Welcome to E-XFL.COM

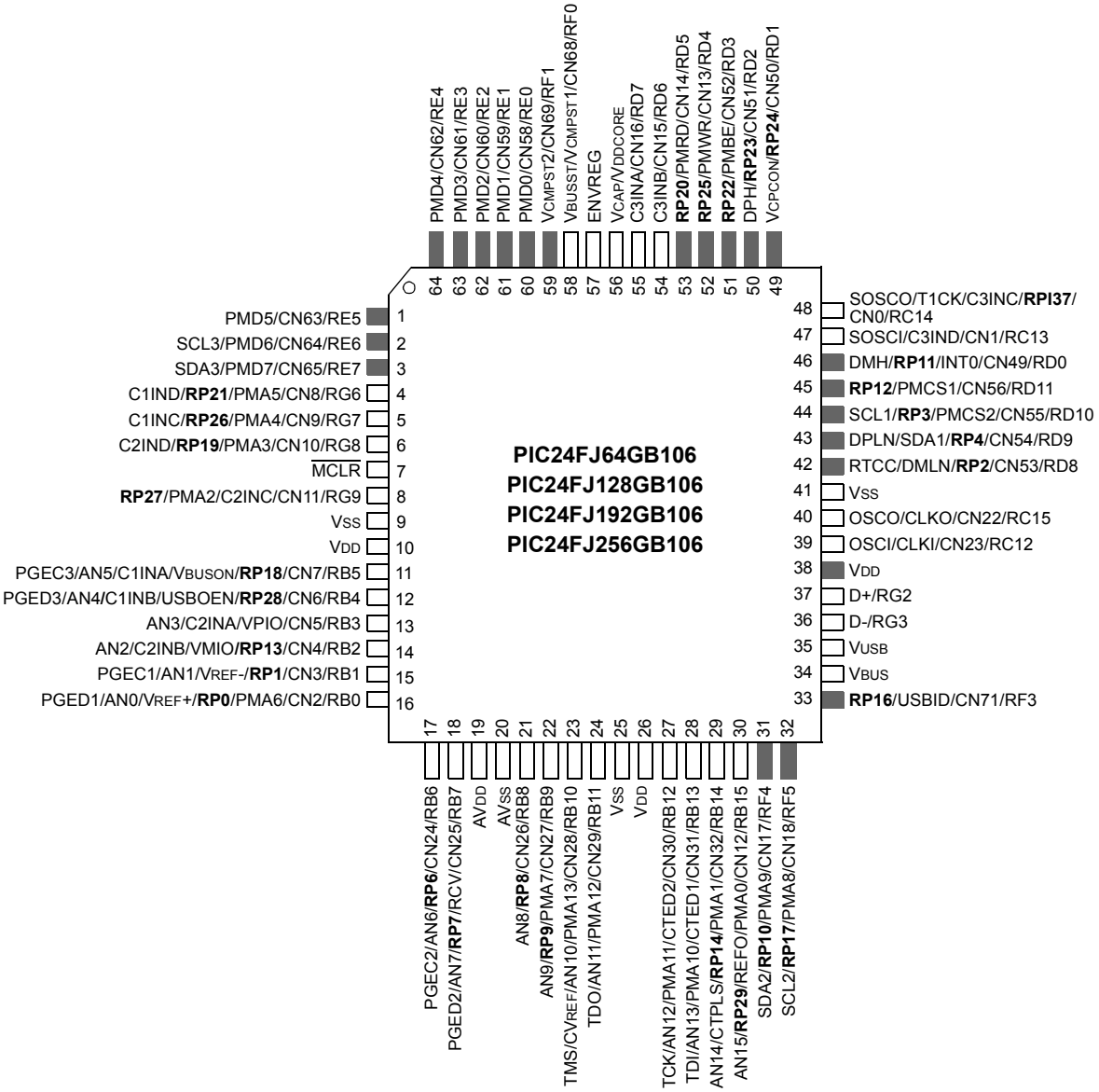
What is "Embedded - Microcontrollers"?

"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, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	65
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256gb108-i-pt

Pin Diagram (64-Pin TQFP and QFN)



Legend: **Shaded pins** indicate pins tolerant to up to +5.5 VDC.
RPn represents remappable pins for the Peripheral Pin Select feature.

Note 1: For QFN devices, the backplane on the underside of the device must also be connected to VSS.

