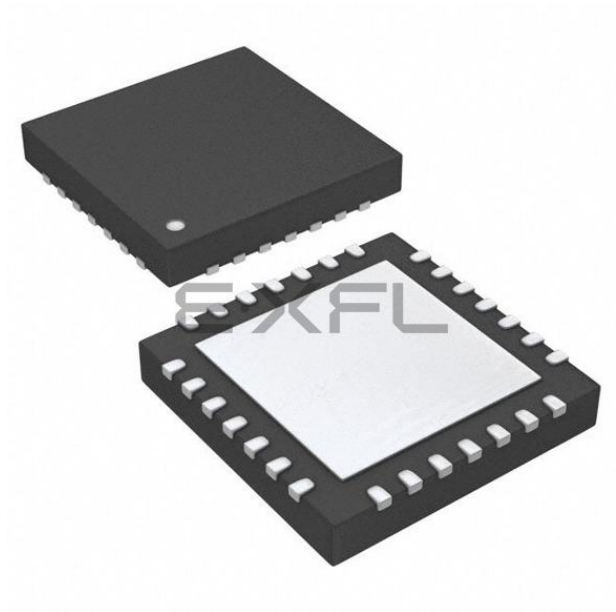
Microchip Technology - PIC24FJ64GA102-I/ML Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

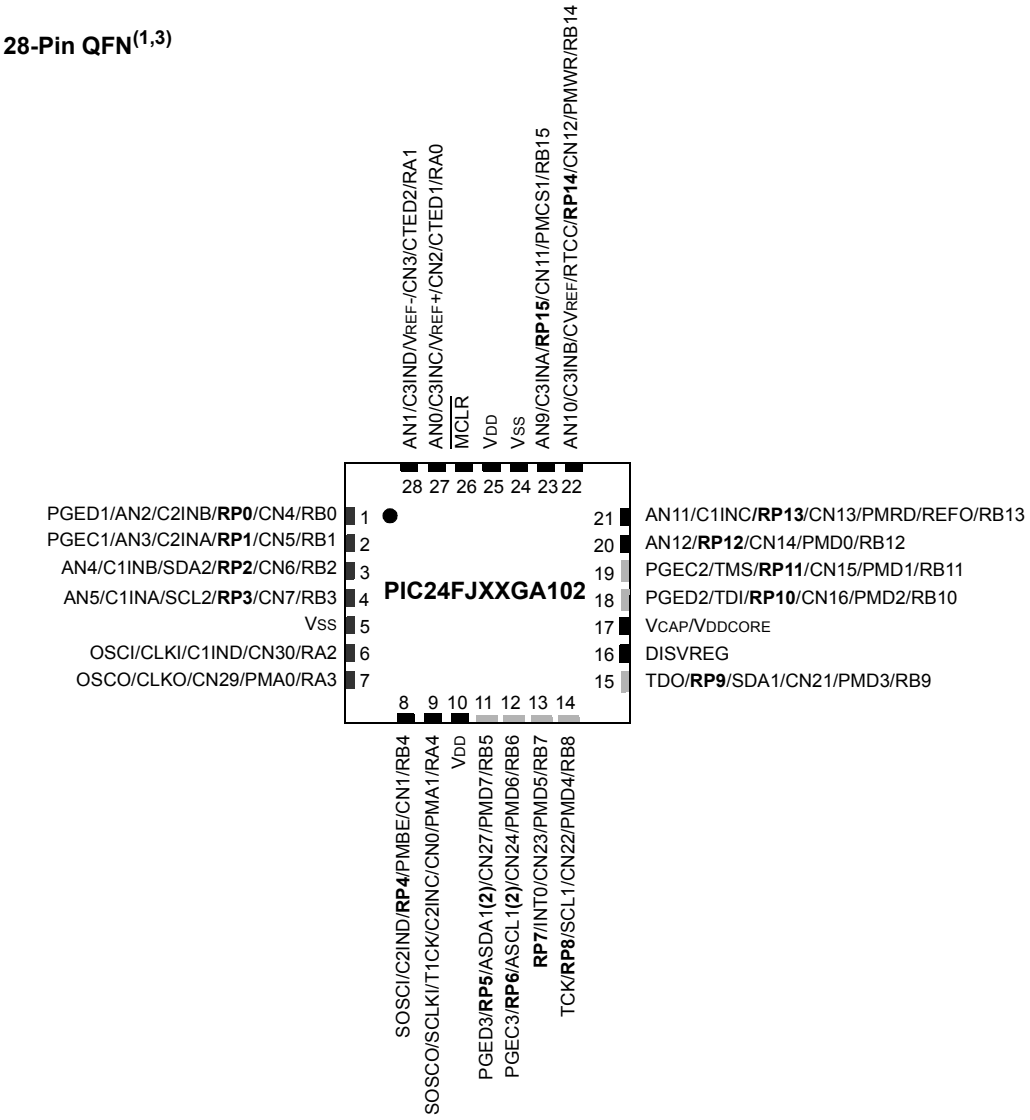
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, 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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj64ga102-i-ml

