1:1

Note 1: Setting EN to '0' does not affect the register contents.

26.2.4 STEERING MODES

In both Synchronous and Asynchronous Steering modes, the modulated input signal can be steered to any combination of four CWG outputs and a fixed-value will be presented on all the outputs not used for the PWM output. Each output has independent polarity, steering, and shutdown options. Dead-band control is not used in either steering mode.

When STRx = 0 (Register 26-5), then the corresponding pin is held at the level defined by OVRx (Register 26-5). When STRx = 1, then the pin is driven by the modulated input signal.

The POLx bits (Register 26-2) control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies to steering modes as described in **Section 26.14 "Register Definitions: CWG Control"**.

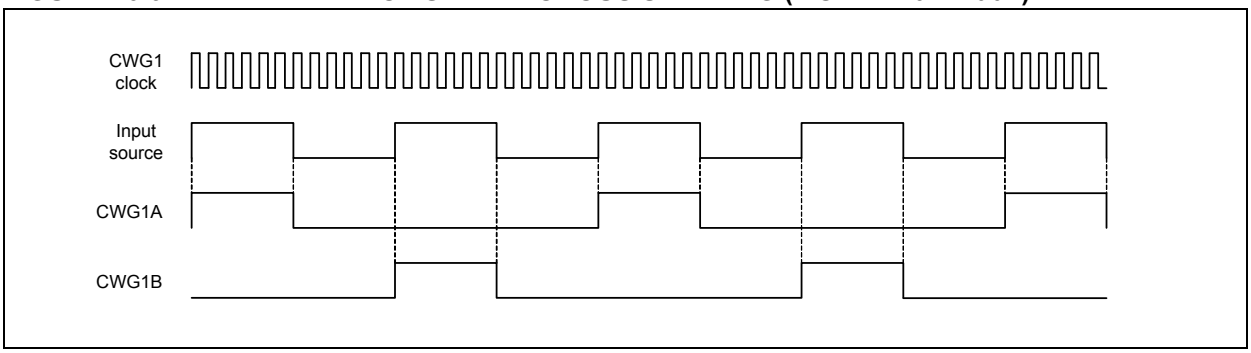
Note:	Only the STRx bits are synchronized; the SDATx (data) bits are not synchronized.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 26.10 "Auto-Shutdown"**. An auto-shutdown event will only affect pins that have STRx = 1.

26.2.4.1 Synchronous Steering Mode

In Synchronous Steering mode (MODE<2:0> bits = 001, Register 26-1), changes to steering selection registers take effect on the next rising edge of the modulated data input (Figure 26-9). In Synchronous Steering mode, the output will always produce a complete waveform.

FIGURE 26-9: EXAMPLE OF SYNCHRONOUS STEERING (MODE<2:0> = 001)



31.2.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error flag bit. A framing error indicates that the Stop bit was not seen at the expected time. The framing error flag is accessed via the FERIF bit in the UxERRIR register. The FERIF bit represents the frame status of the top unread character of the receive FIFO. Therefore, the FERIF bit must be read before reading UxRXB.

The FERIF bit is read-only and only applies to the top unread character of the receive FIFO. A framing error (FERIF = 1) does not preclude reception of additional characters. It is neither necessary nor possible to clear the FERIF bit directly. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERIF bit is cleared when the character at the top of the FIFO does not have a framing error or when all bytes in the receive FIFO have been read. Clearing the ON bit resets the receive FIFO, thereby also clearing the FERIF bit.

A framing error will generate a summary UxERR interrupt when the FERIE bit in the UxERRIE register is set. The summary error is reset when the FERIF bit of the top of the FIFO is '0' or when all FIFO characters have been retrieved.

When FERIE is set, UxRXIF interrupts are suppressed when FERIF is '1'.

