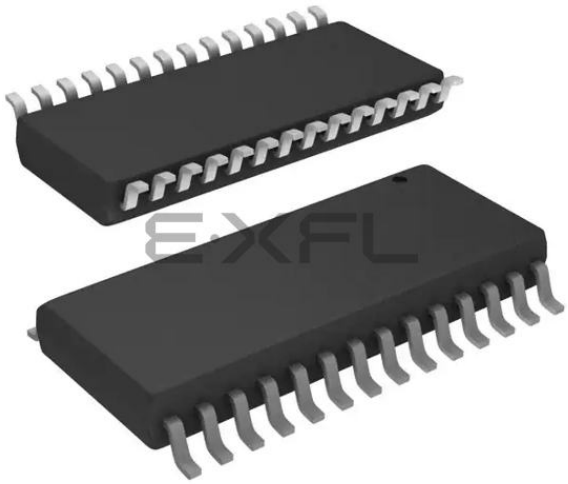


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, PMP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	21
Program Memory Size	64KB (22K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga002-e-so

1.0 DEVICE OVERVIEW

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

- PIC24FJ16GA002
- PIC24FJ32GA002
- PIC24FJ48GA002
- PIC24FJ64GA002
- PIC24FJ16GA004
- PIC24FJ32GA004
- PIC24FJ48GA004
- PIC24FJ64GA004

This family introduces a new line of Microchip devices: a 16-bit microcontroller family with a broad peripheral feature set and enhanced computational performance. The PIC24FJ64GA004 family offers a new migration option for those high-performance applications which may be outgrowing their 8-bit platforms, but don't require the numerical processing power of a digital signal processor.

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 (DSCs). 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 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 PIC24FJ64GA004 family incorporate a range of features that can significantly reduce power consumption during operation. Key items 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 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

All of the devices in the PIC24FJ64GA004 family offer 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.
- A Fast Internal Oscillator (FRC) with a nominal 8 MHz output, which can also be divided under software control to provide clock speeds as low as 31 kHz.
- A Phase Lock Loop (PLL) frequency multiplier, available to the External Oscillator modes and the 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. 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.