Pin Diagrams

28-Pin QFN(1,3)



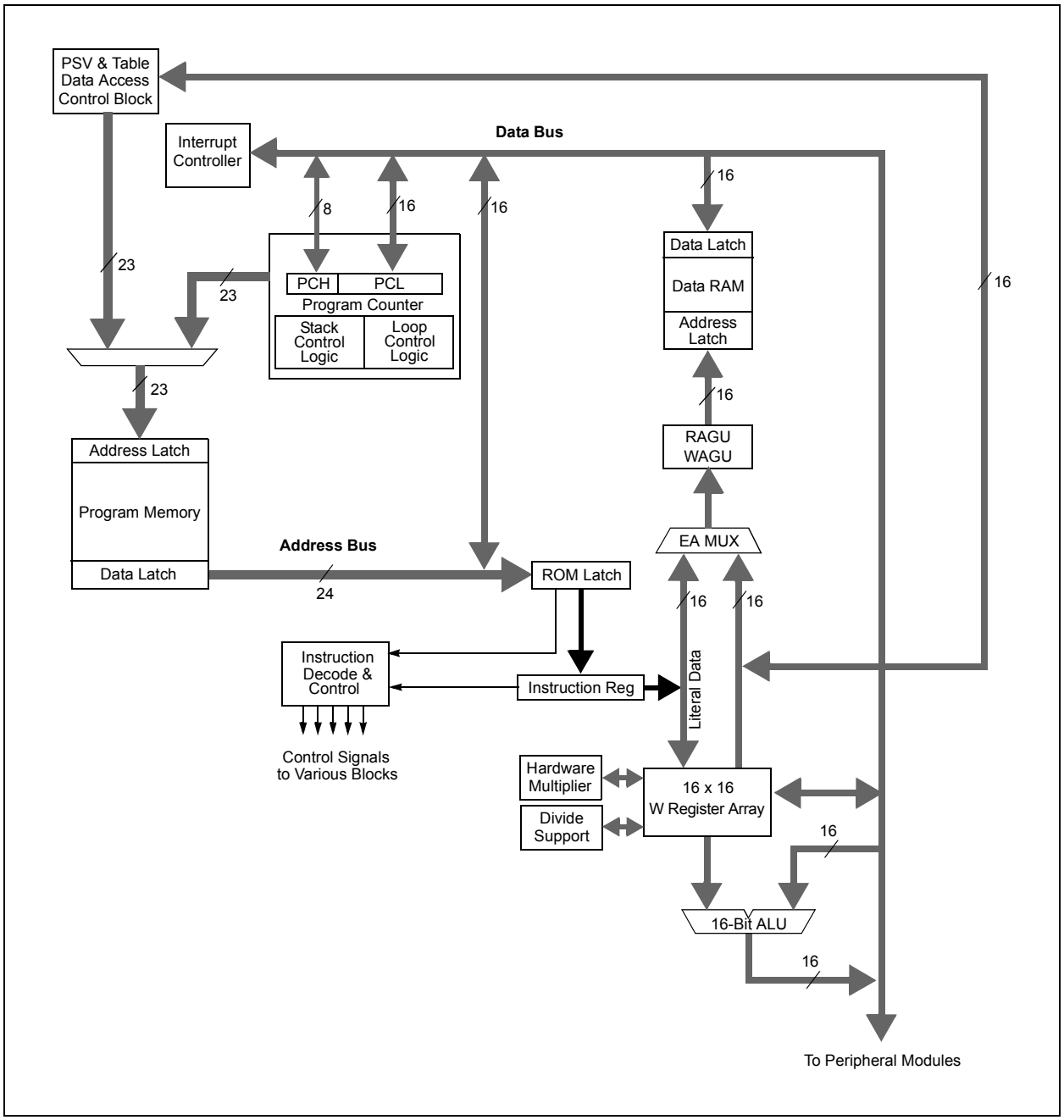
Legend: **RPn** represents remappable peripheral pins.

