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.
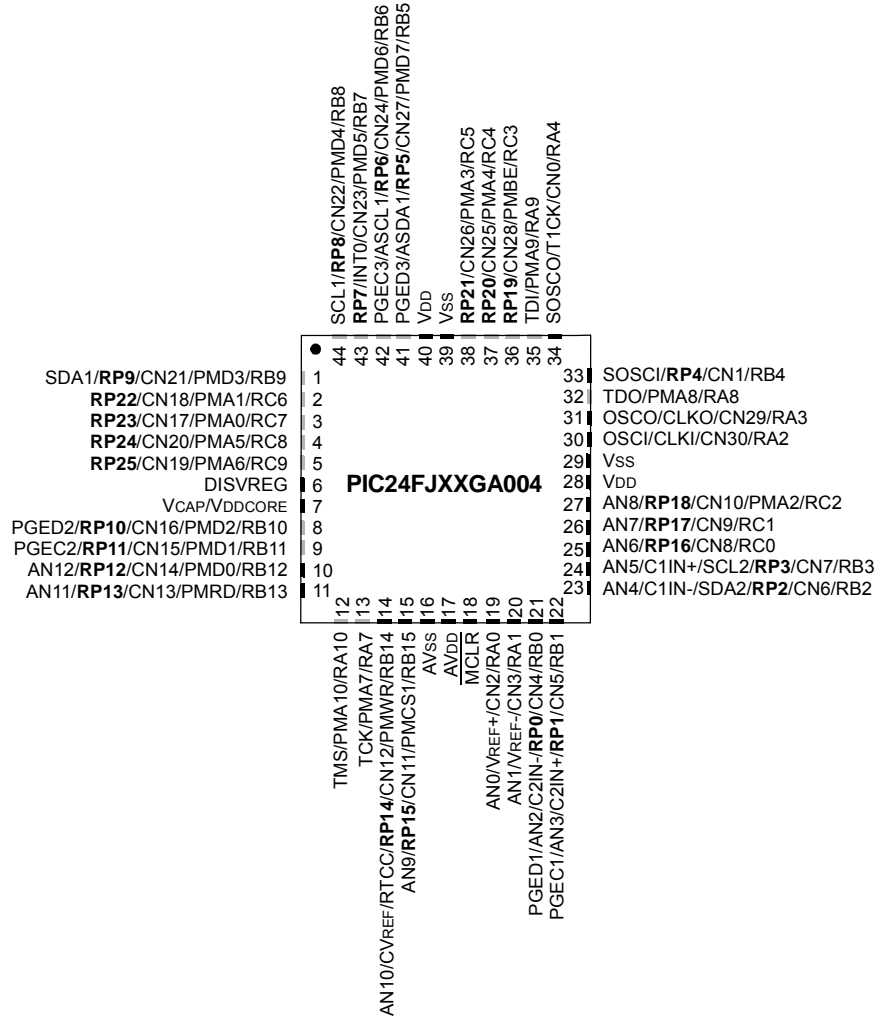
Applications of "Embedded - Microcontrollers"

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	32KB (11K 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 ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj32ga002-i-sp

Pin Diagrams (Continued)

44-Pin QFN(1)

Top row pins (left to right): SCL1/**RP8**/CN22/PMD4/RB8 (44), **RP7**/INT0/CN23/PMD5/RB7 (43), PGEC3/ASCL1/**RP6**/CN24/PMD6/RB6 (42), PGED3/ASDA1/**RP5**/CN27/PMD7/RB5 (41), VDD (40), VSS (39), **RP21**/CN26/PMA3/RC5 (38), **RP20**/CN25/PMA4/RC4 (37), **RP19**/CN28/PMBE/RC3 (36), TDI/PMA9/RA9 (35), SOSCO/T1CK/CN0/RA4 (34)

