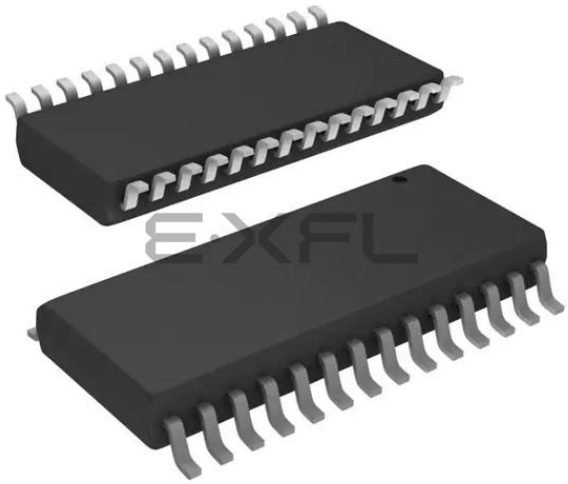


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

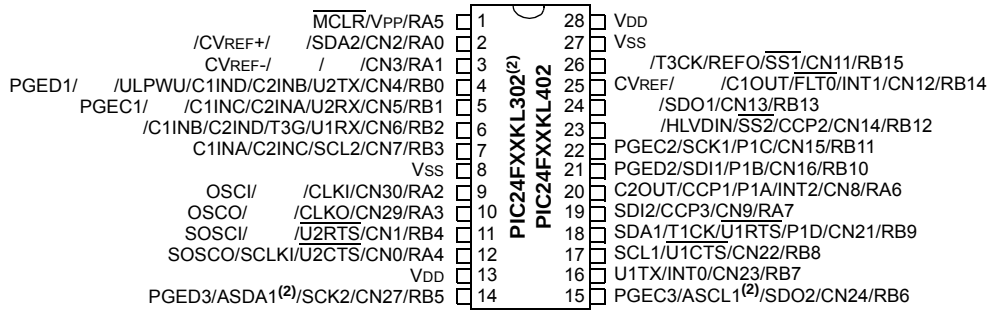
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

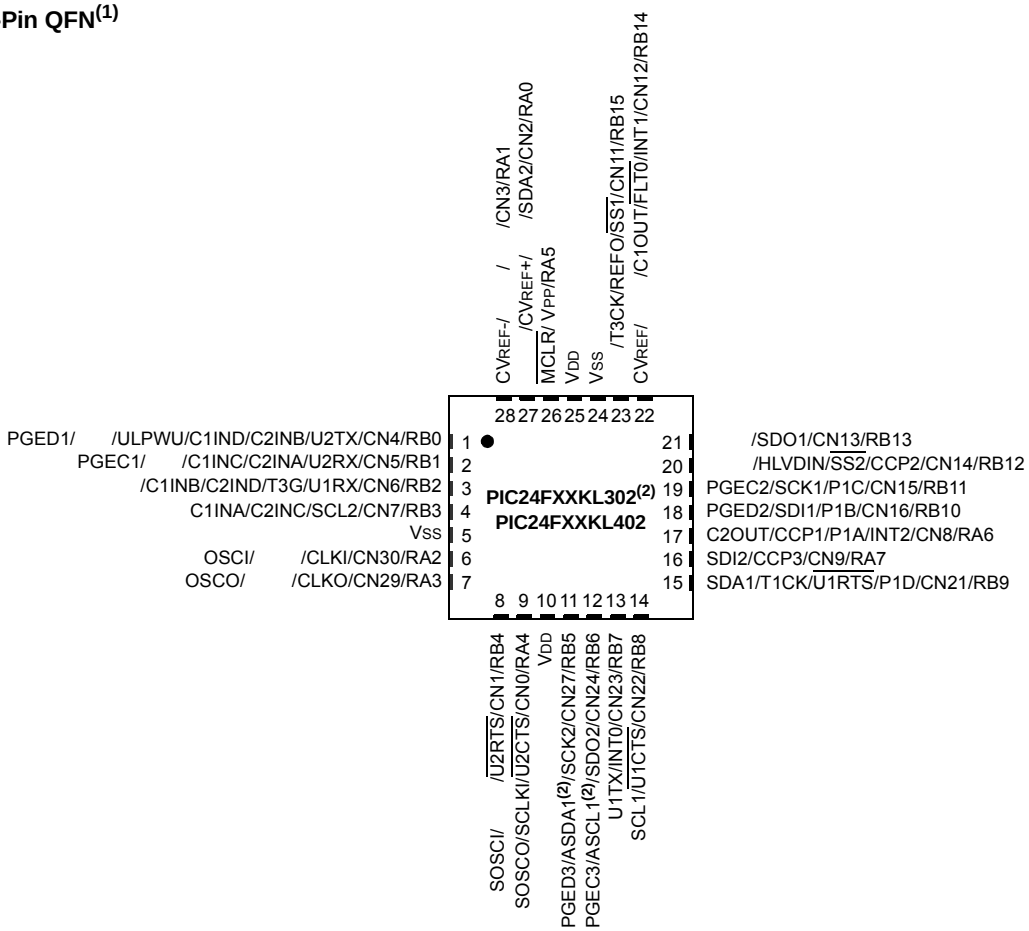
Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16kl402-e-so

