



# PIC24F16KL402 FAMILY

## Low-Power, Low-Cost, General Purpose 16-Bit Flash Microcontrollers with XLP Technology

### Power Management Modes:

- Run – CPU, Flash, SRAM and Peripherals On
- Doze – CPU Clock Runs Slower than Peripherals
- Idle – CPU Off, SRAM and Peripherals On
- Sleep – CPU, Flash and Peripherals Off and SRAM On
- Low-Power Consumption:
  - Run mode currents of 150  $\mu$ A/MHz typical at 1.8V
  - Idle mode currents under 80  $\mu$ A/MHz at 1.8V
  - Sleep mode currents as low as 30 nA at +25°C
  - Watchdog Timer as low as 210 nA at +25°C

### High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator:
  - 4x PLL option
  - Multiple divide options
- 17-Bit x 17-Bit Single-Cycle Hardware Fractional/integer Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture (ISA):
  - 76 base instructions
  - Flexible addressing modes
- Linear Program Memory Addressing
- Linear Data Memory Addressing
- Two Address Generation Units (AGU) for Separate Read and Write Addressing of Data Memory

### Peripheral Features:

- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to Three External Interrupt Sources
- Two 16-Bit Timer/Counters with Selectable Clock Sources
- Up to Two 8-Bit Timers/Counters with Programmable Prescalers
- Two Capture/Compare/PWM (CCP) modules:
  - Modules automatically configure and drive I/O
  - 16-bit Capture with max. resolution 40 ns
  - 16-bit Compare with max. resolution 83.3 ns
  - 1-bit to 10-bit PWM resolution
- Up to One Enhanced CCP module:
  - Backward compatible with CCP
  - 1, 2 or 4 PWM outputs
  - Programmable dead time
  - Auto-shutdown on external event
- Up to Two Master Synchronous Serial Port modules (MSSPs) with Two Modes of Operation:
  - 3-wire SPI (all four modes)
  - I<sup>2</sup>C™ Master, Multi-Master and Slave modes and 7-Bit/10-Bit Addressing
- Up to Two UART modules:
  - Supports RS-485, RS-232 and LIN/J2602
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect (ABD)
  - Two-byte transmit and receive FIFO buffers

Device	Pins	Memory			Peripherals						Ultra Low-Power Wake-up
		Flash Program (bytes)	Data (bytes)	Data EEPROM (bytes)	10-Bit A/D (ch)	Comparators	8/16-Bit Timers	CCP/ECCP	MSSP	UART w/IrDA®	
PIC24F16KL402	28	16K	1024	512	12	2	2/2	2/1	2	2	Y
PIC24F08KL402	28	8K	1024	512	12	2	2/2	2/1	2	2	Y
PIC24F16KL401	20	16K	1024	512	12	2	2/2	2/1	2	2	Y
PIC24F08KL401	20	8K	1024	512	12	2	2/2	2/1	2	2	Y
PIC24F08KL302	28	8K	1024	256	—	2	2/2	2/1	2	2	Y
PIC24F08KL301	20	8K	1024	256	—	2	2/2	2/1	2	2	Y
PIC24F08KL201	20	8K	512	—	12	1	1/2	2/0	1	1	Y
PIC24F08KL200	14	8K	512	—	7	1	1/2	2/0	1	1	Y
PIC24F04KL101	20	4K	512	—	—	1	1/2	2/0	1	1	Y
PIC24F04KL100	14	4K	512	—	—	1	1/2	2/0	1	1	Y

# PIC24F16KL402 FAMILY

---

## Analog Features:

- 10-Bit, up to 12-Channel Analog-to-Digital (A/D) Converter:
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Dual Rail-to-Rail Analog Comparators with Programmable Input/Output Configuration
- On-Chip Voltage Reference

## Special Microcontroller Features:

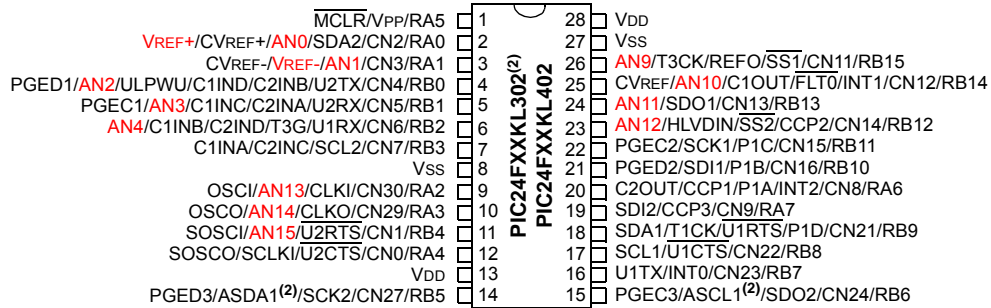
- Operating Voltage Range of 1.8V to 3.6V
- 10,000 Erase/Write Cycle Endurance Flash Program Memory, Typical
- 100,000 Erase/Write Cycle Endurance Data EEPROM, Typical
- Flash and Data EEPROM Data Retention: 40 Years Minimum
- Self-Programmable under Software Control
- Programmable Reference Clock Output

- Fail-Safe Clock Monitor (FSCM) Operation:
  - Detects clock failure and switches to on-chip, Low-Power RC (LPRC) oscillator
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT):
  - Uses its own Low-Power RC oscillator
  - Windowed operating modes
  - Programmable period of 2 ms to 131s
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Emulation (ICE) via 2 Pins
- Programmable High/Low-Voltage Detect (HLVD)
- Programmable Brown-out Reset (BOR):
  - Configurable for software controlled operation and shutdown in Sleep mode
  - Selectable trip points (1.8V, 2.7V and 3.0V)
  - Low-power 2.0V POR re-arm

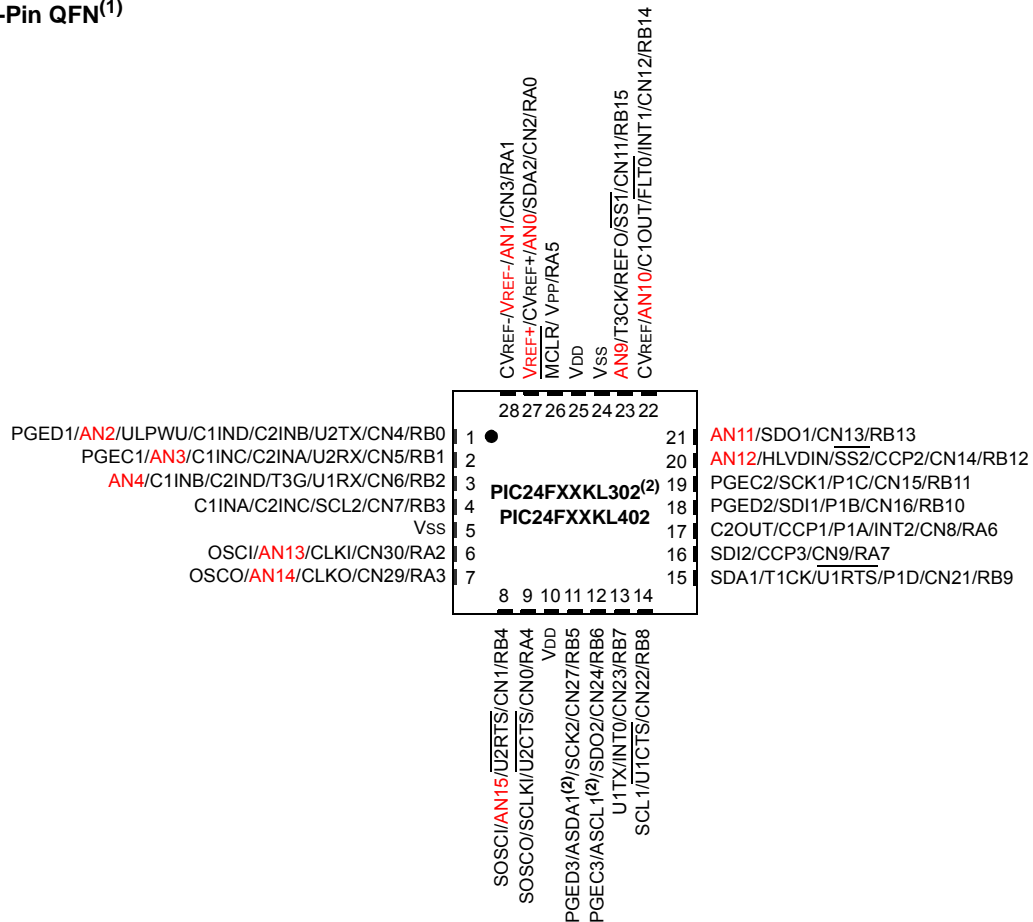
# PIC24F16KL402 FAMILY

## Pin Diagrams: PIC24FXXKL302/402

### 28-Pin SPDIP/SSOP/SOIC<sup>(1)</sup>



### 28-Pin QFN<sup>(1)</sup>



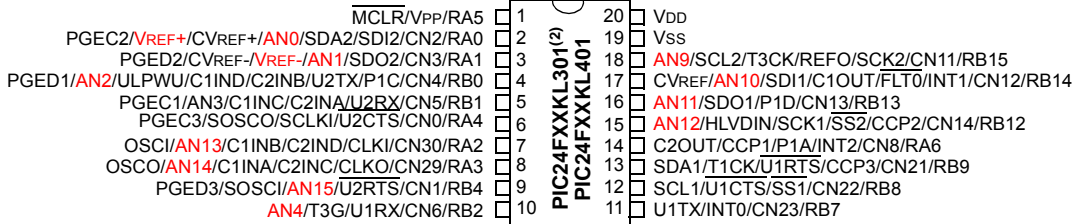
Contact your Microchip sales team for Chip Scale Package (CSP) availability.

- Note 1:** Analog features (indicated in red) are not available on PIC24FXXKL302 devices.  
**Note 2:** Alternate location for I<sup>2</sup>C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

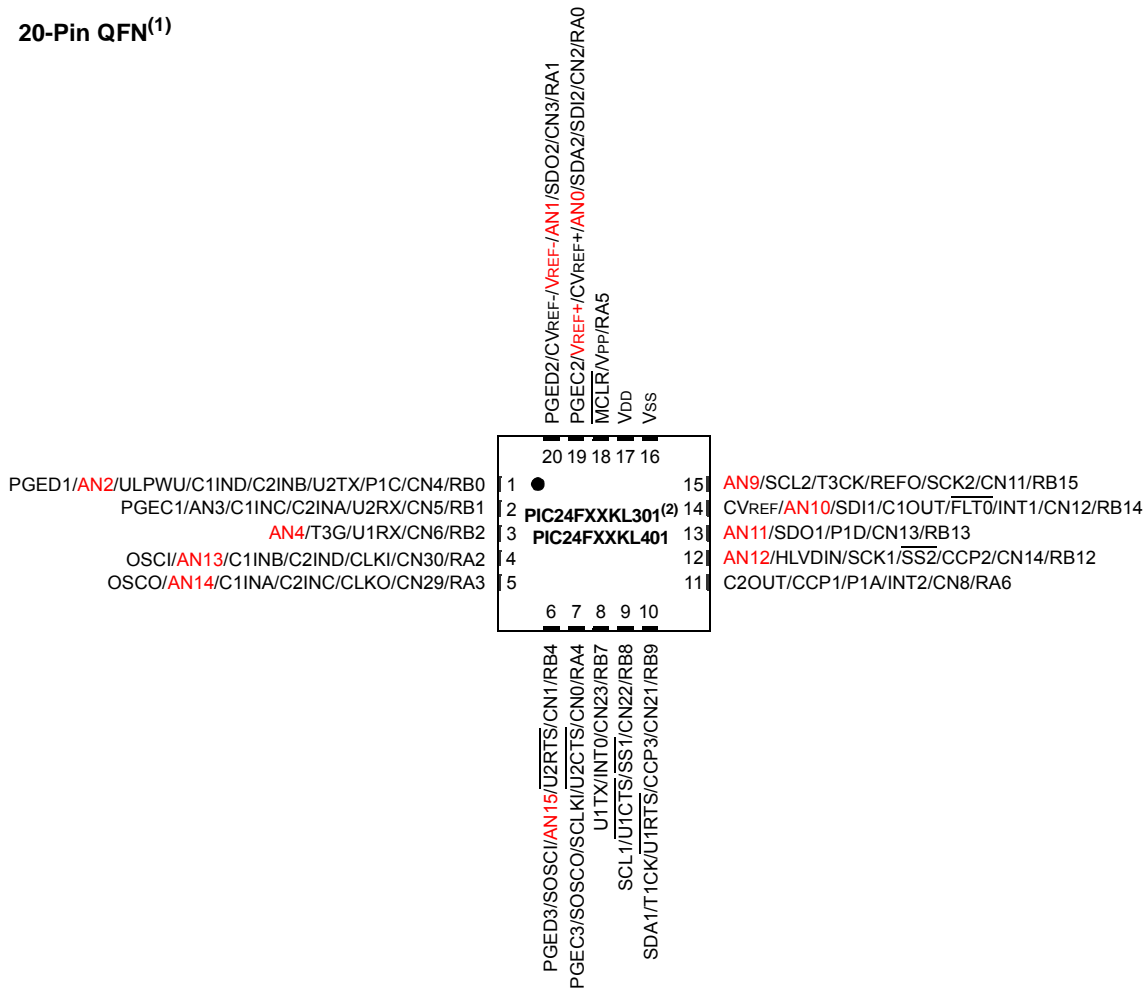
# PIC24F16KL402 FAMILY

## Pin Diagrams: PIC24FXXKL301/401

### 20-Pin PDIP/SSOP/SOIC<sup>(1)</sup>



### 20-Pin QFN<sup>(1)</sup>

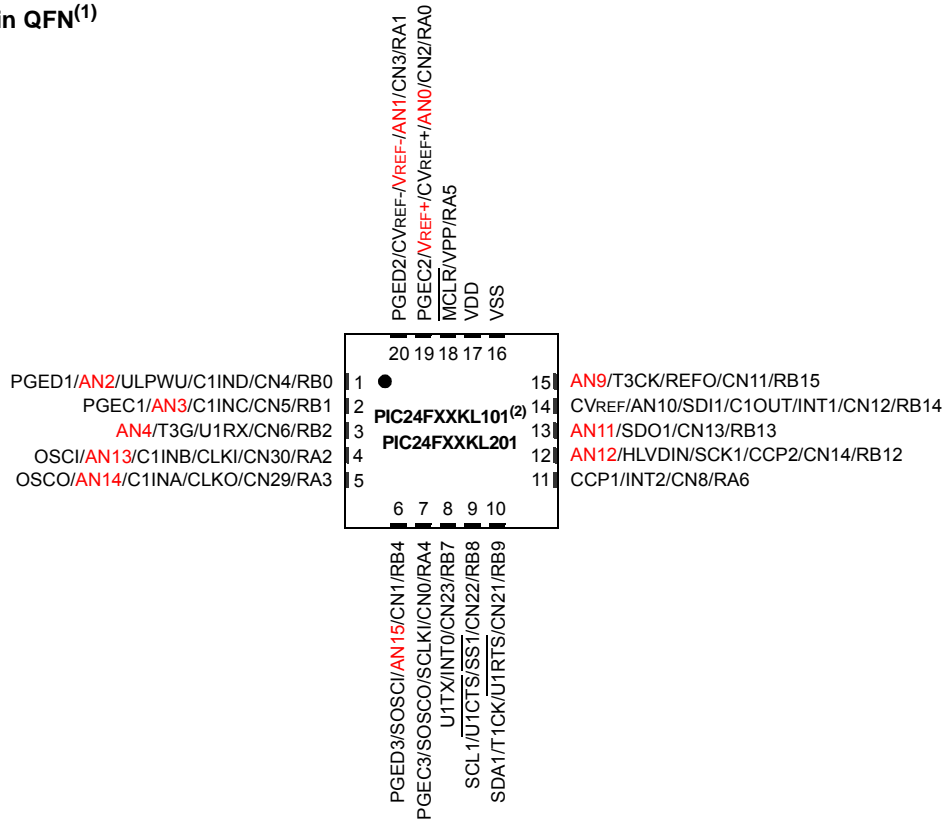


- Note 1:** Analog features (indicated in red) are not available on PIC24FXXKL301 devices.  
**Note 2:** Alternate location for I<sup>2</sup>C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

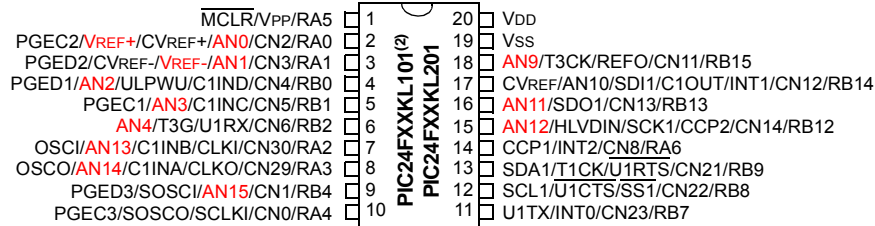
# PIC24F16KL402 FAMILY

## Pin Diagrams: PIC24FXXKL10X/20X

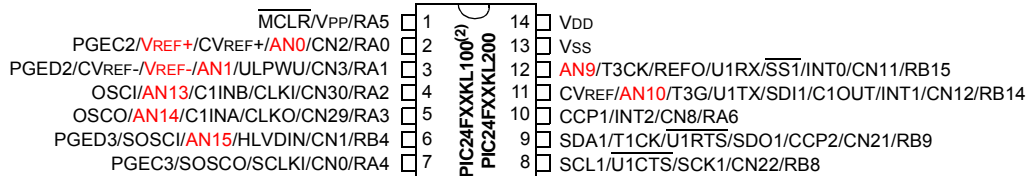
### 20-Pin QFN<sup>(1)</sup>



### 20-Pin PDIP/SSOP/SOIC<sup>(1)</sup>



### 14-Pin PDIP/TSSOP<sup>(1)</sup>



- Note 1:** Analog features (indicated in red) are not available on PIC24FXXKL100/101 devices.  
**Note 2:** Alternate location for I<sup>2</sup>C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

# PIC24F16KL402 FAMILY

---

## Table of Contents

1.0	Device Overview .....	9
2.0	Guidelines for Getting Started with 16-Bit Microcontrollers .....	21
3.0	CPU .....	25
4.0	Memory Organization .....	31
5.0	Flash Program Memory .....	47
6.0	Data EEPROM Memory .....	53
7.0	Resets .....	59
8.0	Interrupt Controller .....	65
9.0	Oscillator Configuration .....	95
10.0	Power-Saving Features .....	105
11.0	I/O Ports .....	111
12.0	Timer1 .....	115
13.0	Timer2 Module .....	117
14.0	Timer3 Module .....	119
15.0	Timer4 Module .....	123
16.0	Capture/Compare/PWM (CCP) and Enhanced CCP Modules .....	125
17.0	Master Synchronous Serial Port (MSSP) .....	135
18.0	Universal Asynchronous Receiver Transmitter (UART) .....	149
19.0	10-Bit High-Speed A/D Converter .....	157
20.0	Comparator Module .....	167
21.0	Comparator Voltage Reference .....	171
22.0	High/Low-Voltage Detect (HLVD) .....	173
23.0	Special Features .....	175
24.0	Development Support .....	187
25.0	Instruction Set Summary .....	191
26.0	Electrical Characteristics .....	199
27.0	Packaging Information .....	225
	Appendix A: Revision History .....	251
	Appendix B: Migrating from PIC18/PIC24 to PIC24F16KL402 .....	251
	Index .....	253
	The Microchip Web Site .....	257
	Customer Change Notification Service .....	257
	Customer Support .....	257
	Product Identification System .....	259

# PIC24F16KL402 FAMILY

---

---

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com). We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24F04KL100
- PIC24F08KL200
- PIC24F08KL301
- PIC24F08KL401
- PIC24F08KL402
- PIC24F04KL101
- PIC24F08KL201
- PIC24F08KL302
- PIC24F16KL401
- PIC24F16KL402

The PIC24F16KL402 family adds an entire range of economical, low pin count and low-power devices to Microchip's portfolio of 16-bit microcontrollers. Aimed at applications that require low-power consumption but more computational ability than an 8-bit platform can provide, these devices offer a range of tailored peripheral sets that allow the designer to optimize both price point and features with no sacrifice of functionality.

### 1.1 Core Features

#### 1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® digital signal controllers. The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 64 Kbytes (data)
- A 16-element Working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32-bit by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as C
- Operational performance up to 16 MIPS

#### 1.1.2 POWER-SAVING TECHNOLOGY

All of the devices in the PIC24F16KL402 family incorporate a range of features that can significantly reduce power consumption during operation. Key features include:

- **On-the-Fly Clock Switching:** The device clock can be changed under software control to the Timer1 source, or the internal, Low-Power RC (LPRC) oscillator during operation, allowing the user to incorporate power-saving ideas into their software designs.

- **Doze Mode Operation:** When timing-sensitive applications, such as serial communications, require the uninterrupted operation of peripherals, the CPU clock speed can be selectively reduced, allowing incremental power savings without missing a beat.
- **Instruction-Based Power-Saving Modes:** The microcontroller can suspend all operations, or selectively shut down its core while leaving its peripherals active, with a single instruction in software.

#### 1.1.3 OSCILLATOR OPTIONS AND FEATURES

The PIC24F16KL402 family offers five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-2 clock output.
- Two Fast Internal Oscillators (FRCs): One with a nominal 8 MHz output and the other with a nominal 500 kHz output. These outputs can also be divided under software control to provide clock speed as low as 31 kHz or 2 kHz.
- A Phase Locked Loop (PLL) frequency multiplier, available to the External Oscillator modes and the 8 MHz FRC Oscillator, which allows clock speeds of up to 32 MHz.
- A separate Internal RC Oscillator (LPRC) with a fixed 31 kHz output, which provides a low-power option for timing-insensitive applications.

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor (FSCM). This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

#### 1.1.4 EASY MIGRATION

The consistent pinout scheme used throughout the entire family also helps in migrating to the next larger device. This is true when moving between devices with the same pin count, or even jumping from 20-pin or 28-pin devices to 44-pin/48-pin devices.

The PIC24F family is pin compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. This extends the ability of applications to grow, from the relatively simple, to the powerful and complex.

# PIC24F16KL402 FAMILY

## 1.2 Other Special Features

- **Communications:** The PIC24F16KL402 family incorporates multiple serial communication peripherals to handle a range of application requirements. The MSSP module implements both SPI and I<sup>2</sup>C™ protocols, and supports both Master and Slave modes of operation for each. Devices also include one of two UARTs with built-in IrDA® encoders/decoders.
- **Analog Features:** Select members of the PIC24F16KL402 family include a 10-bit A/D Converter module. The A/D module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, as well as faster sampling speeds. The comparator modules are configurable for a wide range of operations and can be used as either a single or double comparator module.

## 1.3 Details on Individual Family Members

Devices in the PIC24F16KL402 family are available in 14-pin, 20-pin and 28-pin packages. The general block diagram for all devices is shown in [Figure 1-1](#).

The PIC24F16KL402 family may be thought of as four different device groups, each offering a slightly different set of features. These differ from each other in multiple ways:

- The size of the Flash program memory
- The presence and size of data EEPROM
- The presence of an A/D Converter and the number of external analog channels available
- The number of analog comparators
- The number of general purpose timers
- The number and type of CCP modules (i.e., CCP vs. ECCP)
- The number of serial communications modules (both MSSPs and UARTs)

The general differences between the different sub-families are shown in [Table 1-1](#). The feature sets for specific devices are summarized in [Table 1-2](#) and [Table 1-3](#).

A list of the individual pin features available on the PIC24F16KL402 family devices, sorted by function, is provided in [Table 1-4](#) (for PIC24FXXKL40X/30X devices) and [Table 1-5](#) (for PIC24FXXKL20X/10X devices). Note that these tables show the pin location of individual peripheral features and not how they are multiplexed on the same pin. This information is provided in the pinout diagrams in the beginning of this data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

**TABLE 1-1: FEATURE COMPARISON FOR PIC24F16KL402 FAMILY GROUPS**

Device Group	Program Memory (bytes)	Data EEPROM (bytes)	Timers (8/16-bit)	CCP and ECCP	Serial (MSSP/UART)	A/D (channels)	Comparators
PIC24FXXKL10X	4K	—	1/2	2/0	1/1	—	1
PIC24FXXKL20X	8K	—	1/2	2/0	1/1	7 or 12	1
PIC24FXXKL30X	8K	256	2/2	2/1	2/2	—	2
PIC24FXXKL40X	8K or 16K	512	2/2	2/1	2/2	12	2

# PIC24F16KL402 FAMILY

**TABLE 1-2: DEVICE FEATURES FOR PIC24F16KL40X/30X DEVICES**

Features	PIC24F16KL402	PIC24F08KL402	PIC24F08KL302	PIC24F16KL401	PIC24F08KL401	PIC24F08KL301
Operating Frequency	DC – 32 MHz					
Program Memory (bytes)	16K	8K	8K	16K	8K	8K
Program Memory (instructions)	5632	2816	2816	5632	2816	2816
Data Memory (bytes)	1024	1024	1024	1024	1024	1024
Data EEPROM Memory (bytes)	512	512	256	512	512	256
Interrupt Sources (soft vectors/NMI traps)	31 (27/4)	31 (27/4)	30 (26/4)	31 (27/4)	31 (27/4)	30 (26/4)
I/O Ports	PORTA<7:0> PORTB<15:0>			PORTA<6:0> PORTB<15:12,9:7,4,2:0>		
Total I/O Pins	24			18		
Timers (8/16-bit)	2/2	2/2	2/2	2/2	2/2	2/2
Capture/Compare/PWM modules:						
Total	3	3	3	3	3	3
Enhanced CCP	1	1	1	1	1	1
Input Change Notification Interrupt	23	23	23	17	17	17
Serial Communications:						
UART	2	2	2	2	2	2
MSSP	2	2	2	2	2	2
10-Bit Analog-to-Digital Module (input channels)	12	12	—	12	12	—
Analog Comparators	2	2	2	2	2	2
Resets (and delays)	POR, BOR, RESET Instruction, MCLR, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)					
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations					
Packages	28-Pin SPDIP/SSOP/SOIC/QFN			20-Pin PDIP/SSOP/SOIC/QFN		

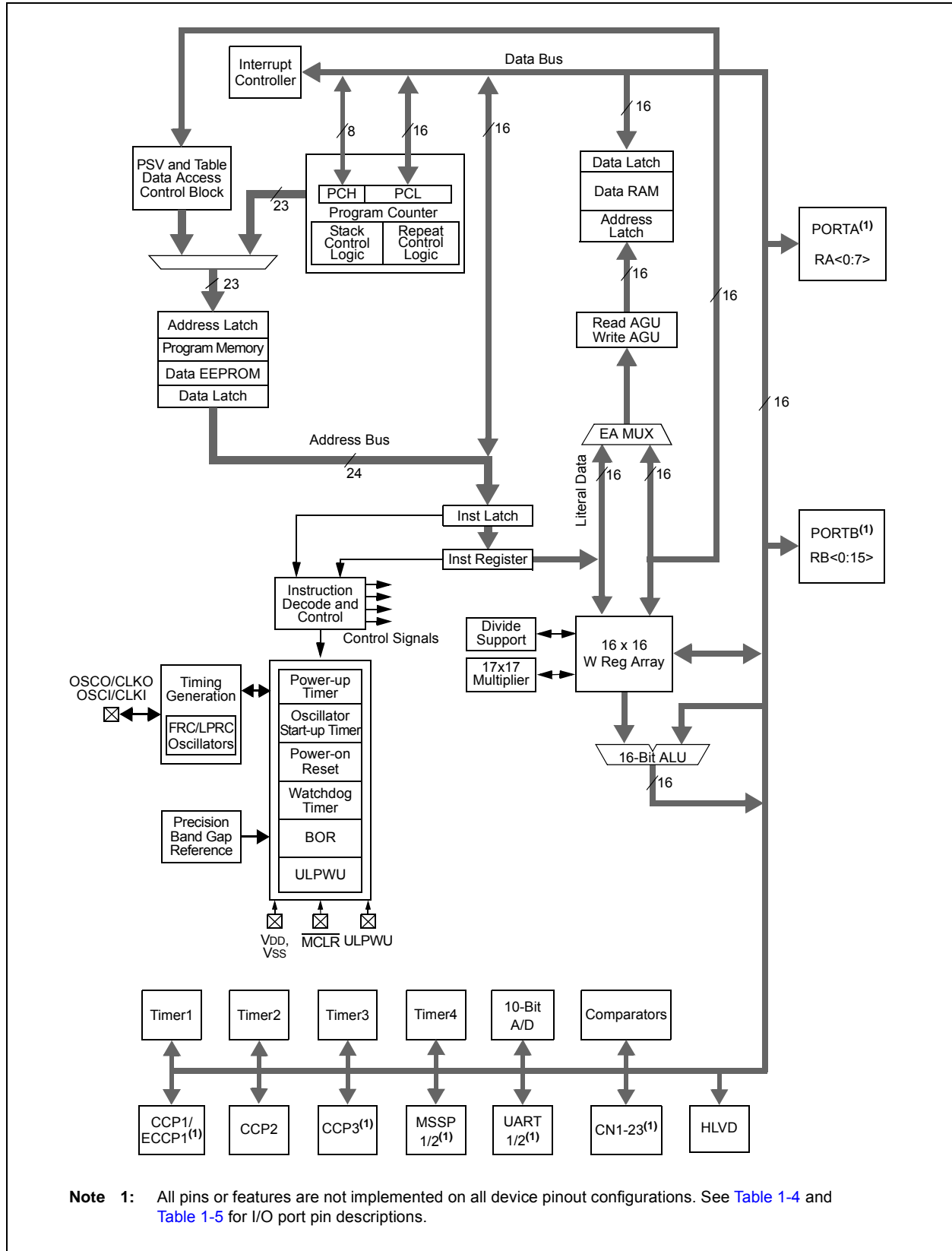
# PIC24F16KL402 FAMILY

**TABLE 1-3: DEVICE FEATURES FOR THE PIC24F16KL20X/10X DEVICES**

Features	PIC24F08KL201	PIC24F04KL101	PIC24F08KL200	PIC24F04KL100
Operating Frequency	DC – 32 MHz			
Program Memory (bytes)	8K	4K	8K	4K
Program Memory (instructions)	2816	1408	2816	1408
Data Memory (bytes)	512	512	512	512
Data EEPROM Memory (bytes)	—	—	—	—
Interrupt Sources (soft vectors/NMI traps)	27 (23/4)	26 (22/4)	27 (23/4)	26 (22/4)
I/O Ports	PORTA<6:0> PORTB<15:12,9:7,4,2:0>		PORTA<5:0> PORTB<15:14,9:8,4,0>	
Total I/O Pins	17		12	
Timers (8/16-bit)	1/2	1/2	1/2	1/2
Capture/Compare/PWM modules:				
Total	2	2	2	2
Enhanced CCP	0	0	0	0
Input Change Notification Interrupt	17	17	11	11
Serial Communications:				
UART	1	1	1	1
MSSP	1	1	1	1
10-Bit Analog-to-Digital Module (input channels)	12	—	7	—
Analog Comparators	1	1	1	1
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{\text{MCLR}}$ , WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	20-Pin PDIP/SSOP/SOIC/QFN		14-Pin PDIP/TSSOP	

# PIC24F16KL402 FAMILY

FIGURE 1-1: PIC24F16KL402 FAMILY GENERAL BLOCK DIAGRAM



# PIC24F16KL402 FAMILY

**TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS**

Function	Pin Number				I/O	Buffer	Description	
	20-Pin PDIP/SSOP/SOIC	20-Pin QFN	28-Pin SPDIP/SSOP/SOIC	28-Pin QFN				
AN0	2	19	2	27	I	ANA	A/D Analog Inputs. Not available on PIC24F16KL30X family devices.	
AN1	3	20	3	28	I	ANA		
AN2	4	1	4	1	I	ANA		
AN3	5	2	5	2	I	ANA		
AN4	6	3	6	3	I	ANA		
AN5	—	—	7	4	I	ANA		
AN9	18	15	26	23	I	ANA		
AN10	17	14	25	22	I	ANA		
AN11	16	13	24	21	I	ANA		
AN12	15	12	23	20	I	ANA		
AN13	7	4	9	6	I	ANA		
AN14	8	5	10	7	I	ANA		
AN15	9	6	11	8	I	ANA		
ASCL1	—	—	15	12	I/O	I <sup>2</sup> C™		Alternate MSSP1 I <sup>2</sup> C Clock Input/Output
ASDA1	—	—	14	11	I/O	I <sup>2</sup> C		Alternate MSSP1 I <sup>2</sup> C Data Input/Output
AVDD	20	17	28	25	I	ANA	Positive Supply for Analog modules	
AVSS	19	16	27	24	I	ANA	Ground Reference for Analog modules	
CCP1	14	11	20	17	I/O	ST	CCP1/ECCP1 Capture Input/Compare and PWM Output	
CCP2	15	12	23	20	I/O	ST	CCP2 Capture Input/Compare and PWM Output	
CCP3	13	10	19	16	I/O	ST	CCP3 Capture Input/Compare and PWM Output	
C1INA	8	5	7	4	I	ANA	Comparator 1 Input A (+)	
C1INB	7	4	6	3	I	ANA	Comparator 1 Input B (-)	
C1INC	5	2	5	2	I	ANA	Comparator 1 Input C (+)	
C1IND	4	1	4	1	I	ANA	Comparator 1 Input D (-)	
C1OUT	17	14	25	22	O	—	Comparator 1 Output	
C2INA	5	2	5	2	I	ANA	Comparator 2 Input A (+)	
C2INB	4	1	4	1	I	ANA	Comparator 2 Input B (-)	
C2INC	8	5	7	4	I	ANA	Comparator 2 Input C (+)	
C2IND	7	4	6	3	I	ANA	Comparator 2 Input D (-)	
C2OUT	14	11	20	17	O	—	Comparator 2 Output	
CLK I	7	4	9	6	I	ANA	Main Clock Input	
CLKO	8	5	10	7	O	—	System Clock Output	

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

Function	Pin Number				I/O	Buffer	Description	
	20-Pin PDIP/ SSOP/ SOIC	20-Pin QFN	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN				
CN0	10	7	12	9	I	ST	Interrupt-on-Change Inputs	
CN1	9	6	11	8	I	ST		
CN2	2	19	2	27	I	ST		
CN3	3	20	3	28	I	ST		
CN4	4	1	4	1	I	ST		
CN5	5	2	5	2	I	ST		
CN6	6	3	6	3	I	ST		
CN7	—	—	7	4	I	ST		
CN8	14	11	20	17	I	ST		
CN9	—	—	19	16	I	ST		
CN11	18	15	26	23	I	ST		
CN12	17	14	25	22	I	ST		
CN13	16	13	24	21	I	ST		
CN14	15	12	23	20	I	ST		
CN15	—	—	22	19	I	ST		
CN16	—	—	21	18	I	ST		
CN21	13	10	18	15	I	ST		
CN22	12	9	17	14	I	ST		
CN23	11	8	16	13	I	ST		
CN24	—	—	15	12	I	ST		
CN27	—	—	14	11	I	ST		
CN29	8	5	10	7	I	ST		
CN30	7	4	9	6	I	ST		
CVREF	17	14	25	22	I	ANA		Comparator Voltage Reference Output
CVREF+	2	19	2	27	I	ANA		Comparator Reference Positive Input Voltage
CVREF-	3	20	3	28	I	ANA		Comparator Reference Negative Input Voltage
FLT0	17	14	25	22	I	ST		ECCP1 Enhanced PWM Fault Input
HLVDIN	15	12	23	20	I	ST		High/Low-Voltage Detect Input
INT0	11	8	16	13	I	ST		Interrupt 0 Input
INT1	17	14	25	22	I	ST		Interrupt 1 Input
INT2	14	11	20	17	I	ST	Interrupt 2 Input	
MCLR	1	18	1	26	I	ST	Master Clear (device Reset) Input. This line is brought low to cause a Reset.	
OSCI	7	4	9	6	I	ANA	Main Oscillator Input	
OSCO	8	5	10	7	O	ANA	Main Oscillator Output	
P1A	14	11	20	17	O	—	ECCP1 Output A (Enhanced PWM Mode)	
P1B	5	2	21	18	O	—	ECCP1 Output B (Enhanced PWM Mode)	
P1C	4	1	22	19	O	—	ECCP1 Output C (Enhanced PWM Mode)	
P1D	16	13	18	15	O	—	ECCP1 Output D (Enhanced PWM Mode)	

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

Function	Pin Number				I/O	Buffer	Description
	20-Pin PDIP/SSOP/SOIC	20-Pin QFN	28-Pin SPDIP/SSOP/SOIC	28-Pin QFN			
PGEC1	5	2	5	2	I/O	ST	ICSP™ Clock 1
PCED1	4	1	4	1	I/O	ST	ICSP Data 1
PGEC2	2	19	22	19	I/O	ST	ICSP Clock 2
PGED2	3	20	21	18	I/O	ST	ICSP Data 2
PGEC3	10	7	15	12	I/O	ST	ICSP Clock 3
PGED3	9	6	14	11	I/O	ST	ICSP Data 3
RA0	2	19	2	27	I/O	ST	PORTA Pins
RA1	3	20	3	28	I/O	ST	
RA2	7	4	9	6	I/O	ST	
RA3	8	5	10	7	I/O	ST	
RA4	10	7	12	9	I/O	ST	
RA5	1	18	1	26	I	ST	
RA6	14	11	20	17	I/O	ST	
RA7	—	—	19	16	I/O	ST	
RB0	4	1	4	1	I/O	ST	PORTB Pins
RB1	5	2	5	2	I/O	ST	
RB2	6	3	6	3	I/O	ST	
RB3	—	—	7	4	I/O	ST	
RB4	9	6	11	8	I/O	ST	
RB5	—	—	14	11	I/O	ST	
RB6	—	—	15	12	I/O	ST	
RB7	11	8	16	13	I/O	ST	
RB8	12	9	17	14	I/O	ST	
RB9	13	10	18	15	I/O	ST	
RB10	—	—	21	18	I/O	ST	
RB11	—	—	22	19	I/O	ST	
RB12	15	12	23	20	I/O	ST	
RB13	16	13	24	21	I/O	ST	
RB14	17	14	25	22	I/O	ST	
RB15	18	15	26	23	I/O	ST	
REFO	18	15	26	23	O	—	Reference Clock Output
SCK1	15	12	22	19	I/O	ST	MSSP1 SPI Serial Input/Output Clock
SCK2	18	15	14	11	I/O	ST	MSSP2 SPI Serial Input/Output Clock
SCL1	12	9	17	14	I/O	I <sup>2</sup> C	MSSP1 I <sup>2</sup> C Clock Input/Output
SCL2	18	15	7	4	I/O	I <sup>2</sup> C	MSSP2 I <sup>2</sup> C Clock Input/Output
SCLKI	10	7	12	9	I	ST	Digital Secondary Clock Input
SDA1	13	10	18	15	I/O	I <sup>2</sup> C	MSSP1 I <sup>2</sup> C Data Input/Output
SDA2	2	19	2	27	I/O	I <sup>2</sup> C	MSSP2 I <sup>2</sup> C Data Input/Output
SDI1	17	14	21	18	I	ST	MSSP1 SPI Serial Data Input
SDI2	2	19	19	16	I	ST	MSSP2 SPI Serial Data Input
SDO1	16	13	24	21	O	—	MSSP1 SPI Serial Data Output
SDO2	3	20	15	12	O	—	MSSP2 SPI Serial Data Output

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

Function	Pin Number				I/O	Buffer	Description
	20-Pin PDIP/ SSOP/ SOIC	20-Pin QFN	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN			
SOSCI	9	6	11	8	I	ANA	Secondary Oscillator Input
SOSCO	10	7	12	9	O	ANA	Secondary Oscillator Output
SS1	12	9	26	23	O	—	SPI1 Slave Select
SS2	15	12	23	20	O	—	SPI2 Slave Select
T1CK	13	10	18	15	I	ST	Timer1 Clock
T3CK	18	15	26	23	I	ST	Timer3 Clock
T3G	6	3	6	3	I	ST	Timer3 External Gate Input
U1CTS	12	9	17	14	I	ST	UART1 Clear-to-Send Input
U1RTS	13	10	18	15	O	—	UART1 Request-to-Send Output
U1RX	6	3	6	3	I	ST	UART1 Receive
U1TX	11	8	16	13	O	—	UART1 Transmit
U2CTS	10	7	12	9	I	ST	UART2 Clear-to-Send Input
U2RTS	9	6	11	8	O	—	UART2 Request-to-Send Output
U2RX	5	2	5	2	I	ST	UART2 Receive
U2TX	4	1	4	1	O	—	UART2 Transmit
ULPWU	4	1	4	1	I	ANA	Ultra Low-Power Wake-up Input
VDD	20	17	13, 28	10, 25	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins
VREF+	2	19	2	27	I	ANA	A/D Reference Voltage Input (+)
VREF-	3	20	3	28	I	ANA	A/D Reference Voltage Input (-)
VSS	19	16	8, 27	5, 24	P	—	Ground Reference for Logic and I/O Pins

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-5: PIC24F16KL20X/10X FAMILY PINOUT DESCRIPTIONS**

Function	Pin Number			I/O	Buffer	Description	
	20-Pin PDIP/ SSOP/ SOIC	20-Pin QFN	14-Pin PDIP/ TSSOP				
AN0	2	19	2	I	ANA	A/D Analog Inputs. Not available on PIC24F16KL10X family devices.	
AN1	3	20	3	I	ANA		
AN2	4	1	—	I	ANA		
AN3	5	2	—	I	ANA		
AN4	6	3	—	I	ANA		
AN9	18	15	12	I	ANA		
AN10	17	14	11	I	ANA		
AN11	16	13	—	I	ANA		
AN12	15	12	—	I	ANA		
AN13	7	4	4	I	ANA		
AN14	8	5	5	I	ANA		
AN15	9	6	6	I	ANA		
AVDD	20	17	14	I	ANA		Positive Supply for Analog modules
AVSS	19	16	13	I	ANA		Ground Reference for Analog modules
CCP1	14	11	10	I/O	ST		CCP1 Capture Input/Compare and PWM Output
CCP2	15	12	9	I/O	ST	CCP2 Capture Input/Compare and PWM Output	
C1INA	8	5	5	I	ANA	Comparator 1 Input A (+)	
C1INB	7	4	4	I	ANA	Comparator 1 Input B (-)	
C1INC	5	2	—	I	ANA	Comparator 1 Input C (+)	
C1IND	4	1	—	I	ANA	Comparator 1 Input D (-)	
C1OUT	17	14	11	O	—	Comparator 1 Output	
CLK I	7	4	9	I	ANA	Main Clock Input	
CLKO	8	5	10	O	—	System Clock Output	
CN0	10	7	7	I	ST	Interrupt-on-Change Inputs	
CN1	9	6	6	I	ST		
CN2	2	19	2	I	ST		
CN3	3	20	3	I	ST		
CN4	4	1	—	I	ST		
CN5	5	2	—	I	ST		
CN6	6	3	—	I	ST		
CN8	14	11	10	I	ST		
CN9	—	—	—	I	ST		
CN11	18	15	12	I	ST		
CN12	17	14	11	I	ST		
CN13	16	13	—	I	ST		
CN14	15	12	—	I	ST		
CN21	13	10	9	I	ST		
CN22	12	9	8	I	ST		
CN23	11	8	—	I	ST		
CN29	8	5	5	I	ST		
CN30	7	4	4	I	ST		

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-5: PIC24F16KL20X/10X FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

Function	Pin Number			I/O	Buffer	Description
	20-Pin PDIP/SSOP/SOIC	20-Pin QFN	14-Pin PDIP/TSSOP			
CVREF	17	14	11	I	ANA	Comparator Voltage Reference Output
CVREF+	2	19	2	I	ANA	Comparator Reference Positive Input Voltage
CVREF-	3	20	3	I	ANA	Comparator Reference Negative Input Voltage
HLVDIN	15	12	6	I	ST	High/Low-Voltage Detect Input
INT0	11	8	12	I	ST	Interrupt 0 Input
INT1	17	14	11	I	ST	Interrupt 1 Input
INT2	14	11	10	I	ST	Interrupt 2 Input
MCLR	1	18	1	I	ST	Master Clear (device Reset) Input. This line is brought low to cause a Reset.
OSCI	7	4	4	I	ANA	Main Oscillator Input
OSCO	8	5	5	O	ANA	Main Oscillator Output
PGEC1	5	2	—	I/O	ST	ICSP™ Clock 1
PCED1	4	1	—	I/O	ST	ICSP Data 1
PGEC2	2	19	2	I/O	ST	ICSP Clock 2
PGED2	3	20	3	I/O	ST	ICSP Data 2
PGEC3	10	7	7	I/O	ST	ICSP Clock 3
PGED3	9	6	6	I/O	ST	ICSP Data 3
RA0	2	19	2	I/O	ST	PORTA Pins
RA1	3	20	3	I/O	ST	
RA2	7	4	4	I/O	ST	
RA3	8	5	5	I/O	ST	
RA4	10	7	7	I/O	ST	
RA5	1	18	1	I	ST	
RA6	14	11	10	I/O	ST	
RB0	4	1	—	I/O	ST	PORTB Pins
RB1	5	2	—	I/O	ST	
RB2	6	3	—	I/O	ST	
RB4	9	6	6	I/O	ST	
RB7	11	8	—	I/O	ST	
RB8	12	9	8	I/O	ST	
RB9	13	10	9	I/O	ST	
RB12	15	12	—	I/O	ST	
RB13	16	13	—	I/O	ST	
RB14	17	14	11	I/O	ST	
RB15	18	15	12	I/O	ST	
REFO	18	15	12	O	—	Reference Clock Output

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

**TABLE 1-5: PIC24F16KL20X/10X FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

Function	Pin Number			I/O	Buffer	Description
	20-Pin PDIP/SSOP/SOIC	20-Pin QFN	14-Pin PDIP/TSSOP			
SCK1	15	12	8	I/O	ST	MSSP1 SPI Serial Input/Output Clock
SCL1	12	9	8	I/O	I <sup>2</sup> C	MSSP1 I <sup>2</sup> C Clock Input/Output
SCLKI	10	7	12	I	ST	Digital Secondary Clock Input
SDA1	13	10	9	I/O	I <sup>2</sup> C	MSSP1 I <sup>2</sup> C Data Input/Output
SDI1	17	14	11	I	ST	MSSP1 SPI Serial Data Input
SDO1	16	13	9	O	—	MSSP1 SPI Serial Data Output
SOSCI	9	6	11	I	ANA	Secondary Oscillator Input
SOSCO	10	7	12	O	ANA	Secondary Oscillator Output
$\overline{SS1}$	12	9	12	O	—	SPI1 Slave Select
T1CK	13	10	9	I	ST	Timer1 Clock
T3CK	18	15	12	I	ST	Timer3 Clock
T3G	6	3	11	I	ST	Timer3 External Gate Input
$\overline{U1CTS}$	12	9	8	I	ST	UART1 Clear-to-Send Input
$\overline{U1RTS}$	13	10	9	O	—	UART1 Request-to-Send Output
U1RX	6	3	12	I	ST	UART1 Receive
U1TX	11	8	11	O	—	UART1 Transmit
ULPWU	3	1	3	I	ANA	Ultra Low-Power Wake-up Input
VDD	20	17	14	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins
VREF+	2	19	2	I	ANA	A/D Reference Voltage Input (+)
VREF-	3	20	3	I	ANA	A/D Reference Voltage Input (-)
VSS	19	16	13	P	—	Ground Reference for Logic and I/O Pins

**Legend:** TTL = TTL input buffer  
ANA = Analog level input/output

ST = Schmitt Trigger input buffer  
I<sup>2</sup>C = I<sup>2</sup>C™/SMBus input buffer

# PIC24F16KL402 FAMILY

## 2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

### 2.1 Basic Connection Requirements

Getting started with the PIC24F16KL402 family of 16-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see [Section 2.2 “Power Supply Pins”](#))
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used (see [Section 2.2 “Power Supply Pins”](#))
- MCLR pin (see [Section 2.3 “Master Clear \(MCLR\) Pin”](#))

These pins must also be connected if they are being used in the end application:

- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see [Section 2.4 “ICSP Pins”](#))
- OSCI and OSCO pins when an external oscillator source is used (see [Section 2.5 “External Oscillator Pins”](#))

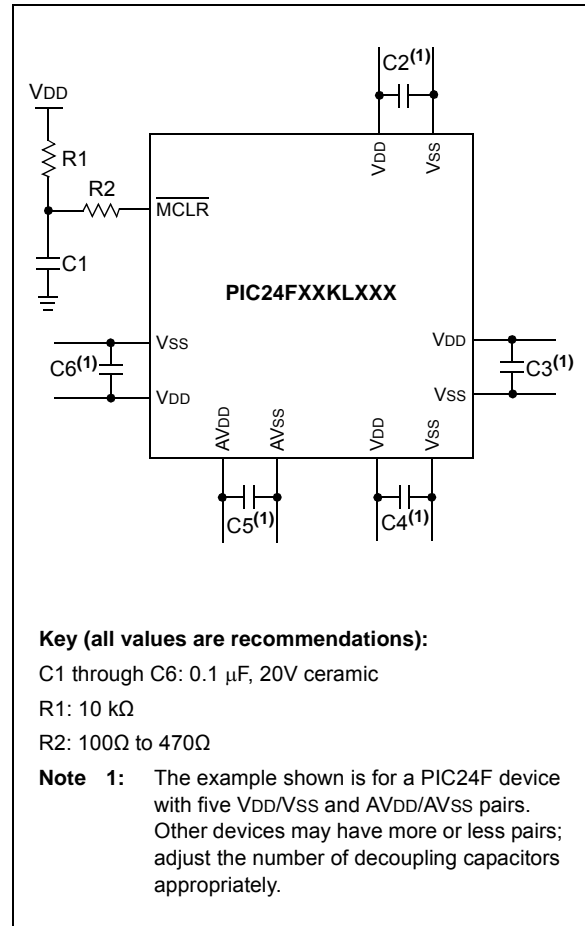
Additionally, the following pins may be required:

- VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

**Note:** The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in [Figure 2-1](#).

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



# PIC24F16KL402 FAMILY

## 2.2 Power Supply Pins

### 2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS, is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1  $\mu\text{F}$  (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01  $\mu\text{F}$  to 0.001  $\mu\text{F}$ . Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1  $\mu\text{F}$  in parallel with 0.001  $\mu\text{F}$ ).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

### 2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7  $\mu\text{F}$  to 47  $\mu\text{F}$ .

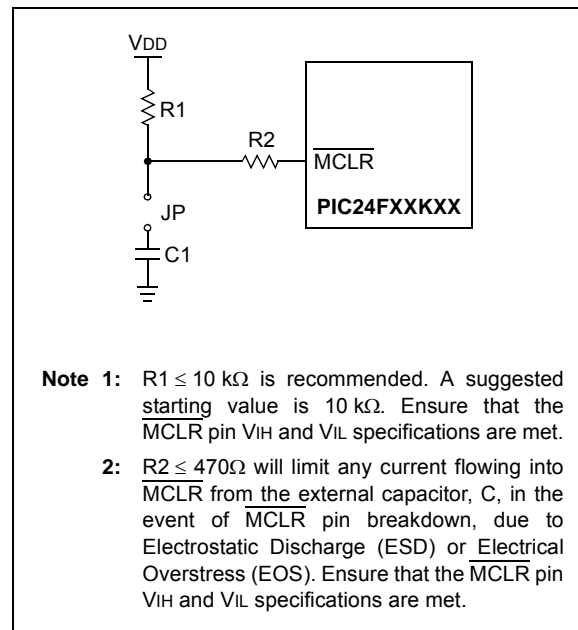
## 2.3 Master Clear ( $\overline{\text{MCLR}}$ ) Pin

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

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

Any components associated with the  $\overline{\text{MCLR}}$  pin should be placed within 0.25 inch (6 mm) of the pin.

**FIGURE 2-2: EXAMPLE OF  $\overline{\text{MCLR}}$  PIN CONNECTIONS**



# PIC24F16KL402 FAMILY

## 2.4 ICSP Pins

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

Pull-up resistors, series diodes and capacitors on the PGC and PGD pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin Input Voltage High (V<sub>IH</sub>) and Input Voltage Low (V<sub>IL</sub>) requirements.

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

For more information on available Microchip development tools connection requirements, refer to [Section 24.0 “Development Support”](#).

## 2.5 External Oscillator Pins

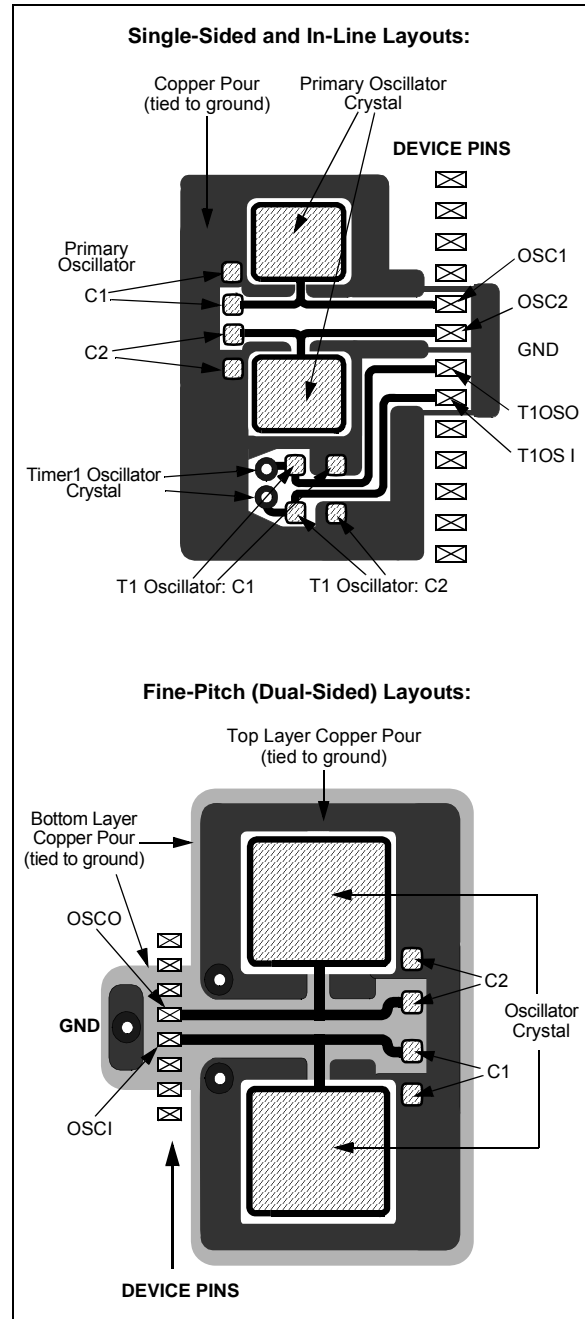
Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to [Section 9.0 “Oscillator Configuration”](#) for details).

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.

**FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**



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).

# PIC24F16KL402 FAMILY

---

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site ([www.microchip.com](http://www.microchip.com)):

- AN826, “Crystal Oscillator Basics and Crystal Selection for rPIC™ 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.

## 3.0 CPU

**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 CPU, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**CPU**” (DS39703).

The PIC24F CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the REPEAT instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit Working registers in the programmer’s model. Each of the Working registers can act as a data, address or address offset register. The 16<sup>th</sup> Working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K word boundary of either program memory or data EEPROM memory, defined by the 8-bit Program Space Visibility Page Address (PSVPAG) register. The program to data space mapping feature lets any instruction access program space as if it were data space.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported, either directly, or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal, Memory Direct and three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a Working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing ternary operations (i.e.,  $A + B = C$ ) to be executed in a single cycle.

A high-speed, 17-bit by 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit by 16-bit or 8-bit by 8-bit integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the REPEAT instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by a 16-bit integer signed and unsigned division. All divide operations require 19 cycles to complete, but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme, with up to eight sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is illustrated in [Figure 3-1](#).

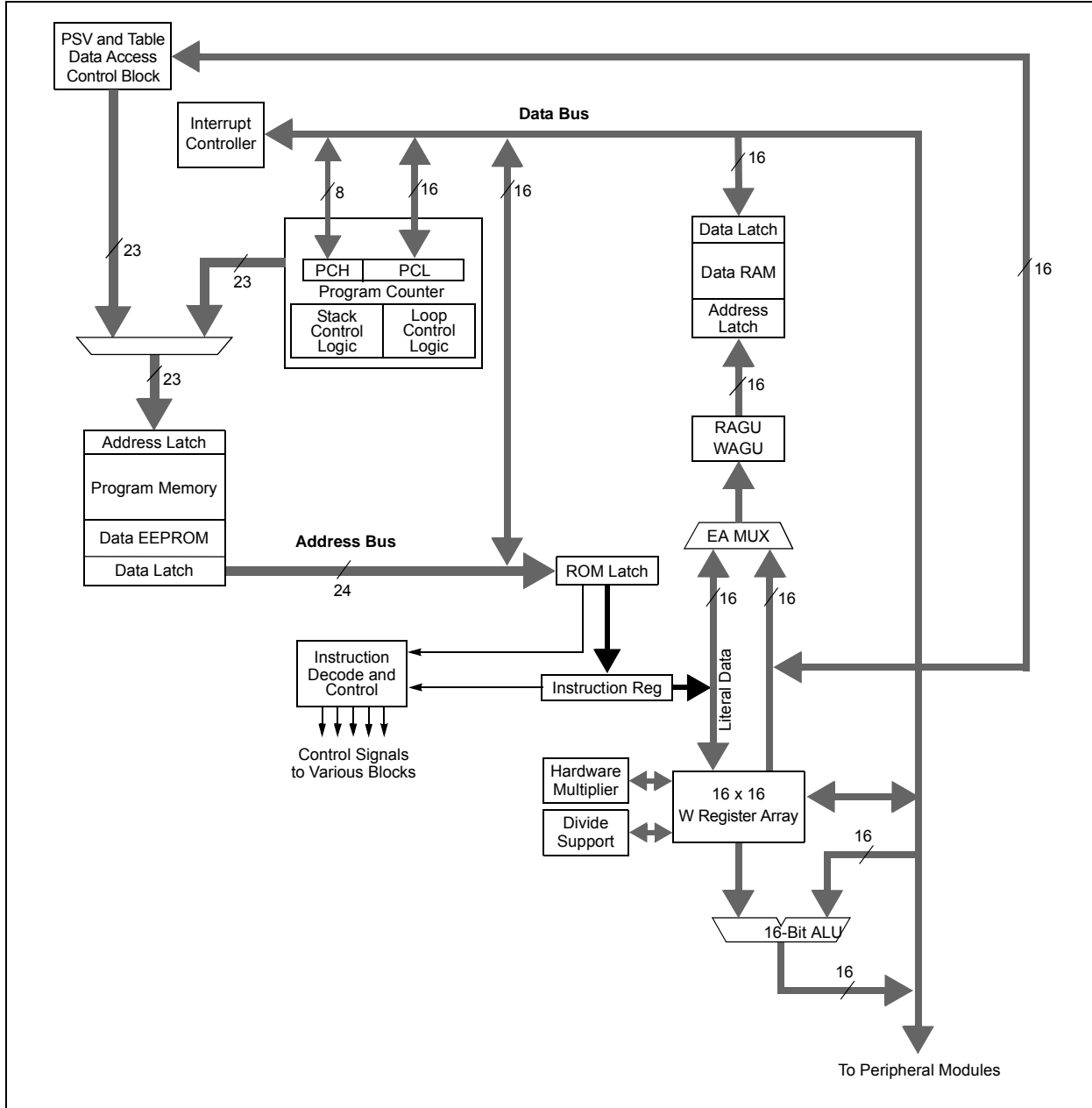
### 3.1 Programmer’s Model

[Figure 3-2](#) displays the programmer’s model for the PIC24F. All registers in the programmer’s model are memory mapped and can be manipulated directly by instructions.

[Table 3-1](#) provides a description of each register. All registers associated with the programmer’s model are memory mapped.