Left side pins (top to bottom):
- 1 SDA1/**RP9**/CN21/PMD3/RB9
- 2 **RP22**/CN18/PMA1/RC6
- 3 **RP23**/CN17/PMA0/RC7
- 4 **RP24**/CN20/PMA5/RC8
- 5 **RP25**/CN19/PMA6/RC9
- 6 DISVREG
- 7 VCAP/VDDCORE
- 8 PGED2/**RP10**/CN16/PMD2/RB10
- 9 PGEC2/**RP11**/CN15/PMD1/RB11
- 10 AN12/**RP12**/CN14/PMD0/RB12
- 11 AN11/**RP13**/CN13/PMRD/RB13

PIC24FJXXGA004

Right side pins (top to bottom):
- 33 SOSCI/**RP4**/CN1/RB4
- 32 TDO/PMA8/RA8
- 31 OSCO/CLKO/CN29/RA3
- 30 OSCI/CLKI/CN30/RA2
- 29 VSS
- 28 VDD
- 27 AN8/**RP18**/CN10/PMA2/RC2
- 26 AN7/**RP17**/CN9/RC1
- 25 AN6/**RP16**/CN8/RC0
- 24 AN5/C1IN+/SCL2/**RP3**/CN7/RB3
- 23 AN4/C1IN-/SDA2/**RP2**/CN6/RB2

Bottom row pins (left to right): TMS/PMA10/RA10 (12), TCK/PMA7/RA7 (13), AN10/CVREF/RTCC/**RP14**/CN12/PMWR/RB14 (14), AN9/**RP15**/CN11/PMCS1/RB15 (15), AVSS (16), AVDD (17), MCLR (18), AN0/VREF+/CN2/RA0 (19), AN1/VREF-/CN3/RA1 (20), PGED1/AN2/C2IN-/**RP0**/CN4/RB0 (21), PGEC1/AN3/C2IN+/**RP1**/CN5/RB1 (22)
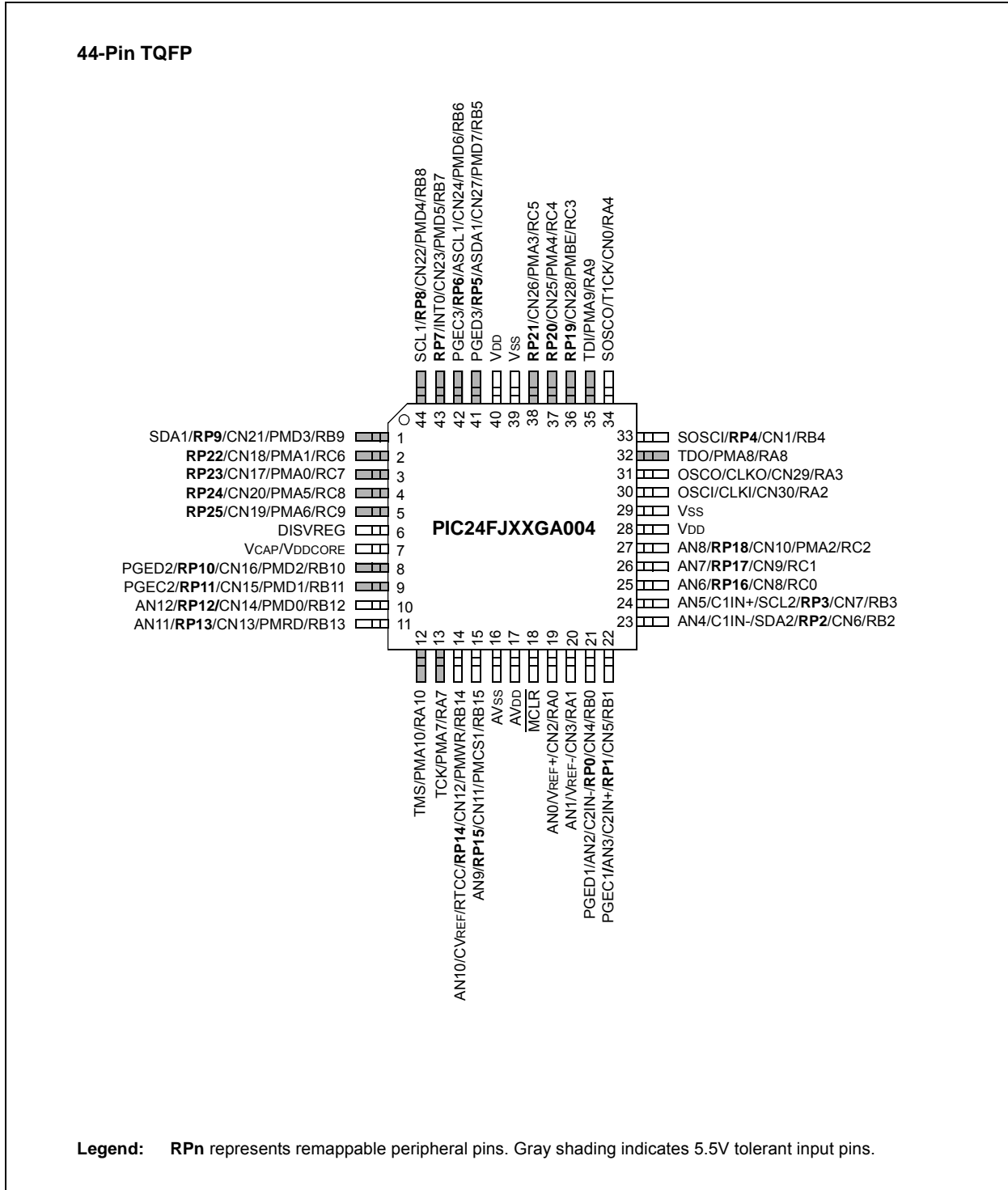
Legend: **RPn** represents remappable peripheral pins. Gray shading indicates 5.5V tolerant input pins.
Note 1: Back pad on QFN devices should be connected to Vss.