Pin Diagrams: PIC24FXXKL302/402

28-Pin SPDIP/SSOP/SOIC(1)



28-Pin QFN(1)



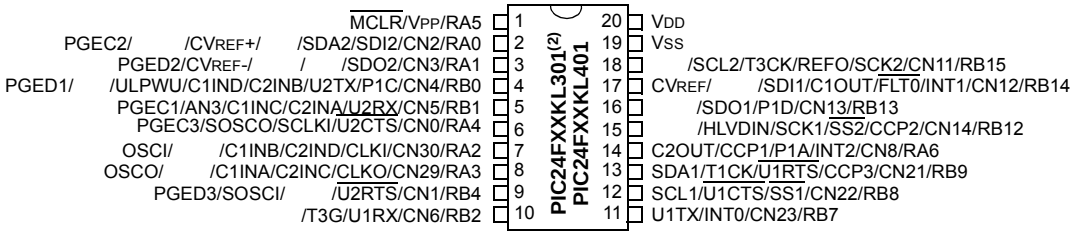
Contact your Microchip sales team for Chip Scale Package (CSP) availability.

Note 1: Analog features (indicated in) are not available on PIC24FXXKL302 devices.

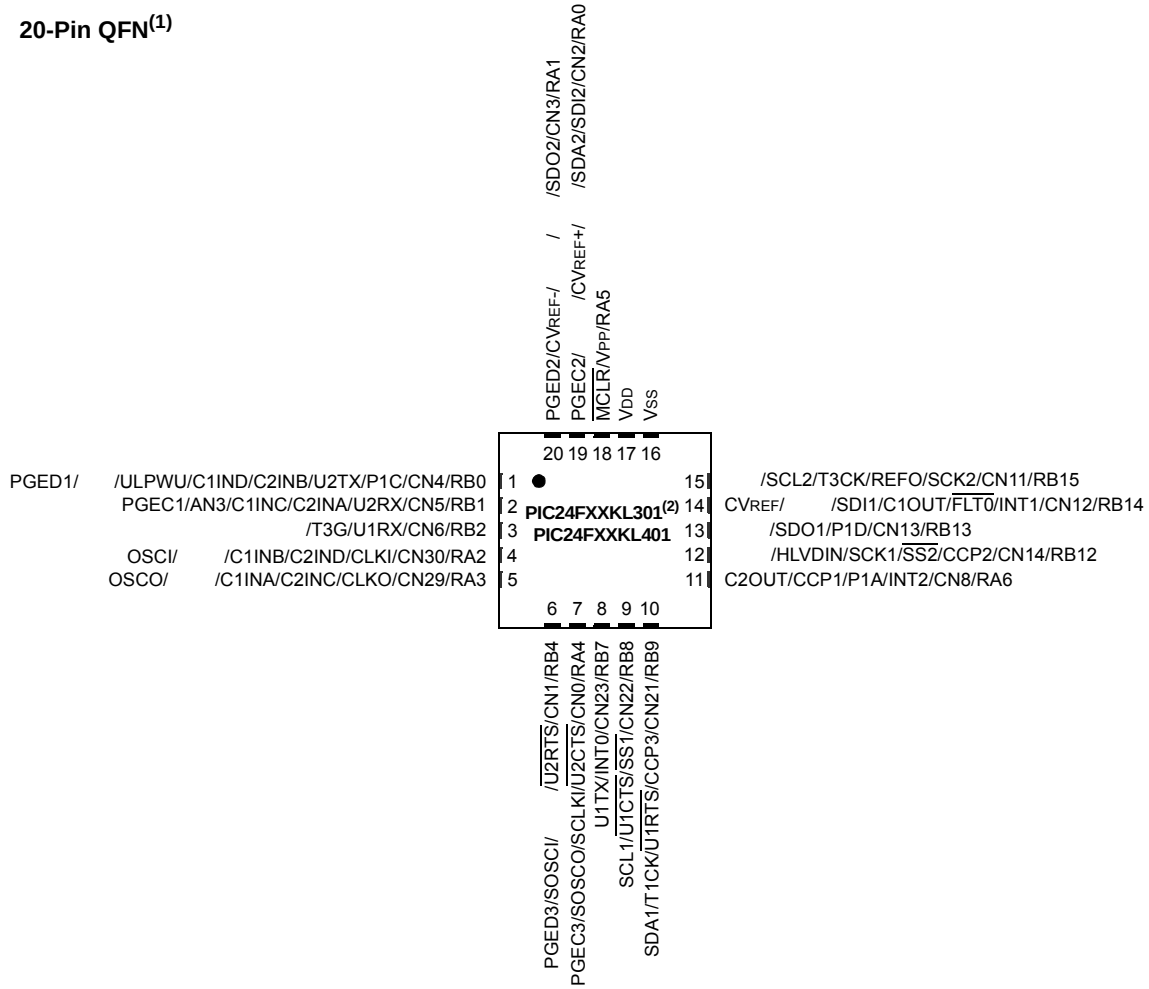
2: Alternate location for I^2C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