# PIC24F16KL402 FAMILY

**FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM**

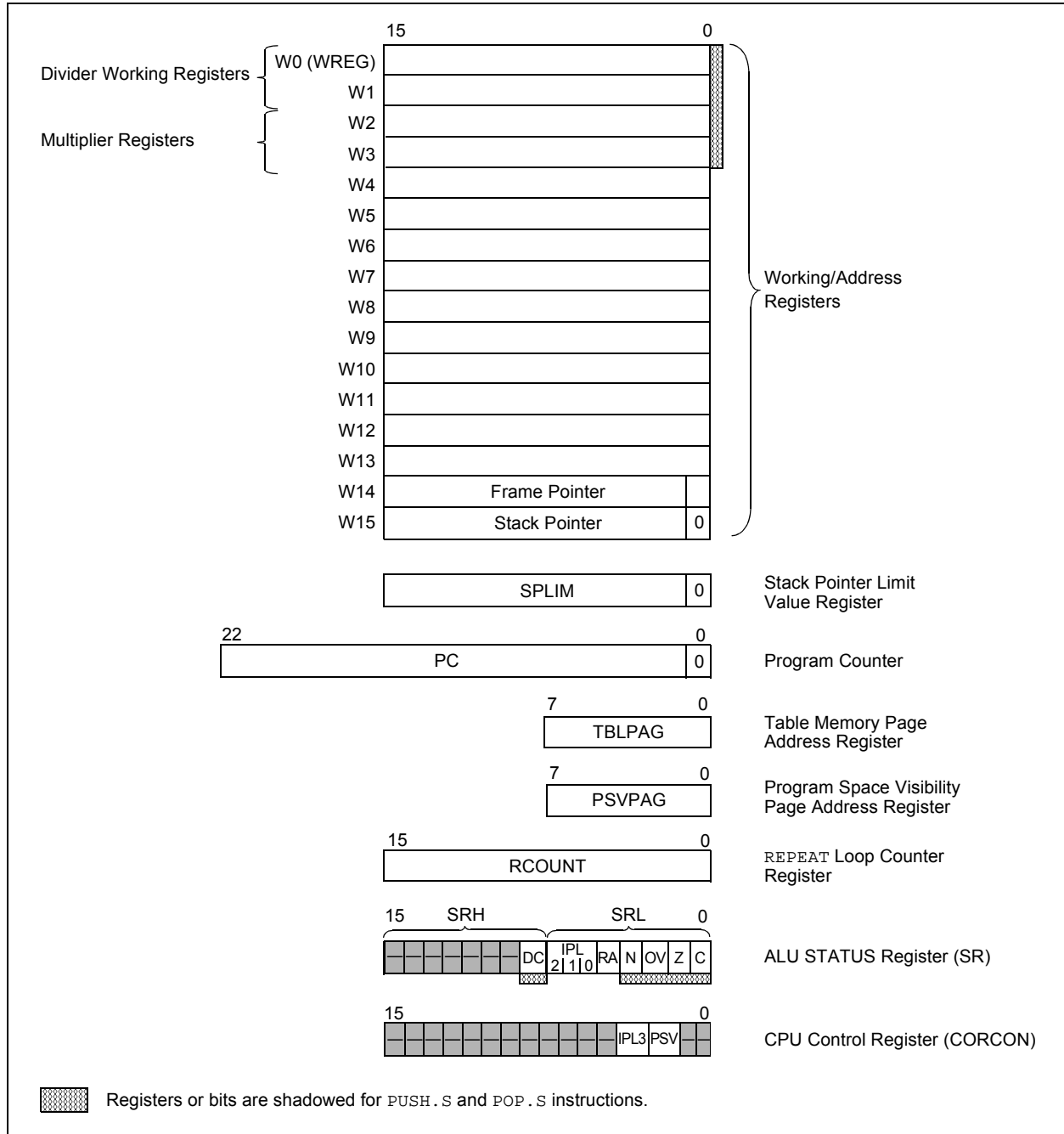


**TABLE 3-1: CPU CORE REGISTERS**

Register(s) Name	Description
W0 through W15	Working Register Array
PC	23-Bit Program Counter
SR	ALU STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
PSVPAG	Program Space Visibility Page Address Register
RCOUNT	REPEAT Loop Counter Register
CORCON	CPU Control Register

# PIC24F16KL402 FAMILY

**FIGURE 3-2: PROGRAMMER'S MODEL**



# PIC24F16KL402 FAMILY

## 3.2 CPU Control Registers

**REGISTER 3-1: SR: ALU STATUS REGISTER**

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

R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2)</sup>	IPL1 <sup>(2)</sup>	IPL0 <sup>(2)</sup>	RA	N	OV	Z	C
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        **DC:** ALU Half Carry/Borrow bit
  - 1 = A carry-out from the 4<sup>th</sup> low-order bit (for byte-sized data) or 8<sup>th</sup> low-order bit (for word-sized data) of the result occurred
  - 0 = No carry-out from the 4<sup>th</sup> or 8<sup>th</sup> low-order bit of the result has occurred
- bit 7-5      **IPL<2:0>:** CPU Interrupt Priority Level (IPL) Status bits<sup>(1,2)</sup>
  - 111 = CPU Interrupt Priority Level is 7 (15); user interrupts disabled
  - 110 = CPU Interrupt Priority Level is 6 (14)
  - 101 = CPU Interrupt Priority Level is 5 (13)
  - 100 = CPU Interrupt Priority Level is 4 (12)
  - 011 = CPU Interrupt Priority Level is 3 (11)
  - 010 = CPU Interrupt Priority Level is 2 (10)
  - 001 = CPU Interrupt Priority Level is 1 (9)
  - 000 = CPU Interrupt Priority Level is 0 (8)
- bit 4        **RA:** REPEAT Loop Active bit
  - 1 = REPEAT loop in progress
  - 0 = REPEAT loop not in progress
- bit 3        **N:** ALU Negative bit
  - 1 = Result was negative
  - 0 = Result was non-negative (zero or positive)
- bit 2        **OV:** ALU Overflow bit
  - 1 = Overflow occurred for signed (2's complement) arithmetic in this arithmetic operation
  - 0 = No overflow has occurred
- bit 1        **Z:** ALU Zero bit
  - 1 = An operation, which effects the Z bit, has set it at some time in the past
  - 0 = The most recent operation, which effects the Z bit, has cleared it (i.e., a non-zero result)
- bit 0        **C:** ALU Carry/Borrow bit
  - 1 = A carry-out from the Most Significant bit (MSb) of the result occurred
  - 0 = No carry-out from the Most Significant bit (MSb) of the result occurred

- Note 1:** The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
- Note 2:** The IPL Status bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

# PIC24F16KL402 FAMILY

## REGISTER 3-2: CORCON: CPU CONTROL REGISTER

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	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3 <sup>(1)</sup>	PSV	—	—
bit 7							bit 0

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

- bit 15-4     **Unimplemented:** Read as '0'
- bit 3        **IPL3:** CPU Interrupt Priority Level Status bit<sup>(1)</sup>  
               1 = CPU Interrupt Priority Level is greater than 7  
               0 = CPU Interrupt Priority Level is 7 or less
- bit 2        **PSV:** Program Space Visibility in Data Space Enable bit  
               1 = Program space is visible in data space  
               0 = Program space is not visible in data space
- bit 1-0     **Unimplemented:** Read as '0'

**Note 1:** User interrupts are disabled when IPL3 = 1.

### 3.3 Arithmetic Logic Unit (ALU)

The PIC24F ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

The PIC24F CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware division for a 16-bit divisor.

#### 3.3.1 MULTIPLIER

The ALU contains a high-speed, 17-bit x 17-bit multiplier. It supports unsigned, signed or mixed sign operation in several Multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

# PIC24F16KL402 FAMILY

---

## 3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. Sixteen-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

## 3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided in [Table 3-2](#).

**TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE AND MULTI-BIT SHIFT OPERATION**

Instruction	Description
ASR	Arithmetic shift right source register by one or more bits.
SL	Shift left source register by one or more bits.
LSR	Logical shift right source register by one or more bits.

# PIC24F16KL402 FAMILY

## 4.0 MEMORY ORGANIZATION

As Harvard architecture devices, the PIC24F microcontrollers feature separate program and data memory space and bussing. This architecture also allows the direct access of program memory from the data space during code execution.

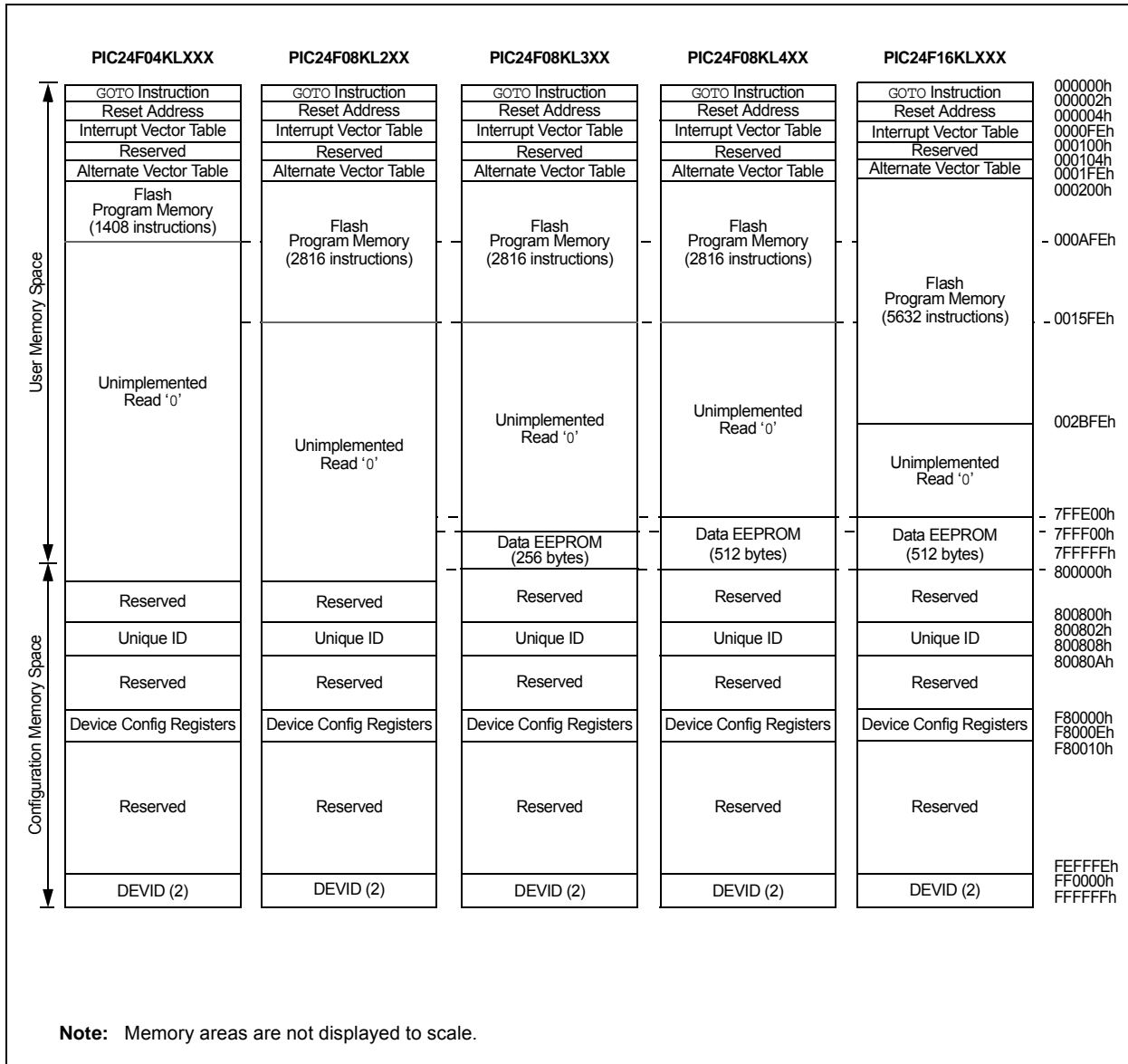
User access to the program memory space is restricted to the lower half of the address range (000000h to 7FFFFFFh). The exception is the use of TBLRD/TBLWT operations, which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space.

### 4.1 Program Address Space

The program address memory space of the PIC24F16KL402 family is 4M instructions. The space is addressable by a 24-bit value derived from either the 23-bit Program Counter (PC) during program execution, or from a table operation or data space remapping, as described in [Section 4.3 “Interfacing Program and Data Memory Spaces”](#).

Memory maps for the PIC24F16KL402 family of devices are shown in [Figure 4-1](#).

**FIGURE 4-1: PROGRAM SPACE MEMORY MAP FOR PIC24F16KL402 FAMILY DEVICES**



# PIC24F16KL402 FAMILY

## 4.1.1 PROGRAM MEMORY ORGANIZATION

The program memory space is organized in word-addressable blocks. Although it is treated as 24 bits wide, it is more appropriate to think of each address of the program memory as a lower and upper word, with the upper byte of the upper word being unimplemented. The lower word always has an even address, while the upper word has an odd address, as shown in [Figure 4-2](#).

Program memory addresses are always word-aligned on the lower word, and addresses are incremented or decremented by two during code execution. This arrangement also provides compatibility with data memory space addressing and makes it possible to access data in the program memory space.

## 4.1.2 HARD MEMORY VECTORS

All PIC24F devices reserve the addresses between 00000h and 000200h for hard-coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A GOTO instruction is programmed by the user at 000000h, with the actual address for the start of code at 000002h.

PIC24F devices also have two Interrupt Vector Tables (IVT), located from 000004h to 0000FFh and 000104h to 0001FFh. These vector tables allow each of the many device interrupt sources to be handled by separate ISRs. A more detailed discussion of the Interrupt Vector Tables is provided in [Section 8.1 “Interrupt Vector Table \(IVT\)”](#).

## 4.1.3 DATA EEPROM

In the PIC24F16KL402 family, the data EEPROM is mapped to the top of the user program memory space, starting at address, 7FFE00, and expanding up to address, 7FFFFF.

The data EEPROM is organized as 16-bit wide memory and 256 words deep. This memory is accessed using Table Read and Table Write operations, similar to the user code memory.

## 4.1.4 DEVICE CONFIGURATION WORDS

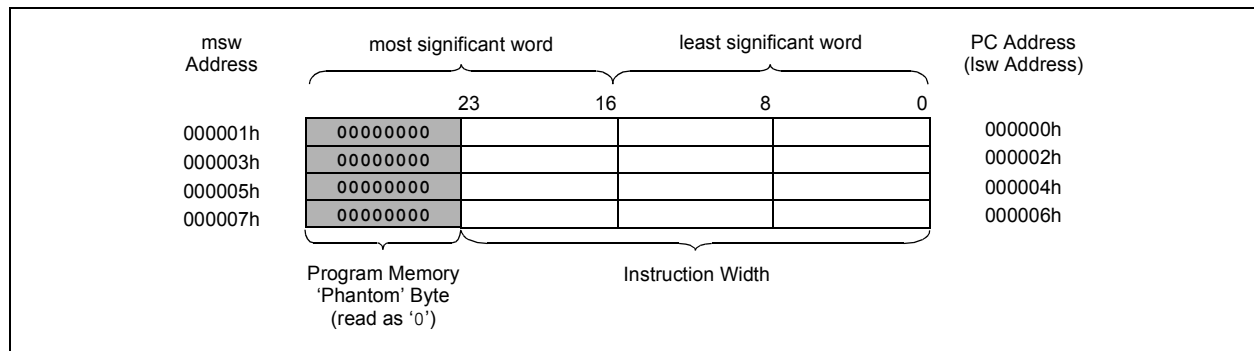
[Table 4-1](#) provides the addresses of the device Configuration Words for the PIC24F16KL402 family. Their location in the memory map is shown in [Figure 4-1](#).

For more information on device Configuration Words, see [Section 23.0 “Special Features”](#).

**TABLE 4-1: DEVICE CONFIGURATION WORDS FOR PIC24F16KL402 FAMILY DEVICES**

Configuration Words	Configuration Word Addresses
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E

**FIGURE 4-2: PROGRAM MEMORY ORGANIZATION**



# PIC24F16KL402 FAMILY

## 4.2 Data Address Space

The PIC24F core has a separate, 16-bit wide data memory space, addressable as a single linear range. The data space is accessed using two Address Generation Units (AGUs); one each for read and write operations. The data space memory map is shown in Figure 4-3.

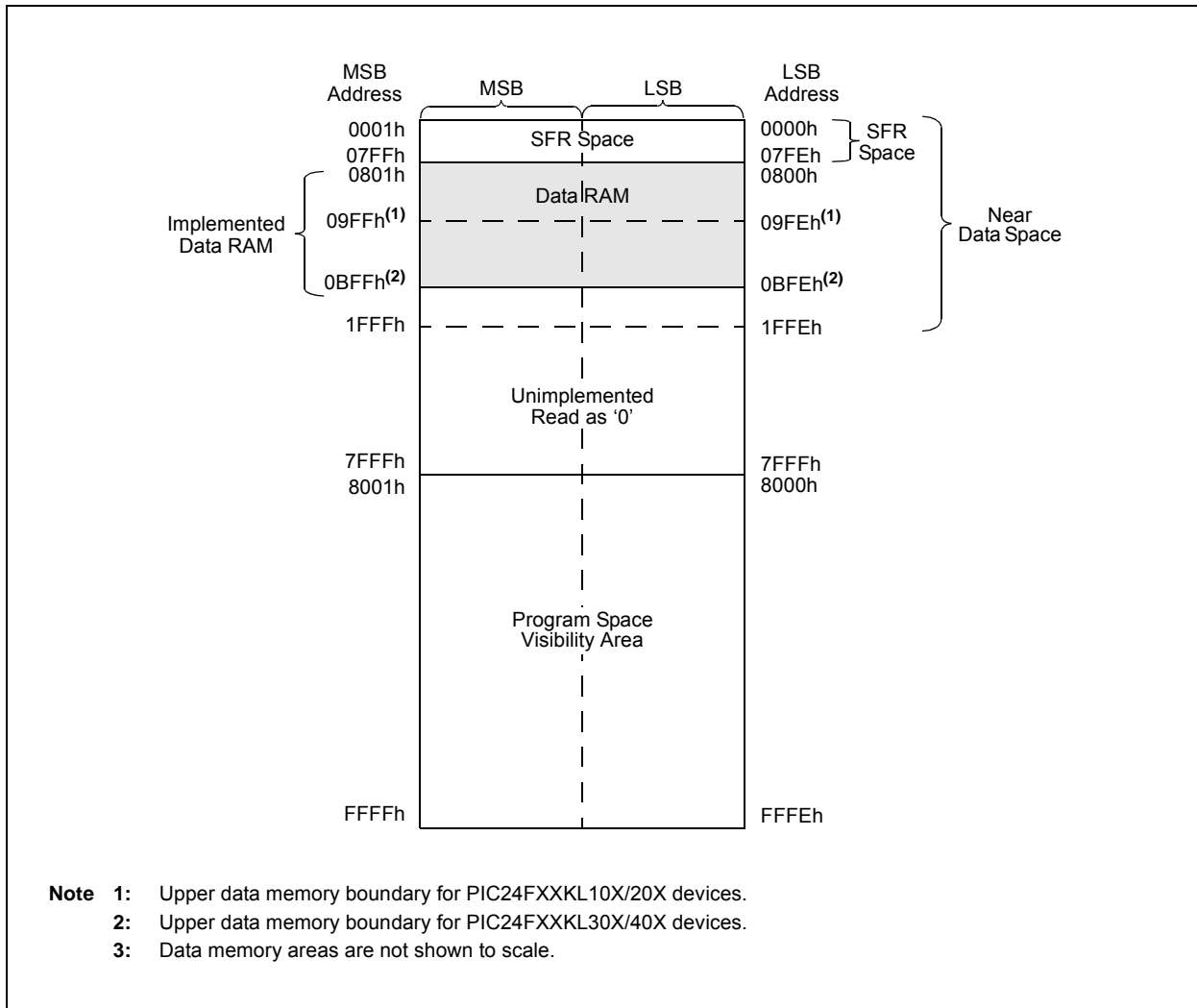
All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when EA<15> = 0) is used for implemented memory addresses, while the upper half (EA<15> = 1) is reserved for the Program Space Visibility (PSV) area (see Section 4.3.3 “Reading Data From Program Memory Using Program Space Visibility”).

Depending on the particular device, PIC24F16KL402 family devices implement either 512 or 1024 words of data memory. If an EA points to a location outside of this area, an all zero word or byte will be returned.

### 4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all the data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

**FIGURE 4-3: DATA SPACE MEMORY MAP FOR PIC24F16KL402 FAMILY DEVICES<sup>(3)</sup>**



# PIC24F16KL402 FAMILY

## 4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® devices and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address (EA) calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word, which contains the byte, using the LSB of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and the registers are organized as two parallel, byte-wide entities with shared (word) address decode, but separate write lines. Data byte writes only write to the corresponding side of the array or register, which matches the byte address.

All word accesses must be aligned to an even address. Mis-aligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a mis-aligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed, but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the LSB; the MSB is not modified.

A Sign-Extend (SE) instruction is provided to allow the users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users

can clear the MSB of any W register by executing a Zero-Extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

## 4.2.3 NEAR DATA SPACE

The 8-Kbyte area between 0000h and 1FFFh is referred to as the Near Data Space (NDS). Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. The remainder of the data space is addressable indirectly. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing (MDA) with a 16-bit address field. For PIC24F16KL402 family devices, the entire implemented data memory lies in Near Data Space.

## 4.2.4 SFR SPACE

The first 2 Kbytes of the Near Data Space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by the module. Much of the SFR space contains unused addresses; these are read as '0'. The SFR space, where the SFRs are actually implemented, is provided in Table 4-2. Each implemented area indicates a 32-byte region, where at least one address is implemented as an SFR. A complete listing of implemented SFRs, including their addresses, is provided in Table 4-3 through Table 4-18.

**TABLE 4-2: IMPLEMENTED REGIONS OF SFR DATA SPACE**

SFR Space Address									
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0	
000h	Core			ICN	Interrupts				—
100h	Timers	—	TMR	—	—	CCP	—	—	—
200h	MSSP	UART		—	—	—	—	I/O	—
300h	A/D			—	—	—	—	—	—
400h	—	—	—	—	—	—	—	—	ANSEL
500h	—	—	—	—	—	—	—	—	—
600h	—	CMP	—	—	—	—	—	—	—
700h	—	—	System	NVM/PMD	—	—	—	—	—

**Legend:** — = No implemented SFRs in this block.

**TABLE 4-3: CPU CORE REGISTERS MAP**

File Name	Start Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	
WREG0	0000	Working Register 0														
WREG1	0002	Working Register 1														
WREG2	0004	Working Register 2														
WREG3	0006	Working Register 3														
WREG4	0008	Working Register 4														
WREG5	000A	Working Register 5														
WREG6	000C	Working Register 6														
WREG7	000E	Working Register 7														
WREG8	0010	Working Register 8														
WREG9	0012	Working Register 9														
WREG10	0014	Working Register 10														
WREG11	0016	Working Register 11														
WREG12	0018	Working Register 12														
WREG13	001A	Working Register 13														
WREG14	001C	Working Register 14														
WREG15	001E	Working Register 15														
SPLIM	0020	Stack Pointer Limit Value Register														
PCL	002E	Program Counter Low Word Register														
PCH	0030	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
TBLPAG	0032	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
PSVPAG	0034	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
RCOUNT	0036	REPEAT Loop Counter Register														
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	
DISICNT	0052	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-4: ICN REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit
CNPD1	0056	CN15PDE <sup>(1)</sup>	CN14PDE <sup>(1)</sup>	CN13PDE <sup>(1)</sup>	CN12PDE	CN11PDE	—	CN9PDE <sup>(2)</sup>	CN8PDE	CN7PDE <sup>(2)</sup>	CN6PDE <sup>(1)</sup>	CN5PDE <sup>(1)</sup>	CN4PDE <sup>(1)</sup>	CN3PDE	CN2
CNPD2	0058	—	CN30PDE	CN29PDE	—	CN27PDE <sup>(2)</sup>	—	—	CN24PDE <sup>(2)</sup>	CN23PDE <sup>(1)</sup>	CN22PDE	CN21PDE	—	—	—
CNEN1	0062	CN15IE <sup>(1)</sup>	CN14IE <sup>(1)</sup>	CN13IE <sup>(1)</sup>	CN12IE	CN11IE	—	CN9IE <sup>(1)</sup>	CN8IE	CN7IE <sup>(1)</sup>	CN6IE <sup>(2)</sup>	CN5PIE <sup>(2)</sup>	CN4IE <sup>(2)</sup>	CN3IE	CN
CNEN2	0064	—	CN30IE	CN29IE	—	CN27IE <sup>(2)</sup>	—	—	CN24IE <sup>(2)</sup>	CN23IE <sup>(1)</sup>	CN22IE	CN21IE	—	—	—
CNPU1	006E	CN15PUE <sup>(1)</sup>	CN14PUE <sup>(1)</sup>	CN13PUE <sup>(1)</sup>	CN12PUE	CN11PUE	—	CN9PUE <sup>(1)</sup>	CN8PUE	CN7PUE <sup>(1)</sup>	CN6PUE <sup>(2)</sup>	CN5PUE <sup>(2)</sup>	CN4PUE <sup>(2)</sup>	CN3PUE	CN2
CNPU2	0070	—	CN30PUE	CN29PUE	—	CN27PUE <sup>(2)</sup>	—	—	CN24PUE <sup>(2)</sup>	CN23PUE <sup>(1)</sup>	CN22PUE	CN21PUE	—	—	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These bits are unimplemented in 14-pin devices; read as '0'.

**2:** These bits are unimplemented in 14-pin and 20-pin devices; read as '0'.

**TABLE 4-5: INTERRUPT CONTROLLER REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR
INTCON2	0082	ALTVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP
IFS0	0084	NVMIF	—	AD1IF	U1TXIF	U1RXIF	—	—	T3IF	T2IF	CCP2IF	—	—	T1IF	CCP1IF
IFS1	0086	U2TXIF	U2RXIF	INT2IF	—	T4IF <sup>(1)</sup>	—	CCP3IF <sup>(1)</sup>	—	—	—	—	INT1IF	CNIF	CMIF
IFS2	0088	—	—	—	—	—	—	—	—	—	—	T3GIF	—	—	—
IFS3	008A	—	—	—	—	—	—	—	—	—	—	—	—	—	BCL2IF <sup>(1)</sup>
IFS4	008C	—	—	—	—	—	—	—	HLVDIF	—	—	—	—	—	U2ERIF
IFS5	008E	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IEC0	0094	NVMIE	—	AD1IE	U1TXIE	U1RXIE	—	—	T3IE	T2IE	CCP2IE	—	—	T1IE	CCP1IE
IEC1	0096	U2TXIE	U2RXIE	INT2IE	—	T4IE <sup>(1)</sup>	—	CCP3IE <sup>(1)</sup>	—	—	—	—	INT1IE	CNIE	CMIE
IEC2	0098	—	—	—	—	—	—	—	—	—	—	T3GIE	—	—	—
IEC3	009A	—	—	—	—	—	—	—	HLVDIE	—	—	—	—	—	BCL2IE <sup>(1)</sup>
IEC4	009C	—	—	—	—	—	—	—	—	—	—	—	—	—	U2ERIE
IEC5	009E	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0	—	—	—	—	—	INT0IP2
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	CCP2IP2	CCP2IP1	CCP2IP0	—	—	—	—	—	—
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	—	—	—	—	—	—	—	—	T3IP2
IPC3	00AA	—	NVMIP2	NVMIP1	NVMIP0	—	—	—	—	—	—	—	AD1IP2	AD1IP1	U1TXIP2
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	BCL1IP2	BCL1IP1	BCL1IP0	—	SSP1IP2
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2
IPC6	00B0	—	T4IP2 <sup>(1)</sup>	T4IP1 <sup>(1)</sup>	T4IP0 <sup>(1)</sup>	—	—	—	—	—	CCP3IP2 <sup>(1)</sup>	CCP3IP1 <sup>(1)</sup>	CCP3IP0 <sup>(1)</sup>	—	—
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	—
IPC9	00B6	—	—	—	—	—	—	—	—	—	T3GIP2	T3GIP1	T3GIP0	—	—
IPC12	00BC	—	—	—	—	—	BCL2IP2 <sup>(1)</sup>	BCL2IP1 <sup>(1)</sup>	BCL2IP0 <sup>(1)</sup>	—	SSP2IP2 <sup>(1)</sup>	SSP2IP1 <sup>(1)</sup>	SSP2IP0 <sup>(1)</sup>	—	—
IPC16	00C4	—	—	—	—	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	HLVDIP2
IPC20	00CC	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIP1
INTTREG	00E0	CPUIRQ	r	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2

**Legend:** — = unimplemented, read as '0', r = reserved. Reset values are shown in hexadecimal.

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X family devices; read as '0'.

**TABLE 4-6: TIMER REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
Timer1 Register															
TMR1	0100														
Timer1 Period Register															
PR1	0102														
T1CON	0104	TON	—	TSIDL	—	T1ECS1	—	T1ECS0	—	TGATE	TCKPS1	TCKPS0	—	—	TSYNK
Timer2 Register															
TMR2	0106	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Timer2 Period Register															
PR2	0108	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Timer3 Register															
T2CON	010A	—	—	—	—	—	—	—	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2O	
Timer4 Register															
TMR3	010C														
T3GCON	010E	—	—	—	—	—	—	—	—	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ T3DONE	T3GVA
Timer4 Period Register															
T3CON	0110	—	—	—	—	—	—	—	—	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCN	T3SYN
TMR4 <sup>(1)</sup>	0112	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PR4 <sup>(1)</sup>	0114	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Timer4 Period Register															
T4CON <sup>(1)</sup>	0116	—	—	—	—	—	—	—	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4O	
CCPTMRSQ <sup>(1)</sup>	013C	—	—	—	—	—	—	—	—	C3TSEL0 <sup>(1)</sup>	—	—	C2TSEL0	—	—

**Legend:** — = unimplemented, read as '0'; Reset values are shown in hexadecimal.

**Note 1:** These bits and/or registers are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X family devices; read as '0'.

**TABLE 4-7: CCP/ECCP REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit
CCP1CON	0190	—	—	—	—	—	—	—	—	PM1 <sup>(1)</sup>	PM0 <sup>(1)</sup>	DC1B1	DC1B0	CCP1M3	CCP1
CCPR1L	0192	—	—	—	—	—	—	—	—	—	—	—	—	—	Register Low By
CCPR1H	0194	—	—	—	—	—	—	—	—	—	—	—	—	—	Register High By
ECCP1DEL <sup>(1)</sup>	0196	—	—	—	—	—	—	—	—	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC
ECCP1AS <sup>(1)</sup>	0198	—	—	—	—	—	—	—	—	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSA1	PSSA
PSTR1CON <sup>(1)</sup>	019A	—	—	—	—	—	—	—	—	CMPL1	CMPL0	—	STRSYNC	STRD	STR
CCP2CON	019C	—	—	—	—	—	—	—	—	—	—	DC2B1	DC2B0	CCP2M3	CCP2
CCPR2L	019E	—	—	—	—	—	—	—	—	—	—	—	—	—	Register Low By
CCPR2H	01A0	—	—	—	—	—	—	—	—	—	—	—	—	—	Register High By
CCP3CON <sup>(1)</sup>	01A8	—	—	—	—	—	—	—	—	—	—	DC3B1	DC3B0	CCP3M3	CCP3
CCPR3L <sup>(1)</sup>	01AA	—	—	—	—	—	—	—	—	—	—	—	—	—	Register Low By
CCPR3H <sup>(1)</sup>	01AC	—	—	—	—	—	—	—	—	—	—	—	—	—	Register High By

**Legend:** — = unimplemented, read as '0'; Reset values are shown in hexadecimal.

**Note 1:** These bits and/or registers are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X family devices; read as '0'.

**TABLE 4-8: MSSP REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit
MSSP1 Receive Buffer/Transmit Register															
SSP1BUF	0200	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSP1CON1	0202	—	—	—	—	—	—	—	—	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSP
SSP1CON2	0204	—	—	—	—	—	—	—	—	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEEN
SSP1CON3	0206	—	—	—	—	—	—	—	—	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBC
SSP1STAT	0208	—	—	—	—	—	—	—	—	SMP	CKE	D/Ā	P	S	R/Ā
SSP1ADD	020A	—	—	—	—	—	—	—	—	MSSP1 Address Register (I <sup>2</sup> C™ Slave Mode) MSSP1 Baud Rate Reload Register (I <sup>2</sup> C Master)					
SSP1MSK	020C	—	—	—	—	—	—	—	—	MSSP1 Address Mask Register (I <sup>2</sup> C Slave Mode)					
SSP2BUF(1)	0210	—	—	—	—	—	—	—	—	MSSP2 Receive Buffer/Transmit Register					
SSP2CON1(1)	0212	—	—	—	—	—	—	—	—	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSP
SSP2CON2(1)	0214	—	—	—	—	—	—	—	—	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEEN
SSP2CON3(1)	0216	—	—	—	—	—	—	—	—	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBC
SSP2STAT(1)	0218	—	—	—	—	—	—	—	—	SMP	CKE	D/Ā	P	S	R/Ā
SSP2ADD(1)	021A	—	—	—	—	—	—	—	—	MSSP2 Address Register (I <sup>2</sup> C Slave Mode) MSSP2 Baud Rate Reload Register (I <sup>2</sup> C Master)					
SSP2MSK(1)	021C	—	—	—	—	—	—	—	—	MSSP2 Address Mask Register (I <sup>2</sup> C Slave Mode)					

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These bits and/or registers are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X family devices; read as '0'.

**TABLE 4-9: UART REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMID	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDISE
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR
U1TXREG	0224	—	—	—	—	—	—	—	—	UART1 Transmit Register					
U1RXREG	0226	—	—	—	—	—	—	—	—	UART1 Receive Register					
U1BRG	0228	Baud Rate Generator Prescaler Register													
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMID	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDISE
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR
U2TXREG	0234	—	—	—	—	—	—	—	—	UART2 Transmit Register					
U2RXREG	0236	—	—	—	—	—	—	—	—	UART2 Receive Register					
U2BRG	0238	Baud Rate Generator Prescaler Register													

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-10: PORTA REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7 <sup>(1)</sup>	Bit 6	Bit 5 <sup>(2)</sup>	Bit 4	Bit 3	Bit 2
TRISA	02C0	—	—	—	—	—	—	—	—	TRISA7	TRISA6	—	TRISA4	TRISA3	TRISA2
PORTA	02C2	—	—	—	—	—	—	—	—	RA7	RA6	RA5	RA4	RA3	RA2
LATA	02C4	—	—	—	—	—	—	—	—	LATA7	LATA6	—	LATA4	LATA3	LATA2
ODCA	02C6	—	—	—	—	—	—	—	—	ODA7	ODA6	—	ODA4	ODA3	ODA2

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These ports and their associated bits are unimplemented on 14-pin and 20-pin devices; read as '0'.

**2:** PORTA<5> is unavailable when MCLR functionality is enabled (MCLRE Configuration bit = 1).

**TABLE 4-11: PORTB REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13 <sup>(1)</sup>	Bit 12 <sup>(1)</sup>	Bit 11 <sup>(2)</sup>	Bit 10 <sup>(2)</sup>	Bit 9	Bit 8	Bit 7 <sup>(1)</sup>	Bit 6 <sup>(2)</sup>	Bit 5 <sup>(2)</sup>	Bit 4	Bit 3 <sup>(2)</sup>	Bit 2
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These ports and their associated bits are unimplemented on 14-pin and 20-pin devices.

**2:** These ports and their associated bits are unimplemented in 14-pin devices.

**TABLE 4-12: PAD CONFIGURATION REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
PADCFG1	02FC	—	—	—	—	SDO2DIS <sup>(1)</sup>	SCK2DIS <sup>(1)</sup>	SDO1DIS	SCK1DIS	—	—	—	—	—	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X family devices; read as '0'.

**TABLE 4-13: A/D REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
ADC1BUF0	0300														
ADC1BUF1	0302														
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAI
AD1CON2	0322	VCFG2	VCFG1	VCFG0	OFFCAL	—	CSCNA	—	—	r	—	SMP13	SMP12	SMP11	SMP
AD1CON3	0324	ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	—	—	ADCS5	ADCS4	ADCS3	ADCS
AD1CHS	0328	CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	—	CH0SA3	CH0S
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12 <sup>(1)</sup>	CSSL11 <sup>(1)</sup>	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	—	CSSL4 <sup>(1)</sup>	—	CSSL

A/D Buffer 0

A/D Buffer 1

**Legend:** — = unimplemented, read as '0', r = reserved bit. Reset values are shown in hexadecimal.

**Note 1:** These bits are unimplemented in 14-pin devices; read as '0'.

**TABLE 4-14: ANALOG SELECT REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
ANCFG	04DE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ANSA	04E0	—	—	—	—	—	—	—	—	—	—	—	—	ANSA3	ANSA
ANSB	04E2	ANSB15	ANSB14	ANSB13	ANSB12 <sup>(1)</sup>	—	—	—	—	—	—	—	ANSB4	ANSB3 <sup>(2)</sup>	ANSB2

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These bits are unimplemented in 14-pin devices; read as '0'.

**2:** These bits are unimplemented in 14-pin and 20-pin devices; read as '0'.

**3:** Reset value for 28-pin devices is shown.

**TABLE 4-15: COMPARATOR REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
CMSTAT	0630	CMIDL	—	—	—	—	—	C2EVT <sup>(1)</sup>	C1EVT	—	—	—	—	—	—
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2
CM1CON	0634	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—
CM2CON <sup>(1)</sup>	0636	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** These bits and/or registers are unimplemented in PIC24FXXKL10X/20X devices; read as '0'.

**TABLE 4-16: SYSTEM REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
RCON	0740	TRAPR	IOPUWR	SBOREN	—	—	—	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	—	LOCK	—	CF	SOSCD
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—
HLVDCON	0756	HLVDEN	—	HLSIDL	—	—	—	—	—	VDJR	BGVST	IRVST	—	HLVDL3	HLVDL

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**Note 1:** RCON register Reset values are dependent on the type of Reset.

**2:** OSCCON register Reset values are dependent on configuration fuses and by type of Reset.

**TABLE 4-17: NVM REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
NVMCON	0760	WR	WREN	WRERR	PGMONLY	—	—	—	—	—	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMO
NVMKEY	0766	—	—	—	—	—	—	—	—	—	—	—	—	—	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-18: ULTRA LOW-POWER WAKE-UP REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
ULPWCON	0768	ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK	—	—	—	—	—	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-19: PMD REGISTER MAP**

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2
PMD1	0770	—	T4MD	T3MD	T2MD	T1MD	—	—	—	SSP1MD	U2MD	U1MD	—	—	—
PMD2	0772	—	—	—	—	—	—	—	—	—	—	—	—	—	CCP3MD
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—
PMD4	0776	—	—	—	—	—	—	—	—	ULPWUMD	—	—	EEMD	REFOMD	—

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

# PIC24F16KL402 FAMILY

## 4.2.5 SOFTWARE STACK

In addition to its use as a Working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer. The pointer always points to the first available free word and grows from lower to higher addresses. It decrements for stack pops and post-increments for stack pushes, as shown in Figure 4-4.

Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

**Note:** A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

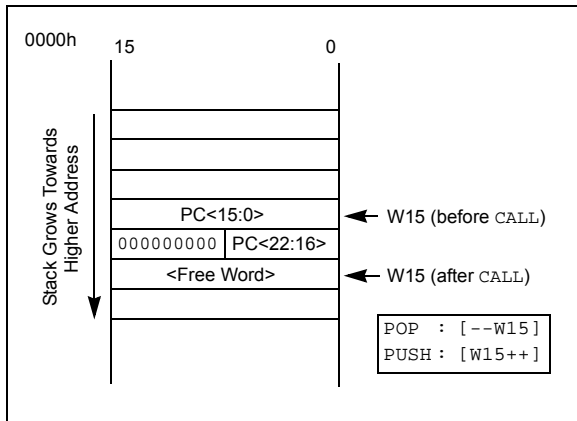
The Stack Pointer Limit Value (SPLIM) register, associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' as all stack operations must be word-aligned. Whenever an EA is generated, using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal, and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation.

Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address, 0DF6, in RAM, initialize the SPLIM with the value, 0DF4.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0800h. This prevents the stack from interfering with the Special Function Register (SFR) space.

**Note:** A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

**FIGURE 4-4: CALL STACK FRAME**



## 4.3 Interfacing Program and Data Memory Spaces

The PIC24F architecture uses a 24-bit wide program space and 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Apart from the normal execution, the PIC24F architecture provides two methods by which the program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space, PSV

Table instructions allow an application to read or write small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look-ups from a large table of static data. It can only access the least significant word (lsw) of the program word.

### 4.3.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Memory Page Address register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the Most Significant bit (MSb) of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 8-bit Program Space Visibility Page Address register (PSVPAG) is used to define a 16K word page in the program space. When the MSb of the EA is '1', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike the table operations, this limits remapping operations strictly to the user memory area.

Table 4-20 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> bits refer to a program space word, whereas the D<15:0> bits refer to a data space word.

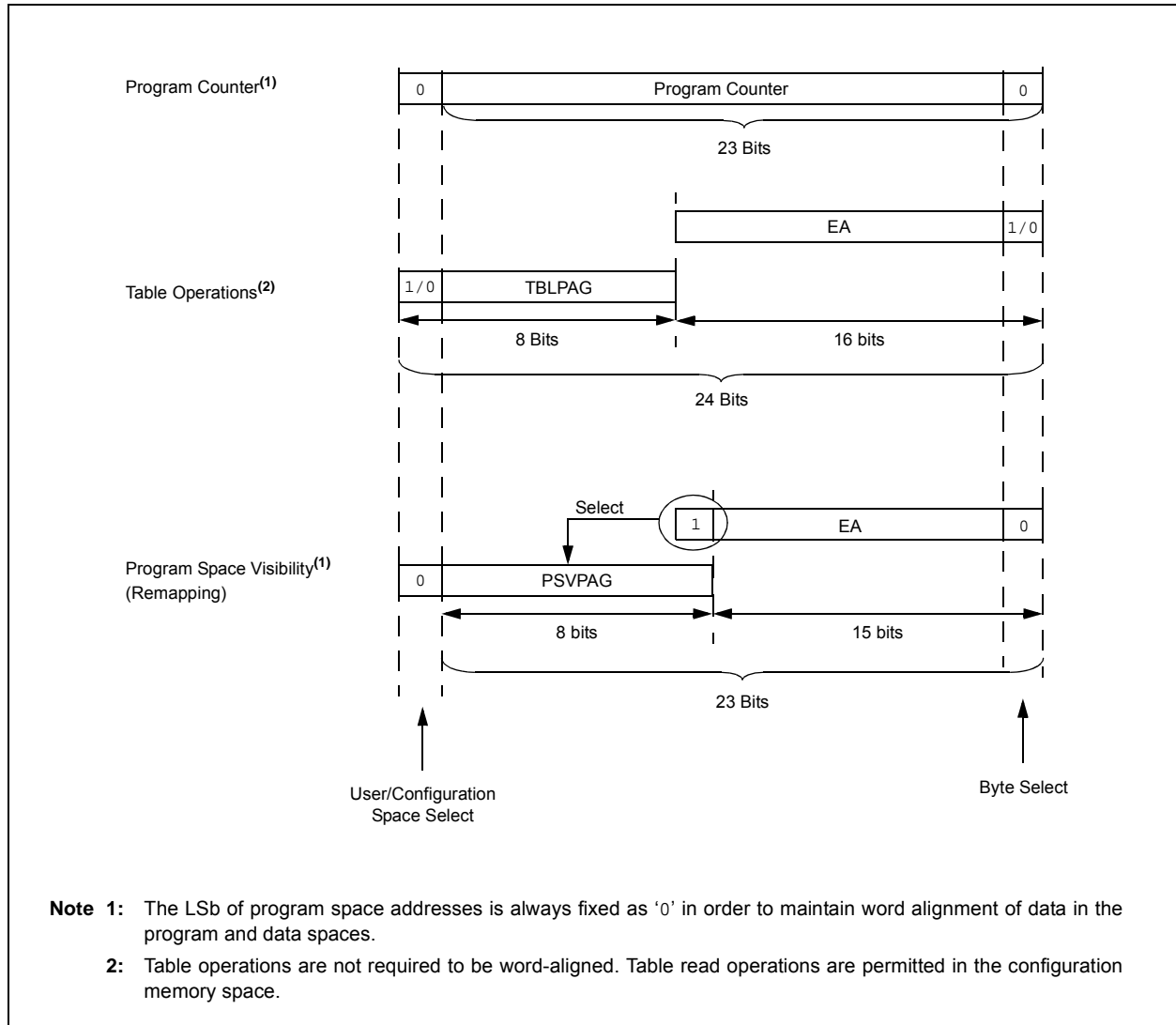
# PIC24F16KL402 FAMILY

**TABLE 4-20: PROGRAM SPACE ADDRESS CONSTRUCTION**

Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access (Code Execution)	User	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT (Byte/Word Read/Write)	User	TBLPAG<7:0>		Data EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	Configuration	TBLPAG<7:0>		Data EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
Program Space Visibility (Block Remap/Read)	User	0	PSVPAG<7:0> <sup>(2)</sup>	Data EA<14:0> <sup>(1)</sup>		
		0	xxxx xxxx	xxx xxxx xxxx xxxx		

- Note 1:** Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is PSVPAG<0>.
- Note 2:** PSVPAG can have only two values ('00' to access program memory and FF to access data EEPROM) on PIC24F16KL402 family devices.

**FIGURE 4-5: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION**



# PIC24F16KL402 FAMILY

## 4.3.2 DATA ACCESS FROM PROGRAM MEMORY AND DATA EEPROM MEMORY USING TABLE INSTRUCTIONS

The `TBLRDL` and `TBLWTL` instructions offer a direct method of reading or writing the lower word of any address within the program memory without going through data space. It also offers a direct method of reading or writing a word of any address within data EEPROM memory. The `TBLRDH` and `TBLWTH` instructions are the only method to read or write the upper 8 bits of a program space word as data.

**Note:** The `TBLRDH` and `TBLWTH` instructions are not used while accessing data EEPROM memory.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to data space addresses. Program memory can thus be regarded as two, 16-bit word-wide address spaces, residing side by side, each with the same address range. `TBLRDL` and `TBLWTL` access the space which contains the least significant data word, and `TBLRDH` and `TBLWTH` access the space which contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

1. `TBLRDL` (Table Read Low): In Word mode, it maps the lower word of the program space location ( $P<15:0>$ ) to a data address ( $D<15:0>$ ).

In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when the byte select is '1'; the lower byte is selected when it is '0'.

2. `TBLRDH` (Table Read High): In Word mode, it maps the entire upper word of a program address ( $P<23:16>$ ) to a data address. Note that  $D<15:8>$ , the 'phantom' byte, will always be '0'.

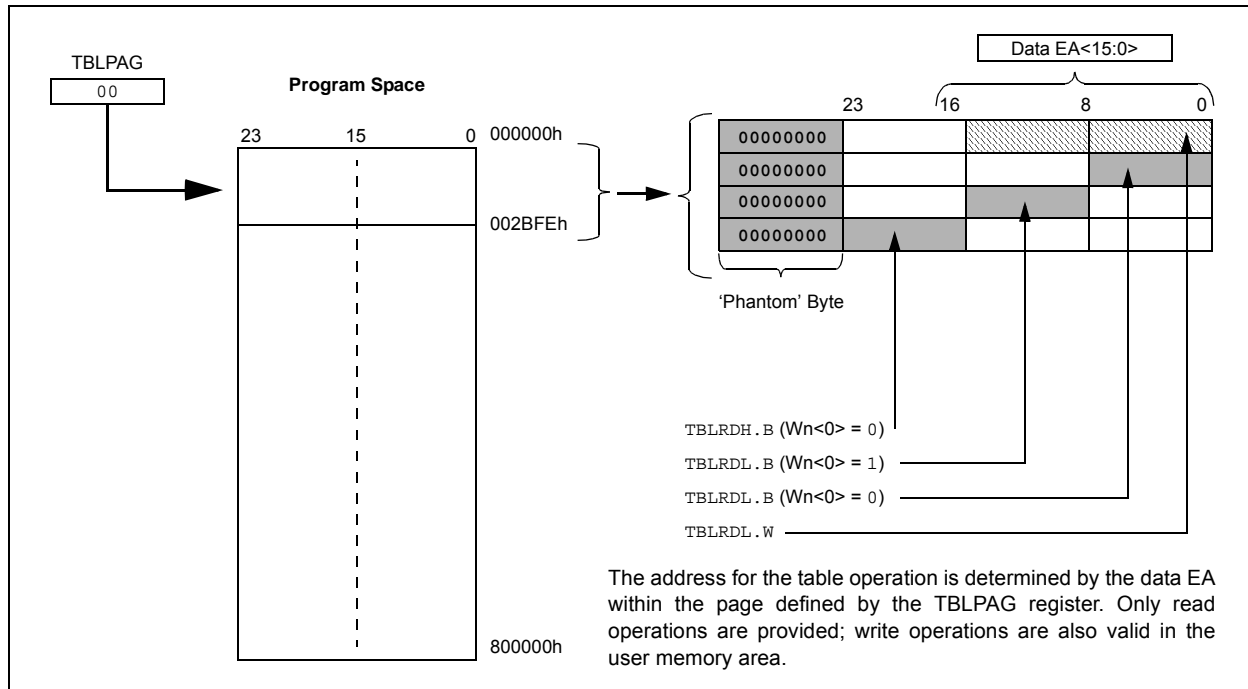
In Byte mode, it maps the upper or lower byte of the program word to  $D<7:0>$  of the data address, as above. Note that the data will always be '0' when the upper 'phantom' byte is selected (byte select = 1).

In a similar fashion, two table instructions, `TBLWTH` and `TBLWTL`, are used to write individual bytes or words to a program space address. The details of their operation are explained in [Section 5.0 "Flash Program Memory"](#).

For all table operations, the area of program memory space to be accessed is determined by the Table Memory Page Address register (`TBLPAG`). `TBLPAG` covers the entire program memory space of the device, including user and configuration spaces. When `TBLPAG<7> = 0`, the table page is located in the user memory space. When `TBLPAG<7> = 1`, the page is located in configuration space.

**Note:** Only Table Read operations will execute in the configuration memory space, and only then, in implemented areas, such as the Device ID. Table write operations are not allowed.

**FIGURE 4-6: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS**



# PIC24F16KL402 FAMILY

## 4.3.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into a 16K word page of the program space. This provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., TBLRDL/H).

Program space access through the data space occurs if the MSb of the data space EA is '1' and PSV is enabled by setting the PSV bit in the CPU Control (CORCON<2>) register. The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page Address (PSVPAG) register. This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with 15 bits of the EA functioning as the lower bits.

By incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads from this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-7), only the lower 16 bits of the

24-bit program word are used to contain the data. The upper 8 bits of any program space location, used as data, should be programmed with '1111 1111' or '0000 0000' to force a NOP. This prevents possible issues should the area of code ever be accidentally executed.

**Note:** PSV access is temporarily disabled during Table Reads/Writes.

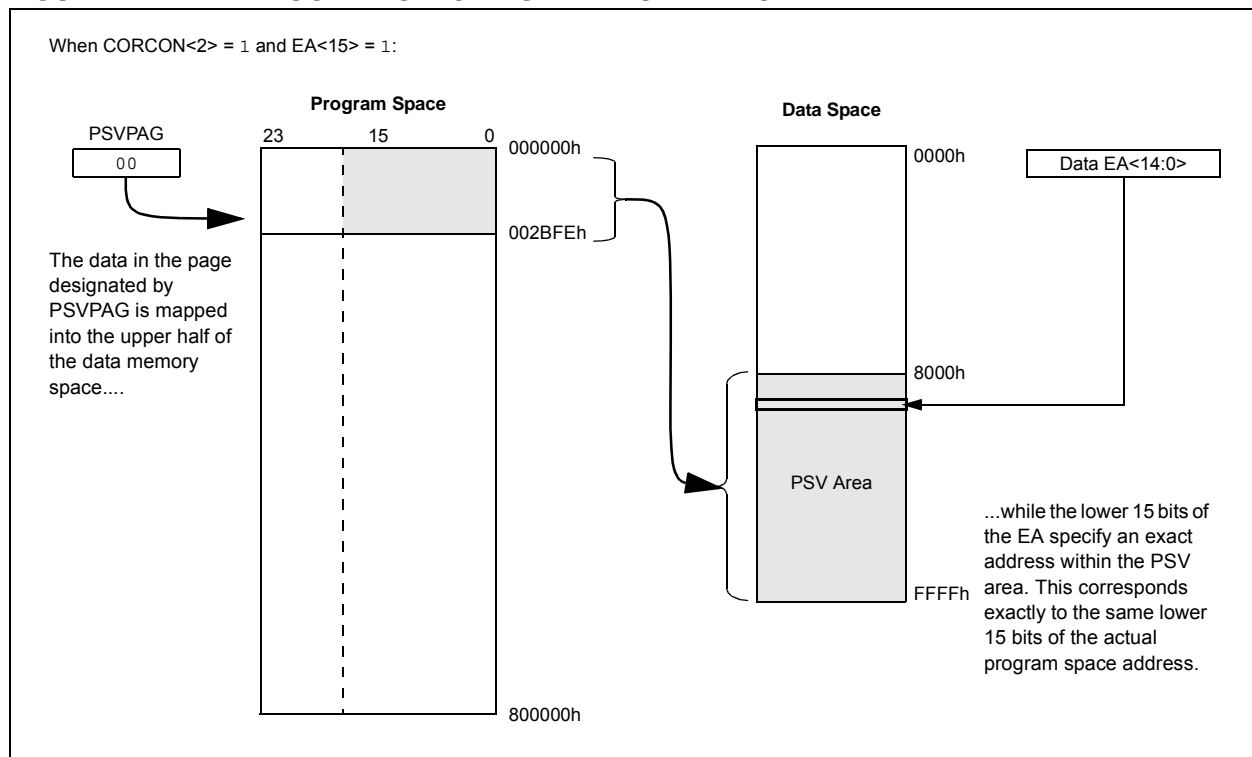
For operations that use PSV and are executed outside of a REPEAT loop, the MOV and MOV.D instructions will require one instruction cycle, in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

For operations that use PSV, which are executed inside a REPEAT loop, there will be some instances that require two instruction cycles, in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction accessing data, using PSV, to execute in a single cycle.

**FIGURE 4-7: PROGRAM SPACE VISIBILITY OPERATION**



# PIC24F16KL402 FAMILY

## 5.0 FLASH PROGRAM MEMORY

**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 Flash Programming, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**Program Memory**” (DS39715).

The PIC24F16KL402 family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 1.8V.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self Programming (RTSP)
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24F device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (which are named PGECx and PGEDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear/Program mode Entry voltage (MCLR/VPP). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or custom firmware to be programmed.

Run-Time Self Programming (RTSP) is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 32 instructions (96 bytes) at a time, and erase program memory in blocks of 32, 64 and 128 instructions (96, 192 and 384 bytes) at a time.

The NVMOP<1:0> (NVMCON<1:0>) bits decide the erase block size.

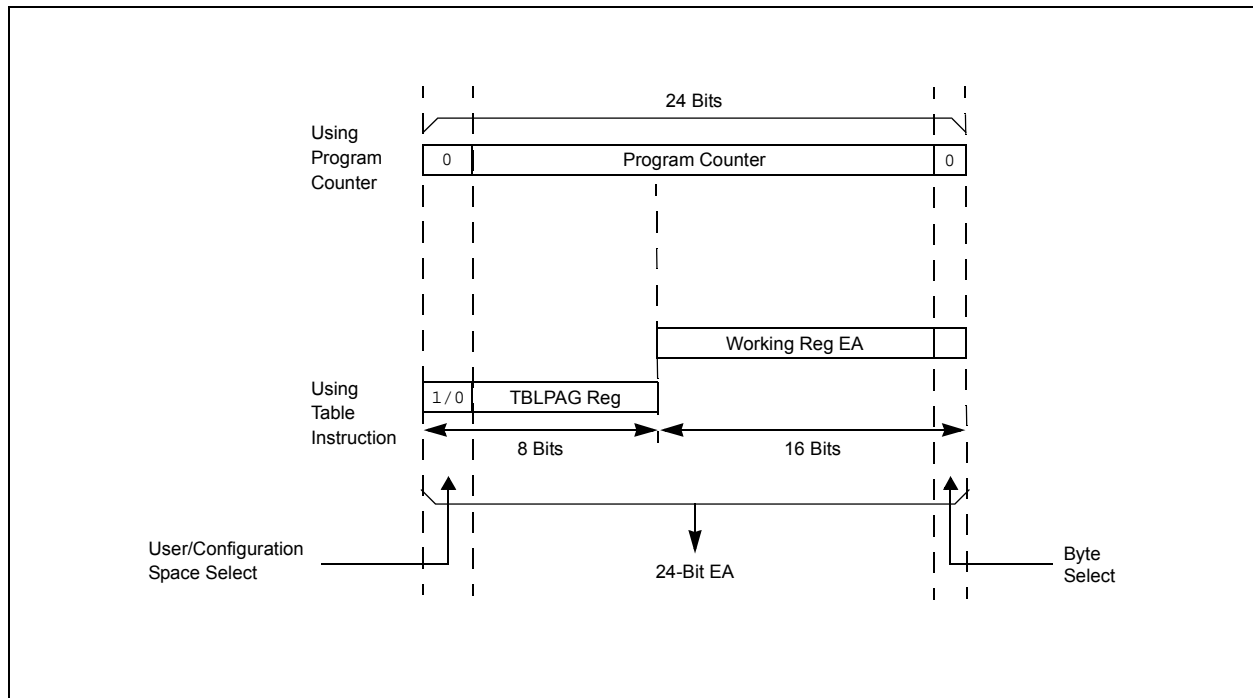
### 5.1 Table Instructions and Flash Programming

Regardless of the method used, Flash memory programming is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as depicted in Figure 5-1.

The TBLRDL and TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



# PIC24F16KL402 FAMILY

---

## 5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 32 instructions or 96 bytes. RTSP allows the user to erase blocks of 1 row, 2 rows and 4 rows (32, 64 and 128 instructions) at a time, and to program one row at a time.

The 1-row (96 bytes), 2-row (192 bytes) and 4-row (384 bytes) erase blocks and single row write block (96 bytes) are edge-aligned, from the beginning of program memory.

When data is written to program memory using `TBLWT` instructions, the data is not written directly to memory. Instead, data written using Table Writes is stored in holding latches until the programming sequence is executed.

Any number of `TBLWT` instructions can be executed and a write will be successfully performed. However, 32 `TBLWT` instructions are required to write the full row of memory.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of `TBLWT` instructions to load the buffers. Programming is performed by setting the control bits in the `NVMCON` register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

<b>Note:</b> Writing to a location multiple times without erasing it is not recommended.
------------------------------------------------------------------------------------------

All of the Table Write operations are single-word writes (two instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

## 5.3 Enhanced In-Circuit Serial Programming

Enhanced ICSP uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

## 5.4 Control Registers

There are two SFRs used to read and write the program Flash memory: `NVMCON` and `NVMKEY`.

The `NVMCON` register ([Register 5-1](#)) controls the blocks that need to be erased, which memory type is to be programmed and when the programming cycle starts.

`NVMKEY` is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the `NVMKEY` register. For more information, refer to [Section 5.5 “Programming Operations”](#).

## 5.5 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the `WR` bit (`NVMCON<15>`) starts the operation and the `WR` bit is automatically cleared when the operation is finished.

# PIC24F16KL402 FAMILY

## REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

R/SO-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY <sup>(4)</sup>	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ERASE	NVMOP5 <sup>(1)</sup>	NVMOP4 <sup>(1)</sup>	NVMOP3 <sup>(1)</sup>	NVMOP2 <sup>(1)</sup>	NVMOP1 <sup>(1)</sup>	NVMOP0 <sup>(1)</sup>
bit 7							bit 0

<b>Legend:</b>	SO = Settable Only bit	HC = Hardware Clearable bit
-n = Value at POR	'1' = Bit is set	R = Readable bit      W = Writable bit
'0' = Bit is cleared	x = Bit is unknown	U = Unimplemented bit, read as '0'

- bit 15      **WR:** Write Control bit  
 1 = Initiates a Flash memory program or erase operation; the operation is self-timed and the bit is cleared by hardware once the operation is complete  
 0 = Program or erase operation is complete and inactive
- bit 14      **WREN:** Write Enable bit  
 1 = Enables Flash program/erase operations  
 0 = Inhibits Flash program/erase operations
- bit 13      **WRERR:** Write Sequence Error Flag bit  
 1 = An improper program or erase sequence attempt, or termination, has occurred (bit is set automatically on any set attempt of the WR bit)  
 0 = The program or erase operation completed normally
- bit 12      **PGMONLY:** Program Only Enable bit<sup>(4)</sup>
- bit 11-7    **Unimplemented:** Read as '0'
- bit 6      **ERASE:** Erase/Program Enable bit  
 1 = Performs the erase operation specified by NVMOP<5:0> on the next WR command  
 0 = Performs the program operation specified by NVMOP<5:0> on the next WR command
- bit 5-0    **NVMOP<5:0>:** Programming Operation Command Byte bits<sup>(1)</sup>  
Erase Operations (when ERASE bit is '1'):  
 1010xx = Erases entire boot block (including code-protected boot block)<sup>(2)</sup>  
 1001xx = Erases entire memory (including boot block, configuration block, general block)<sup>(2)</sup>  
 011010 = Erases 4 rows of Flash memory<sup>(3)</sup>  
 011001 = Erases 2 rows of Flash memory<sup>(3)</sup>  
 011000 = Erases 1 row of Flash memory<sup>(3)</sup>  
 0101xx = Erases entire configuration block (except code protection bits)  
 0100xx = Erases entire data EEPROM<sup>(4)</sup>  
 0011xx = Erases entire general memory block programming operations  
 0001xx = Writes 1 row of Flash memory (when ERASE bit is '0')<sup>(3)</sup>

- Note 1:** All other combinations of the NVMOP<5:0> bits are no operation.  
**Note 2:** Available in ICSP™ mode only. Refer to the device programming specification.  
**Note 3:** The address in the Table Pointer decides which rows will be erased.  
**Note 4:** This bit is used only while accessing data EEPROM. It is implemented only in devices with data EEPROM.

# PIC24F16KL402 FAMILY

## 5.5.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time by erasing the programmable row. The general process is as follows:

1. Read a row of program memory (32 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase a row (see [Example 5-1](#)):
  - a) Set the NVMOPx bits (NVMCON<5:0>) to '011000' to configure for row erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
  - b) Write the starting address of the block to be erased into the TBLPAG and W registers.
  - c) Write 55h to NVMKEY.
  - d) Write AAh to NVMKEY.
  - e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 32 instructions from data RAM into the program memory buffers (see [Example 5-1](#)).
5. Write the program block to Flash memory:
  - a) Set the NVMOPx bits to '000100' to configure for row programming. Clear the ERASE bit and set the WREN bit.
  - b) Write 55h to NVMKEY.
  - c) Write AAh to NVMKEY.
  - d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as shown in [Example 5-5](#).