TABLE 1-1: DEVICE FEATURES FOR THE PIC24FJ64GA004 FAMILY

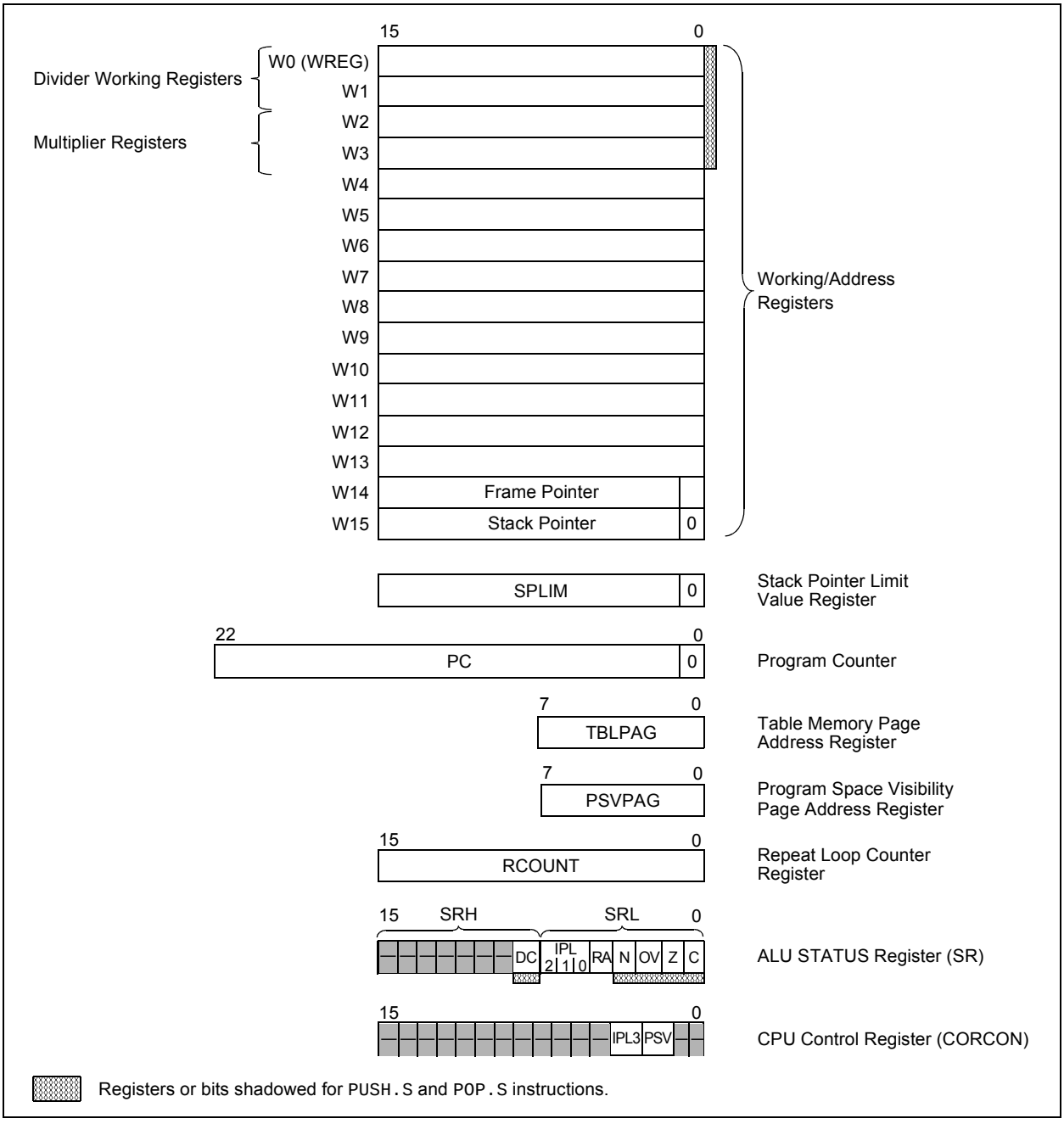
Features	16GA002	32GA002	48GA002	64GA002	16GA004	32GA004	48GA004	64GA004
Operating Frequency	DC – 32 MHz							
Program Memory (bytes)	16K	32K	48K	64K	16K	32K	48K	64K
Program Memory (instructions)	5,504	11,008	16,512	22,016	5,504	11,008	16,512	22,016
Data Memory (bytes)	4096	8192			4096	8192		
Interrupt Sources (soft vectors/NMI traps)	43 (39/4)							
I/O Ports	Ports A, B				Ports A, B, C			
Total I/O Pins	21				35			
Timers:								
Total Number (16-bit)	5(1)							
32-Bit (from paired 16-bit timers)	2							
Input Capture Channels	5(1)							
Output Compare/PWM Channels	5(1)							
Input Change Notification Interrupt	21				30			
Serial Communications:								
UART	2(1)							
SPI (3-wire/4-wire)	2(1)							
I^2C™	2							
Parallel Communications (PMP/PSP)	Yes							
JTAG Boundary Scan	Yes							
10-Bit Analog-to-Digital Module (input channels)	10				13			
Analog Comparators	2							
Remappable Pins	16				26			
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{MCLR}$, 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				44-Pin QFN/TQFP			

Note 1: Peripherals are accessible through remappable pins.

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



3.3.2 DIVIDER

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

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

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

3.3.3 MULTI-BIT SHIFT SUPPORT

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

A full summary of instructions that use the shift operation is provided below in Table 3-2.