Pin Diagrams (Continued)

44-Pin TQFP

PIC24FJXXGA004

Left side (pins 1-11):
- 1 — SDA1/**RP9**/CN21/PMD3/RB9
- 2 — **RP22**/CN18/PMA1/RC6
- 3 — **RP23**/CN17/PMA0/RC7
- 4 — **RP24**/CN20/PMA5/RC8
- 5 — **RP25**/CN19/PMA6/RC9
- 6 — DISVREG
- 7 — VCAP/VDDCORE
- 8 — PGED2/**RP10**/CN16/PMD2/RB10
- 9 — PGEC2/**RP11**/CN15/PMD1/RB11
- 10 — AN12/**RP12**/CN14/PMD0/RB12
- 11 — AN11/**RP13**/CN13/PMRD/RB13

Right side (pins 23-33):
- 33 — SOSCI/**RP4**/CN1/RB4
- 32 — TDO/PMA8/RA8
- 31 — OSCO/CLKO/CN29/RA3
- 30 — OSCI/CLKI/CN30/RA2
- 29 — VSS
- 28 — VDD
- 27 — AN8/**RP18**/CN10/PMA2/RC2
- 26 — AN7/**RP17**/CN9/RC1
- 25 — AN6/**RP16**/CN8/RC0
- 24 — AN5/C1IN+/SCL2/**RP3**/CN7/RB3
- 23 — AN4/C1IN-/SDA2/**RP2**/CN6/RB2

Top side (pins 34-44):
- 44 — SCL1/**RP8**/CN22/PMD4/RB8
- 43 — **RP7**/INT0/CN23/PMD5/RB7
- 42 — PGEC3/**RP6**/ASCL1/CN24/PMD6/RB6
- 41 — PGED3/**RP5**/ASDA1/CN27/PMD7/RB5
- 40 — VDD
- 39 — VSS
- 38 — **RP21**/CN26/PMA3/RC5
- 37 — **RP20**/CN25/PMA4/RC4
- 36 — **RP19**/CN28/PMBE/RC3
- 35 — TDI/PMA9/RA9
- 34 — SOSCO/T1CK/CN0/RA4