### EXAMPLE 5-1: ERASING A PROGRAM MEMORY ROW – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row erase operation
MOV    #0x4058, W0          ;
MOV    W0, NVMCON          ; Initialize NVMCON
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0 ;
MOV    W0, TBLPAG          ; Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR), W0 ; Initialize in-page EA[15:0] pointer
TBLWTL W0, [W0]           ; Set base address of erase block
DISI   #5                  ; Block all interrupts
                          ; for next 5 instructions

MOV    #0x55, W0
MOV    W0, NVMKEY          ; Write the 55 key
MOV    #0xAA, W1
MOV    W1, NVMKEY          ; Write the AA key
BSET   NVMCON, #WR        ; Start the erase sequence
NOP    ; Insert two NOPs after the erase
NOP    ; command is asserted
```

# PIC24F16KL402 FAMILY

## EXAMPLE 5-2: ERASING A PROGRAM MEMORY ROW – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

int __attribute__((space(auto_psv))) progAddr = &progAddr; // Global variable located in Pgm Memory
unsigned int offset;

//Set up pointer to the first memory location to be written

TBLPAG = __builtin_tblpage(&progAddr); // Initialize PM Page Boundary SFR
offset = &progAddr & 0xFFFF; // Initialize lower word of address

__builtin_tblwtl(offset, 0x0000); // Set base address of erase block
// with dummy latch write

NVMCON = 0x4058; // Initialize NVMCON

asm("DISI #5"); // Block all interrupts for next 5
// instructions
__builtin_write_NVM(); // C30 function to perform unlock
// sequence and set WR
```

## EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row programming operations
MOV    #0x4004, W0 ;
MOV    W0, NVMCON ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
MOV    #0x0000, W0 ;
MOV    W0, TBLPAG ; Initialize PM Page Boundary SFR
MOV    #0x6000, W0 ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
MOV    #LOW_WORD_0, W2 ;
MOV    #HIGH_BYTE_0, W3 ;
TBLWTL W2, [W0] ; Write PM low word into program latch
TBLWTH W3, [W0++] ; Write PM high byte into program latch
; 1st_program_word
MOV    #LOW_WORD_1, W2 ;
MOV    #HIGH_BYTE_1, W3 ;
TBLWTL W2, [W0] ; Write PM low word into program latch
TBLWTH W3, [W0++] ; Write PM high byte into program latch
; 2nd_program_word
MOV    #LOW_WORD_2, W2 ;
MOV    #HIGH_BYTE_2, W3 ;
TBLWTL W2, [W0] ; Write PM low word into program latch
TBLWTH W3, [W0++] ; Write PM high byte into program latch
.
.
.
; 32nd_program_word
MOV    #LOW_WORD_31, W2 ;
MOV    #HIGH_BYTE_31, W3 ;
TBLWTL W2, [W0] ; Write PM low word into program latch
TBLWTH W3, [W0] ; Write PM high byte into program latch
```

# PIC24F16KL402 FAMILY

## EXAMPLE 5-4: LOADING THE WRITE BUFFERS – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

#define NUM_INSTRUCTION_PER_ROW 64
int __attribute__((space(auto_psv))) progAddr = &progAddr; // Global variable located in Pgm Memory
unsigned int offset;
unsigned int i;
unsigned int progData[2*NUM_INSTRUCTION_PER_ROW]; // Buffer of data to write

//Set up NVMCON for row programming
NVMCON = 0x4004; // Initialize NVMCON

//Set up pointer to the first memory location to be written
TBLPAG = __builtin_tblpage(&progAddr); // Initialize PM Page Boundary SFR
offset = &progAddr & 0xFFFF; // Initialize lower word of address

//Perform TBLWT instructions to write necessary number of latches
for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
{
    __builtin_tblwtl(offset, progData[i++]); // Write to address low word
    __builtin_tblwth(offset, progData[i]); // Write to upper byte
    offset = offset + 2; // Increment address
}
```

## EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE – ASSEMBLY LANGUAGE CODE

```
DISI    #5                ; Block all interrupts
                        ; for next 5 instructions

MOV     #0x55, W0
MOV     W0, NVMKEY       ; Write the 55 key
MOV     #0xAA, W1
MOV     W1, NVMKEY       ; Write the AA key
BSET    NVMCON, #WR      ; Start the erase sequence
NOP     ; 2 NOPs required after setting WR
NOP     ;
BTSC    NVMCON, #15      ; Wait for the sequence to be completed
BRA     $-2              ;
```

## EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

asm("DISI #5"); // Block all interrupts for next 5 instructions

__builtin_write_NVM(); // Perform unlock sequence and set WR
```

## 6.0 DATA EEPROM MEMORY

**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 Data EEPROM, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**Data EEPROM**” (DS39720).

The data EEPROM memory is a Nonvolatile Memory (NVM), separate from the program and volatile data RAM. Data EEPROM memory is based on the same Flash technology as program memory, and is optimized for both long retention and a higher number of erase/write cycles.

The data EEPROM is mapped to the top of the user program memory space, with the top address at program memory address, 7FFFFFFh. For PIC24FXXKL4XX devices, the size of the data EEPROM is 256 words (7FFE00h to 7FFFFFFh). For PIC24FXXKL3XX devices, the size of the data EEPROM is 128 words (7FFF00h to 7FFFFFFh). The data EEPROM is not implemented in PIC24F08KL20X or PIC24F04KL10X devices.

The data EEPROM is organized as 16-bit wide memory. Each word is directly addressable, and is readable and writable during normal operation over the entire VDD range.

Unlike the Flash program memory, normal program execution is not stopped during a data EEPROM program or erase operation.

The data EEPROM programming operations are controlled using the three NVM Control registers:

- NVMCON: Nonvolatile Memory Control Register
- NVMKEY: Nonvolatile Memory Key Register
- NVMADR: Nonvolatile Memory Address Register

### EXAMPLE 6-1: DATA EEPROM UNLOCK SEQUENCE

```
//Disable Interrupts For 5 instructions
asm volatile("disi #5");
//Issue Unlock Sequence
asm volatile ("mov #0x55, W0      \n"
              "mov W0, NVMKEY     \n"
              "mov #0xAA, W1     \n"
              "mov W1, NVMKEY     \n");
// Perform Write/Erase operations
asm volatile ("bset NVMCON, #WR  \n"
              "nop                \n"
              "nop                \n");
```

## 6.1 NVMCON Register

The NVMCON register ([Register 6-1](#)) is also the primary control register for data EEPROM program/erase operations. The upper byte contains the control bits used to start the program or erase cycle, and the flag bit to indicate if the operation was successfully performed. The lower byte of NVMCOM configures the type of NVM operation that will be performed.

## 6.2 NVMKEY Register

The NVMKEY is a write-only register that is used to prevent accidental writes or erasures of data EEPROM locations.

To start any programming or erase sequence, the following instructions must be executed first, in the exact order provided:

1. Write 55h to NVMKEY.
2. Write AAh to NVMKEY.

After this sequence, a write will be allowed to the NVMCON register for one instruction cycle. In most cases, the user will simply need to set the WR bit in the NVMCON register to start the program or erase cycle. Interrupts should be disabled during the unlock sequence.

The MPLAB® C30 C compiler provides a defined library procedure (`builtin_write_NVM`) to perform the unlock sequence. [Example 6-1](#) illustrates how the unlock sequence can be performed with in-line assembly.

# PIC24F16KL402 FAMILY

## REGISTER 6-1: NVMCON: NONVOLATILE MEMORY CONTROL REGISTER

R/SO-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY	—	—	—	—
bit 15						bit 8	

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ERASE	NVMOP5 <sup>(1)</sup>	NVMOP4 <sup>(1)</sup>	NVMOP3 <sup>(1)</sup>	NVMOP2 <sup>(1)</sup>	NVMOP1 <sup>(1)</sup>	NVMOP0 <sup>(1)</sup>
bit 7							bit 0

<b>Legend:</b>	HC = Hardware Clearable bit	U = Unimplemented bit, read as '0'
R = Readable bit	W = Writable bit	SO = Settable Only bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared      x = Bit is unknown

- bit 15      **WR:** Write Control bit (program or erase)  
 1 = Initiates a data EEPROM erase or write cycle (can be set but not cleared in software)  
 0 = Write cycle is complete (cleared automatically by hardware)
- bit 14      **WREN:** Write Enable bit (erase or program)  
 1 = Enables an erase or program operation  
 0 = No operation allowed (device clears this bit on completion of the write/erase operation)
- bit 13      **WRERR:** Flash Error Flag bit  
 1 = A write operation is prematurely terminated (any  $\overline{\text{MCLR}}$  or WDT Reset during programming operation)  
 0 = The write operation completed successfully
- bit 12      **PGMONLY:** Program Only Enable bit  
 1 = Write operation is executed without erasing target address(es) first  
 0 = Automatic erase-before-write; write operations are preceded automatically by an erase of target address(es)
- bit 11-7    **Unimplemented:** Read as '0'
- bit 6      **ERASE:** Erase Operation Select bit  
 1 = Performs an erase operation when WR is set  
 0 = Performs a write operation when WR is set
- bit 5-0    **NVMOP<5:0>:** Programming Operation Command Byte bits<sup>(1)</sup>  
Erase Operations (when ERASE bit is '1'):  
 011010 = Erases 8 words  
 011001 = Erases 4 words  
 011000 = Erases 1 word  
 0100xx = Erases entire data EEPROM  
Programming Operations (when ERASE bit is '0'):  
 001xxx = Writes 1 word

**Note 1:** These NVMOP configurations are unimplemented on PIC24F04KL10X and PIC24F08KL20X devices.

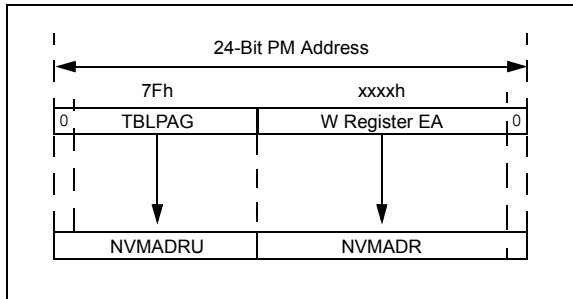
## 6.3 NVM Address Register

As with Flash program memory, the NVM Address Registers, NVMADRU and NVMADR, form the 24-bit Effective Address (EA) of the selected row or word for data EEPROM operations. The NVMADRU register is used to hold the upper 8 bits of the EA, while the NVMADR register is used to hold the lower 16 bits of the EA. These registers are not mapped into the Special Function Register (SFR) space; instead, they directly capture the EA<23:0> of the last Table Write instruction that has been executed and selects the data EEPROM row to erase. Figure 6-1 depicts the program memory EA that is formed for programming and erase operations.

Like program memory operations, the Least Significant bit (LSb) of NVMADR is restricted to even addresses. This is because any given address in the data EEPROM space consists of only the lower word of the program memory width; the upper word, including the uppermost “phantom byte”, is unavailable. This means that the LSb of a data EEPROM address will always be ‘0’.

Similarly, the Most Significant bit (MSb) of NVMADRU is always ‘0’, since all addresses lie in the user program space.

**FIGURE 6-1: DATA EEPROM ADDRESSING WITH TBLPAG AND NVM ADDRESS REGISTERS**



## 6.4 Data EEPROM Operations

The EEPROM block is accessed using Table Read and Table Write operations, similar to those used for program memory. The TBLWTH and TBLRDH instructions are not required for data EEPROM operations since the memory is only 16 bits wide (data on the lower address is valid only). The following programming operations can be performed on the data EEPROM:

- Erase one, four or eight words
- Bulk erase the entire data EEPROM
- Write one word
- Read one word

**Note:** Unexpected results will be obtained if the user attempts to read the EEPROM while a programming or erase operation is underway.

The C30 C compiler includes library procedures to automatically perform the Table Read and Table Write operations, manage the Table Pointer and write buffers, and unlock and initiate memory write sequences. This eliminates the need to create assembler macros or time critical routines in C for each application.

The library procedures are used in the code examples detailed in the following sections. General descriptions of each process are provided for users who are not using the C30 compiler libraries.

# PIC24F16KL402 FAMILY

---

## 6.4.1 ERASE DATA EEPROM

The data EEPROM can be fully erased, or can be partially erased, at three different sizes: one word, four words or eight words. The bits, NVMOP<1:0> (NVMCON<1:0>), decide the number of words to be erased. To erase partially from the data EEPROM, the following sequence must be followed:

1. Configure NVMCON to erase the required number of words: one, four or eight.
2. Load TBLPAG and WREG with the EEPROM address to be erased.
3. Clear the NVMIF status bit and enable the NVM interrupt (optional).
4. Write the key sequence to NVMKEY.
5. Set the WR bit to begin the erase cycle.
6. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical erase sequence is provided in [Example 6-2](#). This example shows how to do a one-word erase. Similarly, a four-word erase and an eight-word erase can be done. This example uses C library procedures to manage the Table Pointer (`builtin_tblpage` and `builtin_tbloffset`) and the Erase Page Pointer (`builtin_tblwtl`). The memory unlock sequence (`builtin_write_NVM`) also sets the WR bit to initiate the operation and returns control when complete.

### EXAMPLE 6-2: SINGLE-WORD ERASE

```
int __attribute__((space(eedata))) eeData = 0x1234; // Global variable located in EEPROM
unsigned int offset;

// Set up NVMCON to erase one word of data EEPROM
NVMCON = 0x4058;

// Set up a pointer to the EEPROM location to be erased
TBLPAG = __builtin_tblpage(&eeData); // Initialize EE Data page pointer
offset = __builtin_tbloffset(&eeData); // Initialize lower word of address
__builtin_tblwtl(offset, 0); // Write EEPROM data to write latch

asm volatile ("disi #5"); // Disable Interrupts For 5 Instructions
__builtin_write_NVM(); // Issue Unlock Sequence & Start Write Cycle
while(NVMCONbits.WR=1); // Optional: Poll WR bit to wait for
// write sequence to complete
```

# PIC24F16KL402 FAMILY

## 6.4.1.1 Data EEPROM Bulk Erase

To erase the entire data EEPROM (bulk erase), the address registers do not need to be configured because this operation affects the entire data EEPROM. The following sequence helps in performing a bulk erase:

1. Configure NVMCON to Bulk Erase mode.
2. Clear the NVMIF status bit and enable the NVM interrupt (optional).
3. Write the key sequence to NVMKEY.
4. Set the WR bit to begin the erase cycle.
5. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical bulk erase sequence is provided in [Example 6-3](#).

## 6.4.2 SINGLE-WORD WRITE

To write a single word in the data EEPROM, the following sequence must be followed:

1. Erase one data EEPROM word (as mentioned in [Section 6.4.1 “Erase Data EEPROM”](#)) if PGMONLY bit (NVMCON<12>) is set to ‘1’.
2. Write the data word into the data EEPROM latch.
3. Program the data word into the EEPROM:
  - Configure the NVMCON register to program one EEPROM word (NVMCON<5:0> = 0001xxx).
  - Clear the NVMIF status bit and enable the NVM interrupt (optional).
  - Write the key sequence to NVMKEY.
  - Set the WR bit to begin the erase cycle.
  - Either poll the WR bit or wait for the NVM interrupt (NVMIF set).
  - To get cleared, wait until NVMIF is set.

A typical single-word write sequence is provided in [Example 6-4](#).

### EXAMPLE 6-3: DATA EEPROM BULK ERASE

```
// Set up NVMCON to bulk erase the data EEPROM
NVMCON = 0x4050;

// Disable Interrupts For 5 Instructions
asm volatile ("disi #5");

// Issue Unlock Sequence and Start Erase Cycle
__builtin_write_NVM();
```

### EXAMPLE 6-4: SINGLE-WORD WRITE TO DATA EEPROM

```
int __attribute__((space(eedata))) eeData = 0x1234; // Global variable located in EEPROM
int newData; // New data to write to EEPROM
unsigned int offset;

// Set up NVMCON to erase one word of data EEPROM
NVMCON = 0x4004;

// Set up a pointer to the EEPROM location to be erased
TBLPAG = __builtin_tblpage(&eeData); // Initialize EE Data page pointer
offset = __builtin_tbloffset(&eeData); // Initialize lower word of address
__builtin_tblwtl(offset, newData); // Write EEPROM data to write latch

asm volatile ("disi #5"); // Disable Interrupts For 5 Instructions
__builtin_write_NVM(); // Issue Unlock Sequence & Start Write Cycle
while(NVMCONbits.WR=1); // Optional: Poll WR bit to wait for
// write sequence to complete
```

# PIC24F16KL402 FAMILY

---

## 6.4.3 READING THE DATA EEPROM

To read a word from data EEPROM, the Table Read instruction is used. Since the EEPROM array is only 16 bits wide, only the TBLRDL instruction is needed. The read operation is performed by loading TBLPAG and WREG with the address of the EEPROM location followed by a TBLRDL instruction.

A typical read sequence using the Table Pointer management (`builtin_tblpage` and `builtin_tbloffset`) and Table Read (`builtin_tblrld`) procedures from the C30 compiler library is provided in [Example 6-5](#).

Program Space Visibility (PSV) can also be used to read locations in the data EEPROM.

### EXAMPLE 6-5: READING THE DATA EEPROM USING THE TBLRD COMMAND

```
int __attribute__((space(eedata))) eeData = 0x1234; // Global variable located in EEPROM
int data; // Data read from EEPROM
unsigned int offset;

// Set up a pointer to the EEPROM location to be erased
TBLPAG = __builtin_tblpage(&eeData); // Initialize EE Data page pointer
offset = __builtin_tbloffset(&eeData); // Initialize lower word of address
data = __builtin_tblrld(offset); // Write EEPROM data to write latch
```

# PIC24F16KL402 FAMILY

## 7.0 RESETS

**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 Resets, refer to the “dsPIC33/PIC24 Family Reference Manual”, “Reset with Programmable Brown-out Reset” (DS39728).

The Reset module combines all Reset sources and controls the device Master Reset Signal,  $\overline{\text{SYSRST}}$ . The following is a list of device Reset sources:

- POR: Power-on Reset
- MCLR: Pin Reset
- SWR: RESET Instruction
- WDTR: Watchdog Timer Reset
- BOR: Brown-out Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode Reset
- UWR: Uninitialized W Register Reset

A simplified block diagram of the Reset module is shown in Figure 7-1.

Any active source of Reset will make the  $\overline{\text{SYSRST}}$  signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state. Most registers are unaffected by a Reset; their status is unknown on a Power-on Reset (POR) and unchanged by all other Resets.

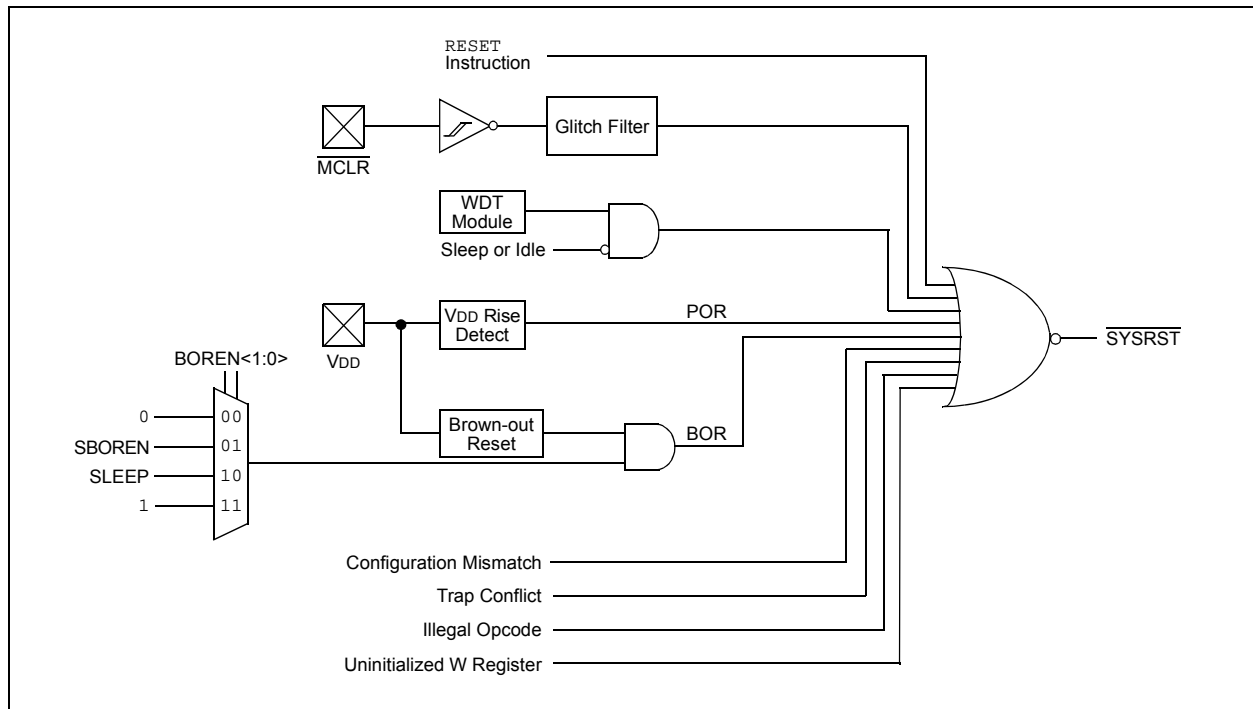
**Note:** Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 7-1). A POR will clear all bits except for the BOR and POR bits ( $\text{RCON}\langle 1:0 \rangle$ ) which are set. The user may set or clear any bit at any time during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software will not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer (WDT) and device power-saving states. The function of these bits is discussed in other sections of this manual.

**Note:** The status bits in the RCON register should be cleared after they are read so that the next RCON register value, after a device Reset, will be meaningful.

FIGURE 7-1: RESET SYSTEM BLOCK DIAGRAM



# PIC24F16KL402 FAMILY

## REGISTER 7-1: RCON: RESET CONTROL REGISTER<sup>(1)</sup>

R/W-0	R/W-0	R/W-0 <sup>(3)</sup>	U-0	U-0	U-0	R/W-0	R/W-0
TRAPR	IOPUWR	SBOREN	—	—	—	CM	PMSLP
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR	SWR	SWDTEN <sup>(2)</sup>	WDTO	SLEEP	IDLE	BOR	POR
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        **TRAPR:** Trap Reset Flag bit  
 1 = A Trap Conflict Reset has occurred  
 0 = A Trap Conflict Reset has not occurred
- bit 14        **IOPUWR:** Illegal Opcode or Uninitialized W Access Reset Flag bit  
 1 = An illegal opcode detection, an illegal address mode or an Uninitialized W register is used as an Address Pointer and caused a Reset  
 0 = An illegal opcode or Uninitialized W register Reset has not occurred
- bit 13        **SBOREN:** Software Enable/Disable of BOR bit<sup>(3)</sup>  
 1 = BOR is turned on in software  
 0 = BOR is turned off in software
- bit 12-10     **Unimplemented:** Read as '0'
- bit 9         **CM:** Configuration Word Mismatch Reset Flag bit  
 1 = A Configuration Word Mismatch Reset has occurred  
 0 = A Configuration Word Mismatch Reset has not occurred
- bit 8         **PMSLP:** Program Memory Power During Sleep bit  
 1 = Program memory bias voltage remains powered during Sleep  
 0 = Program memory bias voltage is powered down during Sleep
- bit 7         **EXTR:** External Reset ( $\overline{\text{MCLR}}$ ) Pin bit  
 1 = A Master Clear (pin) Reset has occurred  
 0 = A Master Clear (pin) Reset has not occurred
- bit 6         **SWR:** Software Reset (Instruction) Flag bit  
 1 = A `RESET` instruction has been executed  
 0 = A `RESET` instruction has not been executed
- bit 5         **SWDTEN:** Software Enable/Disable of WDT bit<sup>(2)</sup>  
 1 = WDT is enabled  
 0 = WDT is disabled
- bit 4         **WDTO:** Watchdog Timer Time-out Flag bit  
 1 = WDT time-out has occurred  
 0 = WDT time-out has not occurred

**Note 1:** All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

**Note 2:** If the `SWDTEN` Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the `SWDTEN` bit setting.

**Note 3:** The `SBOREN` bit is forced to '0' when disabled by the Configuration bits, `BOREN<1:0>` (`FPOR<1:0>`). When the Configuration bits are set to enable `SBOREN`, the default Reset state will be '1'.

# PIC24F16KL402 FAMILY

## REGISTER 7-1: RCON: RESET CONTROL REGISTER<sup>(1)</sup> (CONTINUED)

bit 3	<b>SLEEP:</b> Wake-up from Sleep Flag bit 1 = Device has been in Sleep mode 0 = Device has not been in Sleep mode
bit 2	<b>IDLE:</b> Wake-up from Idle Flag bit 1 = Device has been in Idle mode 0 = Device has not been in Idle mode
bit 1	<b>BOR:</b> Brown-out Reset Flag bit 1 = A Brown-out Reset has occurred (the BOR is also set after a POR) 0 = A Brown-out Reset has not occurred
bit 0	<b>POR:</b> Power-on Reset Flag bit 1 = A Power-up Reset has occurred 0 = A Power-up Reset has not occurred

**Note 1:** All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

**2:** If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

**3:** The SBOREN bit is forced to '0' when disabled by the Configuration bits, BOREN<1:0> (FPOR<1:0>). When the Configuration bits are set to enable SBOREN, the default Reset state will be '1'.

**TABLE 7-1: RESET FLAG BIT OPERATION**

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap Conflict Event	POR
IOPUWR (RCON<14>)	Illegal Opcode or Uninitialized W Register Access	POR
CM (RCON<9>)	Configuration Mismatch Reset	POR
EXTR (RCON<7>)	MCLR Reset	POR
SWR (RCON<6>)	RESET Instruction	POR
WDTO (RCON<4>)	WDT Time-out	PWRSV Instruction, POR
SLEEP (RCON<3>)	PWRSV #SLEEP Instruction	POR
IDLE (RCON<2>)	PWRSV #IDLE Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—

**Note:** All Reset flag bits may be set or cleared by the user software.

### 7.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen, as shown in [Table 7-2](#). If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. For more information, see [Section 9.0 "Oscillator Configuration"](#).

**TABLE 7-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)**

Reset Type	Clock Source Determinant
POR	FNOSC <sub>x</sub> Configuration bits (FNOSC<10:8>)
BOR	
MCLR	COSC <sub>x</sub> Control bits (OSCCON<14:12>)
WDTO	
SWR	

# PIC24F16KL402 FAMILY

## 7.2 Device Reset Times

The Reset times for various types of device Reset are summarized in [Table 7-3](#). Note that the System Reset Signal,  $\overline{\text{SYSRST}}$ , is released after the POR and PWRT delay times expire.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable  $\overline{\text{SYSRST}}$  delay times.

The FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the  $\overline{\text{SYSRST}}$  signal is released.

**TABLE 7-3: RESET DELAY TIMES FOR VARIOUS DEVICE RESETS**

Reset Type	Clock Source	$\overline{\text{SYSRST}}$ Delay	System Clock Delay	Notes
POR <sup>(6)</sup>	EC	TPOR + TPWRT	—	1, 2
	FRC, FRCDIV	TPOR + TPWRT	TFRC	1, 2, 3
	LPRC	TPOR + TPWRT	TLPRC	1, 2, 3
	ECPLL	TPOR + TPWRT	TLOCK	1, 2, 4
	FRCPLL	TPOR + TPWRT	TFRC + TLOCK	1, 2, 3, 4
	XT, HS, SOSC	TPOR + TPWRT	TOST	1, 2, 5
	XTPLL, HSPLL	TPOR + TPWRT	TOST + TLOCK	1, 2, 4, 5
BOR	EC	TPWRT	—	2
	FRC, FRCDIV	TPWRT	TFRC	2, 3
	LPRC	TPWRT	TLPRC	2, 3
	ECPLL	TPWRT	TLOCK	2, 4
	FRCPLL	TPWRT	TFRC + TLOCK	2, 3, 4
	XT, HS, SOSC	TPWRT	TOST	2, 5
	XTPLL, HSPLL	TPWRT	TFRC + TLOCK	2, 3, 4
All Others	Any Clock	—	—	None

**Note 1:** TPOR = Power-on Reset delay.

**2:** TPWRT = 64 ms nominal if the Power-up Timer is enabled; otherwise, it is zero.

**3:** TFRC and TLPRC = RC oscillator start-up times.

**4:** TLOCK = PLL lock time.

**5:** TOST = Oscillator Start-up Timer (OST). A 10-bit counter waits 1024 oscillator periods before releasing the oscillator clock to the system.

**6:** If Two-Speed Start-up is enabled, regardless of the primary oscillator selected, the device starts with FRC, and in such cases, FRC start-up time is valid.

**Note:** For detailed operating frequency and timing specifications, see [Section 26.0 “Electrical Characteristics”](#).

# PIC24F16KL402 FAMILY

## 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  $\overline{\text{SYSRST}}$  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  $\overline{\text{SYSRST}}$  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 RCFGAL 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 VDD below the set threshold point will reset the device. The chip will remain in BOR until VDD rises above the threshold.

If the Power-up Timer is enabled, it will be invoked after VDD rises above the threshold. Then, it will keep the chip in Reset for an additional time delay, TPWRT, if VDD 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 VDD 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.

# PIC24F16KL402 FAMILY

---

## 7.4.2 DETECTING BOR

When BOR is enabled, the BOR bit (RCON<1>) is always reset to '1' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR and BOR bits are reset to '0' in the software, immediately after any POR event. If the BOR bit is '1' while POR is '0', it can be reliably assumed that a BOR event has occurred.

**Note:** Even when the device exits from Deep Sleep mode, both the POR and BOR are set.

## 7.4.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, BOR remains under hardware control and operates as previously described. However, whenever the device enters Sleep mode, BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

## 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.

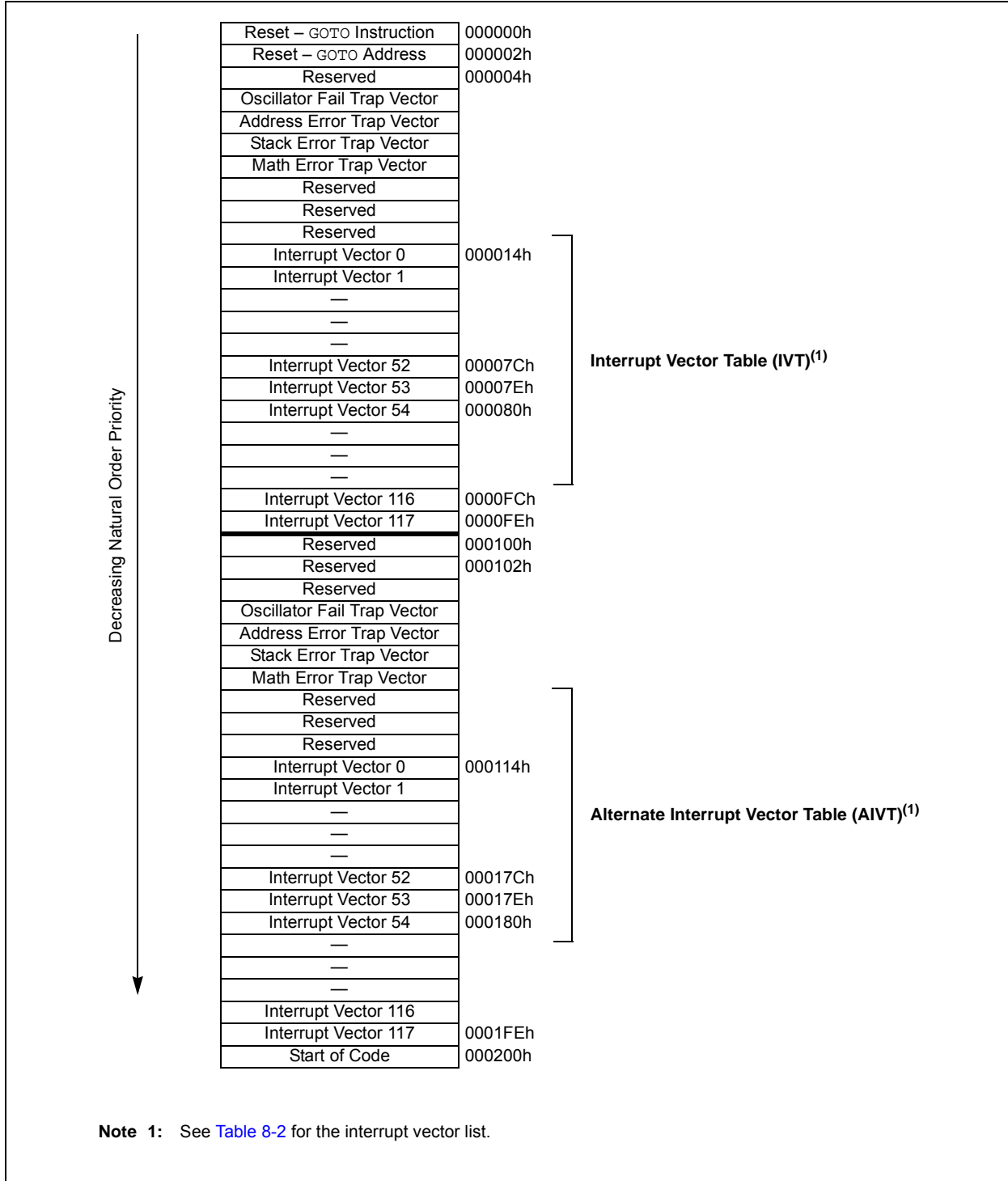
## 8.2 Reset Sequence

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.

# PIC24F16KL402 FAMILY

**FIGURE 8-1: PIC24F INTERRUPT VECTOR TABLE**



# PIC24F16KL402 FAMILY

**TABLE 8-1: TRAP VECTOR DETAILS**

Vector Number	IVT Address	AIVT Address	Trap Source
0	000004h	000104h	Reserved
1	000006h	000106h	Oscillator Failure
2	000008h	000108h	Address Error
3	00000Ah	00010Ah	Stack Error
4	00000Ch	00010Ch	Math Error
5	00000Eh	00010Eh	Reserved
6	000010h	000110h	Reserved
7	000012h	000112h	Reserved

**TABLE 8-2: IMPLEMENTED INTERRUPT VECTORS**

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
MSSP1 Bus Collision Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
MSSP1 SPI or I <sup>2</sup> C™ Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
MSSP2 Bus Collision Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
MSSP2 SPI or I <sup>2</sup> C Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
Input Change Notification	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
HLVD (High/Low-Voltage Detect)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC17<2:0>
NVM (NVM Write Complete)	15	000032h	000132h	IFS0<15>	IEC0<15>	IPC3<14:12>
CCP1/ECCP1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
CCP2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
CCP3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer3 Gate External Count	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
ULPW (Ultra Low-Power Wake-up)	80	0000B4h	0001B4h	IFS5<0>	IEC5<0>	IPC20<2:0>

# PIC24F16KL402 FAMILY

---

## 8.3 Interrupt Control and Status Registers

Depending on the particular device, the PIC24F16KL402 family of devices implements up to 28 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS5
- IEC0 through IEC5
- IPC0 through IPC7, ICP9, IPC12, ICP16, ICP18 and IPC20
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the AIV table.

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit, which is set by the respective peripherals or external signal, and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the Interrupt Priority Level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The INTTREG register contains the associated interrupt vector number and the new CPU Interrupt Priority Level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new Interrupt Priority Level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the same sequence listed in [Table 8-2](#). For example, the INTO (External Interrupt 0) is depicted as having a vector number and a natural order priority of 0. The INTOIF status bit is found in IFS0<0>, the INTOIE enable bit in IEC0<0> and the INTOIP<2:0> priority bits are in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU control registers contain bits that control interrupt functionality. The ALU STATUS Register (SR) contains the IPL<2:0> bits (SR<7:5>). These indicate the current CPU Interrupt Priority Level. The user may change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which together with the IPL<2:0> bits, also indicates the current CPU priority level. IPL3 is a read-only bit so that the trap events cannot be masked by the user's software.

All interrupt registers are described in [Register 8-3](#) through [Register 8-30](#), in the following sections.

# PIC24F16KL402 FAMILY

## REGISTER 8-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC <sup>(1)</sup>
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2,3)</sup>	IPL1 <sup>(2,3)</sup>	IPL0 <sup>(2,3)</sup>	RA <sup>(1)</sup>	N <sup>(1)</sup>	OV <sup>(1)</sup>	Z <sup>(1)</sup>	C <sup>(1)</sup>
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 7-5                      **IPL<2:0>:** CPU Interrupt Priority Level Status bits<sup>(2,3)</sup>

- 111 = CPU Interrupt Priority Level is 7 (15); user interrupts disabled
- 110 = CPU Interrupt Priority Level is 6 (14)
- 101 = CPU Interrupt Priority Level is 5 (13)
- 100 = CPU Interrupt Priority Level is 4 (12)
- 011 = CPU Interrupt Priority Level is 3 (11)
- 010 = CPU Interrupt Priority Level is 2 (10)
- 001 = CPU Interrupt Priority Level is 1 (9)
- 000 = CPU Interrupt Priority Level is 0 (8)

- Note 1:** See [Register 3-1](#) for the description of these bits, which are not dedicated to interrupt control functions.
- 2:** The IPL bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the Interrupt Priority Level if IPL3 = 1.
- 3:** The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

**Note:** Bit 8 and bits 4 through 0 are described in [Section 3.0 "CPU"](#).

# PIC24F16KL402 FAMILY

## REGISTER 8-2: CORCON: CPU CONTROL REGISTER

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	R/C-0	R/W-0	U-0	U-0
—	—	—	—	IPL3 <sup>(2)</sup>	PSV <sup>(1)</sup>	—	—
bit 7						bit 0	

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

- bit 15-4     **Unimplemented:** Read as '0'
- bit 3       **IPL3:** CPU Interrupt Priority Level Status bit<sup>(2)</sup>
  - 1 = CPU Interrupt Priority Level is greater than 7
  - 0 = CPU Interrupt Priority Level is 7 or less
- bit 1-0     **Unimplemented:** Read as '0'

- Note 1:** See [Register 3-2](#) for the description of this bit, which is not dedicated to interrupt control functions.
- Note 2:** The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

**Note:** Bit 2 is described in [Section 3.0 "CPU"](#).

# PIC24F16KL402 FAMILY

## REGISTER 8-3: INTCON1: INTERRUPT CONTROL REGISTER 1

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
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            **NSTDIS:** Interrupt Nesting Disable bit  
                   1 = Interrupt nesting is disabled  
                   0 = Interrupt nesting is enabled
- bit 14-5        **Unimplemented:** Read as '0'
- bit 4            **MATHERR:** Arithmetic Error Trap Status bit  
                   1 = Overflow trap has occurred  
                   0 = Overflow trap has not occurred
- bit 3            **ADDRERR:** Address Error Trap Status bit  
                   1 = Address error trap has occurred  
                   0 = Address error trap has not occurred
- bit 2            **STKERR:** Stack Error Trap Status bit  
                   1 = Stack error trap has occurred  
                   0 = Stack error trap has not occurred
- bit 1            **OSCFAIL:** Oscillator Failure Trap Status bit  
                   1 = Oscillator failure trap has occurred  
                   0 = Oscillator failure trap has not occurred
- bit 0            **Unimplemented:** Read as '0'

# PIC24F16KL402 FAMILY

## REGISTER 8-4: INTCON2: INTERRUPT CONTROL REGISTER2

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7						bit 0	

<b>Legend:</b>	HSC = Hardware Settable/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      **ALTIVT:** Enable Alternate Interrupt Vector Table bit  
1 = Uses Alternate Interrupt Vector Table  
0 = Uses standard (default) vector table
- bit 14      **DISI:** DISI Instruction Status bit  
1 = DISI instruction is active  
0 = DISI instruction is not active
- bit 13-3    **Unimplemented:** Read as '0'
- bit 2        **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit  
1 = Interrupt on negative edge  
0 = Interrupt on positive edge
- bit 1        **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit  
1 = Interrupt on negative edge  
0 = Interrupt on positive edge
- bit 0        **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit  
1 = Interrupt on negative edge  
0 = Interrupt on positive edge

# PIC24F16KL402 FAMILY

## REGISTER 8-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
NVMIF	—	AD1IF	U1TXIF	U1RXIF	—	—	T3IF
bit 15							bit 8

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0
T2IF	CCP2IF	—	—	T1IF	CCP1IF	—	INT0IF
bit 7							bit 0

<b>Legend:</b>			
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      **NVMIF:** NVM Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **AD1IF:** A/D Conversion Complete Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 12      **U1TXIF:** UART1 Transmitter Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 11      **U1RXIF:** UART1 Receiver Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 10-9    **Unimplemented:** Read as '0'
- bit 8        **T3IF:** Timer3 Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 7        **T2IF:** Timer2 Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 6        **CCP2IF:** Capture/Compare/PWM2 Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 5-4     **Unimplemented:** Read as '0'
- bit 3        **T1IF:** Timer1 Interrupt Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 2        **CCP1IF:** Capture/Compare/PWM1 Interrupt Flag Status bit (ECCP1 on PIC24FXXKL40X devices)  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred
- bit 1        **Unimplemented:** Read as '0'
- bit 0        **INT0IF:** External Interrupt 0 Flag Status bit  
1 = Interrupt request has occurred  
0 = Interrupt request has not occurred

# PIC24F16KL402 FAMILY

## REGISTER 8-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0
U2TXIF <sup>(1)</sup>	U2RXIF <sup>(1)</sup>	INT2IF	—	T4IF <sup>(1)</sup>	—	CCP3IF <sup>(1)</sup>	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IF	CNIF	CMIF	BCL1IF	SSP1IF
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      **U2TXIF:** UART2 Transmitter Interrupt Flag Status bit<sup>(1)</sup>  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 14      **U2RXIF:** UART2 Receiver Interrupt Flag Status bit<sup>(1)</sup>  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 13      **INT2IF:** External Interrupt 2 Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 12      **Unimplemented:** Read as '0'
- bit 11      **T4IF:** Timer4 Interrupt Flag Status bit<sup>(1)</sup>  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 10      **Unimplemented:** Read as '0'
- bit 9        **CCP3IF:** Capture/Compare/PWM3 Interrupt Flag Status bit<sup>(1)</sup>  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 8-5     **Unimplemented:** Read as '0'
- bit 4        **INT1IF:** External Interrupt 1 Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 3        **CNIF:** Input Change Notification Interrupt Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 2        **CMIF:** Comparator Interrupt Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 1        **BCL1IF:** MSSP1 I<sup>2</sup>C™ Bus Collision Interrupt Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred
- bit 0        **SSP1IF:** MSSP1 SPI/I<sup>2</sup>C Event Interrupt Flag Status bit  
             1 = Interrupt request has occurred  
             0 = Interrupt request has not occurred

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

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

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	T3GIF	—	—	—	—	—
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 15-6      **Unimplemented:** Read as '0'
- bit 5         **T3GIF:** Timer3 External Gate Interrupt Flag Status bit  
                  1 = Interrupt request has occurred  
                  0 = Interrupt request has not occurred
- bit 4-0      **Unimplemented:** Read as '0'

## REGISTER 8-8: IFS3: INTERRUPT FLAG STATUS REGISTER 3

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	R/W-0	R/W-0	U-0
—	—	—	—	—	BCL2IF <sup>(1)</sup>	SSP2IF <sup>(1)</sup>	—
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 15-3     **Unimplemented:** Read as '0'
- bit 2         **BCL2IF:** MSSP2 I<sup>2</sup>C™ Bus Collision Interrupt Flag Status bit<sup>(1)</sup>  
                  1 = Interrupt request has occurred  
                  0 = Interrupt request has not occurred
- bit 1         **SSP2IF:** MSSP2 SPI/I<sup>2</sup>C Event Interrupt Flag Status bit<sup>(1)</sup>  
                  1 = Interrupt request has occurred  
                  0 = Interrupt request has not occurred
- bit 0         **Unimplemented:** Read as '0'

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## 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 <sup>(1)</sup>	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<sup>(1)</sup>  
             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

# PIC24F16KL402 FAMILY

## REGISTER 8-11: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
NVMIE	—	AD1IE	U1TXIE	U1RXIE	—	—	T3IE
bit 15							bit 8

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0
T2IE	CCP2IE	—	—	T1IE	CCP1IE	—	INT0IE
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      **NVMIE:** NVM Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **AD1IE:** A/D Conversion Complete Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 12      **U1TXIE:** UART1 Transmitter Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 11      **U1RXIE:** UART1 Receiver Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 10-9    **Unimplemented:** Read as '0'
- bit 8        **T3IE:** Timer3 Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 7        **T2IE:** Timer2 Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 6        **CCP2IE:** Capture/Compare/PWM2 Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 5-4     **Unimplemented:** Read as '0'
- bit 3        **T1IE:** Timer1 Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 2        **CCP1IE:** Capture/Compare/PWM1 Interrupt Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled
- bit 1        **Unimplemented:** Read as '0'
- bit 0        **INT0IE:** External Interrupt 0 Enable bit  
1 = Interrupt request is enabled  
0 = Interrupt request is not enabled

# PIC24F16KL402 FAMILY

## REGISTER 8-12: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0
U2TXIE <sup>(1)</sup>	U2RXIE <sup>(1)</sup>	INT2IE	—	T4IE <sup>(1)</sup>	—	CCP3IE <sup>(1)</sup>	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT1IE	CNIE	CMIE	BCL1IE	SSP1IE
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        **U2TXIE:** UART2 Transmitter Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 14        **U2RXIE:** UART2 Receiver Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 13        **INT2IE:** External Interrupt 2 Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 12        **Unimplemented:** Read as '0'
- bit 11        **T4IE:** Timer4 Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 10        **Unimplemented:** Read as '0'
- bit 9         **CCP3IE:** Capture/Compare/PWM3 Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 8-5       **Unimplemented:** Read as '0'
- bit 4         **INT1IE:** External Interrupt 1 Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 3         **CNIE:** Input Change Notification Interrupt Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 2         **CMIE:** Comparator Interrupt Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 1         **BCL1IE:** MSSP1 I<sup>2</sup>C™ Bus Collision Interrupt Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 0         **SSP1IE:** MSSP1 SPI/I<sup>2</sup>C Event Interrupt Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-13: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

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

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	T3GIE	—	—	—	—	—
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 15-6      **Unimplemented:** Read as '0'
- bit 5        **T3GIF:** Timer3 External Gate Interrupt Enable bit  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 4-0      **Unimplemented:** Read as '0'

## REGISTER 8-14: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

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	R/W-0	R/W-0	U-0
—	—	—	—	—	BCL2IE <sup>(1)</sup>	SSP2IE <sup>(1)</sup>	—
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 15-3     **Unimplemented:** Read as '0'
- bit 2        **BCL2IE:** MSSP2 I<sup>2</sup>C™ Bus Collision Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 1        **SSP2IF:** MSSP2 SPI/I<sup>2</sup>C Event Interrupt Enable bit<sup>(1)</sup>  
                  1 = Interrupt request is enabled  
                  0 = Interrupt request is not enabled
- bit 0        **Unimplemented:** Read as '0'

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## 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 <sup>(1)</sup>	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<sup>(1)</sup>  
                   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

# PIC24F16KL402 FAMILY

## 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

# PIC24F16KL402 FAMILY

## REGISTER 8-18: IPC1: INTERRUPT PRIORITY CONTROL REGISTER 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	CCP2IP2	CCP2IP1	CCP2IP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

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

bit 14-12 **T2IP<2:0>:** Timer2 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 **CCP2IP<2:0>:** Capture/Compare/PWM2 Interrupt Priority bits

111 = Interrupt is Priority 7 (highest priority interrupt)

- 
- 
- 

001 = Interrupt is Priority 1

000 = Interrupt source is disabled

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

# PIC24F16KL402 FAMILY

## REGISTER 8-19: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	T3IP2	T3IP1	T3IP0
bit 7						bit 0	

<b>Legend:</b>			
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    **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 11-3     **Unimplemented:** Read as '0'
- bit 2-0      **T3IP<2:0>:** Timer3 Interrupt Priority bits
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled

# PIC24F16KL402 FAMILY

## 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

# PIC24F16KL402 FAMILY

## REGISTER 8-21: IPC4: INTERRUPT PRIORITY CONTROL REGISTER 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
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
—	BCL1IP2	BCL1IP1	BCL1IP0	—	SSP1IP2	SSP1IP1	SSP1IP0
bit 7				bit 0			

<b>Legend:</b>			
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    **CNIP<2:0>:** Input Change Notification 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    **CMIP<2:0>:** Comparator Interrupt Priority bits
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 7      **Unimplemented:** Read as '0'
- bit 6-4    **BCL1IP<2:0>:** MSSP1 I<sup>2</sup>C™ Bus Collision 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    **SSP1IP<2:0>:** MSSP1 SPI/I<sup>2</sup>C Event Interrupt Priority bits
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled

# PIC24F16KL402 FAMILY

## REGISTER 8-22: IPC5: INTERRUPT PRIORITY 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	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
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 15-3

**Unimplemented:** Read as '0'

bit 2-0

**INT1IP<2:0>:** External Interrupt 1 Priority bits

111 = Interrupt is Priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is Priority 1

000 = Interrupt source is disabled

# PIC24F16KL402 FAMILY

## REGISTER 8-23: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	T4IP2 <sup>(1)</sup>	T4IP1 <sup>(1)</sup>	T4IP0 <sup>(1)</sup>	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CCP3IP2 <sup>(1)</sup>	CCP3IP1 <sup>(1)</sup>	CCP3IP0 <sup>(1)</sup>	—	—	—	—
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 **T4IP<2:0>:** Timer4 Interrupt Priority bits<sup>(1)</sup>

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 **CCP3IP:** Capture/Compare/PWM3 Interrupt Priority bits<sup>(1)</sup>

111 = Interrupt is Priority 7 (highest priority interrupt)

•  
•  
•

001 = Interrupt is Priority 1

000 = Interrupt source is disabled

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

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-24: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2 <sup>(1)</sup>	U2TXIP1 <sup>(1)</sup>	U2TXIP0 <sup>(1)</sup>	—	U2RXIP2 <sup>(1)</sup>	U2RXIP1 <sup>(1)</sup>	U2RXIP0 <sup>(1)</sup>
bit 15				bit 8			

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT2IP2	INT2IP1	INT2IP0	—	—	—	—
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    **U2TXIP<2:0>:** UART2 Transmitter Interrupt Priority bits<sup>(1)</sup>  
 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    **U2RXIP<2:0>:** UART2 Receiver Interrupt Priority bits<sup>(1)</sup>  
 111 = Interrupt is Priority 7 (highest priority interrupt)  
 •  
 •  
 •  
 001 = Interrupt is Priority 1  
 000 = Interrupt source is disabled
- bit 7       **Unimplemented:** Read as '0'
- bit 6-4     **INT2IP<2:0>:** External Interrupt 2 Priority bits  
 111 = Interrupt is Priority 7 (highest priority interrupt)  
 •  
 •  
 •  
 001 = Interrupt is Priority 1  
 000 = Interrupt source is disabled
- bit 3-0     **Unimplemented:** Read as '0'

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-25: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	T3GIP2	T3GIP1	T3GIP0	—	—	—	—
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 6-4      **T3GIP<2:0>:** Timer3 External Gate Interrupt Priority bits

111 = Interrupt is Priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is Priority 1

000 = Interrupt source is disabled

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

# PIC24F16KL402 FAMILY

## REGISTER 8-26: IPC12: INTERRUPT PRIORITY CONTROL REGISTER 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	BCL2IP2 <sup>(1)</sup>	BCL2IP1 <sup>(1)</sup>	BCL2IP0 <sup>(1)</sup>
bit 15						bit 8	

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SSP2IP2 <sup>(1)</sup>	SSP2IP1 <sup>(1)</sup>	SSP2IP0 <sup>(1)</sup>	—	—	—	—
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 15-11      **Unimplemented:** Read as '0'
- bit 10-8      **BCL2IP<2:0>:** MSSP2 I<sup>2</sup>C™ Bus Collision Interrupt Priority bits<sup>(1)</sup>
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 7          **Unimplemented:** Read as '0'
- bit 6-4      **SSP2IP<2:0>:** MSSP2 SPI/I<sup>2</sup>C Event Interrupt Priority bits<sup>(1)</sup>
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 3-0      **Unimplemented:** Read as '0'

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-27: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	U2ERIP2 <sup>(1)</sup>	U2ERIP1 <sup>(1)</sup>	U2ERIP0 <sup>(1)</sup>
bit 15						bit 8	

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2 <sup>(1)</sup>	U1ERIP1 <sup>(1)</sup>	U1ERIP0 <sup>(1)</sup>	—	—	—	—
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 15-11     **Unimplemented:** Read as '0'
- bit 10-8     **U2ERIP<2:0>:** UART2 Error Interrupt Priority bits<sup>(1)</sup>
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 7         **Unimplemented:** Read as '0'
- bit 6-4     **U1ERIP<2:0>:** UART1 Error Interrupt Priority bits<sup>(1)</sup>
  - 111 = Interrupt is Priority 7 (highest priority interrupt)
  - 
  - 
  - 
  - 001 = Interrupt is Priority 1
  - 000 = Interrupt source is disabled
- bit 3-0     **Unimplemented:** Read as '0'

**Note 1:** These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

# PIC24F16KL402 FAMILY

## REGISTER 8-28: IPC18: INTERRUPT PRIORITY CONTROL REGISTER 18

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	R/W-1	R/W-0	R/W-0
—	—	—	—	—	HLVDIP2	HLVDIP1	HLVDIP0
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 15-3            **Unimplemented:** Read as '0'  
 bit 2-0            **HLVDIP<2:0>:** High/Low-Voltage Detect Interrupt Priority bits  
                     111 = Interrupt is Priority 7 (highest priority interrupt)  
                     •  
                     •  
                     •  
                     001 = Interrupt is Priority 1  
                     000 = Interrupt source is disabled

## REGISTER 8-29: IPC20: INTERRUPT PRIORITY CONTROL REGISTER 20

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	R/W-1	R/W-0	R/W-0
—	—	—	—	—	ULPWUIP2	ULPWUIP1	ULPWUIP0
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 15-3            **Unimplemented:** Read as '0'  
 bit 6-4            **ULPWUIP<2:0>:** Ultra Low-Power Wake-up Interrupt Priority bits  
                     111 = Interrupt is Priority 7 (highest priority interrupt)  
                     •  
                     •  
                     •  
                     001 = Interrupt is Priority 1  
                     000 = Interrupt source is disabled

# PIC24F16KL402 FAMILY

## REGISTER 8-30: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER

R-0	r-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	r	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

<b>Legend:</b>	r = Reserved 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      **CPUIRQ:** Interrupt Request from Interrupt Controller CPU bit  
 1 = An interrupt request has occurred but has not yet been Acknowledged by the CPU (this will happen when the CPU priority is higher than the interrupt priority)  
 0 = No interrupt request is left unacknowledged
- bit 14      **Reserved:** Maintain as '0'
- bit 13      **VHOLD:** Vector Hold bit  
Allows Vector Number Capture and Changes What Interrupt is Stored in the VECNUM bit:  
 1 = VECNUM<6:0> will contain the value of the highest priority pending interrupt, instead of the current interrupt  
 0 = VECNUM<6:0> will contain the value of the last Acknowledged interrupt (last interrupt that has occurred with higher priority than the CPU, even if other interrupts are pending)
- bit 12      **Unimplemented:** Read as '0'
- bit 11-8    **ILR<3:0>:** New CPU Interrupt Priority Level bits  
 1111 = CPU Interrupt Priority Level is 15  
 •  
 •  
 •  
 0001 = CPU Interrupt Priority Level is 1  
 0000 = CPU Interrupt Priority Level is 0
- bit 7        **Unimplemented:** Read as '0'
- bit 6-0     **VECNUM<6:0>:** Vector Number of Pending Interrupt bits  
 0111111 = Interrupt vector pending is Number 135  
 •  
 •  
 •  
 0000001 = Interrupt vector pending is Number 9  
 0000000 = Interrupt vector pending is Number 8

# PIC24F16KL402 FAMILY

---

## 8.4 Interrupt Setup Procedures

### 8.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS Control bit (INTCON1<15>) if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and the type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits, for all enabled interrupt sources, may be programmed to the same non-zero value.

**Note:** At a device Reset, the IPCx registers are initialized, such that all user interrupt sources are assigned to Priority Level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

### 8.4.2 INTERRUPT SERVICE ROUTINE

The method that is used to declare an ISR and initialize the IVT with the correct vector address depends on the programming language (i.e., C or assembler) and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles. Otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

### 8.4.3 TRAP SERVICE ROUTINE (TSR)

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

### 8.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to Priority Level 7 by inclusive ORing the value, `OEH`, with `SRL`.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

Only user interrupts with a priority level of 7 or less can be disabled. Trap sources (Levels 8-15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of Priority Levels 1-6 for a fixed period. Level 7 interrupt sources are not disabled by the `DISI` instruction.

# PIC24F16KL402 FAMILY

## 9.0 OSCILLATOR CONFIGURATION

**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 Oscillator Configuration, refer to the “dsPIC33/PIC24 Family Reference Manual”, “Oscillator with 500 kHz Low-Power FRC” (DS39726).

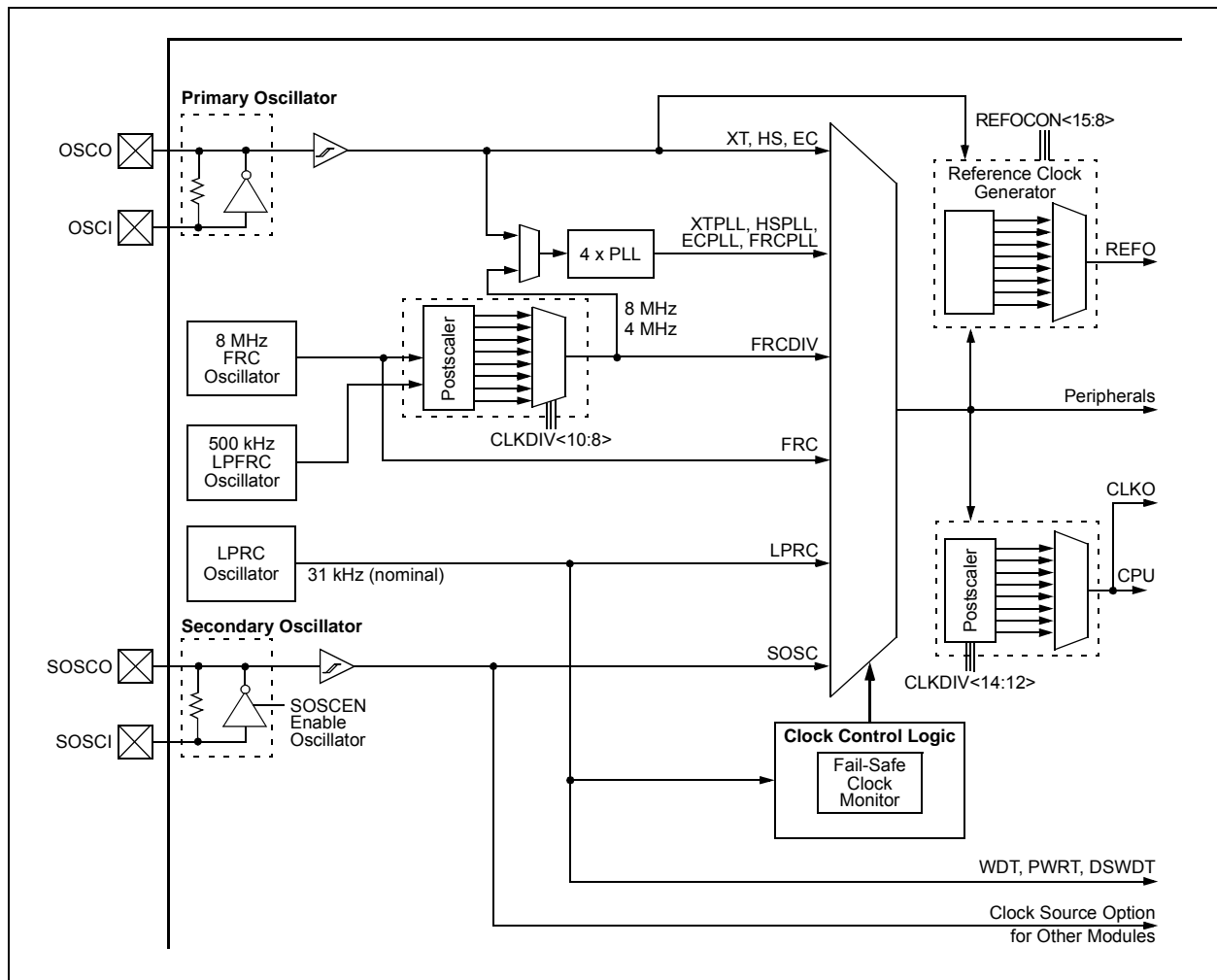
The oscillator system for the PIC24F16KL402 family of devices has the following features:

- A total of five external and internal oscillator options as clock sources, providing 11 different clock modes.
- On-chip, 4x Phase Locked Loop (PLL) to boost internal operating frequency on select internal and external oscillator sources.

- Software-controllable switching between various clock sources.
- Software-controllable postscaler for selective clocking of CPU for system power savings.
- System frequency range declaration bits for EC mode. When using an external clock source, the current consumption is reduced by setting the declaration bits to the expected frequency range.
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown.

A simplified diagram of the oscillator system is shown in Figure 9-1.

**FIGURE 9-1: PIC24F16KL402 FAMILY CLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## 9.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSC1 and OSC0 pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins  
PIC24F16KL402 family devices consist of two types of secondary oscillators:
  - High-Power Secondary Oscillator
  - Low-Power Secondary Oscillator
 These can be selected by using the SOSCSEL (FOSC<5>) bit.
- Fast Internal RC (FRC) Oscillator
  - 8 MHz FRC Oscillator
  - 500 kHz Lower Power FRC Oscillator
- Low-Power Internal RC (LPRC) Oscillator with two modes:
  - High-Power/High-Accuracy mode
  - Low-Power/Low-Accuracy mode

The primary oscillator and 8 MHz FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. The selected clock source generates the processor and peripheral clock sources.

The processor clock source is divided by two to produce the internal instruction cycle clock, Fcy. In this document, the instruction cycle clock is also denoted by Fosc/2. The internal instruction cycle clock, Fosc/2, can be provided on the OSC0 I/O pin for some operating modes of the primary oscillator.

## 9.2 Initial Configuration on POR

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The Oscillator Configuration bit settings are located in the Configuration registers in the program memory (for more information, see [Section 23.2 “Configuration Bits”](#)). The Primary Oscillator Configuration bits, POSCMD<1:0> (FOSC<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (FOSCSEL<2:0>), select the oscillator source that is used at a POR. The FRC Primary Oscillator with Postscaler (FRCDIV) is the default (unprogrammed) selection. The secondary oscillator, or one of the internal oscillators, may be chosen by programming these bit locations. The EC mode Frequency Range Configuration bits, POSCFREQ<1:0> (FOSC<4:3>), optimize power consumption when running in EC mode. The default configuration is “frequency range is greater than 8 MHz”.

The Configuration bits allow users to choose between the various clock modes, shown in [Table 9-1](#).

### 9.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSMx Configuration bits (FOSC<7:6>) are used jointly to configure device clock switching and the FSCM. Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when FCKSM<1:0> are both programmed ('00').

**TABLE 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION**

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Notes
8 MHz FRC Oscillator with Postscaler (FRCDIV)	Internal	11	111	<a href="#">1</a> , <a href="#">2</a>
500 kHz FRC Oscillator with Postscaler (LPFRCDIV)	Internal	11	110	<a href="#">1</a>
Low-Power RC Oscillator (LPRC)	Internal	11	101	<a href="#">1</a>
Secondary (Timer1) Oscillator (SOSC)	Secondary	00	100	<a href="#">1</a>
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (XT)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
8 MHz FRC Oscillator with PLL Module (FRCPLL)	Internal	11	001	<a href="#">1</a>
8 MHz FRC Oscillator (FRC)	Internal	11	000	<a href="#">1</a>

**Note 1:** OSC0 pin function is determined by the OSCIOFNC Configuration bit.

**2:** This is the default oscillator mode for an unprogrammed (erased) device.

# PIC24F16KL402 FAMILY

## 9.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers (SFRs):

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The Clock Divider register (Register 9-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The FRC Oscillator Tune register (Register 9-3) allows the user to fine-tune the FRC oscillator. OSCTUN functionality has been provided to help customers compensate for temperature effects on the FRC frequency over a wide range of temperatures. The tuning step-size is an approximation and is neither characterized nor tested.

### REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0, HSC	R-0, HSC	R-0, HSC	U-0	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15				bit 8			

R/SO-0, HSC	U-0	R-0, HSC <sup>(2)</sup>	U-0	R/CO-0, HS	R/W-0 <sup>(3)</sup>	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN
bit 7				bit 0			

<b>Legend:</b>	HSC = Hardware Settable/Clearable bit		
HS = Hardware Settable bit	CO = Clearable Only bit	SO = Settable Only 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 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits

- 111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = 8 MHz FRC Oscillator (FRC)

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **NOOSC<2:0>:** New Oscillator Selection bits<sup>(1)</sup>

- 111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = 8 MHz FRC Oscillator (FRC)

**Note 1:** Reset values for these bits are determined by the FNOOSC<2:0> Configuration bits.

**2:** Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

**3:** When SOSC is selected to run from a digital clock input rather than an external crystal (SOSCSRC = 0), this bit has no effect.

# PIC24F16KL402 FAMILY

---

## REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

- bit 7      **CLKLOCK:** Clock Selection Lock Enable bit  
    If FSCM is Enabled (FCKSM1 = 1):  
    1 = Clock and PLL selections are locked  
    0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit  
    If FSCM is Disabled (FCKSM1 = 0):  
    Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.
- bit 6      **Unimplemented:** Read as '0'
- bit 5      **LOCK:** PLL Lock Status bit<sup>(2)</sup>  
    1 = PLL module is in lock or the PLL module start-up timer is satisfied  
    0 = PLL module is out of lock, the PLL start-up timer is running or PLL is disabled
- bit 4      **Unimplemented:** Read as '0'
- bit 3      **CF:** Clock Fail Detect bit  
    1 = FSCM has detected a clock failure  
    0 = No clock failure has been detected
- bit 2      **SOSCDRV:** Secondary Oscillator Drive Strength bit<sup>(3)</sup>  
    1 = High-power SOSC circuit is selected  
    0 = Low/high-power select is done via the SOSCSRC Configuration bit
- bit 1      **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit  
    1 = Enables secondary oscillator  
    0 = Disables secondary oscillator
- bit 0      **OSWEN:** Oscillator Switch Enable bit  
    1 = Initiates an oscillator switch to the clock source specified by the NOSC<2:0> bits  
    0 = Oscillator switch is complete

- Note 1:** Reset values for these bits are determined by the FNOSC<2:0> Configuration bits.
- 2:** Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.
- 3:** When SOSC is selected to run from a digital clock input rather than an external crystal (SOSCSRC = 0), this bit has no effect.

# PIC24F16KL402 FAMILY

## REGISTER 9-2: CLKDIV: CLOCK DIVIDER REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN <sup>(1)</sup>	RCDIV2	RCDIV1	RCDIV0
bit 15						bit 8	

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7						bit 0	

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15

**ROI:** Recover on Interrupt bit

1 = Interrupts clear the DOZEN bit, and reset the CPU and peripheral clock ratio to 1:1

0 = Interrupts have no effect on the DOZEN bit

bit 14-12

**DOZE<2:0>:** CPU-to-Peripheral Clock Ratio Select bits

111 = 1:128

110 = 1:64

101 = 1:32

100 = 1:16

011 = 1:8

010 = 1:4

001 = 1:2

000 = 1:1

bit 11

**DOZEN:** DOZE Enable bit<sup>(1)</sup>

1 = DOZE<2:0> bits specify the CPU-to-peripheral clock ratio

0 = CPU and the peripheral clock ratio are set to 1:1

bit 10-8

**RCDIV<2:0>:** FRC Postscaler Select bits

When COSC<2:0> (OSCCON<14:12>) = 111 or 001:

111 = 31.25 kHz (divide-by-256)

110 = 125 kHz (divide-by-64)

101 = 250 kHz (divide-by-32)

100 = 500 kHz (divide-by-16)

011 = 1 MHz (divide-by-8)

010 = 2 MHz (divide-by-4)

001 = 4 MHz (divide-by-2) (default)

000 = 8 MHz (divide-by-1)

When COSC<2:0> (OSCCON<14:12>) = 110:

111 = 1.95 kHz (divide-by-256)

110 = 7.81 kHz (divide-by-64)

101 = 15.62 kHz (divide-by-32)

100 = 31.25 kHz (divide-by-16)

011 = 62.5 kHz (divide-by-8)

010 = 125 kHz (divide-by-4)

001 = 250 kHz (divide-by-2) (default)

000 = 500 kHz (divide-by-1)

bit 7-0

**Unimplemented:** Read as '0'

**Note 1:** This bit is automatically cleared when the ROI bit is set and an interrupt occurs.

# PIC24F16KL402 FAMILY

## REGISTER 9-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5 <sup>(1)</sup>	TUN4 <sup>(1)</sup>	TUN3 <sup>(1)</sup>	TUN2 <sup>(1)</sup>	TUN1 <sup>(1)</sup>	TUN0 <sup>(1)</sup>
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 15-6

**Unimplemented:** Read as '0'

bit 5-0

**TUN<5:0>:** FRC Oscillator Tuning bits<sup>(1)</sup>

011111 = Maximum frequency deviation

011110

•

•

•

000001

000000 = Center frequency, oscillator is running at factory calibrated frequency

111111

•

•

•

100001

100000 = Minimum frequency deviation

**Note 1:** Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

## 9.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

**Note:** The Primary Oscillator mode has three different submodes (XT, HS and EC), which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

### 9.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the FOSC Configuration register must be programmed to '0'. (Refer to [Section 23.0 "Special Features"](#) for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and FSCM function are disabled; this is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled; it is held at '0' at all times.

### 9.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

1. If desired, read the COSCx bits (OSCCON<14:12>) to determine the current oscillator source.
2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
3. Write the appropriate value to the NOSCx bits (OSCCON<10:8>) for the new oscillator source.
4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically, as follows:

1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
3. The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bits value is transferred to the COSCx bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM, with LPRC as a clock source, are enabled) or SOSC (if SOSSEN remains enabled).

**Note 1:** The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.

- 2: Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL mode are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.

# PIC24F16KL402 FAMILY

The following code sequence for a clock switch is recommended:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8>, in two back-to-back instructions.
3. Write the new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0>, in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock-sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in [Example 9-1](#).

## EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV      #OSCCONH, w1
MOV      #0x78, w2
MOV      #0x9A, w3
MOV.b    w2, [w1]
MOV.b    w3, [w1]
;Set new oscillator selection
MOV.b    WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV      #OSCCONL, w1
MOV      #0x46, w2
MOV      #0x57, w3
MOV.b    w2, [w1]
MOV.b    w3, [w1]
;Start oscillator switch operation
BSET     OSCCON,#0
```

## 9.5 Reference Clock Output

In addition to the CLKO output (Fosc/2) available in certain oscillator modes, the device clock in the PIC24F16KL402 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register ([Register 9-4](#)). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSC1 and OSC2, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT). Therefore, if the ROSEL bit is also not set, the oscillator on OSC1 and OSC2 will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

# PIC24F16KL402 FAMILY

## REGISTER 9-4: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

### Legend:

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

- bit 15        **ROEN:** Reference Oscillator Output Enable bit  
                  1 = Reference oscillator is enabled on REFO pin  
                  0 = Reference oscillator is disabled
- bit 14        **Unimplemented:** Read as '0'
- bit 13        **ROSSLP:** Reference Oscillator Output Stop in Sleep bit  
                  1 = Reference oscillator continues to run in Sleep  
                  0 = Reference oscillator is disabled in Sleep
- bit 12        **ROSEL:** Reference Oscillator Source Select bit  
                  1 = Primary oscillator is used as the base clock<sup>(1)</sup>  
                  0 = System clock is used as the base clock; the base clock reflects any clock switching of the device
- bit 11-8     **RODIV<3:0>:** Reference Oscillator Divisor Select bits  
                  1111 = Base clock value divided by 32,768  
                  1110 = Base clock value divided by 16,384  
                  1101 = Base clock value divided by 8,192  
                  1100 = Base clock value divided by 4,096  
                  1011 = Base clock value divided by 2,048  
                  1010 = Base clock value divided by 1,024  
                  1001 = Base clock value divided by 512  
                  1000 = Base clock value divided by 256  
                  0111 = Base clock value divided by 128  
                  0110 = Base clock value divided by 64  
                  0101 = Base clock value divided by 32  
                  0100 = Base clock value divided by 16  
                  0011 = Base clock value divided by 8  
                  0010 = Base clock value divided by 4  
                  0001 = Base clock value divided by 2  
                  0000 = Base clock value
- bit 7-0      **Unimplemented:** Read as '0'

**Note 1:** The crystal oscillator must be enabled using the FOSC<2:0> bits; the crystal maintains the operation in Sleep mode.