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

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

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	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 Register																0000
TMR5HLD	0116	Timer5 Holding Register (for 32-bit operations only)																0000
TMR5	0118	Timer5 Register																0000
PR4	011A	Timer4 Period Register																FFFF
PR5	011C	Timer5 Period Register																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

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

TABLE 4-7: INPUT CAPTURE 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
IC1BUF	0140	Input Capture 1 Register																FFFF
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2BUF	0144	Input Capture 2 Register																FFFF
IC2CON	0146	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3BUF	0148	Input Capture 3 Register																FFFF
IC3CON	014A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4BUF	014C	Input Capture 4 Register																FFFF
IC4CON	014E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5BUF	0150	Input Capture 5 Register																FFFF
IC5CON	0152	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	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 pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 4-4. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

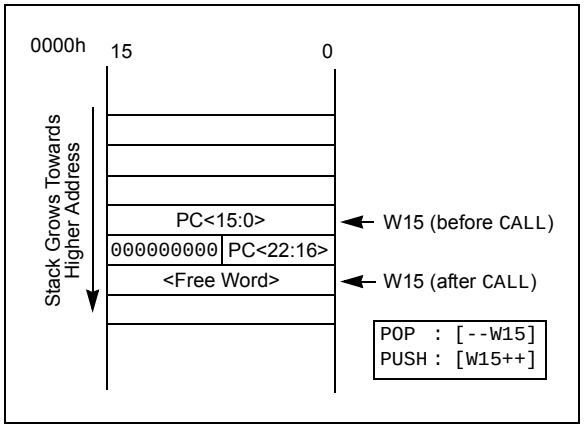
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

The Stack Pointer Limit Value register (SPLIM), 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' because 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 2000h in RAM, initialize the SPLIM with the value, 1FFEh.

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.

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

FIGURE 4-4: CALL STACK FRAME



4.3 Interfacing Program and Data Memory Spaces

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

Aside from normal execution, the PIC24F architecture provides two methods by which 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 (Program Space Visibility)

Table instructions allow an application to read or write to 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 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 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 Most Significant bit 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 table operations, this limits remapping operations strictly to the user memory area.

Table 4-25 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> refers to a program space word, whereas D<15:0> refers to a data space word.

REGISTER 7-23: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

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	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
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

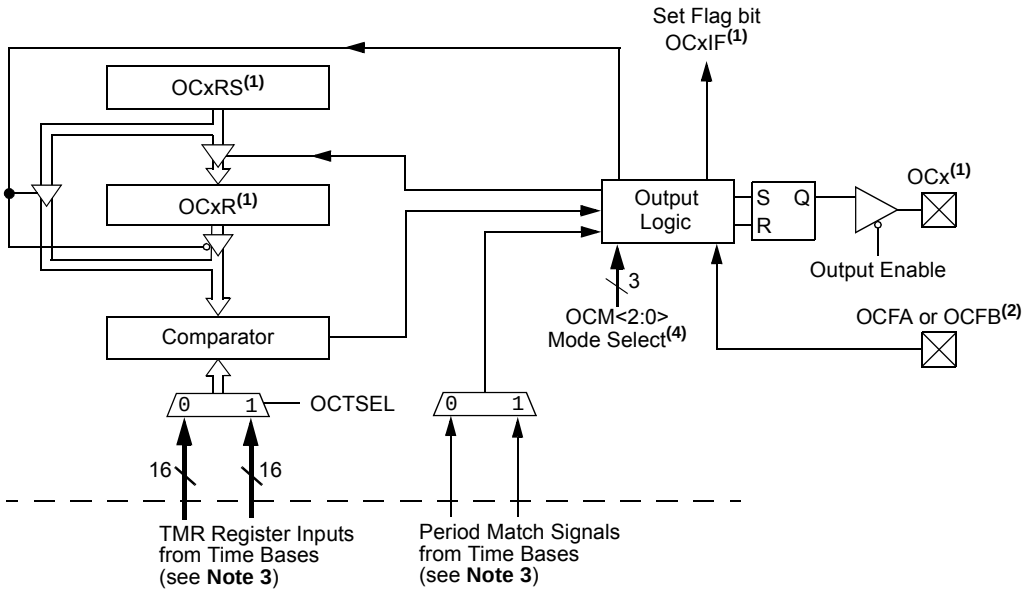
bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **SPI2IP<2:0>:** SPI2 Event 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 **SPF2IP<2:0>:** SPI2 Fault Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

FIGURE 14-1: OUTPUT COMPARE x MODULE BLOCK DIAGRAM



Note 1: Where 'x' is shown, reference is made to the registers associated with the respective Output Compare Channels 1 through 5.