Bottom side (pins 12-22):
- 12 — TMS/PMA10/RA10
- 13 — TCK/PMA7/RA7
- 14 — AN10/CVREF/RTCC/**RP14**/CN12/PMWR/RB14
- 15 — AN9/**RP15**/CN11/PMCS1/RB15
- 16 — AVSS
- 17 — AVDD
- 18 — MCLR
- 19 — AN0/VREF+/CN2/RA0
- 20 — AN1/VREF-/CN3/RA1
- 21 — PGED1/AN2/C2IN-/**RP0**/CN4/RB0
- 22 — PGEC1/AN3/C2IN+/**RP1**/CN5/RB1

Legend: **RPn** represents remappable peripheral pins. Gray shading indicates 5.5V tolerant input pins.

© 2010-2013 Microchip Technology Inc.

PIC24FJ64GA004 FAMILY

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	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	—	—	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	PMPIF	—	—	—	OC5IF	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	—	0000
IFS3	008A	—	RTCIF	—	—	—	—	—	—	—	—	—	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	—	—	—	—	LVDIF	—	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IEC0	0094	—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	PMPIE	—	—	—	OC5IE	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	—	0000
IEC3	009A	—	RTCIE	—	—	—	—	—	—	—	—	—	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	—	—	—	—	LVDIE	—	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4444
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	4444
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	4444
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4444
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	4444
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4444
IPC10	00B8	—	—	—	—	—	—	—	—	—	OC5IP2	OC5IP1	OC5IP0	—	—	—	—	4444
IPC11	00BA	—	—	—	—	—	—	—	—	—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—	4444
IPC12	00BC	—	—	—	—	—	MI2C2P2	MI2C2P1	MI2C2P0	—	SI2C2P2	SI2C2P1	SI2C2P0	—	—	—	—	4444
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	—	—	—	—	—	—	—	4444
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4444
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	4444
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

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

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.
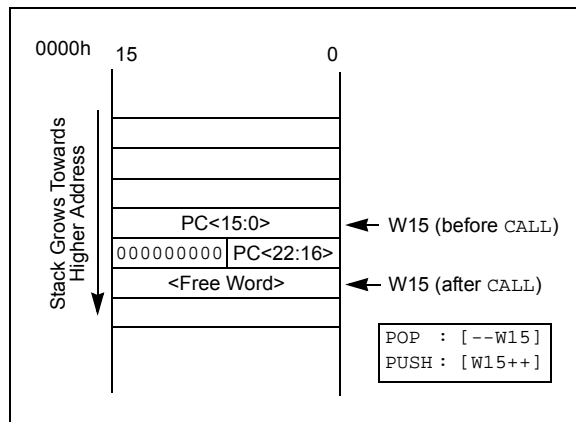
5.0 FLASH PROGRAM MEMORY

> **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, refer to the *"PIC24F Family Reference Manual"*, **"Program Memory"** (DS39715).

The PIC24FJ64GA004 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 2.25V.

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 PIC24FJ64GA004 family 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 PGCx and PGDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear (MCLR). 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.

RTSP is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 64 instructions (192 bytes) at a time and erase program memory in blocks of 512 instructions (1536 bytes) at a time.

5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory 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 shown in Figure 5-1.