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 10.0 POWER-SAVING FEATURES

**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 Power-Saving Features, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**Power-Saving Features with Deep Sleep**” (DS39727).

The PIC24F16KL402 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption using several strategies:

- Clock frequency
- Instruction-based Idle and Sleep modes
- Hardware-based periodic wake-up from Sleep
- Software Controlled Doze mode
- Selective peripheral control in software

Combinations of these methods can be used to selectively tailor an application’s power consumption, while still maintaining critical application features, such as timing-sensitive communications.

## 10.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSCx bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in [Section 9.0 “Oscillator Configuration”](#).

## 10.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special `PWRSV` instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation.

The assembly syntax of the `PWRSV` instruction is shown in [Example 10-1](#).

**Note:** `SLEEP_MODE` and `IDLE_MODE` are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to “wake-up”.

### EXAMPLE 10-1: `PWRSV` INSTRUCTION SYNTAX

```
PWRSV    #SLEEP_MODE    ; Put the device into SLEEP mode
PWRSV    #IDLE_MODE     ; Put the device into IDLE mode
```

# PIC24F16KL402 FAMILY

---

## 10.2.1 SLEEP MODE

Sleep mode includes these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum, provided that no I/O pin is sourcing current.
- The I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if any active module has selected the LPRC as its source, including the WDT, Timer1 and Timer3.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features, or peripherals, may continue to operate in Sleep mode. This includes items, such as the Input Change Notification (ICN) on the I/O ports or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

On wake-up from Sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

## 10.2.2 IDLE MODE

Idle mode has these features:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see [Section 10.5 “Selective Peripheral Module Control”](#)).
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The device will wake from Idle mode on any of these events:

- Any interrupt that is individually enabled
- Any device Reset
- A WDT time-out

On wake-up from Idle, the clock is re-applied to the CPU. Instruction execution begins immediately, starting with the instruction following the `PWRSVAV` instruction or the first instruction in the ISR.

## 10.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a `PWRSVAV` instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

## 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.

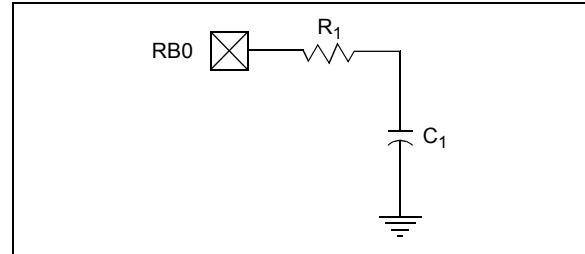
The time-out is dependent on the discharge time of the RC circuit on RB0. When the voltage on RB0 drops below  $V_{IL}$ , the device wakes up and executes the next instruction.

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

```

# PIC24F16KL402 FAMILY

## REGISTER 10-1: ULPWCON: ULPWU CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 15      **ULPEN:** ULPWU Module Enable bit  
             1 = Module is enabled  
             0 = Module is disabled
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **ULPSIDL:** ULPWU Stop in Idle Select bit  
             1 = Discontinues module operation when the device enters Idle mode  
             0 = Continues module operation in Idle mode
- bit 12-9    **Unimplemented:** Read as '0'
- bit 8        **ULPSINK:** ULPWU Current Sink Enable bit  
             1 = Current sink is enabled  
             0 = Current sink is disabled
- bit 7-0     **Unimplemented:** Read as '0'

## 10.4 Doze Mode

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted, synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed, while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default.

It is also possible to use Doze mode to selectively reduce power consumption in event driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption. Meanwhile, the CPU idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

## 10.5 Selective Peripheral Module Control

Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked and thus, consume power. There may be cases where the application needs what these modes do not provide: the allocation of power resources to CPU processing, with minimal power consumption from the peripherals.

PIC24F devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with two control bits:

- The Peripheral Enable bit, generically named, “XXXEN”, located in the module’s main control SFR.
- The Peripheral Module Disable (PMD) bit, generically named, “XXXMD”, located in one of the PMD Control registers.

Both bits have similar functions in enabling or disabling its associated module. Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, the control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect, and read values will be invalid. Many peripheral modules have a corresponding PMD bit.

In contrast, disabling a module by clearing its XXXEN bit, disables its functionality, but leaves its registers available to be read and written to. Power consumption is reduced, but not by as much as when the PMD bits are used.

To achieve more selective power savings, peripheral modules can also be selectively disabled when the device enters Idle mode. This is done through the control bit of the generic name format, “XXXIDL”. By default, all modules that can operate during Idle mode will do so. Using the disable on Idle feature disables the module while in Idle mode, allowing further reduction of power consumption during Idle mode. This enhances power savings for extremely critical power applications.

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 11.0 I/O PORTS

**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 I/O Ports, refer to the “dsPIC33/PIC24 Family Reference Manual”, “I/O Ports with Peripheral Pin Select (PPS)” (DS39711). Note that the PIC24F16KL402 family devices do not support Peripheral Pin Select features.

All of the device pins (except VDD and VSS) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

### 11.1 Parallel I/O (PIO) Ports

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. Figure 11-1 illustrates how ports are shared with other peripherals and the associated I/O pin to which they are connected.

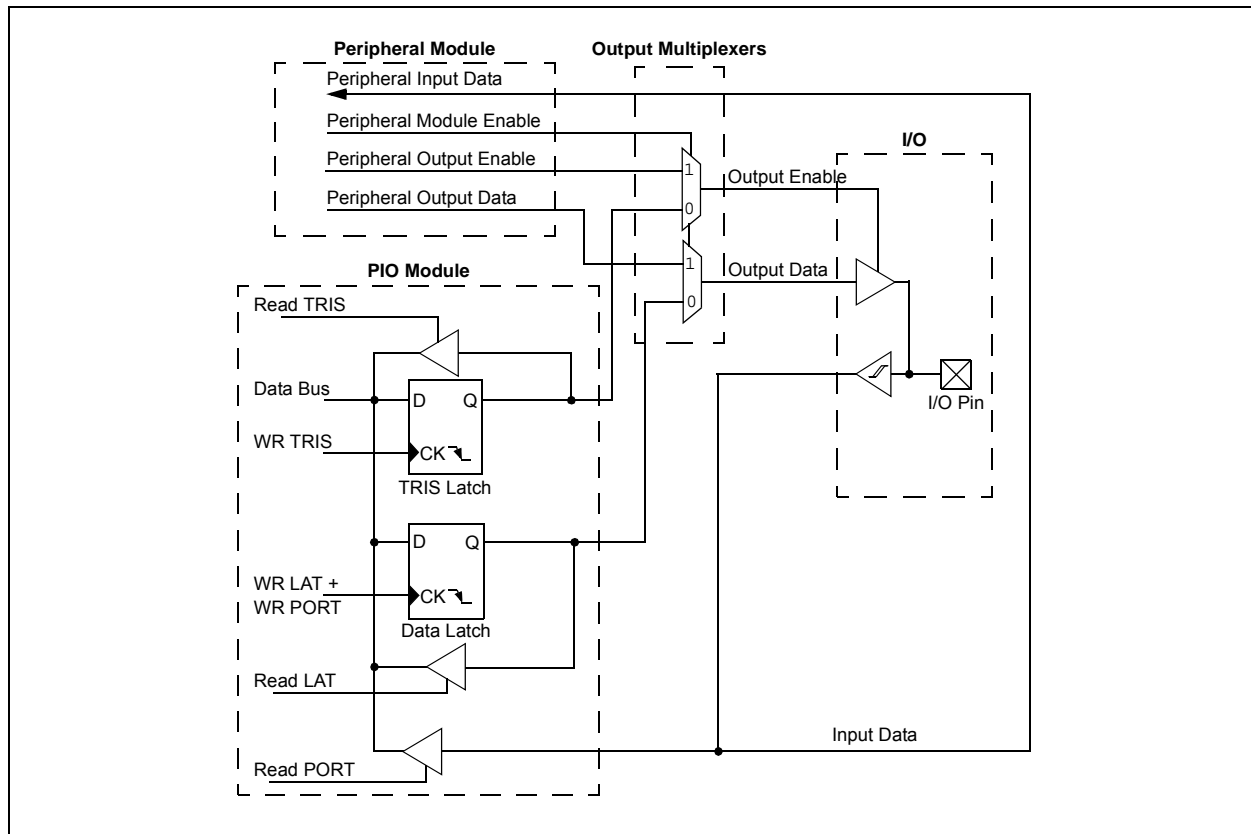
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Data Latch register (LATx), read the latch. Writes to the Data Latch, write the latch. Reads from the port (PORTx), read the port pins, while writes to the port pins, write the latch.

Any bit and its associated data and control registers, that are not valid for a particular device, will be disabled. That means the corresponding LATx and TRISx registers, and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless, regarded as a dedicated port because there is no other competing source of outputs.

**FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED I/O PORT STRUCTURE**



# PIC24F16KL402 FAMILY

---

## 11.1.1 OPEN-DRAIN CONFIGURATION

In addition to the PORTx, LATx and TRISx registers for data control, each port pin can be individually configured for either digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The maximum open-drain voltage allowed is the same as the maximum V<sub>IH</sub> specification.

## 11.1.2 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP.

## 11.2 Configuring Analog Port Pins

The use of the ANSx and TRISx registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRISx bit set (input). If the TRISx bit is cleared (output), the digital output level (V<sub>OH</sub> or V<sub>OL</sub>) will be converted.

When reading the PORTx register, all pins configured as analog input channels will read as cleared (a low level). Analog levels on any pin that is defined as a digital input (including the ANx pins) may cause the input buffer to consume current that exceeds the device specifications.

## 11.2.1 ANALOG SELECTION REGISTER

I/O pins with shared analog functionality, such as A/D inputs and comparator inputs, must have their digital inputs shut off when analog functionality is used. Note that analog functionality includes an analog voltage being applied to the pin externally.

To allow for analog control, the ANSx registers are provided. There is one ANS register for each port (ANSA and ANSB, [Register 11-1](#) and [Register 11-2](#)). Within each ANSx register, there is a bit for each pin that shares analog functionality with the digital I/O functionality. If a particular pin does not have an analog function, that bit is unimplemented.

# PIC24F16KL402 FAMILY

## REGISTER 11-1: ANSA: PORTA ANALOG SELECTION REGISTER

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	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSA3	ANSA2	ANSA1	ANSA0
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 15-4            **Unimplemented:** Read as '0'  
 bit 3-0            **ANSA<3:0>:** Analog Select Control bits  
                     1 = Digital input buffer is not active (use for analog input)  
                     0 = Digital input buffer is active

## REGISTER 11-2: ANSB: PORTB ANALOG SELECTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
ANSB15	ANSB14	ANSB13 <sup>(1)</sup>	ANSB12 <sup>(1)</sup>	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	ANSB4	ANSB3 <sup>(2)</sup>	ANSB2 <sup>(1)</sup>	ANSB1 <sup>(1)</sup>	ANSB0 <sup>(1)</sup>
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 15-12            **ANSB<15:12>:** Analog Select Control bits<sup>(1)</sup>  
                     1 = Digital input buffer is not active (use for analog input)  
                     0 = Digital input buffer is active  
 bit 11-5            **Unimplemented:** Read as '0'  
 bit 4-0            **ANSB<4:0>:** Analog Select Control bits<sup>(2)</sup>  
                     1 = Digital input buffer is not active (use for analog input)  
                     0 = Digital input buffer is active

**Note 1:** ANSB<13:12,2:0> are unimplemented on 14-pin devices.  
**Note 2:** ANSB<3> is unimplemented on 14-pin and 20-pin devices.

# PIC24F16KL402 FAMILY

## 11.3 Input Change Notification

The Input Change Notification (ICN) function of the I/O ports allows the PIC24F16KL402 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 23 external signals that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the Change Notification (CN) module. The CNEN1 and CNEN2 registers contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

On any pin, only the pull-up resistor or the pull-down resistor should be enabled, but not both of them. If the push button or the keypad is connected to VDD, enable the pull-down, or if they are connected to VSS, enable the pull-up resistors. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins.

Setting any of the control bits enables the weak pull-ups for the corresponding pins. The pull-downs are enabled separately, using the CNPD1 and CNPD2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-downs for the corresponding pins.

When the internal pull-up is selected, the pin uses VDD as the pull-up source voltage. When the internal pull-down is selected, the pins are pulled down to VSS by an internal resistor. Make sure that there is no external pull-up source/pull-down sink when the internal pull-ups/pull-downs are enabled.

**Note:** Pull-ups and pull-downs on Change Notification pins should always be disabled whenever the port pin is configured as a digital output.

### EXAMPLE 11-1: PORT WRITE/READ EXAMPLE (ASSEMBLY LANGUAGE)

```
MOV    #0xFF00, W0           ; Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
MOV    W0, TRISB
MOV    #0x00FF, W0           ; Enable PORTB<15:8> digital input buffers
MOV    W0, ANSB
NOP                                ; Delay 1 cycle
BTSS   PORTB, #13           ; Next Instruction
```

### EXAMPLE 11-2: PORT WRITE/READ EXAMPLE (C LANGUAGE)

