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 μA/MHz typical at 1.8V
  - Idle mode currents under 80 μA/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
  - One, two or four PWM outputs
  - Programmable dead time
  - Auto-shutdown on external event
- Up to Two Master Synchronous Serial Port modules (MSSPs) with Two Modes of Operation:
  - Three-wire SPI (all four modes)
  - I^2C 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

Analog Features:

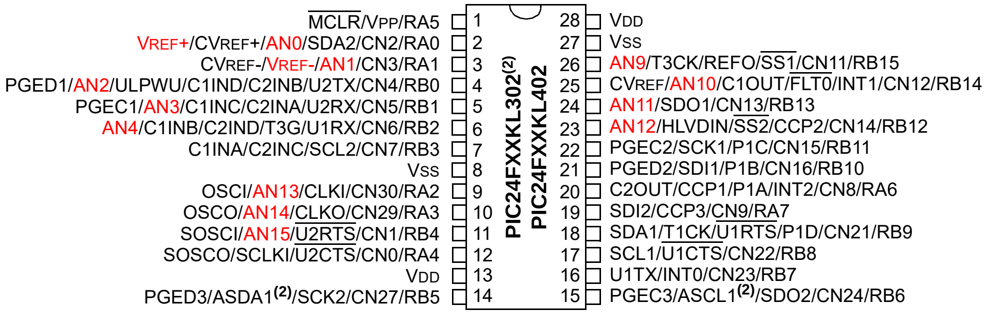
- 10-Bit, Up to 12-Channel Analog-to-Digital (A/D) Converter:
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Dual Rail-to-Rail Analog Comparators with Programmable Input/Output Configuration
- On-Chip Voltage Reference

Special Microcontroller Features:

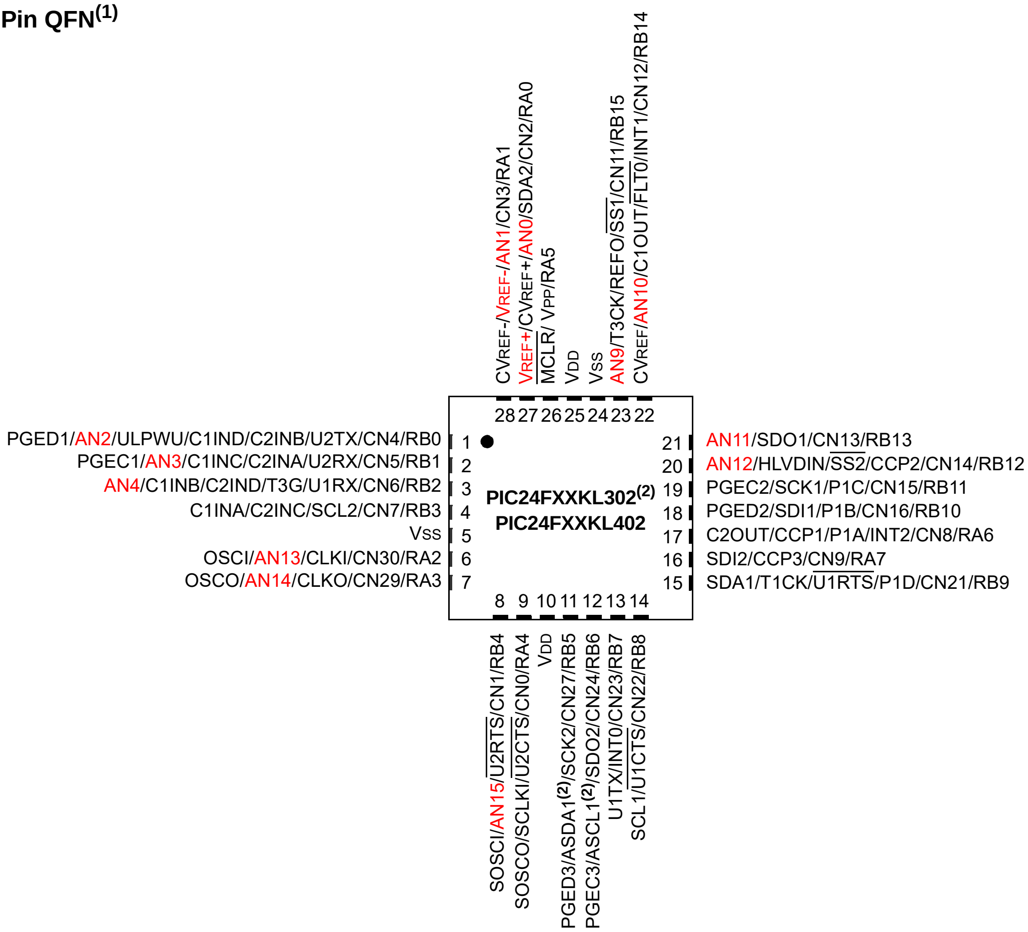
- Operating Voltage Range of 1.8V to 3.6V
- 10,000 Erase/Write Cycle Endurance Flash Program Memory, Typical
- 100,000 Erase/Write Cycle Endurance Data EEPROM, Typical
- Flash and Data EEPROM Data Retention: 40 Years Minimum
- Self-Programmable under Software Control
- Programmable Reference Clock Output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

Pin Diagrams

28-Pin SPDIP/SSOP/SOIC[1]



28-Pin QFN[1]



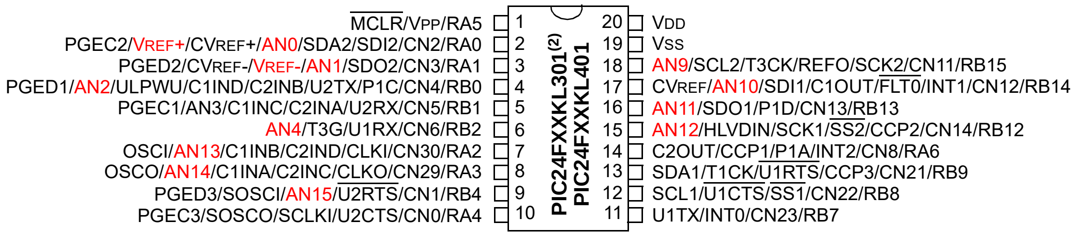
Contact your Microchip sales team for Chip Scale Package (CSP) availability.

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

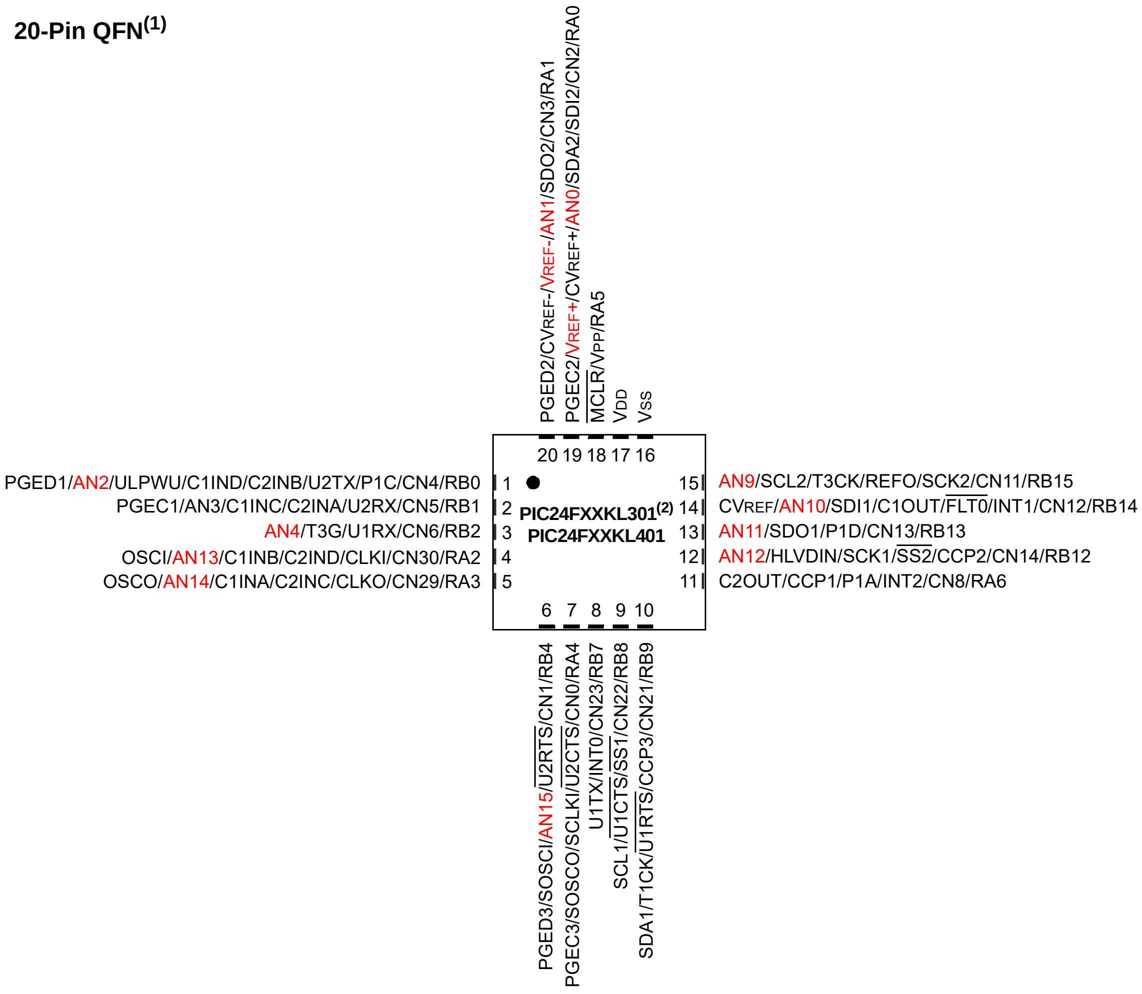
2: Alternate location for I^{2}C functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

Pin Diagrams (Continued)

20-Pin PDIP/SSOP/SOIC[1]



20-Pin QFN[1]

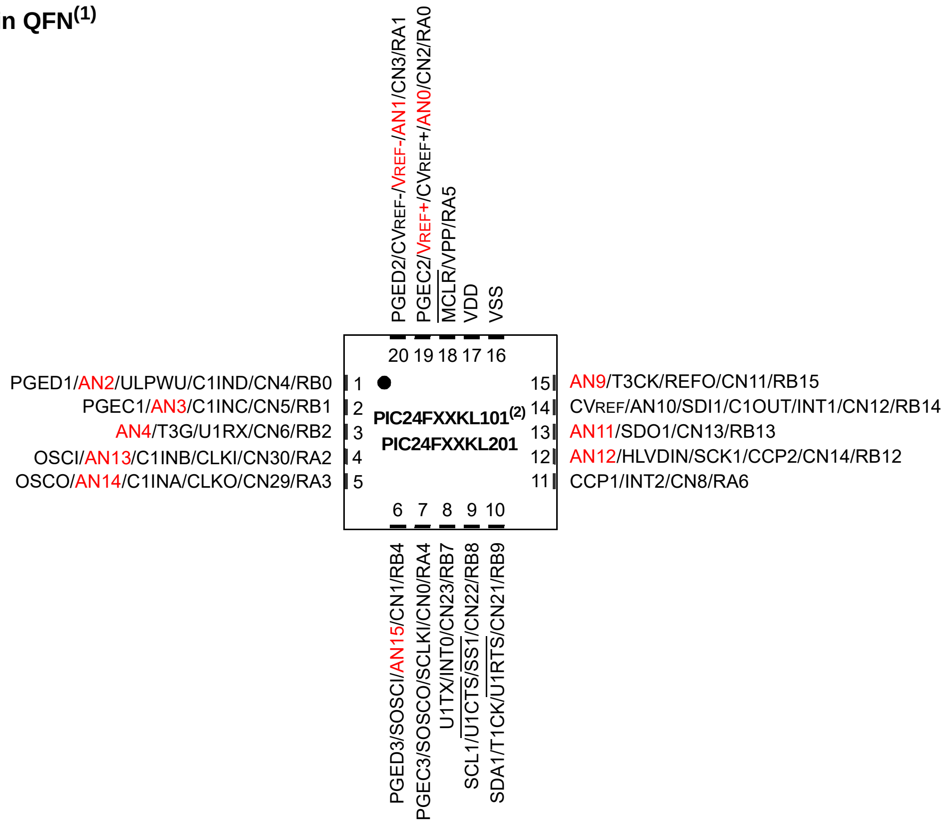


Note 1: Analog features (indicated in red) are not available on PIC24FXXKL301 devices.

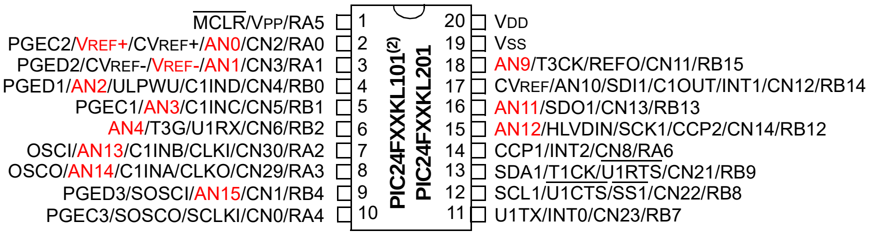
2: Alternate location for I^{2}C functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

Pin Diagrams (Continued)

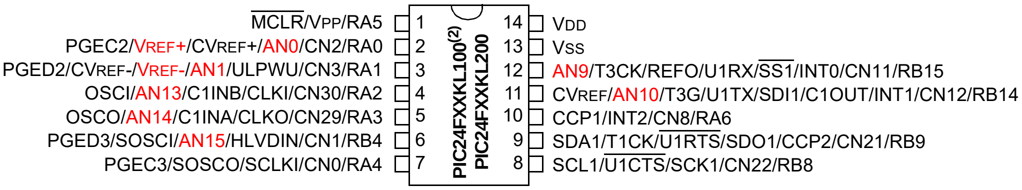
20-Pin QFN(1)



20-Pin PDIP/SSOP/SOIC(1)



14-Pin PDIP/TSSOP(1)



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

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

Table of Contents

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**. 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 Website 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 Website; **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 website at **www.microchip.com** to receive the most current information on all of our products.

NOTES:

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.

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^{2}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

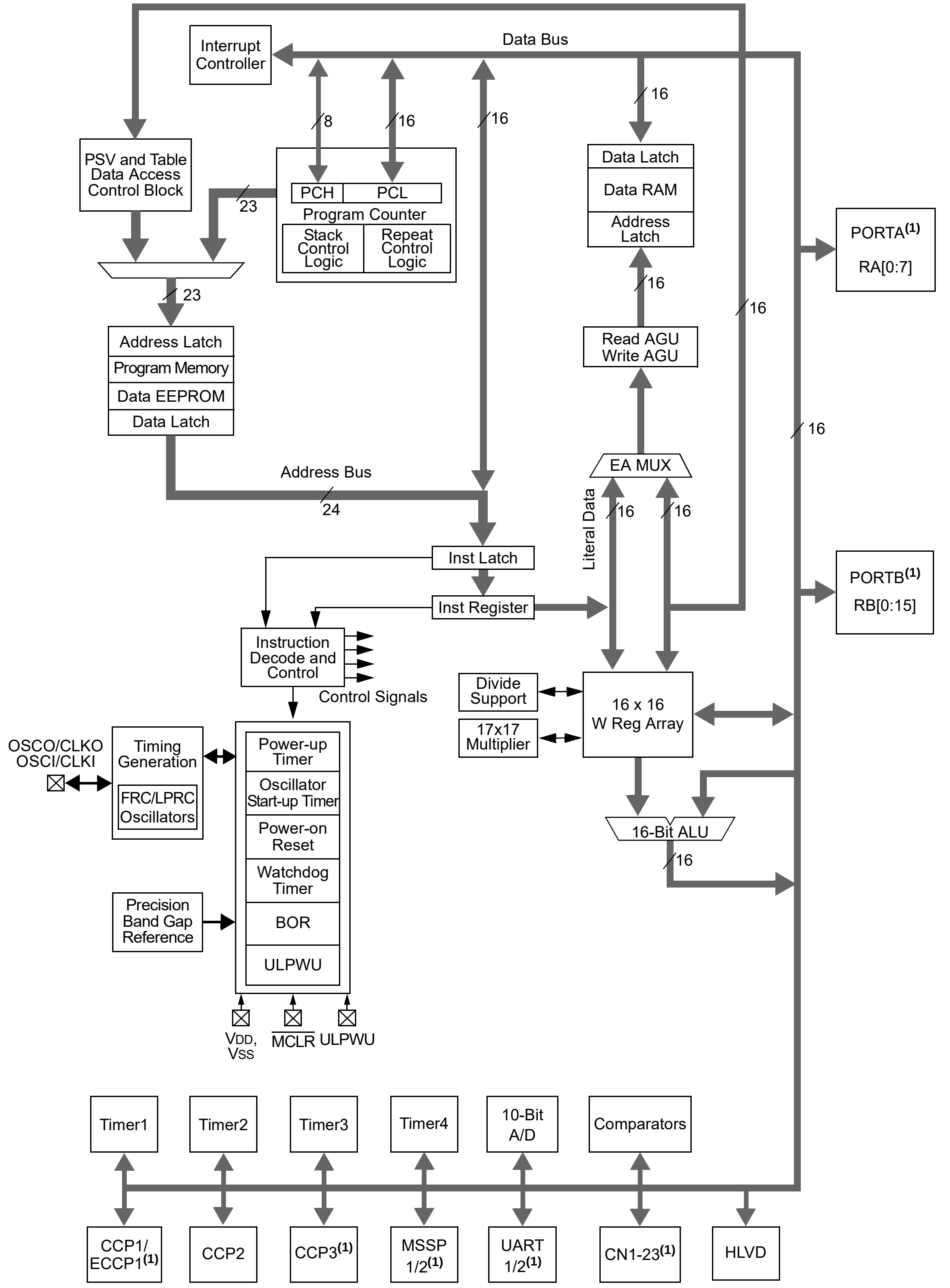
TABLE 1-2: DEVICE FEATURES FOR PIC24F16KL40X/30X DEVICES

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, $\overline{\text{MCLR}}$, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)					
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations					
Packages	28-Pin SPDIP/SSOP/SOIC/QFN			20-Pin PDIP/SSOP/SOIC/QFN		

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	

FIGURE 1-1: PIC24F16KL402 FAMILY GENERAL BLOCK DIAGRAM



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

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^{2}C	Alternate MSSP1 I^{2}C Clock Input/Output
ASDA1	—	—	14	11	I/O	I^{2}C	Alternate MSSP1 I^{2}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^{2}C = I^{2}C/SMBus input buffer

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
$\overline{\text{FLT0}}$	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
$\overline{\text{MCLR}}$	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
ST = Schmitt Trigger input buffer
ANA = Analog level input/output
I^2C = I^2C/SMBus input buffer

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^2C	MSSP1 I^2C Clock Input/Output
SCL2	18	15	7	4	I/O	I^2C	MSSP2 I^2C Clock Input/Output
SCLKI	10	7	12	9	I	ST	Digital Secondary Clock Input
SDA1	13	10	18	15	I/O	I^2C	MSSP1 I^2C Data Input/Output
SDA2	2	19	2	27	I/O	I^2C	MSSP2 I^2C 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^2C = I^2C/SMBus input buffer

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
$\overline{\text{SS1}}$	12	9	26	23	O	—	SPI1 Slave Select
$\overline{\text{SS2}}$	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^2C = I^2C/SMBus input buffer

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
ST = Schmitt Trigger input buffer
ANA = Analog level input/output
I^2C = I^2C/SMBus input buffer

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
$\overline{\text{MCLR}}$	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 ST = Schmitt Trigger input buffer
ANA = Analog level input/output I^2C = I^2C/SMBus input buffer

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^2C	MSSP1 I^2C Clock Input/Output
SCLKI	10	7	12	I	ST	Digital Secondary Clock Input
SDA1	13	10	9	I/O	I^2C	MSSP1 I^2C 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
$\overline{\text{SS1}}$	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
$\overline{\text{U1CTS}}$	12	9	8	I	ST	UART1 Clear-to-Send Input
$\overline{\text{U1RTS}}$	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
ST = Schmitt Trigger input buffer
ANA = Analog level input/output
I^2C = I^2C/SMBus input buffer

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”**)
- $\overline{\text{MCLR}}$ 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”**)

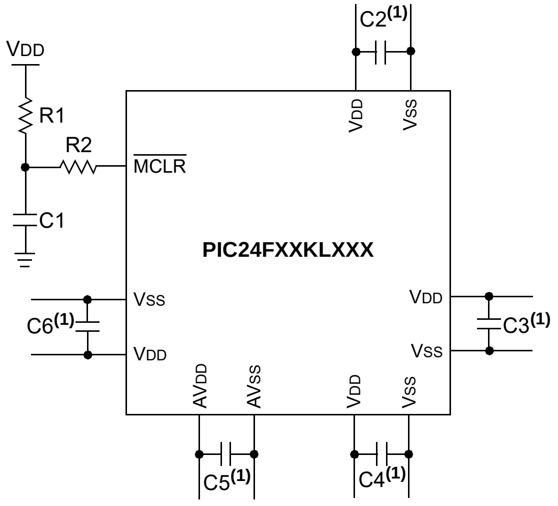
Additionally, the following pins may be required:

- 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



Key (all values are recommendations):

C1 through C6: 0.1 µF, 20V ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: The example shown is for a PIC24F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

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 µF (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 µF to 0.001 µF. 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 µF in parallel with 0.001 µF).
- **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 µF to 47 µF.

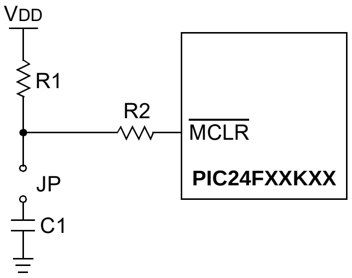
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

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 (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ 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



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor, C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2.4 ICSP Pins

The PGECx and PGEDx 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 PGECx and PGEDx 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_{IH}) and Input Voltage Low (V_{IL}) requirements.

For device emulation, ensure that the “Communication Channel Select” (i.e., PGECx/PGEDx) 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

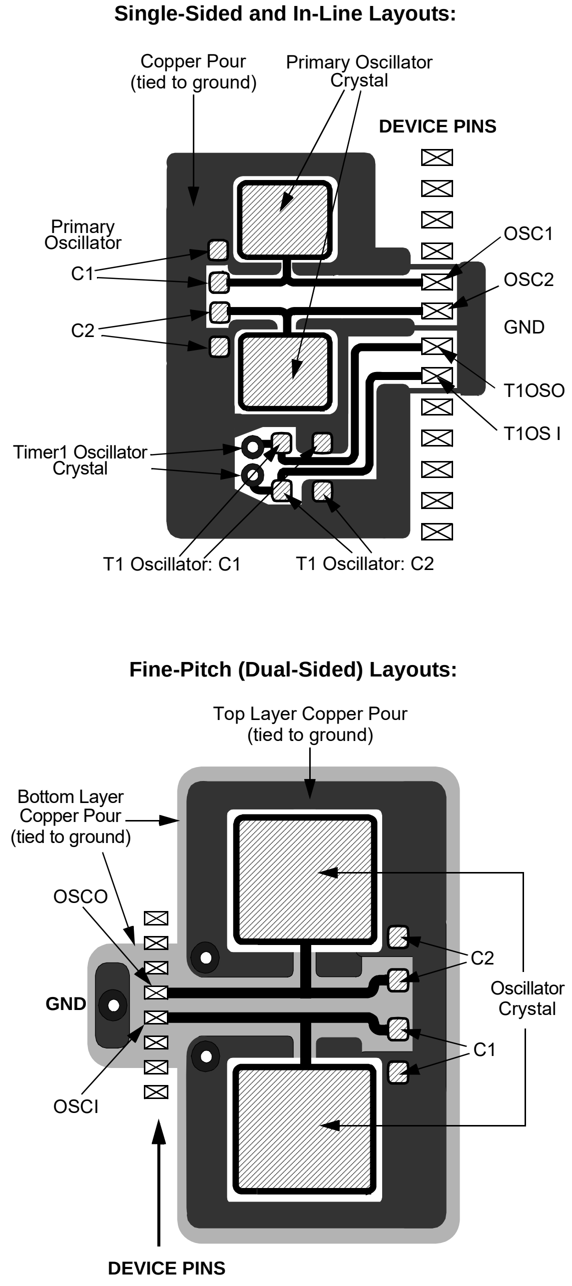
Many microcontrollers have options for at least two oscillators: a high-frequency Primary Oscillator and a low-frequency Secondary Oscillator (refer to **Section 9.0 “Oscillator Configuration”** for details).

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).

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

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

2.6 Unused I/Os

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

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 **"CPU"** (www.microchip.com/DS39703) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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 16th 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 trinary 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.

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM

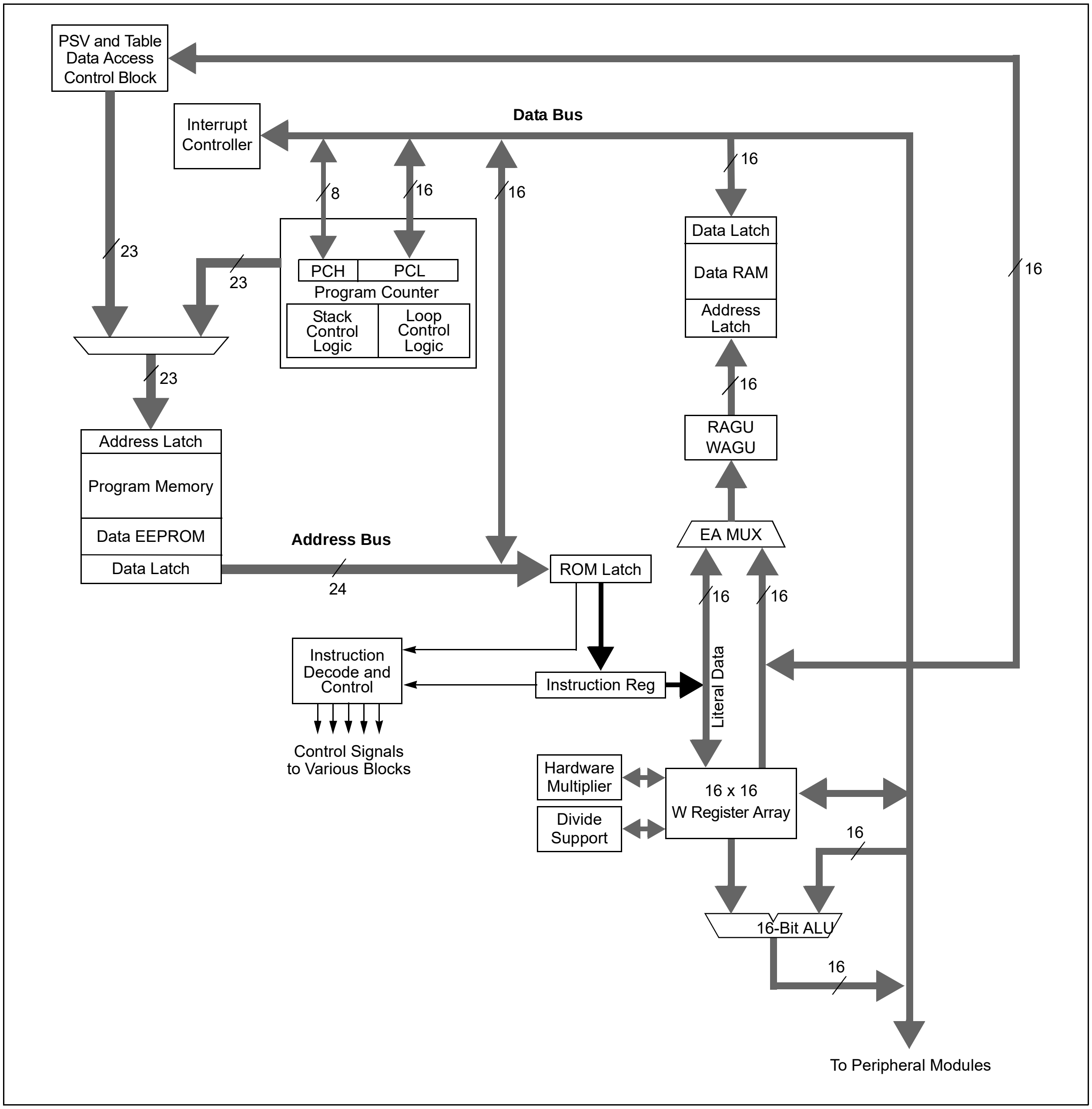
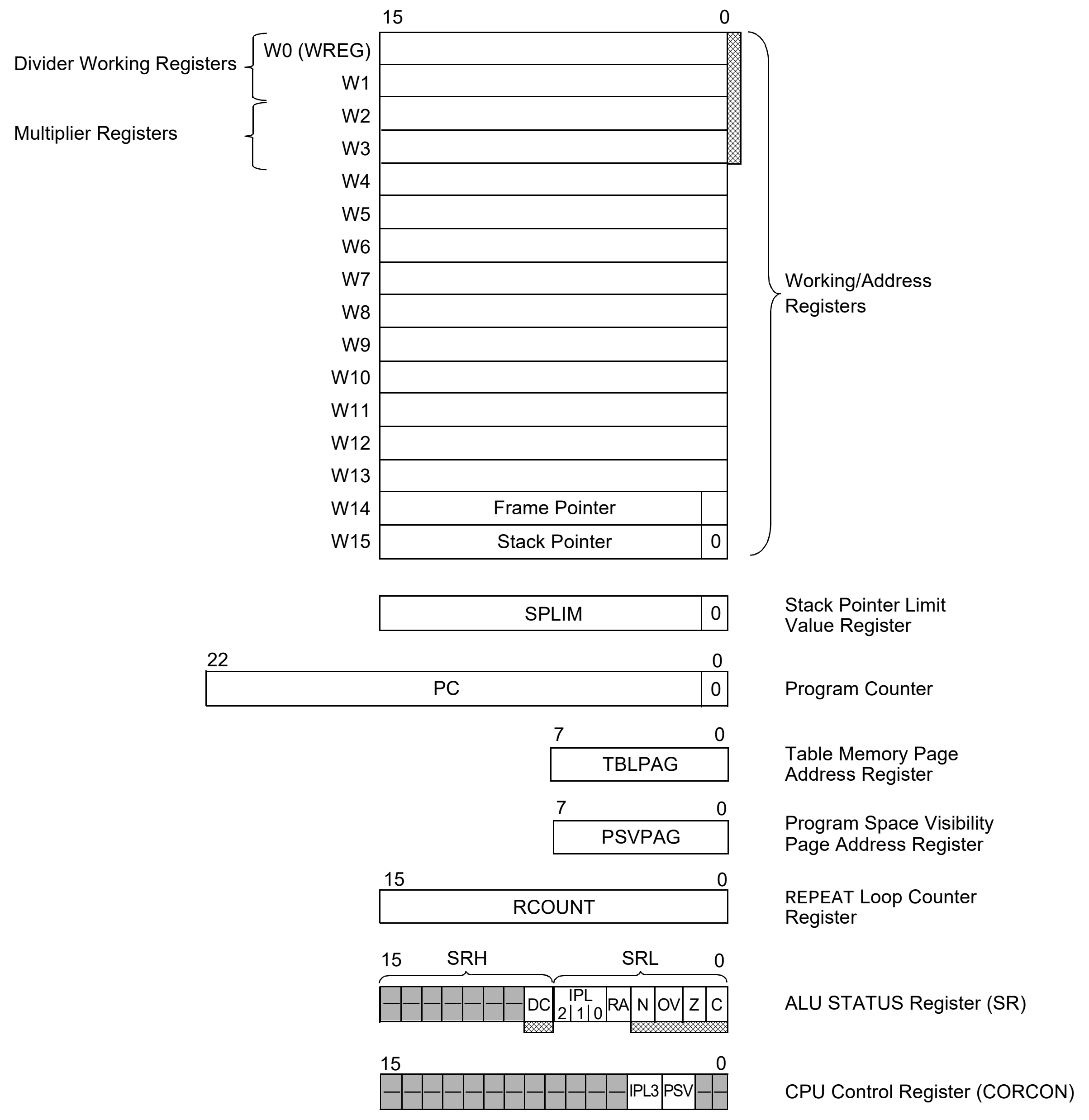


TABLE 3-1: CPU CORE REGISTERS

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

FIGURE 3-2: PROGRAMMER'S MODEL



▨ Registers or bits are shadowed for `PUSH.S` and `POP.S` instructions.

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(1)	R/W-0(1)	R/W-0(1)	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2(2)	IPL1(2)	IPL0(2)	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 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred 0 = No carry-out from the 4th or 8th low-order bit of the result has occurred
bit 7-5	**IPL[2:0]:** CPU Interrupt Priority Level (IPL) Status bits(1,2) 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 (two’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/$\overline{\text{Borrow}}$ 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.

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.

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(1)	PSV	—	—
bit 7							bit 0

Legend:	C = 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 15-4 **Unimplemented:** Read as ‘`0`’

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
`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 two’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

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 MULTIBIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multibit arithmetic and logic shifts. Multibit 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 multibit 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 MULTIBIT 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.

4.0 MEMORY ORGANIZATION

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

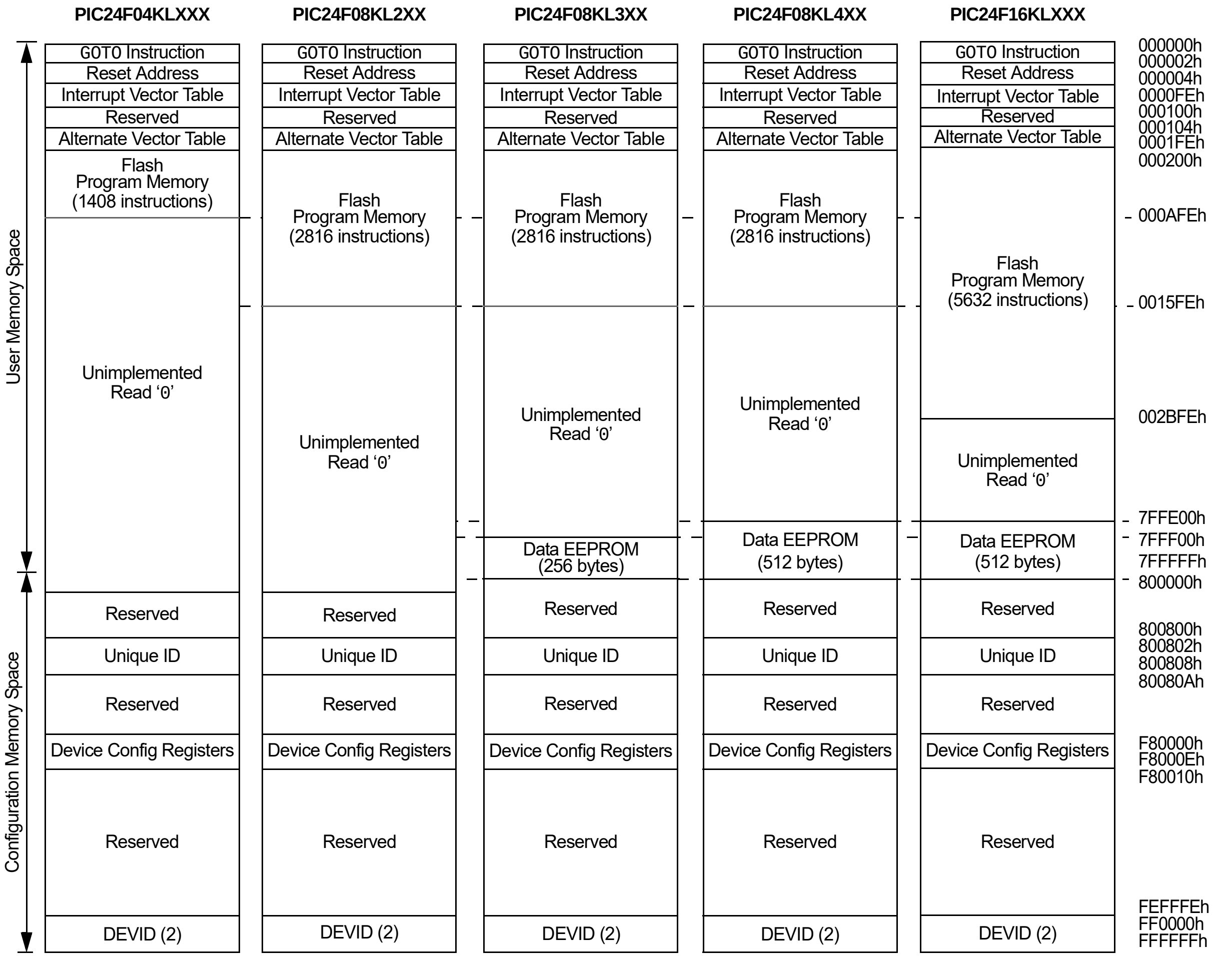
4.1 Program Address Space

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”**.