Note 1: Gray shading indicates 5.5V tolerant input pins.

2: Alternative multiplexing for SDA1 and SCL1 when the I2C1SEL bit is set.

3: The back pad on QFN devices should be connected to VSS.

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM



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 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 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 Least Significant Byte. The Most Significant Byte is not modified.

A Sign-Extend (SE) instruction is provided to allow 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. 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 indirectly addressable. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing with a 16-bit address field.

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 module. Much of the SFR space contains unused addresses; these are read as '0'. A diagram of the SFR space, showing where SFRs are actually implemented, is shown 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 shown in Tables 4-3 through 4-26.

TABLE 4-2: IMPLEMENTED REGIONS OF SFR DATA SPACE

	SFR Space Address							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	Core			ICN	Interrupts			—
100h	Timers		Capture		Compare			—
200h	I^2C™	UART	SPI	—	—	—	I/O	
300h	A/D	A/D/CTMU	—	—	—	—	—	—
400h	—	—	—	—	—	—	—	—
500h	—	—	—	—	—	—	—	—
600h	PMP	RTCC	CRC/Comp	Comparators	PPS			—
700h	—	—	System/DS	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block

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.

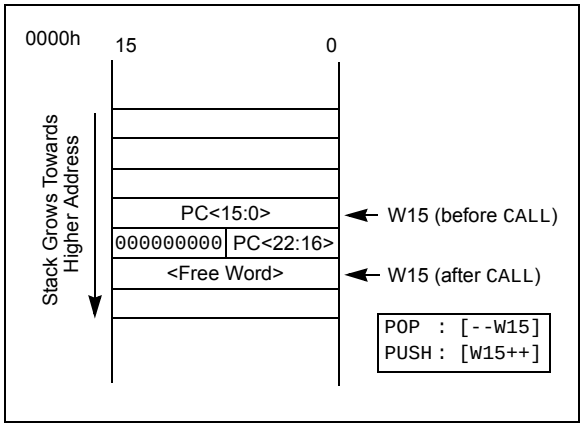
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

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' 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 a 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 (TBLPAG) register 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 (PSVPAG) register 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.