```
TRISB = 0xFF00;           // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
ANSB = 0x00FF;           // Enable PORTB<15:8> digital input buffers
NOP();                   // Delay 1 cycle
if(PORTBbits.RB13 == 1)  // execute following code if PORTB pin 13 is set.
{
}
}
```

# PIC24F16KL402 FAMILY

## 12.0 TIMER1

**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 Timers, refer to the “dsPIC33/PIC24 Family Reference Manual”, “Timers” (DS39704).

The Timer1 module is a 16-bit timer which can operate as a free-running, interval timer/counter, or serve as the time counter for a software-based Real-Time Clock (RTC). Timer1 is only reset on initial VDD power-on events. This allows the timer to continue operating as an RTC clock source through other types of device Reset.

Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

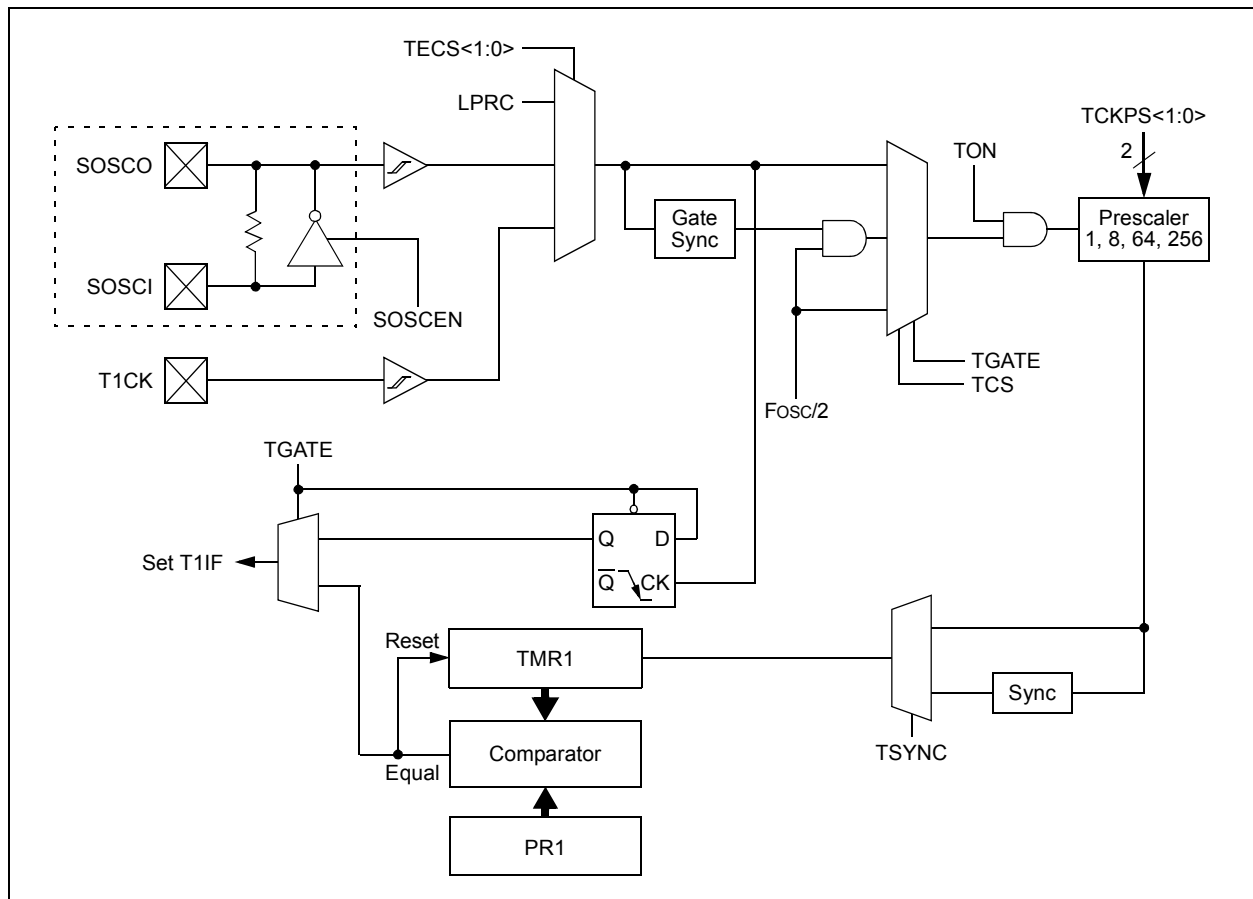
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 12-1 illustrates a block diagram of the 16-bit Timer1 module.

To configure Timer1 for operation:

1. Set the TON bit (= 1).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
5. Load the timer period value into the PR1 register.
6. If interrupts are required, set the Timer1 Interrupt Enable bit, T1IE. Use the Timer1 Interrupt Priority bits, T1IP<2:0>, to set the interrupt priority.

**FIGURE 12-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	T1ECS1 <sup>(1)</sup>	T1ECS0 <sup>(1)</sup>
bit 15						bit 8	

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
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      **TON:** Timer1 On bit  
             1 = Starts 16-bit Timer1  
             0 = Stops 16-bit Timer1
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **TSIDL:** Timer1 Stop in Idle Mode bit  
             1 = Discontinues module operation when device enters Idle mode  
             0 = Continues module operation in Idle mode
- bit 12-10   **Unimplemented:** Read as '0'
- bit 9-8      **T1ECS <1:0>:** Timer1 Extended Clock Select bits<sup>(1)</sup>  
             11 = Reserved; do not use  
             10 = Timer1 uses the LPRC as the clock source  
             01 = Timer1 uses the external clock from T1CK  
             00 = Timer1 uses the Secondary Oscillator (SOSC) as the clock source
- bit 7        **Unimplemented:** Read as '0'
- bit 6        **TGATE:** Timer1 Gated Time Accumulation Enable bit  
             When TCS = 1:  
             This bit is ignored.  
             When TCS = 0:  
             1 = Gated time accumulation is enabled  
             0 = Gated time accumulation is disabled
- bit 5-4      **TCKPS<1:0>:** Timer1 Input Clock Prescale Select bits  
             11 = 1:256  
             10 = 1:64  
             01 = 1:8  
             00 = 1:1
- bit 3        **Unimplemented:** Read as '0'
- bit 2        **TSYNC:** Timer1 External Clock Input Synchronization Select bit  
             When TCS = 1:  
             1 = Synchronizes external clock input  
             0 = Does not synchronize external clock input  
             When TCS = 0:  
             This bit is ignored.
- bit 1        **TCS:** Timer1 Clock Source Select bit  
             1 = Timer1 clock source is selected by T1ECS<1:0>  
             0 = Internal clock (FOSC/2)
- bit 0        **Unimplemented:** Read as '0'

**Note 1:** The T1ECSx bits are valid only when TCS = 1.

# PIC24F16KL402 FAMILY

## 13.0 TIMER2 MODULE

**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 Timers, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**Timers**” (DS39704).

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional Timer3 gate on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP modules

This module is controlled through the T2CON register (Register 13-1), which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

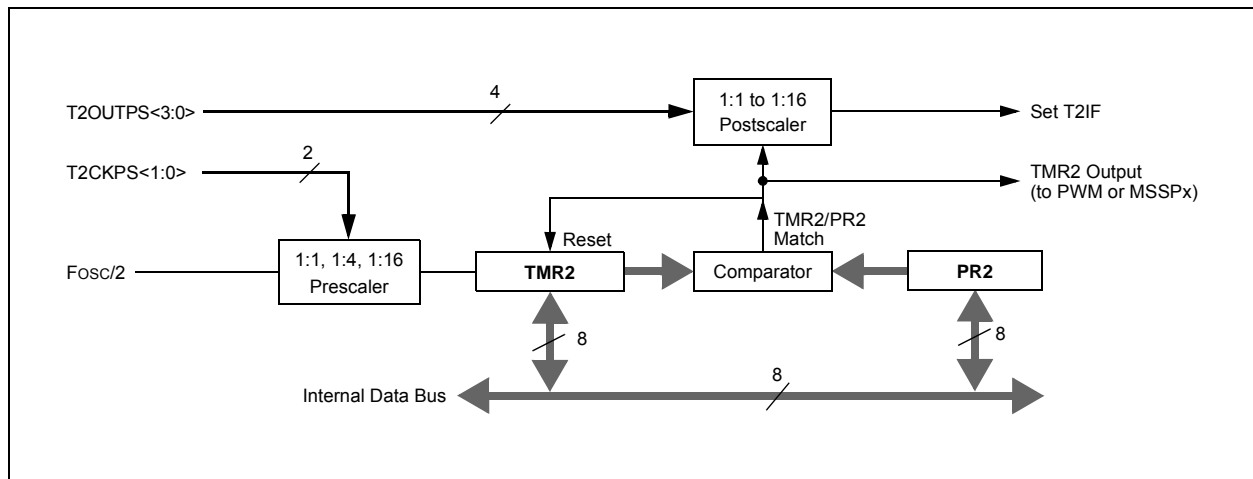
The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset (POR, BOR,  $\overline{\text{MCLR}}$  or WDT Reset)

TMR2 is not cleared when T2CON is written.

A simplified block diagram of the module is shown in Figure 13-1.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



# PIC24F16KL402 FAMILY

## REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 6-3      **T2OUTPS<3:0>:** Timer2 Output Postscale Select bits  
               1111 = 1:16 Postscale  
               1110 = 1:15 Postscale  
               •  
               •  
               •  
               0001 = 1:2 Postscale  
               0000 = 1:1 Postscale
- bit 2         **TMR2ON:** Timer2 On bit  
               1 = Timer2 is on  
               0 = Timer2 is off
- bit 1-0      **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits  
               10 = Prescaler is 16  
               01 = Prescaler is 4  
               00 = Prescaler is 1



# PIC24F16KL402 FAMILY

## REGISTER 14-1: T3CON: TIMER3 CONTROL REGISTER

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	U-0	R/W-0
TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	T3OSCEN	$\overline{T3SYNC}$	—	TMR3ON
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 7-6        **TMR3CS<1:0>:** Timer3 Clock Source Select bits
  - 11 = Low-Power RC Oscillator (LPRC)
  - 10 = External clock source (selected by T3CON<3>)
  - 01 = Instruction clock (FOSC/2)
  - 00 = System clock (Fosc)<sup>(1)</sup>
- bit 5-4        **T3CKPS<1:0>:** Timer3 Input Clock Prescale Select bits
  - 11 = 1:8 Prescale value
  - 10 = 1:4 Prescale value
  - 01 = 1:2 Prescale value
  - 00 = 1:1 Prescale value
- bit 3          **T3OSCEN:** Timer3 Oscillator Enable bit
  - 1 = SOSC (Secondary Oscillator) is used as a clock source
  - 0 = T3CK digital input pin is used as a clock source
- bit 2          **T3SYNC:** Timer3 External Clock Input Synchronization Control bit
  - When TMR3CS<1:0> = 1x:
  - 1 = Does not synchronize the external clock input
  - 0 = Synchronizes the external clock input<sup>(2)</sup>
  - When TMR3CS<1:0> = 0x:
  - This bit is ignored; Timer3 uses the internal clock.
- bit 1          **Unimplemented:** Read as '0'
- bit 0          **TMR3ON:** Timer3 On bit
  - 1 = Enables Timer3
  - 0 = Stops Timer3

- Note 1:** The Fosc clock source should not be selected if the timer will be used with the ECCP capture or compare features.
- 2:** This option must be selected when the timer will be used with ECCP/CCP.

# PIC24F16KL402 FAMILY

## REGISTER 14-2: T3GCON: TIMER3 GATE CONTROL REGISTER<sup>(1)</sup>

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-x	R/W-0	R/W-0
TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/ T3DONE	T3GVAL	T3GSS1	T3GSS0
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      **TMR3GE:** Timer3 Gate Enable bit

If TMR3ON = 0:

This bit is ignored.

If TMR3ON = 1:

1 = Timer counting is controlled by the Timer3 gate function

0 = Timer counts regardless of the Timer3 gate function

bit 6      **T3GPOL:** Timer3 Gate Polarity bit

1 = Timer gate is active-high (Timer3 counts when the gate is high)

0 = Timer gate is active-low (Timer3 counts when the gate is low)

bit 5      **T3GTM:** Timer3 Gate Toggle Mode bit

1 = Timer Gate Toggle mode is enabled.

0 = Timer Gate Toggle mode is disabled and toggle flip-flop is cleared

Timer3 gate flip-flop toggles on every rising edge.

bit 4      **T3GSPM:** Timer3 Gate Single Pulse Mode bit

1 = Timer Gate Single Pulse mode is enabled and is controlling the Timer3 gate

0 = Timer Gate Single Pulse mode is disabled

bit 3      **T3GGO/T3DONE:** Timer3 Gate Single Pulse Acquisition Status bit

1 = Timer gate single pulse acquisition is ready, waiting for an edge

0 = Timer gate single pulse acquisition has completed or has not been started

This bit is automatically cleared when T3GSPM is cleared.

bit 2      **T3GVAL:** Timer3 Gate Current State bit

Indicates the current state of the timer gate that could be provided to the TMR3 register; unaffected by the state of TMR3GE.

bit 1-0      **T3GSS<1:0>:** Timer3 Gate Source Select bits

11 = Comparator 2 output

10 = Comparator 1 output

01 = TMR2 to match PR2 output

00 = T3G input pin

**Note 1:** Initializing T3GCON prior to T3CON is recommended.

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 15.0 TIMER4 MODULE

**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 Timers, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**Timers**” (DS39704).

The Timer4 module is implemented in PIC24FXXKL30X/40X devices only. It has the following features:

- Eight-bit Timer register (TMR4)
- Eight-bit Period register (PR4)
- Readable and writable (all registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR4 match of PR4

The Timer4 module has a control register shown in [Register 15-1](#). Timer4 can be shut off by clearing control bit, TMR4ON (T4CON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4 is controlled by this register.

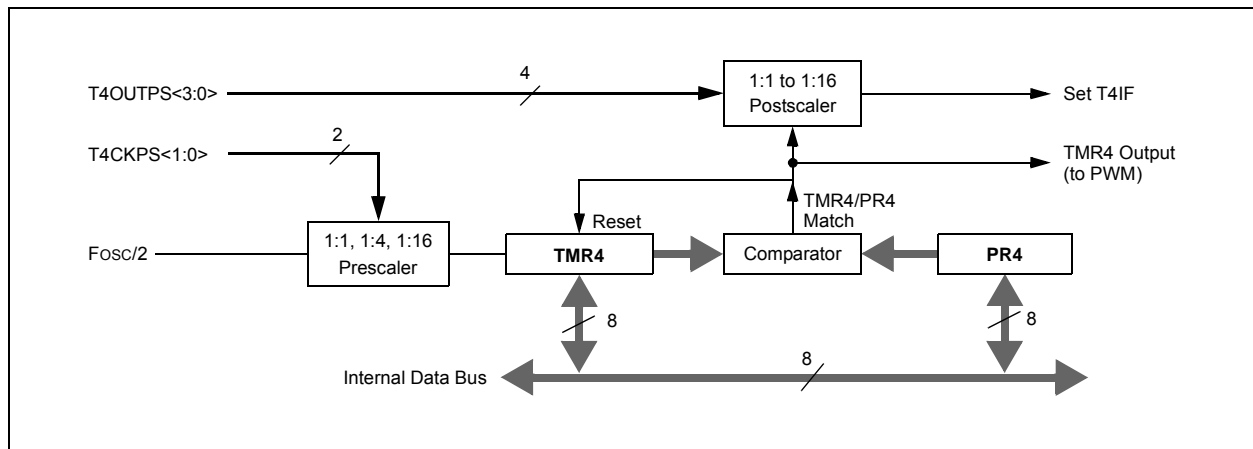
The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR4 register
- A write to the T4CON register
- Any device Reset (POR, BOR,  $\overline{\text{MCLR}}$  or WDT Reset)

TMR4 is not cleared when T4CON is written.

[Figure 15-1](#) is a simplified block diagram of the Timer4 module.

**FIGURE 15-1: TIMER4 BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## REGISTER 15-1: T4CON: TIMER4 CONTROL REGISTER

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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
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 6-3      **T4OUTPS<3:0>:** Timer4 Output Postscale Select bits  
               1111 = 1:16 Postscale  
               1110 = 1:15 Postscale  
               •  
               •  
               •  
               0001 = 1:2 Postscale  
               0000 = 1:1 Postscale
- bit 2         **TMR4ON:** Timer4 On bit  
               1 = Timer4 is on  
               0 = Timer4 is off
- bit 1-0      **T4CKPS<1:0>:** Timer4 Clock Prescale Select bits  
               10 = Prescaler is 16  
               01 = Prescaler is 4  
               00 = Prescaler is 1

# PIC24F16KL402 FAMILY

## 16.0 CAPTURE/COMPARE/PWM (CCP) AND ENHANCED CCP MODULES

**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 Capture/Compare/PWM module, refer to the “dsPIC33/PIC24 Family Reference Manual”.

Depending on the particular device, PIC24F16KL402 family devices include up to three CCP and/or ECCP modules. Key features of all CCP modules include:

- 16-bit input capture for a range of edge events
- 16-bit output compare with multiple output options
- Single-output Pulse-Width Modulation (PWM) with up to 10 bits of resolution
- User-selectable time base from any available timer
- Special Event Trigger on capture and compare events to automatically trigger a range of peripherals

ECCP modules also include these features:

- Operation in Half-Bridge and Full-Bridge (Forward and Reverse) modes
- Pulse steering control across any or all Enhanced PWM pins with user-configurable steering synchronization
- User-configurable external Fault detect with auto-shutdown and auto-restart

PIC24FXXKL40X/30X devices instantiate three CCP modules, one Enhanced (ECCP1) and two standard (CCP2 and CCP3). All other devices instantiate two standard CCP modules (CCP1 and CCP2).

## 16.1 Timer Selection

On all PIC24F16KL402 family devices, the CCP and ECCP modules use Timer3 as the time base for capture and compare operations. PWM and Enhanced PWM operations may use either Timer2 or Timer4. PWM time base selection is done through the CCPTMRS0 register ([Register 16-6](#)).

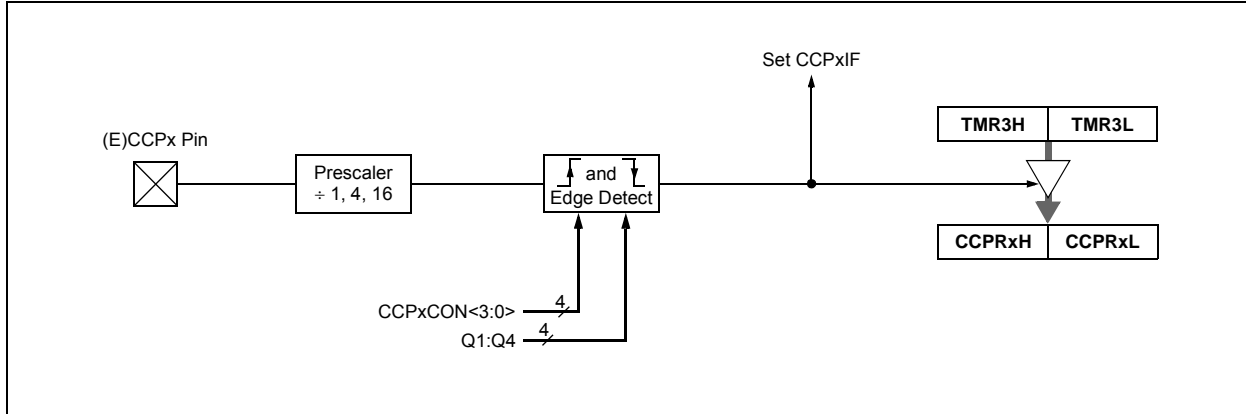
## 16.2 CCP I/O Pins

To configure I/O pins with a CCP function, the proper mode must be selected by setting the CCPxM<3:0> bits.

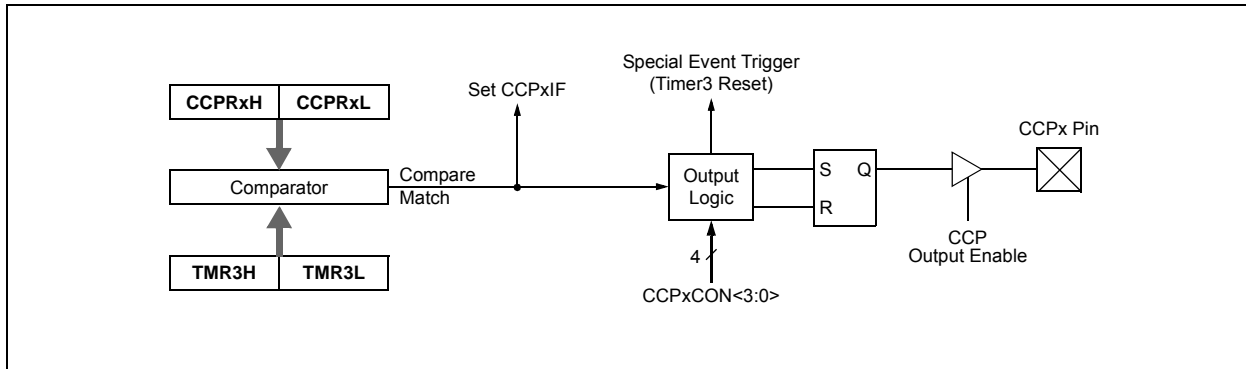
Where the Enhanced CCP module is available, it may have up to four PWM outputs depending on the selected operating mode. These outputs are designated, P1A through P1D. The outputs that are active depend on the ECCP operating mode selected. To configure I/O pins for Enhanced PWM operation, the proper PWM mode must be selected by setting the PM<1:0> and CCPxM<3:0> bits.

# PIC24F16KL402 FAMILY

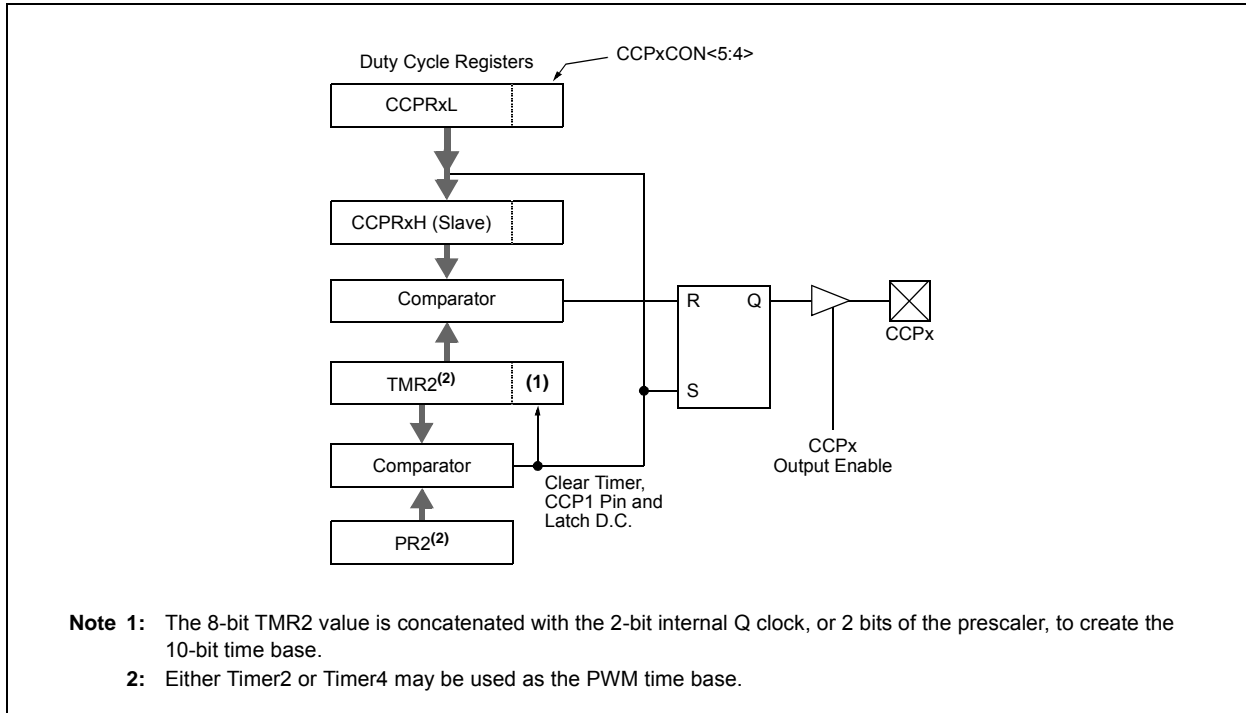
**FIGURE 16-1: GENERIC CAPTURE MODE BLOCK DIAGRAM**



**FIGURE 16-2: GENERIC COMPARE MODE BLOCK DIAGRAM**

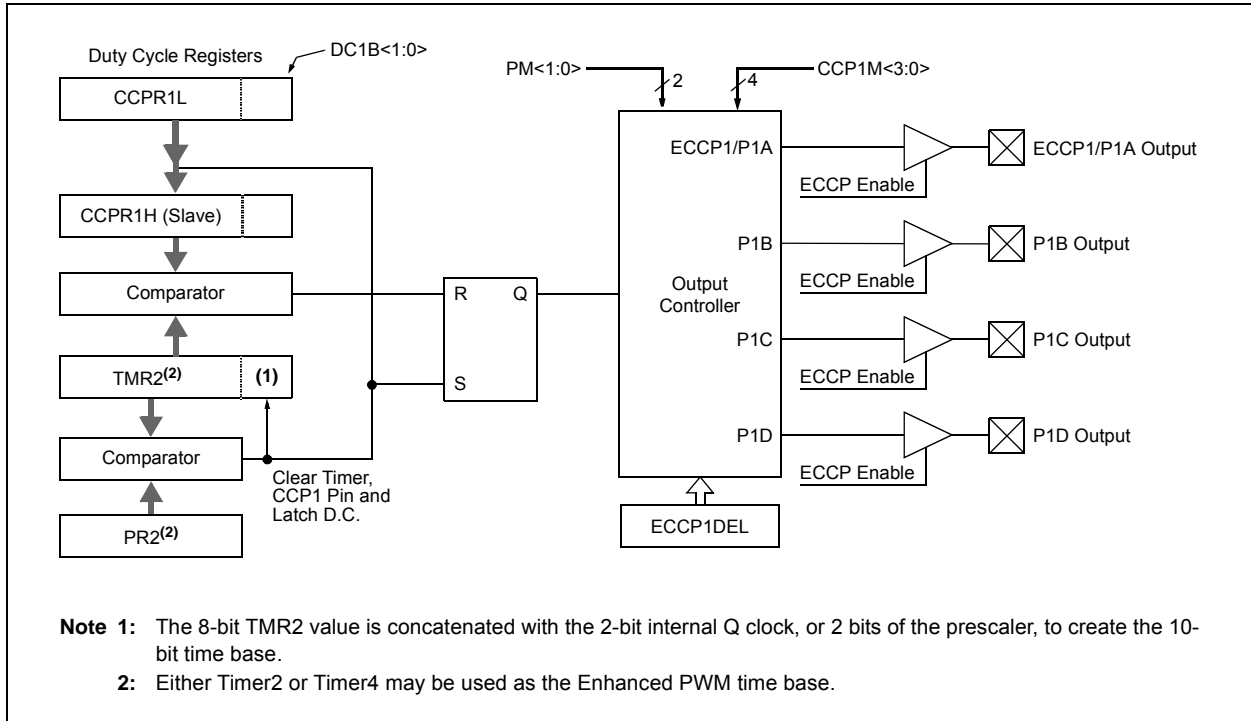


**FIGURE 16-3: SIMPLIFIED PWM BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

**FIGURE 16-4: SIMPLIFIED BLOCK DIAGRAM OF ENHANCED PWM MODE**



# PIC24F16KL402 FAMILY

**REGISTER 16-1: CCPxCON: CCPx CONTROL REGISTER (STANDARD CCP MODULES)**

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3 <sup>(1)</sup>	CCPxM2 <sup>(1)</sup>	CCPxM1 <sup>(1)</sup>	CCPxM0 <sup>(1)</sup>
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 15-6        **Unimplemented:** Read as '0'
- bit 5-4        **DCxB<1:0>:** PWM Duty Cycle Bit 1 and Bit 0 for CCPx Module bits  
                   Capture and Compare modes:  
                   Unused.  
                   PWM mode:  
                   These bits are the two Least Significant bits (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight Most Significant bits (DCxB<9:2>) of the duty cycle are found in CCPRxL.
- bit 3-0        **CCPxM<3:0>:** CCPx Module Mode Select bits<sup>(1)</sup>  
                   1111 = Reserved  
                   1110 = Reserved  
                   1101 = Reserved  
                   1100 = PWM mode  
                   1011 = Compare mode: Special Event Trigger; resets timer on CCPx match (CCPxIF bit is set)  
                   1010 = Compare mode: Generates software interrupt on compare match (CCPxIF bit is set, CCPx pin reflects I/O state)  
                   1001 = Compare mode: Initializes CCPx pin high; on compare match, forces CCPx pin low (CCPxIF bit is set)  
                   1000 = Compare mode: Initializes CCPx pin low; on compare match, forces CCPx pin high (CCPxIF bit is set)  
                   0111 = Capture mode: Every 16th rising edge  
                   0110 = Capture mode: Every 4th rising edge  
                   0101 = Capture mode: Every rising edge  
                   0100 = Capture mode: Every falling edge  
                   0011 = Reserved  
                   0010 = Compare mode: Toggles output on match (CCPxIF bit is set)  
                   0001 = Reserved  
                   0000 = Capture/Compare/PWM is disabled (resets CCPx module)

**Note 1:** CCPxM<3:0> = 1011 will only reset the timer and not start the A/D conversion on a CCPx match.

# PIC24F16KL402 FAMILY

**REGISTER 16-2: CCP1CON: ECCP1 CONTROL REGISTER (ECCP MODULES ONLY)<sup>(1)</sup>**

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
PM1	PM0	DC1B1	DC1B0	CCP1M3 <sup>(2)</sup>	CCP1M2 <sup>(2)</sup>	CCP1M1 <sup>(2)</sup>	CCP1M0 <sup>(2)</sup>
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 7-6 **PM<1:0>:** Enhanced PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A is assigned as a capture input or compare output; P1B, P1C and P1D are assigned as port pins

If CCP1M<3:2> = 11:

11 = Full-bridge output reverse: P1B is modulated; P1C is active; P1A and P1D are inactive

10 = Half-bridge output: P1A, P1B are modulated with dead-band control; P1C and P1D are assigned as port pins

01 = Full-bridge output forward: P1D is modulated; P1A is active; P1B, P1C are inactive

00 = Single output: P1A, P1B, P1C and P1D are controlled by steering

bit 5-4 **DC1B<1:0>:** PWM Duty Cycle bit 1 and bit 0 for CCP1 Module bits

Capture and Compare modes:

Unused.

PWM mode:

These bits are the two Least Significant bits (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight Most Significant bits (DC1B<9:2>) of the duty cycle are found in CCPR1L.

bit 3-0 **CCP1M<3:0>:** ECCP1 Module Mode Select bits<sup>(2)</sup>

1111 = PWM mode: P1A and P1C are active-low; P1B and P1D are active-low

1110 = PWM mode: P1A and P1C are active-low; P1B and P1D are active-high

1101 = PWM mode: P1A and P1C are active-high; P1B and P1D are active-low

1100 = PWM mode: P1A and P1C are active-high; P1B and P1D are active-high

1011 = Compare mode: Special Event Trigger; resets timer on CCP1 match (CCPxIF bit is set)

1010 = Compare mode: Generates software interrupt on compare match (CCP1IF bit is set, CCP1 pin reflects I/O state)

1001 = Compare mode: Initializes CCP1 pin high; on compare match, forces CCP1 pin low (CCP1IF bit is set)

1000 = Compare mode: Initializes CCP1 pin low; on compare match, forces CCP1 pin high (CCP1IF bit is set)

0111 = Capture mode: Every 16th rising edge

0110 = Capture mode: Every 4th rising edge

0101 = Capture mode: Every rising edge

0100 = Capture mode: Every falling edge

0011 = Reserved

0010 = Compare mode: Toggles output on match (CCP1IF bit is set)

0001 = Reserved

0000 = Capture/Compare/PWM is disabled (resets CCP1 module)

**Note 1:** This register is implemented only on PIC24FXXKL40X/30X devices. For all other devices, CCP1CON is configured as [Register 16-1](#).

**2:** CCP1M<3:0> = 1011 will only reset the timer and not start the A/D conversion on a CCP1 match.

# PIC24F16KL402 FAMILY

## REGISTER 16-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN CONTROL REGISTER<sup>(1)</sup>

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 = V<sub>IL</sub> on  $\overline{FLT0}$  pin, or either C1OUT or C2OUT is high

110 = V<sub>IL</sub> on  $\overline{FLT0}$  pin or C2OUT comparator output is high

101 = V<sub>IL</sub> on  $\overline{FLT0}$  pin or C1OUT comparator output is high

100 = V<sub>IL</sub> 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.

# PIC24F16KL402 FAMILY

**REGISTER 16-4: ECCP1DEL: ECCP1 ENHANCED PWM CONTROL REGISTER<sup>(1)</sup>**

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
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
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        **PRSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCPASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCPASE must be cleared by software to restart the PWM

bit 6-0        **PDC<6:0>:** PWM Delay Count bits

PDCn = Number of F<sub>CY</sub> (F<sub>OSC</sub>/2) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it transitions active.

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

# PIC24F16KL402 FAMILY

## REGISTER 16-5: PSTR1CON: ECCP1 PULSE STEERING CONTROL REGISTER<sup>(1)</sup>

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	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA
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 7-6        **CMPL<1:0>:** Complementary Mode Output Assignment Steering bits  
 00 = Complementary output assignment is disabled; the STR<D:A> bits are used to determine Steering mode  
 01 = P1A and P1B are selected as the complementary output pair  
 10 = P1A and P1C are selected as the complementary output pair  
 11 = P1A and P1D are selected as the complementary output pair
- bit 5        **Unimplemented:** Read as '0'
- bit 4        **STRSYNC:** Steering Sync bit  
 1 = Output steering update occurs on the next PWM period  
 0 = Output steering update occurs at the beginning of the instruction cycle boundary
- bit 3        **STRD:** Steering Enable D bit  
 1 = P1D pin has the PWM waveform with polarity control from CCP1M<1:0>  
 0 = P1D pin is assigned to port pin
- bit 2        **STRC:** Steering Enable C bit  
 1 = P1C pin has the PWM waveform with polarity control from CCP1M<1:0>  
 0 = P1C pin is assigned to port pin
- bit 1        **STRB:** Steering Enable B bit  
 1 = P1B pin has the PWM waveform with polarity control from CCP1M<1:0>  
 0 = P1B pin is assigned to port pin
- bit 0        **STRA:** Steering Enable A bit  
 1 = P1A pin has the PWM waveform with polarity control from CCP1M<1:0>  
 0 = P1A pin is assigned to port pin

**Note 1:** This register is only implemented on PIC24FXXKL40X/30X devices. In addition, PWM Steering mode is available only when CCP1M<3:2> = 11 and PM<1:0> = 00.

# PIC24F16KL402 FAMILY

## REGISTER 16-6: CCPTMRS0: CCP TIMER SELECT CONTROL REGISTER 0<sup>(1)</sup>

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'.

# PIC24F16KL402 FAMILY

---

NOTES:

## 17.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP)

**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 MSSP, refer to the “*dsPIC33/PIC24 Family Reference Manual*”.

The Master Synchronous Serial Port (MSSP) module is an 8-bit serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, Shift registers, display drivers, A/D Converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C™)
  - Full Master mode
  - Slave mode (with general address call)

The SPI interface supports these modes in hardware:

- Master mode
- Slave mode
- Daisy-Chaining Operation in Slave mode
- Synchronized Slave operation

The I<sup>2</sup>C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 10-Bit And 7-Bit Addressing and Address Masking
- Byte NACKing
- Selectable Address and Data Hold and Interrupt Masking

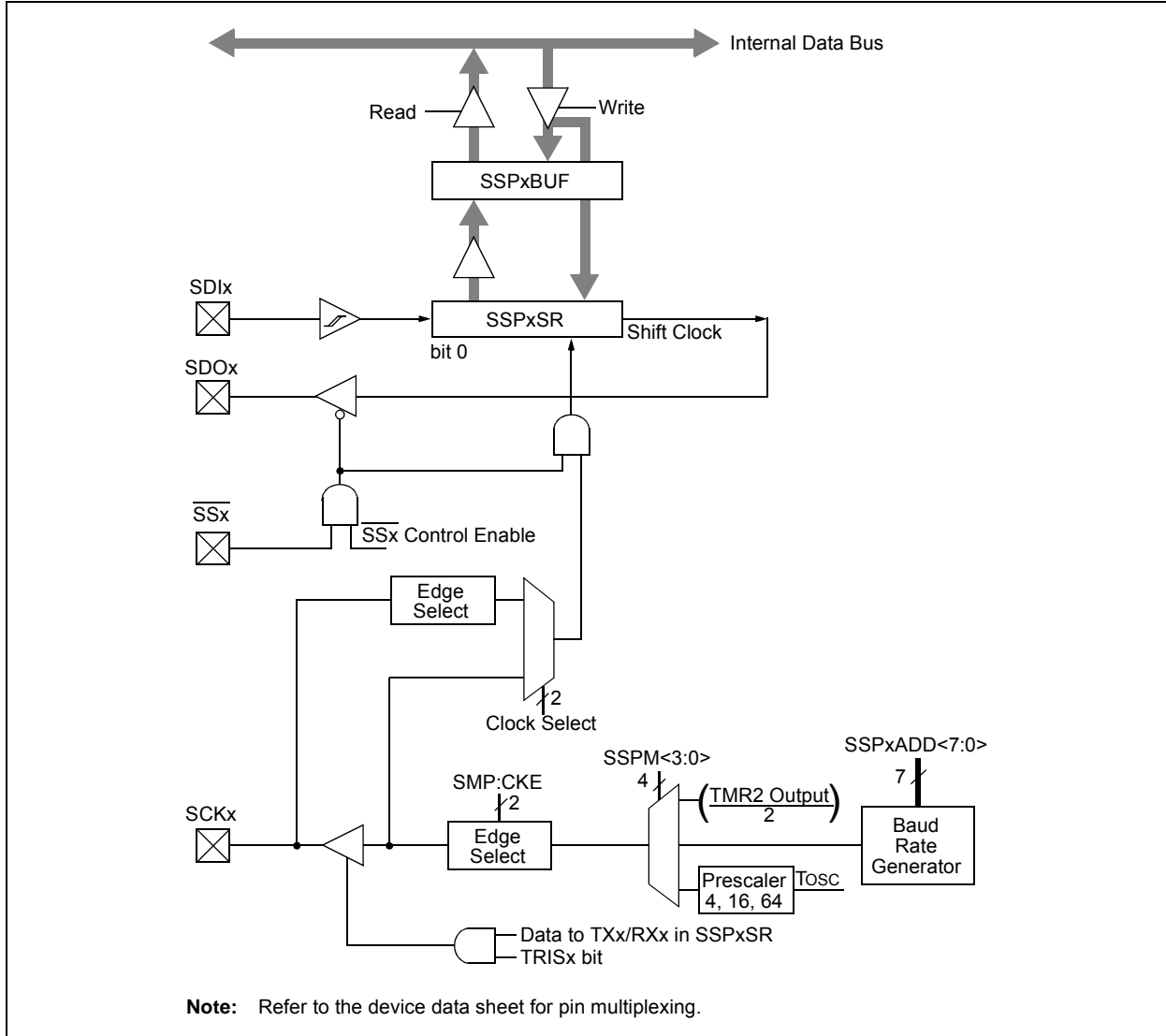
## 17.1 I/O Pin Configuration for SPI

In SPI Master mode, the MSSP module will assert control over any pins associated with the SDOx and SCKx outputs. This does not automatically disable other digital functions associated with the pin, and may result in the module driving the digital I/O port inputs. To prevent this, the MSSP module outputs must be disconnected from their output pins while the module is in SPI Master mode. While disabling the module temporarily may be an option, it may not be a practical solution in all applications.

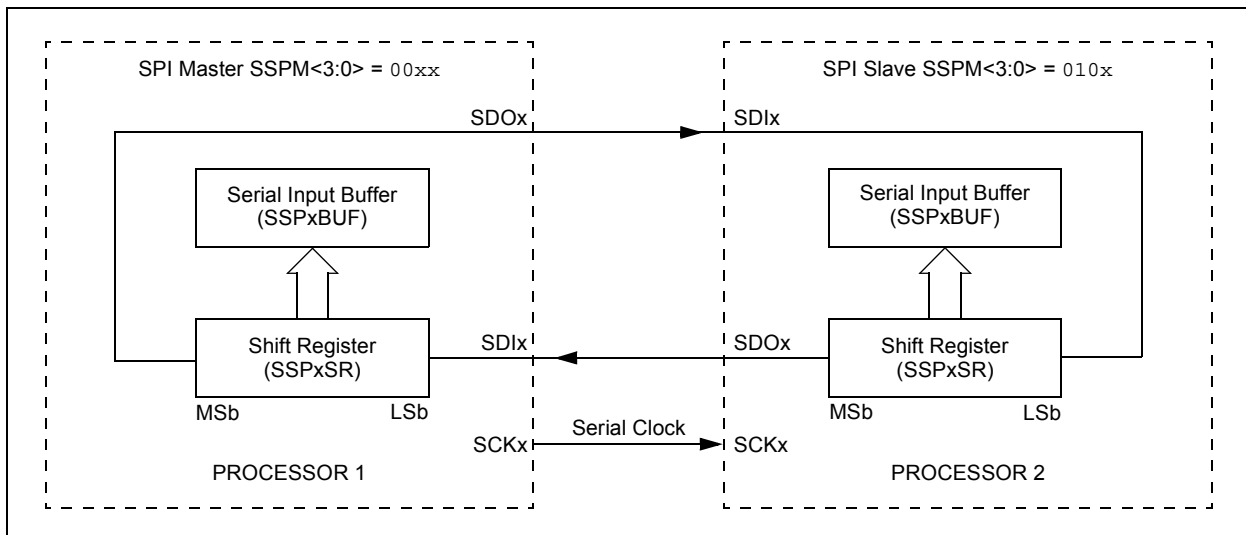
The SDOx and SCKx outputs for the module can be selectively disabled by using the SDOxDIS and SCKxDIS bits in the PADCFG1 register ([Register 17-10](#)). Setting the bit disconnects the corresponding output for a particular module from its assigned pin.

# PIC24F16KL402 FAMILY

**FIGURE 17-1: MSSPx BLOCK DIAGRAM (SPI MODE)**

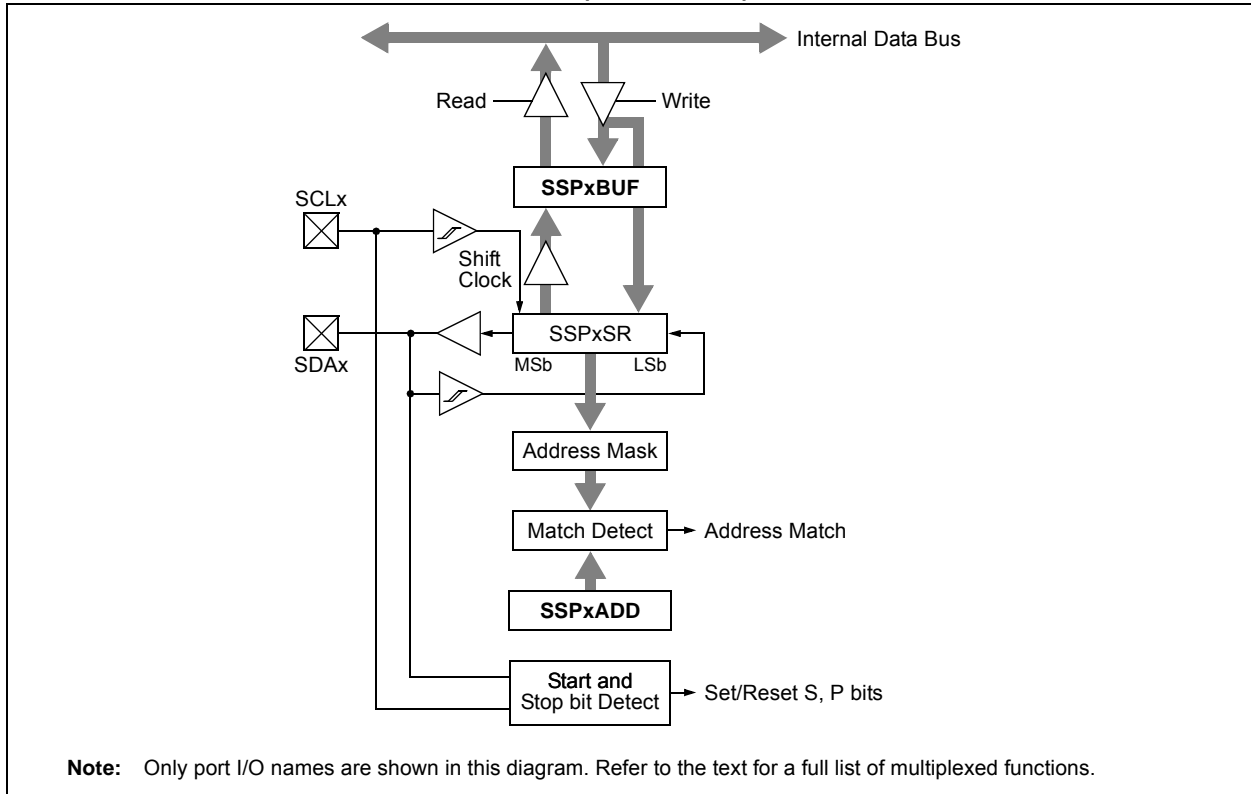


**FIGURE 17-2: SPI MASTER/SLAVE CONNECTION**

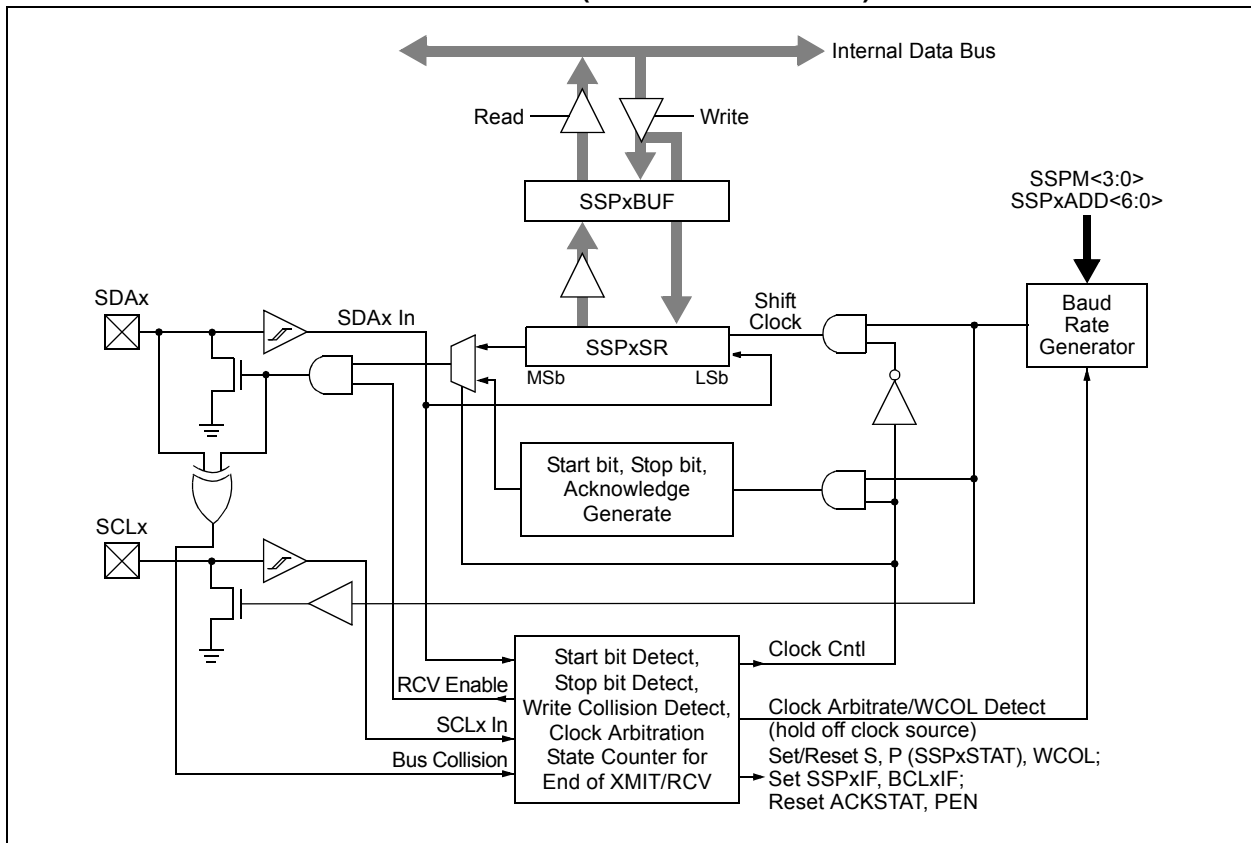


# PIC24F16KL402 FAMILY

**FIGURE 17-3: MSSPx BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)**



**FIGURE 17-4: MSSPx BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



# PIC24F16KL402 FAMILY

## REGISTER 17-1: SSPxSTAT: MSSPx STATUS REGISTER (SPI MODE)

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-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE <sup>(1)</sup>	D $\bar{A}$	P	S	R $\bar{W}$	UA	BF
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            **SMP:** Sample bit

SPI Master mode:

1 = Input data is sampled at the end of data output time  
 0 = Input data is sampled at the middle of data output time

SPI Slave mode:

SMP must be cleared when SPI is used in Slave mode.

bit 6            **CKE:** SPI Clock Select bit<sup>(1)</sup>

1 = Transmit occurs on transition from active to Idle clock state  
 0 = Transmit occurs on transition from Idle to active clock state

bit 5            **D $\bar{A}$ :** Data/Address bit

Used in I<sup>2</sup>C™ mode only.

bit 4            **P:** Stop bit

Used in I<sup>2</sup>C mode only. This bit is cleared when the MSSPx module is disabled; SSPEN is cleared.

bit 3            **S:** Start bit

Used in I<sup>2</sup>C mode only.

bit 2            **R $\bar{W}$ :** Read/Write Information bit

Used in I<sup>2</sup>C mode only.

bit 1            **UA:** Update Address bit

Used in I<sup>2</sup>C mode only.

bit 0            **BF:** Buffer Full Status bit

1 = Receive is complete, SSPxBUF is full  
 0 = Receive is not complete, SSPxBUF is empty

**Note 1:** The polarity of the clock state is set by the CKP bit (SSPxCON1<4>).

# PIC24F16KL402 FAMILY

## REGISTER 17-2: SSPxSTAT: MSSPx STATUS REGISTER (I<sup>2</sup>C™ MODE)

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-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ $\bar{A}$	P <sup>(1)</sup>	S <sup>(1)</sup>	R/ $\bar{W}$	UA	BF
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       **SMP:** Slew Rate Control bit  
In Master or Slave mode:  
 1 = Slew rate control is disabled for Standard Speed mode (100 kHz and 1 MHz)  
 0 = Slew rate control is enabled for High-Speed mode (400 kHz)
- bit 6       **CKE:** SMBus Select bit  
In Master or Slave mode:  
 1 = Enables SMBus specific inputs  
 0 = Disables SMBus specific inputs
- bit 5       **D/ $\bar{A}$ :** Data/Address bit  
In Master mode:  
 Reserved.  
In Slave mode:  
 1 = Indicates that the last byte received or transmitted was data  
 0 = Indicates that the last byte received or transmitted was address
- bit 4       **P:** Stop bit<sup>(1)</sup>  
 1 = Indicates that a Stop bit has been detected last  
 0 = Stop bit was not detected last
- bit 3       **S:** Start bit<sup>(1)</sup>  
 1 = Indicates that a Start bit has been detected last  
 0 = Start bit was not detected last
- bit 2       **R/ $\bar{W}$ :** Read/Write Information bit  
In Slave mode:<sup>(2)</sup>  
 1 = Read  
 0 = Write  
In Master mode:<sup>(3)</sup>  
 1 = Transmit is in progress  
 0 = Transmit is not in progress
- bit 1       **UA:** Update Address bit (10-Bit Slave mode only)  
 1 = Indicates that the user needs to update the address in the SSPxADD register  
 0 = Address does not need to be updated

- Note 1:** This bit is cleared on **RESET** and when **SSPEN** is cleared.
- 2:** This bit holds the **R/ $\bar{W}$**  bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not **ACK** bit.
- 3:** ORing this bit with **SEN**, **RSEN**, **PEN**, **RCEN** or **ACKEN** will indicate if the MSSPx is in Active mode.

# PIC24F16KL402 FAMILY

---

## REGISTER 17-2: SSPxSTAT: MSSPx STATUS REGISTER (I<sup>2</sup>C™ MODE) (CONTINUED)

bit 0      **BF:** Buffer Full Status bit

In Transmit mode:

1 = Transmit is in progress, SSPxBUF is full

0 = Transmit is complete, SSPxBUF is empty

In Receive mode:

1 = SSPxBUF is full (does not include the  $\overline{\text{ACK}}$  and Stop bits)

0 = SSPxBUF is empty (does not include the  $\overline{\text{ACK}}$  and Stop bits)

- Note 1:** This bit is cleared on `RESET` and when `SSPEN` is cleared.
- Note 2:** This bit holds the  $\overline{\text{R/W}}$  bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not  $\overline{\text{ACK}}$  bit.
- Note 3:** ORing this bit with `SEN`, `RSEN`, `PEN`, `RCEN` or `ACKEN` will indicate if the MSSPx is in Active mode.

# PIC24F16KL402 FAMILY

**REGISTER 17-3: SSPxCON1: MSSPx CONTROL REGISTER 1 (SPI MODE)**

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
WCOL	SSPOV <sup>(1)</sup>	SSPEN <sup>(2)</sup>	CKP	SSPM3 <sup>(3)</sup>	SSPM2 <sup>(3)</sup>	SSPM1 <sup>(3)</sup>	SSPM0 <sup>(3)</sup>
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      **WCOL:** Write Collision Detect bit  
             1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)  
             0 = No collision
- bit 6      **SSPOV:** MSSPx Receive Overflow Indicator bit<sup>(1)</sup>  
             SPI Slave mode:  
             1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. The user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).  
             0 = No overflow
- bit 5      **SSPEN:** MSSPx Enable bit<sup>(2)</sup>  
             1 = Enables serial port and configures SCKx, SDOx, SDIx and  $\overline{SSx}$  as serial port pins  
             0 = Disables serial port and configures these pins as I/O port pins
- bit 4      **CKP:** Clock Polarity Select bit  
             1 = Idle state for clock is a high level  
             0 = Idle state for clock is a low level
- bit 3-0      **SSPM<3:0>:** MSSPx Mode Select bits<sup>(3)</sup>  
             1010 = SPI Master mode, Clock =  $F_{osc}/(2 * ([SSPxADD] + 1))$ <sup>(4)</sup>  
             0101 = SPI Slave mode, Clock = SCKx pin;  $\overline{SSx}$  pin control is disabled,  $\overline{SSx}$  can be used as an I/O pin  
             0100 = SPI Slave mode, Clock = SCKx pin;  $\overline{SSx}$  pin control is enabled  
             0011 = SPI Master mode, Clock = TMR2 output/2  
             0010 = SPI Master mode, Clock =  $F_{osc}/32$   
             0001 = SPI Master mode, Clock =  $F_{osc}/8$   
             0000 = SPI Master mode, Clock =  $F_{osc}/2$

- Note 1:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
- 2:** When enabled, these pins must be properly configured as input or output.
- 3:** Bit combinations not specifically listed here are either reserved or implemented in I<sup>2</sup>C mode only.
- 4:** SSPxADD value of 0 is not supported when the Baud Rate Generator is used in SPI mode.

# PIC24F16KL402 FAMILY

## REGISTER 17-4: SSPxCON1: MSSPx CONTROL REGISTER 1 (I<sup>2</sup>C™ MODE)

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
WCOL	SSPOV	SSPEN <sup>(1)</sup>	CKP	SSPM3 <sup>(2)</sup>	SSPM2 <sup>(2)</sup>	SSPM1 <sup>(2)</sup>	SSPM0 <sup>(2)</sup>
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 **WCOL:** Write Collision Detect bit

#### In Master Transmit mode:

1 = A write to the SSPxBUF register was attempted while the I<sup>2</sup>C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

#### In Slave Transmit mode:

1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

#### In Receive mode (Master or Slave modes):

This is a “don't care” bit.

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

#### In Receive mode:

1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

#### In Transmit mode:

This is a “don't care” bit in Transmit mode.

bit 5 **SSPEN:** MSSPx Enable bit<sup>(1)</sup>

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins

0 = Disables the serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCLx Release Control bit

#### In Slave mode:

1 = Releases clock

0 = Holds clock low (clock stretch); used to ensure data setup time

#### In Master mode:

Unused in this mode.

bit 3-0 **SSPM<3:0>:** MSSPx Mode Select bits<sup>(2)</sup>

1111 = I<sup>2</sup>C Slave mode, 10-bit address with Start and Stop bit interrupts is enabled

1110 = I<sup>2</sup>C Slave mode, 7-bit address with Start and Stop bit interrupts is enabled

1011 = I<sup>2</sup>C Firmware Controlled Master mode (Slave Idle)

1000 = I<sup>2</sup>C Master mode, Clock = Fosc/(2 \* ([SSPxADD] + 1))<sup>(3)</sup>

0111 = I<sup>2</sup>C Slave mode, 10-bit address

0110 = I<sup>2</sup>C Slave mode, 7-bit address

**Note 1:** When enabled, the SDAx and SCLx pins must be configured as inputs.

**Note 2:** Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

**Note 3:** SSPxADD values of 0, 1 or 2 are not supported when the Baud Rate Generator is used with I<sup>2</sup>C mode.



# PIC24F16KL402 FAMILY

## REGISTER 17-6: SSPxCON3: MSSPx CONTROL REGISTER 3 (SPI MODE)

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

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACKTIM	PCIE	SCIE	BOEN <sup>(1)</sup>	SDAHT	SBCDE	AHEN	DHEN
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 **ACKTIM:** Acknowledge Time Status bit (I<sup>2</sup>C™ mode only)  
Unused in SPI mode.

bit 6 **PCIE:** Stop Condition Interrupt Enable bit (I<sup>2</sup>C mode only)  
Unused in SPI mode.

bit 5 **SCIE:** Start Condition Interrupt Enable bit (I<sup>2</sup>C mode only)  
Unused in SPI mode.

bit 4 **BOEN:** Buffer Overwrite Enable bit<sup>(1)</sup>

In SPI Slave mode:

1 = SSPxBUF updates every time that a new data byte is shifted in, ignoring the BF bit

0 = If a new byte is received with the BF bit of the SSPxSTAT register already set, the SSPOV bit of the SSPxCON1 register is set and the buffer is not updated

bit 3 **SDAHT:** SDAx Hold Time Selection bit (I<sup>2</sup>C mode only)  
Unused in SPI mode.

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I<sup>2</sup>C Slave mode only)  
Unused in SPI mode.

bit 1 **AHEN:** Address Hold Enable bit (I<sup>2</sup>C Slave mode only)  
Unused in SPI mode.

bit 0 **DHEN:** Data Hold Enable bit (Slave mode only)  
Unused in SPI mode.

**Note 1:** For daisy-chained SPI operation: Allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPxBUF.

# PIC24F16KL402 FAMILY

## REGISTER 17-7: SSPxCON3: MSSPx CONTROL REGISTER 3 (I<sup>2</sup>C™ MODE)

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

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACKTIM <sup>(2)</sup>	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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      **ACKTIM:** Acknowledge Time Status bit<sup>(2)</sup>  
 1 = Indicates the I<sup>2</sup>C bus is in an Acknowledge sequence, set on the 8<sup>th</sup> falling edge of the SCLx clock  
 0 = Not an Acknowledge sequence, cleared on the 9<sup>th</sup> rising edge of the SCLx clock
- bit 6      **PCIE:** Stop Condition Interrupt Enable bit  
 1 = Enables interrupt on detection of a Stop condition  
 0 = Stop detection interrupts are disabled<sup>(1)</sup>
- bit 5      **SCIE:** Start Condition Interrupt Enable bit  
 1 = Enables interrupt on detection of the Start or Restart conditions  
 0 = Start detection interrupts are disabled<sup>(1)</sup>
- bit 4      **BOEN:** Buffer Overwrite Enable bit  
I<sup>2</sup>C Master mode:  
 This bit is ignored.  
I<sup>2</sup>C Slave mode:  
 1 = SSPxBUF is updated and an  $\overline{\text{ACK}}$  is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0  
 0 = SSPxBUF is only updated when SSPOV is clear
- bit 3      **SDAHT:** SDAx Hold Time Selection bit  
 1 = Minimum of 300 ns hold time on SDAx after the falling edge of SCLx  
 0 = Minimum of 100 ns hold time on SDAx after the falling edge of SCLx
- bit 2      **SBCDE:** Slave Mode Bus Collision Detect Enable bit (Slave mode only)  
 1 = Enables slave bus collision interrupts  
 0 = Slave bus collision interrupts are disabled
- bit 1      **AHEN:** Address Hold Enable bit (Slave mode only)  
 1 = Following the 8th falling edge of SCLx for a matching received address byte; the CKP bit of the SSPxCON1 register will be cleared and SCLx will be held low  
 0 = Address holding is disabled
- bit 0      **DHEN:** Data Hold Enable bit (Slave mode only)  
 1 = Following the 8th falling edge of SCLx for a received data byte; slave hardware clears the CKP bit of the SSPxCON1 register and SCLx is held low  
 0 = Data holding is disabled

**Note 1:** This bit has no effect in Slave modes for which Start and Stop condition detection is explicitly listed as enabled.

**2:** The ACKTIM status bit is active only when the AHEN bit or DHEN bit is set.

# PIC24F16KL402 FAMILY

## REGISTER 17-8: SSPxADD: MSSPx SLAVE ADDRESS/BAUD RATE GENERATOR REGISTER

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
ADD<7:0>							
bit 7							bit 0

### Legend:

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

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

bit 7-0                      **ADD<7:0>:** Slave Address/Baud Rate Generator Value bits

#### SPI Master and I<sup>2</sup>C™ Master modes:

Reloads value for Baud Rate Generator. Clock period is  $(([SPxADD] + 1) * 2) / F_{osc}$ .

#### I<sup>2</sup>C Slave modes:

Represents 7 or 8 bits of the slave address, depending on the addressing mode used:

7-Bit mode: Address is ADD<7:1>; ADD<0> is ignored.

10-Bit LSb mode: ADD<7:0> are the Least Significant bits of the address.

10-Bit MSb mode: ADD<2:1> are the two Most Significant bits of the address; ADD<7:3> are always '11110' as a specification requirement, ADD<0> is ignored.

## REGISTER 17-9: SSPxMSK: I<sup>2</sup>C™ SLAVE ADDRESS MASK REGISTER

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

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK<7:0> <sup>(1)</sup>							
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 7-0                      **MSK<7:0>:** Slave Address Mask Select bits<sup>(1)</sup>

1 = Masking of corresponding bit of SSPxADD is enabled

0 = Masking of corresponding bit of SSPxADD is disabled

**Note 1:** MSK0 is not used as a mask bit in 7-bit addressing.

# PIC24F16KL402 FAMILY

## REGISTER 17-10: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SDO2DIS <sup>(1)</sup>	SCK2DIS <sup>(1)</sup>	SDO1DIS	SCK1DIS
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-12 **Unimplemented:** Read as '0'

bit 11 **SDO2DIS:** MSSP2 SDO2 Pin Disable bit<sup>(1)</sup>

1 = The SPI output data (SDO2) of MSSP2 to the pin is disabled

0 = The SPI output data (SDO2) of MSSP2 is output to the pin

bit 10 **SCK2DIS:** MSSP2 SCK2 Pin Disable bit<sup>(1)</sup>

1 = The SPI clock (SCK2) of MSSP2 to the pin is disabled

0 = The SPI clock (SCK2) of MSSP2 is output to the pin

bit 9 **SDO1DIS:** MSSP1 SDO1 Pin Disable bit

1 = The SPI output data (SDO1) of MSSP1 to the pin is disabled

0 = The SPI output data (SDO1) of MSSP1 is output to the pin

bit 8 **SCK1DIS:** MSSP1 SCK1 Pin Disable bit

1 = The SPI clock (SCK1) of MSSP1 to the pin is disabled

0 = The SPI clock (SCK1) of MSSP1 is output to the pin

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

**Note 1:** These bits are implemented only on PIC24FXXKL40X/30X devices.

# PIC24F16KL402 FAMILY

---

NOTES:

## 18.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

**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 Universal Asynchronous Receiver Transmitter, refer to the “dsPIC33/PIC24 Family Reference Manual”, “UART” (DS39708).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in this PIC24F device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. This module also supports a hardware flow control option with the  $\overline{\text{UxCTS}}$  and  $\overline{\text{UxRTS}}$  pins, and also includes an IrDA<sup>®</sup> encoder and decoder.

The primary features of the UART module are:

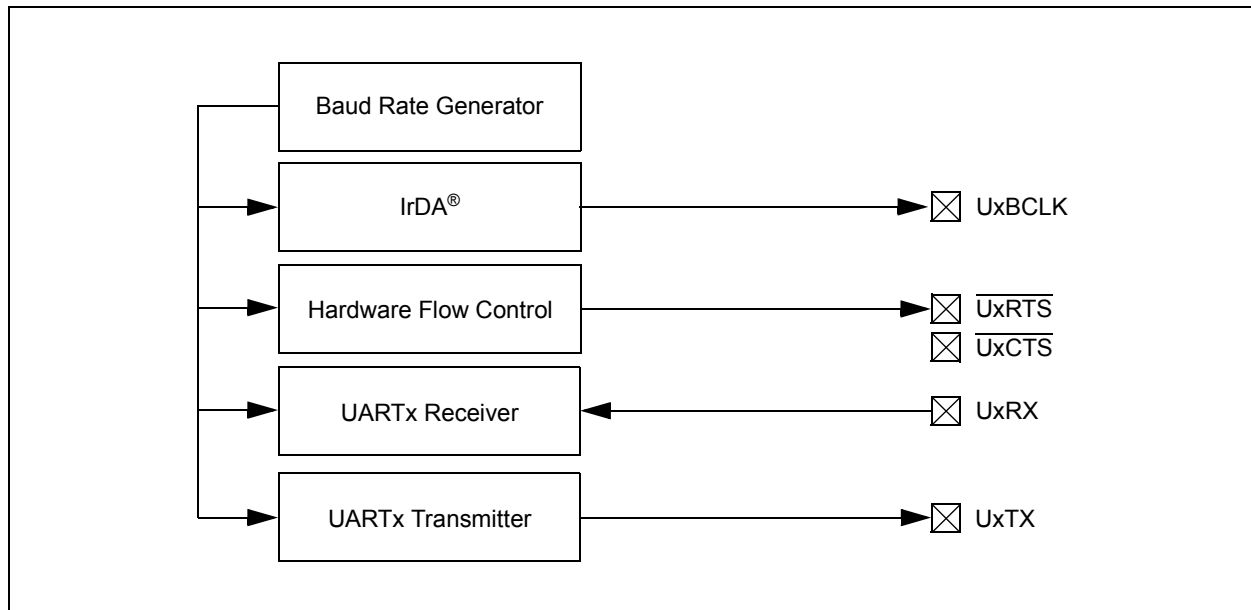
- Full-Duplex, 8-Bit or 9-Bit Data Transmission Through the  $\text{UxTX}$  and  $\text{UxRX}$  Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with  $\overline{\text{UxCTS}}$  and  $\overline{\text{UxRTS}}$  Pins

- Fully Integrated Baud Rate Generator (IBRG) with 16-Bit Prescaler
- Baud Rates Ranging from 1 Mbps to 15 bps at 16 MIPS
- Two-Level Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- Two-Level Deep, FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-Bit mode with Address Detect (9<sup>th</sup> bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA<sup>®</sup> Support

A simplified block diagram of the UART module is shown in Figure 18-1. The UART module consists of these important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

**FIGURE 18-1: UARTx SIMPLIFIED BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## 18.1 UART Baud Rate Generator (BRG)

The UART module includes a dedicated 16-bit Baud Rate Generator (BRG). The UxBRG register controls the period of a free-running, 16-bit timer. Equation 18-1 provides the formula for computation of the baud rate with BRGH = 0.

### EQUATION 18-1: UARTx BAUD RATE WITH BRGH = 0<sup>(1)</sup>

$$\text{Baud Rate} = \frac{\text{FCY}}{16 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{\text{FCY}}{16 \cdot \text{Baud Rate}} - 1$$

**Note 1:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

Example 18-1 provides the calculation of the baud rate error for the following conditions:

- FCY = 4 MHz
- Desired Baud Rate = 9600

### EXAMPLE 18-1: BAUD RATE ERROR CALCULATION (BRGH = 0)<sup>(1)</sup>

$$\text{Desired Baud Rate} = \text{FCY}/(16 (\text{UxBRG} + 1))$$

Solving for UxBRG Value:

$$\text{UxBRG} = ((\text{FCY}/\text{Desired Baud Rate})/16) - 1$$

$$\text{UxBRG} = ((4000000/9600)/16) - 1$$

$$\text{UxBRG} = 25$$

$$\begin{aligned} \text{Calculated Baud Rate} &= 4000000/(16 (25 + 1)) \\ &= 9615 \end{aligned}$$

$$\begin{aligned} \text{Error} &= (\text{Calculated Baud Rate} - \text{Desired Baud Rate}) \\ &\quad \text{Desired Baud Rate} \\ &= (9615 - 9600)/9600 \\ &= 0.16\% \end{aligned}$$

**Note 1:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 0) possible is FCY/16 (for UxBRG = 0) and the minimum baud rate possible is FCY/(16 \* 65536).

Equation 18-2 shows the formula for computation of the baud rate with BRGH = 1.

### EQUATION 18-2: UARTx BAUD RATE WITH BRGH = 1<sup>(1)</sup>

$$\text{Baud Rate} = \frac{\text{FCY}}{4 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{\text{FCY}}{4 \cdot \text{Baud Rate}} - 1$$