User access to the program memory space is restricted to the lower half of the address range (000000h to 7FFFFFh). 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.

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



Note: Memory areas are not displayed to scale.

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

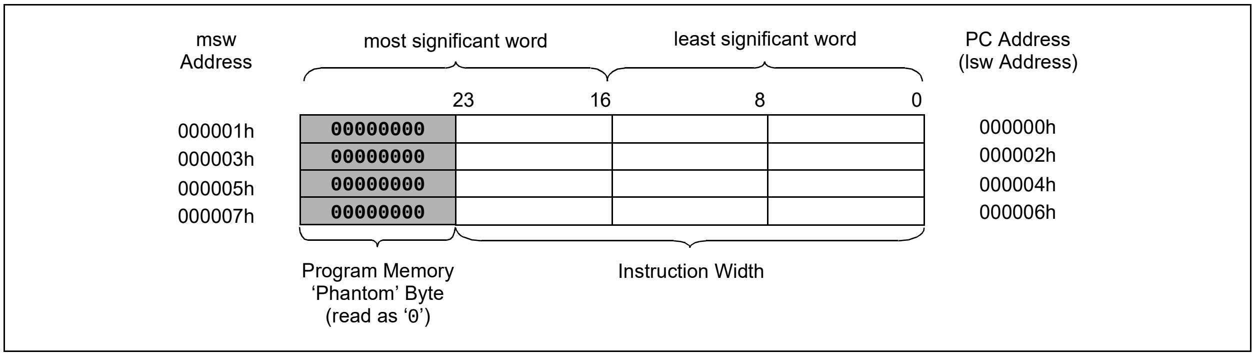
Table 4-1 provides the addresses of the device Configuration Words for the PIC24F16KL402 family. Their location in the memory map is shown in Figure 4-1.

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



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.

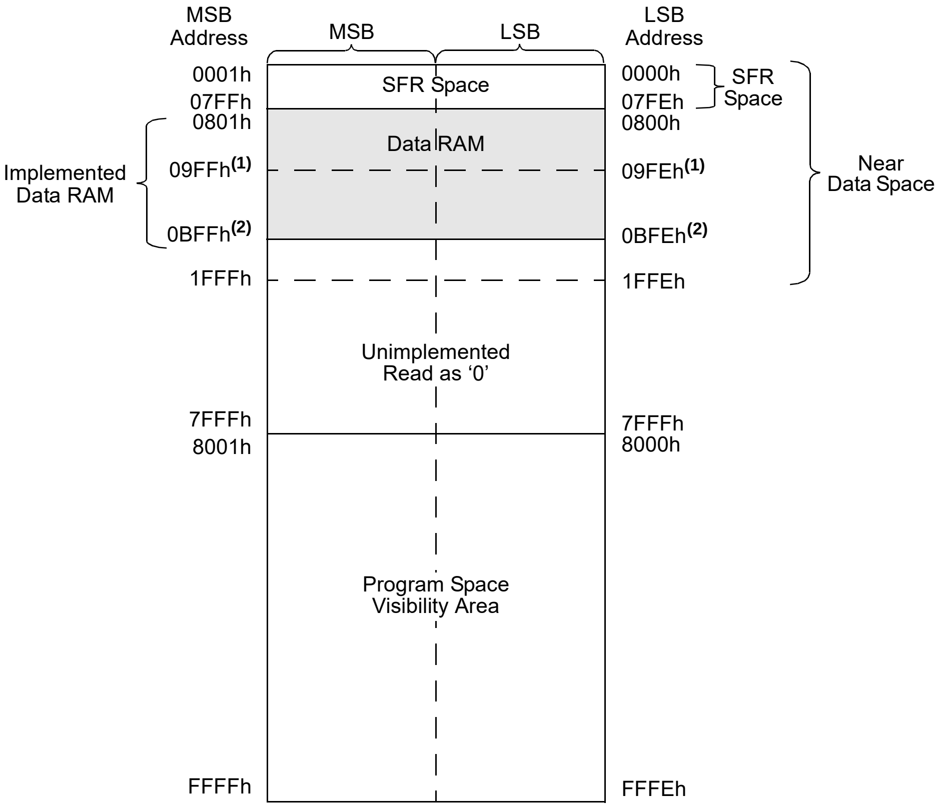
All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the Data Space. This gives a Data Space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when EA[15] = `0`) is used for implemented memory addresses, while the upper 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 are 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[3]



Note 1: Upper data memory boundary for PIC24FXXKL10X/20X devices.
2: Upper data memory boundary for PIC24FXXKL30X/40X devices.
3: Data memory areas are not shown to scale.

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. Misaligned 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 misaligned 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

<table>
<tr><th rowspan="2"></th><th colspan="16">SFR Space Address</th></tr>
<tr><th colspan="2">xx00</th><th colspan="2">xx20</th><th colspan="2">xx40</th><th colspan="2">xx60</th><th colspan="2">xx80</th><th colspan="2">xxA0</th><th colspan="2">xxC0</th><th colspan="2">xxE0</th></tr>
<tr><td>000h</td><td colspan="6">Core</td><td colspan="2">ICN</td><td colspan="6">Interrupts</td><td colspan="2">—</td></tr>
<tr><td>100h</td><td colspan="2">Timers</td><td>—</td><td>TMR</td><td colspan="2">—</td><td colspan="2">—</td><td>—</td><td colspan="2">CCP</td><td>—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>200h</td><td colspan="2">MSSP</td><td colspan="2">UART</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">I/O</td><td colspan="2">—</td></tr>
<tr><td>300h</td><td colspan="4">A/D</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>400h</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td>—</td><td colspan="2">ANSEL</td><td>—</td></tr>
<tr><td>500h</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>600h</td><td colspan="2">—</td><td>CMP</td><td>—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
<tr><td>700h</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">System</td><td colspan="2">NVM/PMD</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td><td colspan="2">—</td></tr>
</table>

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	Bit 1	Bit 0	All Resets
WREG0	0000	Working Register 0																0000
WREG1	0002	Working Register 1																0000
WREG2	0004	Working Register 2																0000
WREG3	0006	Working Register 3																0000
WREG4	0008	Working Register 4																0000
WREG5	000A	Working Register 5																0000
WREG6	000C	Working Register 6																0000
WREG7	000E	Working Register 7																0000
WREG8	0010	Working Register 8																0000
WREG9	0012	Working Register 9																0000
WREG10	0014	Working Register 10																0000
WREG11	0016	Working Register 11																0000
WREG12	0018	Working Register 12																0000
WREG13	001A	Working Register 13																0000
WREG14	001C	Working Register 14																0000
WREG15	001E	Working Register 15															—	0800
SPLIM	0020	Stack Pointer Limit Value Register																xxxx
PCL	002E	Program Counter Low Word Register																0000
PCH	0030	—	—	—	—	—	—	—	—	—	Program Counter Register High Byte							0000
TBLPAG	0032	—	—	—	—	—	—	—	—	Table Memory Page Address Register								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	Program Space Visibility Page Address Register								0000
RCOUNT	0036	REPEAT Loop Counter Register																xxxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL[2:0]			RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	Disable Interrupts Counter Register														xxxx

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 2	Bit 1	Bit 0	All Resets
CNPD1	0056	CN15PDE(1)	CN14PDE(1)	CN13PDE(1)	CN12PDE	CN11PDE	—	CN9PDE(2)	CN8PDE	CN7PDE(2)	CN6PDE(1)	CN5PDE(1)	CN4PDE(1)	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	—	CN30PDE	CN29PDE	—	CN27PDE(2)	—	—	CN24PDE(2)	CN23PDE(1)	CN22PDE	CN21PDE	—	—	—	—	CN16PDE(2)	0000
CNEN1	0062	CN15IE(1)	CN14IE(1)	CN13IE(1)	CN12IE	CN11IE	—	CN9IE(1)	CN8IE	CN7IE(1)	CN6IE(2)	CN5PIE(2)	CN4IE(2)	CN3IE	CNIE	CN1IE	CN0IE	0000
CNEN2	0064	—	CN30IE	CN29IE	—	CN27IE(2)	—	—	CN24IE(2)	CN23IE(1)	CN22IE	CN21IE	—	—	—	—	CN16IE(2)	0000
CNPU1	006E	CN15PUE(1)	CN14PUE(1)	CN13PUE(1)	CN12PUE	CN11PUE	—	CN9PUE(1)	CN8PUE	CN7PUE(1)	CN6PUE(2)	CN5PUE(2)	CN4PUE(2)	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	—	CN30PUE	CN29PUE	—	CN27PUE(2)	—	—	CN24PUE(2)	CN23PUE(1)	CN22PUE	CN21PUE	—	—	—	—	CN16PUE(2)	0000

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	Bit 1	Bit 0	All Resets
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	NVMIF	—	AD1IF	U1TXIF	U1RXIF	—	—	T3IF	T2IF	CCP2IF	—	—	T1IF	CCP1IF	—	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	—	T4IF(1)	—	CCP3IF(1)	—	—	—	—	INT1IF	CNIF	CMIF	BCL1IF	SSP1IF	0000
IFS2	0088	—	—	—	—	—	—	—	—	—	—	T3GIF	—	—	—	—	—	0000
IFS3	008A	—	—	—	—	—	—	—	—	—	—	—	—	—	BCL2IF(1)	SSP2IF(1)	—	0000
IFS4	008C	—	—	—	—	—	—	—	HLVDIF	—	—	—	—	—	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIF	0000
IEC0	0094	NVMIE	—	AD1IE	U1TXIE	U1RXIE	—	—	T3IE	T2IE	CCP2IE	—	—	T1IE	CCP1IE	—	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	—	T4IE(1)	—	CCP3IE(1)	—	—	—	—	INT1IE	CNIE	CMIE	BCL1IE	SSP1IE	0000
IEC2	0098	—	—	—	—	—	—	—	—	—	—	T3GIE	—	—	—	—	—	0000
IEC3	009A	—	—	—	—	—	—	—	—	—	—	—	—	—	BCL2IE(1)	SSP2IE(1)	—	0000
IEC4	009C	—	—	—	—	—	—	—	HLVDIE	—	—	—	—	—	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIE	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0	—	—	—	—	—	INT0IP2	INT0IP1	INT0IP0	4404
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	CCP2IP2	CCP2IP1	CCP2IP0	—	—	—	—	—	—	—	—	4400
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	—	—	—	—	—	—	—	—	T3IP2	T3IP1	T3IP0	4004
IPC3	00AA	—	NVMIP2	NVMIP1	NVMIP0	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	4044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	BCL1IP2	BCL1IP1	BCL1IP0	—	SSP1IP2	SSP1IP1	SS1IP0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	0004
IPC6	00B0	—	T4IP2(1)	T4IP1(1)	T4IP0(1)	—	—	—	—	—	CCP3IP2(1)	CCP3IP1(1)	CCP3IP0(1)	—	—	—	—	4040
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	—	—	—	4440
IPC9	00B6	—	—	—	—	—	—	—	—	—	T3GIP2	T3GIP1	T3GIP0	—	—	—	—	0040
IPC12	00BC	—	—	—	—	—	BCL2IP2(1)	BCL2IP1(1)	BCL2IP0(1)	—	SSP2IP2(1)	SSP2IP1(1)	SSP2IP0(1)	—	—	—	—	0440
IPC16	00C4	—	—	—	—	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	0440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	HLVDIP2	HLVDIP1	HLVDIP0	0004
IPC20	00CC	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIP2	ULPWUIP1	ULPWUIP0	0004
INTTREG	00E0	CPUIRQ	r	VHOLD	—	ILR[3:0]				—	VECNUM[6:0]							0000

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	Bit 1	Bit 0	All Resets
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	T1ECS[1:0]		—	TGATE	TCKPS[1:0]		—	TSYNC	TCS	—	0000
TMR2	0106	—	—	—	—	—	—	—	—	Timer2 Register								0000
PR2	0108	—	—	—	—	—	—	—	—	Timer2 Period Register								00FF
T2CON	010A	—	—	—	—	—	—	—	—	—	T2OUTPS[3:0]				TMR2ON	T2CKPS[1:0]		0000
TMR3	010C	Timer3 Register																0000
T3GCON	010E	—	—	—	—	—	—	—	—	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{T3DONE}}$	T3GVAL	T3GSS[1:0]		0000
T3CON	0110	—	—	—	—	—	—	—	—	TMR3CS[1:0]		T3CKPS[1:0]		T3OSCEN	$\overline{\text{T3SYNC}}$	—	TMR3ON	0000
TMR4[1]	0112	—	—	—	—	—	—	—	—	Timer4 Register								0000
PR4[1]	0114	—	—	—	—	—	—	—	—	Timer4 Period Register								00FF
T4CON[1]	0116	—	—	—	—	—	—	—	—	—	T4OUTPS[3:0]				TMR4ON	T4CKPS[1:0]		0000
CCPTMRS0[1]	013C	—	—	—	—	—	—	—	—	—	C3TSEL0[1]	—	—	C2TSEL0	—	—	C1TSEL0	0000

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 2	Bit 1	Bit 0	All Resets
CCP1CON	0190	—	—	—	—	—	—	—	—	PM[1:0][1]		DC1B[1:0]		CCP1M[3:0]				0000
CCPR1L	0192	—	—	—	—	—	—	—	—	Capture/Compare/PWM1 Register Low Byte								0000
CCPR1H	0194	—	—	—	—	—	—	—	—	Capture/Compare/PWM1 Register High Byte								0000
ECCP1DEL[1]	0196	—	—	—	—	—	—	—	—	PRSEN	PDC[6:0]							0000
ECCP1AS[1]	0198	—	—	—	—	—	—	—	—	ECCPASE	ECCPAS[2:0]			PSSAC[1:0]		PSSBD[1:0]		0000
PSTR1CON[1]	019A	—	—	—	—	—	—	—	—	CMPL[1:0]		—	STRSYNC	STRD	STRC	STRB	STRA	0001
CCP2CON	019C	—	—	—	—	—	—	—	—	—	—	DC2B[1:0]		CCP2M[3:0]				0000
CCPR2L	019E	—	—	—	—	—	—	—	—	Capture/Compare/PWM2 Register Low Byte								0000
CCPR2H	01A0	—	—	—	—	—	—	—	—	Capture/Compare/PWM2 Register High Byte								0000
CCP3CON[1]	01A8	—	—	—	—	—	—	—	—	—	—	DC3B[1:0]		CCP3M[3:0]				0000
CCPR3L[1]	01AA	—	—	—	—	—	—	—	—	Capture/Compare/PWM3 Register Low Byte								0000
CCPR3H[1]	01AC	—	—	—	—	—	—	—	—	Capture/Compare/PWM3 Register High Byte								0000

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 2	Bit 1	Bit 0	All Resets
SSP1BUF	0200	—	—	—	—	—	—	—	—	MSSP1 Receive Buffer/Transmit Register								`00xx`
SSP1CON1	0202	—	—	—	—	—	—	—	—	WCOL	SSPOV	SSPEN	CKP	SSPM[3:0]				`0000`
SSP1CON2	0204	—	—	—	—	—	—	—	—	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	`0000`
SSP1CON3	0206	—	—	—	—	—	—	—	—	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	`0000`
SSP1STAT	0208	—	—	—	—	—	—	—	—	SMP	CKE	$D/\overline{A}$	P	S	$R/\overline{W}$	UA	BF	`0000`
SSP1ADD	020A	—	—	—	—	—	—	—	—	MSSP1 Address Register (I^2C Slave Mode) MSSP1 Baud Rate Reload Register (I^2C Master Mode)								`0000`
SSP1MSK	020C	—	—	—	—	—	—	—	—	MSSP1 Address Mask Register (I^2C Slave Mode)								`00FF`
SSP2BUF(1)	0210	—	—	—	—	—	—	—	—	MSSP2 Receive Buffer/Transmit Register								`00xx`
SSP2CON1(1)	0212	—	—	—	—	—	—	—	—	WCOL	SSPOV	SSPEN	CKP	SSPM[3:0]				`0000`
SSP2CON2(1)	0214	—	—	—	—	—	—	—	—	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	`0000`
SSP2CON3(1)	0216	—	—	—	—	—	—	—	—	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	`0000`
SSP2STAT(1)	0218	—	—	—	—	—	—	—	—	SMP	CKE	$D/\overline{A}$	P	S	$R/\overline{W}$	UA	BF	`0000`
SSP2ADD(1)	021A	—	—	—	—	—	—	—	—	MSSP2 Address Register (I^2C Slave Mode) MSSP2 Baud Rate Reload Register (I^2C Master Mode)								`0000`
SSP2MSK(1)	021C	—	—	—	—	—	—	—	—	MSSP2 Address Mask Register (I^2C Slave Mode)								`00FF`

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	Bit 1	Bit 0	All Resets
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN[1:0]		WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL[1:0]		STSEL	`0000`
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL[1:0]		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	`0110`
U1TXREG	0224	—	—	—	—	—	—	—	UART1 Transmit Register									`xxxx`
U1RXREG	0226	—	—	—	—	—	—	—	UART1 Receive Register									`0000`
U1BRG	0228	Baud Rate Generator Prescaler Register																`0000`
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN[1:0]		WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL[1:0]		STSEL	`0000`
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL[1:0]		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	`0110`
U2TXREG	0234	—	—	—	—	—	—	—	UART2 Transmit Register									`xxxx`
U2RXREG	0236	—	—	—	—	—	—	—	UART2 Receive Register									`0000`
U2BRG	0238	Baud Rate Generator Prescaler Register																`0000`

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[1]	Bit 6	Bit 5[2]	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISA	02C0	—	—	—	—	—	—	—	—	TRISA[7:6]		—	TRISA[4:0]					00DF
PORTA	02C2	—	—	—	—	—	—	—	—	RA[7:0]								xxxx
LATA	02C4	—	—	—	—	—	—	—	—	LATA[7:6]		—	LATA[4:0]					xxxx
ODCA	02C6	—	—	—	—	—	—	—	—	ODA[7:6]		—	ODA[4:0]					0000

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 $\overline{\text{MCLR}}$ functionality is enabled (MCLRE Configuration bit = 1).

TABLE 4-11: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13[2]	Bit 12[2]	Bit 11[1]	Bit 10[1]	Bit 9	Bit 8	Bit 7[2]	Bit 6[1]	Bit 5[1]	Bit 4	Bit 3[1]	Bit 2[2]	Bit 1[2]	Bit 0	All Resets
TRISB	02C8	TRISB[15:0]																FFFF
PORTB	02CA	RB[15:0]																xxxx
LATB	02CC	LATB[15:0]																xxxx
ODCB	02CE	ODB[15:0]																0000

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	Bit 1	Bit 0	All Resets
PADCFG1	02FC	—	—	—	—	SDO2DIS[1]	SCK2DIS[1]	SDO1DIS	SCK1DIS	—	—	—	—	—	—	—	—	0000

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	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	A/D Buffer 0																xxxx
ADC1BUF1	0302	A/D Buffer 1																xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM[1:0]		SSRC[2:0]			—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG[2:0]			OFFCAL	—	CSCNA	—	—	r	—	SMPI[3:0]				r	ALTS	0000
AD1CON3	0324	ADRC	EXTSAM	PUMPEN	SAMC[4:0]					—	—	ADCS[5:0]						0000
AD1CHS	0328	CH0NB	—	—	—	CH0SB[3:0]				CH0NA	—	—	—	CH0SA[3:0]				0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12[1]	CSSL11[1]	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	—	CSSL4[1]	CSSL3[1]	CSSL2[1]	CSSL1	CSSL0	0000

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	Bit 1	Bit 0	All Resets
ANCFG	04DE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VBGEN	0000
ANSA	04E0	—	—	—	—	—	—	—	—	—	—	—	—	ANSA[3:0]				000F
ANSB	04E2	ANSB15	ANSB14	ANSB13	ANSB12[1]	—	—	—	—	—	—	—	ANSB4	ANSB3[2]	ANSB2[1]	ANSB1[1]	ANSB0[1]	F01F[3]

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	Bit 1	Bit 0	All Resets
CMSTAT	0630	CMIDL	—	—	—	—	—	C2EVT[1]	C1EVT	—	—	—	—	—	—	C2OUT	C1OUT	xxxx
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRSS	CVR[4:0]					0000
CM1CON	0634	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL[1:0]		—	CREF	—	—	CCH[1:0]		xxxx
CM2CON[1]	0636	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL[1:0]		—	CREF	—	—	CCH[1:0]		0000

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	Bit 1	Bit 0	All Resets
RCON	0740	TRAPR	IOPUWR	SBOREN	—	—	—	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	**(Note 1)**
OSCCON	0742	—	COSC[2:0]			—	NOSC[2:0]			CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN	**(Note 2)**
CLKDIV	0744	ROI	DOZE[2:0]			DOZEN	RCDIV[2:0]			—	—	—	—	—	—	—	—	`3100`
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN[5:0]						`0000`
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV[3:0]				—	—	—	—	—	—	—	—	`0000`
HLVDCON	0756	HLVDEN	—	HLSIDL	—	—	—	—	—	VDIR	BGVST	IRVST	—	HLVDL[3:0]				`0000`

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	Bit 1	Bit 0	All Resets
NVMCON	0760	WR	WREN	WRERR	PGMONLY	—	—	—	—	—	ERASE	NVMOP[5:0]						`0000`
NVMKEY	0766	—	—	—	—	—	—	—	—	NVM Key Register								`0000`

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	Bit 1	Bit 0	All Resets
ULPWCON	0768	ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK	—	—	—	—	—	—	—	—	`0000`

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	Bit 1	Bit 0	All Resets
PMD1	0770	—	T4MD	T3MD	T2MD	T1MD	—	—	—	SSP1MD	U2MD	U1MD	—	—	—	—	ADC1MD	`0000`
PMD2	0772	—	—	—	—	—	—	—	—	—	—	—	—	—	CCP3MD	CCP2MD	CCP1MD	`0000`
PMD3	0774	—	—	—	—	—	CMPMD	—	—	—	—	—	—	—	—	SSP2MD	—	`0000`
PMD4	0776	—	—	—	—	—	—	—	—	ULPWUMD	—	—	EEMD	REFOMD	—	HLVDMD	—	`0000`

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

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 predecrements 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.

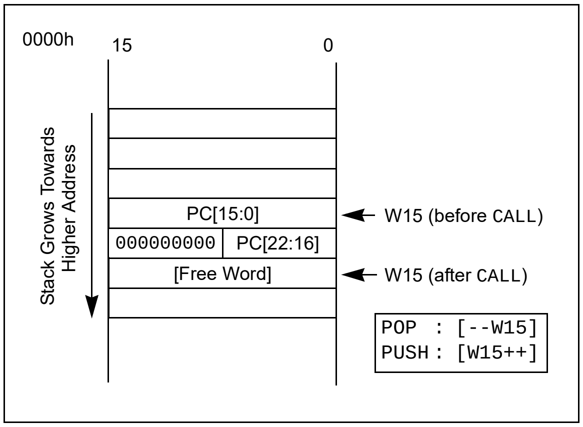
The Stack Pointer Limit Value (SPLIM) register, associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM[0] is forced to '`0`' as all stack operations must be word-aligned. Whenever an EA is generated, 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 these data successfully, they 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.

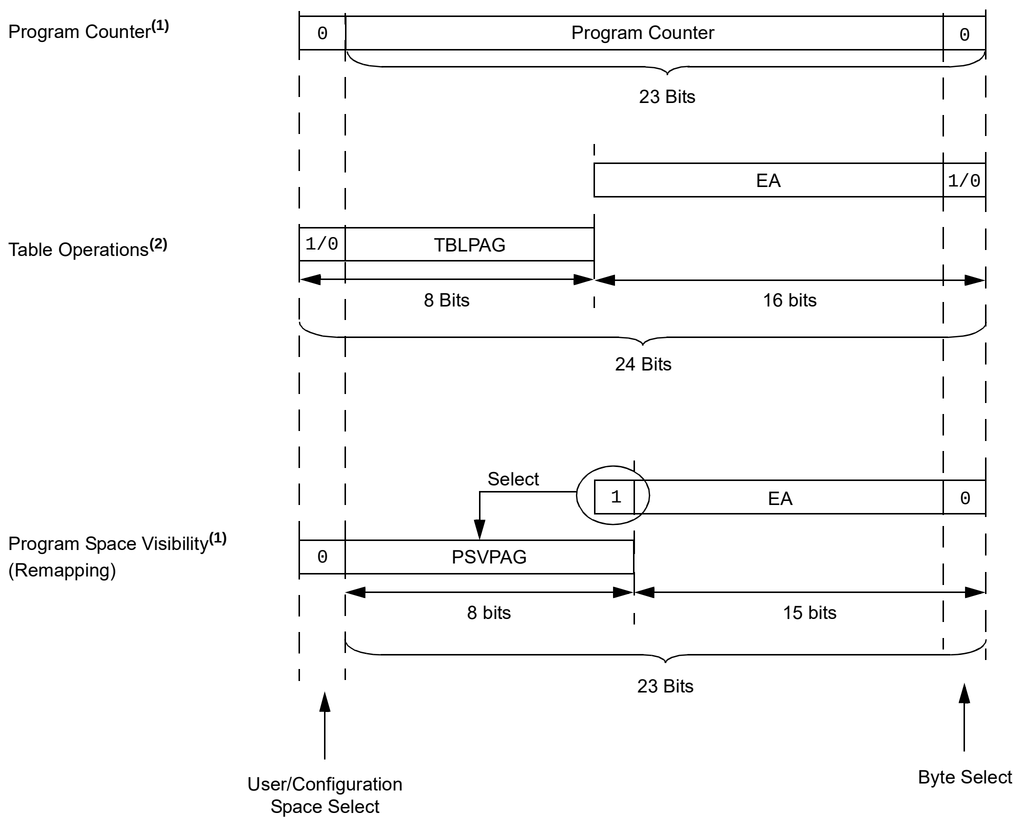
TABLE 4-20: PROGRAM SPACE ADDRESS CONSTRUCTION

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][2]		Data EA[14:0][1]	
		`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].

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



Note 1: The LSb of program space addresses is always fixed as ‘`0`’ in order to maintain word alignment of data in the program and Data Spaces.

2: Table operations are not required to be word-aligned. Table read operations are permitted in the configuration memory space.

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 eight 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`’.

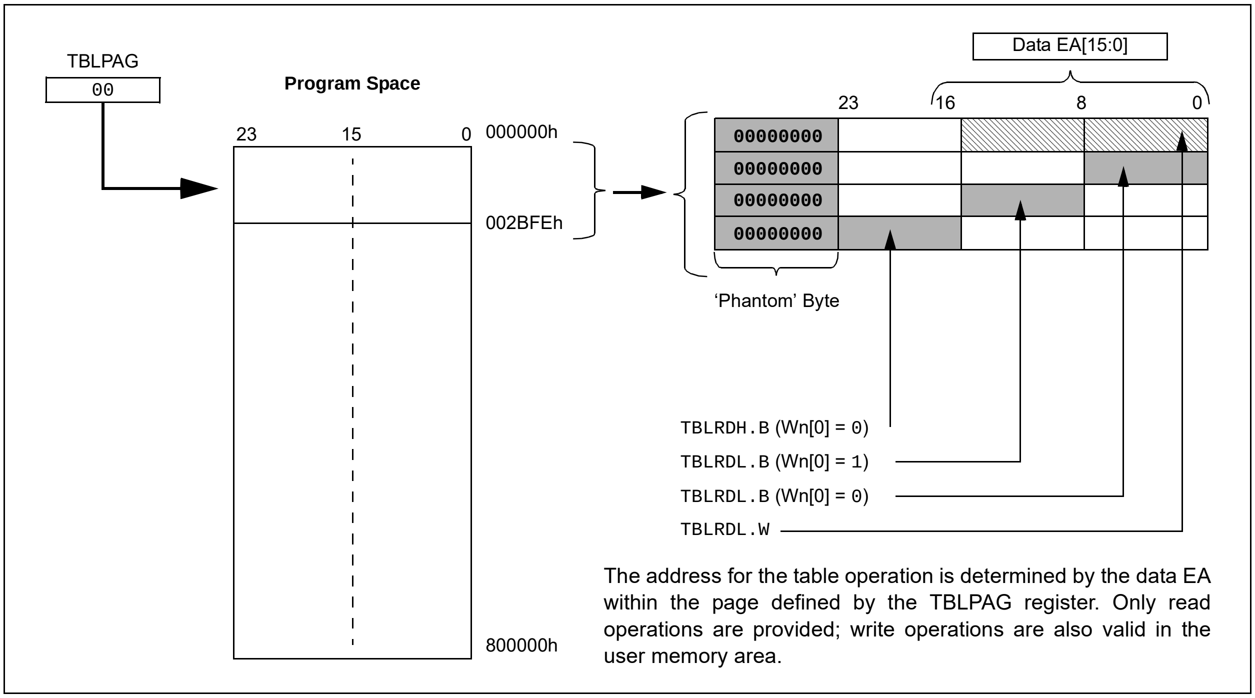
   In Byte mode, it maps the upper or lower byte of the program word to D[7:0] of the data address, as above. Note that the data will always be ‘`0`’ when the upper ‘phantom’ byte is selected (byte select = `1`).

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



The address for the table operation is determined by the data EA within the page defined by the TBLPAG register. Only read operations are provided; write operations are also valid in the user memory area.

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 two 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.

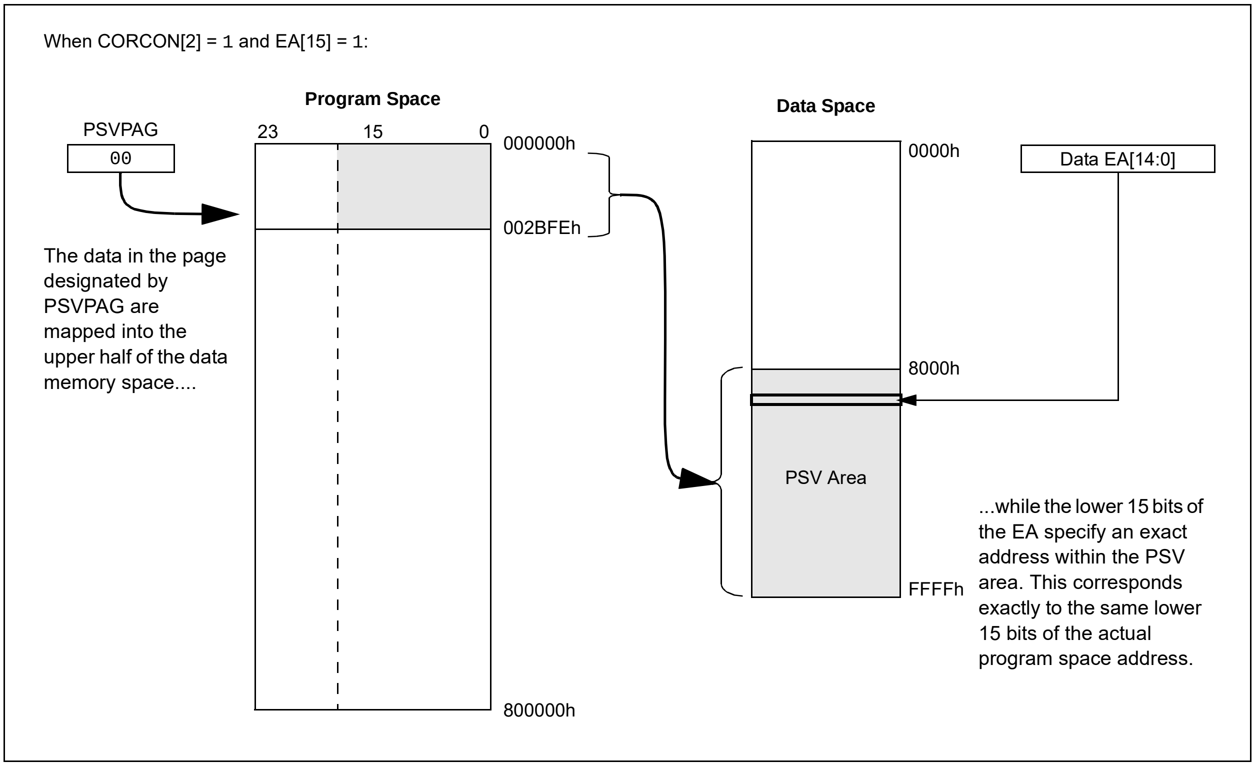
For operations that use PSV and are executed outside of a `REPEAT` loop, the `MOV` and `MOV.D` instructions will require one instruction cycle, in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

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



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 **"PIC24F Flash Program Memory"** (www.microchip.com/DS30009715) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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 ($\overline{\text{MCLR}}$/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.

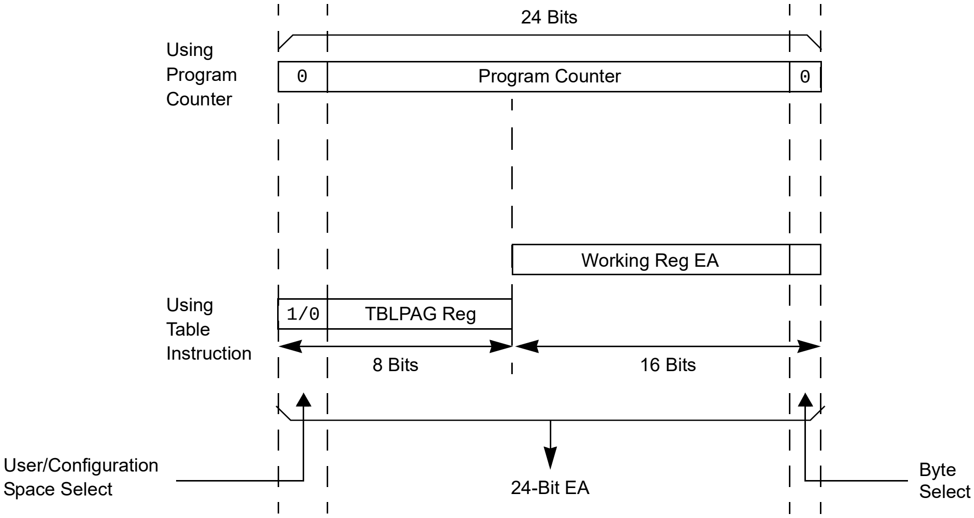
5.1 Table Instructions and Flash Programming

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



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 one row, two rows and four 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 are written to program memory using `TBLWT` instructions, the data are not written directly to memory. Instead, data written using Table Writes are 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.