Pin Diagrams: PIC24FXXKL301/401

20-Pin PDIP/SSOP/SOIC(1)



20-Pin QFN(1)

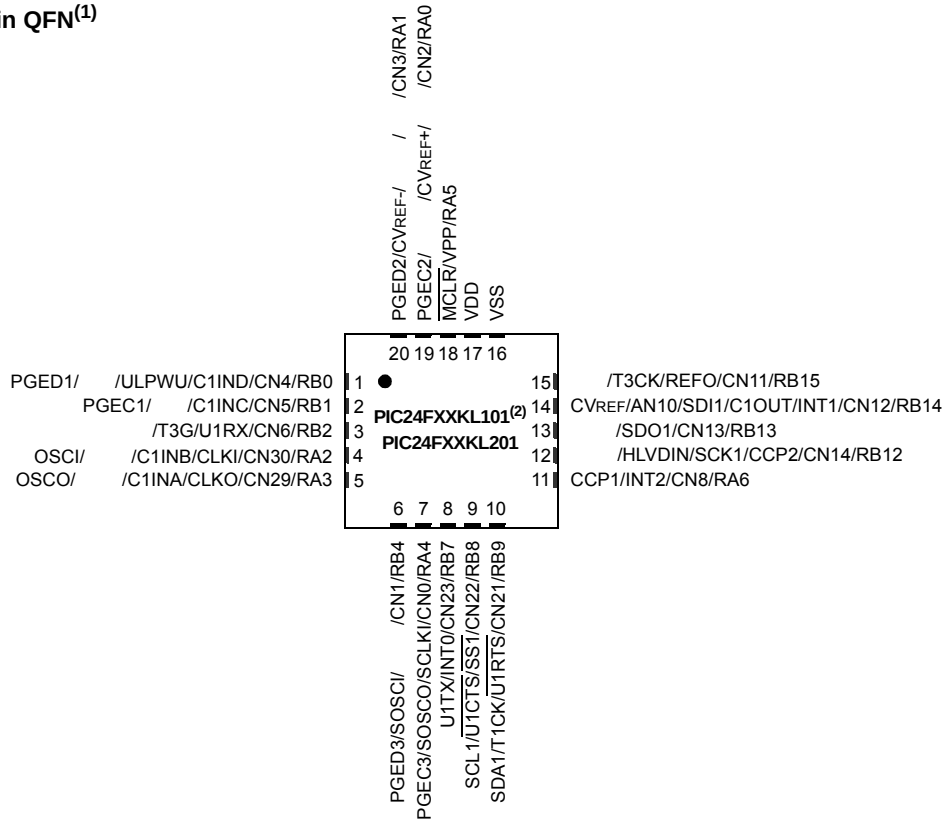


Note 1: Analog features (indicated in) are not available on PIC24FXXKL301 devices.

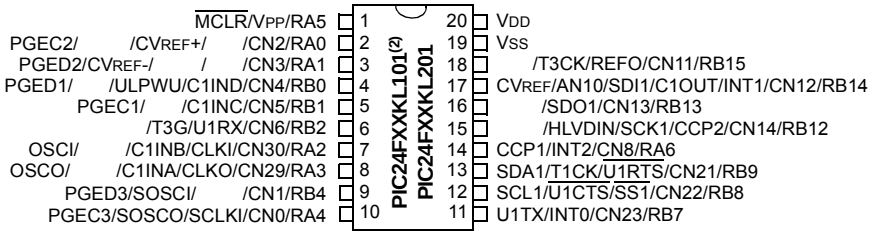
2: Alternate location for I^2C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

Pin Diagrams: PIC24FXXKL10X/20X

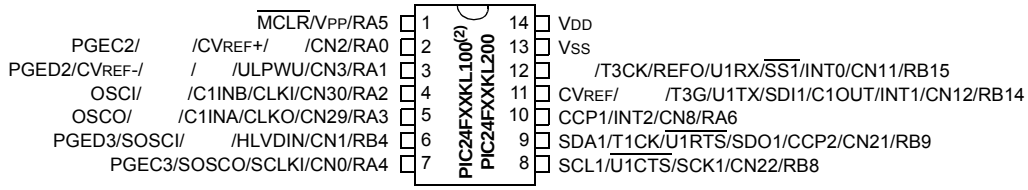
20-Pin QFN(1)



20-Pin PDIP/SSOP/SOIC(1)