**Note 1:** Based on FCY = FOSC/2; Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 1) possible is FCY/4 (for UxBRG = 0) and the minimum baud rate possible is FCY/(4 \* 65536).

Writing a new value to the UxBRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

## 18.2 Transmitting in 8-Bit Data Mode

1. Set up the UART:
  - a) Write appropriate values for data, parity and Stop bits.
  - b) Write appropriate baud rate value to the UxBRG register.
  - c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write data byte to lower byte of UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = 0 and then, the user may set UTXEN. This will cause the serial bit stream to begin immediately, because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bit, UTXISELx.

## 18.3 Transmitting in 9-Bit Data Mode

1. Set up the UART (as described in [Section 18.2 “Transmitting in 8-Bit Data Mode”](#)).
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bit, UTXISELx.

## 18.4 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an auto-baud Sync byte.

1. Configure the UART for the desired mode.
2. Set UTXEN and UTXBRK – sets up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG – loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

## 18.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UART (as described in [Section 18.2 “Transmitting in 8-Bit Data Mode”](#)).
2. Enable the UART.
3. A receive interrupt will be generated when one or more data characters have been received as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

## 18.6 Operation of $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Control Pins

UARTx Clear-to-Send ( $\overline{\text{UxCTS}}$ ) and Request-to-Send ( $\overline{\text{UxRTS}}$ ) are the two hardware-controlled pins that are associated with the UART module. These two pins allow the UART to operate in Simplex and Flow Control modes. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

## 18.7 Infrared Support

The UART module provides two types of infrared UART support: one is the IrDA clock output to support an external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder.

As the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '0'.

### 18.7.1 EXTERNAL IrDA SUPPORT – IrDA CLOCK OUTPUT

To support external IrDA encoder and decoder devices, the UxBCLK pin (same as the  $\overline{\text{UxRTS}}$  pin) can be configured to generate the 16x baud clock. When UEN<1:0> = 11, the UxBCLK pin will output the 16x baud clock if the UART module is enabled; it can be used to support the IrDA codec chip.

### 18.7.2 BUILT-IN IrDA ENCODER AND DECODER

The UART has full implementation of the IrDA encoder and decoder as part of the UART module. The built-in IrDA encoder and decoder functionality is enabled using the IREN bit (UxMODE<12>). When enabled (IREN = 1), the receive pin (UxRX) acts as the input from the infrared receiver. The transmit pin (UxTX) acts as the output to the infrared transmitter.

# PIC24F16KL402 FAMILY

## 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 <sup>(2)</sup>	R/W-0 <sup>(2)</sup>
UARTEN	—	USIDL	IREN <sup>(1)</sup>	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	

<b>Legend:</b>	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<sup>®</sup> Encoder and Decoder Enable bit<sup>(1)</sup>  
 1 = IrDA encoder and decoder are enabled  
 0 = IrDA encoder and decoder are disabled
- bit 11      **RTSMD:** Mode Selection for UxRTS Pin bit  
 1 = UxRTS pin is in Simplex mode  
 0 = UxRTS pin is in Flow Control mode
- bit 10      **Unimplemented:** Read as '0'
- bit 9-8      **UEN<1:0>:** UARTx Enable bits<sup>(2)</sup>  
 11 = UxTX, UxRX and UxBCLK pins are enabled and used; UxCTS pin is controlled by port latches  
 10 = UxTX, UxRX, UxCTS and UxRTS pins are enabled and used  
 01 = UxTX, UxRX and UxRTS pins are enabled and used; UxCTS pin is controlled by port latches  
 00 = UxTX and UxRX pins are enabled and used; UxCTS and 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 only available for the 16x BRG mode (BRGH = 0).

**2:** Bit availability depends on pin availability.

# PIC24F16KL402 FAMILY

---

## REGISTER 18-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

- bit 3        **BRGH:** High Baud Rate Enable bit  
            1 = BRG generates 4 clocks per bit period (4x baud clock, High-Speed mode)  
            0 = BRG generates 16 clocks per bit period (16x baud clock, Standard mode)
- bit 2-1     **PDSEL<1:0>:** Parity and Data Selection bits  
            11 = 9-bit data, no parity  
            10 = 8-bit data, odd parity  
            01 = 8-bit data, even parity  
            00 = 8-bit data, no parity
- bit 0        **STSEL:** Stop Bit Selection bit  
            1 = Two Stop bits  
            0 = One Stop bit

**Note 1:** This feature is only available for the 16x BRG mode (BRGH = 0).

**2:** Bit availability depends on pin availability.

# PIC24F16KL402 FAMILY

## REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0, HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

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

- bit 15,13 **UTXISEL<1:0>**: UARTx Transmission Interrupt Mode Selection bits  
 11 = Reserved; do not use  
 10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR), and as a result, the transmit buffer becomes empty  
 01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed  
 00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)
- bit 14 **UTXINV**: IrDA<sup>®</sup> Encoder Transmit Polarity Inversion bit  
 If IREN = 0:  
 1 = UxTX Idle '0'  
 0 = UxTX Idle '1'  
 If IREN = 1:  
 1 = UxTX Idle '1'  
 0 = UxTX Idle '0'
- bit 12 **Unimplemented**: Read as '0'
- bit 11 **UTXBRK**: UARTx Transmit Break bit  
 1 = Sends Sync Break on next transmission – Start bit, followed by twelve '0' bits; followed by Stop bit; cleared by hardware upon completion  
 0 = Sync Break transmission is disabled or completed
- bit 10 **UTXEN**: UARTx Transmit Enable bit  
 1 = Transmit is enabled; UxTX pin is controlled by UARTx  
 0 = Transmit is disabled; any pending transmission is aborted and the buffer is reset. UxTX pin is controlled by the PORT register.
- bit 9 **UTXBF**: UARTx Transmit Buffer Full Status bit (read-only)  
 1 = Transmit buffer is full  
 0 = Transmit buffer is not full, at least one more character can be written
- bit 8 **TRMT**: Transmit Shift Register Empty bit (read-only)  
 1 = Transmit Shift Register is empty and the transmit buffer is empty (the last transmission has completed)  
 0 = Transmit Shift Register is not empty; a transmission is in progress or queued
- bit 7-6 **URXISEL<1:0>**: UARTx Receive Interrupt Mode Selection bits  
 11 = Interrupt is set on the RSR transfer, making the receive buffer full (i.e., has 2 data characters)  
 10 = Reserved  
 01 = Reserved  
 00 = Interrupt is set when any character is received and transferred from the RSR to the receive buffer; receive buffer has one or more characters

# PIC24F16KL402 FAMILY

---

## REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

- bit 5      **ADDEN:** Address Character Detect bit (bit 8 of the received data = 1)  
1 = Address Detect mode is enabled; if 9-bit mode is not selected, this does not take effect  
0 = Address Detect mode is disabled
- bit 4      **RIDLE:** Receiver Idle bit (read-only)  
1 = Receiver is Idle  
0 = Receiver is active
- bit 3      **PERR:** Parity Error Status bit (read-only)  
1 = Parity error has been detected for the current character (character at the top of the receive FIFO)  
0 = Parity error has not been detected
- bit 2      **FERR:** Framing Error Status bit (read-only)  
1 = Framing error has been detected for the current character (character at the top of the receive FIFO)  
0 = Framing error has not been detected
- bit 1      **OERR:** Receive Buffer Overrun Error Status bit (clear/read-only)  
1 = Receive buffer has overflowed  
0 = Receive buffer has not overflowed (clearing a previously set OERR bit (1 → 0 transition) will reset the receiver buffer and the RSR to the empty state)
- bit 0      **URXDA:** UARTx Receive Buffer Data Available bit (read-only)  
1 = Receive buffer has data; at least one more character can be read  
0 = Receive buffer is empty

# PIC24F16KL402 FAMILY

---

NOTES:

## 19.0 10-BIT HIGH-SPEED A/D CONVERTER

**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 10-Bit High-Speed A/D Converter, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**10-Bit A/D Converter**” (DS39705).

The 10-bit A/D Converter has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 500 ksps
- Up to 12 analog input pins
- External voltage reference input pins
- Internal band gap reference input
- Automatic Channel Scan mode
- Selectable conversion trigger source
- Two-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

Depending on the particular device, PIC24F16KL402 family devices implement up to 12 analog input pins, designated AN0 through AN4 and AN9 through AN15. In addition, there are two analog input pins for external voltage reference connections (VREF+ and VREF-). These voltage reference inputs may be shared with other analog input pins.

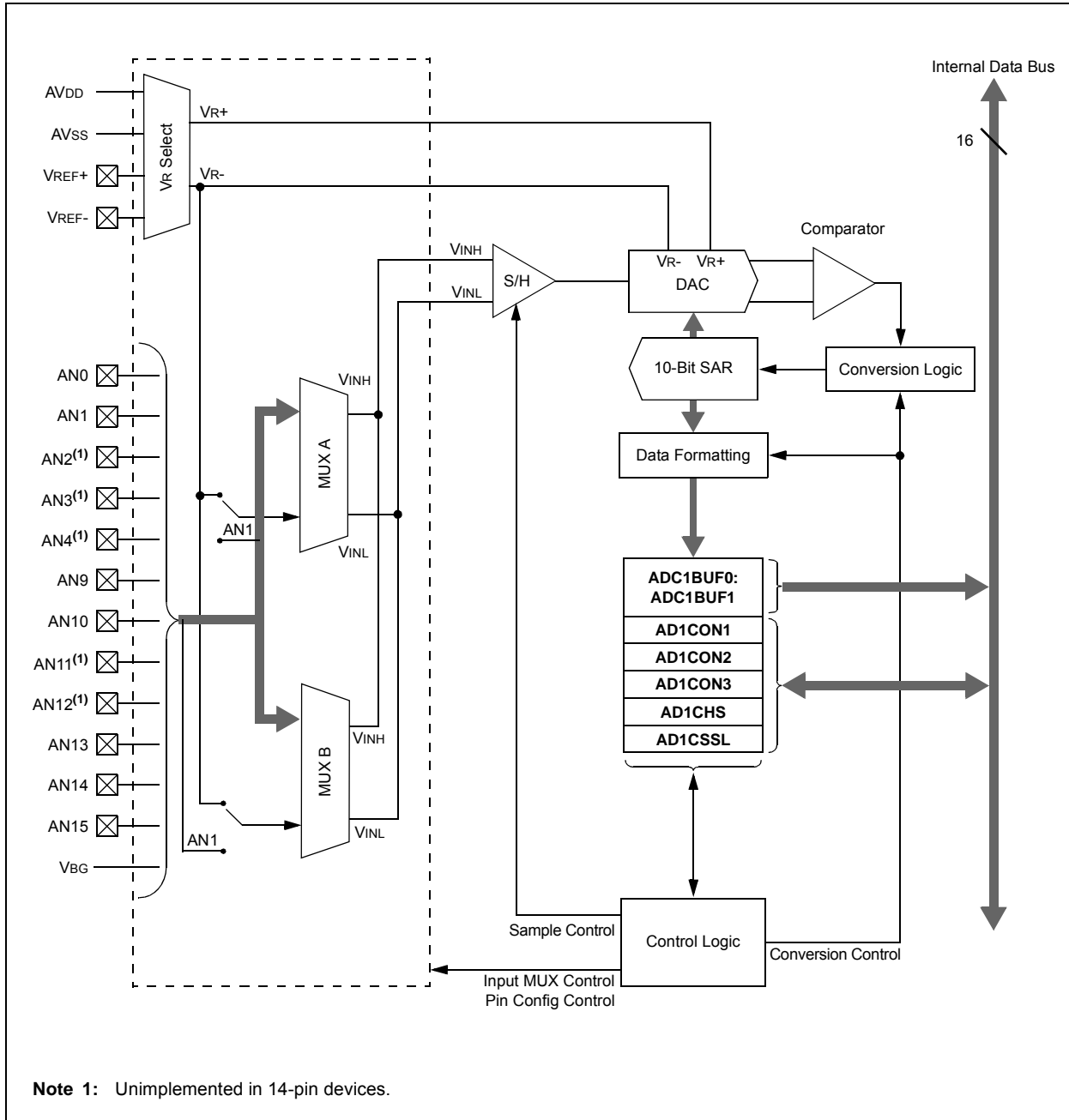
A block diagram of the A/D Converter is displayed in [Figure 19-1](#).

To perform an A/D conversion:

1. Configure the A/D module:
  - a) Configure port pins as analog inputs and/or select band gap reference inputs (ANSA<3:0>, ANSB<15:12,4:0> and ANCFG<0>).
  - b) Select the voltage reference source to match the expected range on analog inputs (AD1CON2<15:13>).
  - c) Select the analog conversion clock to match the desired data rate with the processor clock (AD1CON3<7:0>).
  - d) Select the appropriate sample/conversion sequence (AD1CON1<7:5> and AD1CON3<12:8>).
  - e) Select how conversion results are presented in the buffer (AD1CON1<9:8>).
  - f) Select interrupt rate (AD1CON2<5:2>).
  - g) Turn on A/D module (AD1CON1<15>).
2. Configure A/D interrupt (if required):
  - a) Clear the AD1IF bit.
  - b) Select A/D interrupt priority.

# PIC24F16KL402 FAMILY

**FIGURE 19-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## REGISTER 19-1: AD1CON1: A/D CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON <sup>(1)</sup>	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0, HSC	R-0, HSC
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7						bit 0	

<b>Legend:</b>	HSC = Hardware Settable/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      **ADON:** A/D Operating Mode bit<sup>(1)</sup>  
 1 = A/D Converter module is operating  
 0 = A/D Converter is off
- bit 14      **Unimplemented:** Read as '0'
- bit 13      **ADSIDL:** A/D Stop in Idle Mode bit  
 1 = Discontinues module operation when device enters Idle mode  
 0 = Continues module operation in Idle mode
- bit 12-10   **Unimplemented:** Read as '0'
- bit 9-8     **FORM<1:0>:** Data Output Format bits  
 11 = Signed fractional (sddd dddd dd00 0000)  
 10 = Fractional (dddd dddd dd00 0000)  
 01 = Signed integer (ssss sssd dddd dddd)  
 00 = Integer (0000 00dd dddd dddd)
- bit 7-5     **SSRC<2:0>:** Conversion Trigger Source Select bits  
 111 = Internal counter ends sampling and starts conversion (auto-convert)  
 110 = Reserved  
 101 = Reserved  
 100 = Reserved  
 011 = Reserved  
 010 = Timer1 compare ends sampling and starts conversion  
 001 = Active transition on INTO pin ends sampling and starts conversion  
 000 = Clearing the SAMP bit ends sampling and starts conversion
- bit 4-3     **Unimplemented:** Read as '0'
- bit 2       **ASAM:** A/D Sample Auto-Start bit  
 1 = Sampling begins immediately after the last conversion completes; SAMP bit is auto-set  
 0 = Sampling begins when the SAMP bit is set
- bit 1       **SAMP:** A/D Sample Enable bit  
 1 = A/D Sample-and-Hold amplifier is sampling input  
 0 = A/D Sample-and-Hold amplifier is holding
- bit 0       **DONE:** A/D Conversion Status bit  
 1 = A/D conversion is done  
 0 = A/D conversion is not done

**Note 1:** Values of ADC1BUFx registers will not retain their values once the ADON bit is cleared. Read out the conversion values from the buffer before disabling the module.

# PIC24F16KL402 FAMILY

## REGISTER 19-2: AD1CON2: A/D CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	OFFCAL <sup>(1)</sup>	—	CSCNA	—	—
bit 15						bit 8	

R-x	U-0	R/W-0	R/W-0	R/W-0	R/W-0	r-0	R/W-0
r	—	SMPI3	SMPI2	SMPI1	SMPI0	r	ALTS
bit 7						bit 0	

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

bit 15-13 **VCFG<2:0>**: Voltage Reference Configuration bits

VCFG<2:0>	VR+	VR-
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **OFFCAL**: Offset Calibration bit<sup>(1)</sup>  
 1 = Conversions to get the offset calibration value  
 0 = Conversions to get the actual input value

bit 11 **Unimplemented**: Read as '0'

bit 10 **CSCNA**: Scan Input Selections for MUX A Input Multiplexer bit  
 1 = Scans inputs  
 0 = Does not scan inputs

bit 9-8 **Unimplemented**: Read as '0'

bit 7 **Reserved**: Ignore this value

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

bit 5-2 **SMPI<3:0>**: Sample/Convert Sequences Per Interrupt Selection bits

1111 =

- = Reserved, do not use (may cause conversion data loss)
- 

0010 =

- 0001 = Interrupts at the completion of conversion for each 2<sup>nd</sup> sample/convert sequence
- 0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **Reserved**: Always maintain as '0'

bit 0 **ALTS**: Alternate Input Sample Mode Select bit

- 1 = Uses MUX A input multiplexer settings for the first sample, then alternates between MUX B and MUX A input multiplexer settings for all subsequent samples
- 0 = Always uses MUX A input multiplexer settings

**Note 1:** When the OFFCAL bit is set, inputs are disconnected and tied to AVSS. This sets the inputs of the A/D to zero. Then, the user can perform a conversion. Use of the Calibration mode is not affected by AD1PCFG contents nor channel input selection. Any analog input switches are disconnected from the A/D Converter in this mode. The conversion result is stored by the user software and used to compensate subsequent conversions. This can be done by adding the two's complement of the result obtained with the OFFCAL bit set to all normal A/D conversions.

# PIC24F16KL402 FAMILY

## REGISTER 19-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

### Legend:

R = Readable bit  
-n = Value at POR

W = Writable bit  
'1' = Bit is set

U = Unimplemented bit, read as '0'  
'0' = Bit is cleared  
x = Bit is unknown

bit 15     **ADRC:** A/D Conversion Clock Source bit  
          1 = A/D internal RC clock  
          0 = Clock derived from system clock

bit 14     **EXTSAM:** Extended Sampling Time bit  
          1 = A/D is still sampling after SAMP = 0  
          0 = A/D is finished sampling

bit 13     **PUMPEN:** Charge Pump Enable bit  
          1 = Charge pump for switches is enabled  
          0 = Charge pump for switches is disabled

bit 12-8   **SAMC<4:0>:** Auto-Sample Time bits  
          11111 = 31 TAD  
          •  
          •  
          •  
          00001 = 1 TAD  
          00000 = 0 TAD (not recommended)

bit 7-6     **Unimplemented:** Maintain as '0'

bit 5-0     **ADCS<5:0>:** A/D Conversion Clock Select bits  
          11111 = 64 • Tcy  
          11110 = 63 • Tcy  
          •  
          •  
          •  
          00001 = 2 • Tcy  
          00000 = Tcy

# PIC24F16KL402 FAMILY

**REGISTER 19-4: AD1CHS: A/D INPUT SELECT REGISTER**

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	—	CH0SB3	CH0SB2	CH0SB1	CH0SB0
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	—	CH0SA3	CH0SA2	CH0SA1	CH0SA0
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            **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit  
                   1 = Channel 0 negative input is AN1  
                   0 = Channel 0 negative input is VR-
- bit 14-12       **Unimplemented:** Read as '0'
- bit 11-8        **CH0SB<3:0>:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bits  
                   1111 = AN15  
                   1110 = AN14  
                   1101 = AN13  
                   1100 = AN12<sup>(1)</sup>  
                   1011 = AN11<sup>(1)</sup>  
                   1010 = AN10  
                   1001 = AN9  
                   1000 = Upper guardband rail (0.785 \* VDD)  
                   0111 = Lower guardband rail (0.215 \* VDD)  
                   0110 = Internal band gap reference (V<sub>BG</sub>)  
                   0101 = Reserved; do not use  
                   0100 = AN4<sup>(1)</sup>  
                   0011 = AN3<sup>(1)</sup>  
                   0010 = AN2<sup>(1)</sup>  
                   0001 = AN1  
                   0000 = AN0
- bit 7            **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit  
                   1 = Channel 0 negative input is AN1  
                   0 = Channel 0 negative input is VR-
- bit 6-4        **Unimplemented:** Read as '0'
- bit 3-0        **CH0SA<3:0>:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bits  
                   Bit combinations are identical to those for CH0SB<3:0> (above).

**Note 1:** Unimplemented on 14-pin devices; do not use.

# PIC24F16KL402 FAMILY

## REGISTER 19-5: AD1CSSL: A/D INPUT SCAN SELECT REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL<15:8> <sup>(1)</sup>							
bit 15							bit 8
R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL<7:6>		—	CSSL<4:0> <sup>(1)</sup>				
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 15-6      **CSSL<15:6>**: A/D Input Pin Scan Selection bits<sup>(1)</sup>  
                     1 = Corresponding analog channel selected for input scan  
                     0 = Analog channel omitted from input scan
- bit 5              **Unimplemented**: Read as '0'
- bit 4-0      **CSSL<4:0>**: A/D Input Pin Scan Selection bits<sup>(1)</sup>  
                     1 = Corresponding analog channel selected for input scan  
                     0 = Analog channel omitted from input scan

**Note 1:** CSSL<12:11,4:2> bits are unimplemented on 14-pin devices.

## REGISTER 19-6: ANCFG: ANALOG INPUT CONFIGURATION REGISTER

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
—	—	—	—	—	—	—	VBGEN
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              **VBGEN**: Internal Band Gap Reference Enable bit  
                     1 = Internal band gap voltage is available as a channel input to the A/D Converter  
                     0 = Band gap is not available to the A/D Converter

# PIC24F16KL402 FAMILY

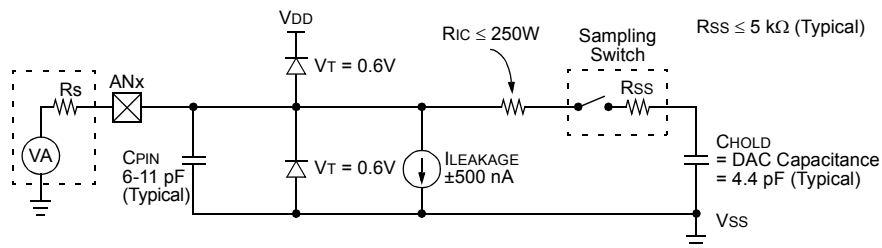
**EQUATION 19-1: A/D CONVERSION CLOCK PERIOD<sup>(1)</sup>**

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

$$T_{AD} = T_{CY} \cdot (ADCS + 1)$$

**Note 1:** Based on  $T_{CY} = 2 \cdot T_{OSC}$ ; Doze mode and PLL are disabled.

**FIGURE 19-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL**

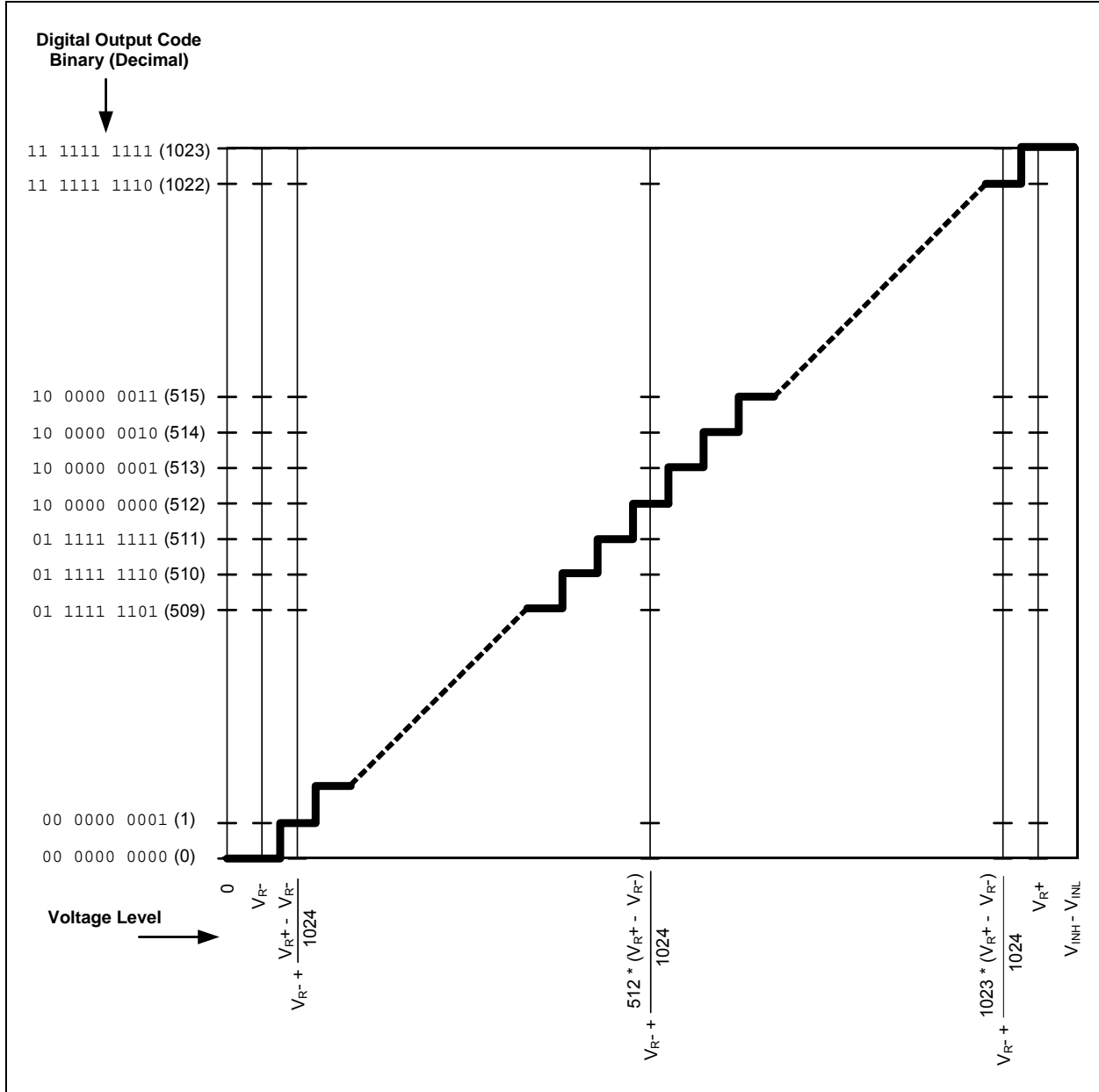


<b>Legend:</b>	CPIN	= Input Capacitance
	VT	= Threshold Voltage
	ILEAKAGE	= Leakage Current at the pin due to Various Junctions
	RIC	= Interconnect Resistance
	Rss	= Sampling Switch Resistance
	CHOLD	= Sample/Hold Capacitance (from DAC)

**Note:** CPIN value depends on device package and is not tested. Effect of CPIN is negligible if  $R_s \leq 5 \text{ k}\Omega$ .

# PIC24F16KL402 FAMILY

FIGURE 19-3: A/D TRANSFER FUNCTION



# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 20.0 COMPARATOR MODULE

**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 Comparator module, refer to the “dsPIC33/PIC24 Family Reference Manual”, “Dual Comparator Module” (DS39710).

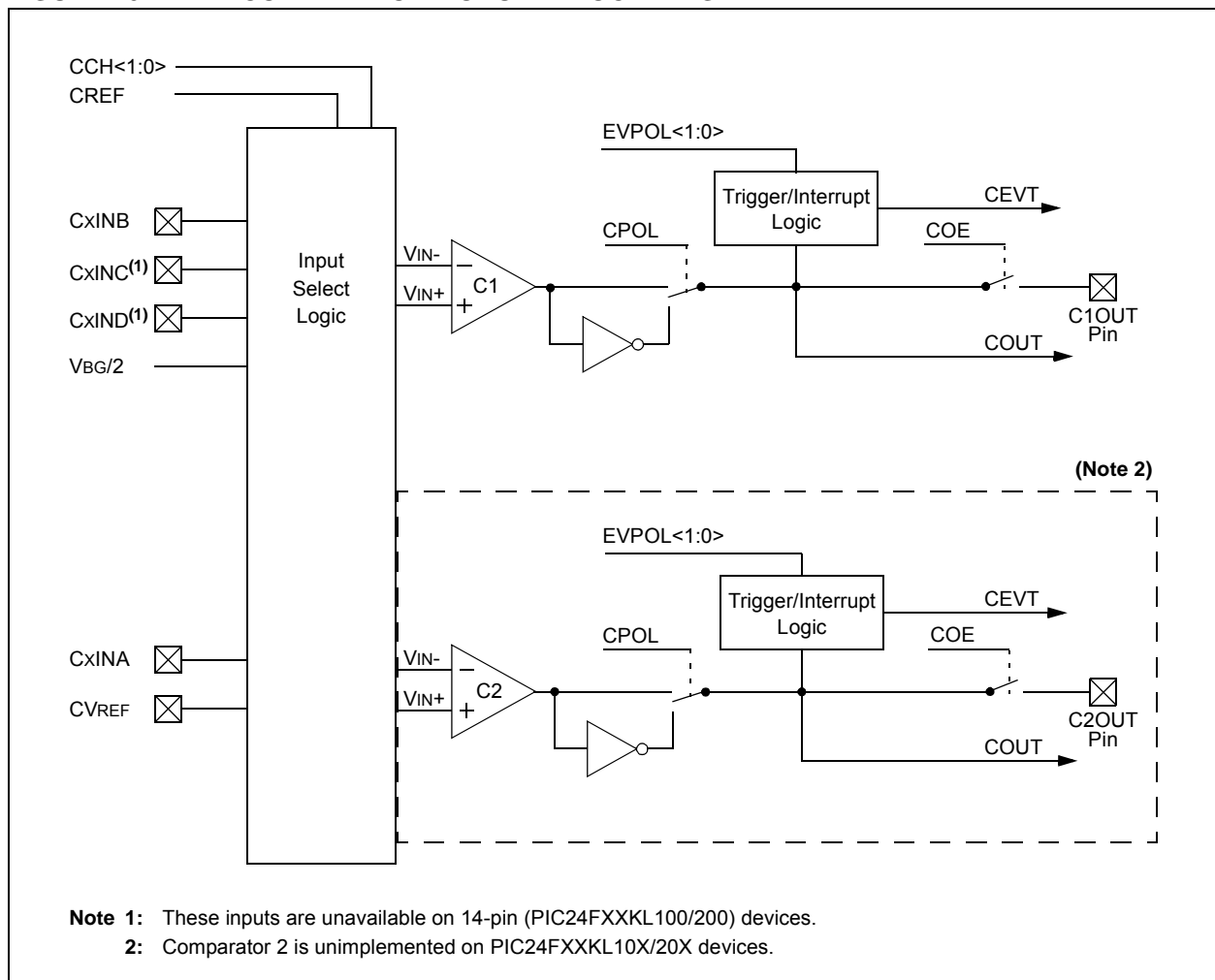
Depending on the particular device, the comparator module provides one or two analog comparators. The inputs to the comparator can be configured to use any one of up to four external analog inputs, as well as a voltage reference input from either the internal band gap reference, divided by 2 ( $V_{BG}/2$ ), or the comparator voltage reference generator.

The comparator outputs may be directly connected to the CxOUT pins. When the respective COE equals ‘1’, the I/O pad logic makes the unsynchronized output of the comparator available on the pin.

A simplified block diagram of the module is displayed in Figure 20-1. Diagrams of the possible individual comparator configurations are displayed in Figure 20-2.

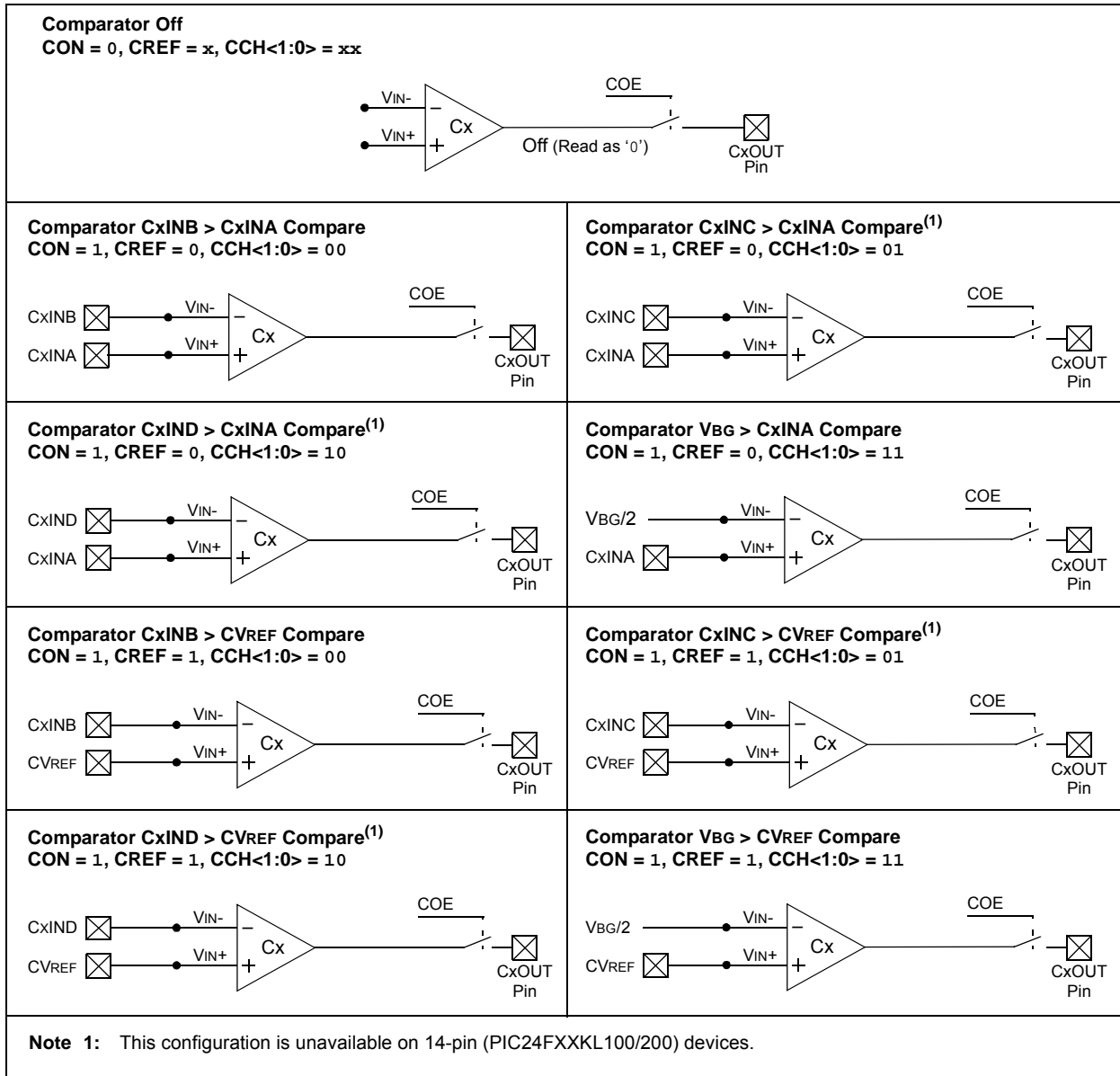
Each comparator has its own control register, CMxCON (Register 20-1), for enabling and configuring its operation. The output and event status of all three comparators is provided in the CMSTAT register (Register 20-2).

**FIGURE 20-1: COMPARATOR MODULE BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

**FIGURE 20-2: INDIVIDUAL COMPARATOR CONFIGURATIONS**



# PIC24F16KL402 FAMILY

## REGISTER 20-1: CMxCON: COMPARATOR x CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R-0
CON	COE	CPOL	CLPWR	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1 <sup>(1)</sup>	EVPOL0 <sup>(1)</sup>	—	CREF	—	—	CCH1	CCH0
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        **CON:** Comparator Enable bit  
                  1 = Comparator is enabled  
                  0 = Comparator is disabled
- bit 14        **COE:** Comparator Output Enable bit  
                  1 = Comparator output is present on the CxOUT pin  
                  0 = Comparator output is internal only
- bit 13        **CPOL:** Comparator Output Polarity Select bit  
                  1 = Comparator output is inverted  
                  0 = Comparator output is not inverted
- bit 12        **CLPWR:** Comparator Low-Power Mode Select bit  
                  1 = Comparator operates in Low-Power mode  
                  0 = Comparator does not operate in Low-Power mode
- bit 11-10    **Unimplemented:** Read as '0'
- bit 9         **CEVT:** Comparator Event bit  
                  1 = Comparator event defined by EVPOL<1:0> has occurred; subsequent triggers and interrupts are disabled until the bit is cleared  
                  0 = Comparator event has not occurred
- bit 8         **COUT:** Comparator Output bit  
                  When CPOL = 0:  
                  1 =  $V_{IN+} > V_{IN-}$   
                  0 =  $V_{IN+} < V_{IN-}$   
                  When CPOL = 1:  
                  1 =  $V_{IN+} < V_{IN-}$   
                  0 =  $V_{IN+} > V_{IN-}$
- bit 7-6       **EVPOL<1:0>:** Trigger/Event/Interrupt Polarity Select bits<sup>(1)</sup>  
                  11 = Trigger/event/interrupt is generated on any change of the comparator output (while CEVT = 0)  
                  10 = Trigger/event/interrupt is generated on the high-to-low transition of the comparator output  
                  01 = Trigger/event/Interrupt is generated on the low-to-high transition of the comparator output  
                  00 = Trigger/event/interrupt generation is disabled
- bit 5         **Unimplemented:** Read as '0'
- bit 4         **CREF:** Comparator Reference Select bits (non-inverting input)  
                  1 = Non-inverting input connects to the internal CVREF voltage  
                  0 = Non-inverting input connects to the CxINA pin

**Note 1:** If EVPOL<1:0> is set to a value other than '00', the first interrupt generated will occur on any transition of COUT, regardless of if it is a rising or falling edge. Subsequent interrupts will occur based on the EVPOLx bits setting.

**2:** Unimplemented on 14-pin (PIC24FXXKL100/200) devices.

# PIC24F16KL402 FAMILY

## REGISTER 20-1: CMxCON: COMPARATOR x CONTROL REGISTER (CONTINUED)

- bit 3-2      **Unimplemented:** Read as '0'
- bit 1-0      **CCH<1:0>:** Comparator Channel Select bits  
 11 = Inverting input of the comparator connects to VBG/2  
 10 = Inverting input of the comparator connects to the CxIND pin<sup>(2)</sup>  
 01 = Inverting input of the comparator connects to the CxINC pin<sup>(2)</sup>  
 00 = Inverting input of the comparator connects to the CxINB pin

**Note 1:** If EVPOL<1:0> is set to a value other than '00', the first interrupt generated will occur on any transition of COUT, regardless of if it is a rising or falling edge. Subsequent interrupts will occur based on the EVPOLx bits setting.

**2:** Unimplemented on 14-pin (PIC24FXXKL100/200) devices.

## REGISTER 20-2: CMSTAT: COMPARATOR MODULE STATUS REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	—	C2EVT <sup>(1)</sup>	C1EVT
bit 15						bit 8	

U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC
—	—	—	—	—	—	C2OUT <sup>(1)</sup>	C1OUT
bit 7						bit 0	

<b>Legend:</b>	HSC = Hardware Settable/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      **CMIDL:** Comparator Stop in Idle Mode bit  
 1 = Discontinues operation of all comparators when device enters Idle mode  
 0 = Continues operation of all enabled comparators in Idle mode
- bit 14-10    **Unimplemented:** Read as '0'
- bit 9        **C2EVT:** Comparator 2 Event Status bit (read-only)<sup>(1)</sup>  
 Shows the current event status of Comparator 2 (CM2CON<9>).
- bit 8        **C1EVT:** Comparator 1 Event Status bit (read-only)  
 Shows the current event status of Comparator 1 (CM1CON<9>).
- bit 7-2      **Unimplemented:** Read as '0'
- bit 1        **C2OUT:** Comparator 2 Output Status bit (read-only)<sup>(1)</sup>  
 Shows the current output of Comparator 2 (CM2CON<8>).
- bit 0        **C1OUT:** Comparator 1 Output Status bit (read-only)  
 Shows the current output of Comparator 1 (CM1CON<8>).

**Note 1:** These bits are unimplemented on PIC24FXXKL10X/20X devices.

# PIC24F16KL402 FAMILY

## 21.0 COMPARATOR VOLTAGE REFERENCE

**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 Comparator Voltage Reference, refer to the “dsPIC33/PIC24 Family Reference Manual”, “**Comparator Voltage Reference Module**” (DS39709).

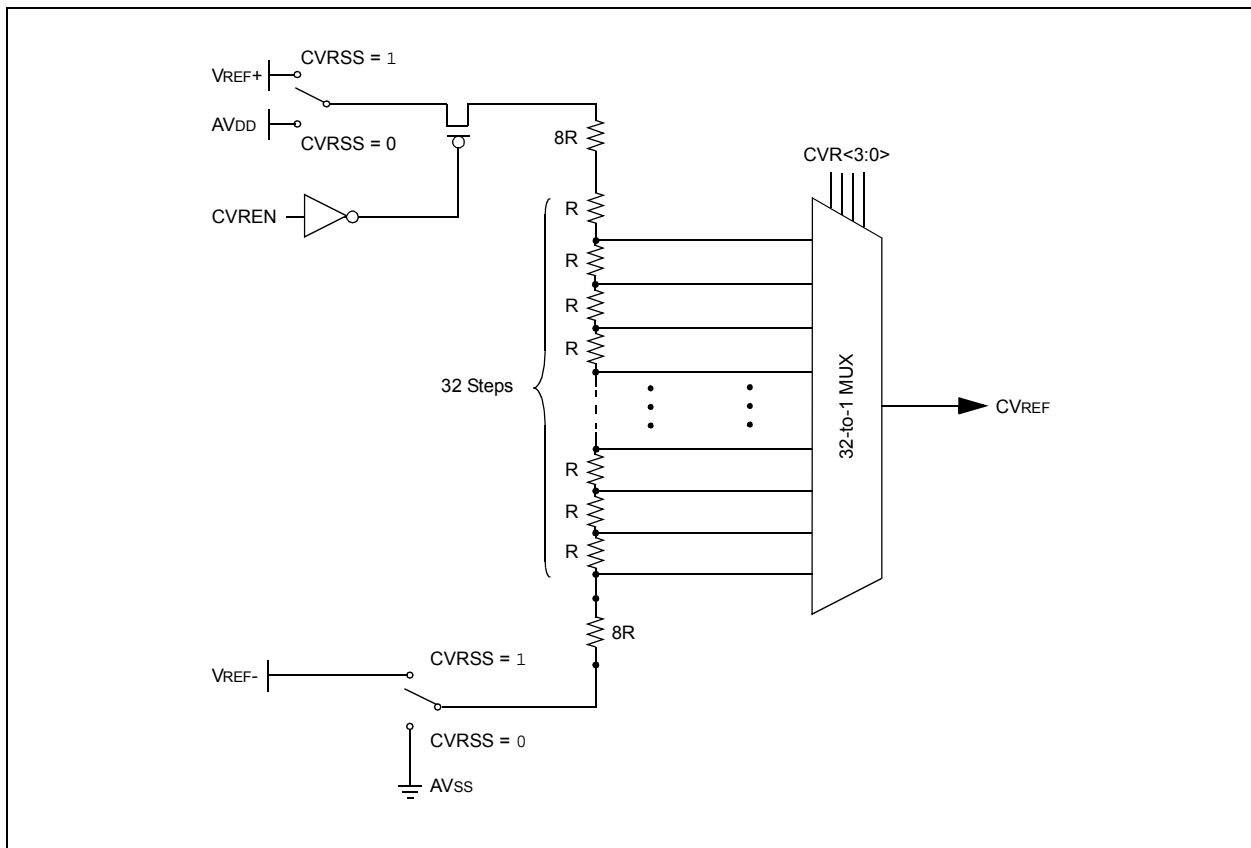
## 21.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register ([Register 21-1](#)). The comparator voltage reference provides a range of output voltages, with 32 distinct levels.

The comparator voltage reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF-. The voltage source is selected by the CVRSS bit (CVRCON<5>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output.

**FIGURE 21-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## REGISTER 21-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

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
CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0
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 **CVREN:** Comparator Voltage Reference Enable bit

1 = CVREF circuit is powered on

0 = CVREF circuit is powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit

1 = CVREF voltage level is output on the CVREF pin

0 = CVREF voltage level is disconnected from the CVREF pin

bit 5 **CVRSS:** Comparator VREF Source Selection bit

1 = Comparator reference source, CVRSRC = VREF+ – VREF-

0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>:** Comparator VREF Value Selection  $0 \leq \text{CVR}<4:0> \leq 31$  bits

When CVRSS = 1:

$\text{CVREF} = (\text{VREF-}) + (\text{CVR}<4:0>/32) \cdot (\text{VREF+} - \text{VREF-})$

When CVRSS = 0:

$\text{CVREF} = (\text{AVSS}) + (\text{CVR}<4:0>/32) \cdot (\text{AVDD} - \text{AVSS})$

## 22.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

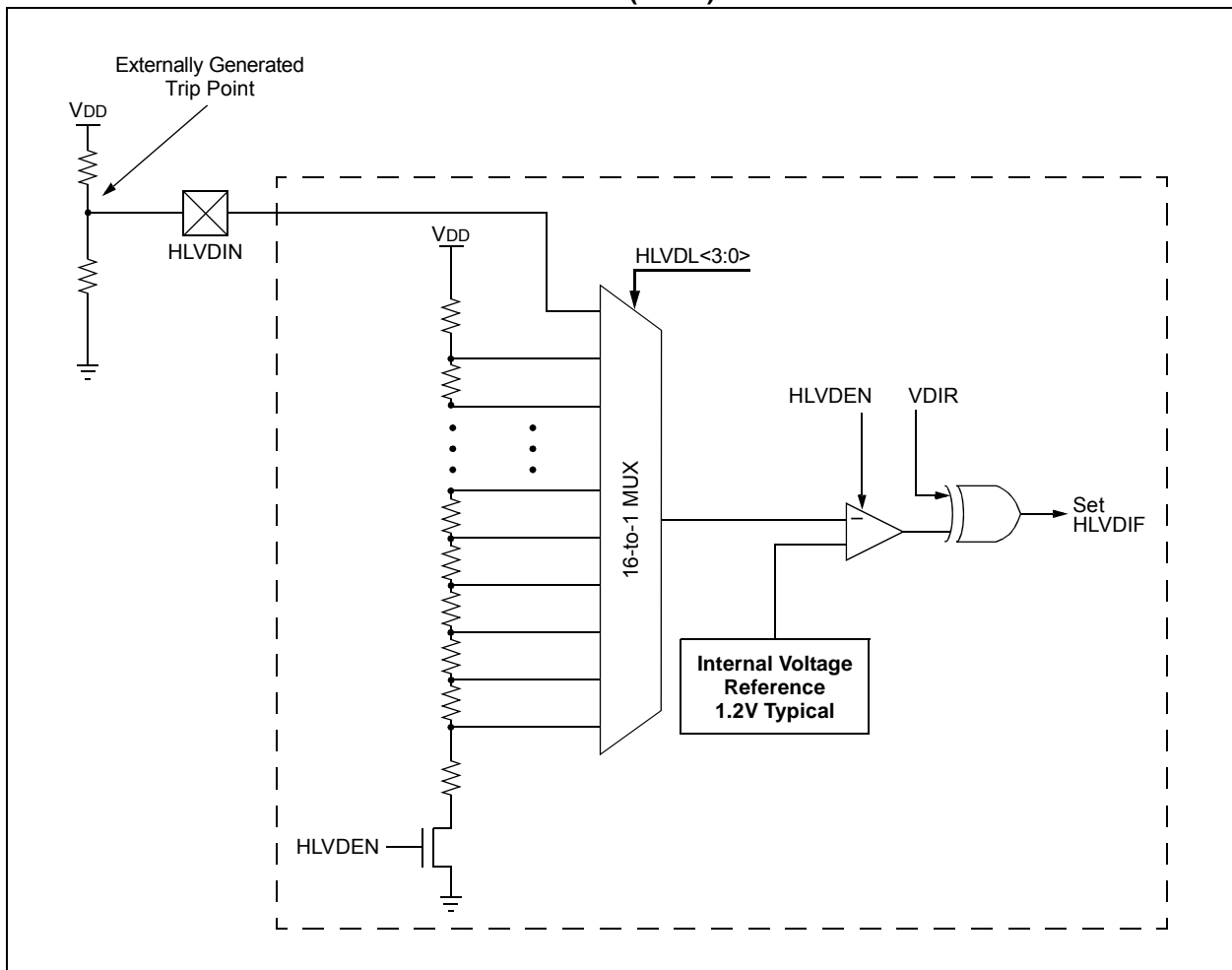
**Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the High/Low-Voltage Detect, refer to the “dsPIC33/PIC24 Family Reference Manual”, “High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)” (DS39725).

An interrupt flag is set if the device experiences an excursion past the trip point in the direction of change. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to the interrupt.

The HLVD Control register (see [Register 22-1](#)) completely controls the operation of the HLVD module. This allows the circuitry to be “turned off” by the user under software control, which minimizes the current consumption for the device.

The High/Low-Voltage Detect module (HLVD) is a programmable circuit that allows the user to specify both the device voltage trip point and the direction of change.

**FIGURE 22-1: HIGH/LOW-VOLTAGE DETECT (HLVD) MODULE BLOCK DIAGRAM**



# PIC24F16KL402 FAMILY

## REGISTER 22-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	HLSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
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            **HLVDEN:** High/Low-Voltage Detect Power Enable bit  
                   1 = HLVD is enabled  
                   0 = HLVD is disabled
- bit 14            **Unimplemented:** Read as '0'
- bit 13            **HLSIDL:** HLVD Stop in Idle Mode bit  
                   1 = Discontinues module operation when the device enters Idle mode  
                   0 = Continues module operation in Idle mode
- bit 12-8        **Unimplemented:** Read as '0'
- bit 7            **VDIR:** Voltage Change Direction Select bit  
                   1 = Event occurs when the voltage equals or exceeds the trip point (HLVDL<3:0>)  
                   0 = Event occurs when the voltage equals or falls below the trip point (HLVDL<3:0>)
- bit 6            **BGVST:** Band Gap Voltage Stable Flag bit  
                   1 = Indicates that the band gap voltage is stable  
                   0 = Indicates that the band gap voltage is unstable
- bit 5            **IRVST:** Internal Reference Voltage Stable Flag bit  
                   1 = Indicates that the internal reference voltage is stable and the High-Voltage Detect logic generates the interrupt flag at the specified voltage range  
                   0 = Indicates that the internal reference voltage is unstable and the High-Voltage Detect logic will not generate the interrupt flag at the specified voltage range, and the HLVD interrupt should not be enabled
- bit 4            **Unimplemented:** Read as '0'
- bit 3-0        **HLVDL<3:0>:** High/Low-Voltage Detection Limit bits  
                   1111 = External analog input is used (input comes from the HLVDIN pin)  
                   1110 = Trip Point 14<sup>(1)</sup>  
                   1101 = Trip Point 13<sup>(1)</sup>  
                   1100 = Trip Point 12<sup>(1)</sup>  
                   .  
                   .  
                   .  
                   0000 = Trip Point 0<sup>(1)</sup>

**Note 1:** For the actual trip point, see [Section 26.0 “Electrical Characteristics”](#).

# PIC24F16KL402 FAMILY

## 23.0 SPECIAL FEATURES

**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 Watchdog Timer, High-Level Device Integration and Programming Diagnostics, refer to the individual sections of the “*dsPIC33/PIC24 Family Reference Manual*” provided below:

- “**Watchdog Timer (WDT)**” (DS39697)
- “**High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)**” (DS39725)
- “**Programming and Diagnostics**” (DS39716)

PIC24F16KL402 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation
- Factory Programmed Unique ID

### 23.1 Code Protect Security Options

The Boot Segment (BS) and General Segment (GS) are two segments on this device with separate programmable security levels. The Boot Segment, configured via the FBS Configuration register, can have three possible levels of security:

- **No Security (BSS = 111):** The Boot Segment is not utilized and all addresses in program memory are part of the General Segment (GS).
- **Standard Security (BSS = 110 or 101):** The Boot Segment is enabled and code-protected, preventing ICSP reads of the Flash memory. Standard security also prevents Flash reads and writes of the BS from the GS. The BS can still read and write to itself.
- **High Security (BSS = 010 or 001):** The Boot Segment is enabled with all of the security provided by Standard Security mode. In addition, in High-Security mode, there are program flow change restrictions in place. While executing from the GS, program flow changes that attempt to enter the BS (e.g., branch (BRA) or CALL instructions) can only enter the BS at one of the first 32 instruction locations (0x200 to 0x23F). Attempting to jump into the BS at an instruction higher than this will result in an Illegal Opcode Reset.

The General Segment, configured via the FGS Configuration register, can have two levels of security:

- **No Security (GSS0 = 1):** The GS is not code-protected and can be read in all modes.
- **Standard Security (GSS0 = 0):** The GS is code-protected, preventing ICSP reads of the Flash memory.

For more detailed information on these Security modes, refer to the “*dsPIC33/PIC24 Family Reference Manual*”, “**CodeGuard™ Security**” (DS70199).

### 23.2 Configuration Bits

The Configuration bits can be programmed (read as ‘0’), or left unprogrammed (read as ‘1’), to select various device configurations. These bits are mapped starting at program memory location, F80000h. A complete list is provided in [Table 23-1](#). A detailed explanation of the various bit functions is provided in [Register 23-1](#) through [Register 23-7](#).

The address, F80000h, is beyond the user program memory space. In fact, it belongs to the configuration memory space (800000h-FFFFFFh), which can only be accessed using Table Reads and Table Writes.

**TABLE 23-1: CONFIGURATION REGISTERS LOCATIONS**

Configuration Register	Address
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E

# PIC24F16KL402 FAMILY

## REGISTER 23-1: FBS: BOOT SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	R/C-1 <sup>(1)</sup>	R/C-1 <sup>(1)</sup>	R/C-1 <sup>(1)</sup>	R/C-1 <sup>(1)</sup>
—	—	—	—	BSS2	BSS1	BSS0	BWRP
bit 7							bit 0

### Legend:

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

bit 7-4                      **Unimplemented:** Read as '0'

bit 3-1                      **BSS<2:0>:** Boot Segment Program Flash Code Protection bits<sup>(1)</sup>  
 111 = No Boot Segment; all program memory space is General Segment  
 110 = Standard security Boot Segment starts at 0200h, ends at 0AFEh  
 101 = Standard security Boot Segment starts at 0200h, ends at 15FEh<sup>(2)</sup>  
 100 = Reserved  
 011 = Reserved  
 010 = High-security Boot Segment starts at 0200h, ends at 0AFEh  
 001 = High-security Boot Segment starts at 0200h, ends at 15FEh<sup>(2)</sup>  
 000 = Reserved

bit 0                      **BWRP:** Boot Segment Program Flash Write Protection bit<sup>(1)</sup>  
 1 = Boot Segment may be written  
 0 = Boot Segment is write-protected

**Note 1:** Code protection bits can only be programmed by clearing them. They can be reset to their default factory state ('1'), but only by performing a bulk erase and reprogramming the entire device.

**2:** This selection is available only on PIC24F16KL40X devices.

## REGISTER 23-2: FGS: GENERAL SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1 <sup>(1)</sup>	R/C-1 <sup>(1)</sup>
—	—	—	—	—	—	GSS0	GWRP
bit 7							bit 0

### Legend:

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

bit 7-2                      **Unimplemented:** Read as '0'

bit 1                      **GSS0:** General Segment Code Flash Code Protection bit<sup>(1)</sup>  
 1 = No protection  
 0 = Standard security is enabled

bit 0                      **GWRP:** General Segment Code Flash Write Protection bit<sup>(1)</sup>  
 1 = General Segment may be written  
 0 = General Segment is write-protected

**Note 1:** Code protection bits can only be programmed by clearing them. They can be reset to their default factory state ('1'), but only by performing a bulk erase and reprogramming the entire device.

# PIC24F16KL402 FAMILY

## REGISTER 23-3: FOSCSEL: OSCILLATOR SELECTION CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-0	R/P-0	R/P-1
IESO	LPRCSEL	SOSCSRC	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

### Legend:

R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 7      **IESO:** Internal External Switchover bit  
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)  
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)
- bit 6      **LPRCSEL:** Internal LPRC Oscillator Power Select bit  
1 = High-Power/High-Accuracy mode  
0 = Low-Power/Low-Accuracy mode
- bit 5      **SOSCSRC:** Secondary Oscillator Clock Source Configuration bit  
1 = SOSC analog crystal function is available on the SOSCI/SOSCO pins  
0 = SOSC crystal is disabled; digital SCLKI function is selected on the SOSCO pin
- bit 4-3    **Unimplemented:** Read as '0'
- bit 2-0    **FNOSC<2:0>:** Oscillator Selection bits  
111 = 8 MHz FRC Oscillator with Divide-by-N (FRCDIV)  
110 = 500 kHz Low-Power FRC Oscillator with Divide-by-N (LPFRCDIV)  
101 = Low-Power RC Oscillator (LPRC)  
100 = Secondary Oscillator (SOSC)  
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)  
010 = Primary Oscillator (XT, HS, EC)  
001 = 8 MHz FRC Oscillator with Divide-by-N with PLL module (FRCDIV+PLL)  
000 = 8 MHz FRC Oscillator (FRC)

# PIC24F16KL402 FAMILY

## REGISTER 23-4: FOSC: OSCILLATOR CONFIGURATION REGISTER

R/P-0	R/P-0	R/P-1	R/P-1	R/P-1	R/P-0	R/P-1	R/P-1
FCKSM1	FCKSM0	SOSCSEL	POSCFREQ1	POSCFREQ0	OSCIOFNC	POSCMD1	POSCMD0
bit 7							bit 0

### Legend:

R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 7-6      **FCKSM<1:0>**: Clock Switching and Monitor Selection Configuration bits  
 1x = Clock switching is disabled, Fail-Safe Clock Monitor is disabled  
 01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled  
 00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled
- bit 5      **SOSCSEL**: Secondary Oscillator Power Selection Configuration bit  
 1 = Secondary oscillator is configured for high-power operation  
 0 = Secondary oscillator is configured for low-power operation
- bit 4-3      **POSCFREQ<1:0>**: Primary Oscillator Frequency Range Configuration bits  
 11 = Primary oscillator/external clock input frequency is greater than 8 MHz  
 10 = Primary oscillator/external clock input frequency is between 100 kHz and 8 MHz  
 01 = Primary oscillator/external clock input frequency is less than 100 kHz  
 00 = Reserved; do not use
- bit 2      **OSCIOFNC**: CLKO Enable Configuration bit  
 1 = CLKO output signal is active on the OSCO pin; primary oscillator must be disabled or configured for the External Clock mode (EC) for the CLKO to be active (POSCMD<1:0> = 11 or 00)  
 0 = CLKO output is disabled
- bit 1-0      **POSCMD<1:0>**: Primary Oscillator Configuration bits  
 11 = Primary Oscillator mode is disabled  
 10 = HS Oscillator mode is selected  
 01 = XT Oscillator mode is selected  
 00 = External Clock mode is selected

# PIC24F16KL402 FAMILY

## REGISTER 23-5: FWDT: WATCHDOG TIMER CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FWDTEN1	WINDIS	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

### Legend:

R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

bit 7,5 **FWDTEN<1:0>**: Watchdog Timer Enable bits

11 = WDT is enabled in hardware  
 10 = WDT is controlled with the SWDTEN bit setting  
 01 = WDT is enabled only while device is active; WDT is disabled in Sleep, SWDTEN bit is disabled  
 00 = WDT is disabled in hardware; SWDTEN bit is disabled

bit 6 **WINDIS**: Windowed Watchdog Timer Disable bit

1 = Standard WDT is selected; windowed WDT is disabled  
 0 = Windowed WDT is enabled; note that executing a CLRWDWT instruction while the WDT is disabled in hardware and software (FWDTEN<1:0> = 00 and SWDTEN (RCON<5> = 0) will not cause a device Reset

bit 4 **FWPSA**: WDT Prescaler bit

1 = WDT prescaler ratio of 1:128  
 0 = WDT prescaler ratio of 1:32

bit 3-0 **WDTPS<3:0>**: Watchdog Timer Postscale Select bits

1111 = 1:32,768  
 1110 = 1:16,384  
 1101 = 1:8,192  
 1100 = 1:4,096  
 1011 = 1:2,048  
 1010 = 1:1,024  
 1001 = 1:512  
 1000 = 1:256  
 0111 = 1:128  
 0110 = 1:64  
 0101 = 1:32  
 0100 = 1:16  
 0011 = 1:8  
 0010 = 1:4  
 0001 = 1:2  
 0000 = 1:1

# PIC24F16KL402 FAMILY

## REGISTER 23-6: FPOR: RESET CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-0	R/P-1	R/P-1
MCLRE <sup>(1)</sup>	BORV1 <sup>(2)</sup>	BORV0 <sup>(2)</sup>	I2C1SEL <sup>(3)</sup>	PWRTEN	—	BOREN1	BOREN0
bit 7							bit 0

### Legend:

R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

- bit 7      **MCLRE:**  $\overline{\text{MCLR}}$  Pin Enable bit<sup>(1)</sup>  
