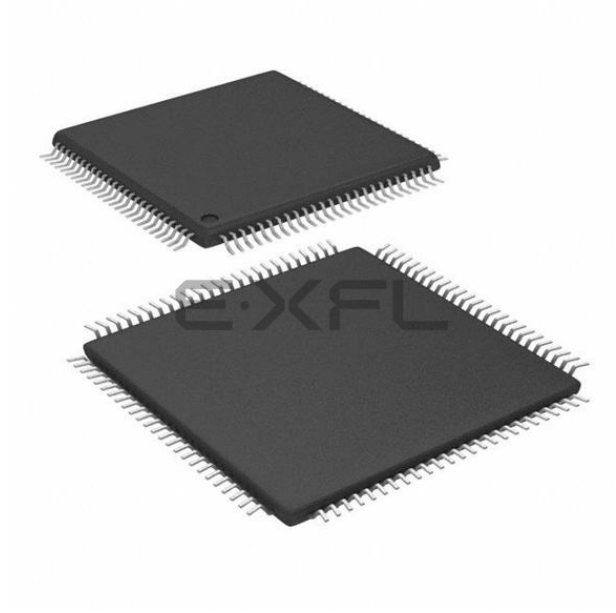
Microchip Technology - PIC24FJ256GB110T-I/PT Datasheet



Welcome to E-XFL.COM

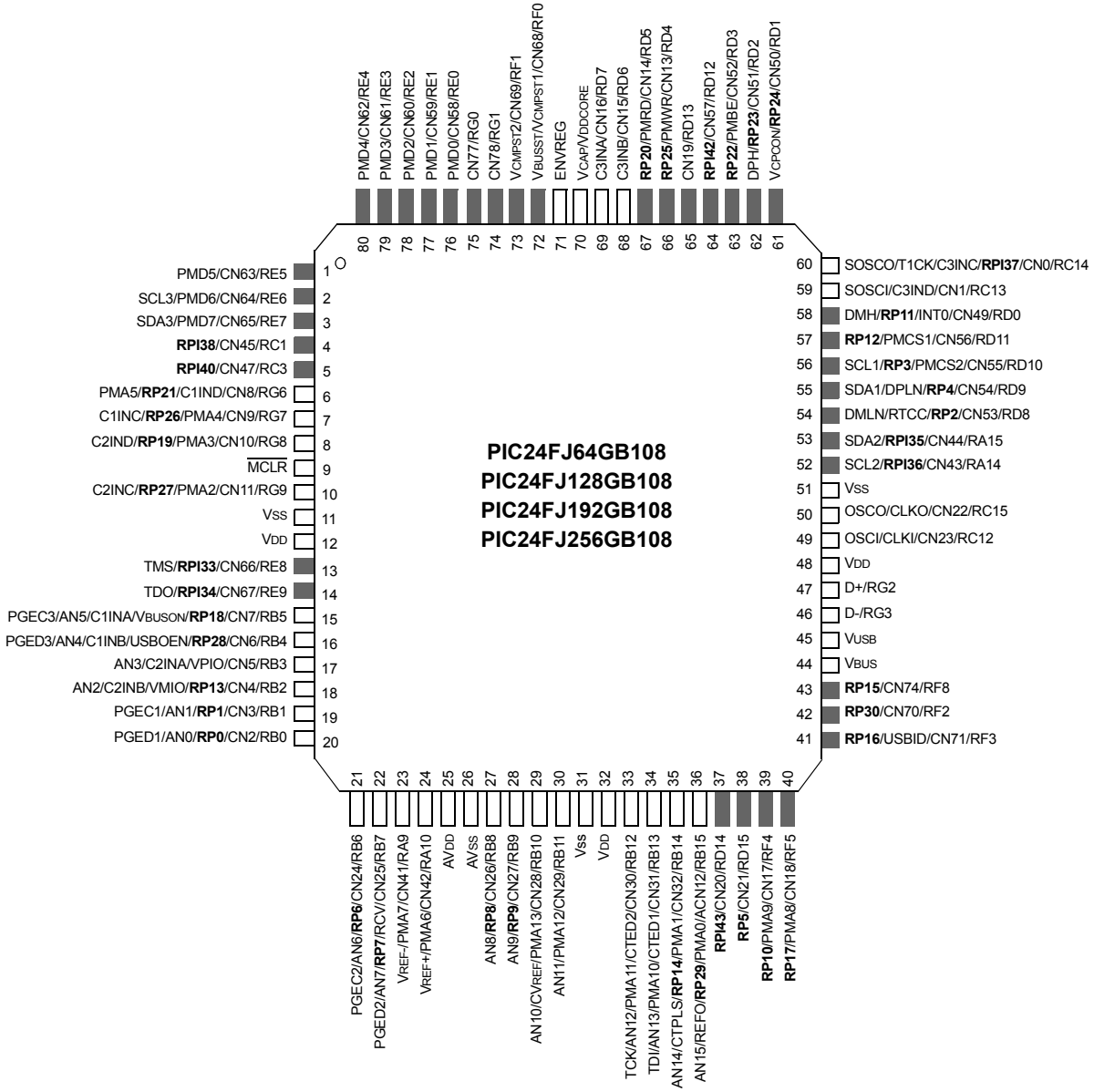
What is "Embedded - Microcontrollers"?