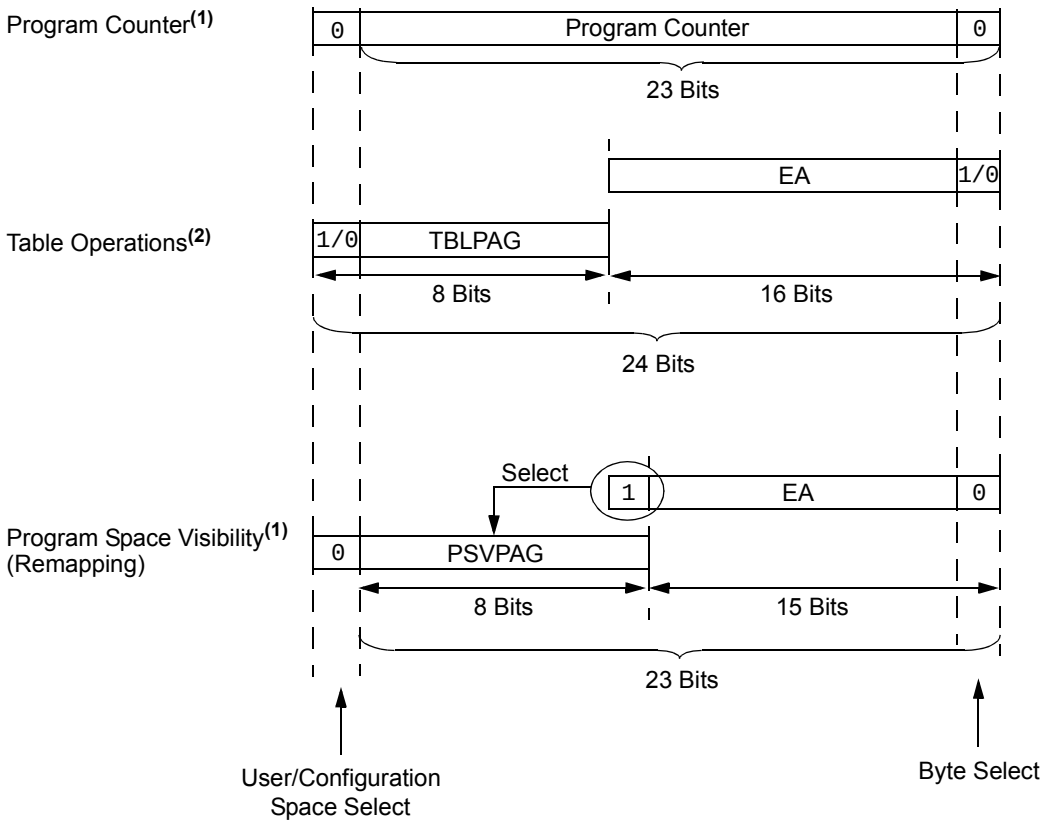
Table 4-27 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

TABLE 4-27: 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>		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>.

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 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 space without going through data space. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

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'.

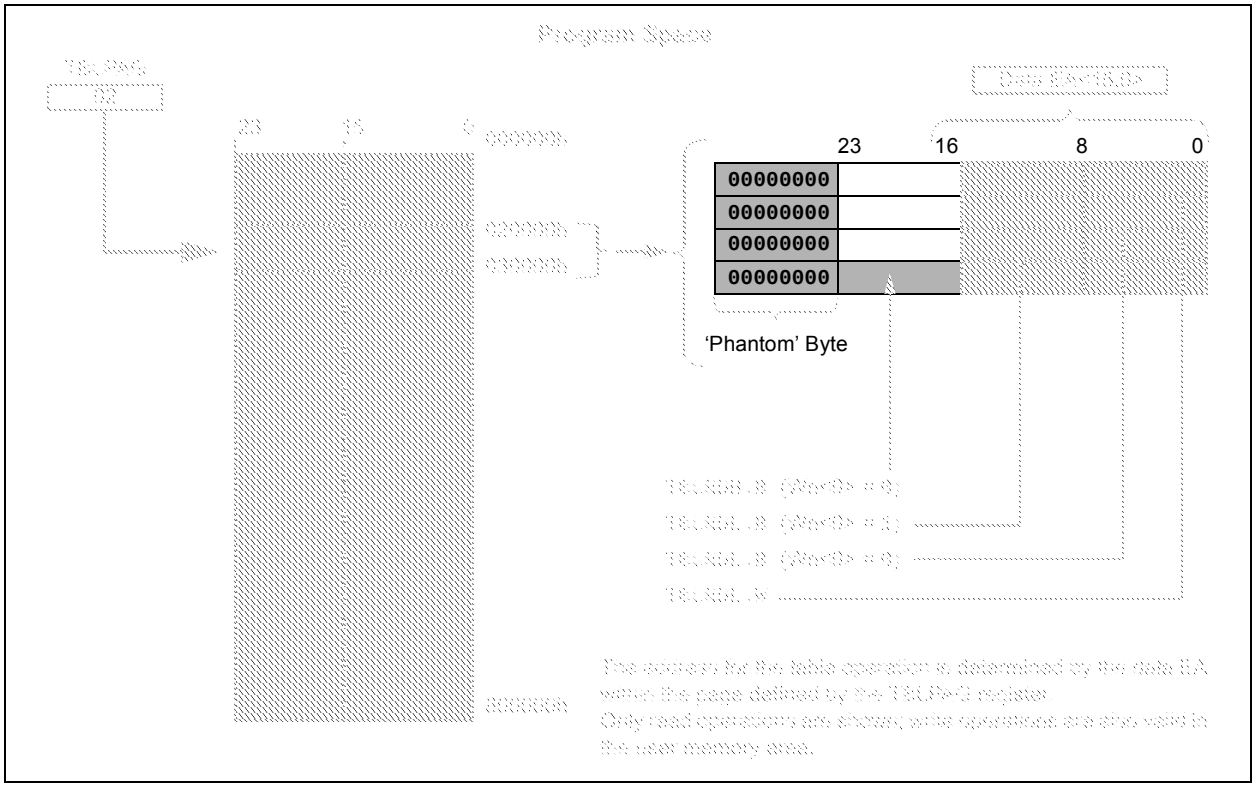
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '0' when the upper 'phantom' byte is selected (byte select = 1).