 1 =  $\overline{\text{MCLR}}$  pin is enabled; RA5 input pin is disabled  
 0 = RA5 input pin is enabled;  $\overline{\text{MCLR}}$  is disabled
- bit 6-5    **BORV<1:0>:** Brown-out Reset Enable bits<sup>(2)</sup>  
 11 = Brown-out Reset is set to the low trip point  
 10 = Brown-out Reset is set to the middle trip point  
 01 = Brown-out Reset is set to the high trip point  
 00 = Downside protection on POR is enabled (Low-Power BOR is selected)
- bit 4      **I2C1SEL:** Alternate MSSP1 I<sup>2</sup>C™ Pin Mapping bit<sup>(3)</sup>  
 1 = Default location for SCL1/SDA1 pins (RB8 and RB9)  
 0 = Alternate location for SCL1/SDA1 pins (ASCL1/RB6 and ASDA1/RB5)
- bit 3      **PWRTEN:** Power-up Timer Enable bit  
 1 = PWRT is enabled  
 0 = PWRT is disabled
- bit 2      **Unimplemented:** Read as '0'
- bit 1-0    **BOREN<1:0>:** Brown-out Reset Enable bits  
 11 = BOR is enabled in hardware; SBOREN bit is disabled  
 10 = BOR is enabled only while device is active and disabled in Sleep; SBOREN bit is disabled  
 01 = BOR is controlled with the SBOREN bit setting  
 00 = BOR is disabled in hardware; SBOREN bit is disabled

- Note 1:** The MCLRE fuse can only be changed when using the VPP-Based ICSP™ mode entry. This prevents a user from accidentally locking out the device from the low-voltage test entry.
- 2:** Refer to [Table 26-5](#) for BOR trip point voltages.
- 3:** Implemented in 28-pin devices only. This bit position must be programmed (= 1) in all other devices for I<sup>2</sup>C functionality to be available.

# PIC24F16KL402 FAMILY

## REGISTER 23-7: FICD: IN-CIRCUIT DEBUGGER CONFIGURATION REGISTER

R/P-1	U-1	U-1	U-0	U-0	U-0	R/P-1	R/P-1
DEBUG	—	—	—	—	—	ICS1	ICS0
bit 7							bit 0

### Legend:

R = Readable bit                      P = Programmable bit                      U = Unimplemented bit, read as '0'  
 -n = Value at POR                      '1' = Bit is set                      '0' = Bit is cleared                      x = Bit is unknown

- bit 7                      **DEBUG:** Background Debugger Enable bit  
                             1 = Background debugger is disabled  
                             0 = Background debugger functions are enabled
- bit 6-5                      **Unimplemented:** Read as '1'
- bit 4-2                      **Unimplemented:** Read as '0'
- bit 1-0                      **ICS<1:0>:** ICD Pin Select bits  
                             11 = PGEC1/PGED1 are used for programming and debugging the device<sup>(1)</sup>  
                             10 = PGEC2/PGED2 are used for programming and debugging the device  
                             01 = PGEC3/PGED3 are used for programming and debugging the device  
                             00 = Reserved; do not use

**Note 1:** PGEC1/PGED1 are not available on PIC24F04KL100 (14-pin) devices.

# PIC24F16KL402 FAMILY

## 23.3 Unique ID

A read-only Unique ID value is stored at addresses, 800802h through 800808h. This factory programmed value is unique to each microcontroller produced in the PIC24F16KL402 family. To access this region, use Table Read instructions or Program Space Visibility.

To ensure a globally Unique ID across other Microchip microcontroller families, the “Unique ID” value should be further concatenated with the family and Device ID values stored at address, FF0000h.

### REGISTER 23-8: DEVID: DEVICE ID REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
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 23-16     **Unimplemented:** Read as '0'

bit 15-8     **FAMID<7:0>:** Device Family Identifier bits  
                   01001011 = PIC24F16KL402 family

bit 7-0       **DEV<7:0>:** Individual Device Identifier bits  
                   00000001 = PIC24F04KL100  
                   00000010 = PIC24F04KL101  
                   00000101 = PIC24F08KL200  
                   00000110 = PIC24F08KL201  
                   00001010 = PIC24F08KL301  
                   00000000 = PIC24F08KL302  
                   00001110 = PIC24F08KL401  
                   00000100 = PIC24F08KL402  
                   00011110 = PIC24F16KL401  
                   00010100 = PIC24F16KL402

# PIC24F16KL402 FAMILY

## REGISTER 23-9: DEVREV: DEVICE REVISION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

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	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
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 23-4      **Unimplemented:** Read as '0'

bit 3-0      **REV<3:0>:** Revision Identifier bits

# PIC24F16KL402 FAMILY

## 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 NOSC<sub>x</sub> 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.

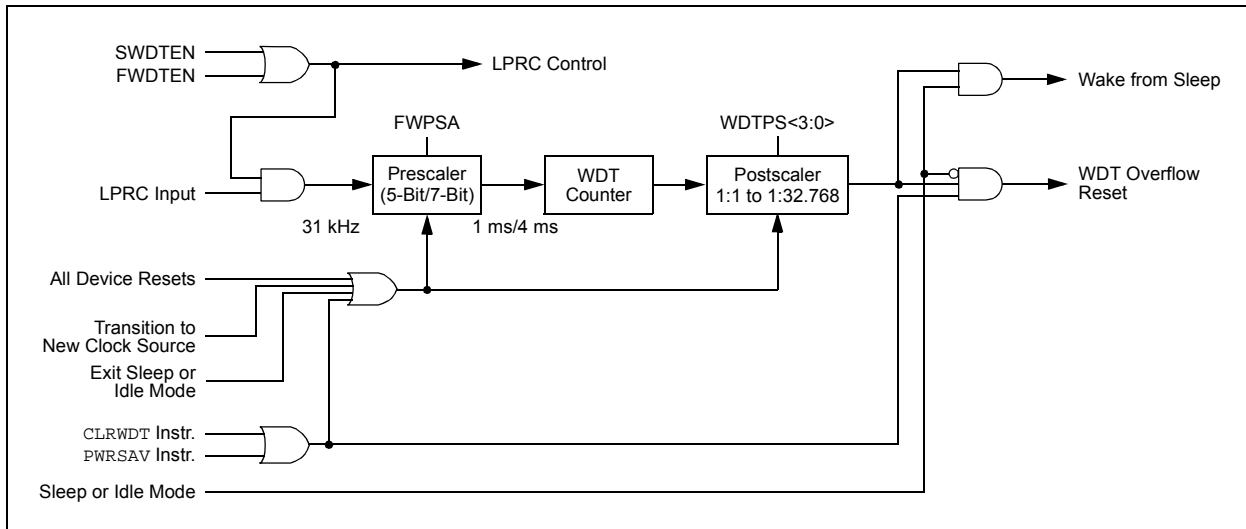
Windowed WDT mode is enabled by programming the Configuration bit, WINDIS (FWDT<6>), to '0'.

### 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**



## 23.5 Program Verification and Code Protection

For all devices in the PIC24F16KL402 family, code protection for the Boot Segment is controlled by the BSS<2:0> Configuration bits and the General Segment by the Configuration bit, GSS0. These bits inhibit external reads and writes to the program memory space. This has no direct effect in normal execution mode.

Write protection is controlled by bit, BWRP, for the Boot Segment and bit, GWRP, for the General Segment in the Configuration Word. When these bits are programmed to '0', internal write and erase operations to program memory are blocked.

## 23.6 In-Circuit Serial Programming

PIC24F16KL402 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock (PGECx) and data (PGEDx), and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

## 23.7 In-Circuit Debugger

When MPLAB® ICD 3, MPLAB REAL ICE™ or PICKit™ 3 is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx and PGEDx pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to MCLR, VDD, VSS, PGECx, PGEDx and the pin pair. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins.

# PIC24F16KL402 FAMILY

---

NOTES:

## 24.0 DEVELOPMENT SUPPORT

The PIC<sup>®</sup> microcontrollers (MCU) and dsPIC<sup>®</sup> digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB<sup>®</sup> X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM<sup>™</sup> Assembler
  - MPLINK<sup>™</sup> Object Linker/  
MPLIB<sup>™</sup> Object Librarian
  - MPLAB Assembler/Linker/Librarian for  
Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE<sup>™</sup> In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICKit<sup>™</sup> 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,  
Evaluation Kits and Starter Kits
- Third-party development tools

## 24.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows<sup>®</sup>, Linux and Mac OS<sup>®</sup> X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

# PIC24F16KL402 FAMILY

---

## 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

# PIC24F16KL402 FAMILY

---

## 24.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 24.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 24.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 24.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 24.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

# PIC24F16KL402 FAMILY

---

## 24.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) for the complete list of demonstration, development and evaluation kits.

## 24.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

## 25.0 INSTRUCTION SET SUMMARY

**Note:** This chapter is a brief summary of the PIC24F Instruction Set Architecture (ISA) and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC<sup>®</sup> MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit 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:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 25-1 lists the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 25-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented *W* register instructions (including barrel shift instructions) have three operands:

- The first source operand, which is typically a register 'Wb' without any address modifier
- The second source operand, which is typically a register 'Ws' with or without an address modifier
- The destination of the result, which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value, 'f'
- The destination, which could either be the file register, 'f', or the *W0* register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The *W* register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the *W* register or file register (specified by a literal value or indirectly by the contents of register, 'Wb')

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a *W* register or file register (specified by the value of 'k')
- The *W* register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand, which is a register 'Wb' without any address modifier
- The second source operand, which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register 'Wd' with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the Table Read and Table Write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all of the required information is available in these 48 bits. In the second word, the 8 MSBs are '0's. If this second word is executed as an instruction (by itself), it will execute as a *NOOP*.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter (PC) 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 *NOOP*. Notable exceptions are the *BRA* (unconditional/computed branch), indirect *CALL/GOTO*, all Table Reads and Table Writes, and *RETURN/RETFIE* instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

# PIC24F16KL402 FAMILY

**TABLE 25-1: SYMBOLS USED IN OPCODE DESCRIPTIONS**

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
bit4	4-bit bit selection field (used in word addressed instructions) $\in \{0...15\}$
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address $\in \{0000h...1FFFh\}$
lit1	1-bit unsigned literal $\in \{0,1\}$
lit4	4-bit unsigned literal $\in \{0...15\}$
lit5	5-bit unsigned literal $\in \{0...31\}$
lit8	8-bit unsigned literal $\in \{0...255\}$
lit10	10-bit unsigned literal $\in \{0...255\}$ for Byte mode, $\{0:1023\}$ for Word mode
lit14	14-bit unsigned literal $\in \{0...16384\}$
lit16	16-bit unsigned literal $\in \{0...65535\}$
lit23	23-bit unsigned literal $\in \{0...8388608\}$ ; LSB must be '0'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal $\in \{-512...511\}$
Slit16	16-bit signed literal $\in \{-32768...32767\}$
Slit6	6-bit signed literal $\in \{-16...16\}$
Wb	Base W register $\in \{W0..W15\}$
Wd	Destination W register $\in \{Wd, [Wd], [Wd++] , [Wd--], [++Wd], [--Wd] \}$
Wdo	Destination W register $\in \{Wnd, [Wnd], [Wnd++] , [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] \}$
Wm,Wn	Dividend, Divisor Working register pair (direct addressing)
Wn	One of 16 Working registers $\in \{W0..W15\}$
Wnd	One of 16 destination Working registers $\in \{W0..W15\}$
Wns	One of 16 source Working registers $\in \{W0..W15\}$
WREG	W0 (Working register used in File register instructions)
Ws	Source W register $\in \{Ws, [Ws], [Ws++] , [Ws--], [++Ws], [--Ws] \}$
Wso	Source W register $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb] \}$

# PIC24F16KL402 FAMILY

**TABLE 25-2: INSTRUCTION SET OVERVIEW**

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
ADD	ADD f	$f = f + WREG$	1	1	C, DC, N, OV, Z
	ADD f, WREG	$WREG = f + WREG$	1	1	C, DC, N, OV, Z
	ADD #lit10, Wn	$Wd = lit10 + Wd$	1	1	C, DC, N, OV, Z
	ADD Wb, Ws, Wd	$Wd = Wb + Ws$	1	1	C, DC, N, OV, Z
	ADD Wb, #lit5, Wd	$Wd = Wb + lit5$	1	1	C, DC, N, OV, Z
ADDC	ADDC f	$f = f + WREG + (C)$	1	1	C, DC, N, OV, Z
	ADDC f, WREG	$WREG = f + WREG + (C)$	1	1	C, DC, N, OV, Z
	ADDC #lit10, Wn	$Wd = lit10 + Wd + (C)$	1	1	C, DC, N, OV, Z
	ADDC Wb, Ws, Wd	$Wd = Wb + Ws + (C)$	1	1	C, DC, N, OV, Z
	ADDC Wb, #lit5, Wd	$Wd = Wb + lit5 + (C)$	1	1	C, DC, N, OV, Z
AND	AND f	$f = f .AND. WREG$	1	1	N, Z
	AND f, WREG	$WREG = f .AND. WREG$	1	1	N, Z
	AND #lit10, Wn	$Wd = lit10 .AND. Wd$	1	1	N, Z
	AND Wb, Ws, Wd	$Wd = Wb .AND. Ws$	1	1	N, Z
	AND Wb, #lit5, Wd	$Wd = Wb .AND. lit5$	1	1	N, Z
ASR	ASR f	f = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR f, WREG	WREG = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR Ws, Wd	Wd = Arithmetic Right Shift Ws	1	1	C, N, OV, Z
	ASR Wb, Wns, Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N, Z
	ASR Wb, #lit5, Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N, Z
BCLR	BCLR f, #bit4	Bit Clear f	1	1	None
	BCLR Ws, #bit4	Bit Clear Ws	1	1	None
BRA	BRA C, Expr	Branch if Carry	1	1 (2)	None
	BRA GE, Expr	Branch if Greater than or Equal	1	1 (2)	None
	BRA GEU, Expr	Branch if Unsigned Greater than or Equal	1	1 (2)	None
	BRA GT, Expr	Branch if Greater than	1	1 (2)	None
	BRA GTU, Expr	Branch if Unsigned Greater than	1	1 (2)	None
	BRA LE, Expr	Branch if Less than or Equal	1	1 (2)	None
	BRA LEU, Expr	Branch if Unsigned Less than or Equal	1	1 (2)	None
	BRA LT, Expr	Branch if Less than	1	1 (2)	None
	BRA LTU, Expr	Branch if Unsigned Less than	1	1 (2)	None
	BRA N, Expr	Branch if Negative	1	1 (2)	None
	BRA NC, Expr	Branch if Not Carry	1	1 (2)	None
	BRA NN, Expr	Branch if Not Negative	1	1 (2)	None
	BRA NOV, Expr	Branch if Not Overflow	1	1 (2)	None
	BRA NZ, Expr	Branch if Not Zero	1	1 (2)	None
	BRA OV, Expr	Branch if Overflow	1	1 (2)	None
	BRA Expr	Branch Unconditionally	1	2	None
	BRA Z, Expr	Branch if Zero	1	1 (2)	None
BRA Wn	Computed Branch	1	2	None	
BSET	BSET f, #bit4	Bit Set f	1	1	None
	BSET Ws, #bit4	Bit Set Ws	1	1	None
BSW	BSW.C Ws, Wb	Write C bit to Ws<Wb>	1	1	None
	BSW.Z Ws, Wb	Write Z bit to Ws<Wb>	1	1	None
BTG	BTG f, #bit4	Bit Toggle f	1	1	None
	BTG Ws, #bit4	Bit Toggle Ws	1	1	None
BTSC	BTSC f, #bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
	BTSC Ws, #bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

# PIC24F16KL402 FAMILY

**TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS $f, \#bit4$	Bit Test $f$ , Skip if Set	1	1 (2 or 3)	None
	BTSS $Ws, \#bit4$	Bit Test $Ws$ , Skip if Set	1	1 (2 or 3)	None
BTST	BTST $f, \#bit4$	Bit Test $f$	1	1	Z
	BTST.C $Ws, \#bit4$	Bit Test $Ws$ to C	1	1	C
	BTST.Z $Ws, \#bit4$	Bit Test $Ws$ to Z	1	1	Z
	BTST.C $Ws, Wb$	Bit Test $Ws < Wb >$ to C	1	1	C
	BTST.Z $Ws, Wb$	Bit Test $Ws < Wb >$ to Z	1	1	Z
BTSTS	BTSTS $f, \#bit4$	Bit Test then Set $f$	1	1	Z
	BTSTS.C $Ws, \#bit4$	Bit Test $Ws$ to C, then Set	1	1	C
	BTSTS.Z $Ws, \#bit4$	Bit Test $Ws$ to Z, then Set	1	1	Z
CALL	CALL $lit23$	Call Subroutine	2	2	None
	CALL $Wn$	Call Indirect Subroutine	1	2	None
CLR	CLR $f$	$f = 0x0000$	1	1	None
	CLR $WREG$	$WREG = 0x0000$	1	1	None
	CLR $Ws$	$Ws = 0x0000$	1	1	None
CLRWDT	CLRWDT	Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM $f$	$f = \bar{f}$	1	1	N, Z
	COM $f, WREG$	$WREG = \bar{f}$	1	1	N, Z
	COM $Ws, Wd$	$Wd = \bar{Ws}$	1	1	N, Z
CP	CP $f$	Compare $f$ with $WREG$	1	1	C, DC, N, OV, Z
	CP $Wb, \#lit5$	Compare $Wb$ with $lit5$	1	1	C, DC, N, OV, Z
	CP $Wb, Ws$	Compare $Wb$ with $Ws$ ( $Wb - Ws$ )	1	1	C, DC, N, OV, Z
CP0	CP0 $f$	Compare $f$ with $0x0000$	1	1	C, DC, N, OV, Z
	CP0 $Ws$	Compare $Ws$ with $0x0000$	1	1	C, DC, N, OV, Z
CPB	CPB $f$	Compare $f$ with $WREG$ , with Borrow	1	1	C, DC, N, OV, Z
	CPB $Wb, \#lit5$	Compare $Wb$ with $lit5$ , with Borrow	1	1	C, DC, N, OV, Z
	CPB $Wb, Ws$	Compare $Wb$ with $Ws$ , with Borrow ( $Wb - Ws - C$ )	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ $Wb, Wn$	Compare $Wb$ with $Wn$ , Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT $Wb, Wn$	Compare $Wb$ with $Wn$ , Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT $Wb, Wn$	Compare $Wb$ with $Wn$ , Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE $Wb, Wn$	Compare $Wb$ with $Wn$ , Skip if $\neq$	1	1 (2 or 3)	None
DAW	DAW.B $Wn$	$Wn =$ Decimal Adjust $Wn$	1	1	C
DEC	DEC $f$	$f = f - 1$	1	1	C, DC, N, OV, Z
	DEC $f, WREG$	$WREG = f - 1$	1	1	C, DC, N, OV, Z
	DEC $Ws, Wd$	$Wd = Ws - 1$	1	1	C, DC, N, OV, Z
DEC2	DEC2 $f$	$f = f - 2$	1	1	C, DC, N, OV, Z
	DEC2 $f, WREG$	$WREG = f - 2$	1	1	C, DC, N, OV, Z
	DEC2 $Ws, Wd$	$Wd = Ws - 2$	1	1	C, DC, N, OV, Z
DISI	DISI $\#lit14$	Disable Interrupts for $k$ Instruction Cycles	1	1	None
DIV	DIV.SW $Wm, Wn$	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD $Wm, Wn$	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW $Wm, Wn$	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD $Wm, Wn$	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH $Wns, Wnd$	Swap $Wns$ with $Wnd$	1	1	None
FF1L	FF1L $Ws, Wnd$	Find First One from Left (MSb) Side	1	1	C
FF1R	FF1R $Ws, Wnd$	Find First One from Right (LSb) Side	1	1	C

# PIC24F16KL402 FAMILY

**TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO Expr	Go to Address	2	2	None
	GOTO Wn	Go to Indirect	1	2	None
INC	INC f	$f = f + 1$	1	1	C, DC, N, OV, Z
	INC f, WREG	WREG = $f + 1$	1	1	C, DC, N, OV, Z
	INC Ws, Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2 f	$f = f + 2$	1	1	C, DC, N, OV, Z
	INC2 f, WREG	WREG = $f + 2$	1	1	C, DC, N, OV, Z
	INC2 Ws, Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR f	$f = f .IOR. WREG$	1	1	N, Z
	IOR f, WREG	WREG = $f .IOR. WREG$	1	1	N, Z
	IOR #lit10, Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR Wb, Ws, Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR Wb, #lit5, Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK #lit14	Link Frame Pointer	1	1	None
LSR	LSR f	$f = \text{Logical Right Shift } f$	1	1	C, N, OV, Z
	LSR f, WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR Ws, Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR Wb, Wns, Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR Wb, #lit5, Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV f, Wn	Move f to Wn	1	1	None
	MOV [Wns+Slit10], Wnd	Move [Wns+Slit10] to Wnd	1	1	None
	MOV f	Move f to f	1	1	N, Z
	MOV f, WREG	Move f to WREG	1	1	None
	MOV #lit16, Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b #lit8, Wn	Move 8-bit Literal to Wn	1	1	None
	MOV Wn, f	Move Wn to f	1	1	None
	MOV Wns, [Wns+Slit10]	Move Wns to [Wns+Slit10]	1	1	None
	MOV Wso, Wdo	Move Ws to Wd	1	1	None
	MOV WREG, f	Move WREG to f	1	1	None
	MOV.D Wns, Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
	MOV.D Ws, Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
MUL	MUL.SS Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{Signed}(Wb) * \text{Signed}(Ws)$	1	1	None
	MUL.SU Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{Signed}(Wb) * \text{Unsigned}(Ws)$	1	1	None
	MUL.US Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{Unsigned}(Wb) * \text{Signed}(Ws)$	1	1	None
	MUL.UU Wb, Ws, Wnd	$\{Wnd+1, Wnd\} = \text{Unsigned}(Wb) * \text{Unsigned}(Ws)$	1	1	None
	MUL.SU Wb, #lit5, Wnd	$\{Wnd+1, Wnd\} = \text{Signed}(Wb) * \text{Unsigned}(lit5)$	1	1	None
	MUL.UU Wb, #lit5, Wnd	$\{Wnd+1, Wnd\} = \text{Unsigned}(Wb) * \text{Unsigned}(lit5)$	1	1	None
	MUL f	W3:W2 = $f * WREG$	1	1	None
NEG	NEG f	$f = \bar{f} + 1$	1	1	C, DC, N, OV, Z
	NEG f, WREG	WREG = $\bar{f} + 1$	1	1	C, DC, N, OV, Z
	NEG Ws, Wd	Wd = $\overline{Ws} + 1$	1	1	C, DC, N, OV, Z
NOP	NOP	No Operation	1	1	None
	NOPR	No Operation	1	1	None
POP	POP f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd+1)	1	2	None
	POP.S	Pop Shadow Registers	1	1	All
PUSH	PUSH f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D Wns	Push W(ns):W(ns+1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S	Push Shadow Registers	1	1	None

# PIC24F16KL402 FAMILY

**TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
PWRSVAV	PWRSVAV #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

# PIC24F16KL402 FAMILY

**TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
TBLRDH	TBLRDH <i>Ws, Wd</i>	Read Prog<23:16> to Wd<7:0>	1	2	None
TBLRDL	TBLRDL <i>Ws, Wd</i>	Read Prog<15:0> to Wd	1	2	None
TBLWTH	TBLWTH <i>Ws, Wd</i>	Write Ws<7:0> to Prog<23:16>	1	2	None
TBLWTL	TBLWTL <i>Ws, Wd</i>	Write Ws to Prog<15:0>	1	2	None
ULNK	ULNK	Unlink Frame Pointer	1	1	None
XOR	XOR <i>f</i>	$f = f \text{ .XOR. WREG}$	1	1	N, Z
	XOR <i>f, WREG</i>	$WREG = f \text{ .XOR. WREG}$	1	1	N, Z
	XOR <i>#lit10, Wn</i>	$Wd = \text{lit}10 \text{ .XOR. Wd}$	1	1	N, Z
	XOR <i>Wb, Ws, Wd</i>	$Wd = Wb \text{ .XOR. Ws}$	1	1	N, Z
	XOR <i>Wb, #lit5, Wd</i>	$Wd = Wb \text{ .XOR. lit}5$	1	1	N, Z
ZE	ZE <i>Ws, Wnd</i>	Wnd = Zero-Extend Ws	1	1	C, Z, N

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## 26.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24F16KL402 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24F16KL402 family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

### Absolute Maximum Ratings<sup>(†)</sup>

Ambient temperature under bias .....	-40°C to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to VSS .....	-0.3V to +4.5V
Voltage on any combined analog and digital pin, with respect to VSS .....	-0.3V to (VDD + 0.3V)
Voltage on any digital only pin with respect to VSS .....	-0.3V to (VDD + 0.3V)
Voltage on $\overline{\text{MCLR}}$ /VPP pin with respect to VSS .....	-0.3V to +9.0V
Maximum current out of VSS pin .....	300 mA
Maximum current into VDD pin <sup>(1)</sup> .....	250 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by all ports .....	200 mA
Maximum current sourced by all ports <sup>(1)</sup> .....	200 mA

**Note 1:** Maximum allowable current is a function of device maximum power dissipation (see [Table 26-1](#)).

† **Notice:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

# PIC24F16KL402 FAMILY

## 26.1 DC Characteristics

FIGURE 26-1: PIC24F16KL402 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)

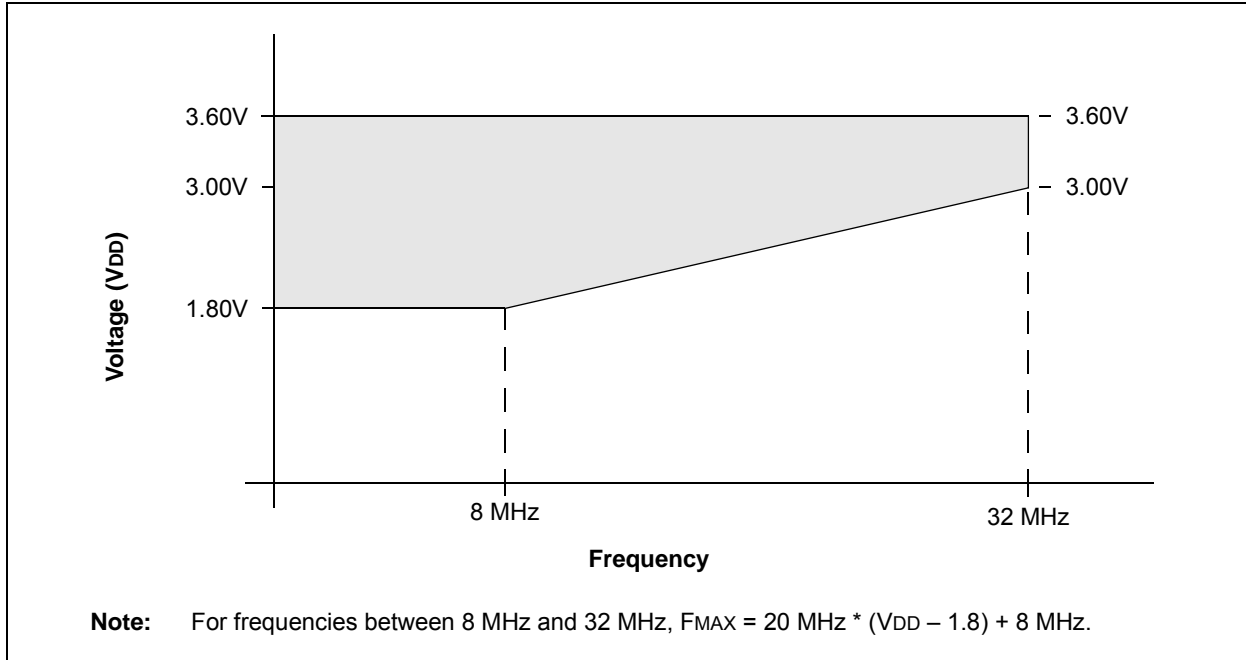
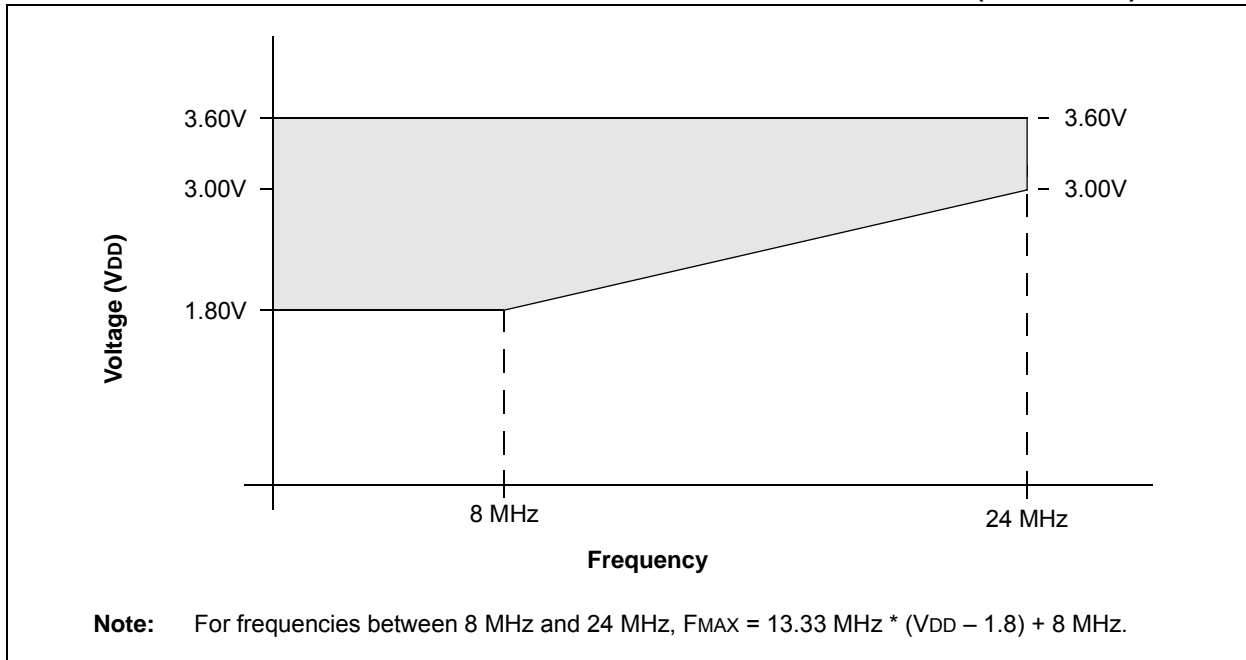


FIGURE 26-2: PIC24F16KL402 FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED)



# PIC24F16KL402 FAMILY

**TABLE 26-1: THERMAL OPERATING CONDITIONS**

Rating	Symbol	Min	Typ	Max	Unit
Operating Junction Temperature Range	TJ	-40	—	+140	°C
Operating Ambient Temperature Range	TA	-40	—	+125	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O Pin Power Dissipation: $P_{I/O} = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	PD	P <sub>INT</sub> + P <sub>I/O</sub>			W
Maximum Allowed Power Dissipation	PD <sub>MAX</sub>	$(T_J - T_A)/\theta_{JA}$			W

**TABLE 26-2: THERMAL PACKAGING CHARACTERISTICS**

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 20-Pin PDIP	$\theta_{JA}$	62.4	—	°C/W	1
Package Thermal Resistance, 28-Pin SPDIP	$\theta_{JA}$	60	—	°C/W	1
Package Thermal Resistance, 20-Pin SSOP	$\theta_{JA}$	108	—	°C/W	1
Package Thermal Resistance, 28-Pin SSOP	$\theta_{JA}$	71	—	°C/W	1
Package Thermal Resistance, 20-Pin SOIC	$\theta_{JA}$	75	—	°C/W	1
Package Thermal Resistance, 28-Pin SOIC	$\theta_{JA}$	80.2	—	°C/W	1
Package Thermal Resistance, 20-Pin QFN	$\theta_{JA}$	43	—	°C/W	1
Package Thermal Resistance, 28-Pin QFN	$\theta_{JA}$	32	—	°C/W	1
Package Thermal Resistance, 14-Pin PDIP	$\theta_{JA}$	62.4	—	°C/W	1
Package Thermal Resistance, 14-Pin TSSOP	$\theta_{JA}$	108	—	°C/W	1

**Note 1:** Junction to ambient thermal resistance, Theta-JA ( $\theta_{JA}$ ) numbers are achieved by package simulations.

**TABLE 26-3: DC CHARACTERISTICS: TEMPERATURE AND VOLTAGE SPECIFICATIONS**

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				
Para m No.	Symbol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
DC10	V <sub>DD</sub>	<b>Supply Voltage</b>	1.8	—	3.6	V	
DC12	V <sub>DR</sub>	<b>RAM Data Retention Voltage<sup>(2)</sup></b>	1.5	—	—	V	
DC16	V <sub>POR</sub>	<b>V<sub>DD</sub> Start Voltage</b> to Ensure Internal Power-on Reset Signal	V <sub>SS</sub>	—	0.7	V	
DC17	SV <sub>DD</sub>	<b>V<sub>DD</sub> Rise Rate</b> to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 0.1s 0-2.5V in 60 ms
	V <sub>BG</sub>	<b>Band Gap Voltage Reference</b>	1.14	1.2	1.26	V	

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**2:** This is the limit to which V<sub>DD</sub> can be lowered without losing RAM data.

# PIC24F16KL402 FAMILY

**TABLE 26-4: HIGH/LOW-VOLTAGE DETECT CHARACTERISTICS**

Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended								
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions	
DC18	VHLVD	HLVD Voltage on VDD Transition	HLVDL<3:0> = 0000	—	1.85	1.94	V	
			HLVDL<3:0> = 0001	1.81	1.90	2.00	V	
			HLVDL<3:0> = 0010	1.85	1.95	2.05	V	
			HLVDL<3:0> = 0011	1.90	2.00	2.10	V	
			HLVDL<3:0> = 0100	1.95	2.05	2.15	V	
			HLVDL<3:0> = 0101	2.06	2.17	2.28	V	
			HLVDL<3:0> = 0110	2.12	2.23	2.34	V	
			HLVDL<3:0> = 0111	2.24	2.36	2.48	V	
			HLVDL<3:0> = 1000	2.31	2.43	2.55	V	
			HLVDL<3:0> = 1001	2.47	2.60	2.73	V	
			HLVDL<3:0> = 1010	2.64	2.78	2.92	V	
			HLVDL<3:0> = 1011	2.74	2.88	3.02	V	
			HLVDL<3:0> = 1100	2.85	3.00	3.15	V	
			HLVDL<3:0> = 1101	2.96	3.12	3.28	V	
HLVDL<3:0> = 1110	3.22	3.39	3.56	V				

**TABLE 26-5: BOR TRIP POINTS**

Standard Operating Conditions: 1.8V to 3.6V Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended								
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions	
DC19		BOR Voltage on VDD Transition	BORV = 00	1.85	2.0	2.15	V	<b>Note 1</b>
			BORV = 01	2.90	3.0	3.38	V	
			BORV = 10	2.53	2.7	3.07	V	
			BORV = 11	1.75	1.85	2.05	V	

**Note 1:** LPBOR re-arms the POR circuit but does not cause a BOR.

# PIC24F16KL402 FAMILY

**TABLE 26-6: DC CHARACTERISTICS: OPERATING CURRENT (IDD)<sup>(2)</sup>**

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 <sup>(1)</sup>	Max	Units	Conditions		
<b>IDD Current</b>						
DC20	0.154	0.350	mA	1.8V	+85°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 (IDLE)<sup>(2)</sup>**

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 <sup>(1)</sup>	Max	Units	Conditions		
<b>Idle Current (IDLE)</b>						
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		3.3V	+125°C	
DC46	0.45	—	mA	1.8V	+85°C	FRC (4 MIPS), Fosc = 8 MHz
	0.76	—		3.3V		
DC50	0.87	18.0	µA	1.8V	+85°C	LPRC (15.5 KIPS), Fosc = 31 kHz
	1.55	40.0		3.3V		
	—	27.0	µA	1.8V	+125°C	
	—	50.0		3.3V		

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

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

# PIC24F16KL402 FAMILY

**TABLE 26-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)**

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 <sup>(1)</sup>	Max	Units	Conditions		
<b>Power-Down Current (IPD)</b>						
DC60	0.01	0.20	μA	-40°C	1.8V	Sleep Mode <sup>(2)</sup>
	0.03	0.20	μA	+25°C		
	0.06	0.87	μA	+60°C		
	0.20	1.35	μA	+85°C		
	—	8.00	μA	+125°C		
	0.01	0.54	μA	-40°C	3.3V	
	0.03	0.54	μA	+25°C		
	0.08	1.68	μA	+60°C		
	0.25	2.45	μA	+85°C		
	—	10.00	μA	+125°C		

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

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

# PIC24F16KL402 FAMILY

**TABLE 26-9: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)**

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 <sup>(1)</sup>	Max	Units	Conditions		
<b>Module Differential Current (<math>\Delta</math>IPD)</b>						
DC71	0.21	0.65	$\mu$ A	1.8V	+85°C	Watchdog Timer Current: $\Delta$ WDT <sup>(2,3)</sup>
	0.45	0.95	$\mu$ A	3.3V		
	—	1.30	$\mu$ A	1.8V	+125°C	
	—	1.50	$\mu$ A	3.3V		
DC72	0.69	1.50	$\mu$ A	1.8V	+85°C	32 kHz Crystal with Timer1: $\Delta$ SOSC (SOSCSEL = 0) <sup>(2)</sup>
	1.00	1.50	$\mu$ A	3.3V		
DC75	5.24	—	$\mu$ A	1.8V	+85°C	$\Delta$ HLVD <sup>(2,3)</sup>
	5.16	11.00	$\mu$ A	3.3V		
	—	12.00	$\mu$ A	1.8V	+125°C	
	—	15.00	$\mu$ A	3.3V		
DC76	4.15	9.00	$\mu$ A	3.3V	+85°C	$\Delta$ BOR <sup>(2,3)</sup>
	—	11.0	$\mu$ A	3.3V		
DC78	0.03	0.20	$\mu$ A	1.8V	+85°C	$\Delta$ LPBOR <sup>(2)</sup>
	0.03	0.20	$\mu$ A	3.3V		
	—	0.40	$\mu$ A	1.8V	+125°C	
	—	0.40	$\mu$ A	3.3V		

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

**2:** The  $\Delta$  current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

**3:** This current applies to Sleep only.

# PIC24F16KL402 FAMILY

TABLE 26-10: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

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				
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
DI10 DI15 DI16 DI17 DI18 DI19	V <sub>IL</sub>	<b>Input Low Voltage<sup>(4)</sup></b>					
		I/O Pins	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
		$\overline{\text{MCLR}}$	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
		OSCI (XT mode)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
		OSCI (HS mode)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
		I/O Pins with I <sup>2</sup> C™ Buffer	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	SMBus disabled
I/O Pins with SMBus Buffer	V <sub>SS</sub>	—	0.8	V	SMBus enabled		
DI20 DI25 DI26 DI27 DI28 DI29	V <sub>IH</sub>	<b>Input High Voltage<sup>(4,5)</sup></b>					
		I/O Pins:					
		with Analog Functions	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
		Digital Only	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
		$\overline{\text{MCLR}}$	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
		OSCI (XT mode)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
		OSCI (HS mode)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
		I/O Pins with I <sup>2</sup> C Buffer:					
with Analog Functions	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V			
Digital Only	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V			
I/O Pins with SMBus	2.1	—	V <sub>DD</sub>	V	2.5V ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>		
DI30	IC <sub>NPU</sub>	<b>CNx Pull-up Current</b>	50	250	500	μA	V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub>
DI31	IPU	<b>Maximum Load Current for Digital High Detection w/Internal Pull-up</b>	—	—	30	μA	V <sub>DD</sub> = 2.0V
			—	—	1000	μA	V <sub>DD</sub> = 3.3V
DI50 DI51	I <sub>IL</sub>	<b>Input Leakage Current<sup>(2,3)</sup></b>					
		I/O Ports	—	0.050	±0.100	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at high-impedance
		V <sub>REF+</sub> , V <sub>REF-</sub> , AN0, AN1	—	0.300	±0.500	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at high-impedance

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

**2:** The leakage current on the  $\overline{\text{MCLR}}$  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.

**3:** Negative current is defined as current sourced by the pin.

**4:** Refer to [Table 1-4](#) and [Table 1-5](#) for I/O pin buffer types.

**5:** V<sub>IH</sub> requirements are met when the internal pull-ups are enabled.

# PIC24F16KL402 FAMILY

**TABLE 26-11: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS**

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					
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions	
DO10	VOL	<b>Output Low Voltage</b> All I/O Pins	—	—	0.4	V	IOL = 4.0 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 3.5 mA	VDD = 2.0V
DO16		OSC2/CLKO	—	—	0.4	V	IOL = 1.2 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 0.4 mA	VDD = 2.0V
DO20	VOH	<b>Output High Voltage</b> All I/O Pins	3	—	—	V	IOH = -3.0 mA	VDD = 3.6V
			1.6	—	—	V	IOH = -1.0 mA	VDD = 2.0V
DO26		OSC2/CLKO	3	—	—	V	IOH = -1.0 mA	VDD = 3.6V
			1.6	—	—	V	IOH = -0.5 mA	VDD = 2.0V

**Note 1:** Data in “Typ” column is at +25°C unless otherwise stated.

**TABLE 26-12: DC CHARACTERISTICS: PROGRAM MEMORY**

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					
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions	
<b>Program Flash Memory</b>								
D130	EP	Cell Endurance	10,000 <sup>(2)</sup>	—	—	E/W	VMIN = Minimum operating voltage  Provided no other specifications are violated	
D131	VPR	VDD for Read	VMIN	—	3.6	V		
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms		
D134	TRETD	Characteristic Retention	40	—	—	Year		
D135	IDDP	Supply Current During Programming	—	10	—	mA		

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

**2:** Self-write and block erase.

# PIC24F16KL402 FAMILY

**TABLE 26-13: DC CHARACTERISTICS: DATA EEPROM MEMORY**

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					
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions	
		<b>Data EEPROM Memory</b>						
D140	EPD	Cell Endurance	100,000	—	—	E/W	V <sub>MIN</sub> = Minimum operating voltage  Provided no other specifications are violated	
D141	VPRD	V <sub>DD</sub> for Read	V <sub>MIN</sub>	—	3.6	V		
D143A	TiWD	Self-Timed Write Cycle Time	—	4	—	ms		
D143B	TREF	Number of Total Write/Erase Cycles Before Refresh	—	10M	—	E/W		
D144	TRETDD	Characteristic Retention	40	—	—	Year		
D145	IDDPD	Supply Current during Programming	—	7	—	mA		

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

**TABLE 26-14: DC CHARACTERISTICS: COMPARATOR**

Standard Operating Conditions: 2.0V < V <sub>DD</sub> < 3.6V Operating temperature -40°C < TA ≤ +85°C (unless otherwise stated) -40°C ≤ TA ≤ +125°C for Extended							
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
D300	V <sub>IOFF</sub>	Input Offset Voltage	—	20	40	mV	
D301	V <sub>ICM</sub>	Input Common-Mode Voltage	0	—	V <sub>DD</sub>	V	
D302	CMRR	Common-Mode Rejection Ratio	55	—	—	dB	

**TABLE 26-15: DC CHARACTERISTICS: COMPARATOR VOLTAGE REFERENCE**

Standard Operating Conditions: 2.0V < V <sub>DD</sub> < 3.6V Operating temperature -40°C < TA ≤ +85°C (unless otherwise stated) -40°C ≤ TA ≤ +125°C for Extended							
Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
VRD310	CVRES	Resolution	—	—	V <sub>DD</sub> /32	LSb	
VRD311	CVR <sub>AA</sub>	Absolute Accuracy	—	—	A <sub>VDD</sub> - 1.5	LSb	
VRD312	CVR <sub>UR</sub>	Unit Resistor Value (R)	—	2k	—	Ω	

# PIC24F16KL402 FAMILY

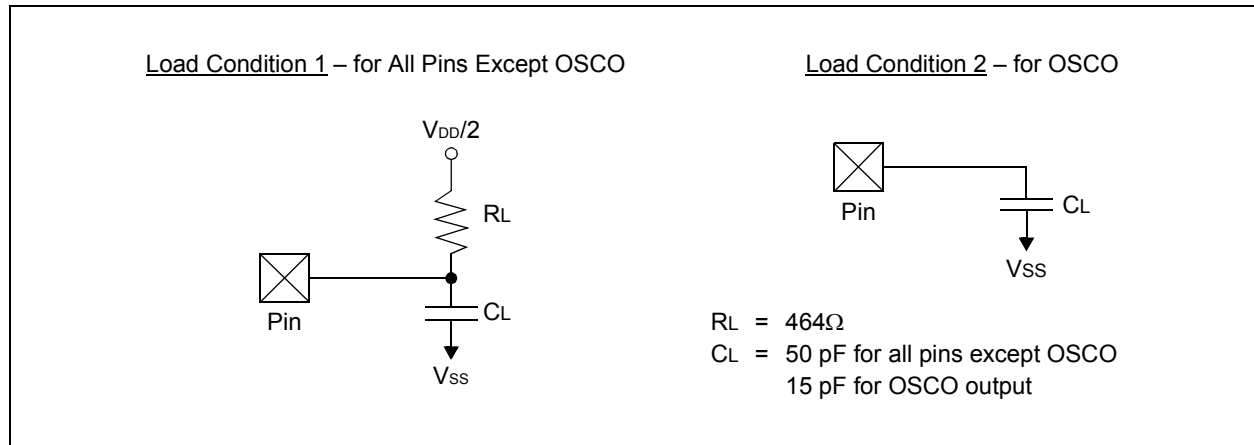
## 26.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24F16KL402 Family AC characteristics and timing parameters.

**TABLE 26-16: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC**

AC CHARACTERISTICS	Standard Operating Conditions: 1.8V to 3.6V
	Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial
	Operating voltage $V_{DD}$ range as described in <a href="#">Section 26.1 “DC Characteristics”</a> .

**FIGURE 26-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS**



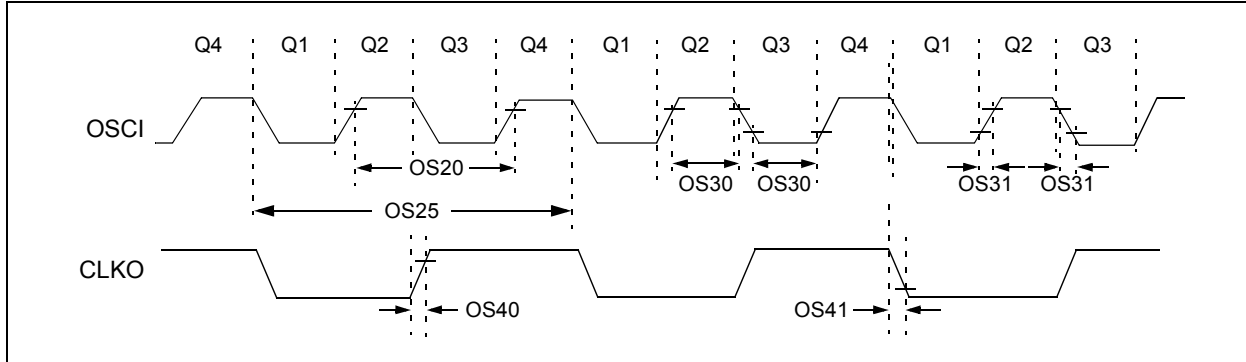
**TABLE 26-17: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS**

Param No.	Symbol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
DO50	Cosc2	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when external clock is used to drive OSC1
DO56	Cio	All I/O Pins and OSCO	—	—	50	pF	EC mode
DO58	CB	SCLx, SDAx	—	—	400	pF	In I <sup>2</sup> C™ mode

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

# PIC24F16KL402 FAMILY

**FIGURE 26-4: EXTERNAL CLOCK TIMING**



**TABLE 26-18: EXTERNAL CLOCK TIMING REQUIREMENTS**

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended				
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
OS10	Fosc	External CLKI Frequency (External clocks allowed only in EC mode)	DC 4	— —	32 8	MHz MHz	EC ECPLL
		Oscillator Frequency	0.2 4 4 31	— — — —	4 25 8 33	MHz MHz MHz kHz	XT HS HSPLL SOSC
OS20	Tosc	$T_{osc} = 1/F_{osc}$	—	—	—	—	See Parameter OS10 for Fosc value
OS25	Tcy	Instruction Cycle Time <sup>(2)</sup>	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	$0.45 \times T_{osc}$	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time <sup>(3)</sup>	—	6	10	ns	
OS41	TckF	CLKO Fall Time <sup>(3)</sup>	—	6	10	ns	

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**2:** Instruction cycle period ( $T_{cy}$ ) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “Min.” values with an external clock applied to the OSCI/CLKI pin. When an external clock input is used, the “Max.” cycle time limit is “DC” (no clock) for all devices.

**3:** Measurements are taken in EC mode. The CLKO signal is measured on the OSCO pin. CLKO is low for the Q1-Q2 period ( $1/2 T_{cy}$ ) and high for the Q3-Q4 period ( $1/2 T_{cy}$ ).

# PIC24F16KL402 FAMILY

**TABLE 26-19: PLL CLOCK TIMING SPECIFICATIONS**

AC 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				
Param No.	Sym	Characteristic <sup>(1)</sup>	Min	Typ <sup>(2)</sup>	Max	Units	Conditions
OS50	FPLLI	PLL Input Frequency Range	4	—	8	MHz	ECPLL, HSPLL modes, -40°C ≤ TA ≤ +85°C
OS51	Fsys	PLL Output Frequency Range	16	—	32	MHz	-40°C ≤ TA ≤ +85°C
OS52	TLOCK	PLL Start-up Time (Lock Time)	—	1	2	ms	
OS53	DCLK	CLKO Stability (Jitter)	-2	1	2	%	Measured over 100 ms period

**Note 1:** These parameters are characterized but not tested in manufacturing.

**Note 2:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

**TABLE 26-20: INTERNAL RC OSCILLATOR ACCURACY**

AC 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				
Param No.	Characteristic	Min	Typ	Max	Units	Conditions
F20	FRC @ 8 MHz <sup>(1)</sup>	-2	—	+2	%	+25°C, 3.0V ≤ VDD ≤ 3.6V
		-5	—	+5	%	-40°C ≤ TA ≤ +85°C, 1.8V ≤ VDD ≤ 3.6V
		-10	—	+10	%	-40°C ≤ TA ≤ +125°C, 1.8V ≤ VDD ≤ 3.6V
F21	LPRC @ 31 kHz <sup>(2)</sup>	-15	—	+15	%	-40°C ≤ TA ≤ +85°C, 1.8V ≤ VDD ≤ 3.6V
		-25	—	+25	%	-40°C ≤ TA ≤ +125°C, 1.8V ≤ VDD ≤ 3.6V

**Note 1:** The frequency is calibrated at +25°C and 3.3V. The OSCTUN bits can be used to compensate for temperature drift.

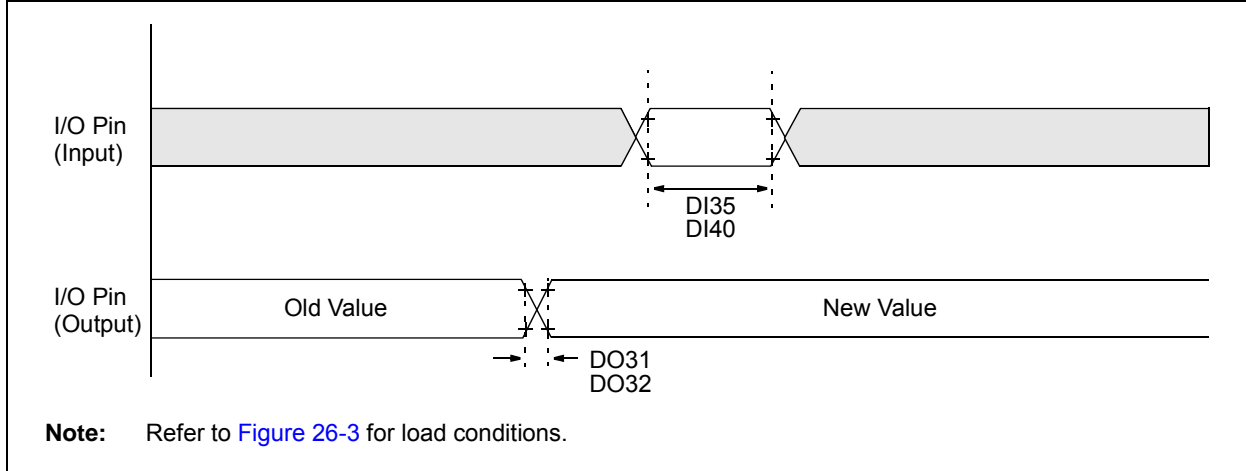
**Note 2:** The change of LPRC frequency as VDD changes.

**TABLE 26-21: INTERNAL RC OSCILLATOR SPECIFICATIONS**

AC 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				
Param No.	Sym	Characteristic	Min	Typ	Max	Units	Conditions
	TFRC	FRC Start-up Time	—	5	—	μs	
	TLPRC	LPRC Start-up Time	—	70	—	μs	

# PIC24F16KL402 FAMILY

**FIGURE 26-5: CLKO AND I/O TIMING CHARACTERISTICS**



**TABLE 26-22: CLKO AND I/O TIMING REQUIREMENTS**

AC 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		
Param No.	Sym	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
DO31	TioR	Port Output Rise Time	—	10	25	ns	
DO32	TioF	Port Output Fall Time	—	10	25	ns	
DI35	TINP	INTx pin High or Low Time (output)	20	—	—	ns	
DI40	TRBP	CNx High or Low Time (input)	2	—	—	Tcy	

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

# PIC24F16KL402 FAMILY

**TABLE 26-23: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET TIMING REQUIREMENTS**

AC 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				
Param No.	Symbol	Characteristic	Min.	Typ <sup>(1)</sup>	Max.	Units	Conditions
SY10	TmCL	MCLR Pulse Width (low)	2	—	—	μs	
SY11	TPWRT	Power-up Timer Period	50	64	90	ms	
SY12	TPOR	Power-on Reset Delay	1	5	10	μs	
SY13	TIOZ	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	—	100	ns	
SY20	TWDT	Watchdog Timer Time-out Period	0.85	1.0	1.15	ms	1.32 prescaler
			3.4	4.0	4.6	ms	1:128 prescaler
SY25	TBOR	Brown-out Reset Pulse Width	1	—	—	μs	
SY45	TRST	Internal State Reset Time	—	5	—	μs	
SY55	TLOCK	PLL Start-up Time	—	100	—	μs	
SY65	TOST	Oscillator Start-up Time	—	1024	—	TOSC	
SY71	TPM	Program Memory Wake-up Time	—	1	—	μs	Sleep wake-up with PMSLP = 0

**Note 1:** Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.

**TABLE 26-24: COMPARATOR TIMINGS**

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
300	TRESP	Response Time <sup>(1,2)</sup>	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid <sup>(2)</sup>	—	—	10	μs	

**Note 1:** Response time is measured with one comparator input at  $(V_{DD} - 1.5)/2$ , while the other input transitions from VSS to VDD.

**2:** Parameters are characterized but not tested.

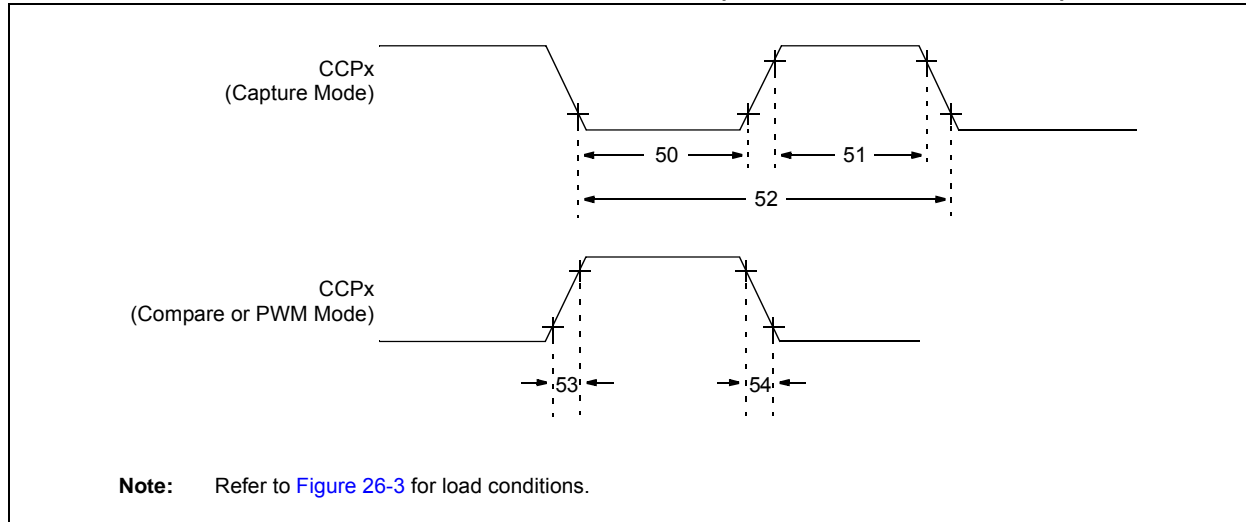
**TABLE 26-25: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS**

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
VR310	TSET	Settling Time <sup>(1)</sup>	—	—	10	μs	

**Note 1:** Settling time is measured while CVRSS = 1 and the CVR<3:0> bits transition from ‘0000’ to ‘1111’.

# PIC24F16KL402 FAMILY

**FIGURE 26-6: CAPTURE/COMPARE/PWM TIMINGS (ECCP1, ECCP2 MODULES)**

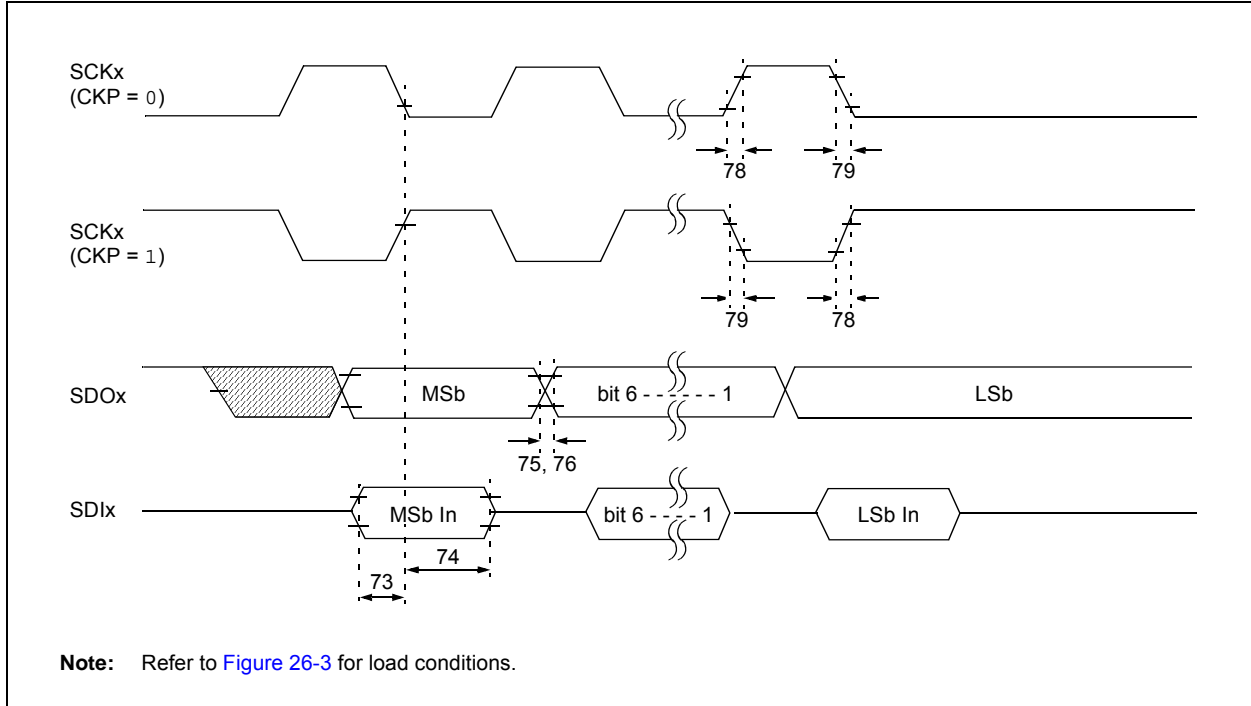


**TABLE 26-26: CAPTURE/COMPARE/PWM REQUIREMENTS (ECCP1, ECCP2 MODULES)**

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
50	TccL	CCPx Input Low Time	No Prescaler	$0.5 T_{CY} + 20$	—	ns	
			With Prescaler	20	—	ns	
51	TccH	CCPx Input High Time	No Prescaler	$0.5 T_{CY} + 20$	—	ns	
			With Prescaler	20	—	ns	
52	TccP	CCPx Input Period		Greater of: 40 or $\frac{2 T_{CY} + 40}{N}$	—	ns	N = prescale value (1, 4 or 16)
53	TccR	CCPx Output Fall Time		—	25	ns	
54	TccF	CCPx Output Rise Time		—	25	ns	

# PIC24F16KL402 FAMILY

**FIGURE 26-7: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)**

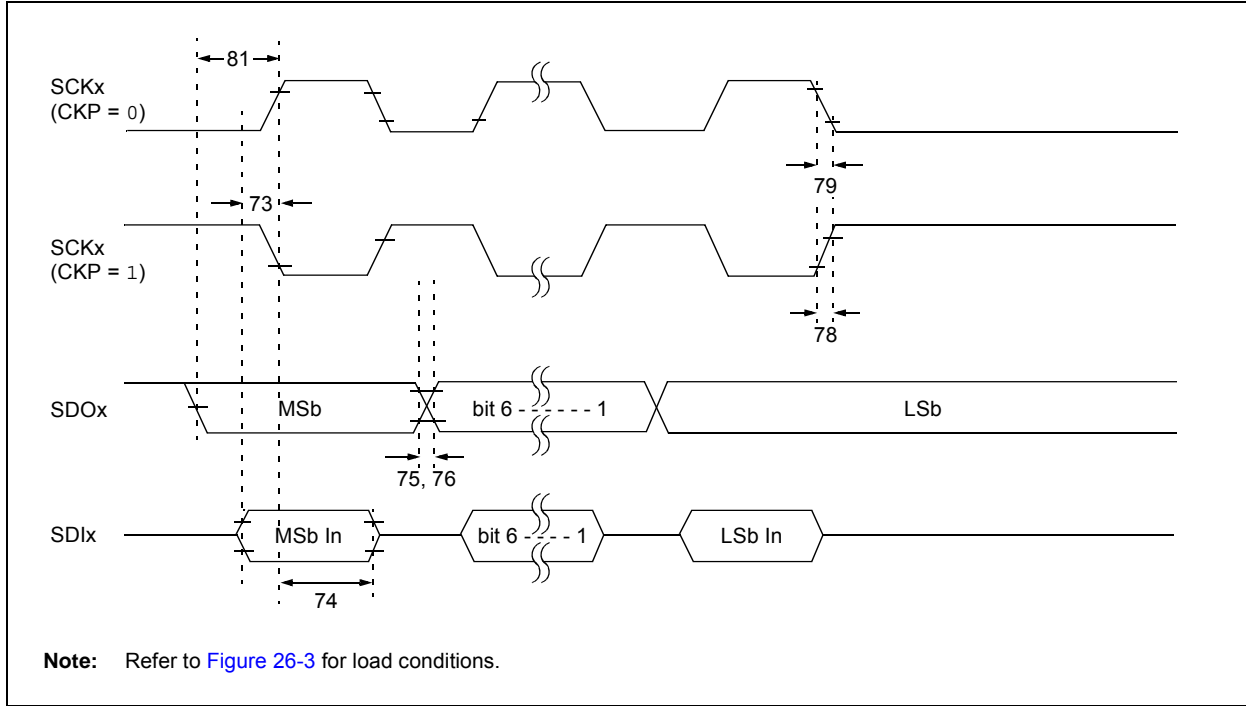


**TABLE 26-27: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)**

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TdIV2scH, TdIV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	ns	
74	Tsch2dIL, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	40	—	ns	
75	TDO <sub>R</sub>	SDOx Data Output Rise Time	—	25	ns	
76	TDO <sub>F</sub>	SDOx Data Output Fall Time	—	25	ns	