The TBLRDL and the 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**
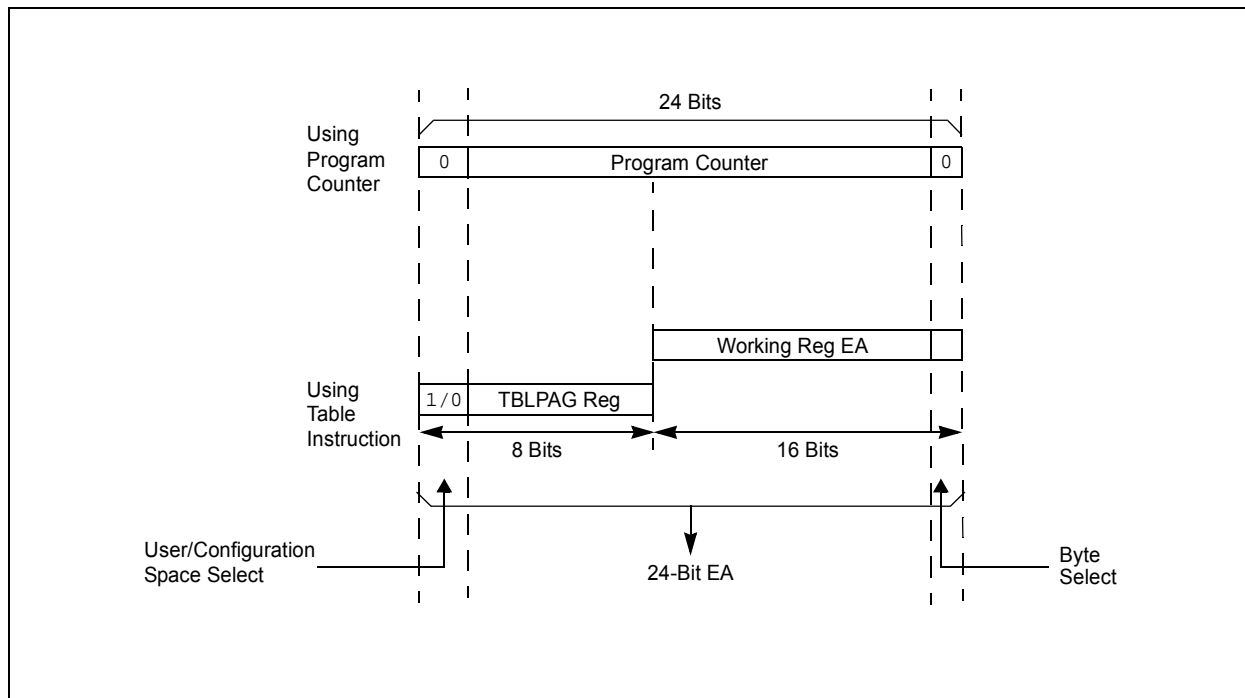
REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	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:		
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) Interrupt 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 7-28: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
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 15-11   **Unimplemented:** Read as '0'

bit 10-8   **RTCIP<2:0>:** Real-Time Clock/Calendar 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'

10.4   Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. The challenge is even greater on low pin count devices similar to the PIC24FJ64GA family. In an application that needs to use more than one peripheral multiplexed on a single pin, inconvenient work arounds in application code or a complete redesign may be the only option.

The Peripheral Pin Select feature provides an alternative to these choices by enabling the user's peripheral set selection and their placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. Peripheral Pin Select is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

10.4.1   AVAILABLE PINS

The Peripheral Pin Select feature is used with a range of up to 26 pins; the number of available pins is dependent on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation, "RPn", in their full pin designation, where "RP" designates a remappable peripheral and "n" is the remappable pin number. See Table 1-2 for pinout options in each package offering.

10.4.2   AVAILABLE PERIPHERALS

The peripherals managed by the Peripheral Pin Select are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer-related peripherals (input capture and output compare) and external interrupt inputs. Also included are the outputs of the comparator module, since these are discrete digital signals.

The Peripheral Pin Select module is not applied to I^2C™, Change Notification inputs, RTCC alarm outputs or peripherals with analog inputs.

A key difference between pin select and non-pin select peripherals is that pin select peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non-pin select peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

10.4.2.1   Peripheral Pin Select Function Priority

Pin-selectable peripheral outputs (for example, OC and UART transmit) take priority over any general purpose digital functions permanently tied to that pin, such as PMP and port I/O. Specialized digital outputs, such as USB functionality, take priority over PPS outputs on the same pin. The pin diagrams at the beginning of this data sheet list peripheral outputs in order of priority. Refer to them for priority concerns on a particular pin.

Unlike devices with fixed peripherals, pin-selectable peripheral inputs never take ownership of a pin. The pin's output buffer is controlled by the pin's TRIS bit setting or by a fixed peripheral on the pin. If the pin is configured in Digital mode, then the PPS input will operate correctly, reading the input. If an analog function is enabled on the same pin, the pin-selectable input will be disabled.