TABLE 1-4: PIC24FJ256GB110 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP, QFN	80-Pin TQFP	100-Pin TQFP			
CN43	—	52	66	I	ST	Interrupt-on-Change Inputs.
CN44	—	53	67	I	ST	
CN45	—	4	6	I	ST	
CN46	—	—	7	I	ST	
CN47	—	5	8	I	ST	
CN48	—	—	9	I	ST	
CN49	46	58	72	I	ST	
CN50	49	61	76	I	ST	
CN51	50	62	77	I	ST	
CN52	51	63	78	I	ST	
CN53	42	54	68	I	ST	
CN54	43	55	69	I	ST	
CN55	44	56	70	I	ST	
CN56	45	57	71	I	ST	
CN57	—	64	79	I	ST	
CN58	60	76	93	I	ST	
CN59	61	77	94	I	ST	
CN60	62	78	98	I	ST	
CN61	63	79	99	I	ST	
CN62	64	80	100	I	ST	
CN63	1	1	3	I	ST	
CN64	2	2	4	I	ST	
CN65	3	3	5	I	ST	
CN66	—	13	18	I	ST	
CN67	—	14	19	I	ST	
CN68	58	72	87	I	ST	
CN69	59	73	88	I	ST	
CN70	—	42	52	I	ST	
CN71	33	41	51	I	ST	
CN74	—	43	53	I	ST	
CN75	—	—	40	I	ST	
CN76	—	—	39	I	ST	
CN77	—	75	90	I	ST	
CN78	—	74	89	I	ST	
CN79	—	—	96	I	ST	
CN80	—	—	97	I	ST	
CN81	—	—	95	I	ST	
CN82	—	—	1	I	ST	
CTED1	28	34	42	I	ANA	CTMU External Edge Input 1.
CTED2	27	33	41	I	ANA	CTMU External Edge Input 2.
CTPLS	29	35	43	O	—	CTMU Pulse Output.
CVREF	23	29	34	O	—	Comparator Voltage Reference Output.

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

2.7 Configuration of Analog and Digital Pins During ICSP Operations

If an ICSP compliant emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as “digital” pins. Depending on the particular device, this is done by setting all bits in the ADnPCFG register(s), or clearing all bit in the ANSx registers.

All PIC24F devices will have either one or more ADnPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to **Section 22.0 “10-Bit High-Speed A/D Converter”** for more specific information.

The bits in these registers that correspond to the A/D pins that initialized the emulator must not be changed by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the ADC module, as follows:

- For devices with an ADnPCFG register, clear the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.
- For devices with ANSx registers, set the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.

When a Microchip debugger/emulator is used as a programmer, the user application firmware must correctly configure the ADnPCFG or ANSx registers. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.8 Unused I/Os

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

5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase blocks of eight rows (512 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

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

Any number of TBLWT instructions can be executed and a write will be successfully performed. However, 64 TBLWT instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused addresses should be programmed with FFFFFFh. This is because the holding latches reset to an unknown state, so if the addresses are left in the Reset state, they may overwrite the locations on rows which were not rewritten.

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

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

Note:	Writing to a location multiple times without erasing is *not* recommended.

All of the table write operations are single-word writes (2 instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 JTAG Operation

The PIC24F family supports JTAG boundary scan. Boundary scan can improve the manufacturing process by verifying pin-to-PCB connectivity.

5.4 Enhanced In-Circuit Serial Programming

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

5.5 Control Registers

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

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.6 "Programming Operations"** for further details.

5.6 Programming Operations

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

TABLE 6-1: RESET FLAG BIT OPERATION

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

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

6.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to **Section 8.0 "Oscillator Configuration"** for further details.

TABLE 6-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

Reset Type	Clock Source Determinant
POR	FNOSC Configuration bits (CW2<10:8>)
BOR	
$\overline{\text{MCLR}}$	COSC Control bits (OSCCON<14:12>)
WDTO	
SWR	

6.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 6-3. Note that the system Reset signal, $\overline{\text{SYSRST}}$, is released after the POR and PWRT delay times expire.

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

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

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIF	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
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 15-14 **Unimplemented:** Read as '0'

bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **OC8IF:** Output Compare Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **OC7IF:** Output Compare Channel 7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC6IF:** Output Compare Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **IC6IF:** Input Capture Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-2 **Unimplemented:** Read as '0'

bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SPF2IF:** SPI2 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-13: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIE	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE
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 15-14 **Unimplemented:** Read as '0'

bit 13 **PMPIE:** Parallel Master Port Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **OC8IE:** Output Compare Channel 8 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **OC7IE:** Output Compare Channel 7 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **OC6IE:** Output Compare Channel 6 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **OC5IE:** Output Compare Channel 5 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **IC6IE:** Input Capture Channel 6 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **IC5IE:** Input Capture Channel 5 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **IC4IE:** Input Capture Channel 4 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **IC3IE:** Input Capture Channel 3 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4-2 **Unimplemented:** Read as '0'

bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 0 **SPF2IE:** SPI2 Fault Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-20: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