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



6.0 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, refer to the *"PIC24F Family Reference Manual"*, **Section 7. "Reset"** (DS39712).

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

- POR: Power-on Reset
- $\overline{MCLR}$: Pin Reset
- SWR: RESET Instruction
- WDT: Watchdog Timer Reset
- BOR: Brown-out Reset
- CM: Configuration Mismatch 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 6-1.

Any active source of Reset will make the $\overline{SYSRST}$ 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 POR and unchanged by all other Resets.

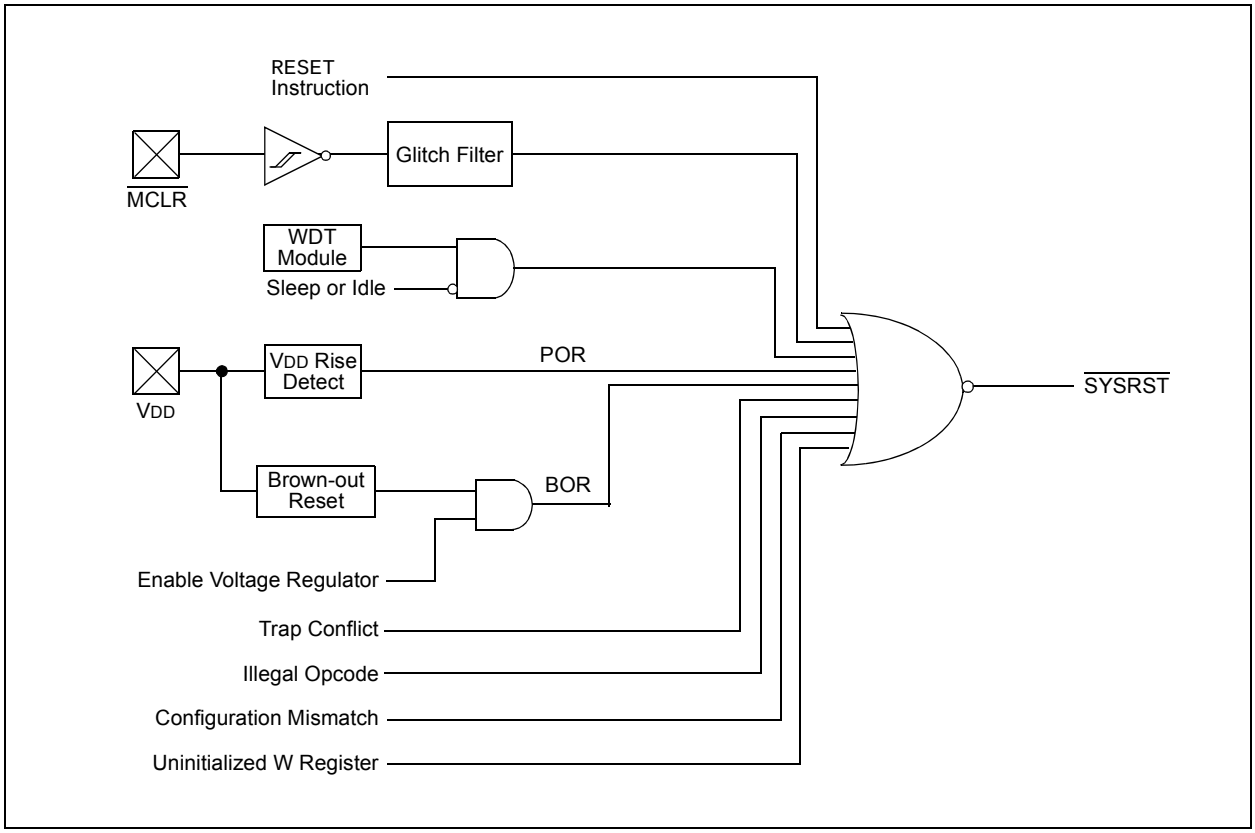
Note:	Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1). A Power-on Reset 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 and device power-saving states. The function of these bits is discussed in other sections of this data sheet.