2: The OCFA pin controls the OC1-OC4 channels. The OCFB pin controls the OC5 channel.

3: Each output compare channel can use one of two selectable time bases. Refer to the device data sheet for the time bases associated with the module.

4: This peripheral's inputs and outputs must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

14.4 Output Compare Register

REGISTER 14-1: OCxCON: OUTPUT COMPARE x CONTROL REGISTER

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

U-0	U-0	U-0	R-0, HC	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	OCFLT	OCTSEL	OCM2(1)	OCM1(1)	OCM0(1)
bit 7							bit 0

Legend:	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 15-14 **Unimplemented:** Read as '0'

bit 13 **OCSIDL:** Output Compare x Stop in Idle Mode Control bit
1 = Output Compare x halts in CPU Idle mode
0 = Output Compare x continues to operate in CPU Idle mode

bit 12-5 **Unimplemented:** Read as '0'

bit 4 **OCFLT:** PWM Fault Condition Status bit
1 = PWM Fault condition has occurred (cleared in HW only)
0 = No PWM Fault condition has occurred (this bit is only used when OCM<2:0> = 111)

bit 3 **OCTSEL:** Output Compare x Timer Select bit
1 = Timer3 is the clock source for Output Compare x
0 = Timer2 is the clock source for Output Compare x
Refer to the device data sheet for specific time bases available to the output compare module.

bit 2-0 **OCM<2:0>:** Output Compare x Mode Select bits(1)
111 = PWM mode on OCx; Fault pin, OCFx, is enabled(2)
110 = PWM mode on OCx; Fault pin, OCFx, is disabled(2)
101 = Initializes OCx pin low, generates continuous output pulses on OCx pin
100 = Initializes OCx pin low, generates single output pulse on OCx pin
011 = Compare event toggles OCx pin
010 = Initializes OCx pin high, compare event forces OCx pin low
001 = Initializes OCx pin low, compare event forces OCx pin high
000 = Output compare channel is disabled

Note 1: RPORx (OCx) must be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

2: The OCFA pin controls the OC1-OC4 channels. The OCFB pin controls the OC5 channel.

REGISTER 17-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

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

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: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: This feature is only available for the 16x BRG mode (BRGH = 0).

3: Bit availability depends on pin availability.

19.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

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, refer to the *"PIC24F Family Reference Manual"*, **"Real-Time Clock and Calendar (RTCC)"** (DS39696).