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

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

bit 6-4 **AD1IP<2:0>:** A/D Conversion Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-28: 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	R/W-1	R/W-0	R/W-0
—	PMPIP2	PMPIP1	PMPIP0	—	OC8IP2	OC8IP1	OC8IP0
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 **Unimplemented:** Read as ‘0’

bit 2-0 **OC8IP<2:0>:** Output Compare Channel 8 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 10-31: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
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 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP19R<5:0>:** RP19 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP19 (see Table 10-3 for peripheral function numbers)

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

bit 5-0 **RP18R<5:0>:** RP18 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP18 (see Table 10-3 for peripheral function numbers)

REGISTER 10-32: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
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

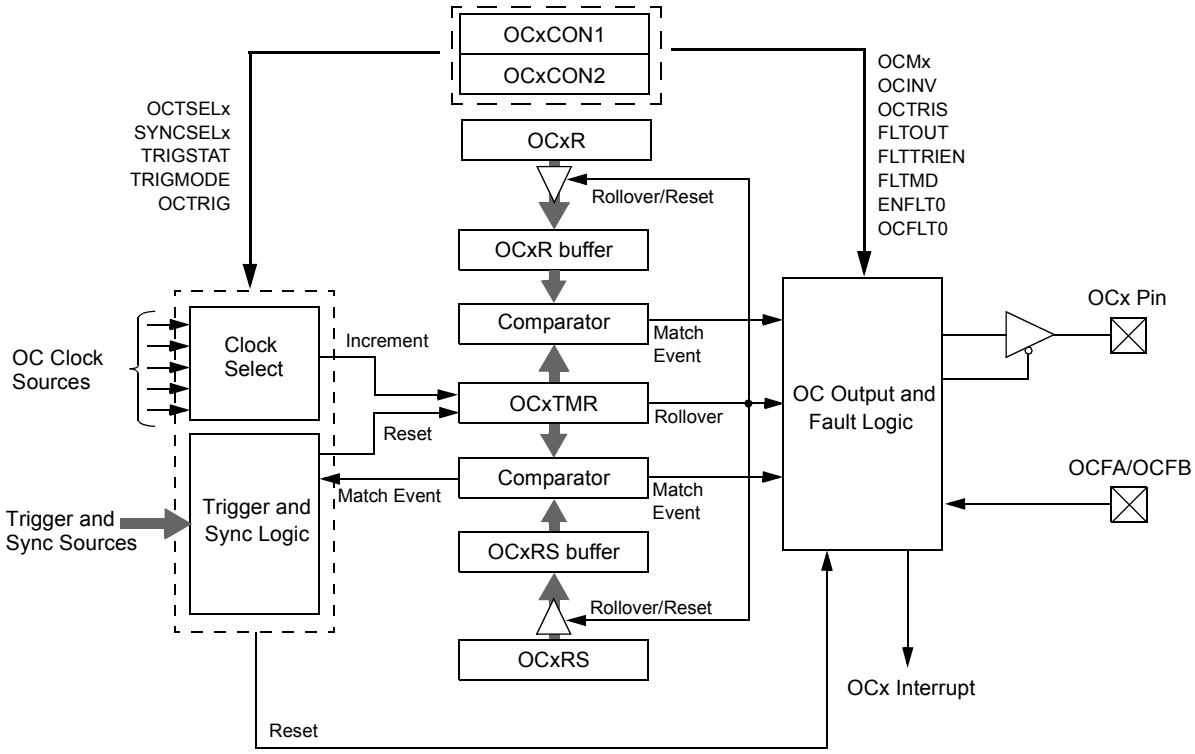
bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP21R<5:0>:** RP21 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP21 (see Table 10-3 for peripheral function numbers)

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

bit 5-0 **RP20R<5:0>:** RP20 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP20 (see Table 10-3 for peripheral function numbers)

FIGURE 14-2: OUTPUT COMPARE BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. Please see **Section 10.4 "Peripheral Pin Select"** for more information.

14.3.1 PWM PERIOD

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

EQUATION 14-1: CALCULATING THE PWM PERIOD(1)

PWM Period = [(PRy) + 1] • TCY • (Timer Prescale Value)

where: PWM Frequency = 1/[PWM Period]

Note 1: Based on TCY = TOSC * 2, 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 and OCxR registers. The OCxRS and OCxR registers can be written to at any time, but the duty cycle value is not latched 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 glitchless PWM operation.

Some important boundary parameters of the PWM duty cycle include:

- If OCxR, OCxRS, and PRy are all loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- ·If OCxRS is greater than PRy, the pin will remain high (100% duty cycle).

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 MIPS and 10 MIPS, respectively.

REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	R-0	R-0	R-0
SPIEN(1)	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0	R/C-0 HS	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

Legend:	C = Clearable bit	HS = Hardware settable 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 **SPIEN:** SPIx Enable bit(1)
1 = Enables module and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables module

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

bit 13 **SPISIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-11 **Unimplemented:** Read as '0'

bit 10-8 **SPIBEC<2:0>:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)
Master mode:
Number of SPI transfers pending.
Slave mode:
Number of SPI transfers unread.

bit 7 **SRMPT:** Shift Register (SPIxSR) Empty bit (valid in Enhanced Buffer mode)
1 = SPIx Shift register is empty and ready to send or receive
0 = SPIx Shift register is not empty

bit 6 **SPIROV:** Receive Overflow Flag bit
1 = A new byte/word is completely received and discarded. The user software has not read the previous data in the SPIxBUF register.
0 = No overflow has occurred

bit 5 **SRXMPT:** Receive FIFO Empty bit (valid in Enhanced Buffer mode)
1 = Receive FIFO is empty
0 = Receive FIFO is not empty

bit 4-2 **SISEL<2:0>:** SPIx Buffer Interrupt Mode bits (valid in Enhanced Buffer mode)
111 = Interrupt when SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when last bit is shifted into SPIxSR, as a result, the TX FIFO is empty
101 = Interrupt when the last bit is shifted out of SPIxSR, now the transmit is complete
100 = Interrupt when one data is shifted into the SPIxSR, as a result, the TX FIFO has one open spot
011 = Interrupt when SPIx receive buffer is full (SPIRBF bit set)
010 = Interrupt when SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in receive buffer (SRMPT bit is set)
000 = Interrupt when the last data in the receive buffer is read, as a result, the buffer is empty (SRXMPT bit set)

Note 1: If SPIEN = 1, these functions must be assigned to available RPn pins before use. See **Section 10.4 "Peripheral Pin Select"** for more information.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

bit 4-2 **SPRE<2:0>:** Secondary Prescale bits (Master mode)
111 = Secondary prescale 1:1
110 = Secondary prescale 2:1
...
000 = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>:** Primary Prescale bits (Master mode)
11 = Primary prescale 1:1
10 = Primary prescale 4:1
01 = Primary prescale 16:1
00 = Primary prescale 64:1

Note 1: If DISSCK = 0, SCKx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

3: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

4: If SSEN = 1, $\overline{SSx}$ must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

REGISTER 15-3: SPIxCON2: SPIx CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
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 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support enabled
0 = Framed SPIx support disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control on $\overline{SSx}$ pin bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

bit 13 **SPIFPOL:** Frame Sync Pulse Polarity bit (Frame mode only)
1 = Frame sync pulse is active-high
0 = Frame sync pulse is active-low

bit 12-2 **Unimplemented:** Read as '0'

bit 1 **SPIFE:** Frame Sync Pulse Edge Select bit
1 = Frame sync pulse coincides with first bit clock
0 = Frame sync pulse precedes first bit clock

bit 0 **SPIBEN:** Enhanced Buffer Enable bit
1 = Enhanced Buffer enabled
0 = Enhanced Buffer disabled (Legacy mode)