Note:	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.

REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

R/SO/HC-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY[4]	—	—	—	—
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[1]	NVMOP4[1]	NVMOP3[1]	NVMOP2[1]	NVMOP1[1]	NVMOP0[1]
bit 7							bit 0

Legend:	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[4]

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[1]

Erase Operations (when ERASE bit is ‘`1`’):
`1010xx` = Erases entire boot block (including code-protected boot block)[2]
`1001xx` = Erases entire memory (including boot block, configuration block, general block)[2]
`011010` = Erases four rows of Flash memory[3]
`011001` = Erases two rows of Flash memory[3]
`011000` = Erases one row of Flash memory[3]
`0101xx` = Erases entire configuration block (except code protection bits)
`0100xx` = Erases entire data EEPROM[4]
`0011xx` = Erases entire general memory block programming operations
`0001xx` = Writes one row of Flash memory (when ERASE bit is ‘`0`’)[3]

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

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 `NOP`s, 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
```

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
```

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 **"Data EEPROM"** (www.microchip.com/DS39720) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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, 7FFFFFh. For PIC24FXXKL4XX devices, the size of the data EEPROM is 256 words (7FFE00h to 7FFFFFh). For PIC24FXXKL3XX devices, the size of the data EEPROM is 128 words (7FFF00h to 7FFFFFh). 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

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.

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");
```

REGISTER 6-1: NVMCON: NONVOLATILE MEMORY CONTROL REGISTER

HC/R/SO-0	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(1)	NVMOP4(1)	NVMOP3(1)	NVMOP2(1)	NVMOP1(1)	NVMOP0(1)
bit 7							bit 0

Legend:	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(1)

Erase Operations (when ERASE bit is ‘1’):
011010 = Erases eight words
011001 = Erases four words
011000 = Erases one word
0100xx = Erases entire data EEPROM

Programming Operations (when ERASE bit is ‘0’):
0001xx = Writes one word

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

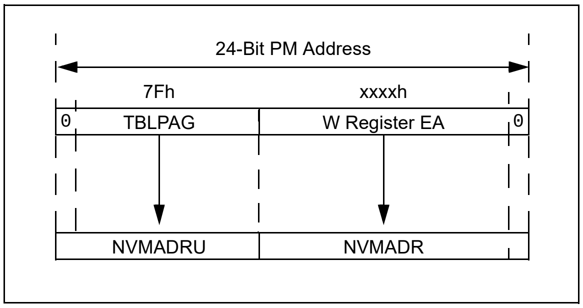
6.3 NVM Address Register

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 are 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.

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);           // Initizlize 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
```

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] = `0001xx`).
   - 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);             // Initizlize 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
```

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_tblrdl`) 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);               // Initizlize lower word of address
    data =   __builtin_tblrdl(offset);                   // Write EEPROM data to write latch
```

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 **“Reset with Programmable Brown-out Reset”** (www.microchip.com/DS39728) in the *“dsPIC33/PIC24 Family Reference Manual”*.

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
- $\overline{\text{MCLR}}$: 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.

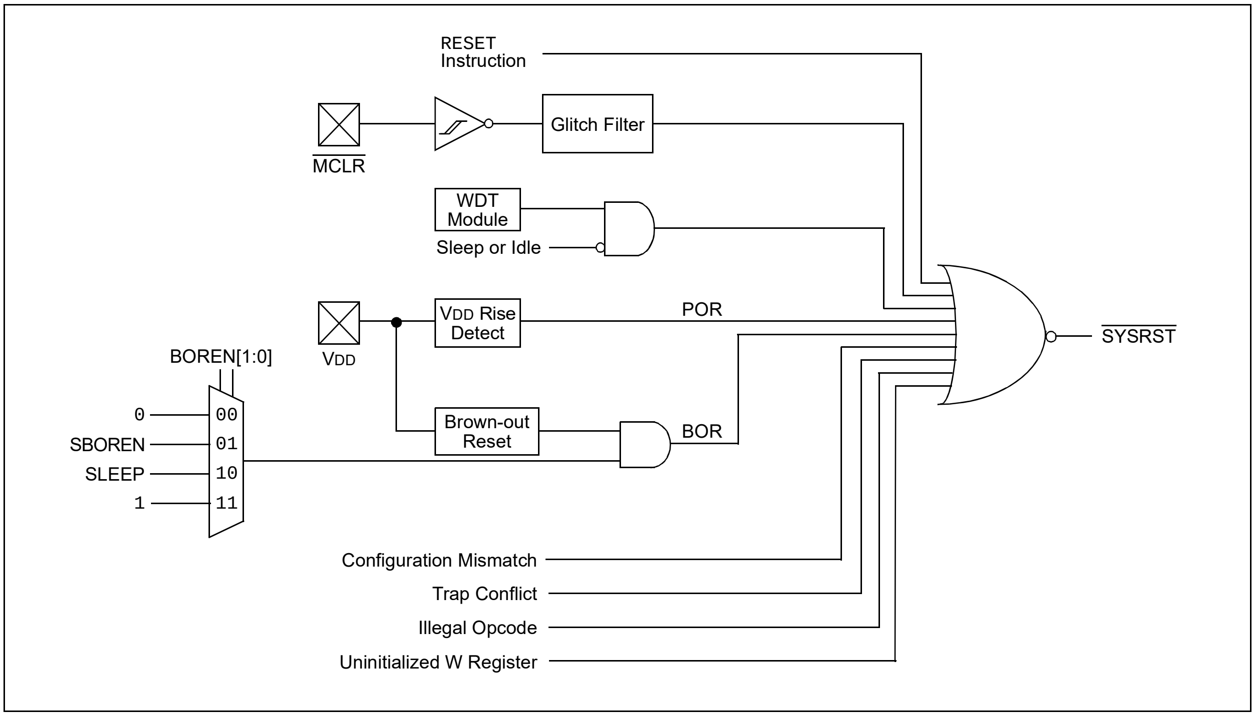
Note: Refer to the specific peripheral or CPU section of this manual for register Reset states.

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 (RCON[1:0]) 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



REGISTER 7-1: RCON: RESET CONTROL REGISTER(1)

R/W-0	R/W-0	R/W-0(3)	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(2)	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(3)
`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(2)
`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.

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`’.

REGISTER 7-1: RCON: RESET CONTROL REGISTER[1] (CONTINUED)

bit 3 **SLEEP:** Wake-up from Sleep Flag bit
`1` = Device has been in Sleep mode
`0` = Device has not been in Sleep mode

bit 2 **IDLE:** Wake-up from Idle Flag bit
`1` = Device has been in Idle mode
`0` = Device has not been in Idle mode

bit 1 **BOR:** 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 **POR:** 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])	$\overline{\text{MCLR}}$ Reset	POR
SWR (RCON[6])	`RESET` Instruction	POR
WDTO (RCON[4])	WDT Time-out	`PWRSAV` Instruction, POR
SLEEP (RCON[3])	`PWRSAV #SLEEP` Instruction	POR
IDLE (RCON[2])	`PWRSAV #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	FNOSCx Configuration bits (FNOSC[10:8])
BOR	
$\overline{\text{MCLR}}$	COSCx Control bits (OSCCON[14:12])
WDTO	
SWR	

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, 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 SYSRST delay times.

The FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the SYSRST signal is released.

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

Reset Type	Clock Source	SYSRST Delay	System Clock Delay	Notes
POR(6)	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"**.

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 FNOSCx bits in the Flash Configuration Word (FOSCSEL); see Table 7-2. The RCFGCAL and NVMCON registers are only affected by a POR.

7.4 Brown-out Reset (BOR)

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

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

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

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

7.4.1 LOW-POWER BOR (LPBOR)

The Low-Power BOR is an alternate setting for the BOR, designed to consume minimal power. In LPBOR mode, BORV[1:0] (FPOR[6:5]) = `00`. The BOR trip point is approximately 2.0V. Due to the low current consumption, the accuracy of the LPBOR mode can vary. Unlike the other BOR modes, LPBOR mode will not cause a device Reset when VDD drops below the trip point. Instead, it re-arms the POR circuit to ensure that the device will reset properly in the event that V_{DD} continues to drop below the minimum operating voltage. The device will continue to execute code when V_{DD} is below the level of the LPBOR trip point. A device that requires falling edge BOR protection to prevent code from improperly executing should use one of the other BOR voltage settings.

7.4.2 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.

7.4.3 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.

7.4.4 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 **"Interrupts"** (www.microchip.com/DS70000600) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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.

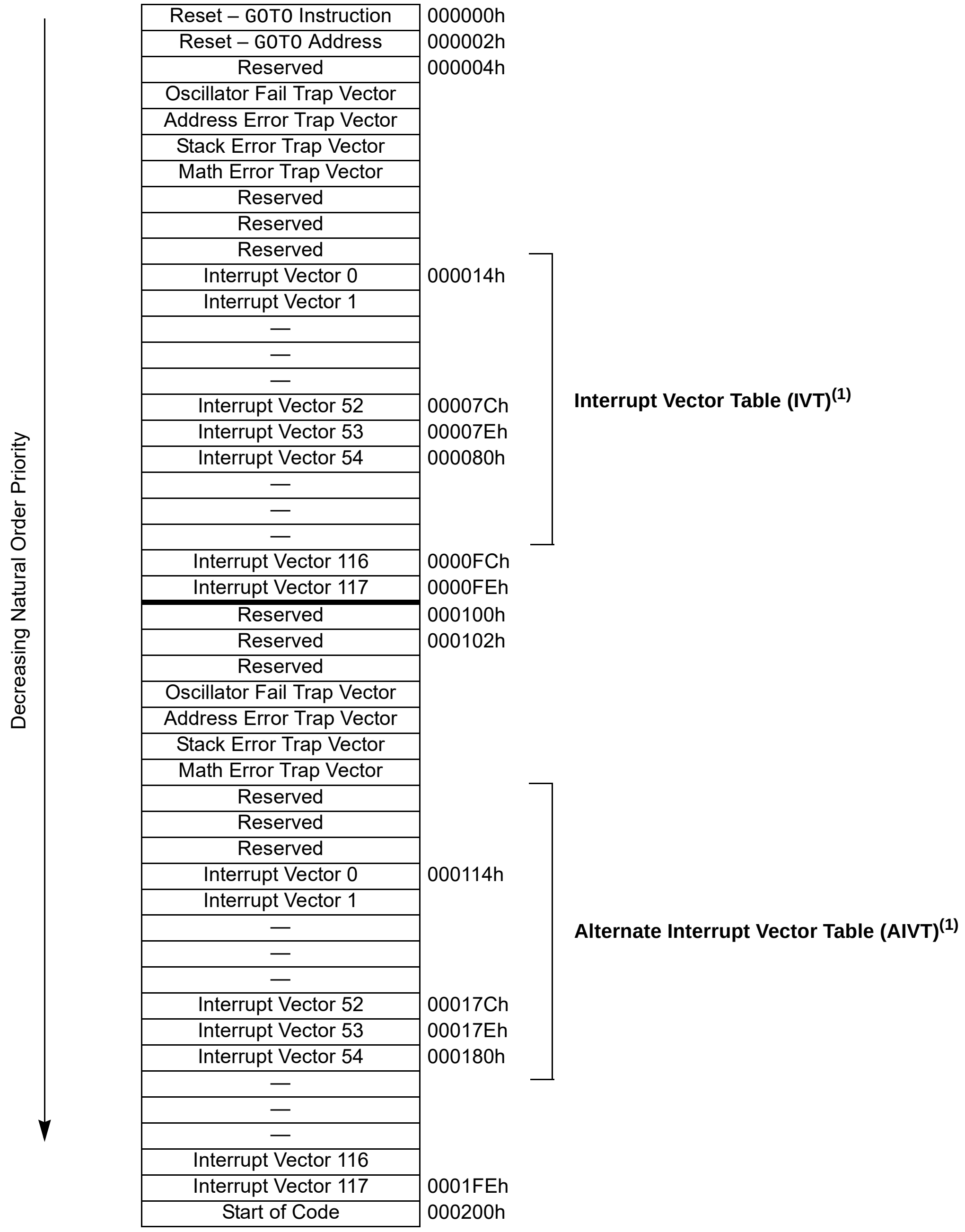
The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run 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.

FIGURE 8-1: PIC24F INTERRUPT VECTOR TABLE



Reset – `GOTO` Instruction	000000h
Reset – `GOTO` Address	000002h
Reserved	000004h
Oscillator Fail Trap Vector	
Address Error Trap Vector	
Stack Error Trap Vector	
Math Error Trap Vector	
Reserved	
Reserved	
Reserved	
Interrupt Vector 0	000014h
Interrupt Vector 1	
—	
—	
—	
Interrupt Vector 52	00007Ch
Interrupt Vector 53	00007Eh
Interrupt Vector 54	000080h
—	
—	
—	
Interrupt Vector 116	0000FCh
Interrupt Vector 117	0000FEh
Reserved	000100h
Reserved	000102h
Reserved	
Oscillator Fail Trap Vector	
Address Error Trap Vector	
Stack Error Trap Vector	
Math Error Trap Vector	
Reserved	
Reserved	
Reserved	
Interrupt Vector 0	000114h
Interrupt Vector 1	
—	
—	
—	
Interrupt Vector 52	00017Ch
Interrupt Vector 53	00017Eh
Interrupt Vector 54	000180h
—	
—	
—	
Interrupt Vector 116	
Interrupt Vector 117	0001FEh
Start of Code	000200h

Note 1: See Table 8-2 for the interrupt vector list.

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 Description	MPLAB® XC16 ISR Name	Vector #	IRQ #	IVT Address	AIVT Address	Interrupt Bit Locations		
						Flag	Enable	Priority
External Interrupt 0	_INT0Interrupt	8	0	000014h	000114h	IFS0[0]	IEC0[0]	IPC0[2:0]
CCP1/ECCP1	_CCP1Interrupt	10	2	000018h	000118h	IFS0[2]	IEC0[2]	IPC0[10:8]
Timer1	_T1Interrupt	11	3	00001Ah	00011Ah	IFS0[3]	IEC0[3]	IPC0[14:12]
CCP2	_CCP2Interrupt	14	6	000020h	000120h	IFS0[6]	IEC0[6]	IPC1[10:8]
Timer2	_T2Interrupt	15	7	000022h	000122h	IFS0[7]	IEC0[7]	IPC1[14:12]
Timer3	_T3Interrupt	16	8	000024h	000124h	IFS0[8]	IEC0[8]	IPC2[2:0]
UART1 Receiver	_U1RXInterrupt	19	11	00002Ah	00012Ah	IFS0[11]	IEC0[11]	IPC2[14:12]
UART1 Transmitter	_U1TXInterrupt	20	12	00002Ch	00012Ch	IFS0[12]	IEC0[12]	IPC3[2:0]
ADC1 Conversion Done	_ADC1Interrupt	21	13	00002Eh	00012Eh	IFS0[13]	IEC0[13]	IPC3[6:4]
NVM (NVM Write Complete)	_NVMWriteInterrupt	23	15	000032h	000132h	IFS0[15]	IEC0[15]	IPC3[14:12]
MSSP1 SPI or I^{2}C Event	_MSSP1Interrupt	24	16	000034h	000134h	IFS1[0]	IEC1[0]	IPC4[2:0]
MSSP1 Bus Collision Event	_MSSP1BCInterrupt	25	17	000036h	000136h	IFS1[1]	IEC1[1]	IPC4[6:4]
Comparator Event	_CompInterrupt	26	18	000038h	000138h	IFS1[2]	IEC1[2]	IPC4[10:8]
Input Change Notification	_CNInterrupt	27	19	00003Ah	00013Ah	IFS1[3]	IEC1[3]	IPC4[14:12]
External Interrupt 1	_INT1Interrupt	28	20	00003Ch	00013Ch	IFS1[4]	IEC1[4]	IPC5[2:0]
CCP3	_CCP3Interrupt	33	25	000046h	000146h	IFS1[9]	IEC1[9]	IPC6[6:4]
Timer4	_T4Interrupt	35	27	00004Ah	00014Ah	IFS1[11]	IEC1[11]	IPC6[14:12]
External Interrupt 2	_INT2Interrupt	37	29	00004Eh	00014Eh	IFS1[13]	IEC1[13]	IPC7[6:4]
UART2 Receiver	_U2RXInterrupt	38	30	000050h	000150h	IFS1[14]	IEC1[14]	IPC7[10:8]
UART2 Transmitter	_U2TXInterrupt	39	31	000052h	000152h	IFS1[15]	IEC1[15]	IPC7[14:12]
Timer3 Gate External Count	_T3GIInterrupt	45	37	00005Eh	00015Eh	IFS2[5]	IEC2[5]	IPC9[6:4]
MSSP2 SPI or I^{2}C Event	_MSSP2Interrupt	57	49	000076h	000176h	IFS3[1]	IEC3[1]	IPC12[6:4]
MSSP2 Bus Collision Event	_MSSP2BCInterrupt	58	50	000078h	000178h	IFS3[2]	IEC3[2]	IPC12[10:8]
UART1 Error	_U1ErrInterrupt	73	65	000096h	000196h	IFS4[1]	IEC4[1]	IPC16[6:4]
UART2 Error	_U2ErrInterrupt	74	66	000098h	000198h	IFS4[2]	IEC4[2]	IPC16[10:8]
HLVD (High/Low-Voltage Detect)	_HLVDInterrupt	80	72	0000A4h	0001A4h	IFS4[8]	IEC4[8]	IPC18[2:0]
ULPW (Ultra Low-Power Wake-up)	_ULPWUInterrupt	88	80	0000B4h	0001B4h	IFS5[0]	IEC5[0]	IPC20[2:0]

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 INT0 (External Interrupt 0) is depicted as having a vector number and a natural order priority of 0. The INT0IF status bit is found in IFS0[0], the INT0IE enable bit in IEC0[0] and the INT0IP[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.

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(1)
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(2,3)	IPL1(2,3)	IPL0(2,3)	RA(1)	N(1)	OV(1)	Z(1)	C(1)
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(2,3) `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”**.

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[(2)]	PSV[(1)]	—	—
bit 7							bit 0

Legend:	C = 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 15-4	**Unimplemented:** Read as ‘`0`’
bit 3	**IPL3:** CPU Interrupt Priority Level Status bit(2) `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.
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”**.

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`’

REGISTER 8-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	HSC/R-0	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

Legend:	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

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

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 **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

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[1]	U2RXIF[1]	INT2IF	—	T4IF[1]	—	CCP3IF[1]	—
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[1]
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 14 **U2RXIF:** UART2 Receiver Interrupt Flag Status bit[1]
`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[1]
`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[1]
`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^{2}C Bus Collision Interrupt Flag Status bit
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 0 **SSP1IF:** MSSP1 SPI/I^{2}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.

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(1)	SSP2IF(1)	—
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^{2}C Bus Collision Interrupt Flag Status bit(1)
`1` = Interrupt request has occurred
`0` = Interrupt request has not occurred

bit 1 **SSP2IF:** MSSP2 SPI/I^{2}C Event Interrupt Flag Status bit(1)
`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.

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

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

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

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

bit 15-9 **Unimplemented:** Read as ‘`0`’

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

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

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

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

bit 0 **Unimplemented:** Read as ‘`0`’

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

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

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

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

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

bit 15-1 **Unimplemented:** Read as ‘`0`’

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

REGISTER 8-11: 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

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[1]	U2RXIE[1]	INT2IE	—	T4IE[1]	—	CCP3IE[1]	—
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[1]
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit[1]
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[1]
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[1]
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^2C Bus Collision Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SSP1IE:** MSSP1 SPI/I^2C 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.

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(1)	SSP2IE(1)	—
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^2C Bus Collision Interrupt Enable bit(1)

`1` = Interrupt request is enabled
`0` = Interrupt request is not enabled

bit 1 **SSP2IF:** MSSP2 SPI/I^2C Event Interrupt Enable bit(1)

`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.

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

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

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

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

bit 15-9 **Unimplemented:** Read as ‘0’

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

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

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

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

bit 0 **Unimplemented:** Read as ‘0’

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

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

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

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

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

bit 15-1 **Unimplemented:** Read as ‘0’

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

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

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

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

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

bit 15 **Unimplemented:** Read as ‘`0`’

bit 14-12 **T1IP[2:0]:** Timer1 Interrupt Priority bits

`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘`0`’

bit 10-8 **CCP1IP[2:0]:** Capture/Compare/PWM1 Interrupt Priority bits

`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

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

bit 2-0 **INT0IP[2:0]:** External Interrupt 0 Priority bits

`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

REGISTER 8-18: 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`’

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

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 **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

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

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

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 **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^2C 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^2C Event Interrupt Priority bits

111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

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[1]	T4IP1[1]	T4IP0[1]	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CCP3IP2[1]	CCP3IP1[1]	CCP3IP0[1]	—	—	—	—
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[1]

`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[1]

`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.

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[1]	U2TXIP1[1]	U2TXIP0[1]	—	U2RXIP2[1]	U2RXIP1[1]	U2RXIP0[1]
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[1]

`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[1]

`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.

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`’

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(1)	BCL2IP1(1)	BCL2IP0(1)
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SSP2IP2(1)	SSP2IP1(1)	SSP2IP0(1)	—	—	—	—
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^2C Bus Collision Interrupt Priority bits(1) 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^2C Event Interrupt Priority bits(1) 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.

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(1)	U2ERIP1(1)	U2ERIP0(1)
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2(1)	U1ERIP1(1)	U1ERIP0(1)	—	—	—	—
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(1)

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(1)

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.

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

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
—	VECNUM[6:0]						
bit 7							bit 0

Legend:	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 VECNUMx 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

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.

9.0 OSCILLATOR CONFIGURATION

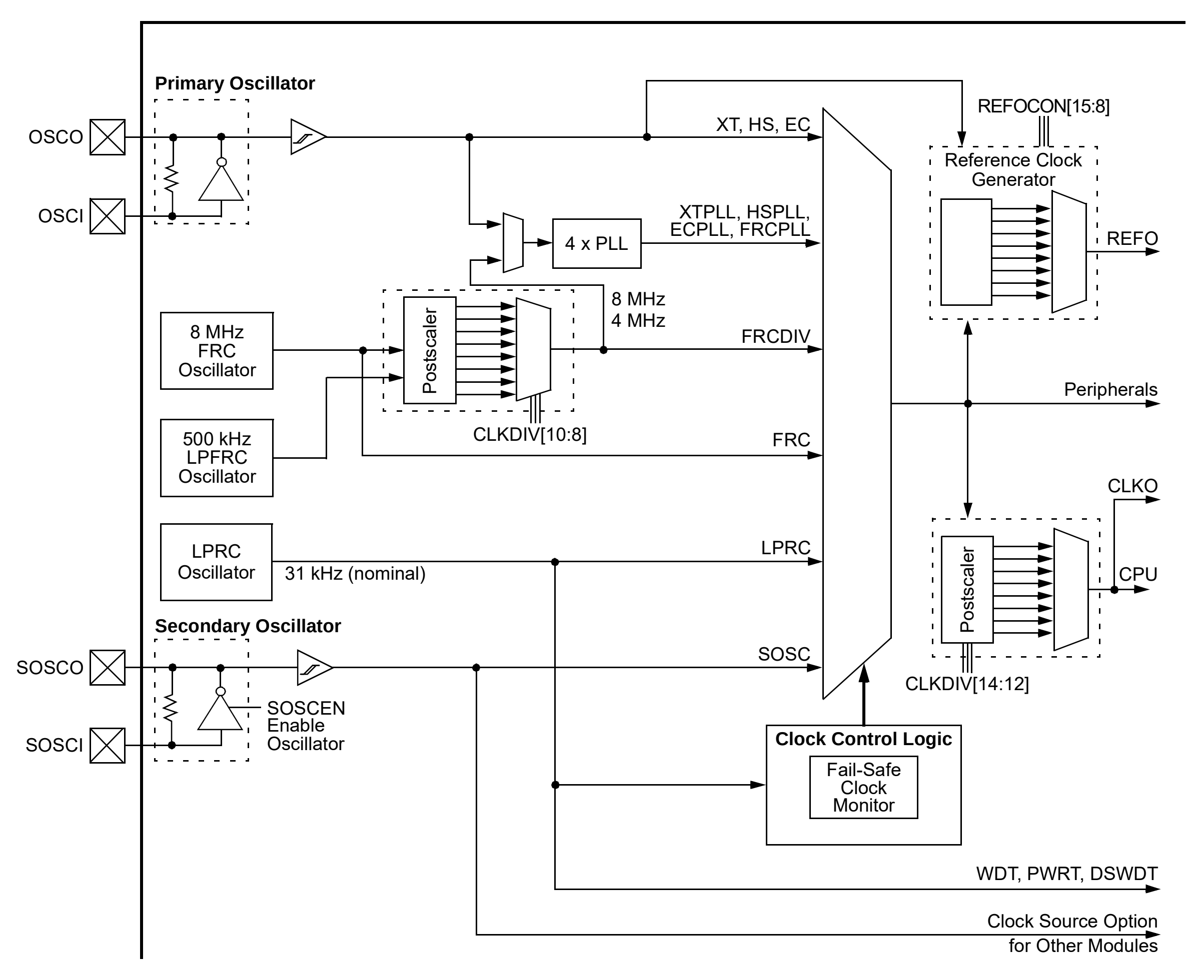
> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Oscillator Configuration, refer to **"Oscillator"** (www.microchip.com/DS39700) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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



9.1 CPU Clocking Scheme

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

- Primary Oscillator (POSC) on the OSCI and OSCO 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 OSCO 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	1, 2
500 kHz FRC Oscillator with Postscaler (LPFRCDIV)	Internal	11	110	1
Low-Power RC Oscillator (LPRC)	Internal	11	101	1
Secondary (Timer1) Oscillator (SOSC)	Secondary	00	100	1
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	1
8 MHz FRC Oscillator (FRC)	Internal	11	000	1

Note 1: OSCO pin function is determined by the OSCIOFNC Configuration bit.
2: This is the default oscillator mode for an unprogrammed (erased) device.

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	HSC/R-0	HSC/R-0	HSC/R-0	U-0	R/W-x[1]	R/W-x[1]	R/W-x[1]
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

HSC/R/SO-0	U-0	HSC/R-0[2]	U-0	HS/R/CO-0	R/W-0[3]	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN
bit 7							bit 0

Legend:	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 **NOSC[2:0]:** New Oscillator Selection bits[1]

`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 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.

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(2) `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(3) `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.

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[1]	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[1]

`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.

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
—	—	TUN[5:0][(1)]					
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(1)

`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 ten 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 SOSCEN 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.

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.

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[1]
`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.

NOTES:

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 **"Power-Saving Features with Deep Sleep"** (www.microchip.com/DS39727) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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 `PWRSAV` 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 `PWRSAV` 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: PWRSAV INSTRUCTION SYNTAX

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

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 `PWRSAV` 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 `PWRSAV` 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.

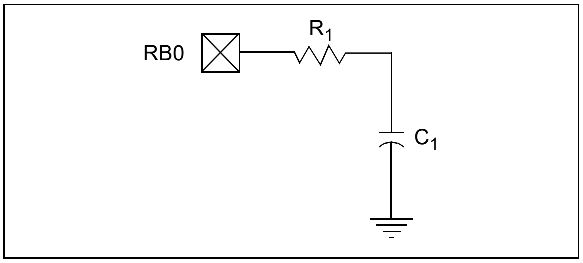
The time-out is dependent on the discharge time of the RC circuit on RB0. When the voltage on RB0 drops below V_{IL}, the device wakes up and executes the next instruction.

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
```

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.

NOTES:

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 **"I/O Ports with Peripheral Pin Select (PPS)"** (www.microchip.com/DS30009711) in the *"dsPIC33/PIC24 Family Reference Manual"*. 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.

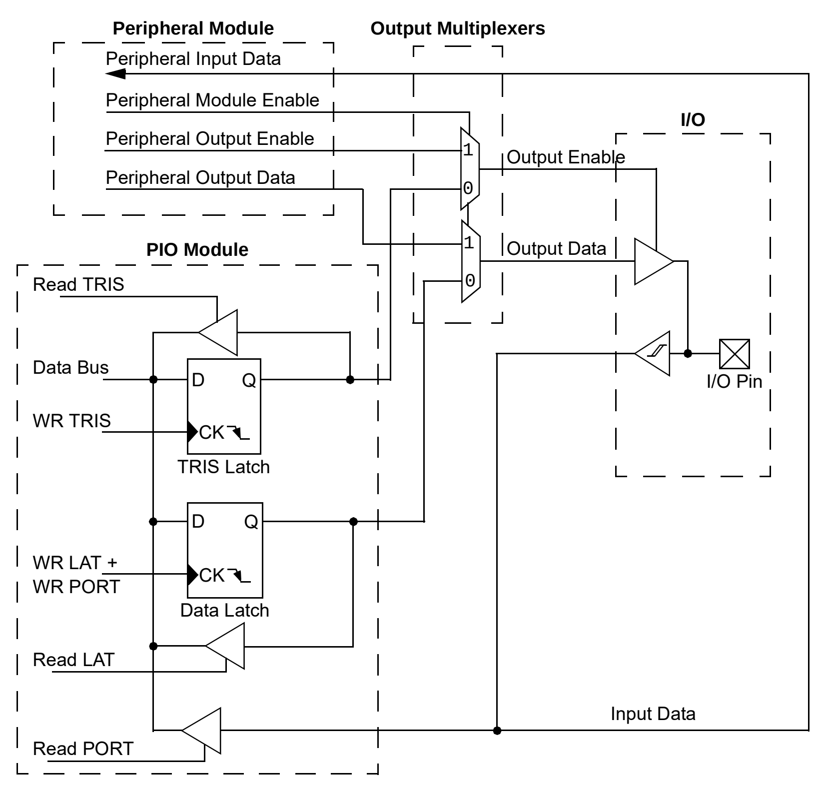
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively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



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 VIH 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 (VOH or VOL) 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.

On devices that do not have an A/D Converter, it is still necessary to configure the ANSx registers in order to enable digital input buffers. Any I/O pins with an ANx function listed in red in the device Pin Diagrams will default to have the digital input buffer disabled.

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
—	—	—	—	ANSA[3:0]			
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 15-4 **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
ANSB[15:12](1)				—	—	—	—
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
—	—	—	ANSB[4:0](1,2)				
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(1)
`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(1,2)
`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.
2: ANSB[3] is unimplemented on 14-pin and 20-pin devices.

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.
{
}
```

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 **"Timers"** (www.microchip.com/DS39704) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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:

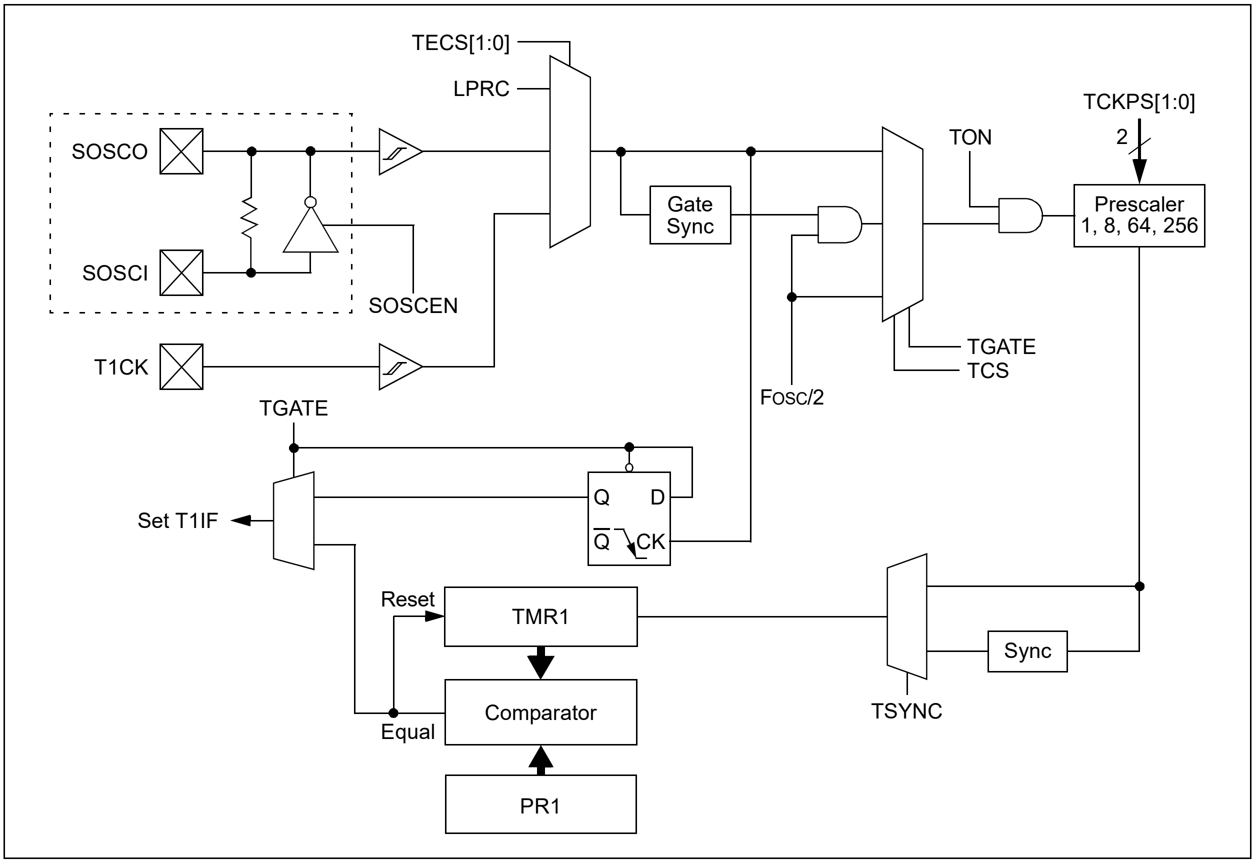
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

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



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[1]	T1ECS0[1]
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[1]
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.

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 **"Timers"** (www.microchip.com/DS39704) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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.

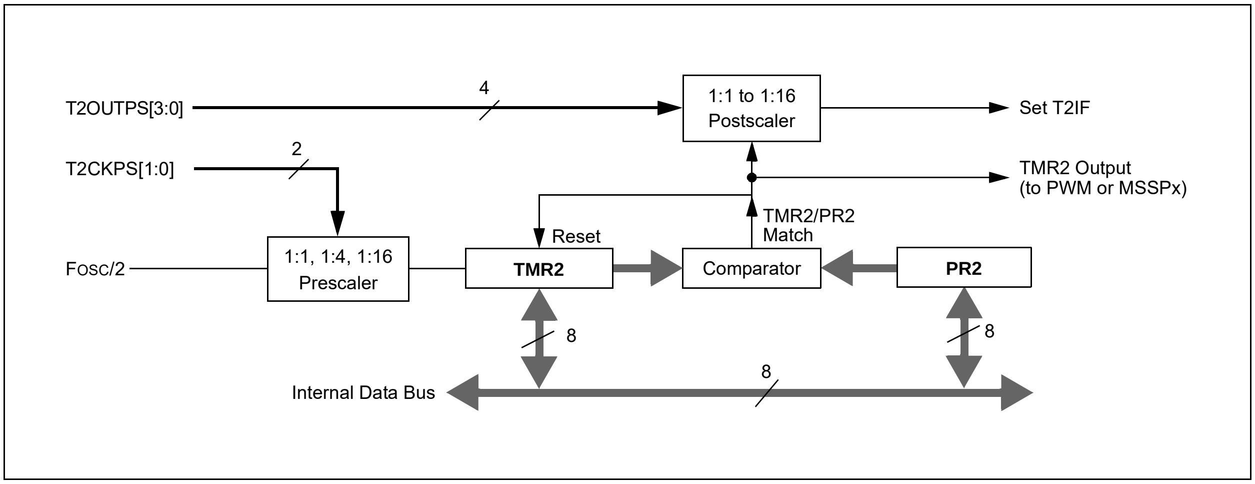
The prescaler and postscaler counters are cleared when any of the following occurs:

- 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



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

14.0 TIMER3 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 **"Timers"** (www.microchip.com/DS39704) in the *"dsPIC33/PIC24 Family Reference Manual"*.