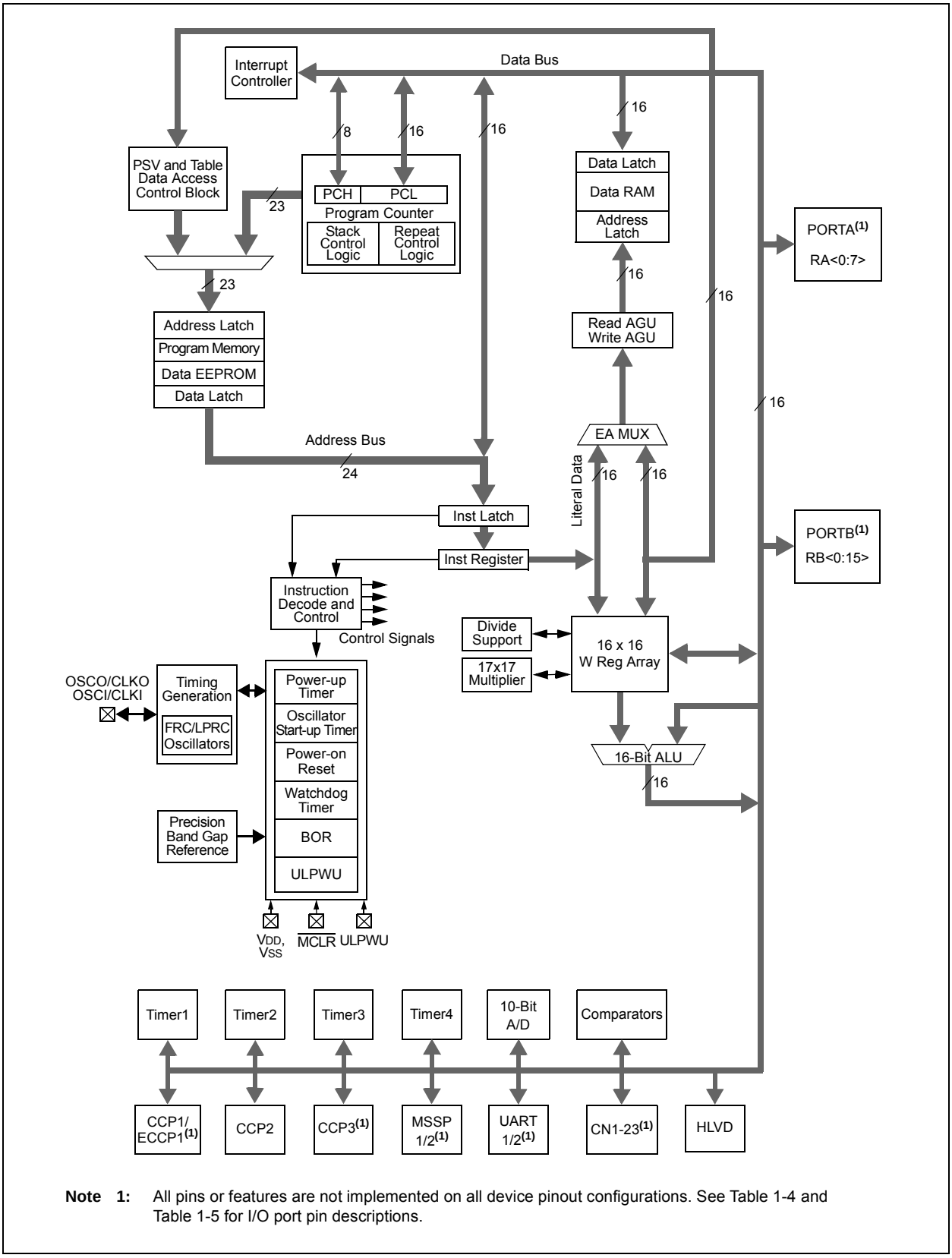
14-Pin PDIP/TSSOP(1)



Note 1: Analog features (indicated in) are not available on PIC24FXXKL100/101 devices.

2: Alternate location for I^2C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

FIGURE 1-1: PIC24F16KL402 FAMILY GENERAL BLOCK DIAGRAM



Note 1: All pins or features are not implemented on all device pinout configurations. See Table 1-4 and Table 1-5 for I/O port pin descriptions.

7.2.1 POR AND LONG OSCILLATOR START-UP TIMES

The oscillator start-up circuitry and its associated delay timers are not linked to the device Reset delays that occur at power-up. Some crystal circuits (especially low-frequency crystals) will have a relatively long start-up time. Therefore, one or more of the following conditions is possible after SYSRST is released:

- The oscillator circuit has not begun to oscillate.
- The Oscillator Start-up Timer (OST) has not expired (if a crystal oscillator is used).
- The PLL has not achieved a lock (if PLL is used).

The device will not begin to execute code until a valid clock source has been released to the system. Therefore, the oscillator and PLL start-up delays must be considered when the Reset delay time must be known.

7.2.2 FAIL-SAFE CLOCK MONITOR (FSCM) AND DEVICE RESETS

If the FSCM is enabled, it will begin to monitor the system clock source when SYSRST is released. If a valid clock source is not available at this time, the device will automatically switch to the FRC oscillator and the user can switch to the desired crystal oscillator in the Trap Service Routine (TSR).

7.3 Special Function Register Reset States

Most of the Special Function Registers (SFRs) associated with the PIC24F CPU and peripherals are reset to a particular value at a device Reset. The SFRs are grouped by their peripheral or CPU function and their Reset values are specified in each section of this manual.