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 6-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 7-20: 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 7-25: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

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
—	OC5IP2	OC5IP1	OC5IP0	—	—	—	—
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 **OC5IP<2:0>:** Output Compare Channel 5 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 7-26: IPC11: INTERRUPT PRIORITY CONTROL REGISTER 11

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
—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—
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 **PMPIP<2:0>:** Parallel Master Port 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 7-27: 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
—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—
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 **MI2C2IP<2:0>:** Master I2C2 Event 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 **SI2C2IP<2:0>:** Slave I2C2 Event 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 10-7: RPINR9: PERIPHERAL PIN SELECT INPUT REGISTER 9

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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **IC5R<4:0>:** Assign Input Capture 5 (IC5) to Corresponding RPn or RPIn Pin bits

REGISTER 10-8: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
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
—	—	—	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
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 **OCFBR<4:0>:** Assign Output Compare Fault B (OCFB) to Corresponding RPn or RPIn Pin bits

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

bit 4-0 **OCFAR<4:0>:** Assign Output Compare Fault A (OCFA) to Corresponding RPn or RPIn Pin bits

REGISTER 10-9: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
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
—	—	—	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
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 **U1CTSR<4:0>:** Assign UART1 Clear to Send ($\overline{\text{U1CTS}}$) to Corresponding RPn or RPIn Pin bits

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

bit 4-0 **U1RXR<4:0>:** Assign UART1 Receive (U1RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-10: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
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
—	—	—	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
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 **U2CTSR<4:0>:** Assign UART2 Clear to Send ($\overline{\text{U2CTS}}$) to Corresponding RPn or RPIn Pin bits

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

bit 4-0 **U2RXR<4:0>:** Assign UART2 Receive (U2RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-21: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
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
—	—	—	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
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 **RP13R<4:0>:** RP13 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP13 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP12R<4:0>:** RP12 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP12 (see Table 10-3 for peripheral function numbers).

REGISTER 10-22: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP15R4	RP15R3	RP15R2	RP15R1	RP15R0
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
—	—	—	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
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 **RP15R<4:0>:** RP15 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP0 (see Table 10-3 for peripheral function numbers).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP14R<4:0>:** RP14 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP14 (see Table 10-3 for peripheral function numbers).

14.0 OUTPUT COMPARE WITH DEDICATED TIMERS

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"*, **Section 35. "Output Capture with Dedicated Timer"** (DS39723).

All devices in the PIC24FJ64GA104 family features 5 independent output compare modules. Each of these modules offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce Pulse-Width Modulated (PWM) waveforms for driving power applications.

Key features of the output compare module include:

- Hardware-configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 21 user-selectable trigger/sync sources available
- Two separate Period registers (a main register, OCxR, and a secondary register, OCxRS) for greater flexibility in generating pulses of varying widths
- Configurable for single pulse or continuous pulse generation on an output event or continuous PWM waveform generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

14.1 General Operating Modes

14.1.1 SYNCHRONOUS AND TRIGGER MODES

By default, the output compare module operates in a Free-Running mode. The internal 16-bit counter, OCxTMR, runs counts up continuously, wrapping around from FFFFh to 0000h on each overflow with its period synchronized to the selected external clock source. Compare or PWM events are generated each time a match between the internal counter and one of the Period registers occurs.