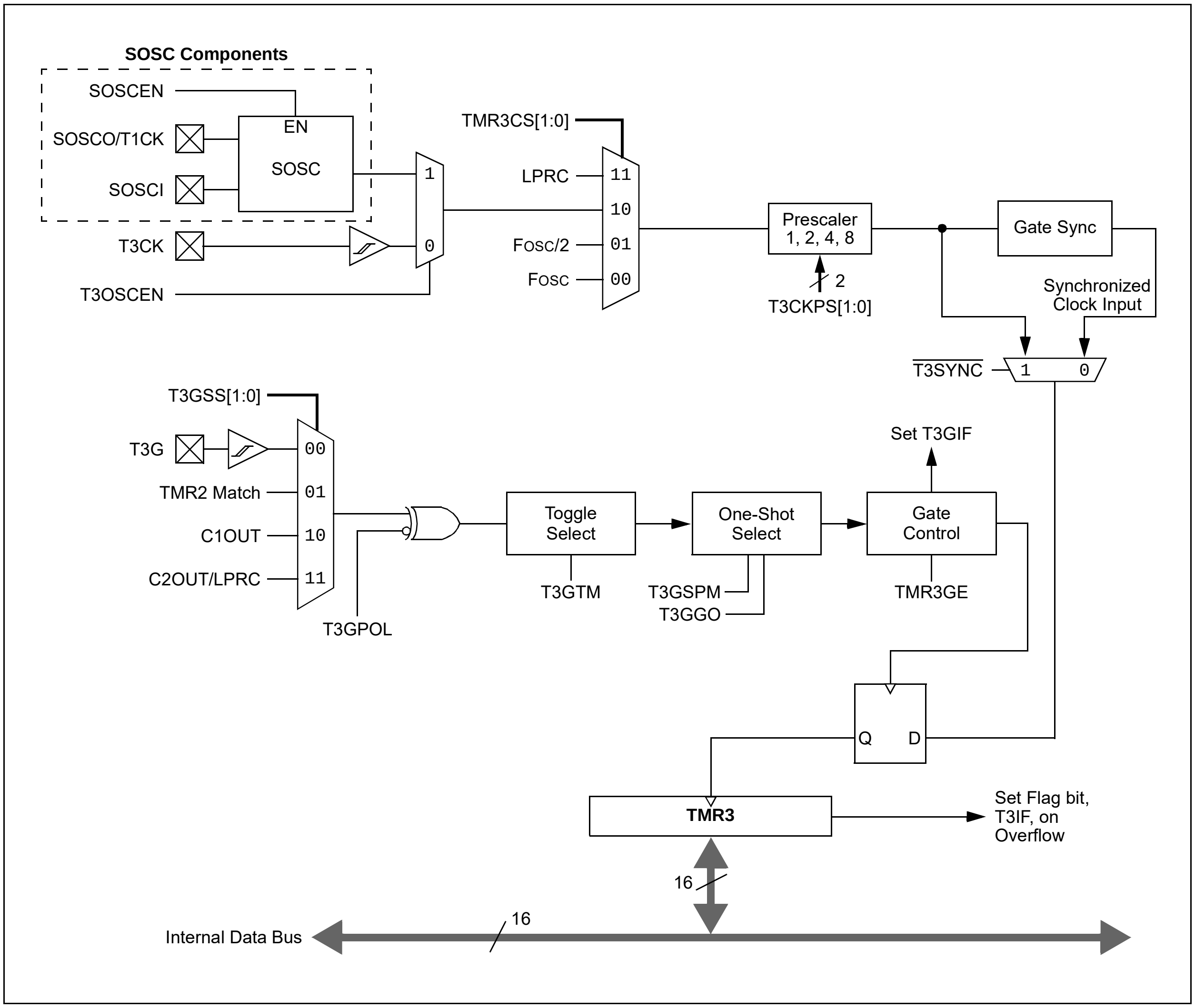
The Timer3 timer/counter modules incorporate these features:

- Software-Selectable Operation as a 16-Bit Timer or Counter
- One 16-Bit Readable and Writable Timer Value Register
- Selectable Clock Source (internal or external) with Device Clock, SOSC or LPRC Oscillator Options
- Interrupt-on-Overflow
- Multiple Timer Gating Options, including:
  - User-selectable gate sources and polarity
  - Gate/toggle operation
  - Single Pulse (One-Shot) mode
- Module Reset on ECCP Special Event Trigger

The Timer3 module is controlled through the T3CON register (Register 14-1). A simplified block diagram of the Timer3 module is shown in Figure 14-1.

The FOSC clock source should not be used with the ECCP capture/compare features. If the timer will be used with the capture or compare features, always select one of the other timer clocking options.

FIGURE 14-1: TIMER3 BLOCK DIAGRAM



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	$\overline{\text{T3SYNC}}$	—	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` = System clock (FOSC)(1)
`00` = Instruction clock (FOSC/2)

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 **$\overline{\text{T3SYNC}}$:** 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(2)

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.

REGISTER 14-2: T3GCON: TIMER3 GATE CONTROL REGISTER(1)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{T3DONE}}$	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/$\overline{\textbf{T3DONE}}$:** 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.

NOTES:

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 **"Timers"** (www.microchip.com/DS39704) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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

- 8-Bit Timer Register (TMR4)
- 8-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.

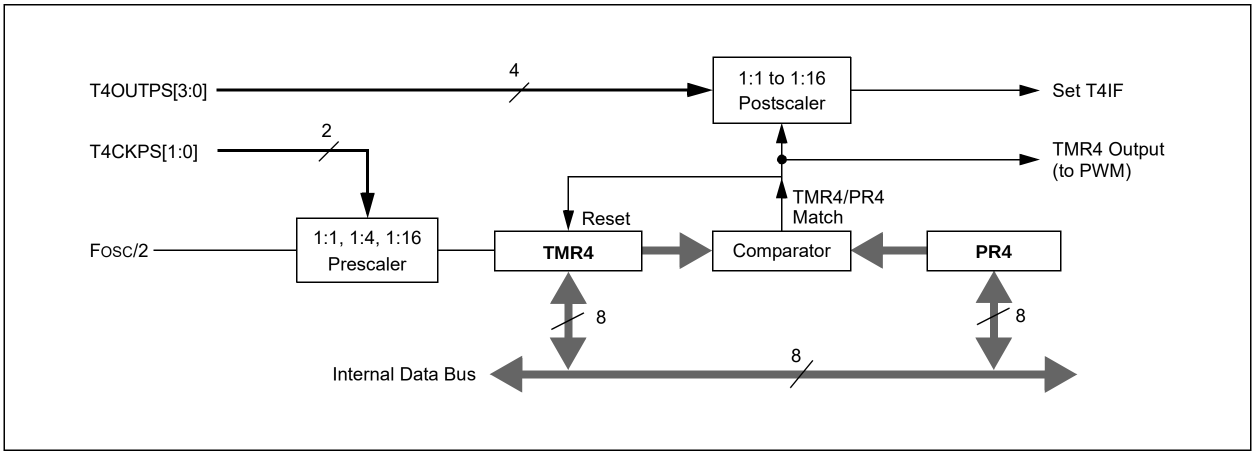
The prescaler and postscaler counters are cleared when any of the following occurs:

- 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



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

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 **"Capture/Compare/PWM Modules (CCP and ECCP)"** (www.microchip.com/DS30673) in 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 Ten 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.

FIGURE 16-1: GENERIC CAPTURE MODE BLOCK DIAGRAM

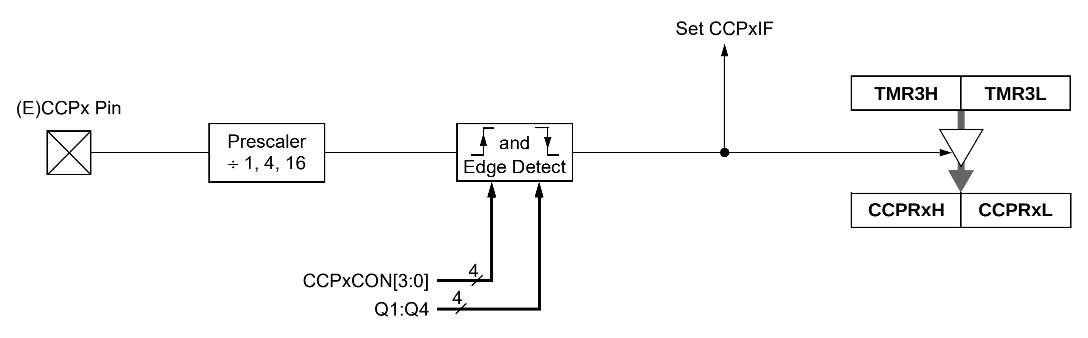


FIGURE 16-2: GENERIC COMPARE MODE BLOCK DIAGRAM

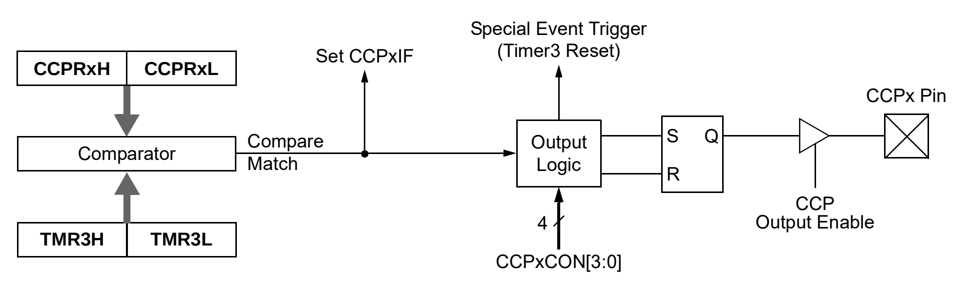
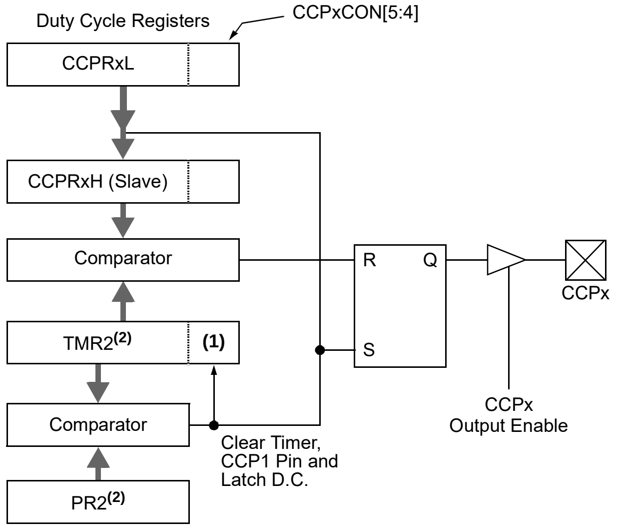


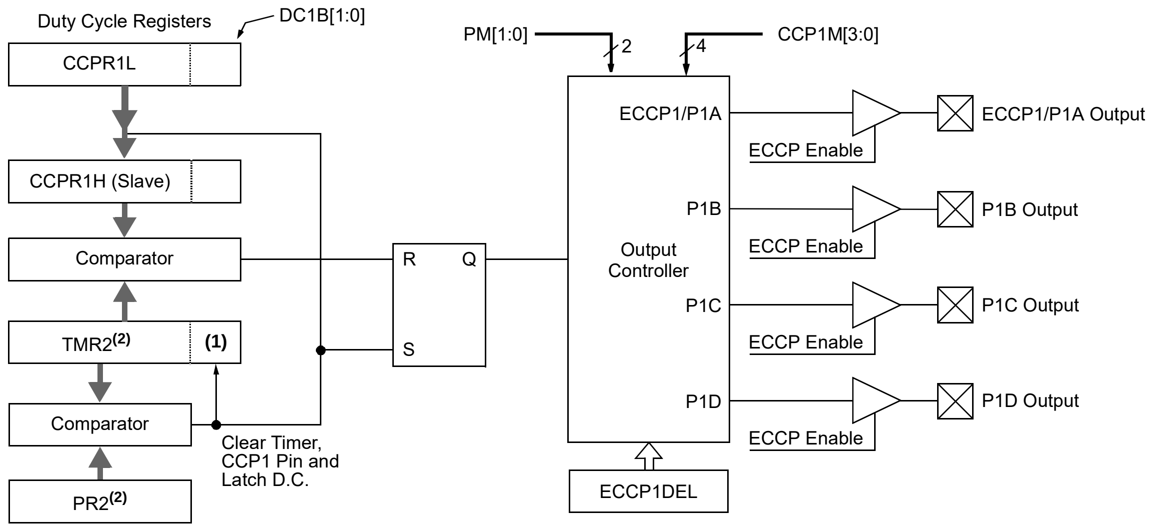
FIGURE 16-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

2: Either Timer2 or Timer4 may be used as the PWM time base.

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



Note 1: The 8-bit TMR2 value is concatenated with the 2-bit internal Q clock, or 2 bits of the prescaler, to create the 10-bit time base.

2: Either Timer2 or Timer4 may be used as the Enhanced PWM time base.

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(1)	CCPxM2(1)	CCPxM1(1)	CCPxM0(1)
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(1)

`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.

REGISTER 16-2: CCP1CON: ECCP1 CONTROL REGISTER (ECCP MODULES ONLY)(1)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PM1	PM0	DC1B1	DC1B0	CCP1M3(2)	CCP1M2(2)	CCP1M1(2)	CCP1M0(2)
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(2)

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.

REGISTER 16-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN CONTROL REGISTER[1]

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

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

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

bit 15-8 **Unimplemented:** Read as ‘`0`’

bit 7 **ECCPASE:** ECCP1 Auto-Shutdown Event Status bit

`1` = A shutdown event has occurred; ECCP outputs are in a shutdown state
`0` = ECCP outputs are operating

bit 6-4 **ECCPAS[2:0]:** ECCP1 Auto-Shutdown Source Select bits

`111` = V_{IL} on $\overline{\text{FLT0}}$ pin, or either C1OUT or C2OUT is high
`110` = V_{IL} on $\overline{\text{FLT0}}$ pin or C2OUT comparator output is high
`101` = V_{IL} on $\overline{\text{FLT0}}$ pin or C1OUT comparator output is high
`100` = V_{IL} on $\overline{\text{FLT0}}$ pin
`011` = Either C1OUT or C2OUT is high
`010` = C2OUT comparator output is high
`001` = C1OUT comparator output is high
`000` = Auto-shutdown is disabled

bit 3-2 **PSSAC[1:0]:** P1A and P1C Pins Shutdown State Control bits

`1x` = P1A and P1C pins tri-state
`01` = Drive pins, P1A and P1C, to ‘`1`’
`00` = Drive pins, P1A and P1C, to ‘`0`’

bit 1-0 **PSSBD[1:0]:** P1B and P1D Pins Shutdown State Control bits

`1x` = P1B and P1D pins tri-state
`01` = Drive pins, P1B and P1D, to ‘`1`’
`00` = Drive pins, P1B and P1D, to ‘`0`’

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

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.

2: Writing to the ECCPASE bit is disabled while an auto-shutdown condition persists.

3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 16-4: ECCP1DEL: ECCP1 ENHANCED PWM CONTROL REGISTER(1)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
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 FCY (FOSC/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.

REGISTER 16-5: PSTR1CON: ECCP1 PULSE STEERING CONTROL REGISTER[1]

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

R/W-0	R/W-0	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`.

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

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

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

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

bit 15-7	**Unimplemented:** Read as ‘`0`’
bit 6	**C3TSEL0:** CCP3 Timer Selection bit `1` = CCP3 uses TMR3/TMR4 `0` = CCP3 uses TMR3/TMR2
bit 5-4	**Unimplemented:** Read as ‘`0`’
bit 3	**C2TSEL0:** CCP2 Timer Selection bit `1` = CCP2 uses TMR3/TMR4 `0` = CCP2 uses TMR3/TMR2
bit 2-1	**Unimplemented:** Read as ‘`0`’
bit 0	**C1TSEL0:** CCP1/ECCP1 Timer Selection bit `1` = CCP1/ECCP1 uses TMR3/TMR4 `0` = CCP1/ECCP1 uses TMR3/TMR2

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

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 **"Master Synchronous Serial Port (MSSP)"** (www.microchip.com/DS30627) in 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^2C)
  - 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^2C interface supports the following modes in hardware:

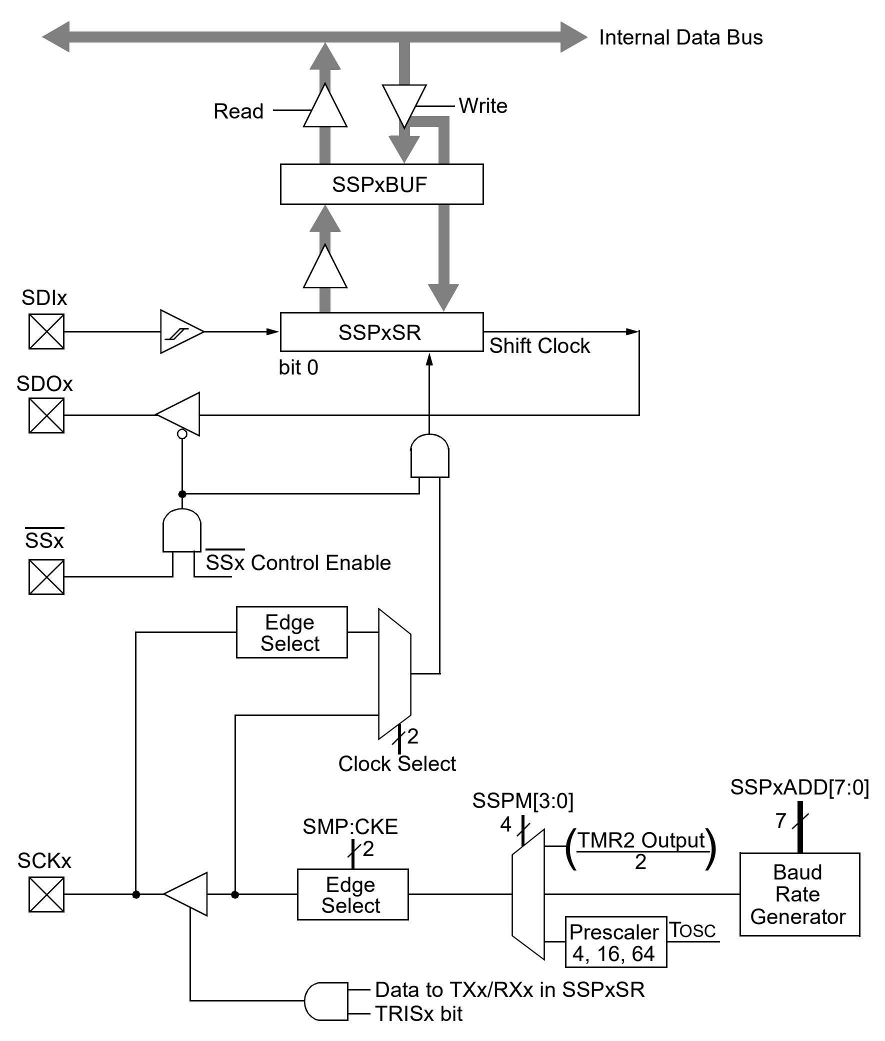
- Master mode
- Multi-Master mode
- Slave mode with 10-Bit And 7-Bit Addressing and Address Masking
- Byte NACKing
- Selectable Address and Data Hold and Interrupt Masking

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.

FIGURE 17-1: MSSPx BLOCK DIAGRAM (SPI MODE)



Note: Refer to the device data sheet for pin multiplexing.

FIGURE 17-2: SPI MASTER/SLAVE CONNECTION

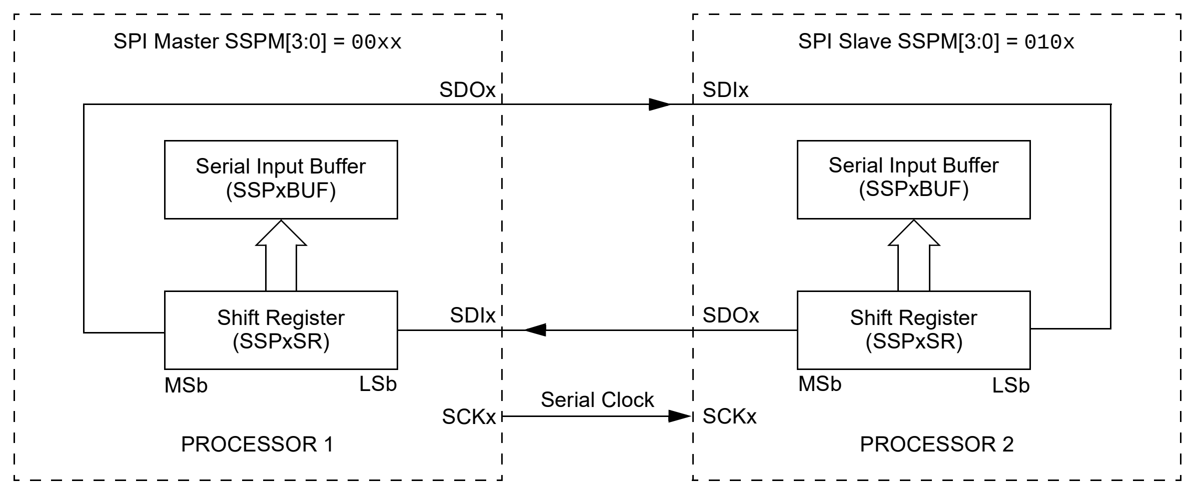
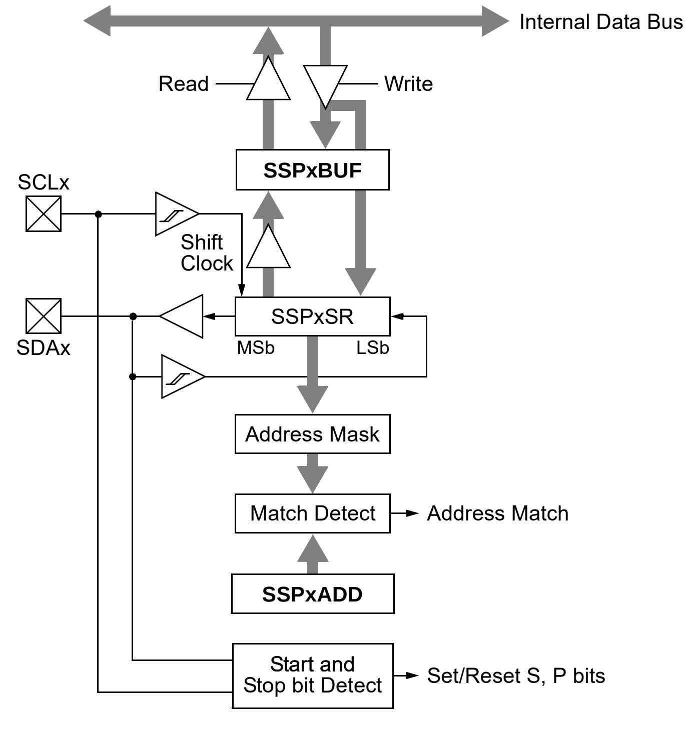
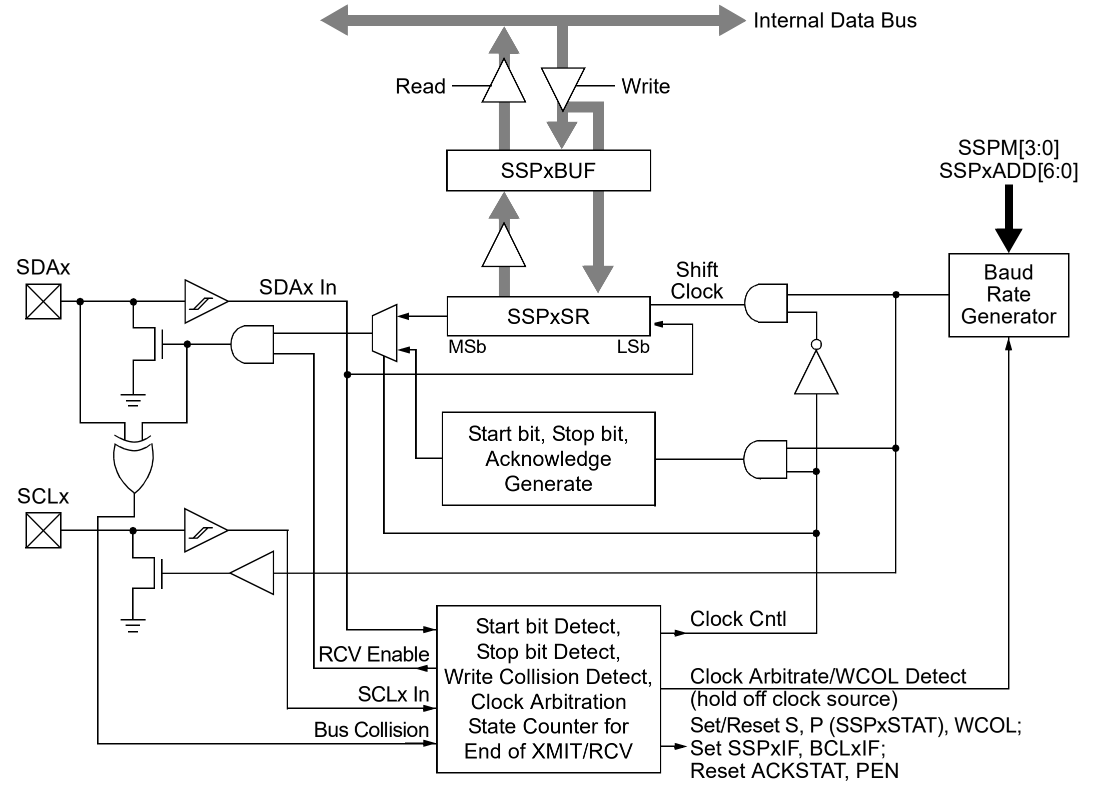


FIGURE 17-3: MSSPx BLOCK DIAGRAM (I^2C MODE)



Note: Only port I/O names are shown in this diagram. Refer to the text for a full list of multiplexed functions.

FIGURE 17-4: MSSPx BLOCK DIAGRAM (I^2C MASTER MODE)



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(1)	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	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 are sampled at the end of data output time `0` = Input data are 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(1) `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/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ bit Used in I^{2}C mode only.
bit 4	**P:** Stop bit Used in I^{2}C mode only. This bit is cleared when the MSSPx module is disabled; SSPEN is cleared.
bit 3	**S:** Start bit Used in I^{2}C mode only.
bit 2	**R/$\overline{\text{W}}$:** Read/$\overline{\text{Write}}$ Information bit Used in I^{2}C mode only.
bit 1	**UA:** Update Address bit Used in I^{2}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]).

REGISTER 17-2: SSPxSTAT: MSSPx STATUS REGISTER (I^2C 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/$\overline{\text{A}}$	P(1)	S(1)	R/$\overline{\text{W}}$	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/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ 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(1)

`1` = Indicates that a Stop bit has been detected last
`0` = Stop bit was not detected last

bit 3 **S:** Start bit(1)

`1` = Indicates that a Start bit has been detected last
`0` = Start bit was not detected last

bit 2 **R/$\overline{\text{W}}$:** Read/$\overline{\text{Write}}$ Information bit

In Slave mode:(2)
`1` = Read
`0` = Write

In Master mode:(3)
`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/$\overline{\text{W}}$ 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.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

REGISTER 17-2: SSPxSTAT: MSSPx STATUS REGISTER (I^2C 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.

2: This bit holds the R/$\overline{\text{W}}$ 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.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSPx is in Active mode.

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(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
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(1)

<u>SPI Slave mode:</u>
`1` = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR are 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(2)

`1` = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{\text{SSx}}$ 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(3)

`1010` = SPI Master mode, Clock = FOSC/(2 * ([SSPxADD] + 1))(4)
`0101` = SPI Slave mode, Clock = SCKx pin; $\overline{\text{SSx}}$ pin control is disabled, $\overline{\text{SSx}}$ can be used as an I/O pin
`0100` = SPI Slave mode, Clock = SCKx pin; $\overline{\text{SSx}}$ pin control is enabled
`0011` = SPI Master mode, Clock = TMR2 output/2
`0010` = SPI Master mode, Clock = FOSC/32
`0001` = SPI Master mode, Clock = FOSC/8
`0000` = SPI Master mode, Clock = FOSC/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^{2}C mode only.

4: SSPxADD value of ‘`0`’ is not supported when the Baud Rate Generator is used in SPI mode.

REGISTER 17-4: SSPxCON1: MSSPx CONTROL REGISTER 1 (I^2C 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(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
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^2C 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(1)

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(2)

1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts is enabled
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts is enabled
1011 = I^2C Firmware Controlled Master mode (Slave Idle)
1000 = I^2C Master mode, Clock = FOSC/(2 * ([SSPxADD] + 1))(3)
0111 = I^2C Slave mode, 10-bit address
0110 = I^2C Slave mode, 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.
2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.
3: SSPxADD values of 0, 1 or 2 are not supported when the Baud Rate Generator is used with I^2C mode.

REGISTER 17-5: SSPxCON2: MSSPx CONTROL REGISTER 2 (I^2C 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
GCEN	ACKSTAT	ACKDT(1)	ACKEN(2)	RCEN(2)	PEN(2)	RSEN(2)	SEN(2)
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 **GCEN:** General Call Enable bit (Slave mode only)
1 = Enables interrupt when a general call address (0000h) is received in the SSPxSR
0 = General call address is disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)(1)
1 = No Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master mode only)(2)
1 = Initiates Acknowledge sequence on SDAx and SCLx pins, and transmits ACKDT data bit; automatically cleared by hardware
0 = Acknowledge sequence is Idle

bit 3 **RCEN:** Receive Enable bit (Master Receive mode only)(2)
1 = Enables Receive mode for I^2C
0 = Receive is Idle

bit 2 **PEN:** Stop Condition Enable bit (Master mode only)(2)
1 = Initiates Stop condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Stop condition is Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (Master mode only)(2)
1 = Initiates Repeated Start condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Repeated Start condition is Idle

bit 0 **SEN:** Start Condition Enable bit(2)
<u>Master Mode:</u>
1 = Initiates Start condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Start condition is Idle
<u>Slave Mode:</u>
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch is enabled)
0 = Clock stretching is disabled

Note 1: The value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.

2: If the I^2C module is active, these bits may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

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(1)	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^2C mode only)
Unused in SPI mode.

bit 6 **PCIE:** Stop Condition Interrupt Enable bit (I^2C mode only)
Unused in SPI mode.

bit 5 **SCIE:** Start Condition Interrupt Enable bit (I^2C mode only)
Unused in SPI mode.

bit 4 **BOEN:** Buffer Overwrite Enable bit(1)
<u>In SPI Slave mode:</u>
`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^2C mode only)
Unused in SPI mode.

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^2C Slave mode only)
Unused in SPI mode.

bit 1 **AHEN:** Address Hold Enable bit (I^2C 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.

REGISTER 17-7: SSPxCON3: MSSPx CONTROL REGISTER 3 (I^2C 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(2)	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(2)

`1` = Indicates the I^2C bus is in an Acknowledge sequence, set on the 8th falling edge of the SCLx clock
`0` = Not an Acknowledge sequence, cleared on the 9th 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(1)

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(1)

bit 4 **BOEN:** Buffer Overwrite Enable bit

I^2C Master mode:
This bit is ignored.
I^2C 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.

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^{2}C™ Master modes:

Reloads value for Baud Rate Generator. Clock period is (([SPxADD] + 1) *2)/FOSC.

I^{2}C Slave modes:

Represents seven or eight 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^{2}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](1)							
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(1)

`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.

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[(1)]	SCK2DIS[(1)]	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(1)

`1` = The SPI output data (SDO2) of MSSP2 to the pin are disabled
`0` = The SPI output data (SDO2) of MSSP2 are output to the pin

bit 10 **SCK2DIS:** MSSP2 SCK2 Pin Disable bit(1)

`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 are disabled
`0` = The SPI output data (SDO1) of MSSP1 are 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.

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 **"UART"** (www.microchip.com/DS39708) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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 UxCTS and UxRTS pins, and also includes an IrDA® encoder and decoder.

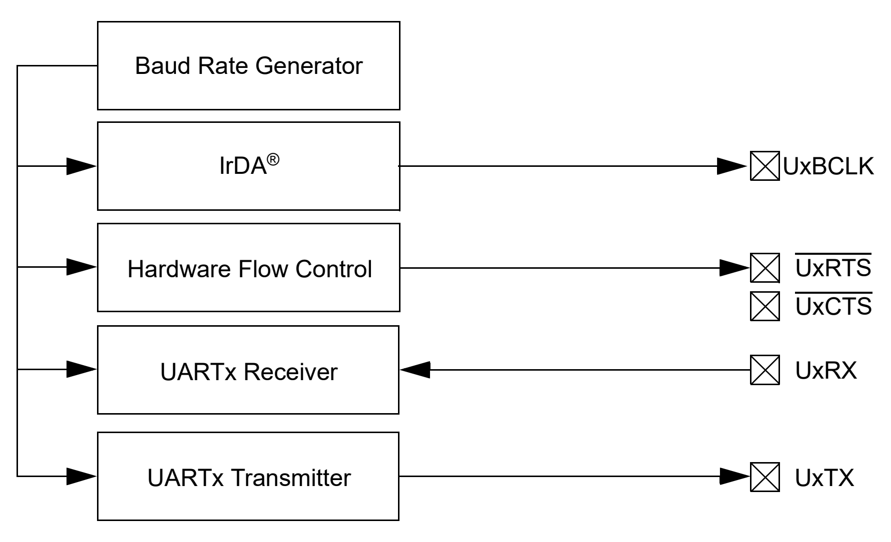
The primary features of the UART module are:

- Full-Duplex, 8-Bit or 9-Bit Data Transmission Through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with UxCTS and 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 (9th 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® 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



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`(1)

$$\text{Baud Rate} = \frac{F_{CY}}{16 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{16 \cdot \text{Baud Rate}} - 1$$

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

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`(1)

$$\text{Baud Rate} = \frac{F_{CY}}{4 \cdot (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{4 \cdot \text{Baud Rate}} - 1$$

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.

EXAMPLE 18-1: BAUD RATE ERROR CALCULATION (BRGH = `0`)(1)

$$\begin{array}{lll}
\text{Desired Baud Rate} & = & F_{CY}/(16\ (\text{UxBRG} + 1)) \\
\text{Solving for UxBRG Value:} & & \\
\quad\text{UxBRG} & = & ((F_{CY}/\text{Desired Baud Rate})/16) - 1 \\
\quad\text{UxBRG} & = & ((4000000/9600)/16) - 1 \\
\quad\text{UxBRG} & = & 25 \\
\text{Calculated Baud Rate} & = & 4000000/(16\ (25 + 1)) \\
 & = & 9615 \\
\text{Error} & = & \dfrac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}} \\
 & = & (9615 - 9600)/9600 \\
 & = & 0.16\%
\end{array}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

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.