31.2.2.5 Receiver Parity Modes

Even and odd parity is automatically detected when the MODE<3:0> bits are set to '0011' and '0010', respectively. Parity modes receive eight data bits and one parity bit for a total of nine bits for each character. The PERIF bit in the UxERRIR register represents the parity error of the top unread character of the receive FIFO rather than the parity bit itself. The parity error must be read before reading the UxRXB register advances the FIFO.

A parity error will generate a summary UxERR interrupt when the PERIE bit in the UxERRIE register is set.The summary error is reset when the PERIF bit of the top of the FIFO is '0' or when all FIFO characters have been retrieved.

When PERIE is set, UxRXIF interrupts are suppressed when PERIF is '1'.

31.2.2.6 Receive FIFO Overflow

When more characters are received than the receive FIFO can hold, the RXFOIF bit in the UxERRIR register is set. The character causing the overflow condition is discarded. The RUNOVF bit in the UxCON2 register determines how the receive circuit responds to characters while the overflow condition persists. When RUNOVF is set, the receive shifter stays synchronized to the incoming data stream by responding to Start, data, and Stop bits. However, all received bytes not already in the FIFO are discarded. When RUNOVF is cleared, the receive shifter ceases operation and Start, data, and Stop bits are ignored. The receive overflow condition is cleared by reading the UxRXB register and clearing the RXFOIF bit. If the UxRXB register is not read to open a space in the FIFO, the next character received will be discarded and cause another overflow condition.

A receive overflow error will generate a summary UxEIF interrupt when the RXFOIE bit in the UxERRIE register is set.

31.2.2.7 Asynchronous Reception Setup

1. Initialize the UxBRGH, UxBRGL register pair and the BRGS bit to achieve the desired baud rate (see **Section 31.17 "UART Baud Rate Generator (BRG)"**).
2. Configure the RXPPS register for the desired RX pin
3. Clear the ANSEL bit for the RX pin (if applicable).
4. Set the MODE<3:0> bits to the desired asynchronous mode.
5. Set the RXPOL bit if the data stream is inverted.
6. Enable the serial port by setting the ON bit.
7. If interrupts are desired, set the UxRXIE bit in the PIEx register and the GIE bits in the INTCON0 register.
8. Enable reception by setting the RXEN bit.
9. The UxRXIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the UxRXIE interrupt enable bit is also set.
10. Read the UxERRIR register to get the error flags.
11. Read the UxRXB register to get the received byte.
12. If an overrun occurred, clear the RXFOIF bit.

34.3 Register Definitions: FVR Control

REGISTER 34-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	RDY(1)	TSEN(3)	TSRNG(3)	CDAFVR<1:0>		ADFVR<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	q = Value depends on condition

bit 7 **EN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **RDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = VOUT = 3VT (High Range)
0 = VOUT = 2VT (Low Range)

bit 3-2 **CDAFVR<1:0>:** Comparator FVR Buffer Gain Selection bits
11 = FVR Buffer 1 Gain is 4x, (4.096V)(2)
10 = FVR Buffer 1 Gain is 2x, (2.048V)(2)
01 = FVR Buffer 1 Gain is 1x, (1.024V)
00 = FVR Buffer 1 is off

bit 1-0 **ADFVR<1:0>:** ADC FVR Buffer Gain Selection bit
11 = FVR Buffer 2 Gain is 4x, (4.096V)(2)
10 = FVR Buffer 2 Gain is 2x, (2.048V)(2)
01 = FVR Buffer 2 Gain is 1x, (1.024V)
00 = FVR Buffer 2 is off

Note 1: FVRRDY is always '1'.
2: Fixed Voltage Reference output cannot exceed VDD.
3: See **Section 35.0 "Temperature Indicator Module"** for additional information.

TABLE 34-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	EN	RDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		597

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used with the Fixed Voltage Reference.