The Reset value for each SFR does not depend on the type of Reset, with the exception of four registers. The Reset value for the Reset Control register, RCON, will depend on the type of device Reset. The Reset value for the Oscillator Control register, OSCCON, will depend on the type of Reset and the programmed values of the FNOSC bits in the Flash Configuration Word (FOSCSEL); see Table 7-2. The RCFGCAL and NVMCON registers are only affected by a POR.

7.4 Brown-out Reset (BOR)

PIC24F16KL402 family devices implement a BOR circuit, which provides the user several configuration and power-saving options. The BOR is controlled by the BORV<1:0> and BOREN<1:0> Configuration bits (FPOR<6:5,1:0>). There are a total of four BOR configurations, which are provided in Table 7-3.

The BOR threshold is set by the BORV<1:0> bits. If BOR is enabled (any values of BOREN<1:0>, except '00'), any drop of V_{DD} below the set threshold point will reset the device. The chip will remain in BOR until V_{DD} rises above the threshold.

If the Power-up Timer is enabled, it will be invoked after V_{DD} rises above the threshold. Then, it will keep the chip in Reset for an additional time delay, T_{PWRT}, if V_{DD} drops below the threshold while the power-up timer is running. The chip goes back into a BOR and the Power-up Timer will be initialized. Once V_{DD} rises above the threshold, the Power-up Timer will execute the additional time delay.

BOR and the Power-up Timer (PWRT) are independently configured. Enabling the BOR Reset does not automatically enable the PWRT.

7.4.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = 01, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<13>). Setting SBOREN enables the BOR to function, as previously described. Clearing the SBOREN disables the BOR entirely. The SBOREN bit only operates in this mode; otherwise, it is read as '0'.

Placing BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change the BOR configuration. It also allows the user to tailor the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note:	Even when the BOR is under software control, the BOR Reset voltage level is still set by the BORV<1:0> Configuration bits; it can not be changed in software.

8.0 INTERRUPT CONTROLLER

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the Interrupt Controller, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Interrupts"** (DS39707).

The PIC24F interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the CPU. It has the following features:

- Up to eight processor exceptions and software traps
- Seven user-selectable priority levels
- Interrupt Vector Table (IVT) with up to 118 vectors
- Unique vector for each interrupt or exception source
- Fixed priority within a specified user priority level
- Alternate Interrupt Vector Table (AIVT) for debug support
- Fixed interrupt entry and return latencies

8.1 Interrupt Vector Table (IVT)

The IVT is shown in Figure 8-1. The IVT resides in the program memory, starting at location, 000004h. The IVT contains 126 vectors, consisting of eight non-maskable trap vectors, plus up to 118 sources of interrupt. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

Interrupt vectors are prioritized in terms of their natural priority; this is linked to their position in the vector table. All other things being equal, lower addresses have a higher natural priority. For example, the interrupt associated with vector 0 will take priority over interrupts at any other vector address.

PIC24F16KL402 family devices implement 32 non-maskable traps and unique interrupts; these are summarized in Table 8-1 and Table 8-2.

8.1.1 ALTERNATE INTERRUPT VECTOR TABLE (AIVT)

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 8-1. Access to the AIVT is provided by the ALTIVT control bit (INTCON2<15>). If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors.

The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not needed, the AIVT should be programmed with the same addresses used in the IVT.

8.2 Reset Sequence

A device Reset is not a true exception, because the interrupt controller is not involved in the Reset process. The PIC24F devices clear their registers in response to a Reset, which forces the Program Counter (PC) to zero. The microcontroller then begins program execution at location, 000000h. The user programs a `GOTO` instruction at the Reset address, which redirects the program execution to the appropriate start-up routine.

Note:	Any unimplemented or unused vector locations in the IVT and AIVT should be programmed with the address of a default interrupt handler routine that contains a `RESET` instruction.

REGISTER 8-9: IFS4: INTERRUPT FLAG STATUS REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	HLVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	U2ERIF(1)	U1ERIF	—
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **U2ERIF:** UART2 Error Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U1ERIF:** UART1 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

Note 1: This bit is unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

REGISTER 8-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	ULPWUIF
bit 7							bit 0

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

bit 15-1 **Unimplemented:** Read as '0'

bit 0 **ULPWUIF:** Ultra Low-Power Wake-up Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 8-15: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	HLVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	U2ERIE(1)	U1ERIE	—
bit 7							bit 0

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

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **U2ERIE:** UART2 Error Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **U1ERIE:** UART1 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

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

Note 1: This bit is unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

REGISTER 8-16: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	ULPWUIE
bit 7							bit 0

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

bit 15-1 **Unimplemented:** Read as '0'

bit 0 **ULPWUIE:** Ultra Low-Power Wake-up Interrupt Enable Bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 8-17: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **CCP1IP<2:0>:** Capture/Compare/PWM1 Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 8-20: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	NVMIP2	NVMIP1	NVMIP0	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **NVMIP<2:0>:** NVM Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11-7 **Unimplemented:** Read as ‘0’

bit 6-4 **AD1IP<2:0>:** A/D Conversion Complete Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

10.3 Ultra Low-Power Wake-up

The Ultra Low-Power Wake-up (ULPWU) on pin, RB0, allows a slow falling voltage to generate an interrupt without excess current consumption. This feature provides a low-power technique for periodically waking up the device from Sleep mode.