REGISTER 18-1: UxMODE: UARTx MODE REGISTER

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

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

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

bit 15 **UARTEN:** UARTx Enable bit

1 = UARTx is enabled; all UARTx pins are controlled by UARTx as defined by UEN[1:0]
0 = UARTx is disabled; all UARTx pins are controlled by port latches, UARTx power consumption is minimal

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **USIDL:** UARTx Stop in Idle Mode bit

1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **IREN:** IrDA® Encoder and Decoder Enable bit(1)

1 = IrDA encoder and decoder are enabled
0 = IrDA encoder and decoder are disabled

bit 11 **RTSMD:** Mode Selection for $\overline{\text{UxRTS}}$ Pin bit

1 = $\overline{\text{UxRTS}}$ pin is in Simplex mode
0 = $\overline{\text{UxRTS}}$ pin is in Flow Control mode

bit 10 **Unimplemented:** Read as ‘0’

bit 9-8 **UEN[1:0]:** UARTx Enable bits(2)

11 = UxTX, UxRX and UxBCLK pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
10 = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
01 = UxTX, UxRX and $\overline{\text{UxRTS}}$ pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
00 = UxTX and UxRX pins are enabled and used; $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$/UxBCLK pins are controlled by port latches

bit 7 **WAKE:** Wake-up on Start Bit Detect During Sleep Mode Enable bit

1 = UARTx will continue to sample the UxRX pin; interrupt is generated on the falling edge, bit is cleared in hardware on the following rising edge
0 = No wake-up is enabled

bit 6 **LPBACK:** UARTx Loopback Mode Select bit

1 = Enables Loopback mode
0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto-Baud Enable bit

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

bit 4 **RXINV:** Receive Polarity Inversion bit

1 = UxRX Idle state is ‘0’
0 = UxRX Idle state is ‘1’

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

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 is only available for the 16x BRG mode (BRGH = `0`).

2: Bit availability depends on pin availability.

REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	HC/R/W-0	R/W-0	HSC/R-0	HSC/R-1
UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	HSC/R-1	HSC/R-0	HSC/R-0	HS/R/C-0	HSC/R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
HS = Hardware Settable bit	C = Clearable bit	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 15,13 **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® 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 two 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

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

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 **"10-Bit A/D Converter"** (www.microchip.com/DS39705) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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

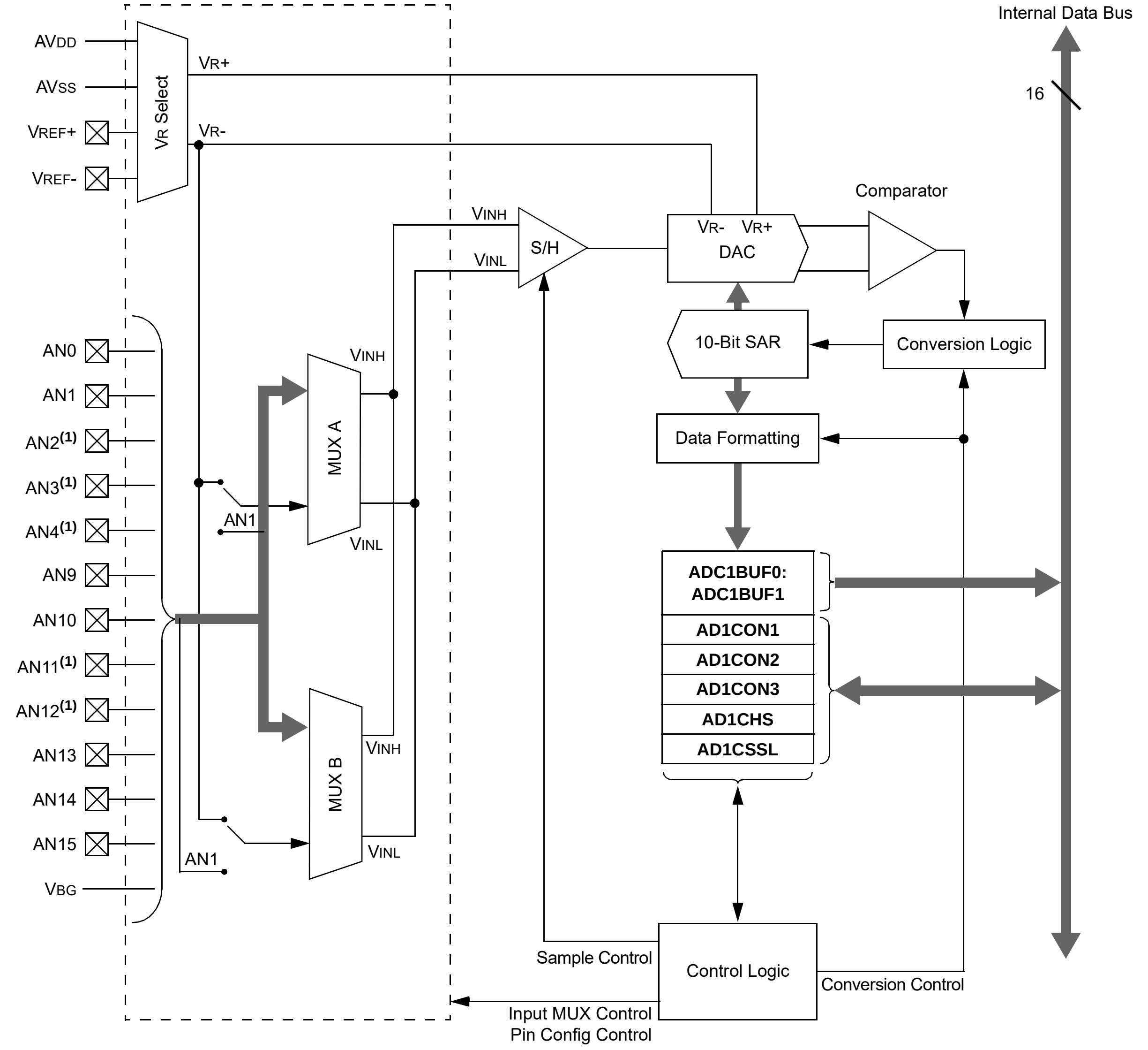
Depending on the particular device, PIC24F16KL402 family devices implement up to 12 analog input pins, designated AN0 through AN4 and AN9 through AN15. In addition, there are two analog input pins for external voltage reference connections (VREF+ and VREF-). These voltage reference inputs may be shared 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.

FIGURE 19-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM



Note 1: Unimplemented in 14-pin devices.

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[(1)]	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	HSC/R/W-0	HSC/R-0
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	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(1) `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 INT0 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.

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[1]	—	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
—	—	SMPI3	SMPI2	SMPI1	SMPI0	—	ALTS
bit 7							bit 0

Legend:	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 15-13 **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[1]

`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 2nd 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.

REGISTER 19-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	—	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
—	—	ADCS[5:0]					
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 **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 **Unimplemented:** Maintain as ‘`0`’

bit 12-8 **SAMC[4:0]:** Auto-Sample Time bits

`11111` = 31 T_{AD}
•
•
•
`00001` = 1 T_{AD}
`00000` = 0 T_{AD} (not recommended)

bit 7-6 **Unimplemented:** Maintain as ‘`0`’

bit 5-0 **ADCS[5:0]:** A/D Conversion Clock Select bits

`111111` = 64 • T_{CY}
`111110` = 63 • T_{CY}
•
•
•
`000001` = 2 • T_{CY}
`000000` = T_{CY}

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(1)
`1011` = AN11(1)
`1010` = AN10
`1001` = AN9
`1000` = Upper guardband rail (0.785 * VDD)
`0111` = Lower guardband rail (0.215 * VDD)
`0110` = Internal band gap reference (VBG)
`0101` = Reserved; do not use
`0100` = AN4(1)
`0011` = AN3(1)
`0010` = AN2(1)
`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.

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](1)							
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](1)				
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(1)

`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(1)

`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

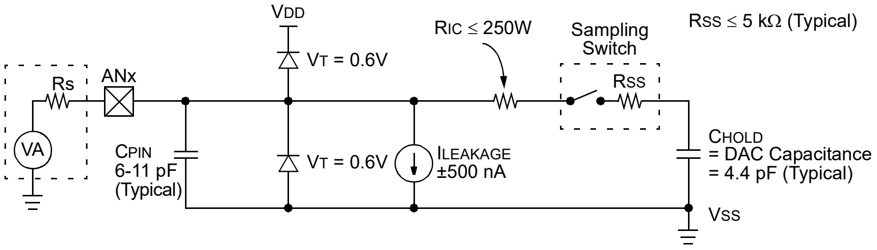
EQUATION 19-1: A/D CONVERSION CLOCK PERIOD[1]

$$\text{ADCS} = \frac{T_{AD}}{T_{CY}} - 1$$

$$T_{AD} = T_{CY} \bullet (\text{ADCS} + 1)$$

Note 1: Based on $T_{CY} = 2 * T_{OSC}$; Doze mode and PLL are disabled.

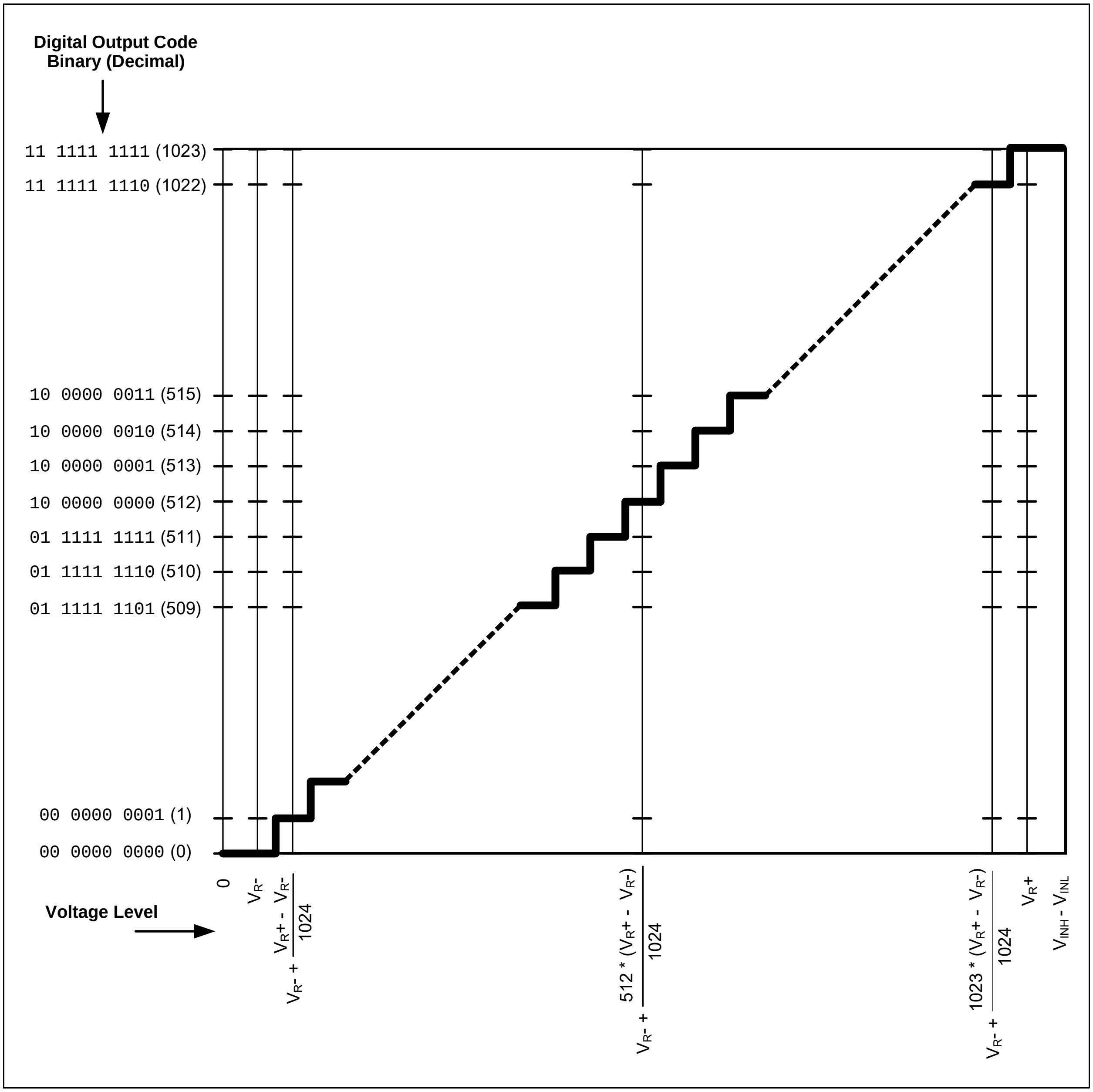
FIGURE 19-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL



Legend:
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 Rs ≤ 5 kΩ.

FIGURE 19-3: A/D TRANSFER FUNCTION



NOTES:

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 **"Scalable Comparator Module"** (www.microchip.com/DS39734) in the *"dsPIC33/PIC24 Family Reference Manual"*.

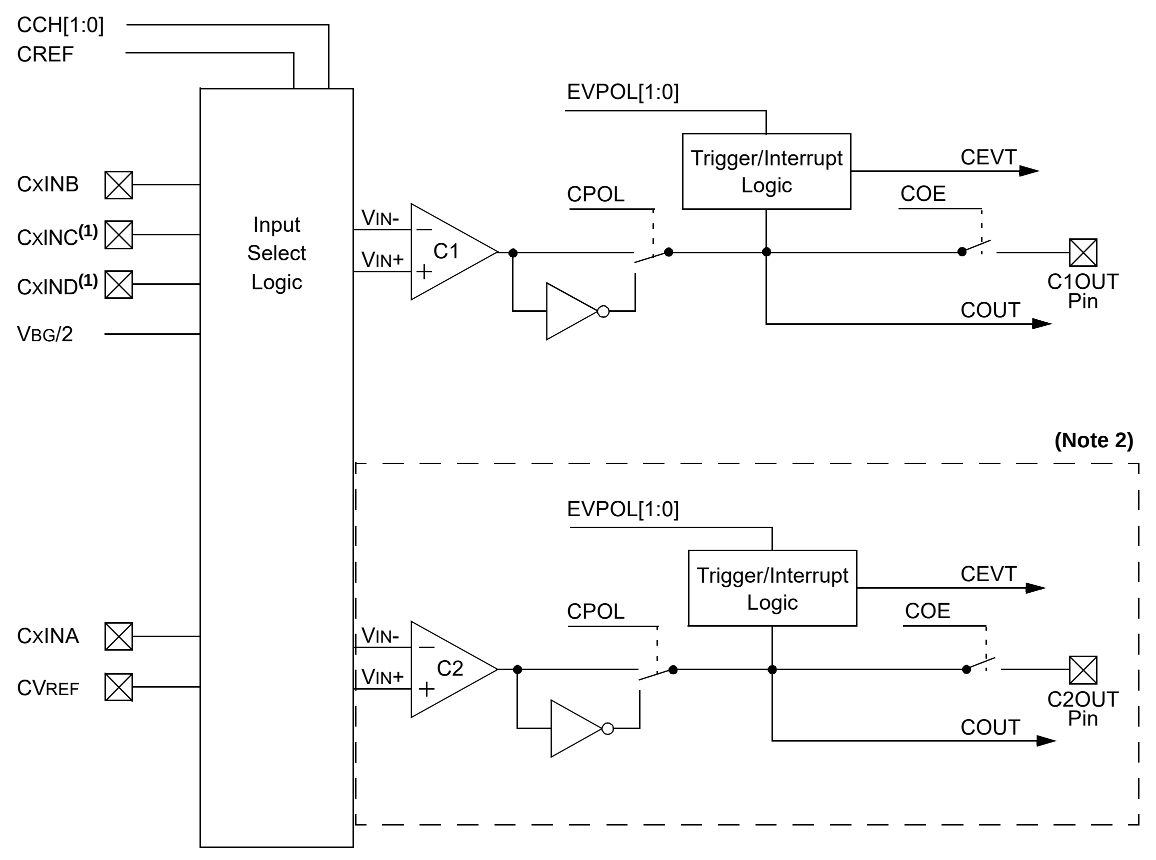
Depending on the particular device, the comparator module provides one or two analog comparators. The inputs to the comparator can be configured to use any one of up to four external analog inputs, as well as a voltage reference input from either the internal band gap reference, divided by 2 (VBG/2), 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).

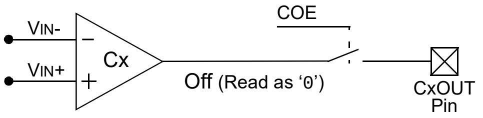
FIGURE 20-1: COMPARATOR MODULE BLOCK DIAGRAM



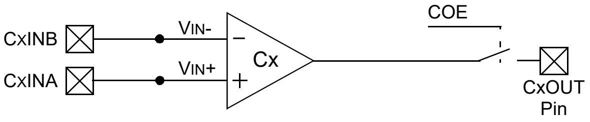
Note 1: These inputs are unavailable on 14-pin (PIC24FXXKL100/200) devices.
2: Comparator 2 is unimplemented on PIC24FXXKL10X/20X devices.

FIGURE 20-2: INDIVIDUAL COMPARATOR CONFIGURATIONS

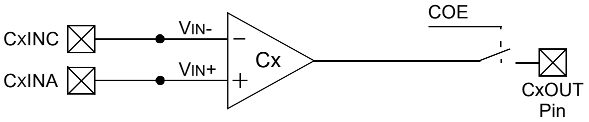
Comparator Off
CON = 0, CREF = x, CCH[1:0] = xx



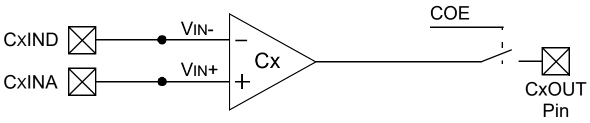
Comparator CxINB > CxINA Compare
CON = 1, CREF = 0, CCH[1:0] = 00



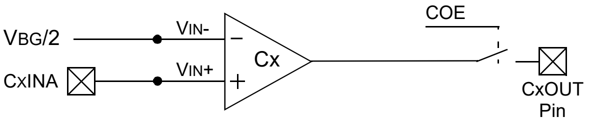
Comparator CxINC > CxINA Compare(1)
CON = 1, CREF = 0, CCH[1:0] = 01



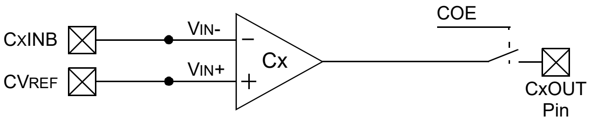
Comparator CxIND > CxINA Compare(1)
CON = 1, CREF = 0, CCH[1:0] = 10



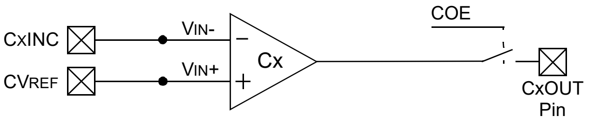
Comparator VBG > CxINA Compare
CON = 1, CREF = 0, CCH[1:0] = 11



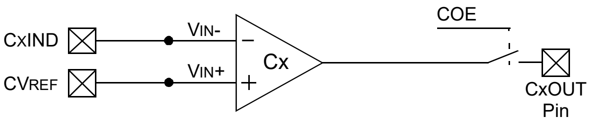
Comparator CxINB > CVREF Compare
CON = 1, CREF = 1, CCH[1:0] = 00



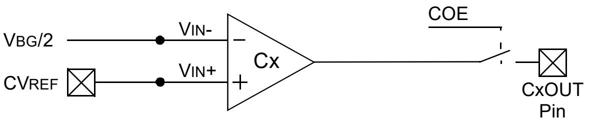
Comparator CxINC > CVREF Compare(1)
CON = 1, CREF = 1, CCH[1:0] = 01



Comparator CxIND > CVREF Compare(1)
CON = 1, CREF = 1, CCH[1:0] = 10



Comparator VBG > CVREF Compare
CON = 1, CREF = 1, CCH[1:0] = 11



Note 1: This configuration is unavailable on 14-pin (PIC24FXXKL100/200) devices.

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(1)	EVPOL0(1)	—	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 = VIN+ > VIN-
0 = VIN+ < VIN-
When CPOL = 1:
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6 **EVPOL[1:0]:** Trigger/Event/Interrupt Polarity Select bits(1)
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 (noninverting input)
1 = Noninverting input connects to the internal CVREF voltage
0 = Noninverting 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.

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 V_{BG}/2 `10` = Inverting input of the comparator connects to the CxIND pin(2) `01` = Inverting input of the comparator connects to the CxINC pin(2) `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	HSC/R-0	HSC/R-0
CMIDL	—	—	—	—	—	C2EVT(1)	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	HSC/R-0	HSC/R-0
—	—	—	—	—	—	C2OUT(1)	C1OUT
bit 7							bit 0

Legend:	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)(1) 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)(1) 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.

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 **"Comparator Voltage Reference Module"** (www.microchip.com/DS39709) in the *"dsPIC33/PIC24 Family Reference Manual"*.

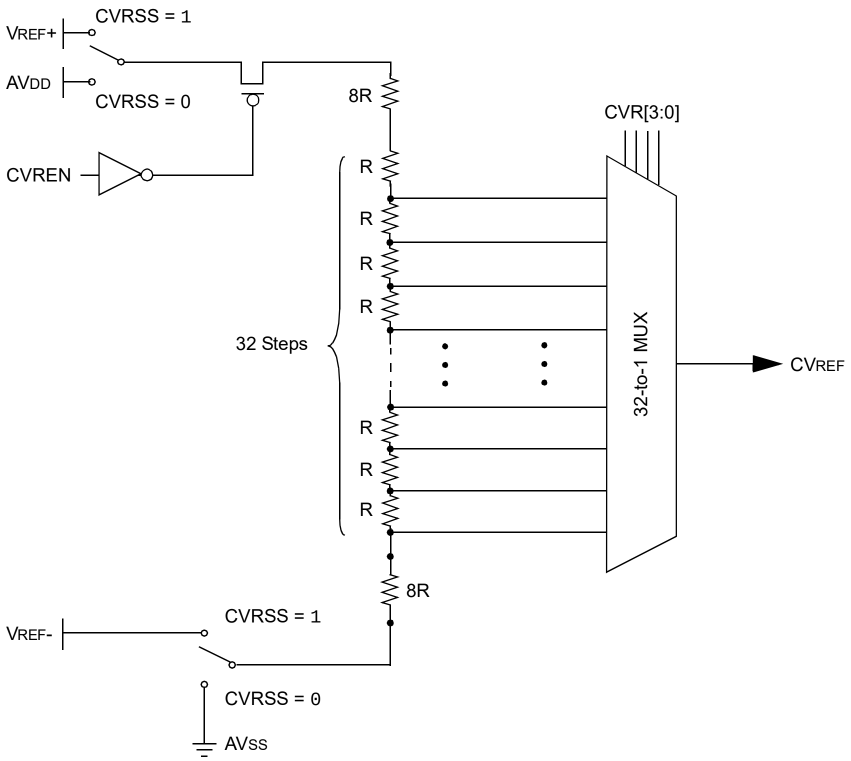
21.1 Configuring the Comparator Voltage Reference

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



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 ≤ CVR[4:0] ≤ 31 bits When CVRSS = 1: CVREF = (VREF-) + (CVR[4:0]/32) • (VREF+ – VREF-) When CVRSS = 0: CVREF = (AVSS) + (CVR[4:0]/32) • (AVDD – AVSS)

22.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

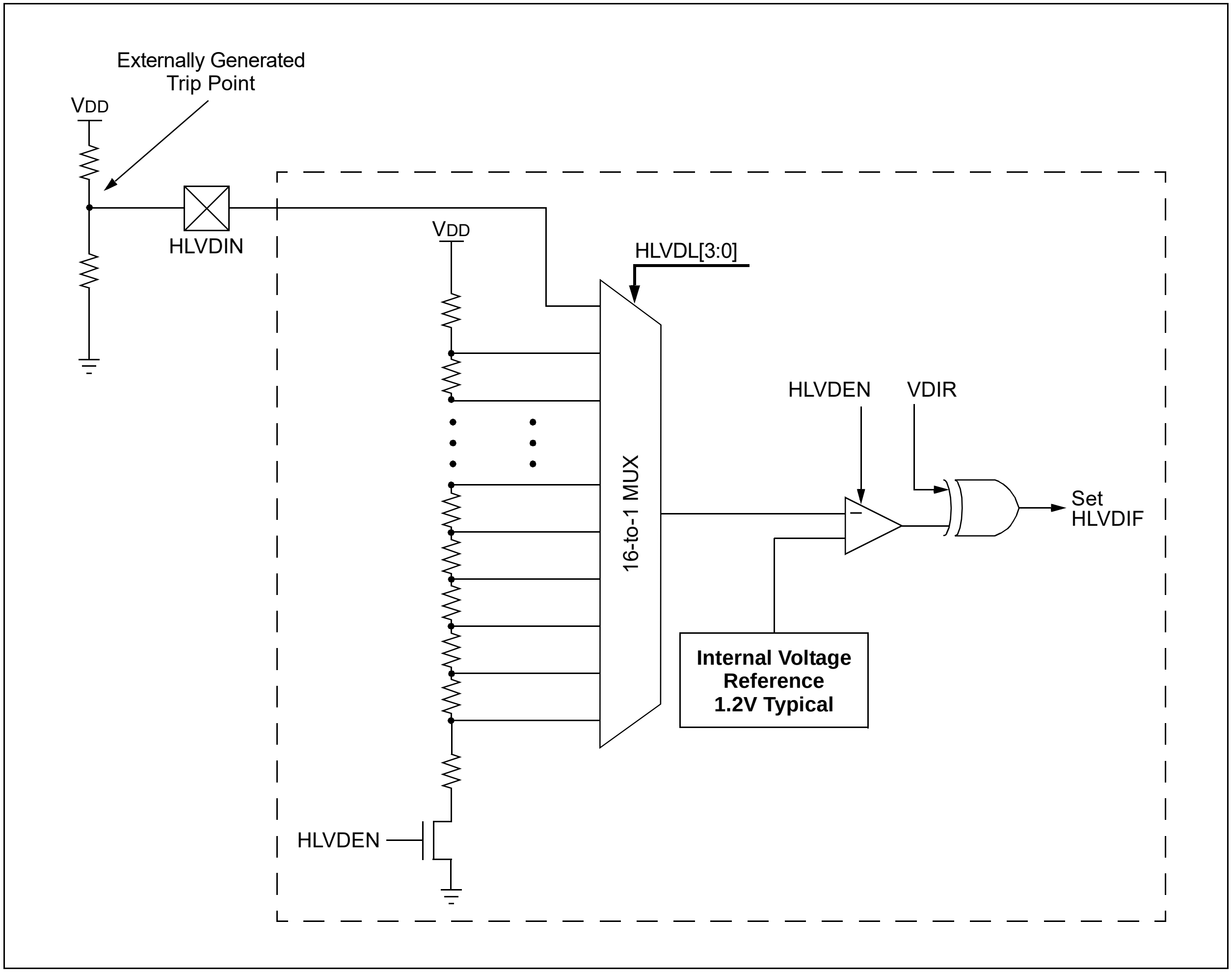
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the High/Low-Voltage Detect, refer to **"High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)"** (www.microchip.com/DS39725) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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.

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.

FIGURE 22-1: HIGH/LOW-VOLTAGE DETECT (HLVD) MODULE BLOCK DIAGRAM



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(1)
`1101` = Trip Point 13(1)
`1100` = Trip Point 12(1)
.
.
.
`0000` = Trip Point 0(1)

Note 1: For the actual trip point, see **Section 26.0 “Electrical Characteristics”**.

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)"** (www.microchip.com/DS39697)
> - **"High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)"** (www.microchip.com/DS39725)
> - **"Programming and Diagnostics"** (www.microchip.com/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

> **Note:** Code-protect bits (BSS, BWRP, GSS, GWRP) are in a group that are subject to write restrictions. If any bit is cleared, the rest cannot be cleared on a subsequent operation. All bits must be cleared using one operation.

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 **"CodeGuard™ Security"** (www.microchip.com/DS70199) in the *"dsPIC33/PIC24 Family Reference Manual"*.

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 REGISTER LOCATIONS

Configuration Register	Address
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E

REGISTER 23-1: FBS: BOOT SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	R/C-1[1]	R/C-1[1]	R/C-1[1]	R/C-1[1]
—	—	—	—	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[1]

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[2]
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[2]
000 = Reserved

bit 0 **BWRP:** Boot Segment Program Flash Write Protection bit[1]

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[1]	R/C-1[1]
—	—	—	—	—	—	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[1]

1 = No protection
0 = Standard security is enabled

bit 0 **GWRP:** General Segment Code Flash Write Protection bit[1]

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.

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)

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

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 `CLRWDT` 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

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(1)	BORV1(2)	BORV0(2)	I2C1SEL(3)	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(1)

`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 Voltage Threshold bits(2)

`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^2C Pin Mapping bit(3)

`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^2C functionality to be available.

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
$\overline{\text{DEBUG}}$	—	—	—	—	—	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 **$\overline{\text{DEBUG}}$:** 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(1)
`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.

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
FAMID[7:0]							
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV[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 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

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
—	—	—	—	REV[3:0]			
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 23-4	**Unimplemented:** Read as ‘0’
bit 3-0	**REV[3:0]:** Revision Identifier bits

23.4 Watchdog Timer (WDT)

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

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

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

The WDT, prescaler and postscaler are reset:

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

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

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

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

23.4.1 WINDOWED OPERATION

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

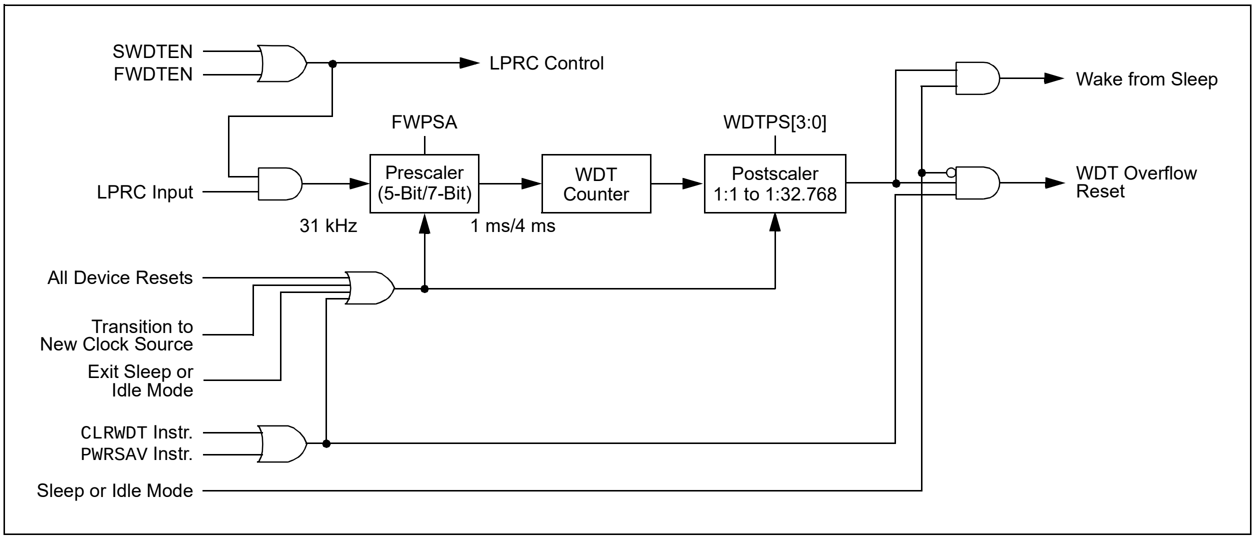
Windowed WDT mode is enabled by programming the Configuration bit, WINDIS (FWDT[6]), to ‘`0`’.

23.4.2 CONTROL REGISTER

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

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

FIGURE 23-1: WDT BLOCK DIAGRAM



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 $\overline{\text{MCLR}}$, V_{DD}, V_{SS}, 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.

NOTES:

24.0 DEVELOPMENT SUPPORT

Move a design from concept to production in record time with Microchip's award-winning development tools. Microchip tools work together to provide state of the art debugging for any project with easy-to-use Graphical User Interfaces (GUIs) in our free MPLAB® X and Atmel Studio Integrated Development Environments (IDEs), and our code generation tools. Providing the ultimate ease-of-use experience, Microchip's line of programmers, debuggers and emulators work seamlessly with our software tools. Microchip development boards help evaluate the best silicon device for an application, while our line of third party tools round out our comprehensive development tool solutions.

Microchip's MPLAB X and Atmel Studio ecosystems provide a variety of embedded design tools to consider, which support multiple devices, such as PIC® MCUs, AVR® MCUs, SAM MCUs and dsPIC® DSCs. MPLAB X tools are compatible with Windows®, Linux® and Mac® operating systems while Atmel Studio tools are compatible with Windows.

Go to the following website for more information and details:

https://www.microchip.com/development-tools/

NOTES:

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® 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 eight MSbs are ‘`0`’s. If this second word is executed as an instruction (by itself), it will execute as a `NOP`.

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 `NOP`. 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.

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) ∈ {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 ∈ {0000h...1FFFh}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSB must be '`0`'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor Working register pair (direct addressing)
Wn	One of 16 Working registers ∈ {W0..W15}
Wnd	One of 16 Destination Working registers ∈ {W0..W15}
Wns	One of 16 Source Working registers ∈ {W0..W15}
WREG	W0 (Working register used in File register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

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

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 = $\overline{f}$	1	1	N, Z
	COM f,WREG	WREG = $\overline{f}$	1	1	N, Z
	COM Ws,Wd	Wd = $\overline{Ws}$	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 – $\overline{C}$)	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 ≠	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

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 = 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} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL f	W3:W2 = f * WREG	1	1	None
NEG	NEG f	f = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG f,WREG	WREG = $\overline{f}$ + 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

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

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

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
`TBLRDH`	`TBLRDH Ws,Wd`	Read Prog[23:16] to Wd[7:0]	1	2	None
`TBLRDL`	`TBLRDL Ws,Wd`	Read Prog[15:0] to Wd	1	2	None
`TBLWTH`	`TBLWTH Ws,Wd`	Write Ws[7:0] to Prog[23:16]	1	2	None
`TBLWTL`	`TBLWTL Ws,Wd`	Write Ws to Prog[15:0]	1	2	None
`ULNK`	`ULNK`	Unlink Frame Pointer	1	1	None
`XOR`	`XOR f`	f = f .XOR. WREG	1	1	N, Z
	`XOR f,WREG`	WREG = f .XOR. WREG	1	1	N, Z
	`XOR #lit10,Wn`	Wd = lit10 .XOR. Wd	1	1	N, Z
	`XOR Wb,Ws,Wd`	Wd = Wb .XOR. Ws	1	1	N, Z
	`XOR Wb,#lit5,Wd`	Wd = Wb .XOR. lit5	1	1	N, Z
`ZE`	`ZE Ws,Wnd`	Wnd = Zero-Extend Ws	1	1	C, Z, N

NOTES:

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(†)

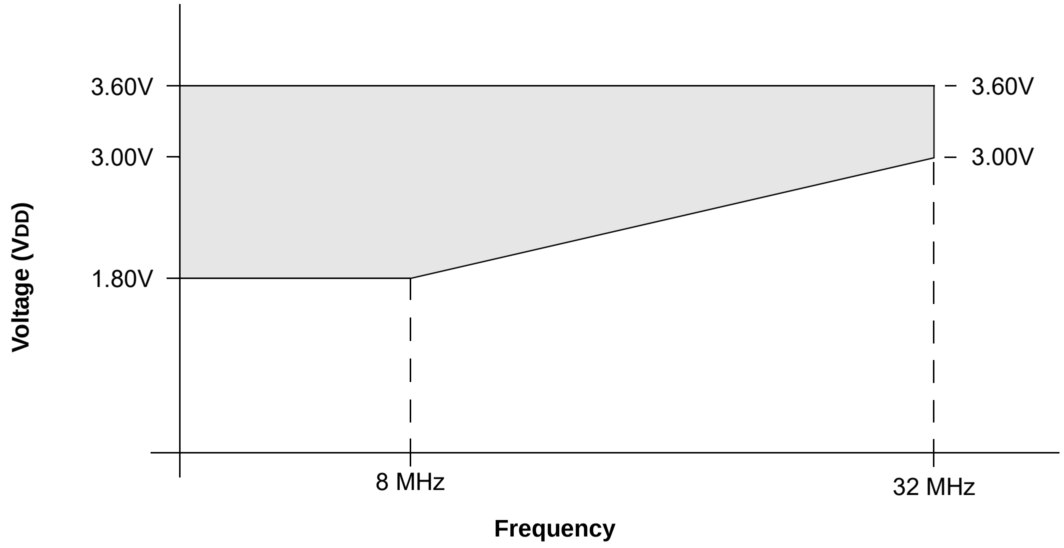
Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on V_{DD} with respect to V_{SS}	-0.3V to +4.5V
Voltage on any combined analog and digital pin with respect to V_{SS}	-0.3V to (V_{DD} + 0.3V)
Voltage on any digital only pin with respect to V_{SS}	-0.3V to (V_{DD} + 0.3V)
Voltage on $\overline{MCLR}$/V_{PP} pin with respect to V_{SS}	-0.3V to +9.0V
Maximum current out of V_{SS} pin	300 mA
Maximum current into V_{DD} pin(1)	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(1)	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.