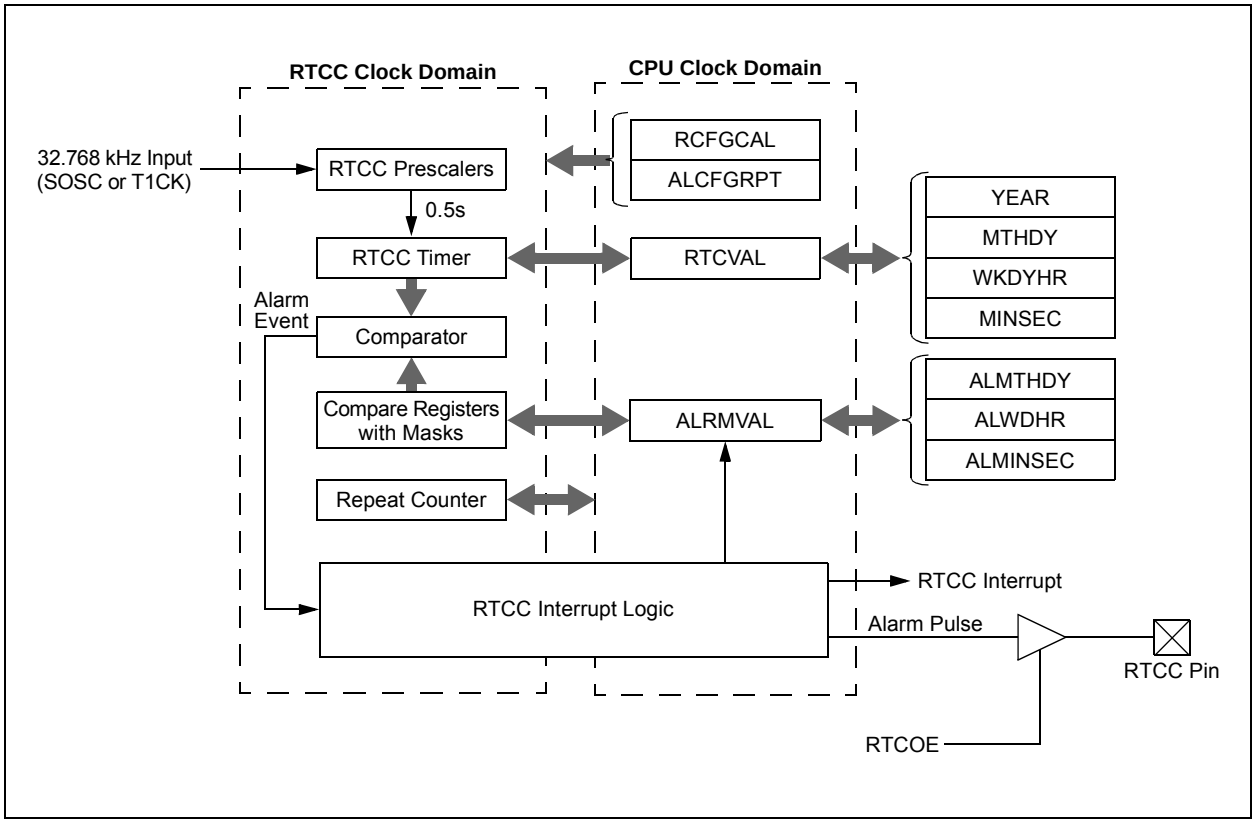
The Real-Time Clock and Calendar (RTCC) provides on-chip, hardware-based clock and calendar functionality with little or no CPU overhead. It is intended for applications where accurate time must be maintained for extended periods, with minimal CPU activity and with limited power resources, such as battery-powered applications.

Key features include:

- Time data in hours, minutes and seconds, with a granularity of one-half second
- 24-hour format (military time) display option
- Calendar data as date, month and year
- Automatic, hardware-based day of week and leap year calculations for dates from 2000 through 2099
- Time and calendar data in BCD format for compact firmware
- Highly configurable alarm function
- External output pin with selectable alarm signal or seconds "tick" signal output
- Time base input from Secondary Oscillator (SOSC) or the T1CK digital clock input (32.768 kHz)
- User calibration feature with auto-adjust

A simplified block diagram of the module is shown in Figure 19-1.The SOSC and RTCC will both remain running while the device is held in Reset with $\overline{\text{MCLR}}$, and will continue running after $\overline{\text{MCLR}}$ is released.

FIGURE 19-1: RTCC BLOCK DIAGRAM



19.1 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

19.1.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR<1:0> bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 19-1).

By writing the RTCVALH byte, the RTCC Pointer value (the RTCPTR<1:0> bits) decrements by one until the bits reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 19-1: RTCVAL REGISTER MAPPING

RTCPTR <1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 19-2).

By writing the ALRMVALH byte, the Alarm Pointer value (the ALRMPTR<1:0> bits) decrements by one until the bits reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 19-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL, the bytes will decrement the ALRMPTR<1:0> value. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

Note:	This only applies to read operations and not write operations.

19.1.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RCFGCAL<13>) must be set (refer to Example 19-1).

Note:	To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only 1 instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 19-1.