To use this feature:

1. Charge the capacitor on RB0 by configuring the RB0 pin to an output and setting it to '1'.
2. Stop charging the capacitor by configuring RB0 as an input.
3. Discharge the capacitor by setting the ULPEN and ULPSINK bits in the ULPWCON register.
4. Configure Sleep mode.
5. Enter Sleep mode.

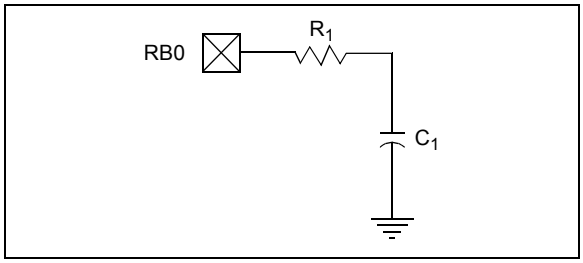
The time-out is dependent on the discharge time of the RC circuit on RB0. When the voltage on RB0 drops below V_{IL}, the device wakes up and executes the next instruction.

When the ULPWU module wakes the device from Sleep mode, the ULPWUIF bit (IFS5<0>) is set. Software can check this bit upon wake-up to determine the wake-up source.

See Example 10-2 for initializing the ULPWU module.

A series resistor, between RB0 and the external capacitor, provides overcurrent protection for the RB0/AN2/ULPWU pin and enables software calibration of the time-out (see Figure 10-1).

FIGURE 10-1: SERIES RESISTOR



A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired delay in Sleep. This technique compensates for the affects of temperature, voltage and component accuracy. The peripheral can also be configured as a simple, programmable Low-Voltage Detect (LVD) or temperature sensor.