10.4.3   CONTROLLING PERIPHERAL PIN SELECT

Peripheral Pin Select features are controlled through two sets of Special Function Registers: one to map peripheral inputs and one to map outputs. Because they are separately controlled, a particular peripheral's input and output (if the peripheral has both) can be placed on any selectable function pin without constraint.

The association of a peripheral to a peripheral-selectable pin is handled in two different ways, depending on if an input or an output is being mapped.

10.4.3.1   Input Mapping

The inputs of the Peripheral Pin Select options are mapped on the basis of the peripheral; that is, a control register associated with a peripheral dictates the pin it will be mapped to. The RPINRx registers are used to configure peripheral input mapping (see Register 10-1 through Register 10-14). Each register contains two sets of 5-bit fields, with each set associated with one of the pin-selectable peripherals. Programming a given peripheral's bit field with an appropriate 5-bit value maps the RPn pin with that value to that peripheral. For any given device, the valid range of values for any of the bit fields corresponds to the maximum number of Peripheral Pin Selections supported by the device.

REGISTER 10-5:   RPINR7: PERIPHERAL PIN SELECT INPUT REGISTER 7

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
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
—	—	—	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
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 15-13   **Unimplemented:** Read as '0'

bit 12-8   **IC2R<4:0>:** Assign Input Capture 2 (IC2) to the Corresponding RPn Pin bits

bit 7-5   **Unimplemented:** Read as '0'

bit 4-0   **IC1R<4:0>:** Assign Input Capture 1 (IC1) to the Corresponding RPn Pin bits

REGISTER 10-6:   RPINR8: PERIPHERAL PIN SELECT INPUT REGISTER 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
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
—	—	—	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
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 15-13   **Unimplemented:** Read as '0'

bit 12-8   **IC4R<4:0>:** Assign Input Capture 4 (IC4) to the Corresponding RPn Pin bits

bit 7-5   **Unimplemented:** Read as '0'

bit 4-0   **IC3R<4:0>:** Assign Input Capture 3 (IC3) to the Corresponding RPn Pin bits

REGISTER 10-17: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R4	RP5R3	RP5R2	RP5R1	RP5R0
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
—	—	—	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
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 15-13 **Unimplemented:** Read as '0'

bit 12-8 **RP5R<4:0>:** Peripheral Output Function is Assigned to RP5 Output Pin bits
(see Table 10-3 for peripheral function numbers)

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RP4R<4:0>:** Peripheral Output Function is Assigned to RP4 Output Pin bits
(see Table 10-3 for peripheral function numbers)

REGISTER 10-18: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
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
—	—	—	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
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 15-13 **Unimplemented:** Read as '0'

bit 12-8 **RP7R<4:0>:** Peripheral Output Function is Assigned to RP7 Output Pin bits
(see Table 10-3 for peripheral function numbers)

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **RP6R<4:0>:** Peripheral Output Function is Assigned to RP6 Output Pin bits
(see Table 10-3 for peripheral function numbers)

14.3 Pulse-Width Modulation Mode

Note: This peripheral contains input and output functions that may need to be configured by the Peripheral Pin Select. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

The following steps should be taken when configuring the output compare module for PWM operation:

1. Set the PWM period by writing to the selected Timery Period register (PRy).
2. Set the PWM duty cycle by writing to the OCxRS register.
3. Write the OCxR register with the initial duty cycle.
4. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
5. Configure the output compare module for one of two PWM Operation modes by writing to the Output Compare Mode bits, OCM<2:0> (OCxCON<2:0>).
6. Set the TMRy prescale value and enable the time base by setting TON (TyCON<15>) = 1.

Note: The OCxR register should be initialized before the output compare module is first enabled. The OCxR register becomes a read-only Duty Cycle register when the module is operated in the PWM modes. The value held in OCxR will become the PWM duty cycle for the first PWM period. The contents of the Output Compare x Secondary register, OCxRS, will not be transferred into OCxR until a time base period match occurs.