78	Tsc <sub>R</sub>	SCKx Output Rise Time (Master mode)	—	25	ns	
79	Tsc <sub>F</sub>	SCKx Output Fall Time (Master mode)	—	25	ns	
	F <sub>SCK</sub>	SCKx Frequency	—	10	MHz	

# PIC24F16KL402 FAMILY

**FIGURE 26-8: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)**

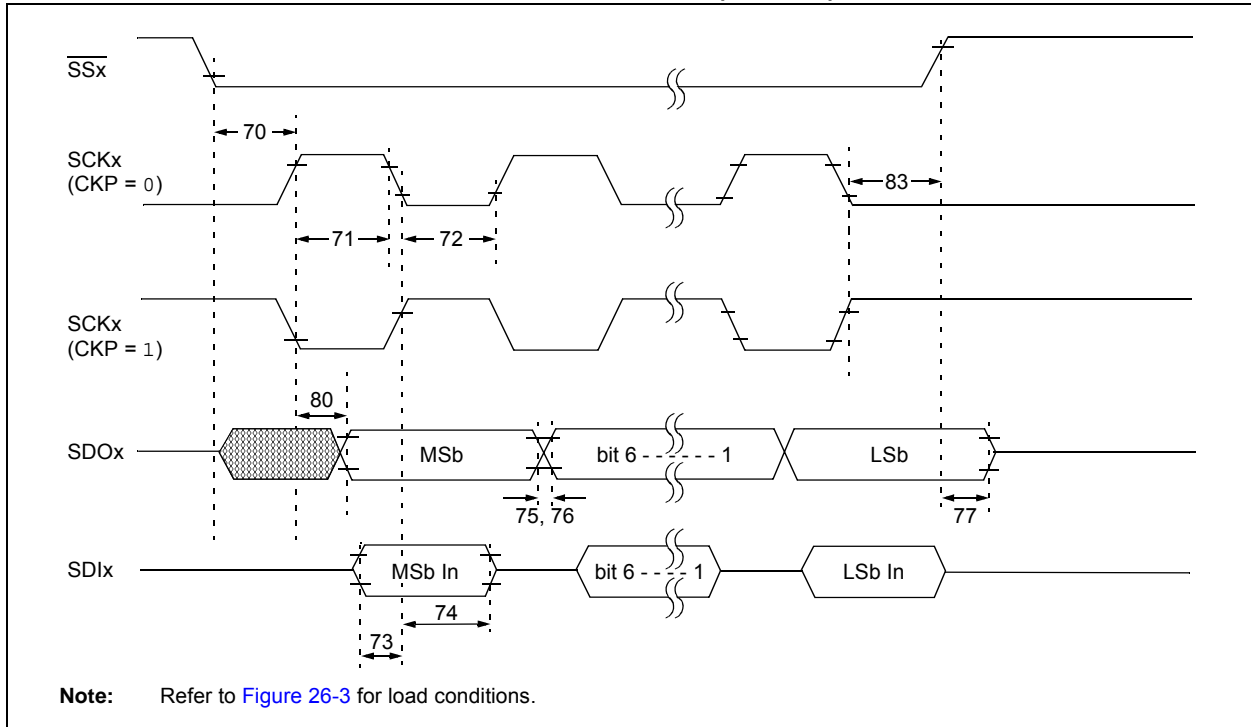


**TABLE 26-28: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
73	TdIV2sCH, TdIV2sCL	Setup Time of SDIx Data Input to SCKx Edge	35	—	ns	
74	TsCH2dIL, TsCL2dIL	Hold Time of SDIx Data Input to SCKx Edge	40	—	ns	
75	TdoR	SDOx Data Output Rise Time	—	25	ns	
76	TdoF	SDOx Data Output Fall Time	—	25	ns	
78	TscR	SCKx Output Rise Time (Master mode)	—	25	ns	
79	TscF	SCKx Output Fall Time (Master mode)	—	25	ns	
81	TdoV2sCH, TdoV2sCL	SDOx Data Output Setup to SCKx Edge	TcY	—	ns	
	FsCK	SCKx Frequency	—	10	MHz	

# PIC24F16KL402 FAMILY

**FIGURE 26-9: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)**



**TABLE 26-29: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)**

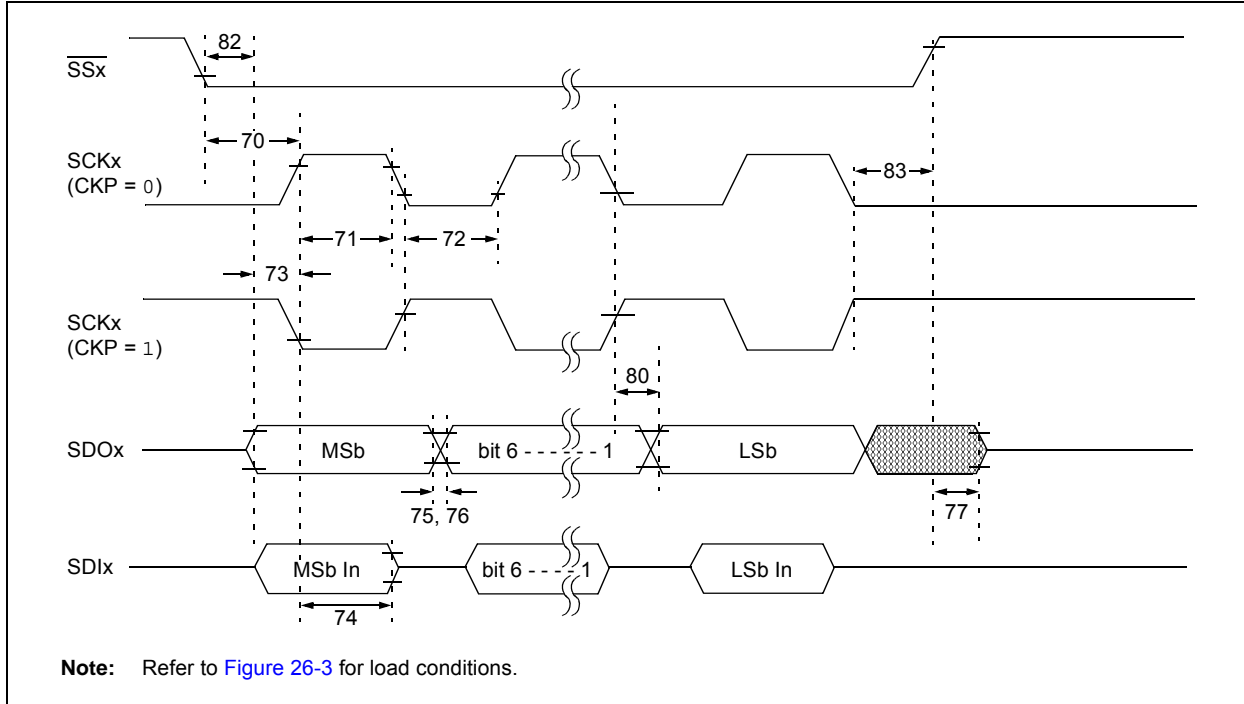
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
70	Tssl2sch, Tssl2scl	$\overline{SSx} \downarrow$ to SCKx $\downarrow$ or SCKx $\uparrow$ Input	3 Tcy	—	ns	
70A	Tssl2wb	$\overline{SSx}$ to Write to SSPxBUF	3 Tcy	—	ns	
71	Tsch	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns
71A			Single Byte	40	—	ns (Note 1)
72	Tscl	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns
72A			Single Byte	40	—	ns (Note 1)
73	Tdiv2sch, Tdiv2scl	Setup Time of SDIx Data Input to SCKx Edge	20	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 Tcy + 40	—	ns	(Note 2)
74	Tsch2dil, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	40	—	ns	
75	TdoR	SDOx Data Output Rise Time	—	25	ns	
76	TdoF	SDOx Data Output Fall Time	—	25	ns	
77	TssH2doZ	$\overline{SSx} \uparrow$ to SDOx Output High-Impedance	10	50	ns	
80	Tsch2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	50	ns	
83	Tsch2ssH, TscL2ssH	$\overline{SSx} \uparrow$ after SCKx Edge	1.5 Tcy + 40	—	ns	
	Fsck	SCKx Frequency	—	10	MHz	

**Note 1:** Requires the use of Parameter 73A.

**2:** Only if Parameters 71A and 72A are used.

# PIC24F16KL402 FAMILY

**FIGURE 26-10: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)**



**TABLE 26-30: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)**

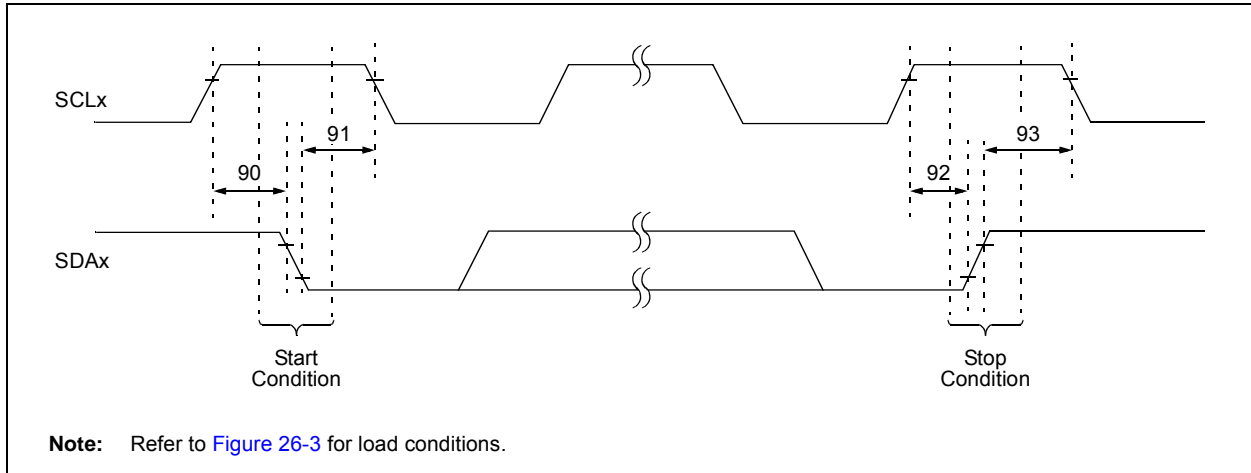
Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
70	TssL2sch, TssL2scl	$\overline{SSx} \downarrow$ to SCKx $\downarrow$ or SCKx $\uparrow$ Input	3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ to Write to SSPxBUF	3 Tcy	—	ns	
71	Tsch	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns
71A			Single Byte	40	—	ns
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns
72A			Single Byte	40	—	ns
73A	Tb2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 Tcy + 40	—	ns	(Note 2)
74	Tsch2dIL, TscL2dIL	Hold Time of SDIx Data Input to SCKx Edge	40	—	ns	
75	TdoR	SDOx Data Output Rise Time	—	25	ns	
76	TdoF	SDOx Data Output Fall Time	—	25	ns	
77	TssH2doZ	$\overline{SSx} \uparrow$ to SDOx Output High-Impedance	10	50	ns	
80	Tsch2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge	—	50	ns	
82	TssL2doV	SDOx Data Output Valid After $\overline{SSx} \downarrow$ Edge	—	50	ns	
83	Tsch2ssH, TscL2ssH	$\overline{SSx} \uparrow$ After SCKx Edge	1.5 Tcy + 40	—	ns	
	Fsck	SCKx Frequency	—	10	MHz	

**Note 1:** Requires the use of Parameter 73A.

**Note 2:** Only if Parameters 71A and 72A are used.

# PIC24F16KL402 FAMILY

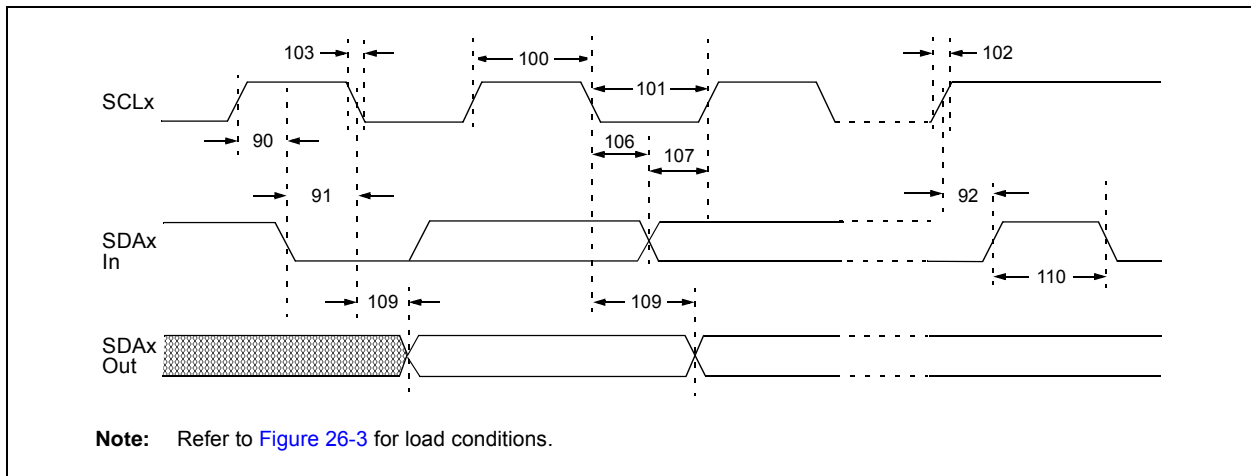
**FIGURE 26-11: I<sup>2</sup>C™ BUS START/STOP BITS TIMING**



**TABLE 26-31: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		

**FIGURE 26-12: I<sup>2</sup>C™ BUS DATA TIMING**



# PIC24F16KL402 FAMILY

**TABLE 26-32: I<sup>2</sup>C™ BUS DATA REQUIREMENTS (SLAVE MODE)**

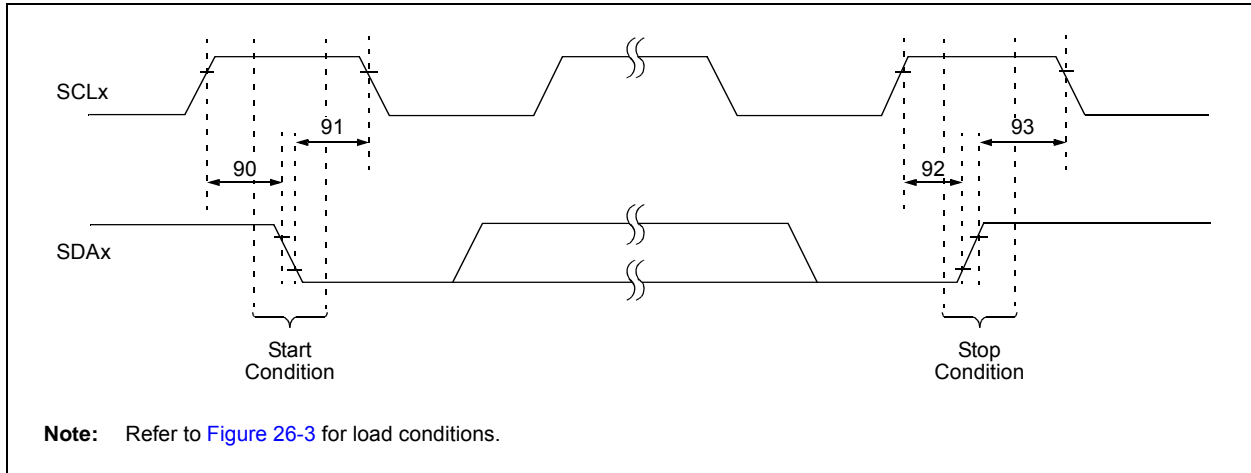
Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	Must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Must operate at a minimum of 10 MHz
			MSSP module	1.5	—	T <sub>CY</sub>	
101	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	Must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Must operate at a minimum of 10 MHz
			MSSP module	1.5	—	T <sub>CY</sub>	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 to 400 pF
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns	C <sub>B</sub> is specified to be from 10 to 400 pF
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	<b>(Note 2)</b>
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	<b>(Note 1)</b>
			400 kHz mode	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
D102	CB	Bus Capacitive Loading		—	400	pF	

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCLx to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode I<sup>2</sup>C™ bus device can be used in a Standard mode I<sup>2</sup>C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I<sup>2</sup>C bus specification), before the SCLx line is released.

# PIC24F16KL402 FAMILY

**FIGURE 26-13: MSSPx I<sup>2</sup>C™ BUS START/STOP BITS TIMING WAVEFORMS**

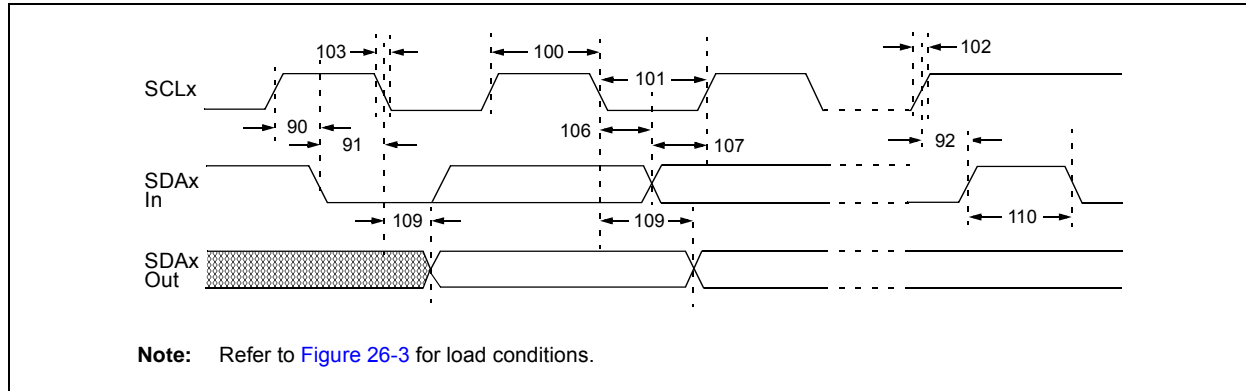


**TABLE 26-33: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS (MASTER MODE)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	ns Only relevant for Repeated Start condition
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	
91	THD:STA	Start Condition Hold Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	ns After this period, the first clock pulse is generated
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	
93	THD:STO	Stop Condition Hold Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	ns
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	

# PIC24F16KL402 FAMILY

**FIGURE 26-14: MSSPx I<sup>2</sup>C™ BUS DATA TIMING**



**TABLE 26-34: I<sup>2</sup>C™ BUS DATA REQUIREMENTS (MASTER MODE)**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions	
100	THIGH	Clock High Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
101	TLOW	Clock Low Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	$20 + 0.1 C_B$	300	ns	
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	$20 + 0.1 C_B$	300	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	Only relevant for Repeated Start condition
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
91	THD:STA	Start Condition Hold Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	After this period, the first clock pulse is generated
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	<b>(Note 1)</b>
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
			400 kHz mode	$2(T_{osc})(BRG + 1)$	—	—	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
D102	CB	Bus Capacitive Loading	—	400	pF		

**Note 1:** A Fast mode I<sup>2</sup>C bus device can be used in a Standard mode I<sup>2</sup>C bus system, but Parameter 107  $\geq$  250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, Parameter 102 + Parameter 107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

# PIC24F16KL402 FAMILY

**TABLE 26-35: A/D MODULE SPECIFICATIONS**

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
<b>Device Supply</b>							
AD01	AVDD	Module VDD Supply	Greater of: VDD – 0.3 or 1.8	—	Lesser of: VDD + 0.3 or 3.6	V	
AD02	AVSS	Module VSS Supply	VSS – 0.3	—	VSS + 0.3	V	
<b>Reference Inputs</b>							
AD05	VREFH	Reference Voltage High	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	Reference Voltage Low	AVSS	—	AVDD – 1.7	V	
AD07	VREF	Absolute Reference Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
<b>Analog Input</b>							
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	<b>(Note 1)</b>
AD11	VIN	Absolute Input Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
AD12	VINL	Absolute VINL Input Voltage	AVSS – 0.3	—	AVDD/2	V	
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	2.5K	$\Omega$	10-bit
<b>A/D Accuracy</b>							
AD20b	NR	Resolution	—	10	—	bits	
AD21b	INL	Integral Nonlinearity	—	$\pm 1$	$\pm 2$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	Differential Nonlinearity	—	$\pm 1$	$\pm 1.5$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	Gain Error	—	$\pm 1$	$\pm 3$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	Offset Error	—	$\pm 1$	$\pm 2$	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b		Monotonicity	—	—	—	—	<b>(Note 2)</b>

**Note 1:** Measurements are taken with external VREF+ and VREF- used as the A/D voltage reference.

**2:** The A/D conversion result never decreases with an increase in the input voltage.

# PIC24F16KL402 FAMILY

**TABLE 26-36: A/D CONVERSION TIMING REQUIREMENTS<sup>(1)</sup>**

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial				
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions
<b>Clock Parameters</b>							
AD50	TAD	A/D Clock Period	75	—	—	ns	$T_{CY} = 75 \text{ ns}$ , AD1CON3 is in default state
AD51	TRC	A/D Internal RC Oscillator Period	—	250	—	ns	
<b>Conversion Rate</b>							
AD55	TCONV	Conversion Time	—	12	—	TAD	
AD56	FCNV	Throughput Rate	—	—	500	ksps	$AV_{DD} \geq 2.7\text{V}$
AD57	TSAMP	Sample Time	—	1	—	TAD	
AD58	TACQ	Acquisition Time	750	—	—	ns	<b>(Note 2)</b>
AD59	TSWC	Switching Time from Convert to Sample	—	—	<b>(Note 3)</b>	—	
AD60	TDIS	Discharge Time	0.5	—	—	TAD	
<b>Clock Parameters</b>							
AD61	TPSS	Sample Start Delay from Setting Sample bit (SAMP)	2	—	3	TAD	

**Note 1:** Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

**2:** The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion ( $V_{DD}$  to  $V_{SS}$  or  $V_{SS}$  to  $V_{DD}$ ).

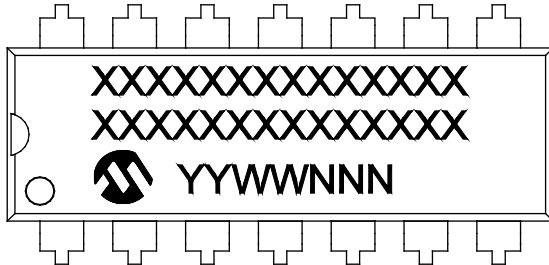
**3:** On the following cycle of the device clock.

# PIC24F16KL402 FAMILY

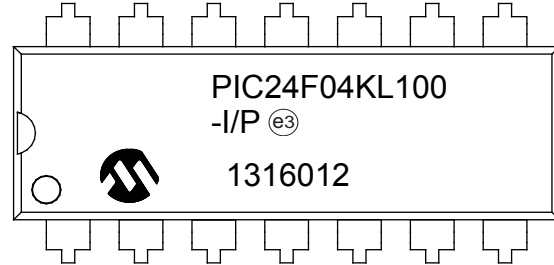
## 27.0 PACKAGING INFORMATION

### 27.1 Package Marking Information

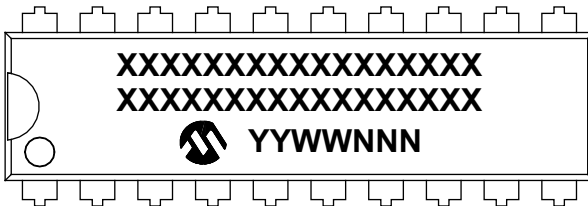
14-Lead PDIP (300 mil)



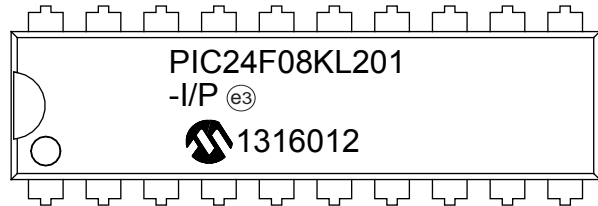
Example



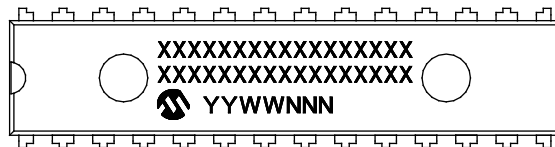
20-Lead PDIP (300 mil)



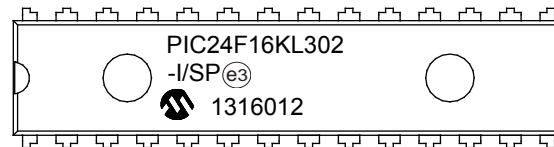
Example



28-Lead SPDIP (.300")



Example

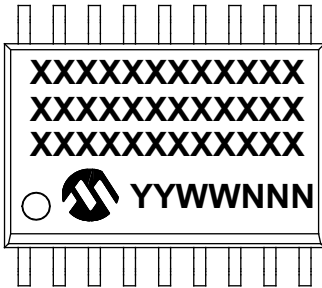


<b>Legend:</b>	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	* (e3)	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

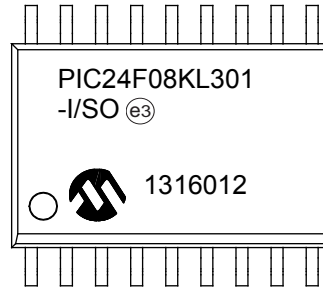
**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# PIC24F16KL402 FAMILY

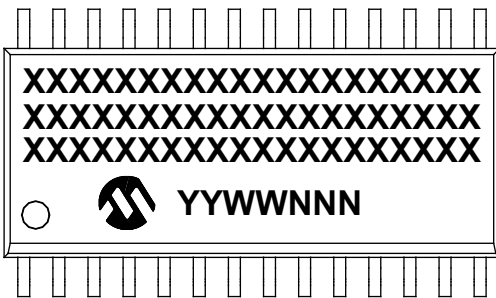
20-Lead SOIC (7.50 mm)



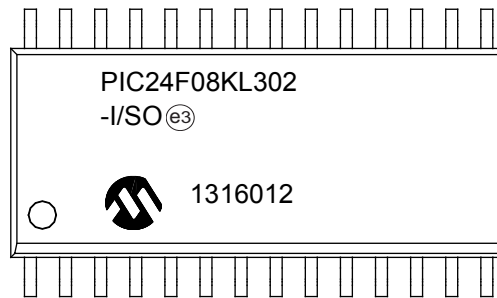
Example



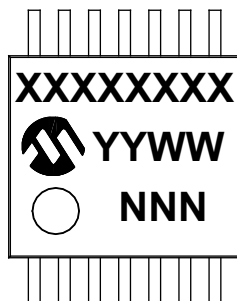
28-Lead SOIC (7.50 mm)



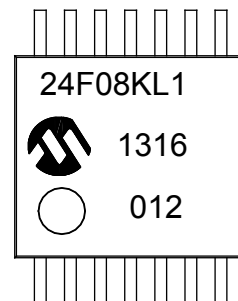
Example



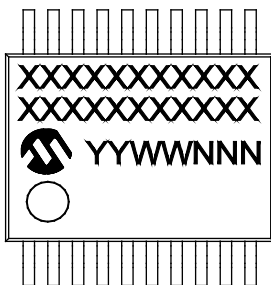
14-Lead TSSOP (4.4 mm)



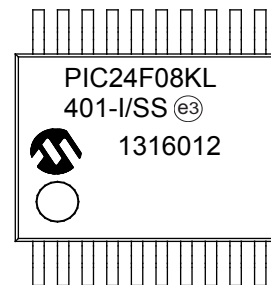
Example



20-Lead SSOP (5.30 mm)

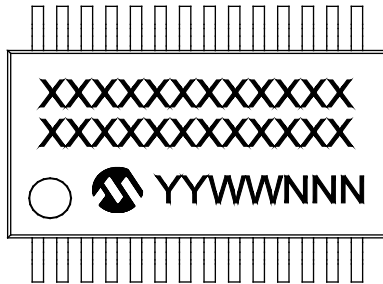


Example

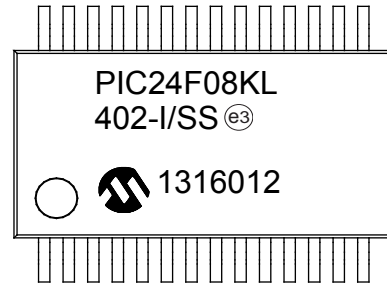


# PIC24F16KL402 FAMILY

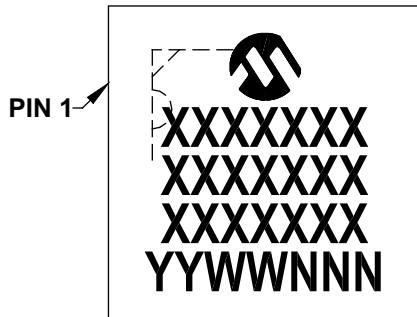
28-Lead SSOP (5.30 mm)



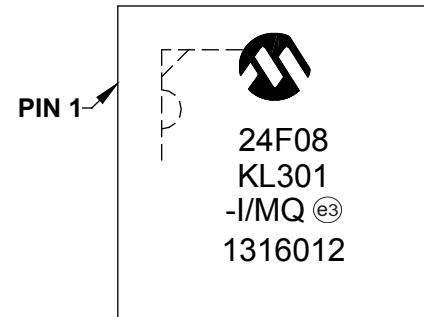
Example



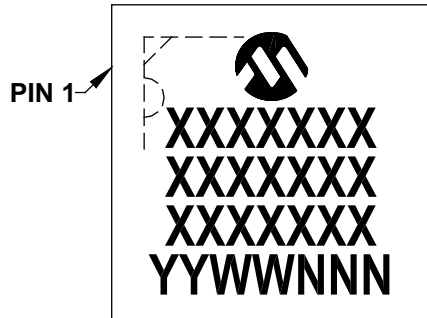
20-Lead QFN (5x5x0.9 mm)



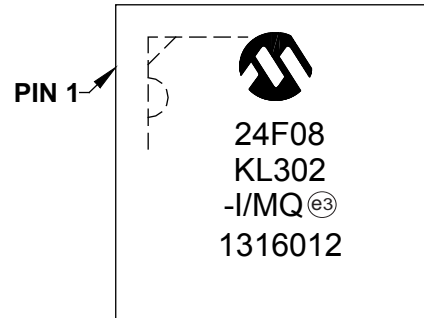
Example



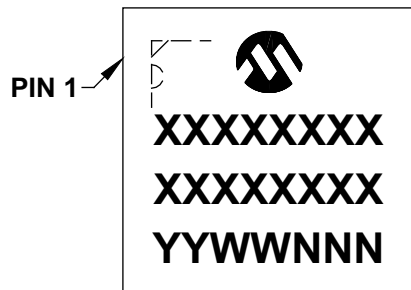
28-Lead QFN (5x5x0.9 mm)



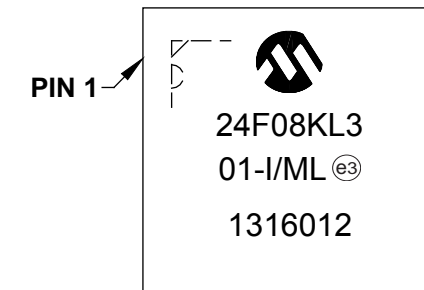
Example



28-Lead QFN (6x6 mm)



Example



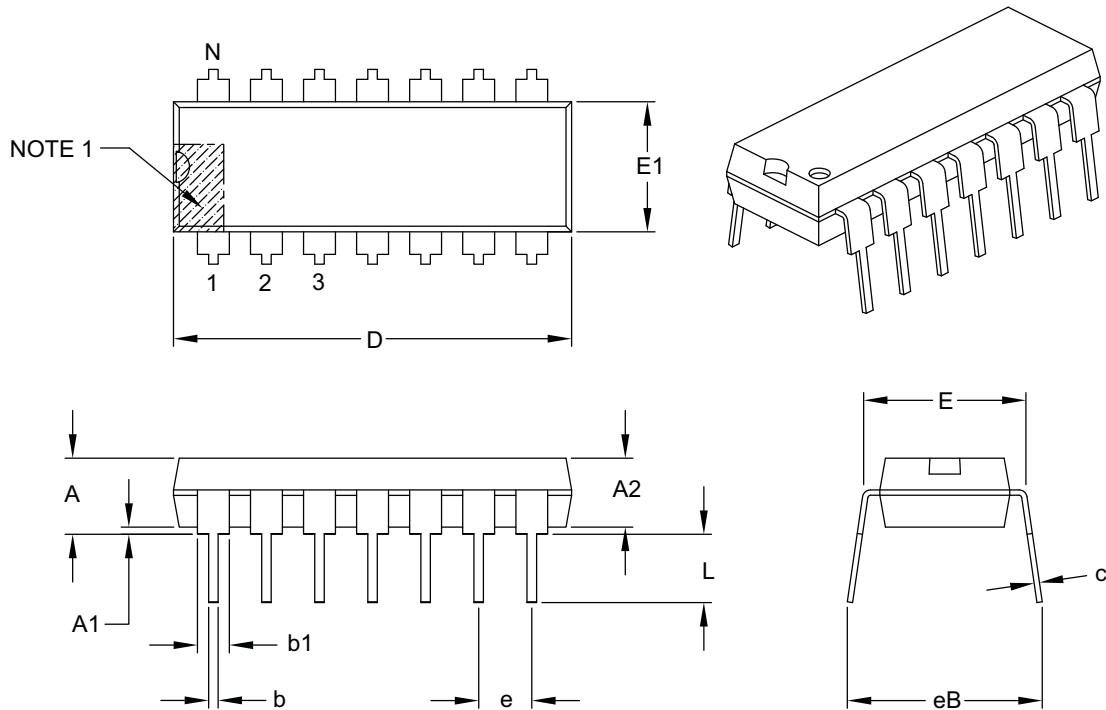
# PIC24F16KL402 FAMILY

## 27.2 Package Details

The following sections give the technical details of the packages.

### 14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

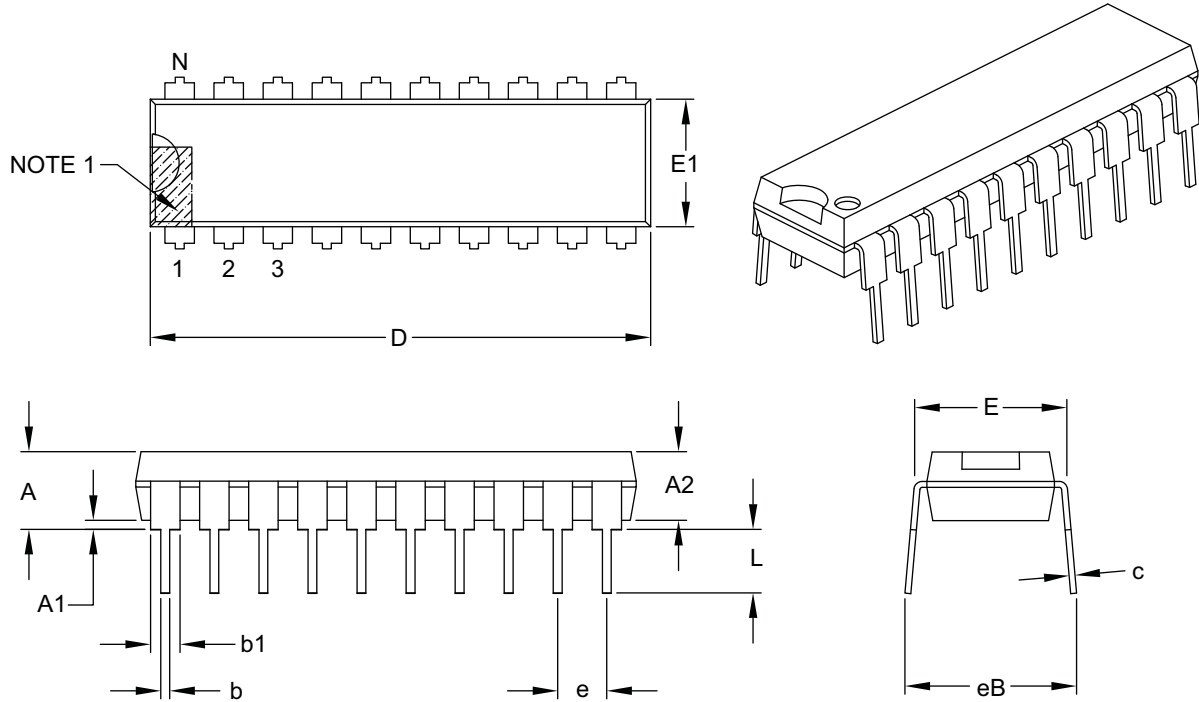
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

# PIC24F16KL402 FAMILY

## 20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

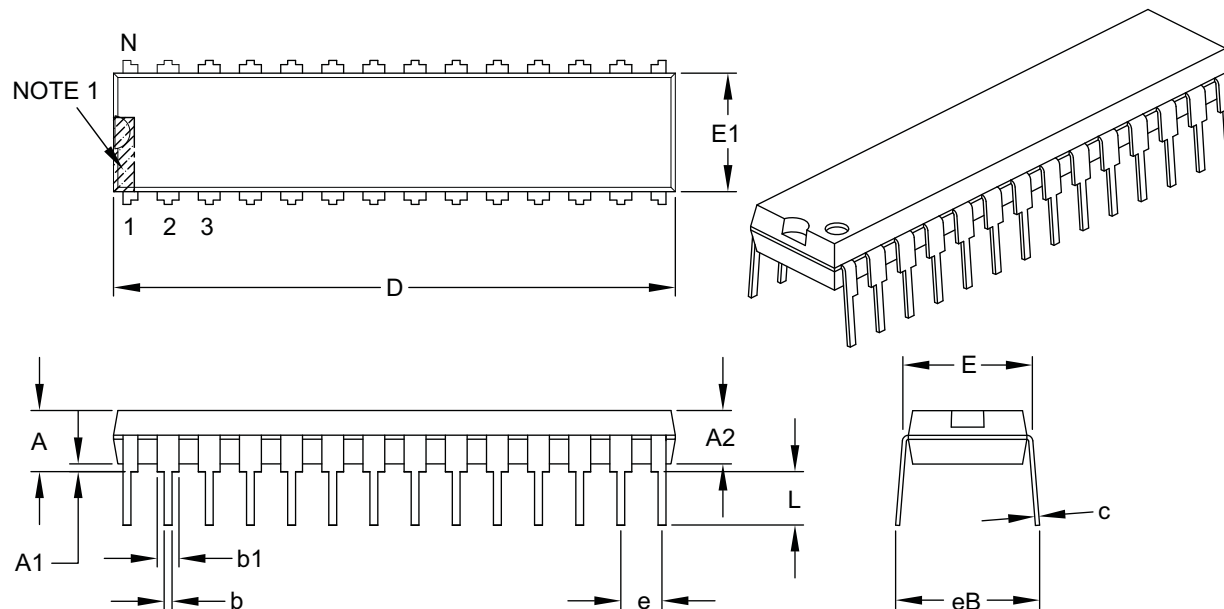
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

# PIC24F16KL402 FAMILY

## 28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

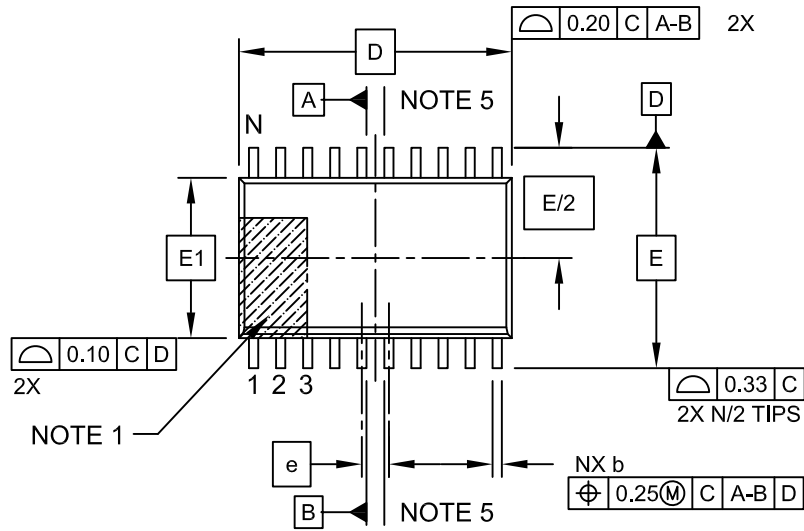
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

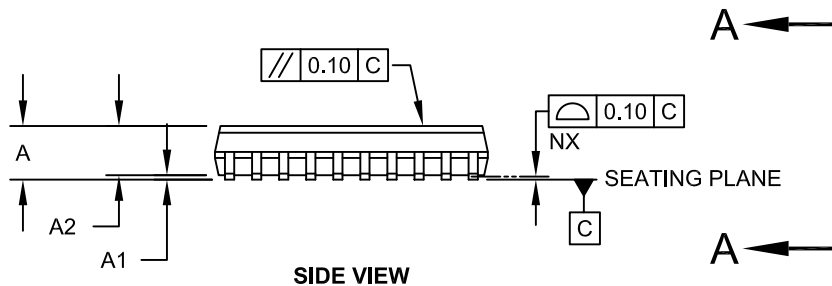
# PIC24F16KL402 FAMILY

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

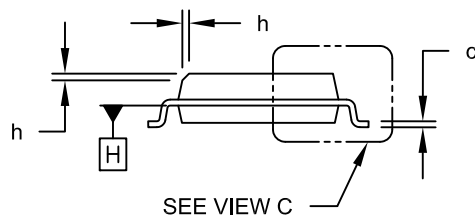
**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



TOP VIEW



SIDE VIEW



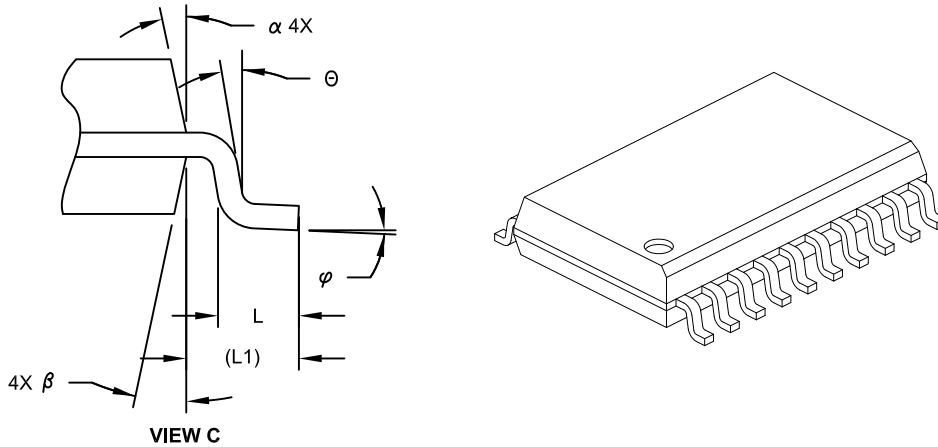
VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

# PIC24F16KL402 FAMILY

## 20-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>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

### Notes:

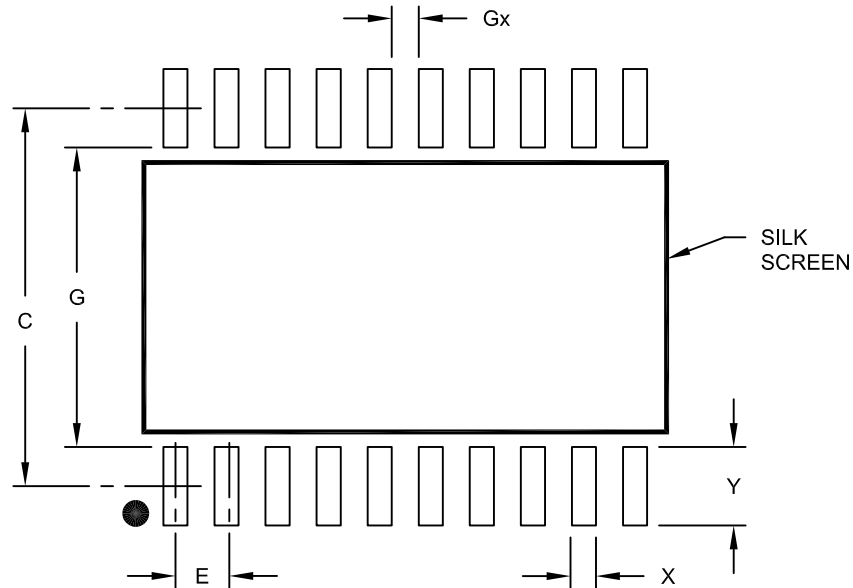
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

# PIC24F16KL402 FAMILY

20-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>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

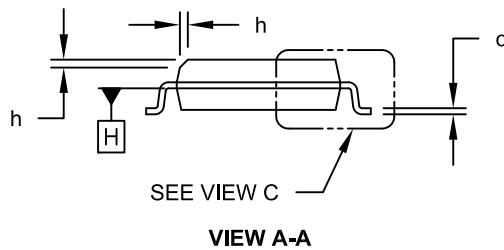
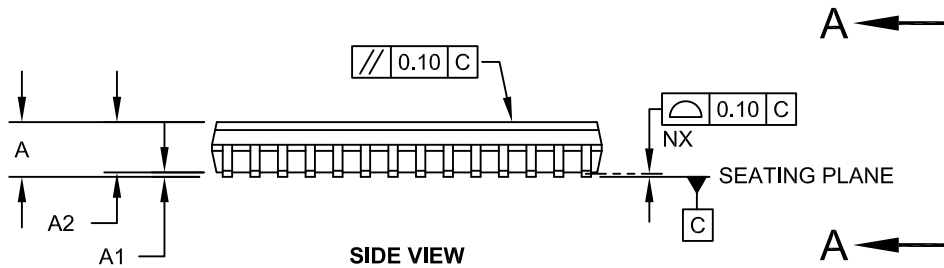
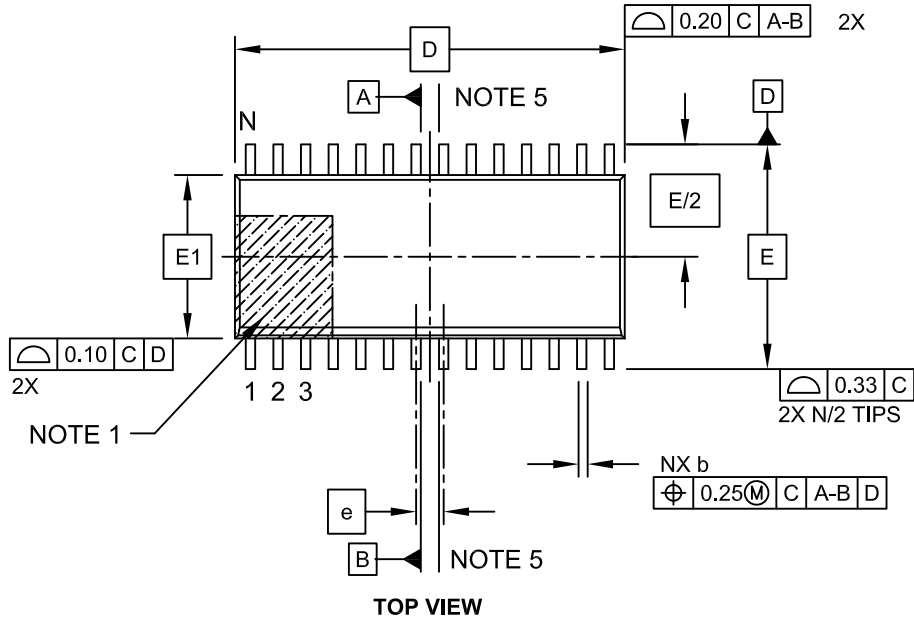
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

# PIC24F16KL402 FAMILY

## 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>

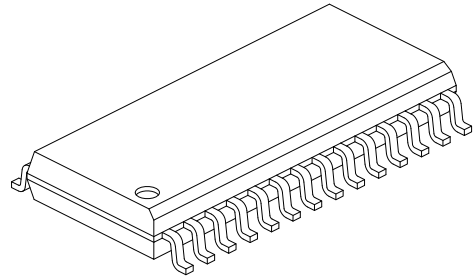
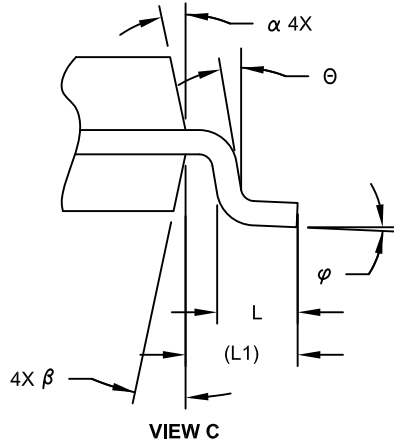


Microchip Technology Drawing C04-052C Sheet 1 of 2

# PIC24F16KL402 FAMILY

## 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>



		MILLIMETERS		
Units				
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	17.90 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

**Notes:**

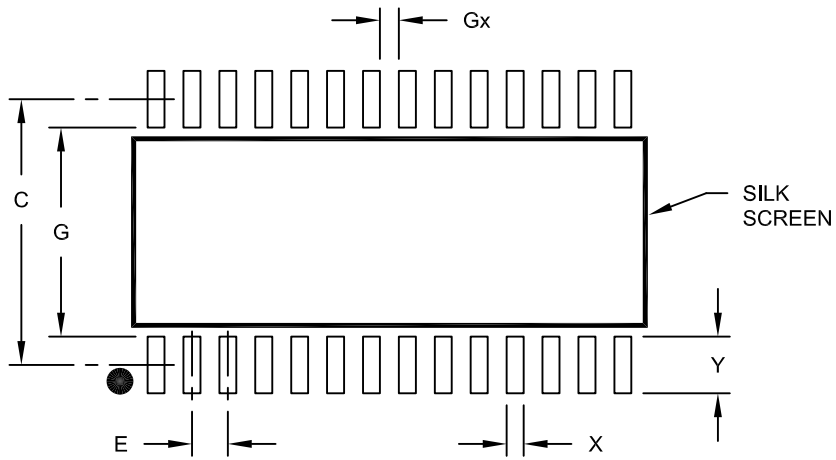
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
 BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
 REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

# PIC24F16KL402 FAMILY

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>



## RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

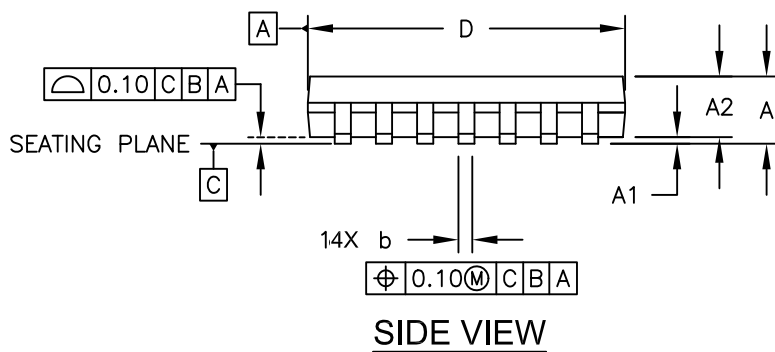
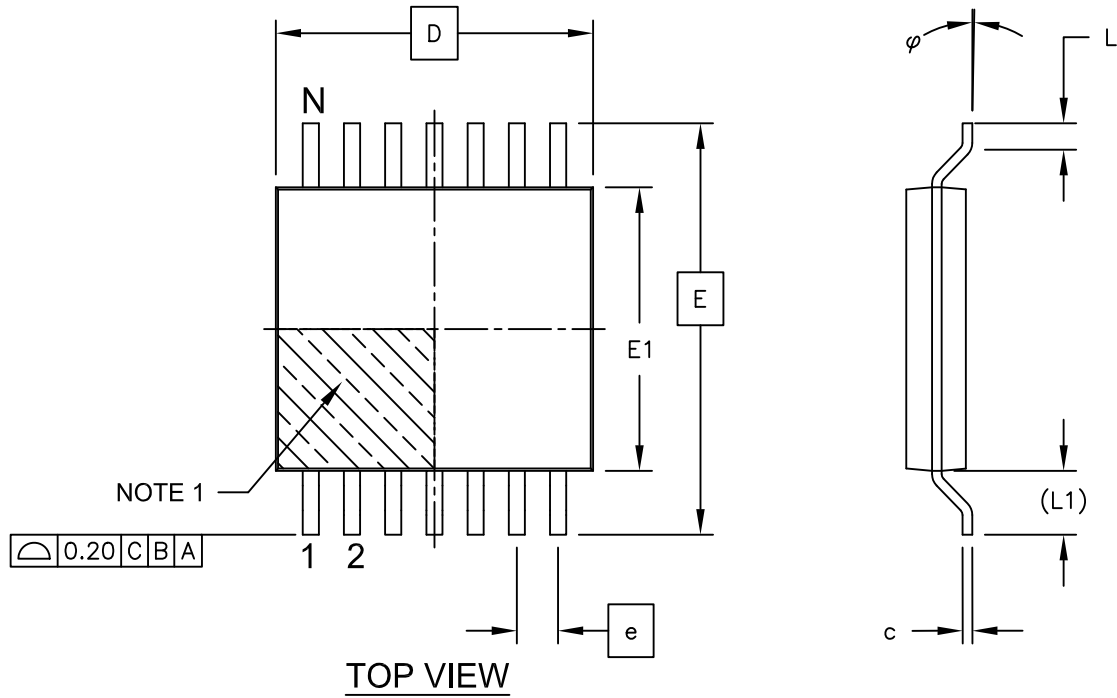
BSC: Basic Dimension, Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

# PIC24F16KL402 FAMILY

## 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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

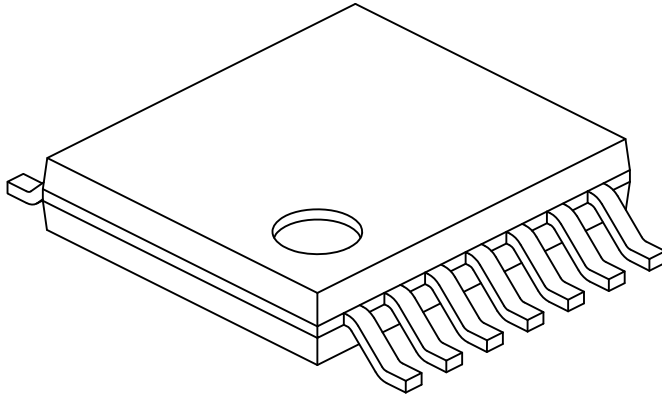


Microchip Technology Drawing C04-087C Sheet 1 of 2

# PIC24F16KL402 FAMILY

## 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	14		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.80	1.00	1.05
Standoff	A1	0.05	-	0.15
Overall Width	E	6.40 BSC		
Molded Package Width	E1	4.30	4.40	4.50
Molded Package Length	D	4.90	5.00	5.10
Foot Length	L	0.45	0.60	0.75
Footprint	(L1)	1.00 REF		
Foot Angle	$\varphi$	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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

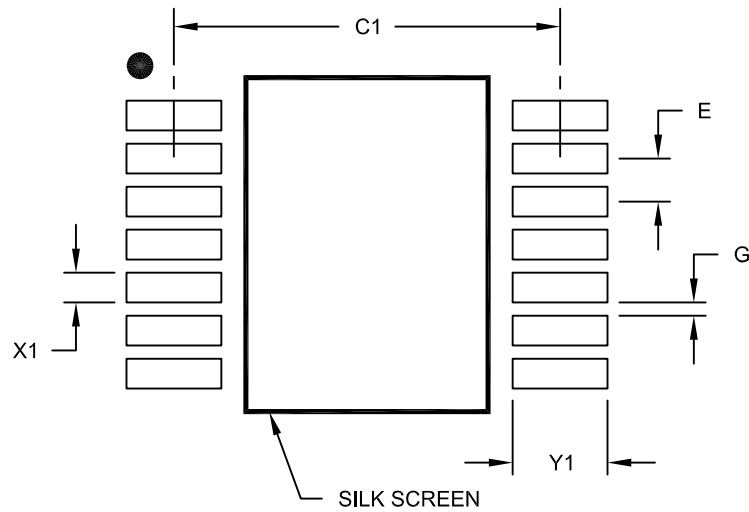
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing No. C04-087C Sheet 2 of 2

# PIC24F16KL402 FAMILY

## 14-Lead Plastic Thin Shrink Small Outline (ST) - 4.4 mm Body [TSSOP]

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C1		5.90	
Contact Pad Width (X14)	X1			0.45
Contact Pad Length (X14)	Y1			1.45
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

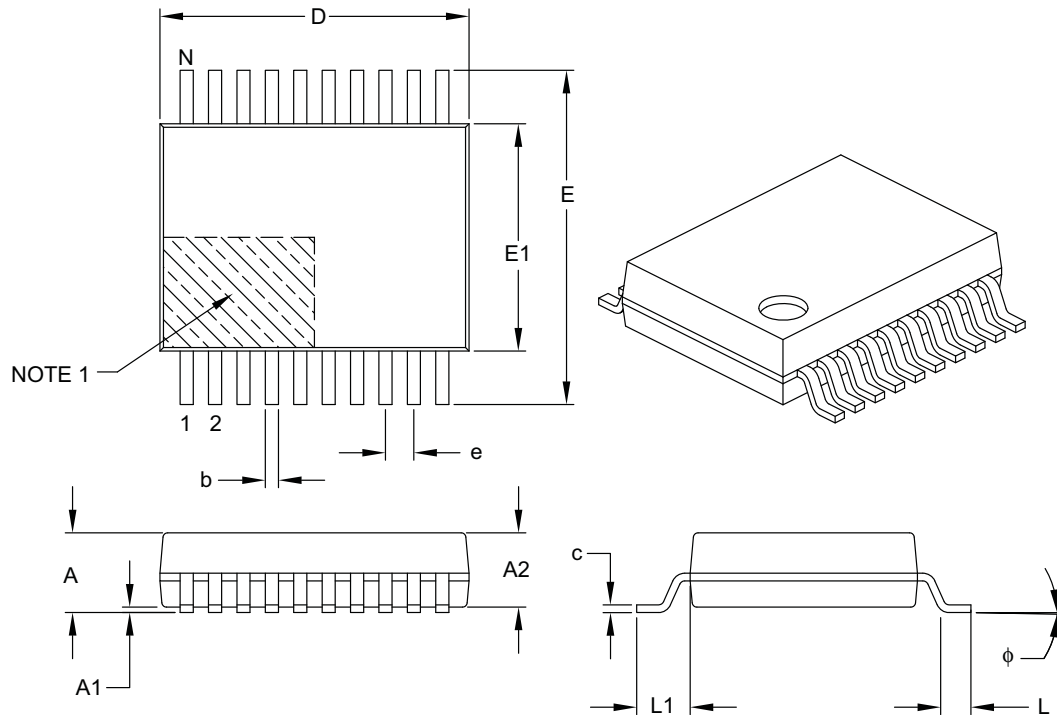
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2087A

# PIC24F16KL402 FAMILY

## 20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		20		
Pitch	e		0.65 BSC		
Overall Height	A	–	–	–	2.00
Molded Package Thickness	A2		1.65	1.75	1.85
Standoff	A1		0.05	–	–
Overall Width	E		7.40	7.80	8.20
Molded Package Width	E1		5.00	5.30	5.60
Overall Length	D		6.90	7.20	7.50
Foot Length	L		0.55	0.75	0.95
Footprint	L1		1.25 REF		
Lead Thickness	c		0.09	–	0.25
Foot Angle	$\phi$		0°	4°	8°
Lead Width	b		0.22	–	0.38

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

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

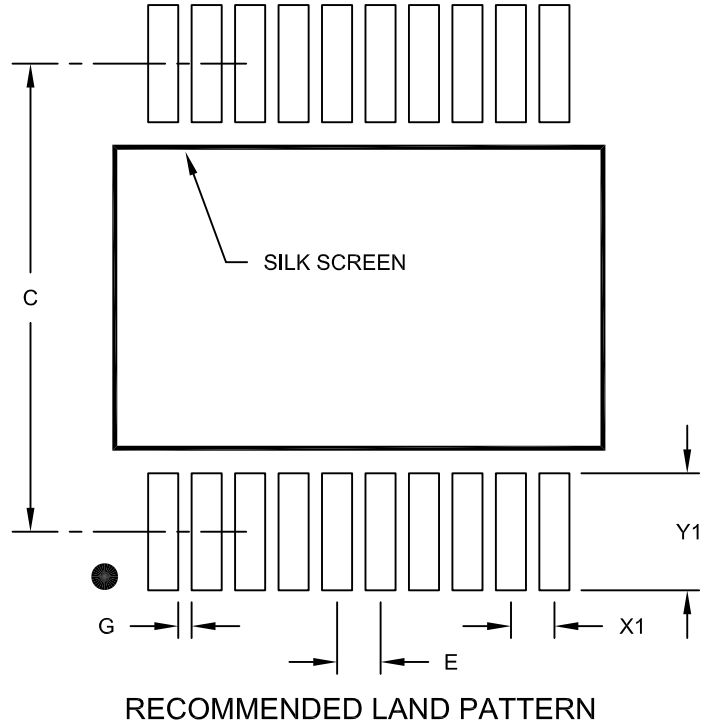
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

# PIC24F16KL402 FAMILY

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

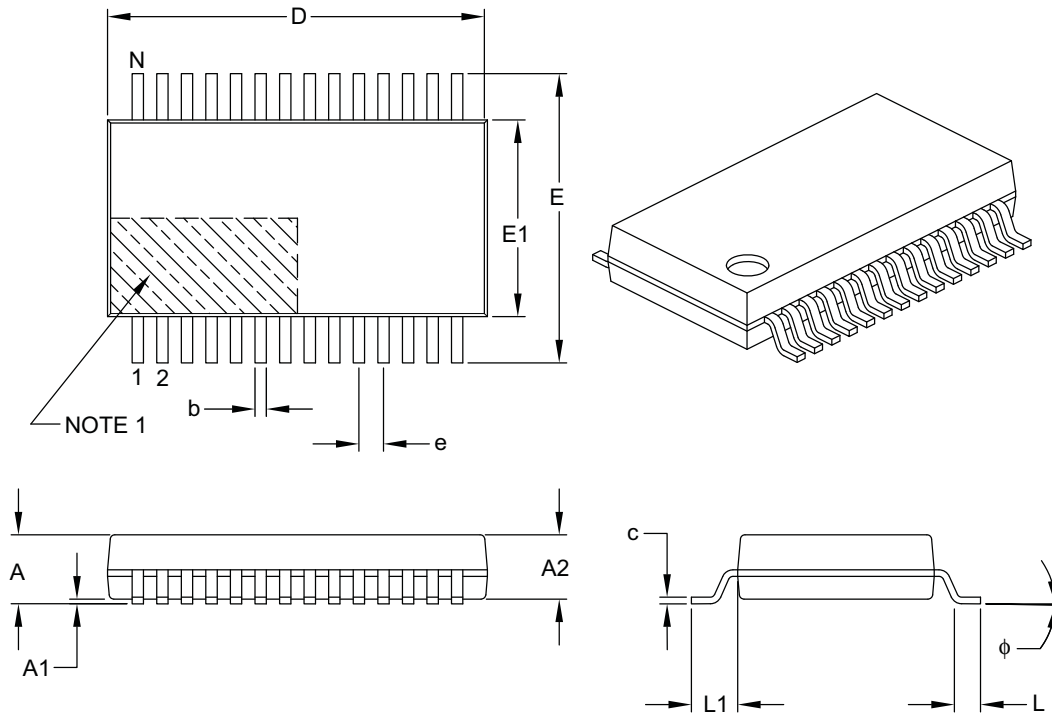
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	$\phi$	0°	4°	8°
Lead Width	b	0.22	–	0.38

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

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

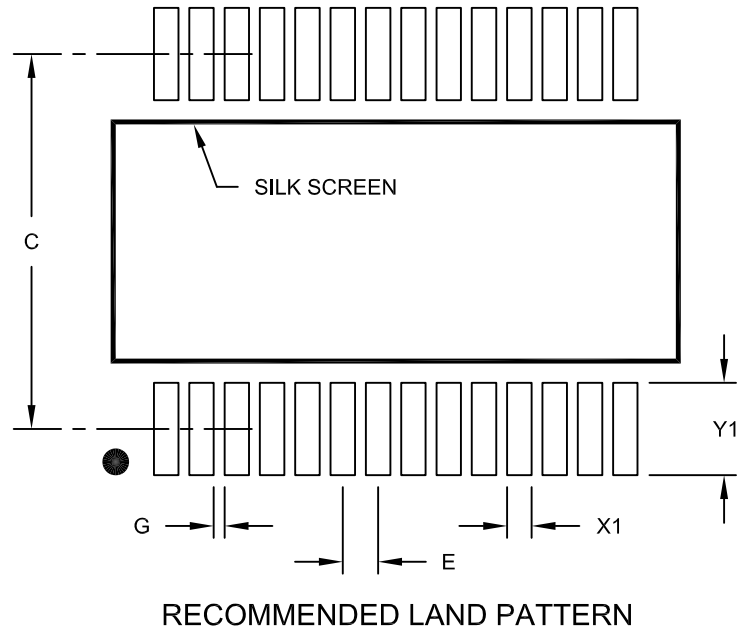
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC24F16KL402 FAMILY

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

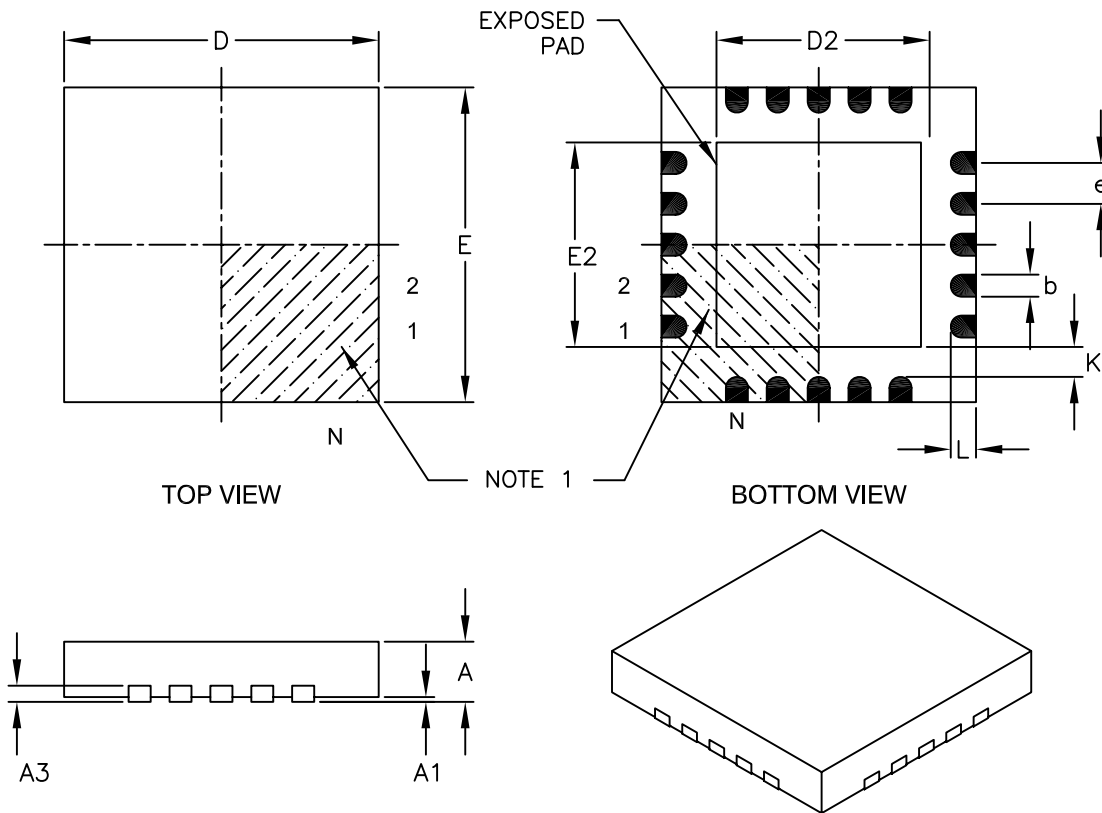
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

# PIC24F16KL402 FAMILY

## 20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	3.15	3.25	3.35
Overall Length	D	5.00 BSC		
Exposed Pad Length	D2	3.15	3.25	3.35
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.35	0.40	0.45
Contact-to-Exposed Pad	K	0.20	-	-

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

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

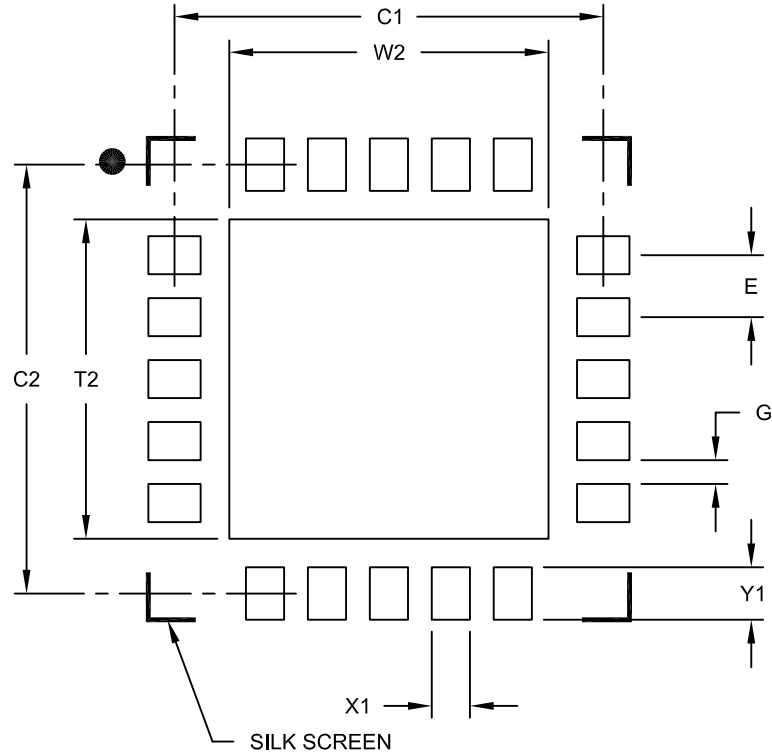
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-139B

# PIC24F16KL402 FAMILY

20-Lead Plastic Quad Flat, No Lead Package (MQ) - 5x5 mm Body [QFN]  
With 0.40mm Contact Length

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.50	
Contact Pad Spacing	C2		4.50	
Contact Pad Width (X20)	X1			0.40
Contact Pad Length (X20)	Y1			0.55
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

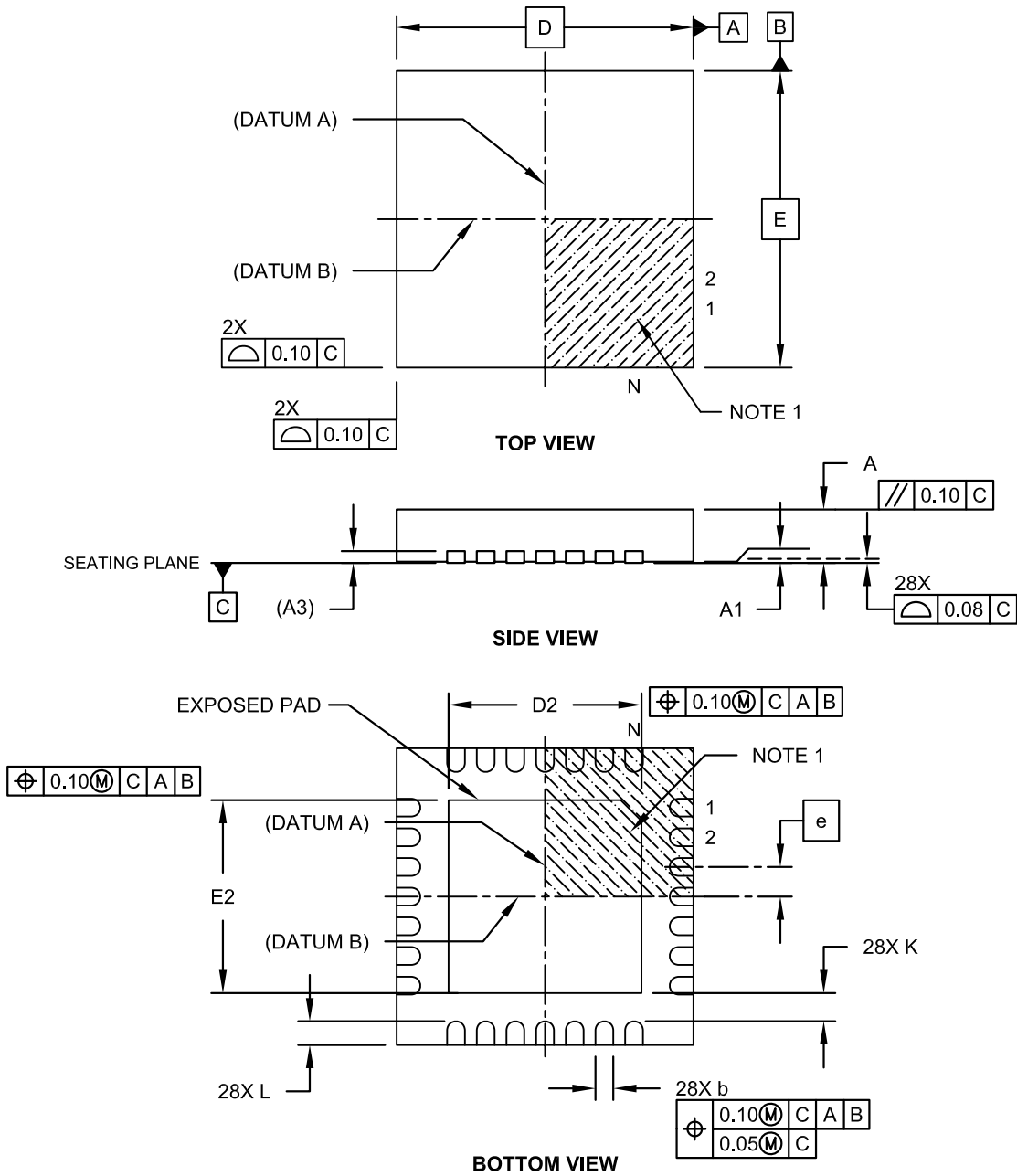
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2139A

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN]

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

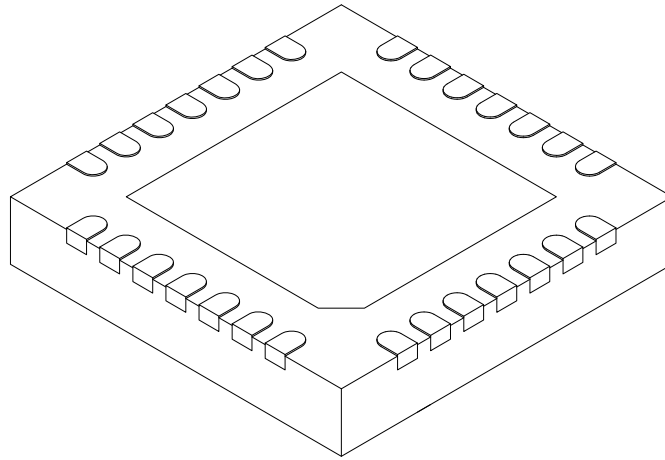


Microchip Technology Drawing C04-140B Sheet 1 of 2

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN]

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.50 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	3.15	3.25	3.35
Overall Length	D	5.00 BSC		
Exposed Pad Length	D2	3.15	3.25	3.35
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.35	0.40	0.45
Contact-to-Exposed Pad	K	0.20	-	-

**Notes:**

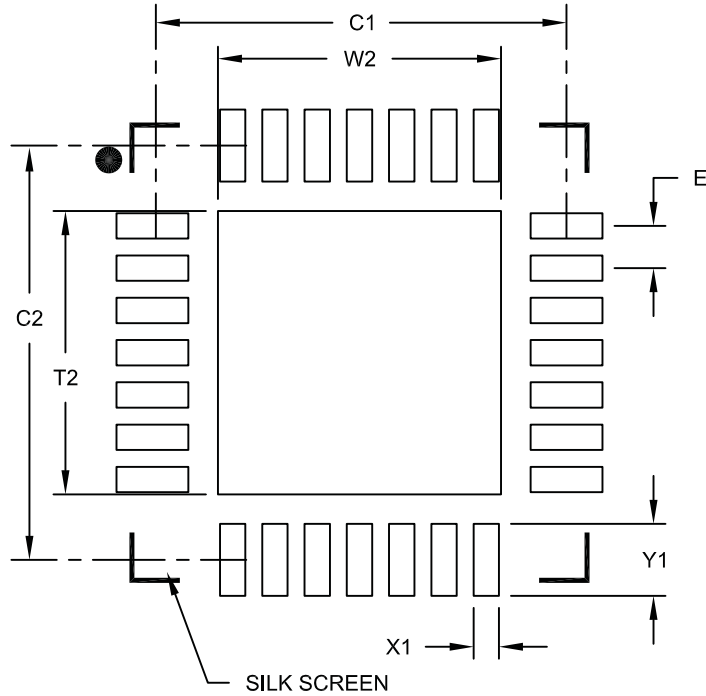
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-140B Sheet 2 of 2

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5 mm Body [QFN] Land Pattern With 0.55 mm Contact Length

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.90	
Contact Pad Spacing	C2		4.90	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			0.85

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

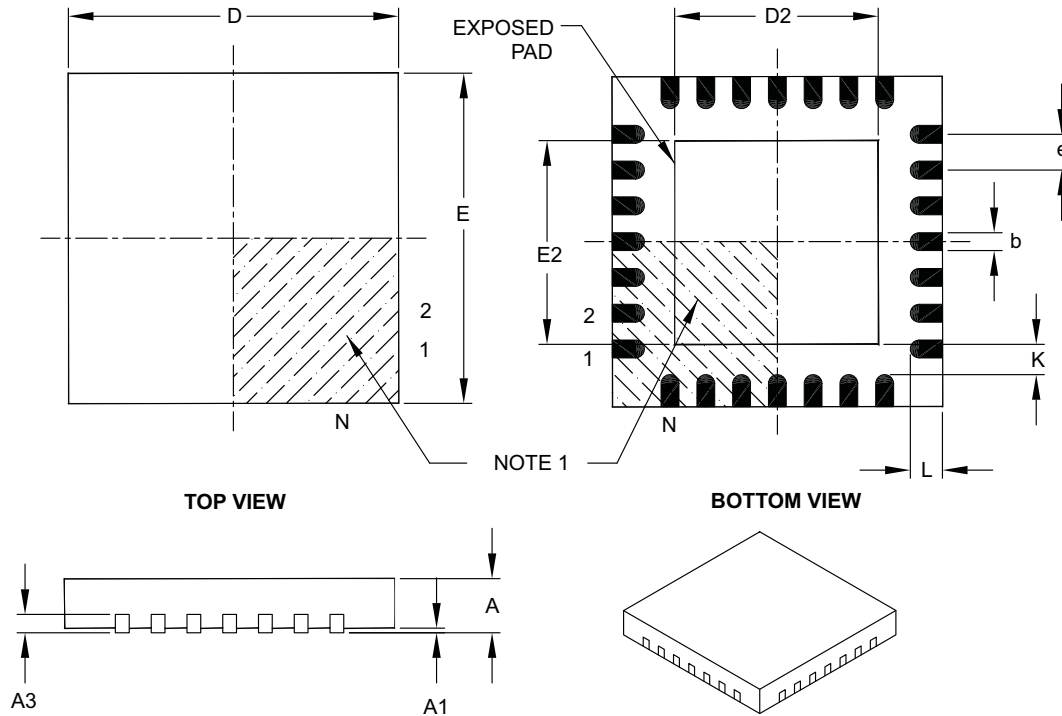
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2140A

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		28		
Pitch	e		0.65 BSC		
Overall Height	A		0.80	0.90	1.00
Standoff	A1		0.00	0.02	0.05
Contact Thickness	A3		0.20 REF		
Overall Width	E		6.00 BSC		
Exposed Pad Width	E2		3.65	3.70	4.20
Overall Length	D		6.00 BSC		
Exposed Pad Length	D2		3.65	3.70	4.20
Contact Width	b		0.23	0.30	0.35
Contact Length	L		0.50	0.55	0.70
Contact-to-Exposed Pad	K		0.20	-	-

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

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

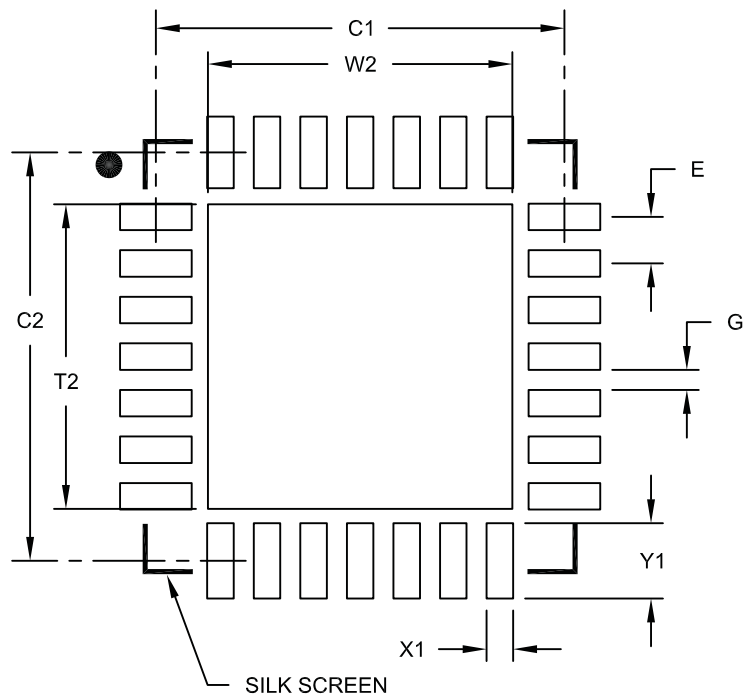
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

# PIC24F16KL402 FAMILY

## 28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 mm Contact Length

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
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. C04-2105A

# PIC24F16KL402 FAMILY

## APPENDIX A: REVISION HISTORY

### Revision A (September 2011)

Original data sheet for the PIC24F16KL402 family of devices.

### Revision B (November 2011)

Updates DC Specifications in Tables 26-6 through 26-9 (all Typical and Maximum values).

Updates AC Specifications in Tables 26-7 through 26-30 (SPI Timing Requirements) with the addition of the FSK specification.

Other minor typographic corrections throughout.

### Revision C (October 2013)

Adds +125°C Extended Temperature information.

Updates several packaging drawings in **Section 27.0 “Packaging Information”**. Other minor typographic corrections throughout.

## APPENDIX B: MIGRATING FROM PIC18/PIC24 TO PIC24F16KL402

The PIC24F16KL402 family combines traditional PIC18 peripherals with a faster PIC24 core to provide a low-cost, high-performance microcontroller with low-power consumption.

Code written for PIC18 devices can be migrated to the PIC24F16KL402 by using a C compiler that generates PIC24 machine level instructions. Assembly language code will need to be rewritten using PIC24 instructions. The PIC24 instruction set shares similarities to the PIC18 instruction set, which should ease porting of assembly code. Application code will require changes to support certain PIC24 peripherals.

Code written for PIC24 devices can be migrated to the PIC24F16KL402 without many code changes. Certain peripherals, however, will require application changes to support modules that were traditionally available only on PIC18 devices.

Refer to [Table B-1](#) for a list of peripheral modules on the PIC24F16KL402 and where they originated from.

**TABLE B-1: TABLE B-1: PIC24F16KL402 PERIPHERAL MODULE ORIGINATING ARCHITECTURE**

Peripheral Module	PIC18	PIC24
ECCP/CCP	X	—
MSSP (I <sup>2</sup> C™/SPI)	X	—
Timer2/4 (8-bit)	X	—
Timer3 (16-bit)	X	—
Timer1 (16-bit)	—	X
10-Bit A/D Converter	—	X
Comparator	—	X
Comparator Voltage Reference	—	X
UART	—	X
HLVD	—	X

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## INDEX

### A

A/D	
10-Bit High-Speed A/D Converter .....	157
Conversion Timing Requirements .....	224
Module Specifications .....	223
A/D Converter	
Analog Input Model .....	164
Transfer Function .....	165
AC Characteristics	
A/D Module .....	223
Capacitive Loading Requirements on	
Output Pins .....	209
Internal RC Oscillator Accuracy .....	211
Internal RC Oscillator Specifications .....	211
Load Conditions and Requirements .....	209
Temperature and Voltage Specifications .....	209
Assembler	
MPASM Assembler .....	188

### B

Block Diagrams	
10-Bit High-Speed A/D Converter .....	158
16-Bit Timer1 .....	115
Accessing Program Memory with	
Table Instructions .....	45
CALL Stack Frame .....	43
Capture Mode Operation .....	126
Comparator Module .....	167
Comparator Voltage Reference Module .....	171
Compare Mode Operation .....	126
CPU Programmer's Model .....	27
Data Access From Program Space	
Address Generation .....	44
Data EEPROM Addressing with TBLPAG and	
NVM Registers .....	55
Enhanced PWM Mode .....	127
High/Low-Voltage Detect (HLVD) Module .....	173
Individual Comparator Configurations .....	168
MCLR Pin Connections Example .....	22
MSSPx Module (I <sup>2</sup> C Master Mode) .....	137
MSSPx Module (I <sup>2</sup> C Mode) .....	137
MSSPx Module (SPI Mode) .....	136
PIC24F CPU Core .....	26
PIC24F16KL402 Family (General) .....	13
PSV Operation .....	46
PWM Operation (Simplified) .....	126
Recommended Minimum Connections .....	21
Reset System .....	59
Serial Resistor .....	107
Shared I/O Port Structure .....	111
Simplified UARTx .....	149
SPI Master/Slave Connection .....	136
Suggested Placement of Oscillator Circuit .....	23
System Clock .....	95
Table Register Addressing .....	47
Timer2 .....	117
Timer3 .....	119
Timer4 .....	123
Watchdog Timer (WDT) .....	184

### C

C Compilers	
MPLAB XC Compilers .....	188
Capture/Compare/PWM (CCP) .....	125
CCP/ECCP	
CCP I/O Pins .....	125
Timer Selection .....	125
Code Examples	
Data EEPROM Bulk Erase .....	57
Data EEPROM Unlock Sequence .....	53
Erasing a Program Memory Row,	
Assembly Language .....	50
Erasing a Program Memory Row, C Language .....	51
I/O Port Write/Read (Assembly Language) .....	114
I/O Port Write/Read (C Language) .....	114
Initiating a Programming Sequence,	
Assembly Language .....	52
Initiating a Programming Sequence, C Language .....	52
Loading the Write Buffers, Assembly Language .....	51
Loading the Write Buffers, C Language .....	52
PWSAV Instruction Syntax .....	105
Reading Data EEPROM Using the	
TBLRD Command .....	58
Sequence for Clock Switching .....	102
Single-Word Erase .....	56
Single-Word Write to Data EEPROM .....	57
Ultra Low-Power Wake-up Initialization .....	107
Code Protection .....	185
Comparator .....	167
Comparator Voltage Reference .....	171
Configuring .....	171
Configuration Bits .....	175
Core Features .....	9
CPU	
ALU .....	29
Control Registers .....	28
Core Registers .....	26
Programmer's Model .....	25
Customer Change Notification Service .....	257
Customer Notification Service .....	257
Customer Support .....	257

### D

Data EEPROM Memory .....	53
Erasing .....	56
Nonvolatile Memory Registers	
NVMCON .....	53
NVMKEY .....	53
NVMADR(U) .....	55
Operations .....	55
Programming	
Bulk Erase .....	57
Reading Data EEPROM .....	58
Single-Word Write .....	57
Data Memory	
Address Space .....	33
Memory Map .....	33
Near Data Space .....	34
Organization .....	34
SFR Space .....	34
Software Stack .....	43
Space Width .....	33

# PIC24F16KL402 FAMILY

DC Characteristics			
BOR Trip Points .....	202		
Comparator .....	208		
Comparator Voltage Reference .....	208		
Data EEPROM Memory .....	208		
High/Low-Voltage Detect .....	202		
I/O Pin Input Specifications .....	206		
I/O Pin Output Specifications .....	207		
Idle Current (I <sub>IDLE</sub> ) .....	203		
Operating Current (I <sub>DD</sub> ) .....	203		
Power-Down Current (I <sub>PD</sub> ) .....	204, 205		
Program Memory .....	207		
Temperature and Voltage Specifications .....	201		
Demo/Development Boards, Evaluation and Starter Kits .....	190		
Development Support .....	187		
Third-Party Tools .....	190		
Device Features for PIC24F16KL20X/10X Devices (Summary) .....	12		
Device Features for PIC24F16KL40X/30X Devices (Summary) .....	11		
<b>E</b>			
Electrical Characteristics			
Absolute Maximum Ratings .....	199		
Thermal Operating Conditions .....	201		
Thermal Packaging Characteristics .....	201		
V/F Graph, Extended .....	200		
V/F Graph, Industrial .....	200		
Enhanced CCP .....	125		
Equations			
A/D Conversion Clock Period .....	164		
UARTx Baud Rate with BRGH = 0 .....	150		
UARTx Baud Rate with BRGH = 1 .....	150		
Errata .....	7		
Examples			
Baud Rate Error Calculation (BRGH = 0) .....	150		
<b>F</b>			
Flash Program Memory			
Control Registers .....	48		
Enhanced ICSP Operation .....	48		
Programming Algorithm .....	50		
Programming Operations .....	48		
RTSP Operation .....	48		
Table Instructions .....	47		
<b>G</b>			
Getting Started Guidelines for 16-Bit MCUs .....	21		
<b>H</b>			
High/Low-Voltage Detect (HLVD) .....	173		
<b>I</b>			
I/O Ports			
Analog Port Configuration .....	112		
Analog Selection Registers .....	112		
Input Change Notification .....	114		
Open-Drain Configuration .....	112		
Parallel (PIO) .....	111		
In-Circuit Debugger .....	185		
In-Circuit Serial Programming (ICSP) .....	185		
Instruction Set			
Opcode Symbols .....	192		
Overview .....	193		
Summary .....	191		
Inter-Integrated Circuit. See I <sup>2</sup> C.			
Internet Address .....	257		
Interrupt Sources			
TMR3 Overflow .....	119		
TMR4 to PR4 Match (PWM) .....	123		
Interrupts			
Alternate Interrupt Vector Table (AIVT) .....	65		
Control and Status Registers .....	68		
Implemented Vectors .....	67		
Interrupt Vector Table (IVT) .....	65		
Reset Sequence .....	65		
Setup Procedures .....	94		
Trap Vectors .....	67		
Vector Table .....	66		
<b>M</b>			
Master Synchronous Serial Port (MSSP) .....	135		
I/O Pin Configuration for SPI .....	135		
Microchip Internet Web Site .....	257		
MPLAB Assembler, Linker, Librarian .....	188		
MPLAB ICD 3 In-Circuit Debugger .....	189		
MPLAB PM3 Device Programmer .....	189		
MPLAB REAL ICE In-Circuit Emulator System .....	189		
MPLAB X Integrated Development Environment Software .....	187		
MPLAB X SIM Software Simulator .....	189		
MPLIB Object Librarian .....	188		
MPLINK Object Linker .....	188		
<b>N</b>			
Near Data Space .....	34		
<b>O</b>			
Oscillator Configuration			
Clock Switching .....	101		
Sequence .....	101		
Configuration Bit Values for Clock Selection .....	96		
CPU Clocking Scheme .....	96		
Initial Configuration on POR .....	96		
Reference Clock Output .....	102		
Oscillator, Timer3 .....	119		
<b>P</b>			
Packaging			
Details .....	228		
Marking .....	225		
PICkit 3 In-Circuit Debugger/Programmer .....	189		
Pinout Descriptions			
PIC24F16KL20X/10X Devices .....	18		
PIC24F16KL40X/30X Devices .....	14		
Power-Saving .....	109		
Power-Saving Features .....	105		
Clock Frequency, Clock Switching .....	105		
Coincident Interrupts .....	106		
Instruction-Based Modes .....	105		
Idle .....	106		
Sleep .....	106		
Selective Peripheral Control .....	109		
Ultra Low-Power Wake-up (ULPWU) .....	107		
Product Identification System .....	259		
Program and Data Memory			
Access Using Table Instructions .....	45		
Program Space Visibility .....	46		
Program and Data Memory Spaces			
Addressing .....	43		
Interfacing .....	43		

# PIC24F16KL402 FAMILY

Program Memory			
Address Space.....	31	IEC4 (Interrupt Enable Control 4).....	80
Data EEPROM.....	32	IEC5 (Interrupt Enable Control 5).....	80
Device Configuration Words .....	32	IFS0 (Interrupt Flag Status 0).....	73
Hard Memory Vectors .....	32	IFS1 (Interrupt Flag Status 1).....	74
Organization.....	32	IFS2 (Interrupt Flag Status 2).....	75
Program Space		IFS3 (Interrupt Flag Status 3).....	75
Memory Map .....	31	IFS4 (Interrupt Flag Status 4).....	76
Program Verification .....	185	IFS5 (Interrupt Flag Status 5).....	76
PWM (CCP Module)		INTCON 2 (Interrupt Control 2).....	72
TMR4 to PR4 Match .....	123	INTCON1 (Interrupt Control 1).....	71
<b>R</b>		INTTREG (Interrupt Control and Status).....	93
Register Maps		IPC0 (Interrupt Priority Control 0).....	81
A/D Converter .....	41	IPC1 (Interrupt Priority Control 1).....	82
Analog Select.....	41	IPC12 (Interrupt Priority Control 12).....	90
CCP/ECCP .....	38	IPC16 (Interrupt Priority Control 16).....	91
Comparator.....	41	IPC18 (Interrupt Priority Control 18).....	92
CPU Core.....	35	IPC2 (Interrupt Priority Control 2).....	83
ICN.....	36	IPC20 (Interrupt Priority Control 20).....	92
Interrupt Controller.....	37	IPC3 (Interrupt Priority Control 3).....	84
MSSP .....	39	IPC4 (Interrupt Priority Control 4).....	85
NVM.....	42	IPC5 (Interrupt Priority Control 5).....	86
Pad Configuration .....	40	IPC6 (Interrupt Priority Control 6).....	87
PMD.....	42	IPC7 (Interrupt Priority Control 7).....	88
PORTA.....	40	IPC9 (Interrupt Priority Control 9).....	89
PORTB.....	40	NVMCON (Flash Memory Control).....	49
System, Clock Control .....	42	NVMCON (Nonvolatile Memory Control).....	54
Timer.....	38	OSCCON (Oscillator Control).....	97
UART .....	39	OSCTUN (FRC Oscillator Tune) .....	100
Ultra Low-Power Wake-up .....	42	PADCFG1 (Pad Configuration Control).....	147
Registers		PSTR1CON (ECCP1 Pulse Steering Control).....	132
AD1CHS (A/D Input Select).....	162	RCON (Reset Control).....	60
AD1CON1 (A/D Control 1).....	159	REFOCON (Reference Oscillator Control).....	103
AD1CON2 (A/D Control 2).....	160	SR (ALU STATUS).....	28, 69
AD1CON3 (A/D Control 3).....	161	SSPxADD (MSSPx Slave Address/Baud	
AD1CSSL (A/D Input Scan Select).....	163	Rate Generator).....	146
ANCFG (Analog Input Configuration) .....	163	SSPxCON1 (MSSPx Control 1, I <sup>2</sup> C Mode) .....	142
ANSA (PORTA Analog Selection) .....	113	SSPxCON1 (MSSPx Control 1, SPI Mode).....	141
ANSB (PORTB Analog Selection) .....	113	SSPxCON2 (MSSPx Control 2, I <sup>2</sup> C Mode) .....	143
CCP1CON (ECCP1 Control, Enhanced CCP).....	129	SSPxCON3 (MSSPx Control 3, I <sup>2</sup> C Mode) .....	145
CCPTMRS0 (CCP Timer Select Control 0) .....	133	SSPxCON3 (MSSPx Control 3, SPI Mode).....	144
CCPxCON (CCPx Control, Standard CCP).....	128	SSPxMSK (I <sup>2</sup> C Slave Address Mask) .....	146
CLKDIV (Clock Divider) .....	99	SSPxSTAT (MSSPx Status, I <sup>2</sup> C Mode).....	139
CMSTAT (Comparator Status).....	170	SSPxSTAT (MSSPx Status, SPI Mode).....	138
CMxCON (Comparator x Control).....	169	T1CON (Timer1 Control) .....	116
CORCON (CPU Control) .....	29, 70	T2CON (Timer2 Control) .....	118
CVRCON (Comparator Voltage		T3CON (Timer3 Control) .....	120
Reference Control) .....	172	T3GCON (Timer3 Gate Control).....	121
DEVID (Device ID).....	182	T4CON (Timer4 Control) .....	124
DEVREV (Device Revision).....	183	ULPWCON (ULPWU Control) .....	108
ECCP1AS (ECCP1 Auto-Shutdown Control).....	130	UxMODE (UARTx Mode) .....	152
ECCP1DEL (ECCP1 Enhanced PWM Control).....	131	UxSTA (UARTx Status and Control) .....	154
FBS (Boot Segment Configuration) .....	176	Resets	
FGS (General Segment Configuration).....	176	Brown-out Reset (BOR).....	63
FICD (In-Circuit Debugger Configuration).....	181	Clock Source Selection .....	61
FOSC (Oscillator Configuration) .....	178	Delay Times.....	62
FOSCSEL (Oscillator Selection Configuration).....	177	Device Times.....	62
FPOR (Reset Configuration).....	180	RCON Flag Operation .....	61
FWDT (Watchdog Timer Configuration) .....	179	SFR States .....	63
HLVDCON (High/Low-Voltage Detect Control).....	174	Revision History.....	251
IEC0 (Interrupt Enable Control 0) .....	77	<b>S</b>	
IEC1 (Interrupt Enable Control 1) .....	78	Serial Peripheral Interface. See SPI Mode.	
IEC2 (Interrupt Enable Control 2) .....	79	SFR Space .....	34
IEC3 (Interrupt Enable Control 3) .....	79	Software Stack .....	43

# PIC24F16KL402 FAMILY

---

## T

Timer1 .....	115
Timer2 .....	117
Timer3 .....	119
Oscillator .....	119
Overflow Interrupt .....	119
Timer4 .....	123
PR4 Register .....	123
TMR4 Register .....	123
TMR4 to PR4 Match Interrupt .....	123
Timing Diagrams	
Capture/Compare/PWM (ECCP1, ECCP2) .....	214
CLKO and I/O .....	212
Example SPI Master Mode (CKE = 0) .....	215
Example SPI Master Mode (CKE = 1) .....	216
Example SPI Slave Mode (CKE = 0) .....	217
Example SPI Slave Mode (CKE = 1) .....	218
External Clock .....	210
I <sup>2</sup> C Bus Data .....	219
I <sup>2</sup> C Bus Start/Stop Bits .....	219
MSSPx I <sup>2</sup> C Bus Data .....	222
MSSPx I <sup>2</sup> C Bus Start/Stop Bits .....	221
Timing Requirements	
A/D Conversion .....	224
Capture/Compare/PWM (ECCP1, ECCP2) .....	214
CLKO and I/O .....	212
Comparator .....	213
Comparator Voltage Reference Settling Time .....	213
External Clock .....	210
I <sup>2</sup> C Bus Data (Slave Mode) .....	220
I <sup>2</sup> C Bus Data Requirements (Master Mode) .....	222
I <sup>2</sup> C Bus Start/Stop Bits (Master Mode) .....	221
I <sup>2</sup> C Bus Start/Stop Bits (Slave Mode) .....	219
PLL Clock Specifications .....	211
Reset, Watchdog Timer, Oscillator Start-up Timer, Power-up Timer and Brown-out Reset .....	213
SPI Mode (Master Mode, CKE = 0) .....	215
SPI Mode (Master Mode, CKE = 1) .....	216
SPI Slave Mode (CKE = 1) .....	218
Timing Requirements SPI Mode (Slave Mode, CKE = 0) .....	217

## U

UART .....	149
Baud Rate Generator (BRG) .....	150
Break and Sync Transmit Sequence .....	151
IrDA Support .....	151
Operation of UxCTS and UxRTS Control Pins .....	151
Receiving in 8-Bit or 9-Bit Data Mode .....	151
Transmitting in 8-Bit Data Mode .....	151
Transmitting in 9-Bit Data Mode .....	151
Unique ID .....	182

## W

Watchdog Timer (WDT) .....	184
Windowed Operation .....	184
WWW Address .....	257
WWW, On-Line Support .....	7

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com). Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://microchip.com/support>**

# PIC24F16KL402 FAMILY

---

NOTES:

# PIC24F16KL402 FAMILY

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

	PIC	24	F	16	KL4	02	T	- I / PT	- XXX
<b>Microchip Trademark</b>	_____								
<b>Architecture</b>	_____								
<b>Flash Memory Family</b>	_____								
<b>Program Memory Size (Kbytes)</b>	_____								
<b>Product Group</b>	_____								
<b>Pin Count</b>	_____								
<b>Tape and Reel Flag (if applicable)</b>	_____								
<b>Temperature Range</b>	_____								
<b>Package</b>	_____								
<b>Pattern</b>	_____								

<b>Architecture</b>	24	= 16-bit modified Harvard without DSP
<b>Flash Memory Family</b>	F	= Standard voltage range Flash program memory
<b>Product Group</b>	KL4	= General purpose microcontrollers
	KL3	
	KL2	
	KL1	
<b>Pin Count</b>	00	= 14-pin
	01	= 20-pin
	02	= 28-pin
<b>Temperature Range</b>	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
<b>Package</b>	SP	= SPDIP
	SO	= SOIC
	SS	= SSOP
	ST	= TSSOP
	ML, MQ	= QFN
	P	= PDIP
<b>Pattern</b>	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise)	
	ES	= Engineering Sample

### Examples:

- PIC24F16KL402-I/ML: General Purpose, 16-Kbyte Program Memory, 28-Pin, Industrial Temperature, QFN Package
- PIC24F04KL101T-I/SS: General Purpose, 4-Kbyte Program Memory, 20-Pin, Industrial Temperature, SSOP Package, Tape-and-Reel

# PIC24F16KL402 FAMILY

---

NOTES:

---

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

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

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

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.


Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICTail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2011-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-62077-620-9

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

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



# MICROCHIP

## Worldwide Sales and Service

### AMERICAS

**Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://www.microchip.com/support>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

**Atlanta**  
Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

**Austin, TX**  
Tel: 512-257-3370

**Boston**  
Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

**Chicago**  
Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

**Cleveland**  
Independence, OH  
Tel: 216-447-0464  
Fax: 216-447-0643

**Dallas**  
Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

**Detroit**  
Novi, MI  
Tel: 248-848-4000

**Houston, TX**  
Tel: 281-894-5983

**Indianapolis**  
Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453

**Los Angeles**  
Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

**New York, NY**  
Tel: 631-435-6000

**San Jose, CA**  
Tel: 408-735-9110

**Canada - Toronto**  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

**Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**China - Beijing**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**China - Chengdu**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**China - Chongqing**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**China - Hangzhou**  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

**China - Hong Kong SAR**  
Tel: 852-2943-5100  
Fax: 852-2401-3431

**China - Nanjing**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**China - Qingdao**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**China - Shanghai**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**China - Shenyang**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**China - Shenzhen**  
Tel: 86-755-8864-2200  
Fax: 86-755-8203-1760

**China - Wuhan**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**China - Xian**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**China - Xiamen**  
Tel: 86-592-2388138  
Fax: 86-592-2388130

**China - Zhuhai**  
Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

**India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**India - Pune**  
Tel: 91-20-3019-1500

**Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

**Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**Taiwan - Hsin Chu**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7830

**Taiwan - Taipei**  
Tel: 886-2-2508-8600  
Fax: 886-2-2508-0102

**Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

**Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**Denmark - Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**Germany - Dusseldorf**  
Tel: 49-2129-3766400

**Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**Germany - Pforzheim**  
Tel: 49-7231-424750

**Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**Italy - Venice**  
Tel: 39-049-7625286

**Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**Poland - Warsaw**  
Tel: 48-22-3325737

**Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91



**Sweden - Stockholm**  
Tel: 46-8-5090-4654

**UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820





10/28/13

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

-  [View PIC24F08KL402-I/SS on WIN SOURCE](#)
-  [Microchip Technology](#) Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

-  Global Sourcing Solution
-  Obsolete Management
-  Cost Control Management
-  Shortage Management
-  Alternative Solution
-  Excess Inventory Management