41.0 INSTRUCTION SET SUMMARY

PIC18(L)F26/27/45/46/47/55/56/57K42 devices incorporate the standard set of PIC18 core instructions, as well as an extended set of instructions, for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

41.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from these PIC® MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two-program memory locations and two that require three-program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 41-3 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 41-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator ‘f’ specifies which file register is to be used by the instruction. The destination designator ‘d’ specifies where the result of the operation is to be placed. If ‘d’ is zero, the result is placed in the WREG register. If ‘d’ is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator ‘f’ represents the number of the file in which the bit is located.

The literal instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The control instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the four MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 41-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 41-3, lists the standard instructions recognized by the Microchip Assembler (MPASM™).

Section 41.1.1 “Standard Instruction Set” provides a description of each instruction.

DCFSNZ **Decrement f, skip if not 0**

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

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

0100	11da	ffff	ffff

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).
If the result is not '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead, making it 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 ≤ 95 (5Fh). See **Section 41.2.3 "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	Write to destination

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    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```
Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP – 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

GOTO **Unconditional Branch**

Syntax: GOTO k

Operands: $0 \le k \le 1048575$

Operation: k → PC<20:1>

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	1111	k_7kkk	$kkkk_0$
2nd word(k<19:8>)	1111	$k_{19}kkk$	kkkk	$kkkk_8$

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a 2-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

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

Example:
```
GOTO THERE
```
After Instruction
PC = Address (THERE)

41.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note: Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing mode (**Section 4.8.1 "Indexed Addressing with Literal Offset"**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank ('a' = 0), or in a GPR bank designated by the BSR ('a' = 1). When the extended instruction set is enabled and 'a' = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 41.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands"**).

Although the Indexed Literal Offset Addressing mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset Addressing mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

41.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument, 'f', in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value, 'k'. As already noted, this occurs only when 'f' is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[ ]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within brackets, will generate an error in the MPASM assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled) when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM assembler.

The destination argument, 'd', functions as before.

In the latest versions of the MPASM™ assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, /y, or the PE directive in the source listing.

41.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18(L)F2x/4x/5xK42, it is very important to consider the type of code. A large, re-entrant application that is written in 'C' and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

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
3F66h	PWM7CON	EN	—	OUT	POL	—	—	—	—	358
3F65h	PWM7DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	360
3F65h	PWM7DCH	DC								360
3F64h	PWM7DCL	DC1	DC0	—	—	—	—	—	—	360
3F64h	PWM7DCL	DC		—	—	—	—	—	—	360
3F63h	—	Unimplemented								
3F62h	PWM8CON	EN	—	OUT	POL	—	—	—	—	358
3F61h	PWM8DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	360
3F61h	PWM8DCH	DC								360
3F60h	PWM8DCL	DC1	DC0	—	—	—	—	—	—	360
3F60h	PWM8DCL	DC		—	—	—	—	—	—	360
3F5Fh	CCPTMRS1	P8TSEL		P7TSEL		P6TSEL		P5TSEL		359
3F5Eh	CCPTMRS0	C4TSEL		C3TSEL		C2TSEL		C1TSEL		359
3F5Dh - 3F5Bh	—	Unimplemented								
3F5Ah	CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F59h	CWG1AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F58h	CWG1AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F57h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F56h	CWG1CON0	EN	LD	—	—	—	MODE			424
3F55h	CWG1DBF	—	—	DBF						431
3F54h	CWG1DBR	—	—	DBR						431
3F53h	CWG1ISM	—	—	—	—	IS				427
3F52h	CWG1CLK	—	—	—	—	—	—	—	CS	426
3F51h	CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F50h	CWG2AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F4Fh	CWG2AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F4Eh	CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F4Dh	CWG2CON0	EN	LD	—	—	—	MODE			424
3F4Ch	CWG2DBF	—	—	DBF						431
3F4Bh	CWG2DBR	—	—	DBR						431
3F4Ah	CWG2ISM	—	—	—	—	IS				427
3F49h	CWG2CLK	—	—	—	—	—	—	—	CS	426
3F48h	CWG3STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
3F47h	CWG3AS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
3F46h	CWG3AS0	SHUTDOWN	REN	LSBD		LSAC		—	—	429
3F45h	CWG3CON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
3F44h	CWG3CON0	EN	LD	—	—	—	MODE			424
3F43h	CWG3DBF	—	—	DBF						431
3F42h	CWG3DBR	—	—	DBR						431
3F41h	CWG3ISM	—	—	—	—	IS				427
3F40h	CWG3CLK	—	—	—	—	—	—	—	CS	426
3F3Fh	NCO1CLK	PWS			—	CKS				454
3F3Eh	NCO1CON	EN	—	OUT	POL	—	—	—	PFM	453
3F3Dh	NCO1INCU	INC								457
3F3Ch	NCO1INCH	INC								456
3F3Bh	NCO1INCL	INC								456