FIGURE 15-5: SPI MASTER, FRAME MASTER CONNECTION DIAGRAM

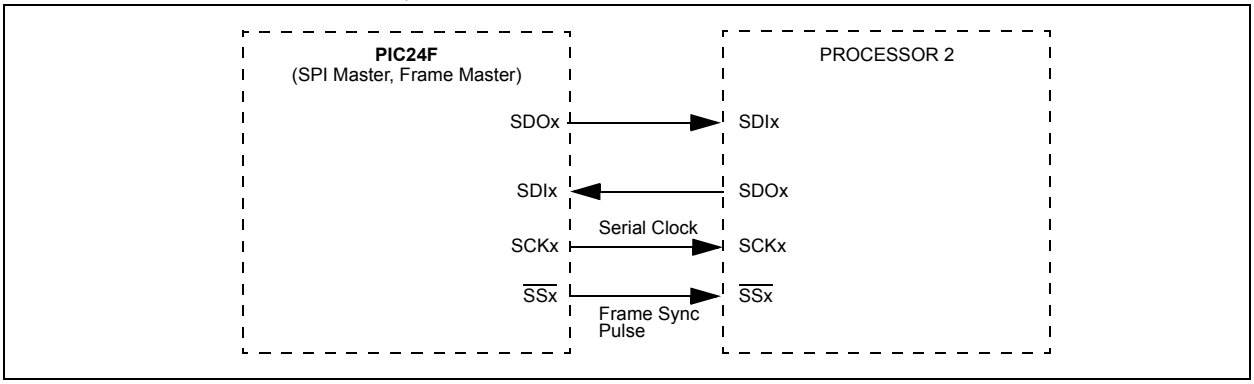


FIGURE 15-6: SPI MASTER, FRAME SLAVE CONNECTION DIAGRAM

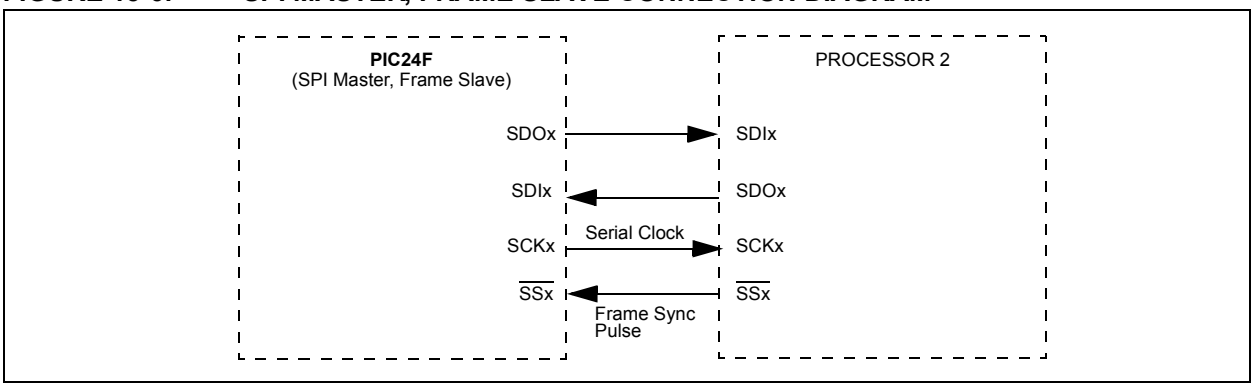


FIGURE 15-7: SPI SLAVE, FRAME MASTER CONNECTION DIAGRAM

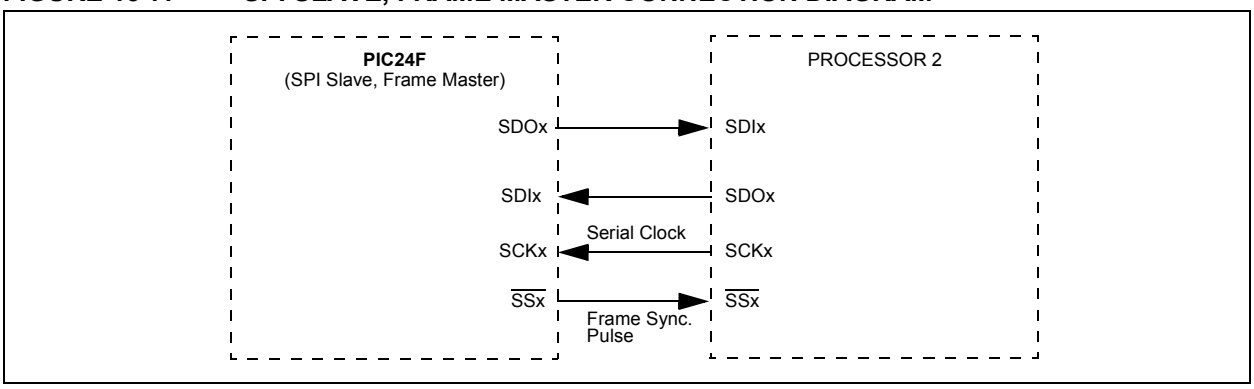
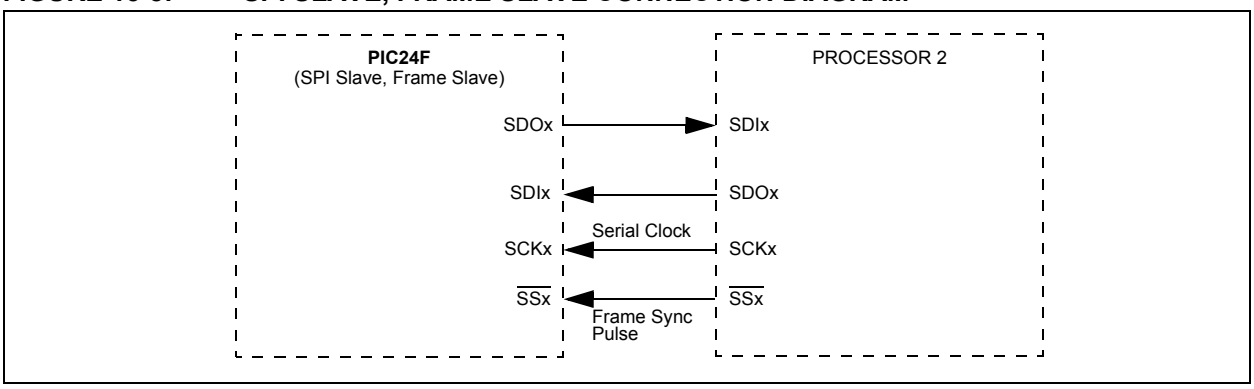