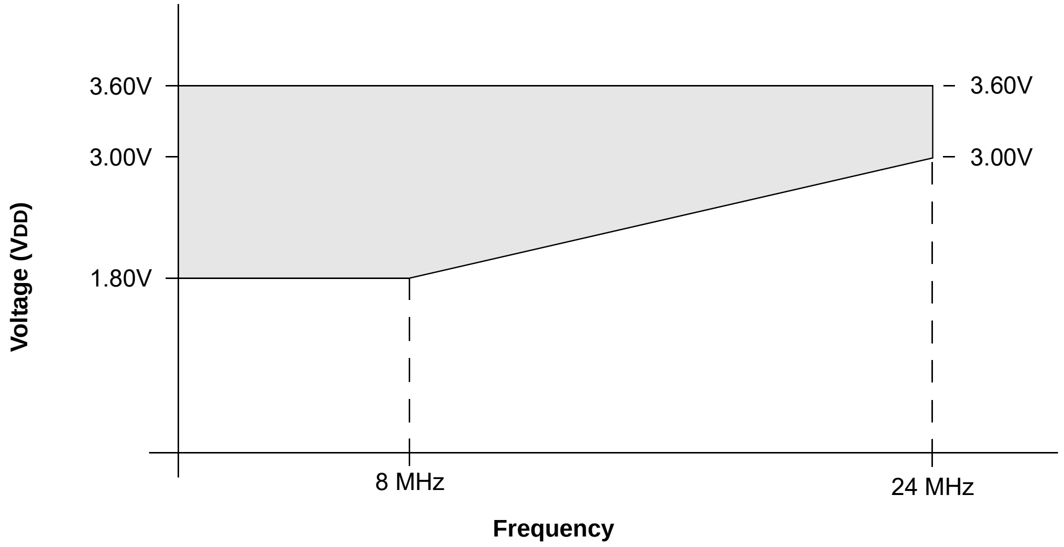
26.1 DC Characteristics

FIGURE 26-1: PIC24F16KL402 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



Note: For frequencies between 8 MHz and 32 MHz, FMAX = 20 MHz * (VDD – 1.8) + 8 MHz.

FIGURE 26-2: PIC24F16KL402 FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED)



Note: For frequencies between 8 MHz and 24 MHz, FMAX = 13.33 MHz * (VDD – 1.8) + 8 MHz.

TABLE 26-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
Operating Junction Temperature Range	T_J	-40	—	+140	°C
Operating Ambient Temperature Range	T_A	-40	—	+125	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

TABLE 26-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 20-Pin PDIP	θ_{JA}	62.4	—	°C/W	1
Package Thermal Resistance, 28-Pin SPDIP	θ_{JA}	60	—	°C/W	1
Package Thermal Resistance, 20-Pin SSOP	θ_{JA}	108	—	°C/W	1
Package Thermal Resistance, 28-Pin SSOP	θ_{JA}	71	—	°C/W	1
Package Thermal Resistance, 20-Pin SOIC	θ_{JA}	75	—	°C/W	1
Package Thermal Resistance, 28-Pin SOIC	θ_{JA}	80.2	—	°C/W	1
Package Thermal Resistance, 20-Pin QFN	θ_{JA}	43	—	°C/W	1
Package Thermal Resistance, 28-Pin QFN	θ_{JA}	32	—	°C/W	1
Package Thermal Resistance, 14-Pin PDIP	θ_{JA}	62.4	—	°C/W	1
Package Thermal Resistance, 14-Pin TSSOP	θ_{JA}	108	—	°C/W	1

Note 1: Junction to ambient thermal resistance, Theta-JA (θ_{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 ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ[(1)]**	**Max**	**Units**	**Conditions**
DC10	V_{DD}	**Supply Voltage**	1.8	—	3.6	V	
DC12	V_{DR}	**RAM Data Retention Voltage[(2)]**	1.5	—	—	V	
DC16	V_{POR}	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	V_{SS}	—	0.7	V	
DC17	SV_{DD}	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 0.1s 0-2.5V in 60 ms
	V_{BG}	**Band Gap Voltage Reference**	1.14	1.2	1.26	V	

Note 1: Data in “Typ” column are 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 VDD can be lowered without losing RAM data.

TABLE 26-4: HIGH/LOW–VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ T_A ≤ +85°C for industrial
-40°C ≤ T_A ≤ +125°C for Extended

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
DC18	V_{HLVD}	HLVD Voltage on V_{DD} 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°C ≤ T_A ≤ +85°C for Industrial
-40°C ≤ T_A ≤ +125°C for Extended

Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
DC19		BOR Voltage on V_{DD} Transition	BORV = `00`	1.85	2.0	2.15	V	Note 1
			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.

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

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
I_{DD} Current						
DC20	0.154	0.350	mA	1.8V	+85V°C	0.5 MIPS, F_{OSC} = 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, F_{OSC} = 2 MHz
	0.585	—		3.3V		
DC24	7.76	12.0	mA	3.3V	+85°C	16 MIPS, F_{OSC} = 32 MHz
	—	18.0		3.3V	+125°C	
DC26	1.44	—	mA	1.8V	+85°C	FRC (4 MIPS), F_{OSC} = 8 MHz
	2.71	—		3.3V		
DC30	4.00	28.0	µA	1.8V	+85°C	LPRC (15.5 KIPS), F_{OSC} = 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 are at 3.3V, +25°C, unless otherwise stated.

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

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

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

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

2: I_{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.

TABLE 26-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (I_{PD})

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended			
Parameter No.	**Typical**[1]	**Max**	**Units**	**Conditions**		
Power-Down Current (I_{PD})						
DC60	0.01	0.20	µA	-40°C	1.8V	Sleep Mode[2]
	0.03	0.20	µA	+25°C		
	0.06	0.87	µA	+60°C		
	0.20	1.35	µA	+85°C		
	—	8.00	µA	+125ºC		
	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	+125ºC		

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

2: Base I_{PD} 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

TABLE 26-9: DC CHARACTERISTICS: POWER-DOWN CURRENT (I_{PD})

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Module Differential Current (ΔI_{PD})						
DC71	0.21	0.65	μA	1.8V	+85°C	Watchdog Timer Current: ΔWDT(2,3)
	0.45	0.95	μA	3.3V		
	—	1.30	μA	1.8V	+125°C	
	—	1.50	μA	3.3V		
DC72	0.69	1.50	μA	1.8V	+85°C	32 kHz Crystal with Timer1: ΔSOSC (SOSCSEL = `0`)(2)
	1.00	1.50	μA	3.3V		
DC75	5.24	—	μA	1.8V	+85°C	ΔHLVD(2,3)
	5.16	11.00	μA	3.3V		
	—	12.00	μA	1.8V	+125°C	
	—	15.00	μA	3.3V		
DC76	4.15	9.00	μA	3.3V	+85°C	ΔBOR(2,3)
	—	11.0	μA	3.3V	+125°C	
DC78	0.03	0.20	μA	1.8V	+85°C	ΔLPBOR(2)
	0.03	0.20	μA	3.3V		
	—	0.40	μA	1.8V	+125°C	
	—	0.40	μA	3.3V		

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

2: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base I_{PD} current.

3: This current applies to Sleep only.

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[1]**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage[4]**					
DI10		I/O Pins	VSS	—	0.2 VDD	V	
DI15		$\overline{\text{MCLR}}$	VSS	—	0.2 VDD	V	
DI16		OSCI (XT mode)	VSS	—	0.2 VDD	V	
DI17		OSCI (HS mode)	VSS	—	0.2 VDD	V	
DI18		I/O Pins with I^{2}C Buffer	VSS	—	0.3 VDD	V	SMBus disabled
DI19		I/O Pins with SMBus Buffer	VSS	—	0.8	V	SMBus enabled
	VIH	**Input High Voltage[4,5]**					
DI20		I/O Pins:					
		with Analog Functions	0.8 VDD	—	VDD	V	
		Digital Only	0.8 VDD	—	VDD	V	
DI25		$\overline{\text{MCLR}}$	0.8 VDD	—	VDD	V	
DI26		OSCI (XT mode)	0.7 VDD	—	VDD	V	
DI27		OSCI (HS mode)	0.7 VDD	—	VDD	V	
DI28		I/O Pins with I^{2}C Buffer:					
		with Analog Functions	0.7 VDD	—	VDD	V	
		Digital Only	0.7 VDD	—	VDD	V	
DI29		I/O Pins with SMBus	2.1	—	VDD	V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	**CNx Pull-up Current**	50	250	500	μA	VDD = 3.3V, VPIN = VSS
DI31	IPU	**Maximum Load Current for Digital High Detection w/Internal Pull-up**	—	—	30	μA	VDD = 2.0V
			—	—	1000	μA	VDD = 3.3V
	IIL	**Input Leakage Current[2,3]**					
DI50		I/O Ports	—	0.050	±0.100	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI51		VREF+, VREF-, AN0, AN1	—	0.300	±0.500	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance

Note 1: Data in "Typ" column are 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: VIH requirements are met when the internal pull-ups are enabled.

TABLE 26-11: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended					
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**	
	V_{OL}	**Output Low Voltage**						
DO10		All I/O Pins	—	—	0.4	V	I_{OL} = 4.0 mA	V_{DD} = 3.6V
			—	—	0.4	V	I_{OL} = 3.5 mA	V_{DD} = 2.0V
DO16		OSC2/CLKO	—	—	0.4	V	I_{OL} = 1.2 mA	V_{DD} = 3.6V
			—	—	0.4	V	I_{OL} = 0.4 mA	V_{DD} = 2.0V
	V_{OH}	**Output High Voltage**						
DO20		All I/O Pins	3	—	—	V	I_{OH} = -3.0 mA	V_{DD} = 3.6V
			1.6	—	—	V	I_{OH} = -1.0 mA	V_{DD} = 2.0V
DO26		OSC2/CLKO	3	—	—	V	I_{OH} = -1.0 mA	V_{DD} = 3.6V
			1.6	—	—	V	I_{OH} = -0.5 mA	V_{DD} = 2.0V

Note 1: Data in “Typ” column are 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 ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
		Program Flash Memory					
D130	E_P	Cell Endurance	10,000[2]	—	—	E/W	
D131	V_{PR}	V_{DD} for Read	V_{MIN}	—	3.6	V	V_{MIN} = Minimum operating voltage
D133A	T_{IW}	Self-Timed Write Cycle Time	—	2	—	ms	
D134	T_{RETD}	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D135	I_{DDP}	Supply Current During Programming	—	10	—	mA	

Note 1: Data in “Typ” column are at 3.3V, +25°C unless otherwise stated.
2: Self-write and block erase.

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[1]**	**Max**	**Units**	**Conditions**
		Data EEPROM Memory					
D140	EPD	Cell Endurance	100,000	—	—	E/W	
D141	VPRD	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
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	Provided no other specifications are violated
D145	IDDPD	Supply Current During Programming	—	7	—	mA	

Note 1: Data in "Typ" column are at 3.3V, +25°C unless otherwise stated.

TABLE 26-14: DC CHARACTERISTICS: COMPARATOR

Standard Operating Conditions: 2.0V < VDD < 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	VIOFF	Input Offset Voltage	—	20	40	mV	
D301	VICM	Input Common-Mode Voltage	0	—	VDD	V	
D302	CMRR	Common-Mode Rejection Ratio	55	—	—	dB	

TABLE 26-15: DC CHARACTERISTICS: COMPARATOR VOLTAGE REFERENCE

Standard Operating Conditions: 2.0V < VDD < 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	—	—	VDD/32	LSb	
VRD311	CVRAA	Absolute Accuracy	—	—	AVDD – 1.5	LSb	
VRD312	CVRUR	Unit Resistor Value (R)	—	2k	—	Ω	

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°C ≤ TA ≤ +85°C for Industrial** Operating voltage VDD range as described in **Section 26.1 "DC Characteristics"**.

FIGURE 26-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

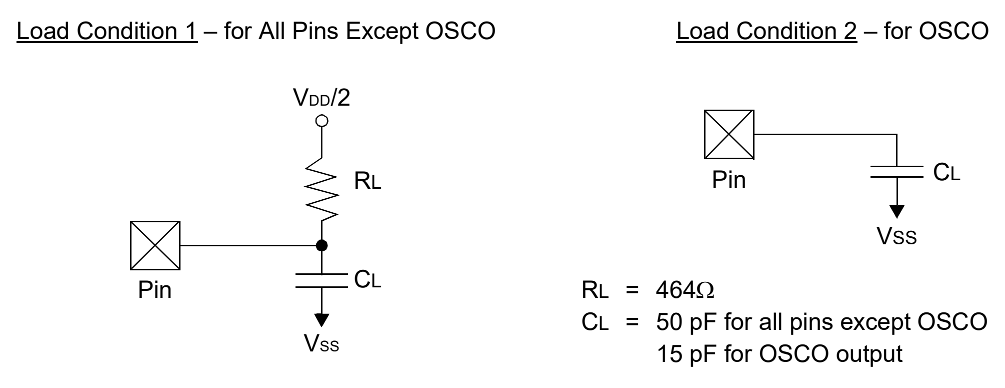


TABLE 26-17: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
DO50	COSC2	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when external clock is used to drive OSCI
DO56	CIO	All I/O Pins and OSCO	—	—	50	pF	EC mode
DO58	CB	SCLx, SDAx	—	—	400	pF	In I^{2}C mode

Note 1: Data in "Typ" column are at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 26-4: EXTERNAL CLOCK TIMING

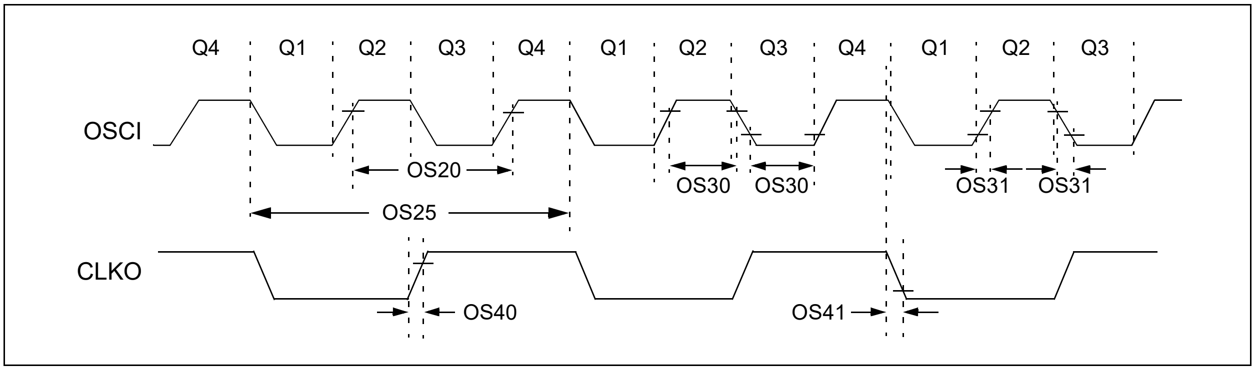


TABLE 26-18: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**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	TOSC = 1/FOSC	—	—	—	—	See Parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time(2)	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time(3)	—	6	10	ns	
OS41	TckF	CLKO Fall Time(3)	—	6	10	ns	

Note 1: Data in “Typ” column are at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Instruction cycle period (TCY) 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 TCY) and high for the Q3-Q4 period (1/2 TCY).

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[1]**	**Min**	**Typ[2]**	**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.

2: Data in “Typ” column are 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[1]	-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[2]	-15	—	+15	%	-40°C ≤ TA ≤ +85°C	1.8V ≤ VDD ≤ 3.6V
		-25	—	+25	%	-40°C ≤ TA ≤ +125°C	1.8V ≤ VDD ≤ 3.6V

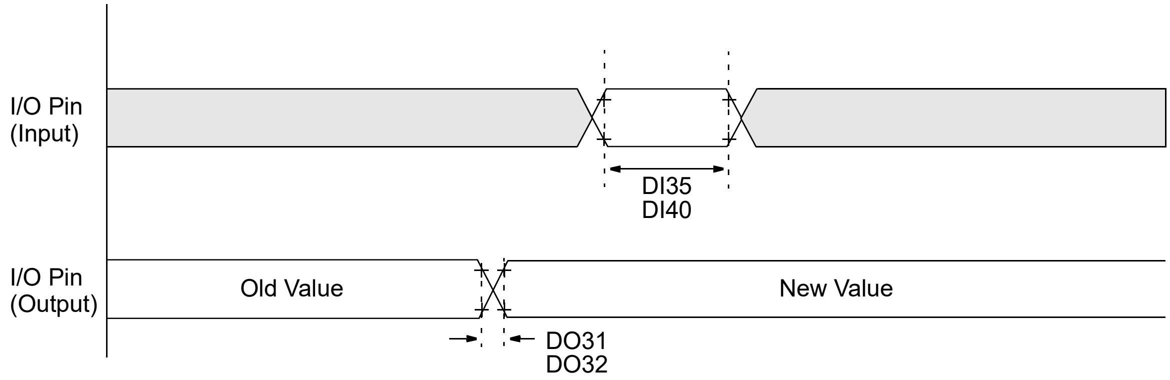
Note 1: The frequency is calibrated at +25°C and 3.3V. The OSCTUN bits can be used to compensate for temperature drift.

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	

FIGURE 26-5: CLKO AND I/O TIMING CHARACTERISTICS



Note: Refer to Figure 26-3 for load conditions.

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[1]**	**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 are at 3.3V, +25°C unless otherwise stated.

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(1)**	**Max.**	**Units**	**Conditions**
SY10	TmcL	$\overline{\text{MCLR}}$ 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 $\overline{\text{MCLR}}$ 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 are 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(1,2)	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid(2)	—	—	10	μs	

Note 1: Response time is measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from VSS to VDD.

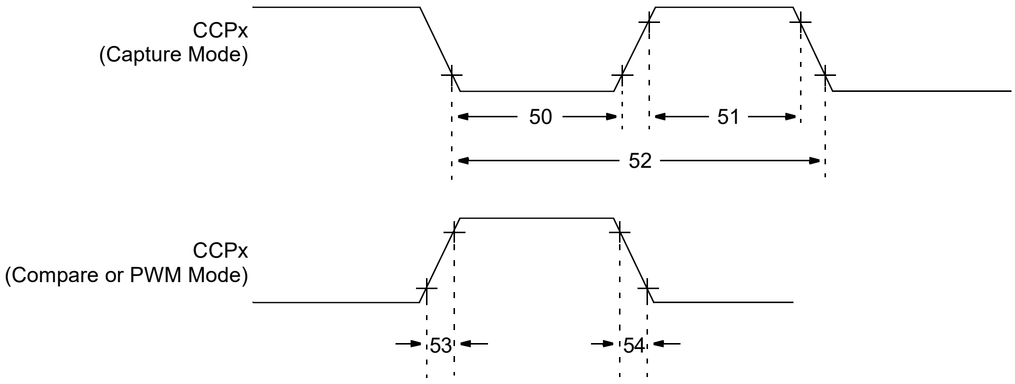
2: Parameters are characterized but not tested.

TABLE 26-25: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS

Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
VR310	TSET	Settling Time(1)	—	—	10	μs	

Note 1: Settling time is measured while CVRSS = `1` and the CVR[3:0] bits transition from ‘`0000`’ to ‘`1111`’.

FIGURE 26-6: CAPTURE/COMPARE/PWM TIMINGS (ECCP1, ECCP2 MODULES)

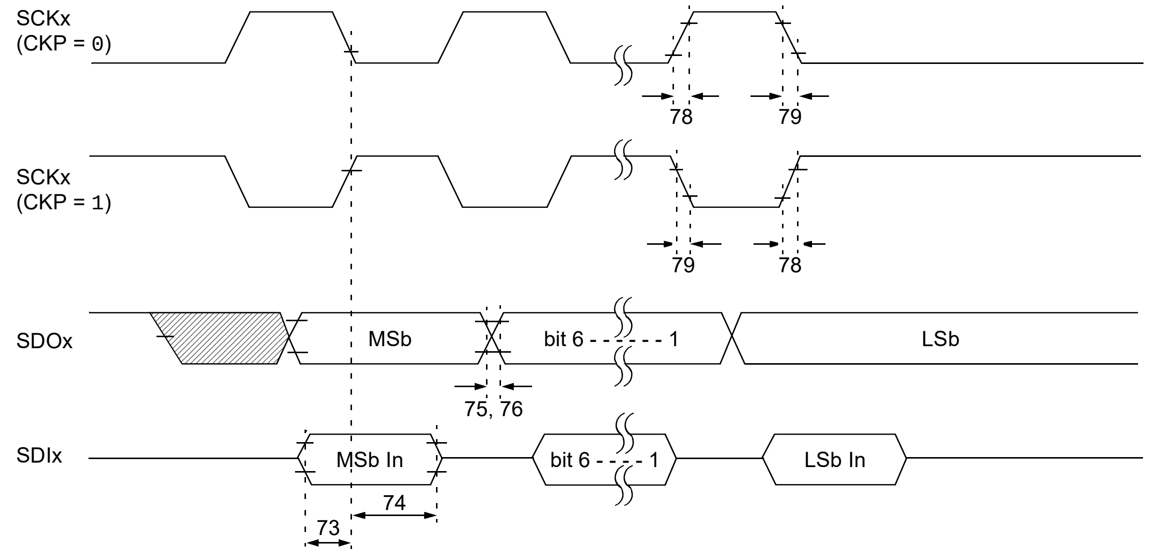


Note: Refer to Figure 26-3 for load conditions.

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 TCY + 20	—	ns	
			With Prescaler	20	—	ns	
51	TCCH	CCPx Input High Time	No Prescaler	0.5 TCY + 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 Fall Time		—	25	ns	

FIGURE 26-7: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)

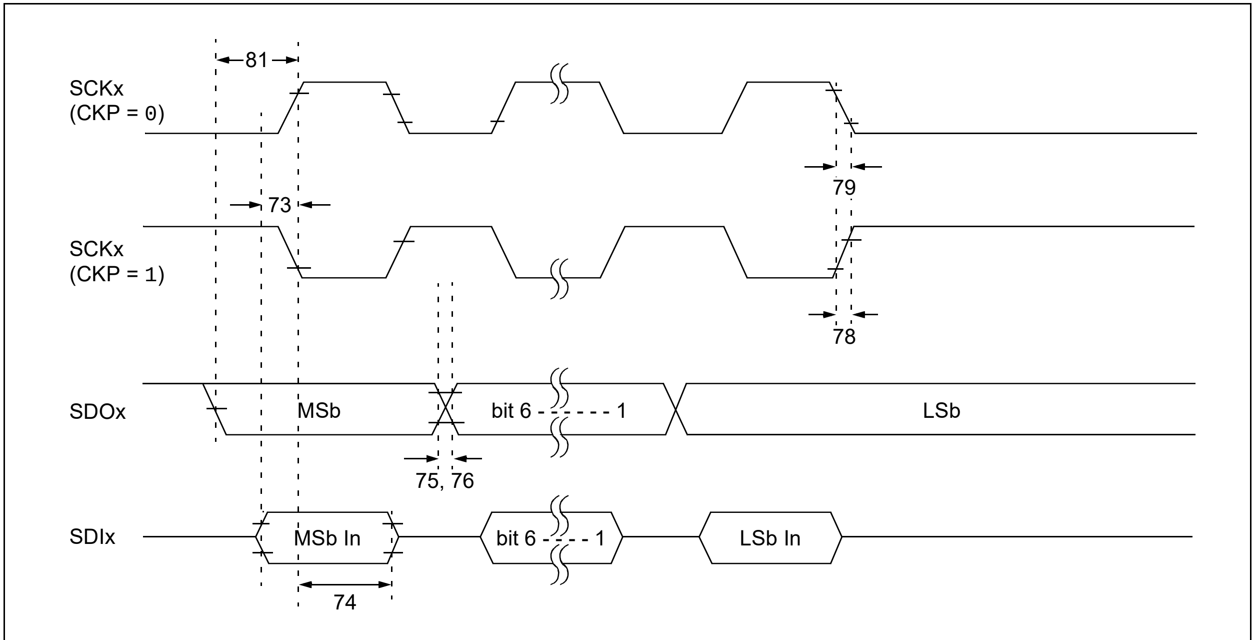


Note: Refer to Figure 26-3 for load conditions.

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	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	
	FSCK	SCKx Frequency	—	10	MHz	

FIGURE 26-8: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)

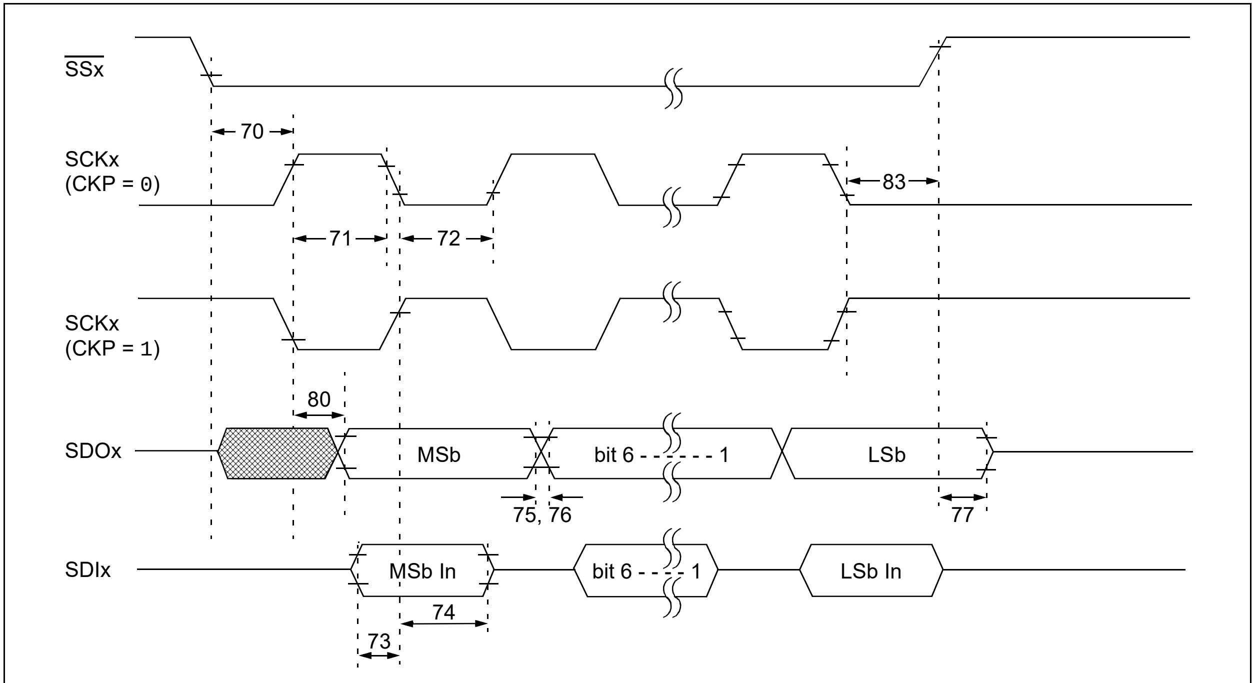


Note: Refer to Figure 26-3 for load conditions.

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	

FIGURE 26-9: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)



Note: Refer to Figure 26-3 for load conditions.

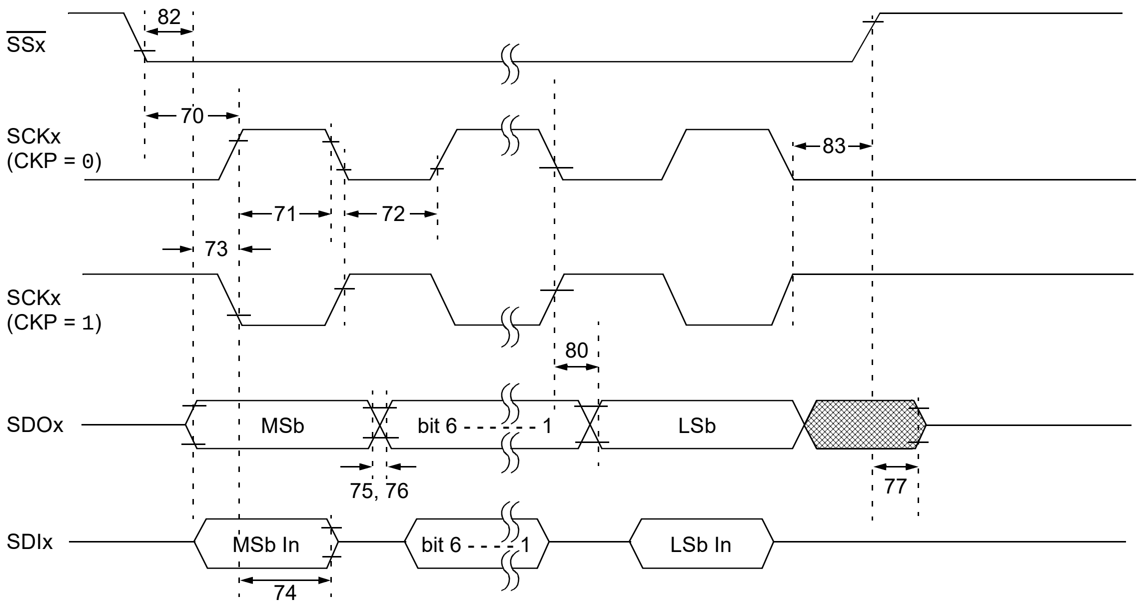
TABLE 26-29: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{\text{SSx}}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 TCY	—	ns	
70A	TssL2WB	$\overline{\text{SSx}}$ 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	$\overline{\text{SSx}}$ ↑ to SDOx Output High-Impedance		10	50	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{\text{SSx}}$ ↑ 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.

FIGURE 26-10: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)



Note: Refer to Figure 26-3 for load conditions.

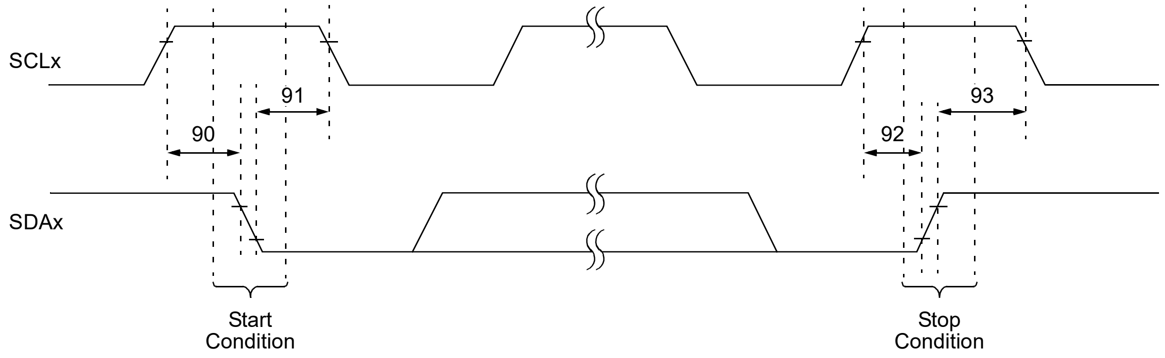
TABLE 26-30: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TSSL2SCH, TSSL2SCL	$\overline{\text{SSx}}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 TCY	—	ns	
70A	TSSL2WB	$\overline{\text{SSx}}$ 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**
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	$\overline{\text{SSx}}$ ↑ 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 $\overline{\text{SSx}}$ ↓ Edge		—	50	ns	
83	TSCH2SSH, TSCL2SSH	$\overline{\text{SSx}}$ ↑ 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.

FIGURE 26-11: I^{2}C BUS START/STOP BITS TIMING

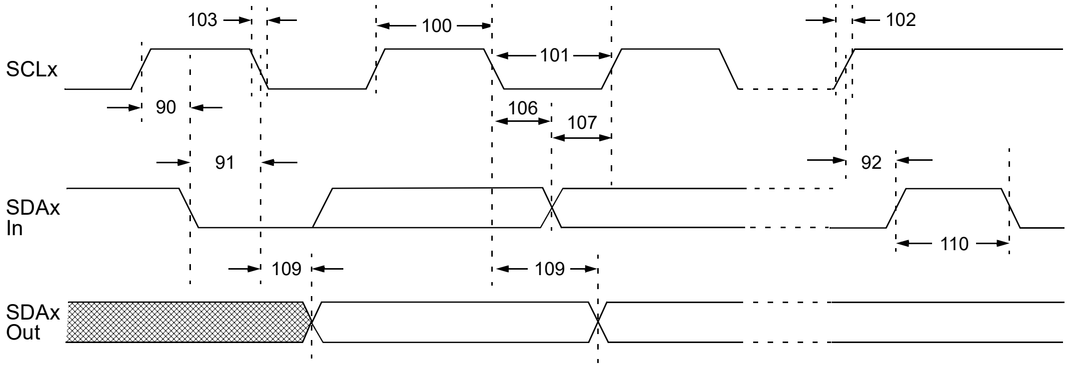


Note: Refer to Figure 26-3 for load conditions.

TABLE 26-31: I^{2}C BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
		Setup Time	400 kHz mode	600	—		
91	THD:STA	Start Condition	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	600	—		
92	TSU:STO	Stop Condition	100 kHz mode	4700	—	ns	
		Setup Time	400 kHz mode	600	—		
93	THD:STO	Stop Condition	100 kHz mode	4000	—	ns	
		Hold Time	400 kHz mode	600	—		

FIGURE 26-12: I^{2}C BUS DATA TIMING



Note: Refer to Figure 26-3 for load conditions.