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.

TABLE 44-6: I/O PORTS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O PORT:					
D300		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D301			—	—	0.15 VDD	V	1.8V ≤ VDD < 4.5V
D302		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D303		with I2C levels	—	—	0.3 VDD	V	
D304		with SMBus 2.0	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D305		with SMBus 3.0	—	—	0.8	V	1.8V ≤ VDD ≤ 5.5V
D306		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O PORT:					
D320		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D321			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD < 4.5V
D322		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
D323		with I2C levels	0.7 VDD	—	—	V	
D324		with SMBus 2.0	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D325		with SMBus 3.0	1.35	—	—	V	1.8V ≤ VDD ≤ 5.5V
D326		$\overline{MCLR}$	0.7 VDD	—	—	V	
	IIL	**Input Leakage Current(1)**					
D340		I/O Ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
D341			—	± 5	± 1000	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C
D342		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
	IPUR	**Weak Pull-up Current**					
D350			25	120	200	µA	VDD = 3.0V, VPIN = VSS
	VOL	**Output Low Voltage**					
D360		I/O ports	—	—	0.6	V	IOL = 10.0mA, VDD = 3.0V
	VOH	**Output High Voltage**					
D370		I/O ports	VDD - 0.7	—	—	V	IOH = 6.0 mA, VDD = 3.0V
D380	CIO	**All I/O pins**	—	5	50	pF	

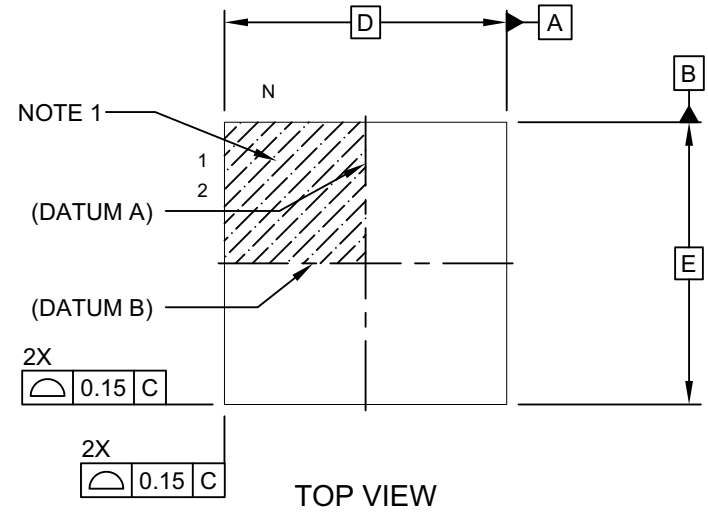
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Negative current is defined as current sourced by the pin.