FIGURE 15-8: SPI SLAVE, FRAME SLAVE CONNECTION DIAGRAM



REGISTER 18-6: U1STAT: USB STATUS REGISTER

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

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	U-0	U-0
ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI(1)	—	—
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **ENDPT<3:0>:** Number of the Last Endpoint Activity bits
(Represents the number of the BDT updated by the last USB transfer).
1111 = Endpoint 15
1110 = Endpoint 14
....
0001 = Endpoint 1
0000 = Endpoint 0

bit 3 **DIR:** Last BD Direction Indicator bit
1 = The last transaction was a transmit transfer (Tx)
0 = The last transaction was a receive transfer (Rx)

bit 2 **PPBI:** Ping-Pong BD Pointer Indicator bit(1)
1 = The last transaction was to the ODD BD bank
0 = The last transaction was to the EVEN BD bank

bit 1-0 **Unimplemented:** Read as '0'

Note 1: This bit is only valid for endpoints with available EVEN and ODD BD registers.

18.7.2 USB INTERRUPT REGISTERS

REGISTER 18-14: U1OTGIR: USB OTG INTERRUPT STATUS REGISTER (HOST MODE ONLY)

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

R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	U-0	R/K-0, HS
IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **IDIF:** ID State Change Indicator bit
- 1 = Change in ID state detected
- 0 = No ID state change

bit 6 **T1MSECIF:** 1 Millisecond Timer bit
- 1 = The 1 millisecond timer has expired
- 0 = The 1 millisecond timer has not expired

bit 5 **LSTATEIF:** Line State Stable Indicator bit
- 1 = USB line state (as defined by the SE0 and JSTATE bits) has been stable for 1 ms, but different from last time
- 0 = USB line state has not been stable for 1 ms

bit 4 **ACTVIF:** Bus Activity Indicator bit
- 1 = Activity on the D+/D- lines or VBUS detected
- 0 = No activity on the D+/D- lines or VBUS detected

bit 3 **SESVDIF:** Session Valid Change Indicator bit
- 1 = VBUS has crossed VA_SESS_END (as defined in the USB OTG Specification)(1)
- 0 = VBUS has not crossed VA_SESS_END

bit 2 **SESENDIF:** B-Device VBUS Change Indicator bit
- 1 = VBUS change on B-device detected; VBUS has crossed VB_SESS_END (as defined in the USB OTG Specification)(1)
- 0 = VBUS has not crossed VA_SESS_END

bit 1 **Unimplemented:** Read as '0'

bit 0 **VBUSVDIF** A-Device VBUS Change Indicator bit
- 1 = VBUS change on A-device detected; VBUS has crossed VA_VBUS_VLD (as defined in the USB OTG Specification)(1)
- 0 = No VBUS change on A-device detected

Note 1: VBUS threshold crossings may be either rising or falling.

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

FIGURE 19-9: EXAMPLE OF AN 8-BIT MULTIPLEXED ADDRESS AND DATA APPLICATION

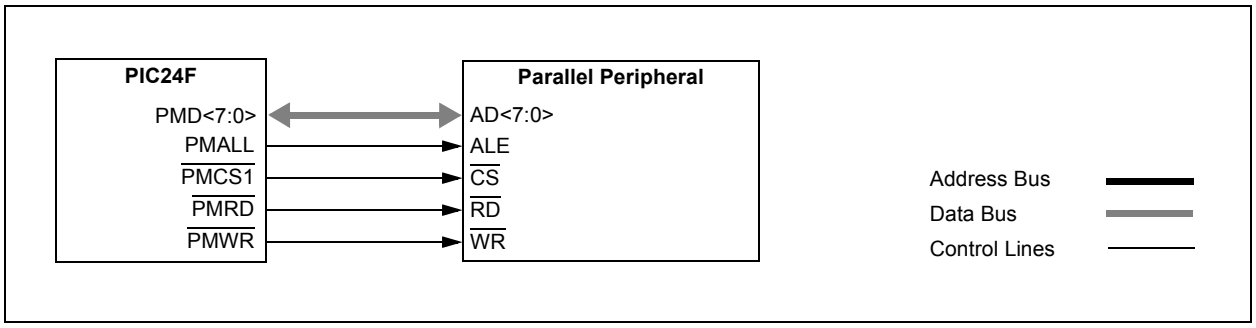


FIGURE 19-10: PARALLEL EEPROM EXAMPLE (UP TO 15-BIT ADDRESS, 8-BIT DATA)