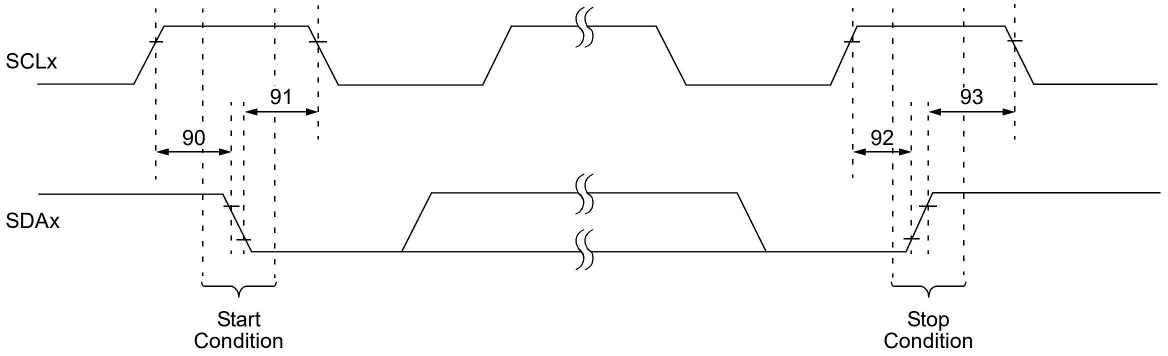
TABLE 26-32: I^{2}C BUS DATA REQUIREMENTS (SLAVE MODE)

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	—	TCY	
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	—	TCY	
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB 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 CB	300	ns	CB 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	**Note 2**
			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	**Note 1**
			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^{2}C bus device can be used in a Standard mode I^{2}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^{2}C bus specification), before the SCLx line is released.

FIGURE 26-13: MSSPx I²C BUS START/STOP BITS TIMING WAVEFORMS

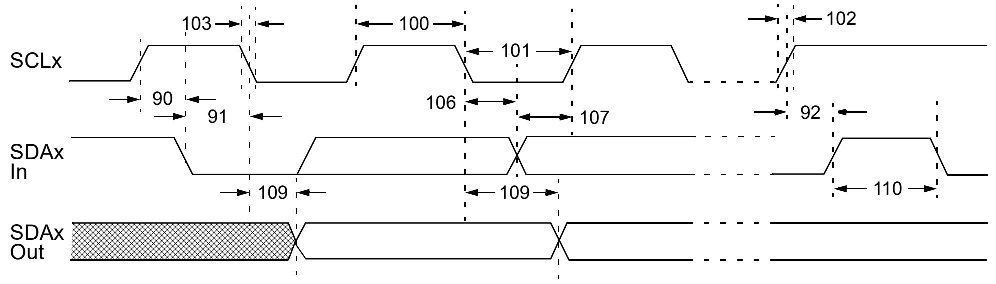


Note: Refer to Figure 26-3 for load conditions.

TABLE 26-33: I²C BUS START/STOP BITS REQUIREMENTS (MASTER MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	Only relevant for Repeated Start condition
		Setup Time	400 kHz mode	2(TOSC)(BRG + 1)	—		
91	THD:STA	Start Condition	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
		Hold Time	400 kHz mode	2(TOSC)(BRG + 1)	—		
92	TSU:STO	Stop Condition	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
		Setup Time	400 kHz mode	2(TOSC)(BRG + 1)	—		
93	THD:STO	Stop Condition	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
		Hold Time	400 kHz mode	2(TOSC)(BRG + 1)	—		

FIGURE 26-14: MSSPx I^{2}C BUS DATA TIMING



Note: Refer to Figure 26-3 for load conditions.

TABLE 26-34: I^{2}C BUS DATA REQUIREMENTS (MASTER MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
101	TLOW	Clock Low Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	
			400 kHz mode	2(TOSC)(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 CB	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 CB	300	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—	—	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(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	**Note 1**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	—	
			400 kHz mode	2(TOSC)(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^{2}C bus device can be used in a Standard mode I^{2}C bus system, but Parameter 107 ≥ 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.

TABLE 26-35: A/D MODULE SPECIFICATIONS

AC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Device Supply							
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	
Reference Inputs							
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	
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	**Note 1**
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	Ω	10-bit
A/D Accuracy							
AD20b	NR	Resolution	—	10	—	bits	
AD21b	INL	Integral Nonlinearity	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22b	DNL	Differential Nonlinearity	—	±1	±1.5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23b	GERR	Gain Error	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24b	EOFF	Offset Error	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25b		Monotonicity	—	—	—	—	**Note 2**

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.

TABLE 26-36: A/D CONVERSION TIMING REQUIREMENTS[1]

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Clock Parameters							
AD50	TAD	A/D Clock Period	75	—	—	ns	TCY = 75 ns, AD1CON3 is in default state
AD51	TRC	A/D Internal RC Oscillator Period	—	250	—	ns	
Conversion Rate							
AD55	TCONV	Conversion Time	—	12	—	TAD	
AD56	FCNV	Throughput Rate	—	—	500	ksps	AVDD ≥ 2.7V
AD57	TSAMP	Sample Time	—	1	—	TAD	
AD58	TACQ	Acquisition Time	750	—	—	ns	Note 2
AD59	TSWC	Switching Time from Convert to Sample	—	—	Note 3	—	
AD60	TDIS	Discharge Time	0.5	—	—	TAD	
Clock Parameters							
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.

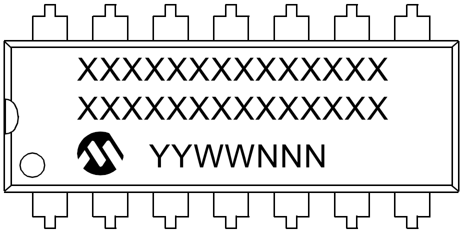
2: The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD).

3: On the following cycle of the device clock.

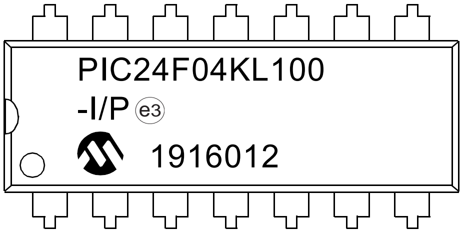
27.0 PACKAGING INFORMATION

27.1 Package Marking Information

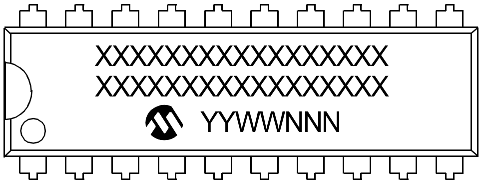
14-Lead PDIP (300 mil)



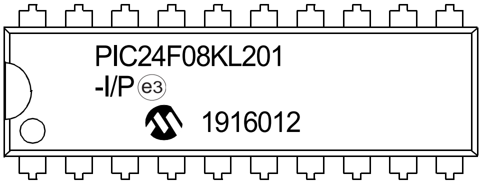
Example



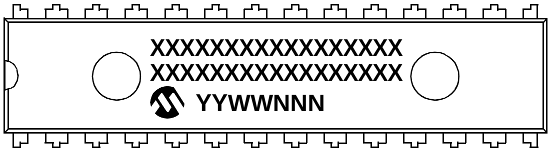
20-Lead PDIP (300 mil)



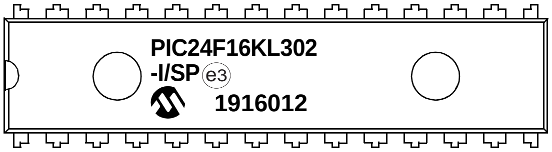
Example



28-Lead SPDIP (.300 in)



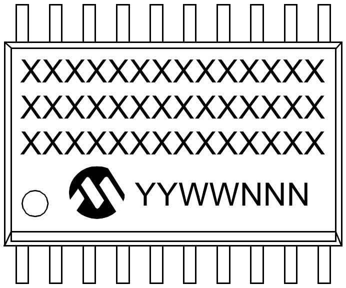
Example



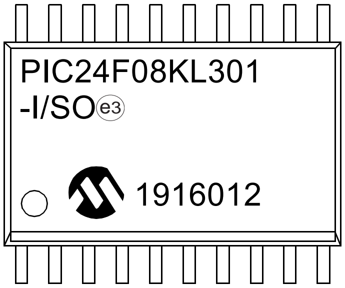
Legend:	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	e3	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free 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.

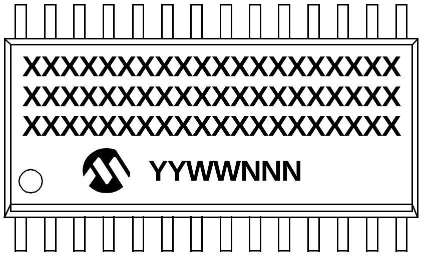
20-Lead SOIC (7.50 mm)



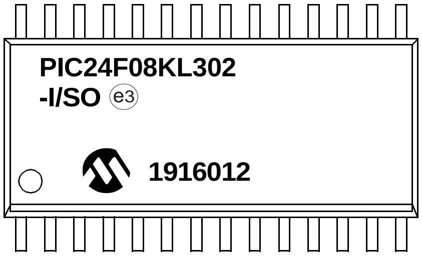
Example



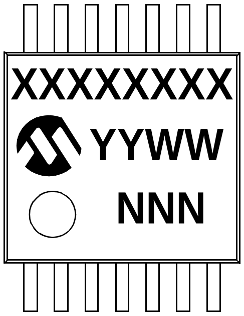
28-Lead SOIC (7.50 mm)



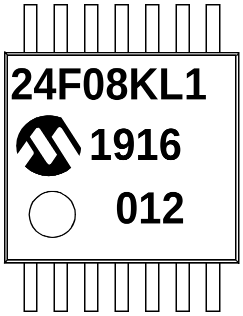
Example



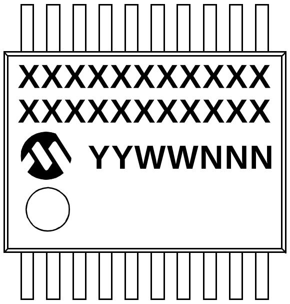
14-Lead TSSOP (4.4 mm)



Example

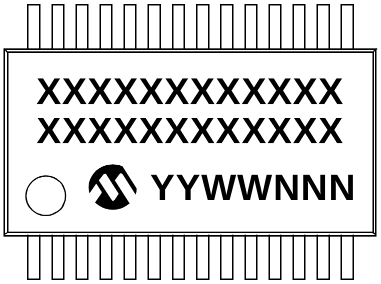


20-Lead SSOP (5.30 mm)



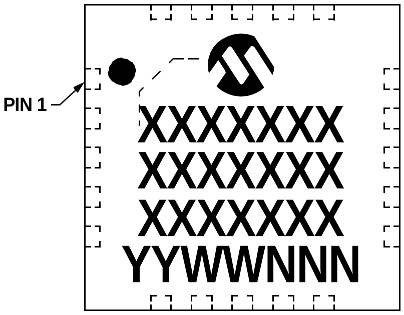
Example

28-Lead SSOP (5.30 mm)

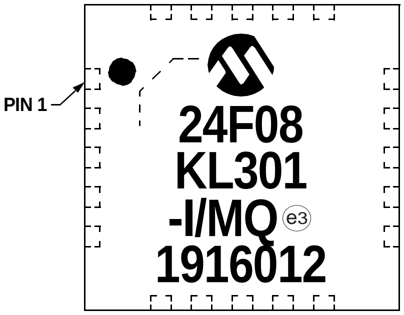


Example

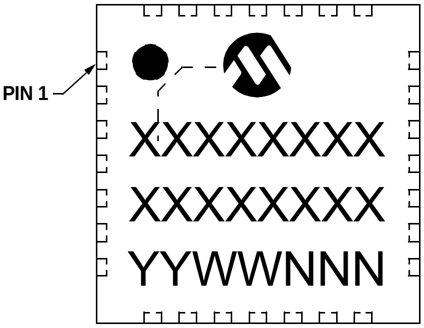
20-Lead VQFN (5x5x0.9 mm)



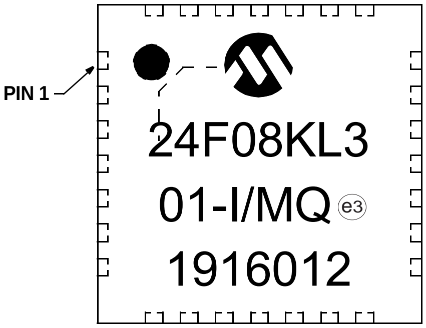
Example



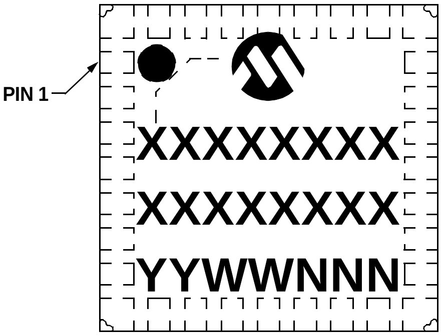
28-Lead QFN (5x5x0.9 mm)



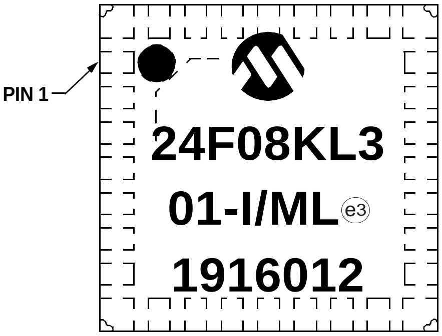
Example



28-Lead QFN (6x6 mm)



Example

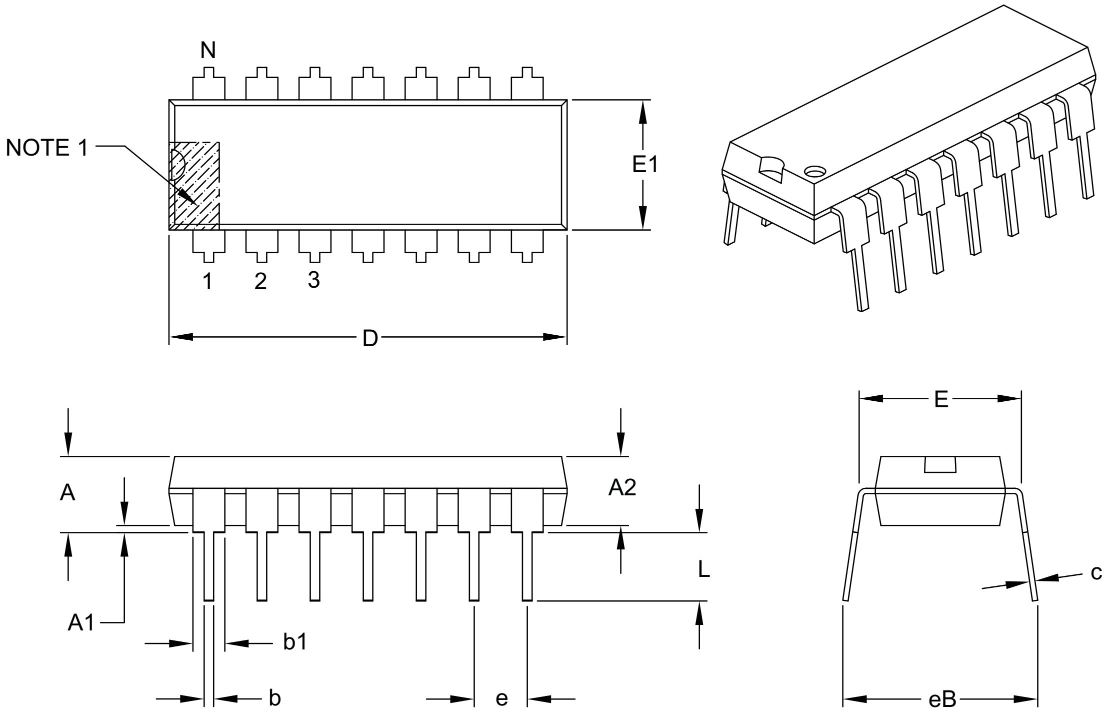


27.2 Package Details

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



Units		INCHES		
Dimension Limits		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:

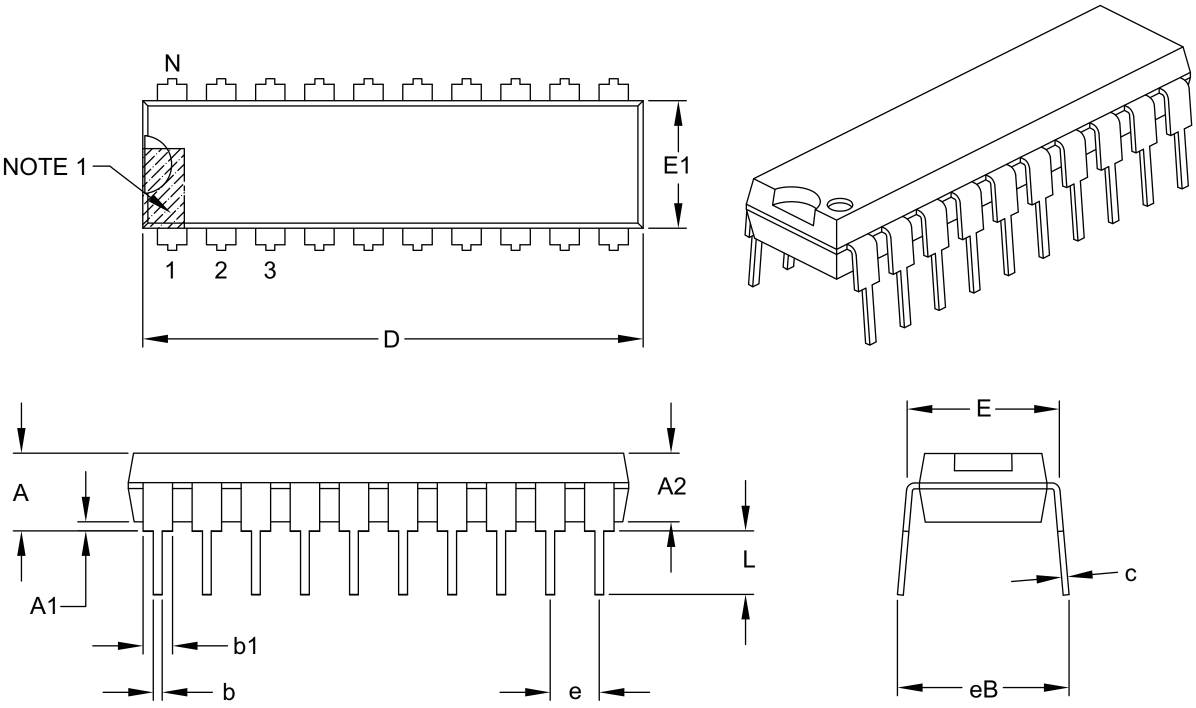
1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-005B

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



Units		INCHES		
Dimension Limits		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:

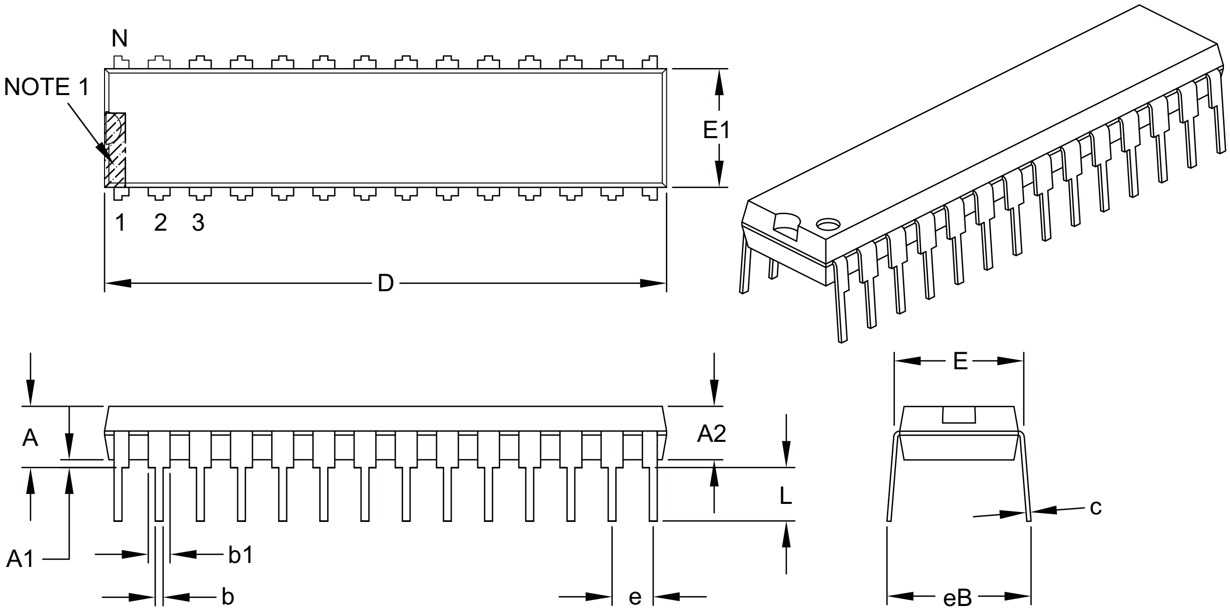
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-019B

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



Units		INCHES		
Dimension Limits		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:

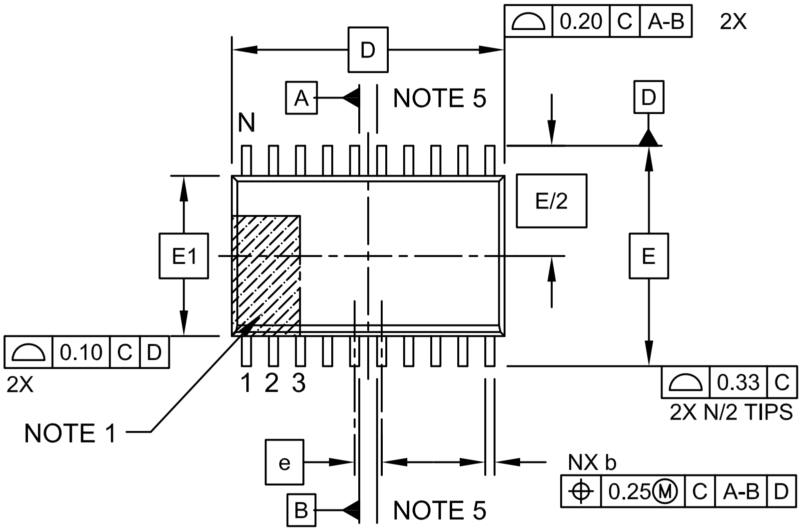
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

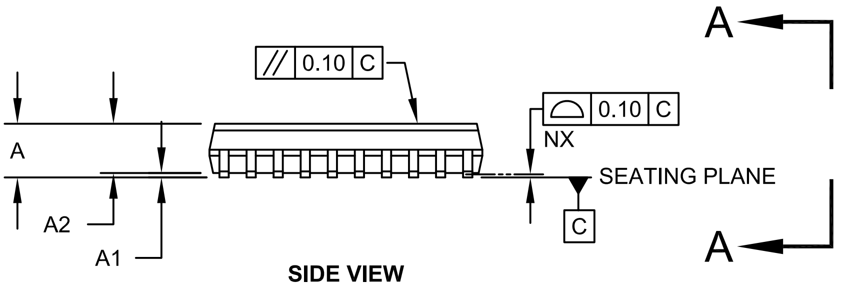
Microchip Technology Drawing C04-070B

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

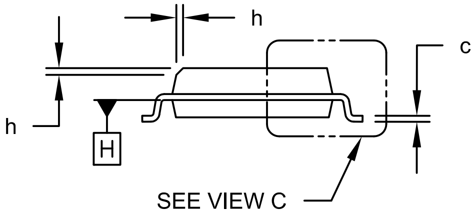
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

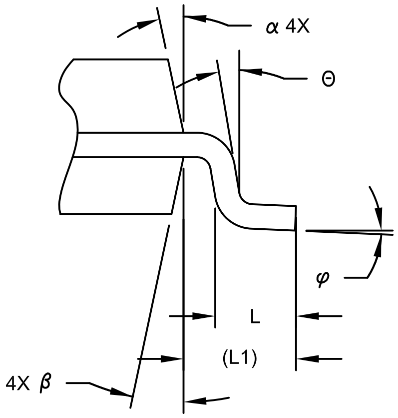


VIEW A-A

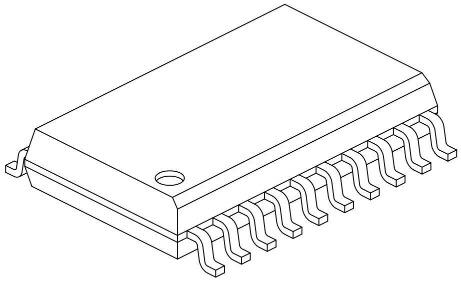
Microchip Technology Drawing C04-094C Sheet 1 of 2

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		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:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. 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.
4. Dimensioning and tolerancing per ASME Y14.5M

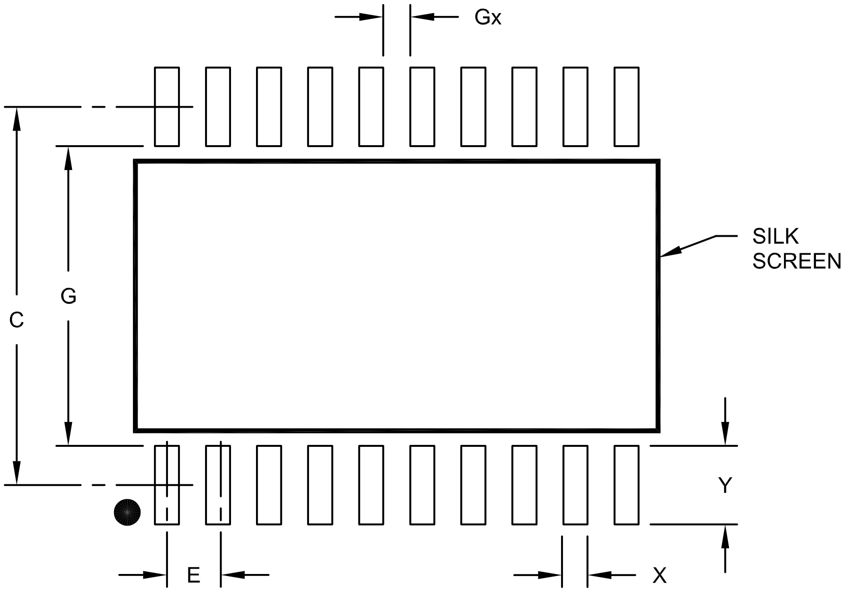
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

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

Units		MILLIMETERS		
Dimension Limits		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:

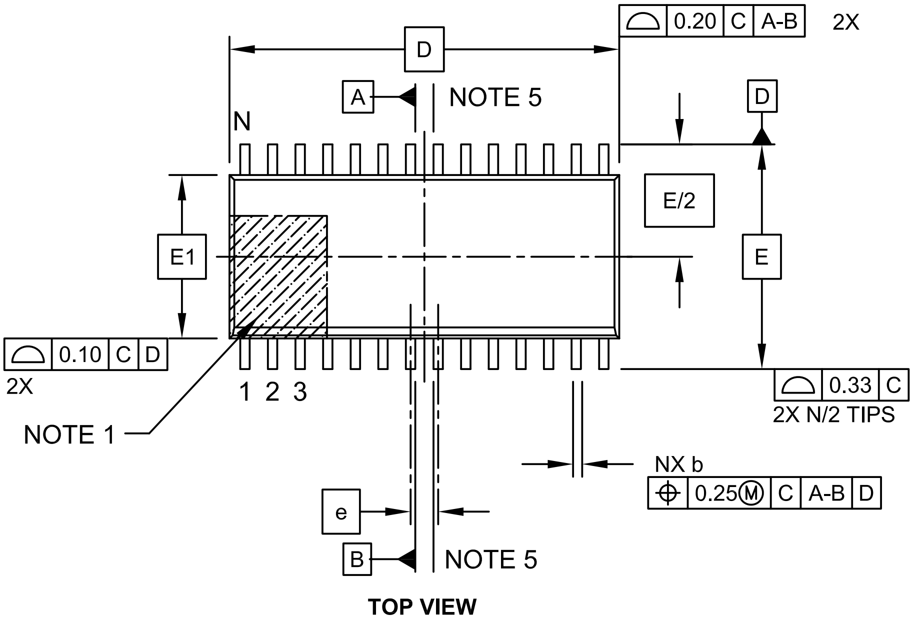
1. Dimensioning and tolerancing per ASME Y14.5M

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

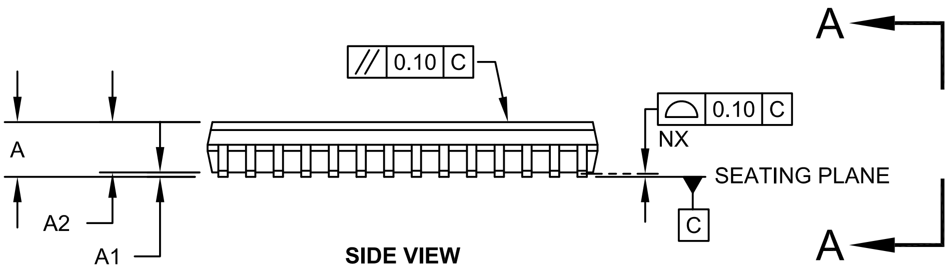
Microchip Technology Drawing No. C04-2094A

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

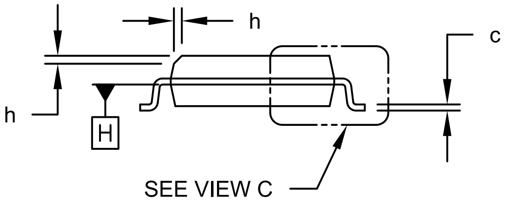
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

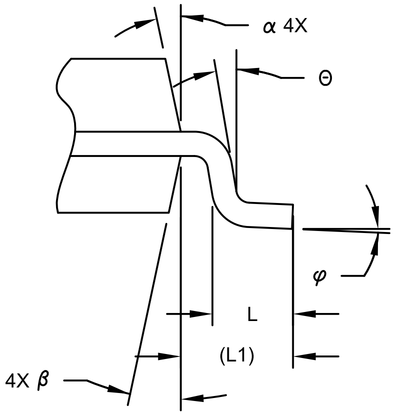


VIEW A-A

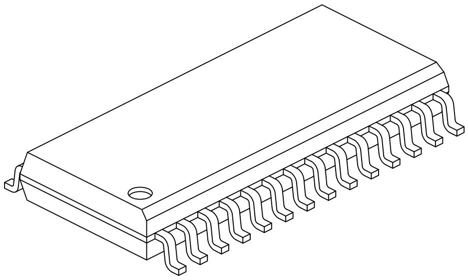
Microchip Technology Drawing C04-052C Sheet 1 of 2

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



VIEW C



Units		MILLIMETERS		
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°

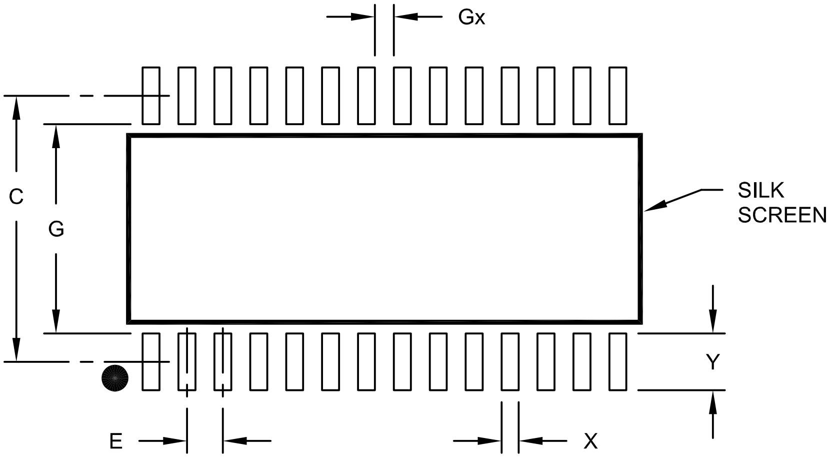
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. 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.
4. 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.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

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

	Units	MILLIMETERS		
	Dimension Limits	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:

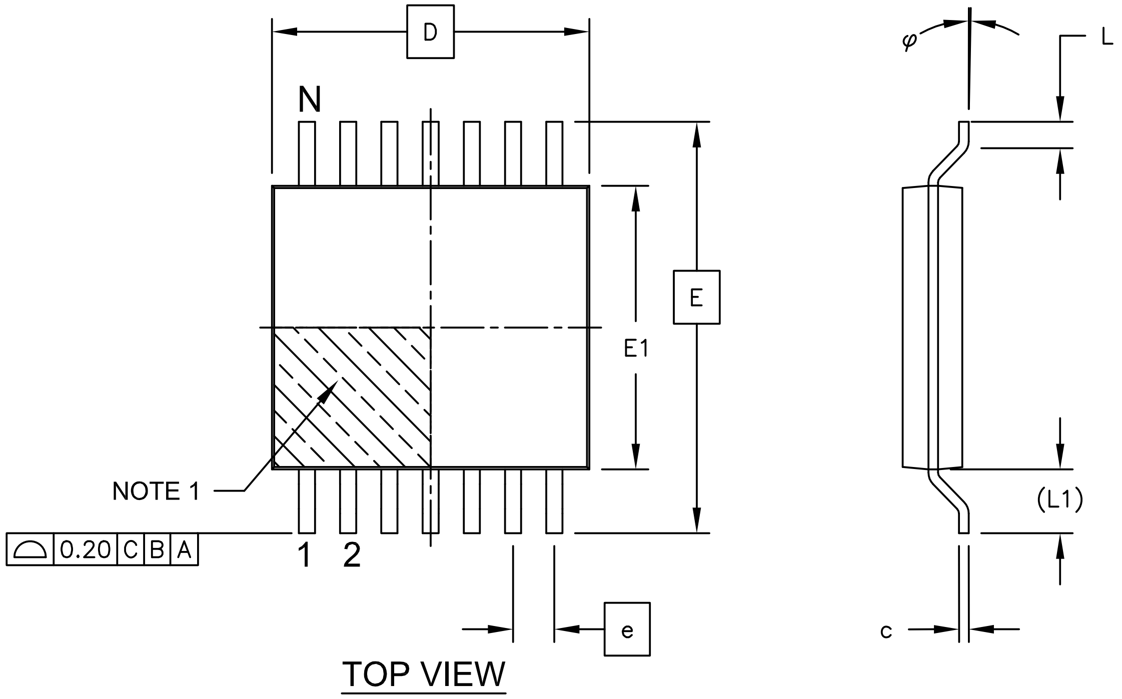
1. Dimensioning and tolerancing per ASME Y14.5M

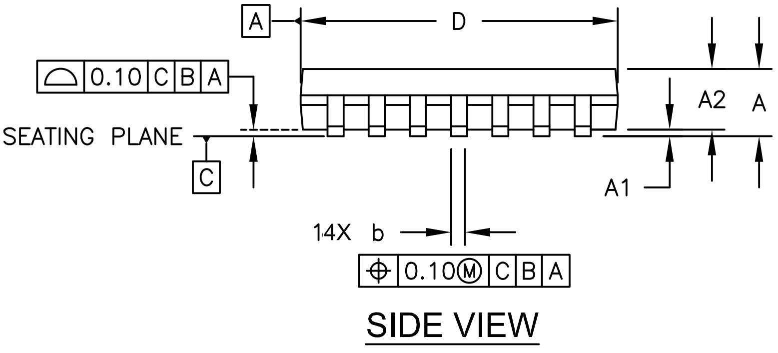
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

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

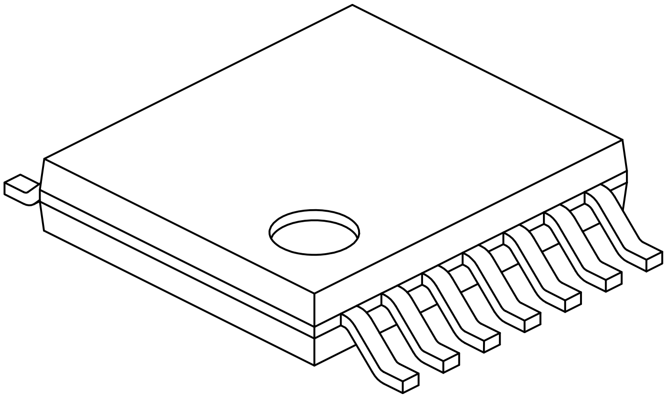




Microchip Technology Drawing C04-087C Sheet 1 of 2

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



Units		MILLIMETERS		
Dimension Limits		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	φ	0°	-	8°
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.19	-	0.30