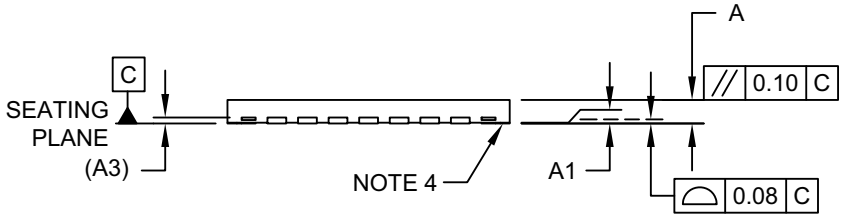
2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

28-Lead Plastic Quad Flat, No Lead Package (MX) - 6x6x0.5mm Body [UQFN]
Ultra-Thin with 0.40 x 0.60 mm Terminal Width/Length and Corner Anchors

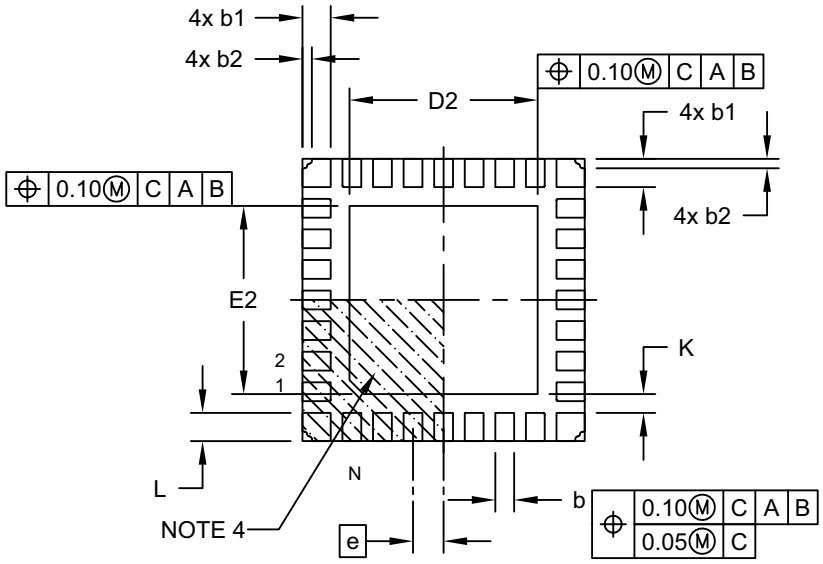
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

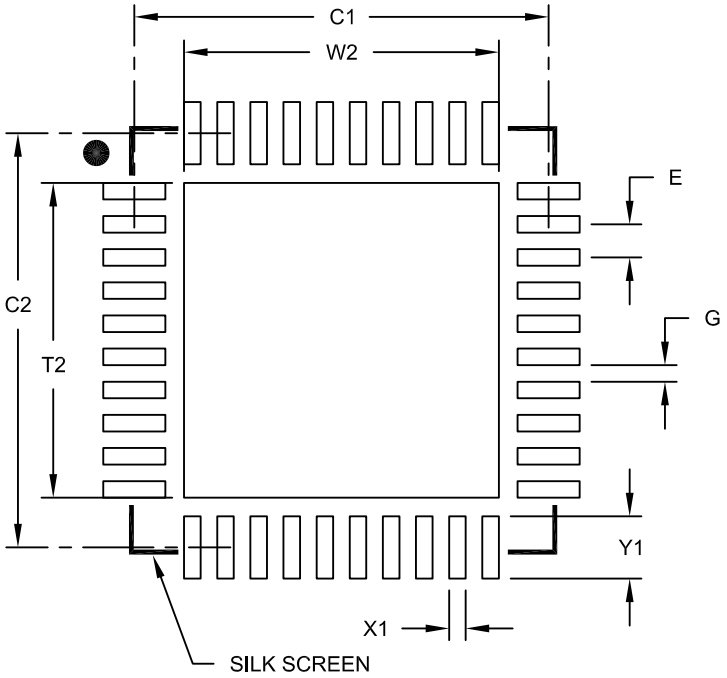


BOTTOM VIEW

Microchip Technology Drawing C04-0209 Rev C Sheet 1 of 2

40-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) - 5x5 mm Body [UQFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	W2			3.80
Optional Center Pad Length	T2			3.80
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X40)	X1			0.20
Contact Pad Length (X40)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2156B