In Synchronous mode, the module begins performing its compare or PWM operation as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the module's internal counter is reset. In Trigger mode, the module waits for a sync event from another internal module to occur before allowing the counter to run.

Free-Running mode is selected by default or any time that the SYNCSEL bits (OCxCON2<4:0>) are set to '00000'. Synchronous or Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The OCTRIG bit (OCxCON2<7>) selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

14.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own set of 16-bit Timer and Duty Cycle registers. To increase the range, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd-numbered module (OCx) provides the Least Significant 16 bits of the 32-bit register pairs and the even-numbered module (OCy) provides the Most Significant 16 bits. Wrap-arounds of the OCx registers cause an increment of their corresponding OCy registers.

Cascaded operation is configured in hardware by setting the OC32 bit (OCxCON2<8>) for both modules.

REGISTER 18-2: PMMODE: PARALLEL PORT MODE REGISTER

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM1	IRQM0	INCM1	INCM0	MODE16	MODE1	MODE0
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
WAITB1(1)	WAITB0(1)	WAITM3	WAITM2	WAITM1	WAITM0	WAITE1(1)	WAITE0(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 **BUSY:** Busy bit (Master mode only)
1 = Port is busy (not useful when the processor stall is active)
0 = Port is not busy

bit 14-13 **IRQM<1:0>:** Interrupt Request Mode bits
11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode), or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = No interrupt is generated, processor stall activated
01 = Interrupt is generated at the end of the read/write cycle
00 = No interrupt is generated

bit 12-11 **INCM<1:0>:** Increment Mode bits
11 = PSP read and write buffers auto-increment (Legacy PSP mode only)
10 = Decrement ADDR<10:0> by 1 every read/write cycle
01 = Increment ADDR<10:0> by 1 every read/write cycle
00 = No increment or decrement of address

bit 10 **MODE16:** 8/16-Bit Mode bit
1 = 16-bit mode: Data register is 16 bits; a read or write to the Data register invokes two 8-bit transfers
0 = 8-bit mode: Data register is 8 bits; a read or write to the Data register invokes one 8-bit transfer

bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master Mode 1 (PMCS1, PMRD/$\overline{\text{PMWR}}$, PMENB, PMBE, PMA<x:0> and PMD<7:0>)
10 = Master Mode 2 (PMCS1, PMRD, $\overline{\text{PMWR}}$, $\overline{\text{PMBE}}$, PMA<x:0> and PMD<7:0>)
01 = Enhanced PSP control signals ($\overline{\text{PMRD}}$, $\overline{\text{PMWR}}$, $\overline{\text{PMCS1}}$, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port control signals ($\overline{\text{PMRD}}$, $\overline{\text{PMWR}}$, $\overline{\text{PMCS1}}$ and PMD<7:0>)

bit 7-6 **WAITB<1:0>:** Data Setup to Read/Write Wait State Configuration bits(1)
11 = Data wait of 4 TCY; multiplexed address phase of 4 TCY
10 = Data wait of 3 TCY; multiplexed address phase of 3 TCY
01 = Data wait of 2 TCY; multiplexed address phase of 2 TCY
00 = Data wait of 1 TCY; multiplexed address phase of 1 TCY

bit 5-2 **WAITM<3:0>:** Read to Byte Enable Strobe Wait State Configuration bits
1111 = Wait of additional 15 TCY
...
0001 = Wait of additional 1 TCY
0000 = No additional wait cycles (operation forced into one TCY)

bit 1-0 **WAITE<1:0>:** Data Hold After Strobe Wait State Configuration bits(1)
11 = Wait of 4 TCY
10 = Wait of 3 TCY
01 = Wait of 2 TCY
00 = Wait of 1 TCY

Note 1: WAITB and WAITE bits are ignored whenever WAITM<3:0> = 0000.

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"*, **Section 29. "Real-Time Clock and Calendar (RTCC)"** (DS39696).

The RTCC provides the user with a Real-Time Clock and Calendar (RTCC) function that can be calibrated.