Notes:

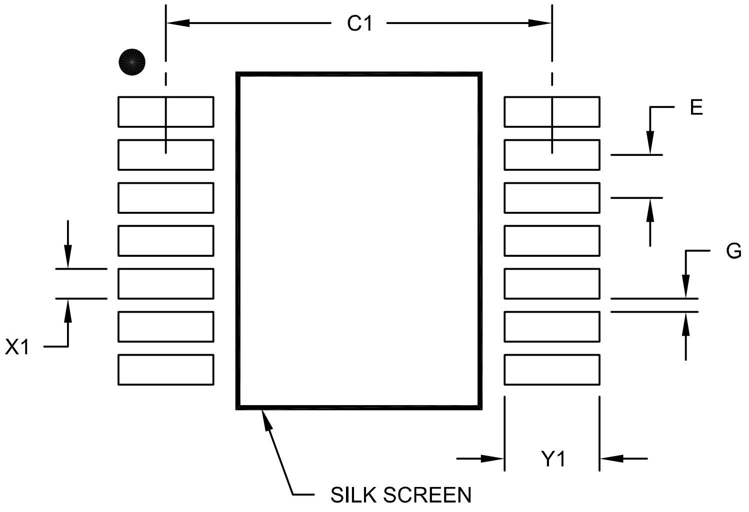
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

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

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

Units		MILLIMETERS		
Dimension Limits		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:

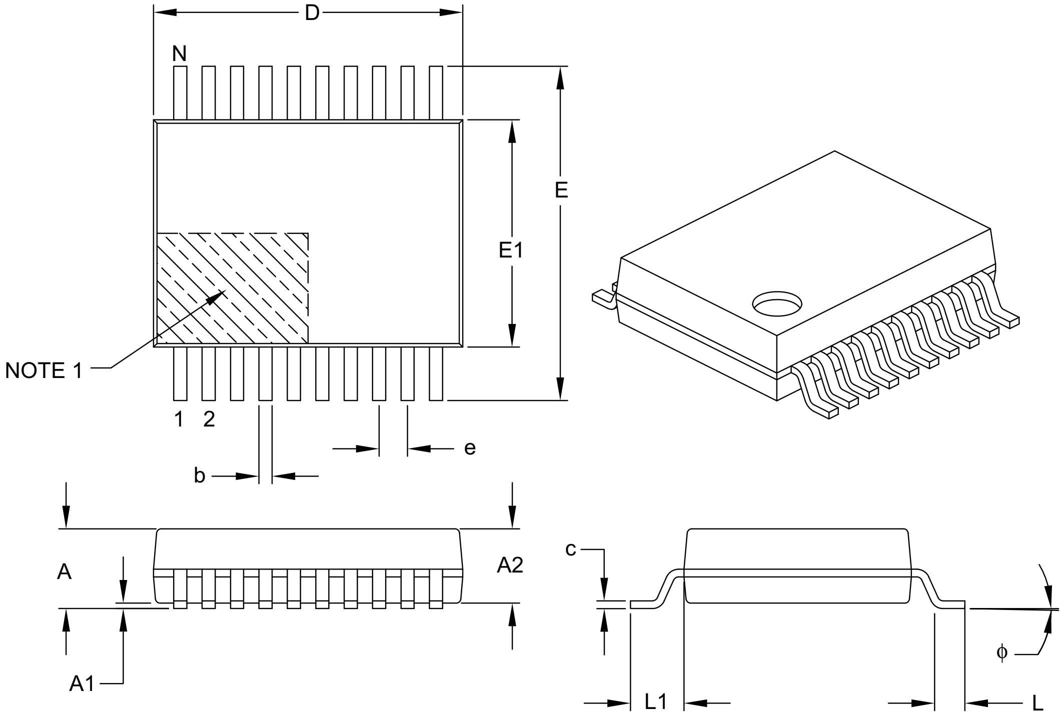
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2087A

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	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

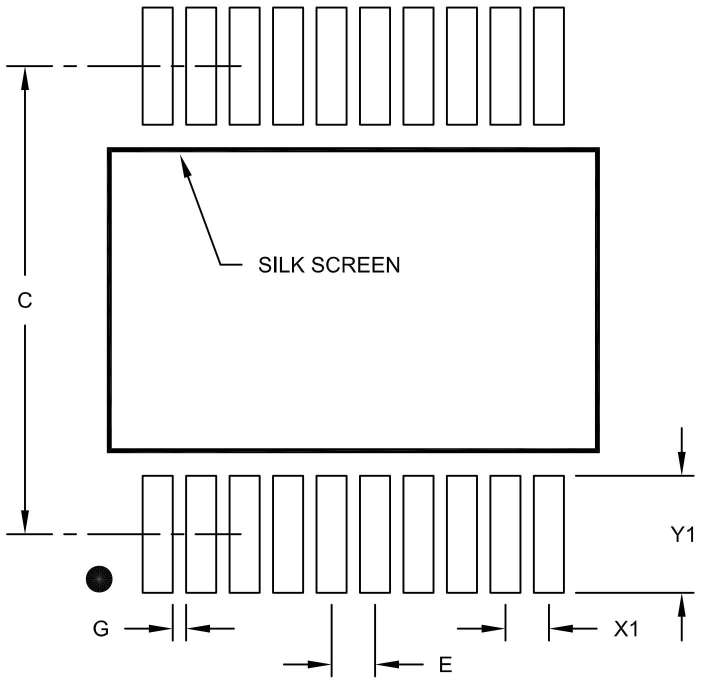
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		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:

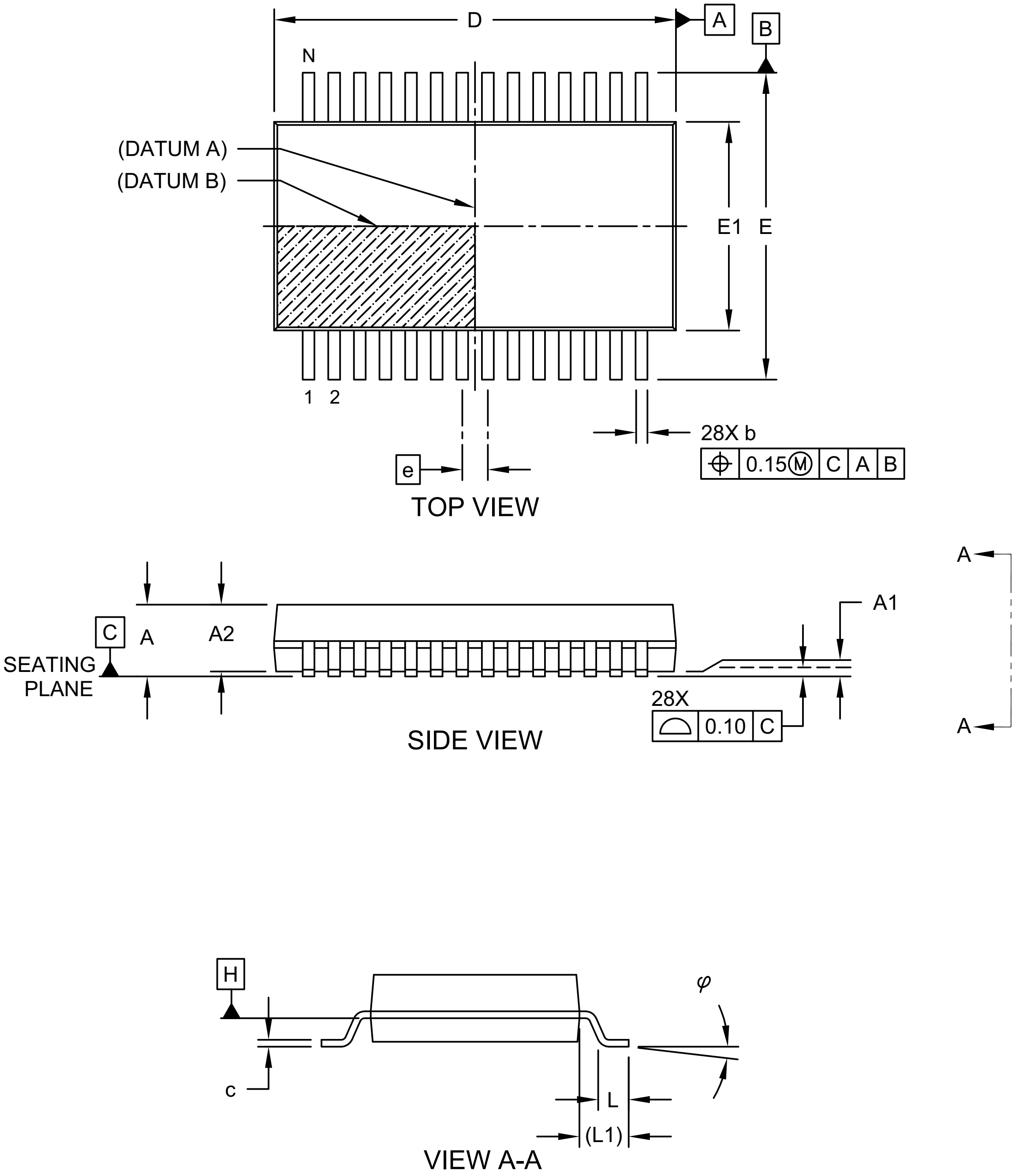
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2072A

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

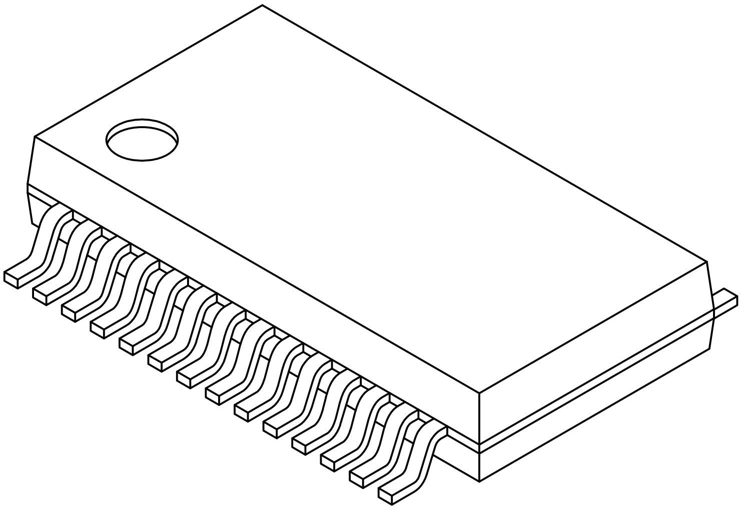
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-073 Rev C Sheet 1 of 2

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



Units		MILLIMETERS		
Dimension Limits		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	φ	0°	4°	8°
Lead Width	b	0.22	-	0.38

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 0.20mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M

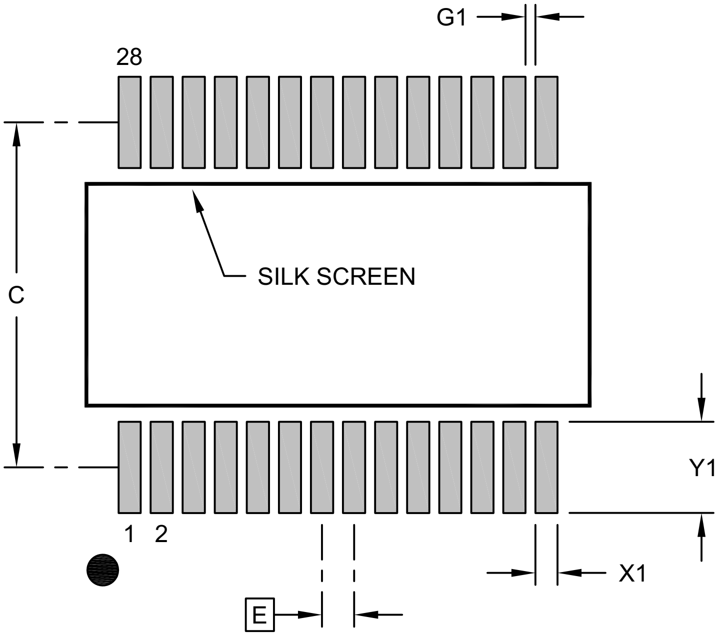
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073 Rev C Sheet 2 of 2

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.00	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.85
Contact Pad to Center Pad (X26)	G1	0.20		

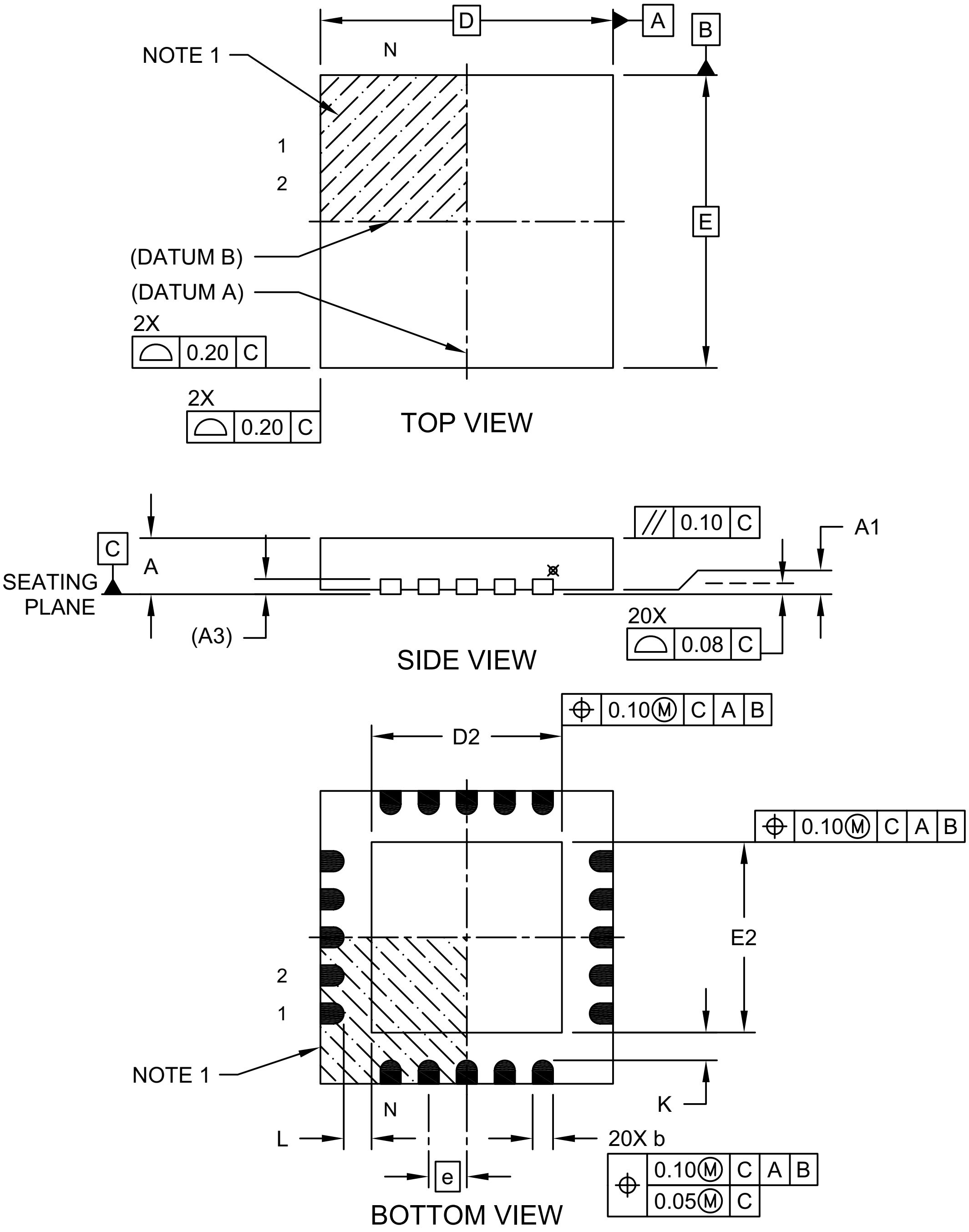
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2073 Rev B

20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x1.0 mm Body [VQFN] With 0.40 mm Contact Length

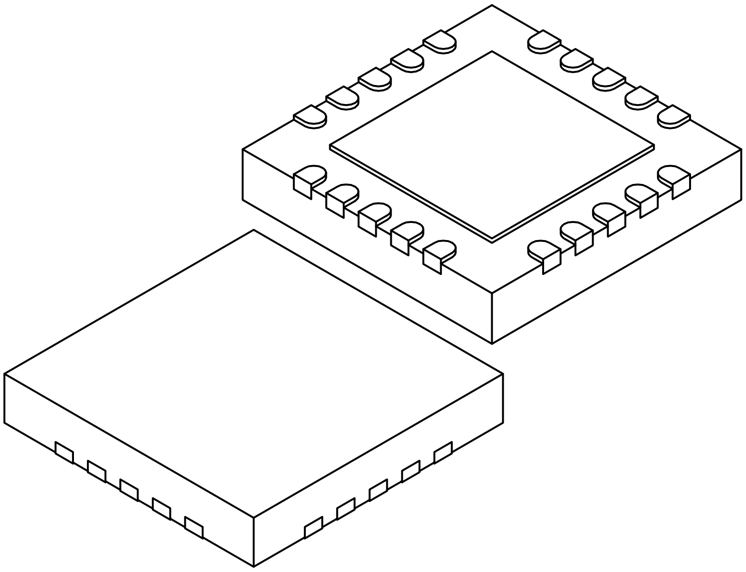
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-139C (MQ) Sheet 1 of 2

20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x1.0 mm Body [VQFN] With 0.40 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 Terminals	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 Length	D	5.00 BSC		
Exposed Pad Length	D2	3.15	3.25	3.35
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	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:

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

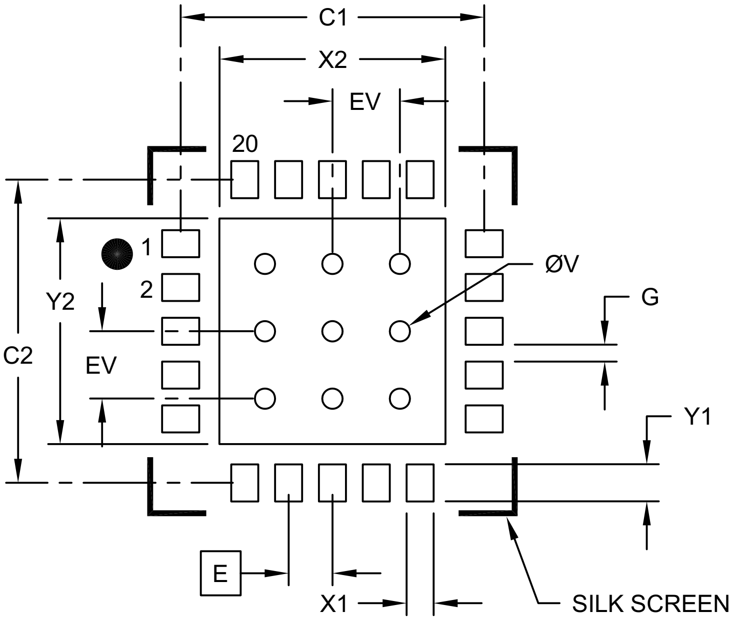
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-139C (MQ) Sheet 2 of 2

20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x1.0 mm Body [VQFN] With 0.40 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

Units		MILLIMETERS		
Dimension Limits		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		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

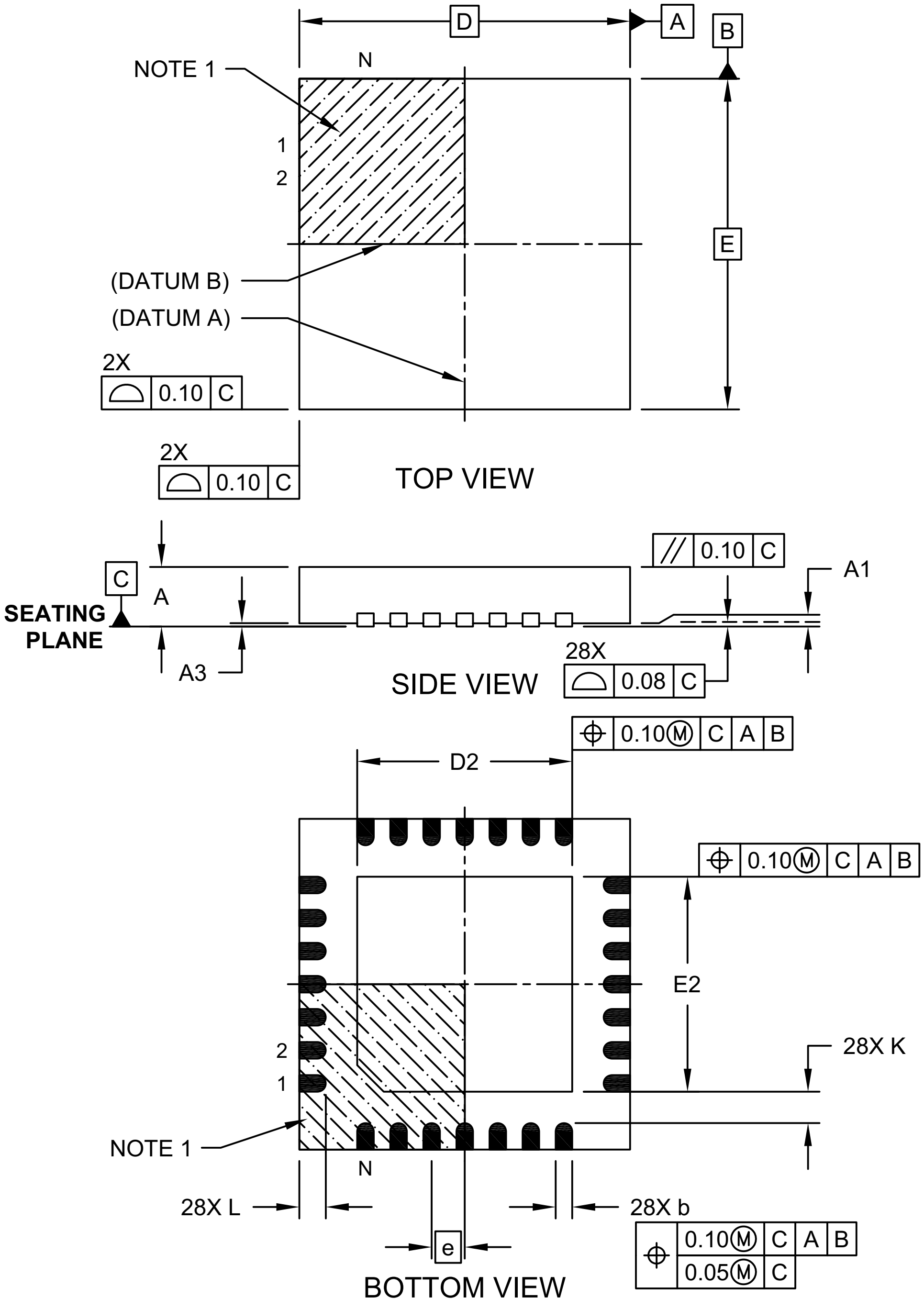
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2139B (MQ)

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN or VQFN]

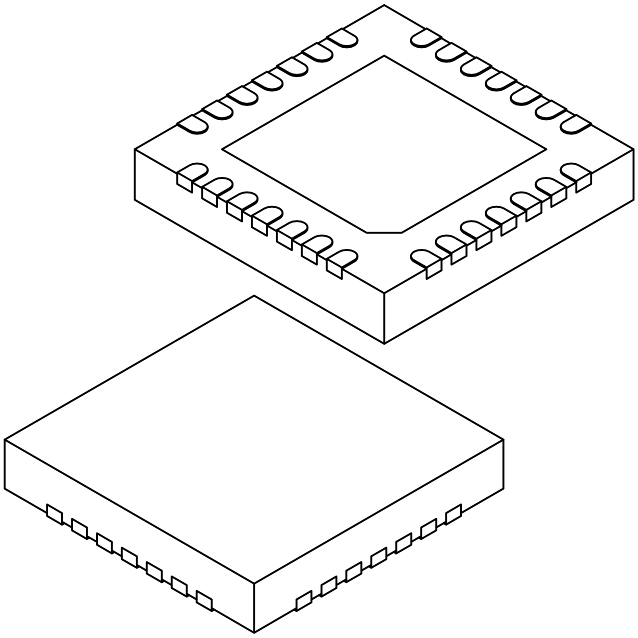
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-140C Sheet 1 of 2

28-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN or VQFN]

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		0.50 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	-	-

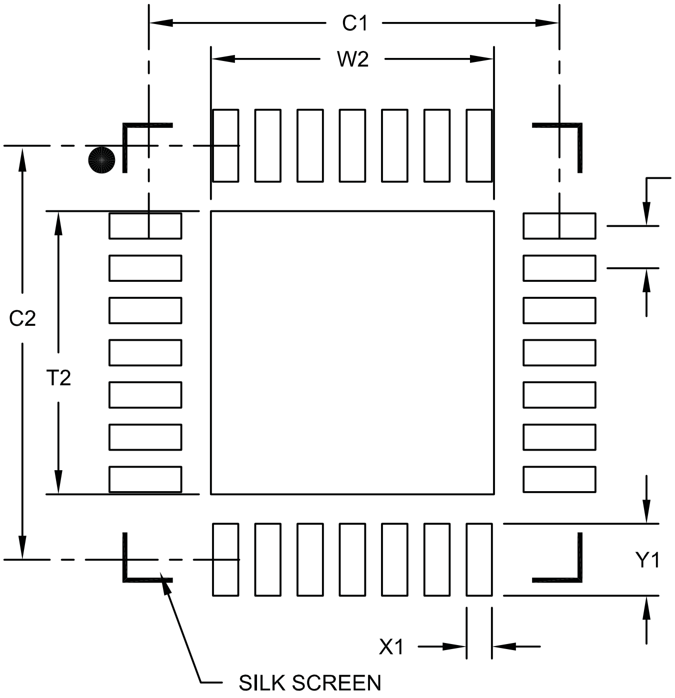
Notes:

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 C04-140C Sheet 2 of 2

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

Units		MILLIMETERS		
Dimension Limits		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:

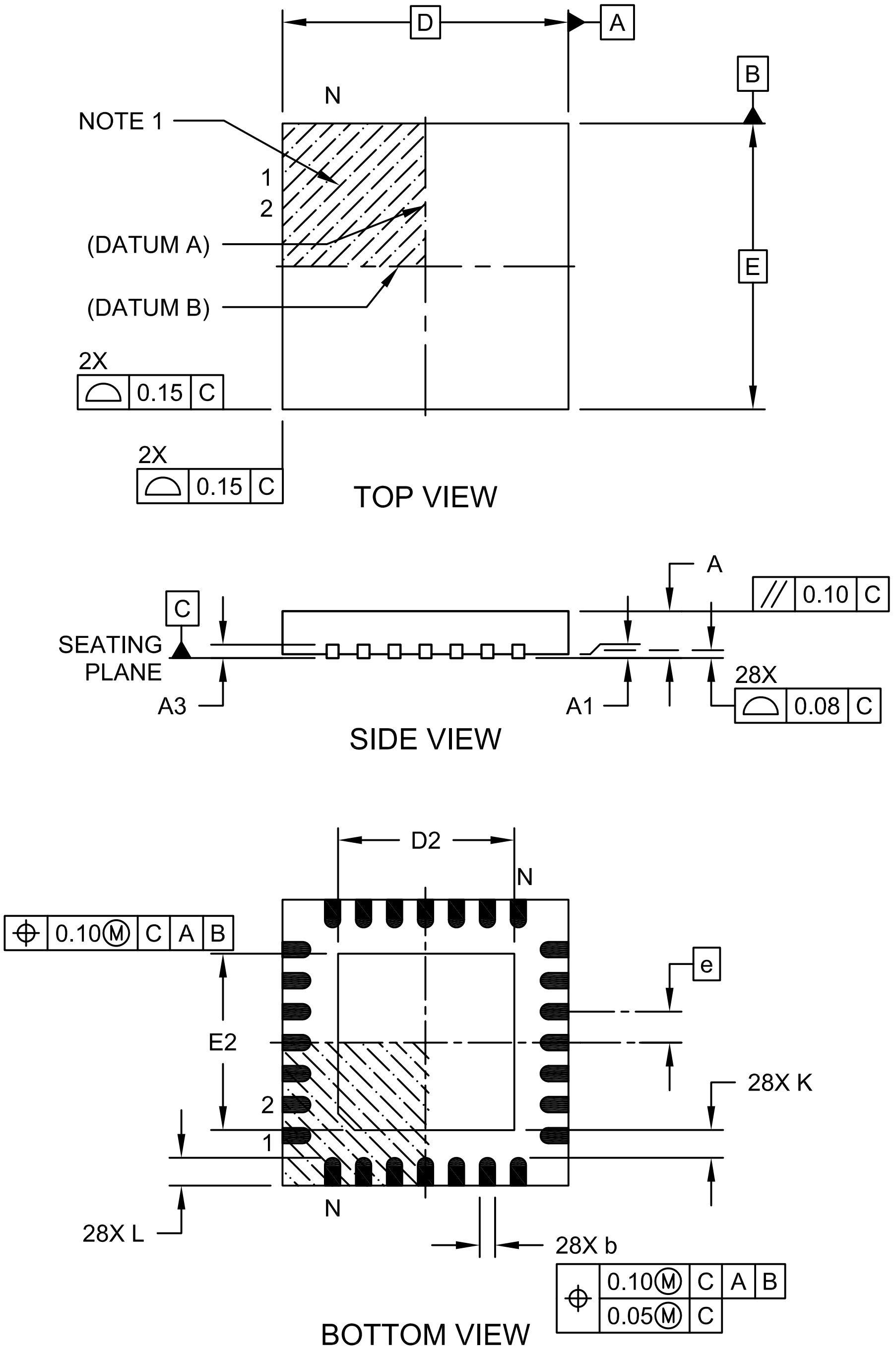
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2140A

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

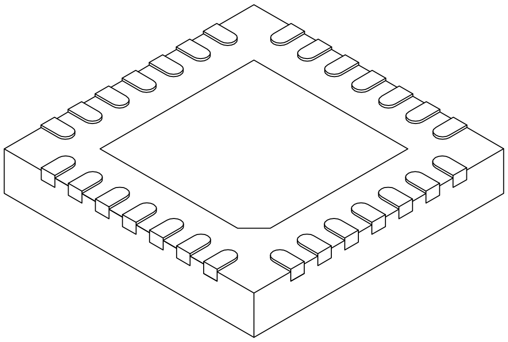
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-105C Sheet 1 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal 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
Terminal 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
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

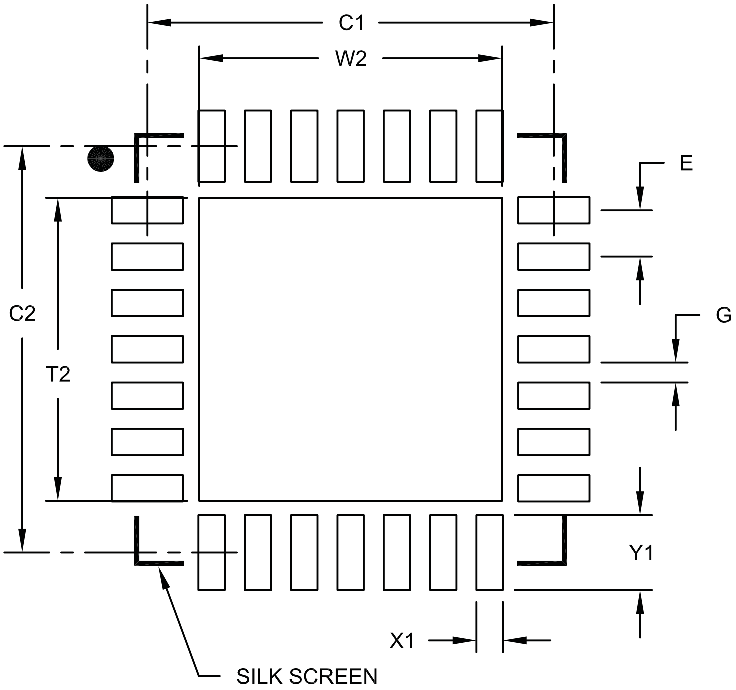
Notes:

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 C04-105C Sheet 2 of 2

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		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

NOTES:

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 FSCK 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.

Revision D (December 2019)

Updates the PIC24FXXKL301/401 pin diagram.

Adds **Section 7.4.1 “Low-Power BOR (LPBOR)”**, adds reference manual information to **Section 16.0 “Capture/Compare/PWM (CCP) and Enhanced CCP Modules”**, adds reference manual information to **Section 17.0 “Master Synchronous Serial Port (MSSP)”**, adds reference manual information to **Section 20.0 “Comparator Module”** and adds a note to **Section 23.1 “Code-Protect Security Options”**.

Updates **Section 11.2.1 “Analog Selection Register”** and **Appendix B: “Migrating from PIC18/PIC24 to PIC24F16KL402”**.

Updates Table 4-11 and Table 8-2.

Updates Register 6-1, Register 14-1, Register 19-3 and Register 23-6.

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.

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: PIC24F16KL402 PERIPHERAL MODULE ORIGINATING ARCHITECTURE

Peripheral Module	PIC18	PIC24
ECCP/CCP	X	—
MSSP (I^2C/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

NOTES:

INDEX

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website 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 website at 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

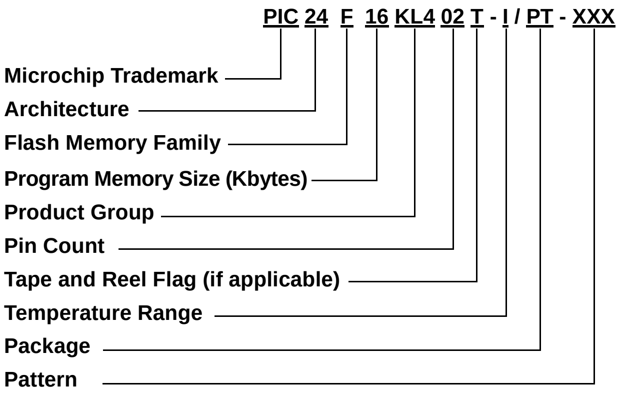
Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture	24	= 16-bit modified Harvard without DSP
Flash Memory Family	F	= Standard voltage range Flash program memory
Product Group	KL4 KL3 KL2 KL1	= General purpose microcontrollers
Pin Count	00 01 02	= 14-pin = 20-pin = 28-pin
Temperature Range	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package	SP SO SS ST ML MQ P	= SPDIP = SOIC = SSOP = TSSOP = QFN = VQFN = PDIP
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) ES = Engineering Sample	

Examples:

a) PIC24F16KL402-I/ML: General Purpose, 16-Kbyte Program Memory, 28-Pin, Industrial Temperature, QFN Package

b) PIC24F04KL101T-I/SS: General Purpose, 4-Kbyte Program Memory, 20-Pin, Industrial Temperature, SSOP Package, Tape-and-Reel

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 unless otherwise stated.

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA 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.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark 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.

© 2011-2019, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-5376-5

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

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

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
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820