"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, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	83
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	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256gb110t-i-pt

Pin Diagram (80-Pin TQFP)



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

TABLE 4-3: CPU CORE REGISTERS 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
WREG0	0000	Working Register 0																0000
WREG1	0002	Working Register 1																0000
WREG2	0004	Working Register 2																0000
WREG3	0006	Working Register 3																0000
WREG4	0008	Working Register 4																0000
WREG5	000A	Working Register 5																0000
WREG6	000C	Working Register 6																0000
WREG7	000E	Working Register 7																0000
WREG8	0010	Working Register 8																0000
WREG9	0012	Working Register 9																0000
WREG10	0014	Working Register 10																0000
WREG11	0016	Working Register 11																0000
WREG12	0018	Working Register 12																0000
WREG13	001A	Working Register 13																0000
WREG14	001C	Working Register 14																0000
WREG15	001E	Working Register 15																0800
SPLIM	0020	Stack Pointer Limit Value Register																xxxx
PCL	002E	Program Counter Low Word Register																0000
PCH	0030	—	—	—	—	—	—	—	—	Program Counter Register High Byte								0000
TBLPAG	0032	—	—	—	—	—	—	—	—	Table Memory Page Address Register								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	Program Space Visibility Page Address Register								0000
RCOUNT	0036	Repeat Loop Counter Register																xxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	Disable Interrupts Counter Register														xxxx

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

TABLE 4-20: ADC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	ADC Data Buffer 0																xxxx
ADC1BUF1	0302	ADC Data Buffer 1																xxxx
ADC1BUF2	0304	ADC Data Buffer 2																xxxx
ADC1BUF3	0306	ADC Data Buffer 3																xxxx
ADC1BUF4	0308	ADC Data Buffer 4																xxxx
ADC1BUF5	030A	ADC Data Buffer 5																xxxx
ADC1BUF6	030C	ADC Data Buffer 6																xxxx
ADC1BUF7	030E	ADC Data Buffer 7																xxxx
ADC1BUF8	0310	ADC Data Buffer 8																xxxx
ADC1BUF9	0312	ADC Data Buffer 9																xxxx
ADC1BUFA	0314	ADC Data Buffer 10																xxxx
ADC1BUFB	0316	ADC Data Buffer 11																xxxx
ADC1BUFC	0318	ADC Data Buffer 12																xxxx
ADC1BUFD	031A	ADC Data Buffer 13																xxxx
ADC1BUFE	031C	ADC Data Buffer 14																xxxx
ADC1BUFF	031E	ADC Data Buffer 15																xxxx
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	—	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1PCFGH	032A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCFG17	PCFG16	0000
AD1PCFGL	032C	PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

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

TABLE 4-21: CTMU 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
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

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

TABLE 4-22: USB OTG 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
U1OTGIR	0480	—	—	—	—	—	—	—	—	IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF	0000
U1OTGIE	0482	—	—	—	—	—	—	—	—	IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE	0000
U1OTGSTAT	0484	—	—	—	—	—	—	—	—	ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD	0000
U1OTGCON	0486	—	—	—	—	—	—	—	—	DPPULUP	DMPULUP	DPPULDWN	DMPULDWN	VBUSON	OTGEN	VBUSCHG	VBUSDIS	0000
U1PWRC	0488	—	—	—	—	—	—	—	—	UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR	0000
U1IR	048A(1)	—	—	—	—	—	—	—	—	STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF	0000
		—	—	—	—	—	—	—	—	STALLIF	ATTACHIF(1)	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF(1)	0000
U1IE	048C(1)	—	—	—	—	—	—	—	—	STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE	0000
		—	—	—	—	—	—	—	—	STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	DETACHIE(1)	0000
U1EIR	048E(1)	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0000
		—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	EOFEF(1)	PIDEF	0000
U1EIE	0490(1)	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0000
		—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	EOFEE(1)	PIDEE	0000
U1STAT	0492	—	—	—	—	—	—	—	—	ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI	—	—	0000
U1CON	0494(1)	—	—	—	—	—	—	—	—	—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN	0000
		—	—	—	—	—	—	—	—	JSTATE(1)	SE0	TOKBUSY	RESET	HOSTEN	RESUME	PPBRST	SOFEN(1)	0000
U1ADDR	0496	—	—	—	—	—	—	—	—	LSPDEN(1)	USB Device Address (DEVADDR) Register							0000
U1BDTP1	0498	—	—	—	—	—	—	—	—	Buffer Descriptor Table Base Address Register							—	0000
U1FRML	049A	—	—	—	—	—	—	—	—	Frame Count Register Low Byte								0000
U1FRMH	049C	—	—	—	—	—	—	—	—	Frame Count Register High Byte								0000
U1TOK(2)	049E	—	—	—	—	—	—	—	—	PID3	PID2	PID1	PID0	EP3	EP2	EP1	EP0	0000
U1SOF(2)	04A0	—	—	—	—	—	—	—	—	Start-Of-Frame Count Register								0000
U1CNFG1	04A6	—	—	—	—	—	—	—	—	UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0	0000
U1CNFG2	04A8	—	—	—	—	—	—	—	—	—	—	—	PUVBUS	EXTI2CEN	UVBUSDIS	UVCMPDIS	UTRDIS	0000

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

Note 1: Alternate register or bit definitions when the module is operating in Host mode.