14.3.1 PWM PERIOD

The PWM period is specified by writing to PRy, the Timery Period register. The PWM period can be calculated using Equation 14-1.

EQUATION 14-1: CALCULATING THE PWM PERIOD[1]

$$\text{PWM Period} = [(\text{PRy}) + 1] \bullet \text{TCY} \bullet (\text{Timer Prescale Value})$$

Where:
$$\text{PWM Frequency} = 1/[\text{PWM Period}]$$

Note 1: Based on $\text{TCY} = 2 * \text{TOSC}$; Doze mode and PLL are disabled.

Note: A PRy value of N will produce a PWM period of N + 1 time base count cycles. For example, a value of 7 written into the PRy register will yield a period consisting of 8 time base cycles.

14.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS register. The OCxRS register can be written to at any time, but the duty cycle value is not latched into OCxR until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitch-less PWM operation. In the PWM mode, OCxR is a read-only register.

Some important boundary parameters of the PWM duty cycle include:

- If the Output Compare x register, OCxR, is loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- If OCxR is greater than PRy (Timery Period register), the pin will remain high (100% duty cycle).
- If OCxR is equal to PRy, the OCx pin will be low for one time base count value and high for all other count values.

See Example 14-1 for PWM mode timing details. Table 14-1 and Table 14-2 show example PWM frequencies and resolutions for a device operating at 4 and 16 MIPS.

EQUATION 14-2: CALCULATION FOR MAXIMUM PWM RESOLUTION[1]

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{\text{FCY}}{\text{FPWM} \bullet (\text{Timer Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on $\text{FCY} = \text{FOSC}/2$; Doze mode and PLL are disabled.

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.