EXAMPLE 10-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
//*****************************************************************************
// 1. Charge the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 0;
    LATBbits.LATB0 = 1;
    for(i = 0; i < 10000; i++) Nop();
//*****************************************************************************
//2. Stop Charging the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 1;
//*****************************************************************************
//3. Enable ULPWU Interrupt
//*****************************************************************************
IFS5bits.ULPWUIF = 0;
IEC5bits.ULPWUIE = 1;
IPC20bits.ULPWUIP = 0x7;
//*****************************************************************************
//4. Enable the Ultra Low Power Wakeup module and allow capacitor discharge
//*****************************************************************************
    ULPWCONbits.ULPEN = 1;
    ULPWCONbits.ULPSINK = 1;
//*****************************************************************************
//5. Enter Sleep Mode
//*****************************************************************************
    Sleep();
//for Sleep, execution will resume here
```

NOTES:

REGISTER 16-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ECCPASE:** ECCP1 Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>:** ECCP1 Auto-Shutdown Source Select bits
111 = VIL on $\overline{FLT0}$ pin, or either C1OUT or C2OUT is high
110 = VIL on $\overline{FLT0}$ pin or C2OUT comparator output is high
101 = VIL on $\overline{FLT0}$ pin or C1OUT comparator output is high
100 = VIL on $\overline{FLT0}$ pin
011 = Either C1OUT or C2OUT is high
010 = C2OUT comparator output is high
001 = C1OUT comparator output is high
000 = Auto-shutdown is disabled

bit 3-2 **PSSAC<1:0>:** P1A and P1C Pins Shutdown State Control bits
1x = P1A and P1C pins tri-state
01 = Drive pins, P1A and P1C, to '1'
00 = Drive pins, P1A and P1C, to '0'

bit 1-0 **PSSBD<1:0>:** P1B and P1D Pins Shutdown State Control bits
1x = P1B and P1D pins tri-state
01 = Drive pins, P1B and P1D, to '1'
00 = Drive pins, P1B and P1D, to '0'

Note 1: This register is implemented only on PIC24FXXKL40X/30X devices.

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCPASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 16-6: CCPTMRS0: CCP TIMER SELECT CONTROL REGISTER 0(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0	R/W-0
—	C3TSEL0	—	—	C2TSEL0	—	—	C1TSEL0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6 **C3TSEL0:** CCP3 Timer Selection bit
1 = CCP3 uses TMR3/TMR4
0 = CCP3 uses TMR3/TMR2

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3 **C2TSEL0:** CCP2 Timer Selection bit
1 = CCP2 uses TMR3/TMR4
0 = CCP2 uses TMR3/TMR2

bit 2-1 **Unimplemented:** Read as ‘0’

bit 0 **C1TSEL0:** CCP1/ECCP1 Timer Selection bit
1 = CCP1/ECCP1 uses TMR3/TMR4
0 = CCP1/ECCP1 uses TMR3/TMR2

Note 1: This register is unimplemented on PIC24FXXKL20X/10X devices; maintain as ‘0’.

REGISTER 18-1: UxMODE: UARTx MODE REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0(2)	R/W-0(2)
UARTEN	—	USIDL	IREN(1)	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/C-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

Legend:	C = Clearable bit	HC = Hardware Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **UARTEN:** UARTx Enable bit
- 1 = UARTx is enabled; all UARTx pins are controlled by UARTx as defined by UEN<1:0>
- 0 = UARTx is disabled; all UARTx pins are controlled by port latches, UARTx power consumption is minimal

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** UARTx Stop in Idle Mode bit
- 1 = Discontinues module operation when device enters Idle mode
- 0 = Continues module operation in Idle mode

bit 12 **IREN:** IrDA® Encoder and Decoder Enable bit(1)
- 1 = IrDA encoder and decoder are enabled
- 0 = IrDA encoder and decoder are disabled

bit 11 **RTSMD:** Mode Selection for $\overline{\text{UxRTS}}$ Pin bit
- 1 = $\overline{\text{UxRTS}}$ pin is in Simplex mode
- 0 = $\overline{\text{UxRTS}}$ pin is in Flow Control mode

bit 10 **Unimplemented:** Read as '0'

bit 9-8 **UEN<1:0>:** UARTx Enable bits(2)
- 11 = UxTX, UxRX and UxBCLK pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 10 = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
- 01 = UxTX, UxRX and $\overline{\text{UxRTS}}$ pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 00 = UxTX and UxRX pins are enabled and used; $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$/UxBCLK pins are controlled by port latches

bit 7 **WAKE:** Wake-up on Start Bit Detect During Sleep Mode Enable bit
- 1 = UARTx will continue to sample the UxRX pin; interrupt is generated on the falling edge, bit is cleared in hardware on the following rising edge
- 0 = No wake-up is enabled

bit 6 **LPBACK:** UARTx Loopback Mode Select bit
- 1 = Enables Loopback mode
- 0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto-Baud Enable bit
- 1 = Enables baud rate measurement on the next character – requires reception of a Sync field (55h); cleared in hardware upon completion
- 0 = Baud rate measurement is disabled or completed

bit 4 **RXINV:** Receive Polarity Inversion bit
- 1 = UxRX Idle state is '0'
- 0 = UxRX Idle state is '1'

Note 1: This feature is is only available for the 16x BRG mode (BRGH = 0).
2: Bit availability depends on pin availability.

23.4 Watchdog Timer (WDT)

For the PIC24F16KL402 family of devices, the WDT is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