EXAMPLE 19-1: SETTING THE RTCWREN BIT

```
asm  volatile("push w7");
asm  volatile("push w8");
asm  volatile("disi #5");
asm  volatile("mov #0x55, w7");
asm  volatile("mov w7, _NVMKEY");
asm  volatile("mov #0xAA, w8");
asm  volatile("mov w8, _NVMKEY");
asm  volatile("bset _RCFGCAL, #13");  //set the RTCWREN bit
asm  volatile("pop w8");
asm  volatile("pop w7");
```

REGISTER 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER(1)

bit 7-0 **CAL<7:0>:** RTCC Drift Calibration bits
01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every one minute
...
01111111 = Minimum positive adjustment; adds 4 RTC clock pulses every one minute
00000000 = No adjustment
11111111 = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute
...
10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute

Note 1: The RCFGCAL register is only affected by a POR.
2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
3: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 19-2: PADCFG1: PAD CONFIGURATION 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	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL(1)	PMPTTL
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 15-2 **Unimplemented:** Read as '0'
bit 1 **RTSECSEL:** RTCC Seconds Clock Output Select bit(1)
1 = RTCC seconds clock is selected for the RTCC pin
0 = RTCC alarm pulse is selected for the RTCC pin
bit 0 **PMPTTL:** PMP Module TTL Input Buffer Select bit
1 = PMP module uses TTL input buffers
0 = PMP module uses Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL) bit needs to be set.

REGISTER 19-6: WKDYHR: WEEKDAY AND HOURS VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
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 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 19-7: MINSEC: MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
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 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits
Contains a value from 0 to 9.

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

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits
Contains a value from 0 to 9.

NOTES:

REGISTER 21-5: AD1PCFG: A/D PORT CONFIGURATION REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	—	—	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8(1)
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
PCFG7(1)	PCFG6(1)	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
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 **PCFG15:** Analog Input Pin Configuration Control bit
1 = Band gap voltage reference is disabled
0 = Band gap voltage reference is enabled

bit 14-13 **Unimplemented:** Read as '0'

bit 12-0 **PCFG<12:0>:** Analog Input Pin Configuration Control bits(1)
1 = Pin for corresponding analog channel is configured in Digital mode; I/O port read is enabled
0 = Pin is configured in Analog mode; I/O port read is disabled, A/D samples pin voltage

Note 1: Analog Channels, AN6, AN7 and AN8, are unavailable on 28-pin devices; leave these corresponding bits set.

REGISTER 21-6: AD1CSSL: A/D INPUT SCAN SELECT REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	—	—	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8(1)
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
CSSL7(1)	CSSL6(1)	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
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 **CSSL15:** Band Gap Reference Input Pin Scan Selection bit
1 = Band gap voltage reference channel is selected for input scan
0 = Band gap voltage reference channel is omitted from input scan

bit 14-13 **Unimplemented:** Read as '0'

bit 12-0 **CSSL<12:0>:** A/D Input Pin Scan Selection bits(1)
1 = Corresponding analog channel is selected for input scan
0 = Analog channel is omitted from input scan

Note 1: Analog Channels, AN6, AN7 and AN8, are unavailable on 28-pin devices; leave these corresponding bits cleared.

REGISTER 22-1: CMCON: COMPARATOR CONTROL REGISTER (CONTINUED)

bit 6 **C1OUT:** Comparator 1 Output bit
When C1INV = 0:
1 = C1 VIN+ > C1 VIN-
0 = C1 VIN+ < C1 VIN-
When C1INV = 1:
0 = C1 VIN+ > C1 VIN-
1 = C1 VIN+ < C1 VIN-

bit 5 **C2INV:** Comparator 2 Output Inversion bit
1 = C2 output is inverted
0 = C2 output is not inverted

bit 4 **C1INV:** Comparator 1 Output Inversion bit
1 = C1 output is inverted
0 = C1 output is not inverted

bit 3 **C2NEG:** Comparator 2 Negative Input Configure bit
1 = Input is connected to VIN+
0 = Input is connected to VIN-
See Figure 22-1 for the Comparator modes.