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");
```

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.

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.

EQUATION 21-1: **A/D CONVERSION CLOCK PERIOD[1]**

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

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

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

FIGURE 21-2: **10-BIT A/D CONVERTER ANALOG INPUT MODEL**
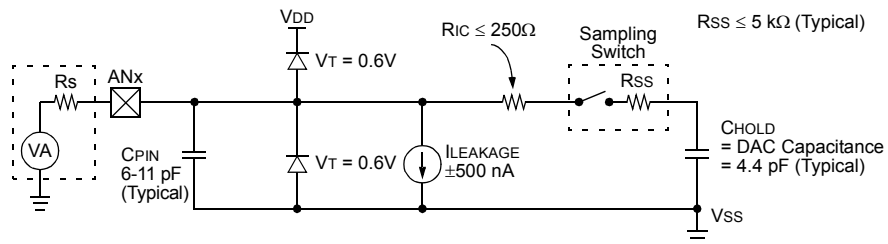
Legend: C_{PIN} = Input Capacitance
V_T = Threshold Voltage
$I_{LEAKAGE}$ = Leakage Current at the pin due to various junctions
R_{IC} = Interconnect Resistance
R_{SS} = Sampling Switch Resistance
C_{HOLD} = Sample/Hold Capacitance (from DAC)

Note: C_{PIN} value depends on device package and is not tested. The effect of C_{PIN} is negligible if Rs $\leq$ 5 kΩ.

25.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

25.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

TABLE 26-1: 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] }

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
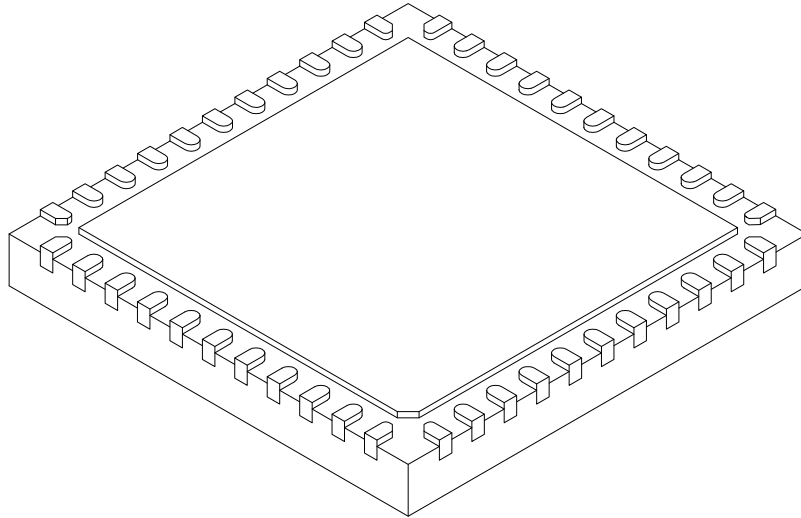
	Units		MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		44	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		8.00 BSC	
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.25	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

APPENDIX A: REVISION HISTORY

Revision A (March 2007)

Original data sheet for the PIC24FJ64GA004 family of devices.

Revision B (March 2007)

Changes to Table 26-8; packaging diagrams updated.

Revision C (January 2008)

- Update of electrical specifications to include DC characteristics for Extended Temperature devices.
- Update for A/D converter chapter to include information on internal band gap voltage reference.
- Added "**Appendix B: "Additional Guidance for PIC24FJ64GA004 Family Applications"**.
- General revisions to incorporate corrections included in document errata to date (DS80333).

Revision D (January 2010)

- Update of electrical specifications to include 60°C specifications for power-down current to DC characteristics.
- Removes references to JTAG programming throughout the document.
- Other minor typographic corrections throughout.

Revision E (May 2013)

- Updates all pin diagrams to indicate 5V tolerant pins.
- Updates all package labeling diagrams.
- Changes the VREGS bit name (RCON<8>) to PMSLP in all occurrences throughout the data sheet; also updates the description of the bit's functionality in Register 6-1. (The actual operation of the bit remains unchanged.)
- Adds additional explanatory text to the following sections:
  - **Section 9.2.1 "Sleep Mode"**
  - **Section 10.4.2.1 "Peripheral Pin Select Function Priority"**
  - **Section 24.2.3 "On-Chip Regulator and POR"**

- Updates **Section 2.0 "Guidelines for Getting Started with 16-Bit Microcontrollers"** with the most current information on V$_{CAP}$ selection.
- Replaces Table 6-3 (Reset Delay Times) with an updated version.
- Updates **Section 7.0 "Interrupt Controller"** by adding a description of the INTTREG register (Register 7-31).
- Updates **Section 8.0 "Oscillator Configuration"** by correcting the external oscillator inputs in Figure 8-1 and a new unlock code sequence in Example 8-1.
- Replaces Example 10-2 with a new code example.
- Updates **Section 19.0 "Real-Time Clock and Calendar (RTCC)"** to add introductory text and amend input sources in Figure 19-1.
- Updates **Section 20.0 "Programmable Cyclic Redundancy Check (CRC) Generator"** with a more current version (no technical changes to the module or its operation).
- Updates **Section 26.0 "Instruction Set Summary"**:
  - Updates syntax of ASR, DAW, LSR, MOV and SL instructions to conform with the *Programmer's Reference Manual*
  - Adds previously omitted instruction, FBCL
- Adds to **Section 27.0 "Electrical Characteristics"**:
  - New Specification DC18 (V$_{BOR}$) to Table 27-3
  - New Specifications DI60a (I$_{ICL}$), DI60b (I$_{ICH}$) and DI60c (ΣI$_{ICT}$) to Table 27-7
  - New Table 27-10 (Comparator Specifications) and Table 27-11 (Comparator Voltage Reference Specifications); previous Table 27-10 is now renumbered as Table 27-12, and all subsequent tables renumbered accordingly
  - New Table 27-17 (Internal RC Oscillator Specifications)
  - New specifications, AD08 (I$_{VREF}$), AD09 (Z$_{REF}$) and AD13 (Leakage Current), to Table 27-20
  - Combines previous Table 27-15 (AC Characteristics: Internal RC Accuracy) and Table 27-16 (Internal RC Accuracy) into a new Table 27-18 (AC Characteristics: Internal RC Accuracy)
- Other minor typographic corrections throughout.

PIC24FJ64GA004 FAMILY