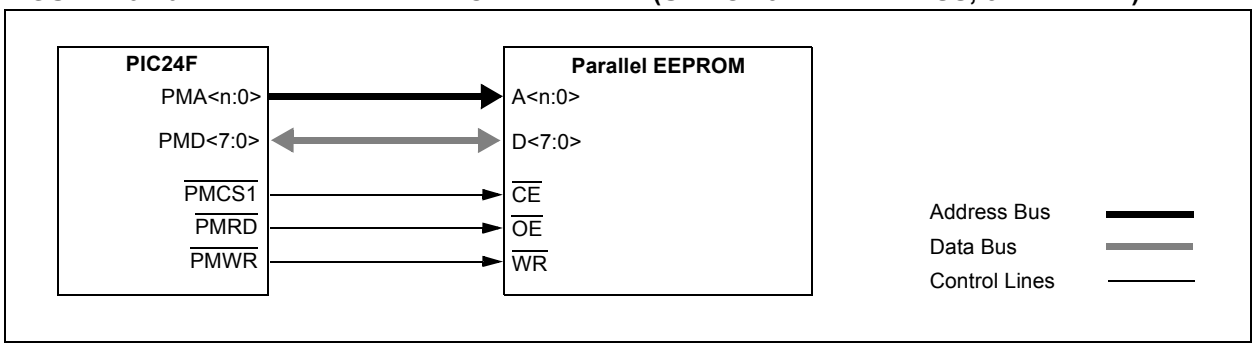


FIGURE 19-11: PARALLEL EEPROM EXAMPLE (UP TO 15-BIT ADDRESS, 16-BIT DATA)

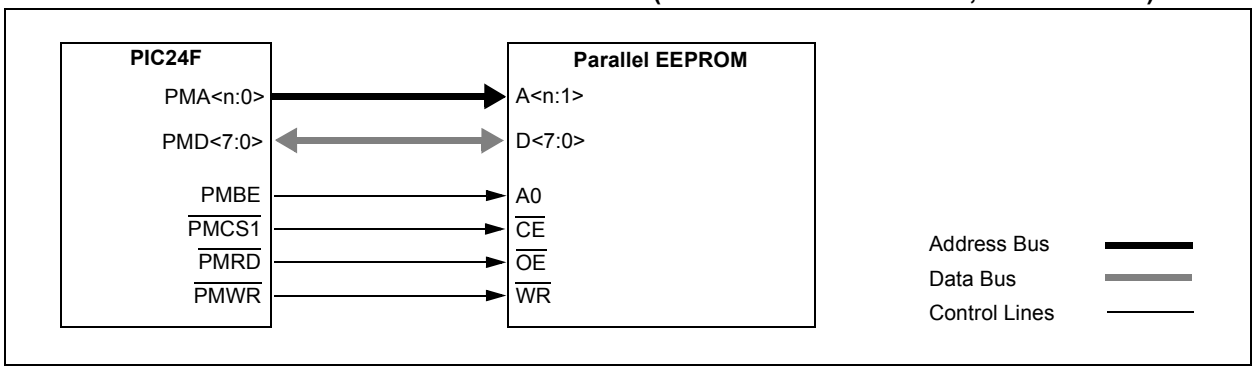
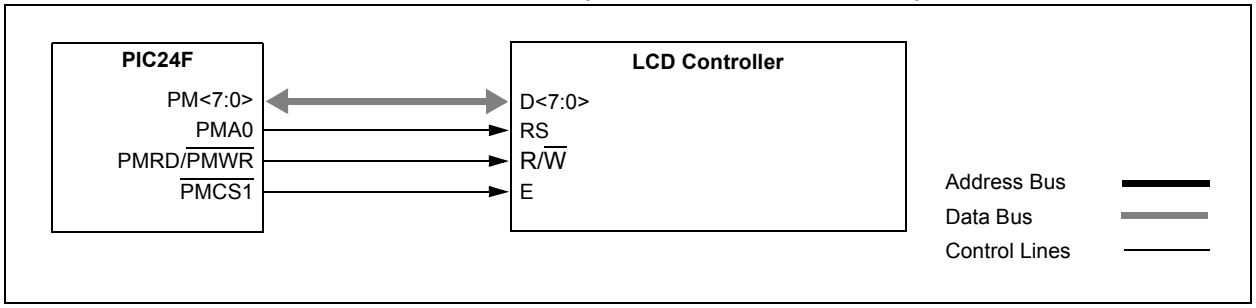


FIGURE 19-12: LCD CONTROL EXAMPLE (BYTE MODE OPERATION)



REGISTER 20-3: ALCFGRPT: ALARM CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
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
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
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 **ALRMEN:** Alarm Enable bit
1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)
0 = Alarm is disabled

bit 14 **CHIME:** Chime Enable bit
1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h

bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits
0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every 4 years)
101x = Reserved – do not use
11xx = Reserved – do not use

bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits
Points to the corresponding Alarm Value registers when reading ALRMVALH and ALRMVALL registers; the ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.
ALRMVAL<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented
ALRMVAL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits
11111111 = Alarm will repeat 255 more times
...
00000000 = Alarm will not repeat
The counter decrements on any alarm event. The counter is prevented from rolling over from 00h to FFh unless CHIME = 1.