The nominal WDT clock source from LPRC is 31 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the FWPSA Configuration bit. With a 31 kHz input, the prescaler yields a nominal WDT time-out period (T_{WDT}) of 1 ms in 5-bit mode or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the Configuration bits, WDTPS<3:0> (FWDT<3:0>), which allow the selection of a total of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler time-out periods, ranges from 1 ms to 131 seconds can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSCx bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a PWRSAV instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a CLRWDT instruction during normal execution

If the WDT is enabled in hardware (FWDTEN<1:0> = 11), it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE bits (RCON<3:2>) will need to be cleared in software after the device wakes up.

The WDT Time-out Flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

Note:	The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

23.4.1 WINDOWED OPERATION

The Watchdog Timer has an optional Fixed Window mode of operation. In this Windowed mode, CLRWDT instructions can only reset the WDT during the last 1/4 of the programmed WDT period. A CLRWDT instruction, executed before that window, causes a WDT Reset similar to a WDT time-out.

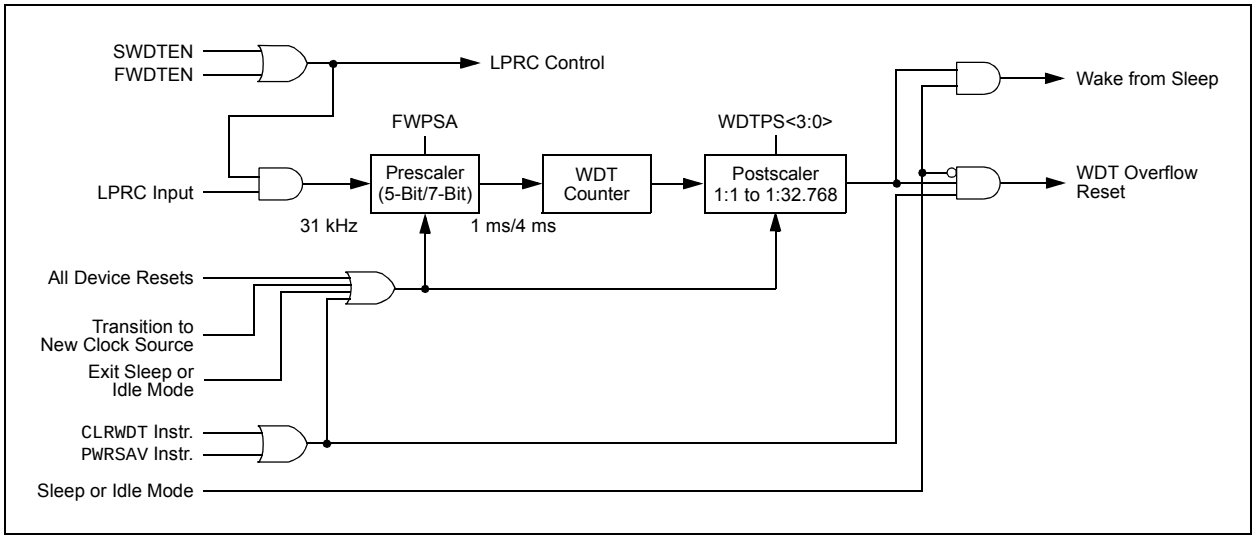
Windowed WDT mode is enabled by programming the Configuration bit, WINDIS (FWDT<6>), to '0'.

23.4.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN<1:0> Configuration bits. When both the FWDTEN<1:0> Configuration bits are set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN<1:0> Configuration bits have been programmed to '10'. The WDT is enabled in software by setting the SWDTEN control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments, and disable the WDT during non-critical segments, for maximum power savings. When the FWTEN<1:0> bits are set to '01', the WDT is enabled only in Run and Idle modes, and is disabled in Sleep. Software control of the WDT SWDTEN bit (RCON<5>) is disabled with this setting.

FIGURE 23-1: WDT BLOCK DIAGRAM



24.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

24.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

24.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

24.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
RCALL	RCALL	Expr	Relative Call	1	2	None
	RCALL	Wn	Computed Call	1	2	None
REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
	REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
RESET	RESET		Software Device Reset	1	1	None
RETFIE	RETFIE		Return from Interrupt	1	3 (2)	None
RETLW	RETLW	#lit10,Wn	Return with Literal in Wn	1	3 (2)	None
RETURN	RETURN		Return from Subroutine	1	3 (2)	None
RLC	RLC	f	f = Rotate Left through Carry f	1	1	C, N, Z
	RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
	RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC	RRC	f	f = Rotate Right through Carry f	1	1	C, N, Z
	RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
	RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
SE	SE	Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM	SETM	f	f = FFFFh	1	1	None
	SETM	WREG	WREG = FFFFh	1	1	None
	SETM	Ws	Ws = FFFFh	1	1	None
SL	SL	f	f = Left Shift f	1	1	C, N, OV, Z
	SL	f,WREG	WREG = Left Shift f	1	1	C, N, OV, Z
	SL	Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
	SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
	SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
SUB	SUB	f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB	f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	#lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SUBR	SUBR	f	f = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	f,WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR	f	f = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
	SWAP	Wn	Wn = Byte Swap Wn	1	1	None

TABLE 26-6: DC CHARACTERISTICS: OPERATING CURRENT (I_{DD})(2)

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
IDD Current						
DC20	0.154	0.350	mA	1.8V	+85V°C	0.5 MIPS, FOSC = 1 MHz
	0.301	0.630		3.3V		
	—	.500	mA	1.8V	+125°C	
	—	.800		3.3V		
DC22	0.300	—	mA	1.8V	+85°C	1 MIPS, FOSC = 2 MHz
	0.585	—		3.3V		
DC24	7.76	12.0	mA	3.3V	+85°C	16 MIPS, FOSC = 32 MHz
	—	18.0		3.3V	+125°C	
DC26	1.44	—	mA	1.8V	+85°C	FRC (4 MIPS), FOSC = 8 MHz
	2.71	—		3.3V		
DC30	4.00	28.0	µA	1.8V	+85°C	LPRC (15.5 KIPS), FOSC = 31 kHz
	9.00	55.0		3.3V		
	—	45.0	µA	1.8V	+125°C	
	—	90.0		3.3V		

Note 1: Data in the Typical column is at 3.3V, +25°C, unless otherwise stated.

2: IDD is measured with all peripherals disabled. All I/Os are configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled.

TABLE 26-7: DC CHARACTERISTICS: IDLE CURRENT (I_{IDLE})(2)

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Idle Current (IIDLE)						
DC40	0.035	0.080	mA	1.8V	+85°C	0.5 MIPS, FOSC = 1 MHz
	0.077	0.150		3.3V		
	—	0.160	mA	1.8V	+125°C	
	—	0.300		3.3V		
DC42	0.076	—	mA	1.8V	+85°C	1 MIPS, FOSC = 2 MHz
	0.146	—		3.3V		
DC44	2.52	3.20	mA	3.3V	+85°C	16 MIPS, FOSC = 32 MHz
	—	5.00	mA	3.3V	+125°C	
DC46	0.45	—	mA	1.8V	+85°C	FRC (4 MIPS), FOSC = 8 MHz
	0.76	—	mA	3.3V		
DC50	0.87	18.0	µA	1.8V	+85°C	LPRC (15.5 KIPS), FOSC = 31 kHz
	1.55	40.0	µA	3.3V		
	—	27.0	µA	1.8V	+125°C	
	—	50.0	µA	3.3V		

Note 1: Data in the Typical column is at 3.3V, +25°C, unless otherwise stated.

2: IIDLE is measured with all I/Os configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled.

T

U

W