Key features of the RTCC module are:

- Operates in Deep Sleep mode
- Selectable clock source
- Provides hours, minutes and seconds using 24-hour format
- Visibility of one half second period
- Provides calendar – weekday, date, month and year
- Alarm-configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year 2000 to 2099 leap year correction
- BCD format for smaller software overhead
- Optimized for long-term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust

19.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator or the LPRC Low-Power Internal Oscillator as the clock reference for the RTCC module. This is configured using the RTCOSC (CW4<5>) Configuration bit. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

The SOSC and RTCC will both remain running while the device is held in Reset with $\overline{MCLR}$ and will continue running after $\overline{MCLR}$ is released.

FIGURE 19-1: RTCC BLOCK DIAGRAM

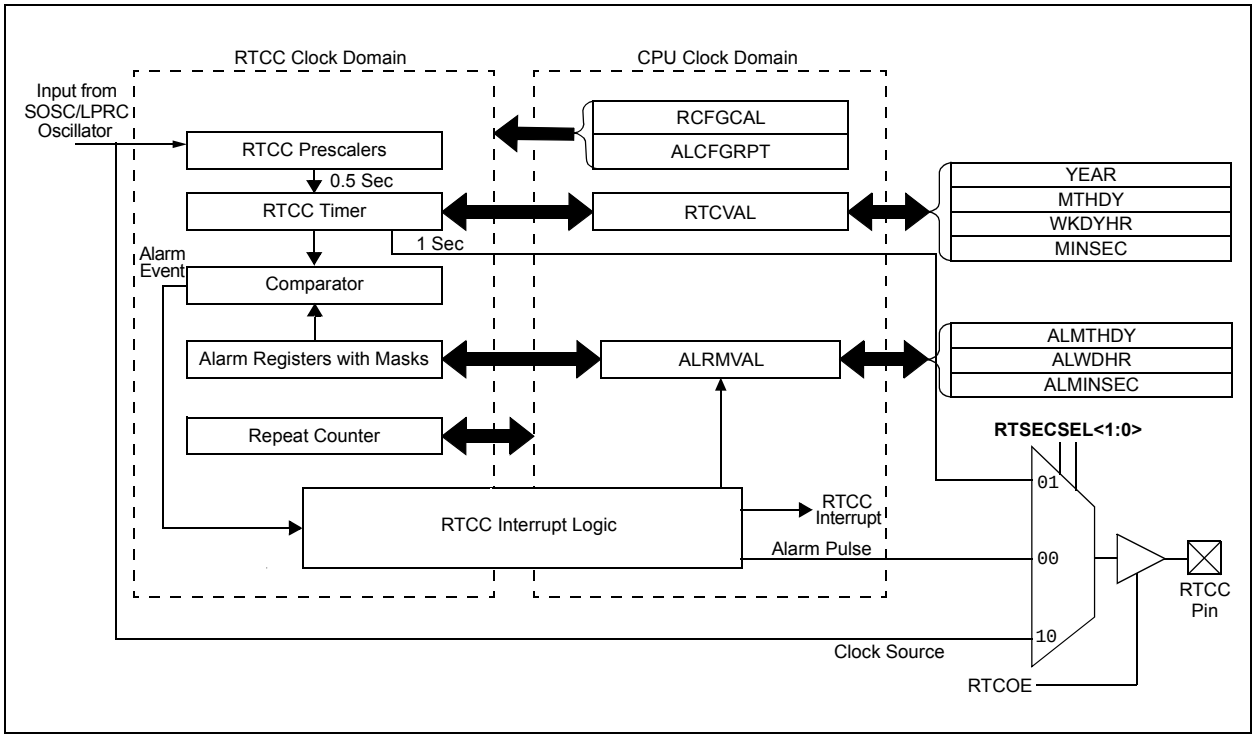


TABLE 27-2: INSTRUCTION SET OVERVIEW (CONTINUED)

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

28.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FJ64GA104 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FJ64GA104 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 +135°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +4.0V
Voltage on any combined analog and digital pin, and $\overline{MCLR}$, 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 +6.0V
Voltage on V_{DDCORE} with respect to V_{SS} ..... -0.3V to +3.0V
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin **(Note 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 **(Note 1)** ..... 200 mA

Note 1: Maximum allowable current is a function of device maximum power dissipation (see Table 28-1).

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.

S

T

U