20.1.5 ALRMVAL REGISTER MAPPINGS

REGISTER 20-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
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
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
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 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit bit
Contains a value of 0 or 1.

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

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

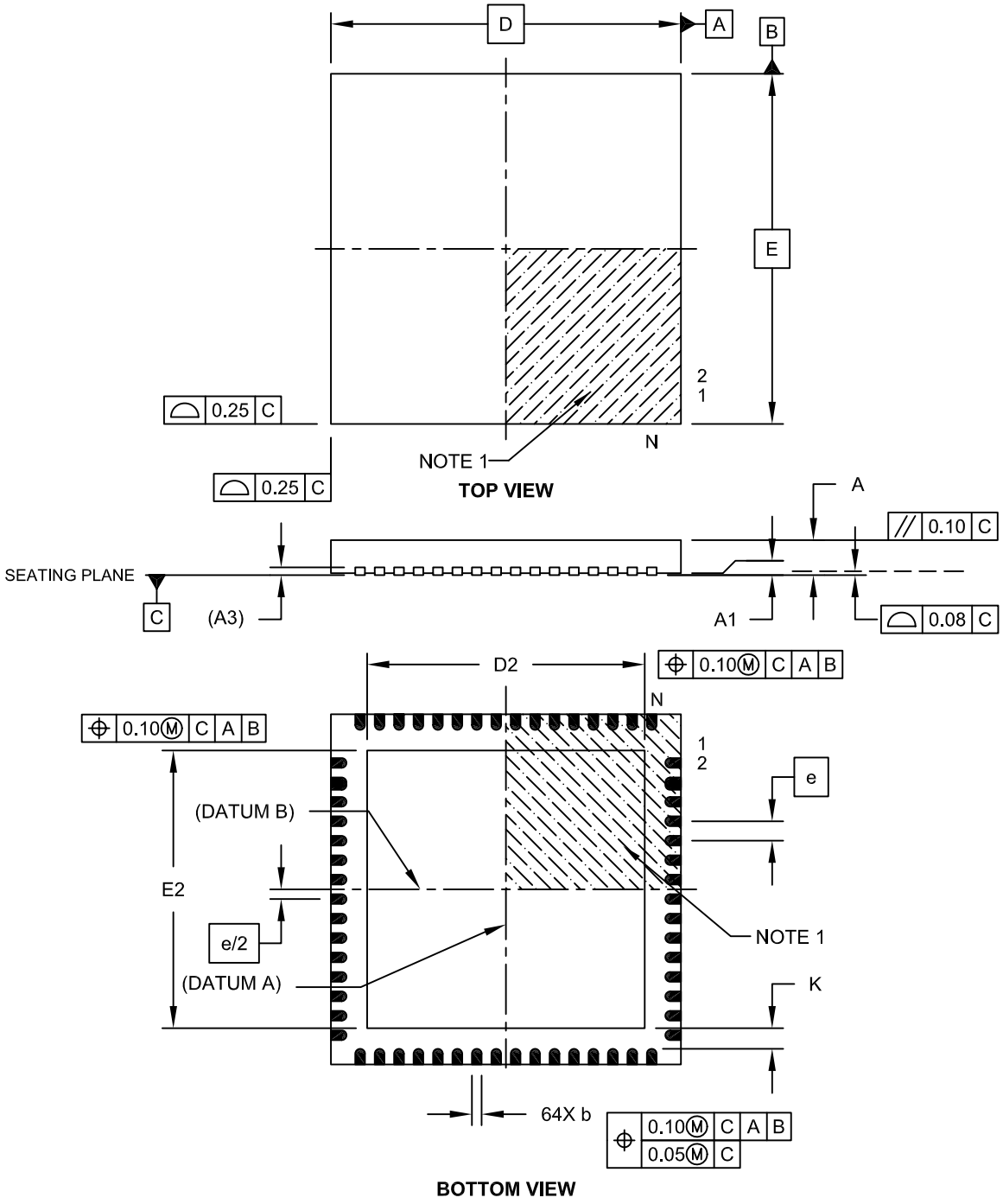
bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit bits
Contains a value from 0 to 3.

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

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

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]

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



Microchip Technology Drawing C04-149B Sheet 1 of 2