bit 2 **C2POS:** Comparator 2 Positive Input Configure bit
1 = Input is connected to VIN+
0 = Input is connected to CVREF
See Figure 22-1 for the Comparator modes.

bit 1 **C1NEG:** Comparator 1 Negative Input Configure bit
1 = Input is connected to VIN+
0 = Input is connected to VIN-
See Figure 22-1 for the Comparator modes.

bit 0 **C1POS:** Comparator 1 Positive Input Configure bit
1 = Input is connected to VIN+
0 = Input is connected to CVREF
See Figure 22-1 for the Comparator modes.

Note 1: If C2OUTEN = 1, the C2OUT peripheral output must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

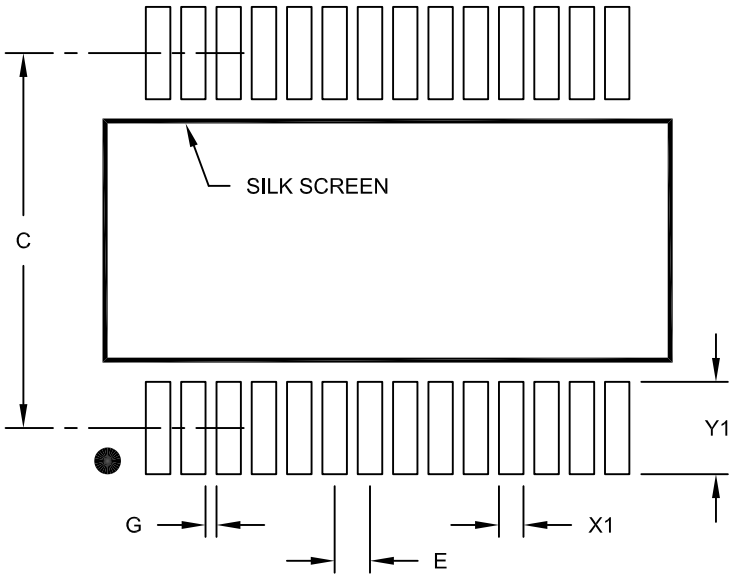
2: If C1OUTEN = 1, the C1OUT peripheral output must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

TABLE 26-2: 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,#lit4,Wnd	Wnd = Arithmetic Right Shift Wb by lit4	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

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		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

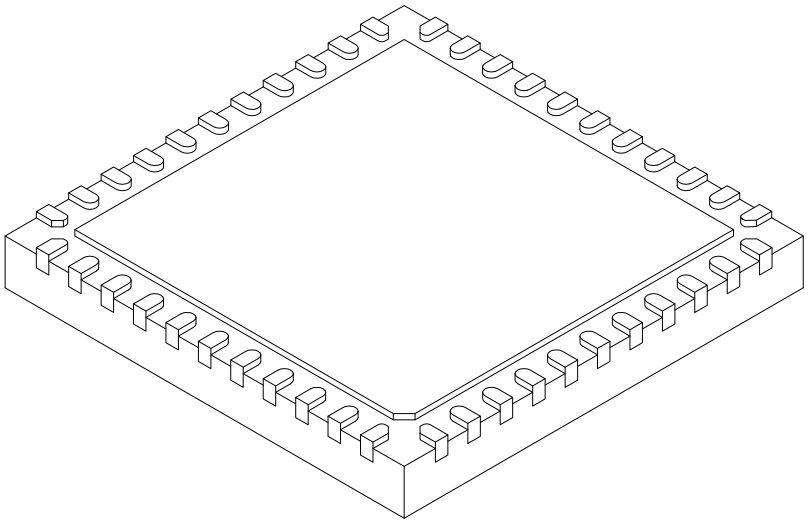
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

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		44	
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		8.00 BSC	
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.25	6.45	6.60
Terminal Width	b	0.20	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
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 C04-103C Sheet 2 of 2