2: This register is available in Host mode only.

4.3.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word page of the program space. This provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., TBLRDL/H).

Program space access through the data space occurs if the Most Significant bit of the data space EA is '1', and program space visibility is enabled by setting the PSV bit in the CPU Control register (CORCON<2>). The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page Address register (PSVPAG). This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with the 15 bits of the EA functioning as the lower bits. Note that by incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads to this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-7), only the lower 16 bits of the 24-bit program word are used to contain the data. The upper 8 bits of any program space locations used as data should be programmed with '1111 1111' or '0000 0000' to force a NOP. This prevents possible issues should the area of code ever be accidentally executed.

Note:	PSV access is temporarily disabled during table reads/writes.

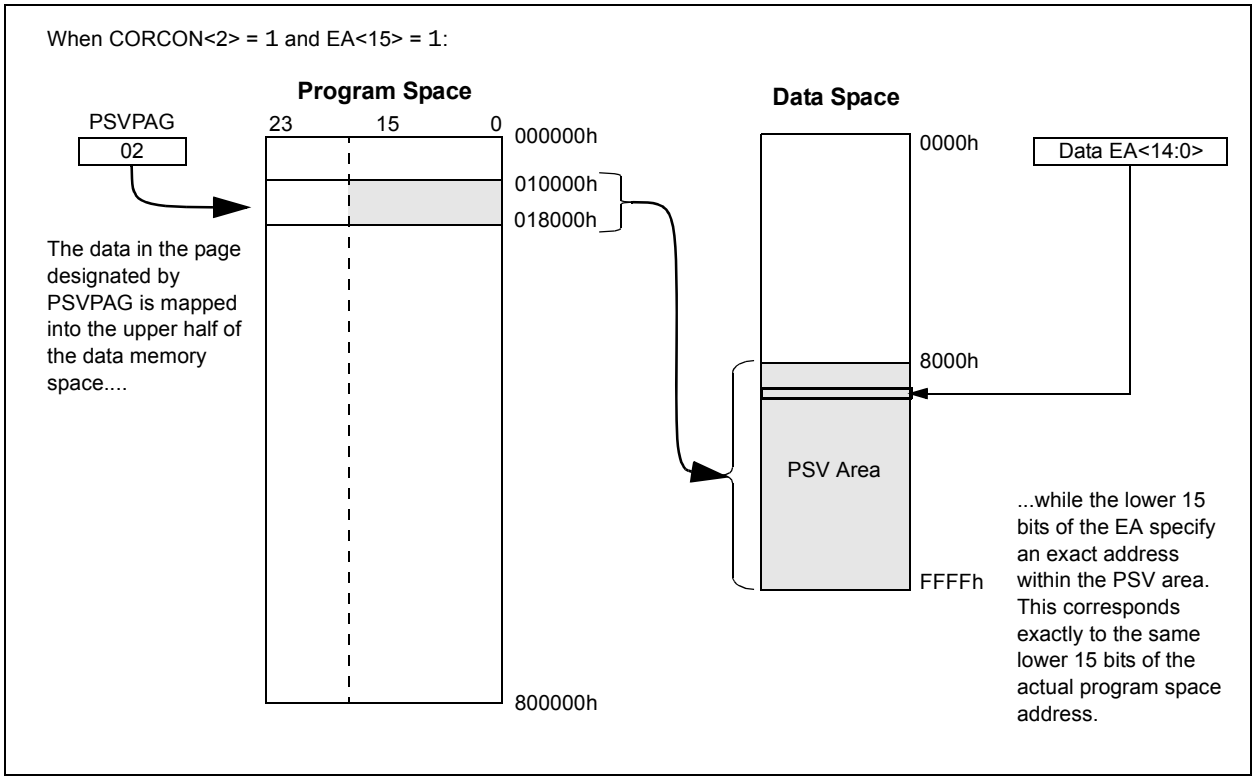
For operations that use PSV and are executed outside a REPEAT loop, the MOV and MOV.D instructions will require one instruction cycle in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

For operations that use PSV which are executed inside a REPEAT loop, there will be some instances that require two instruction cycles in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction accessing data, using PSV, to execute in a single cycle.

FIGURE 4-7: PROGRAM SPACE VISIBILITY OPERATION



REGISTER 7-17: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
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
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 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 **IC1IP<2:0>:** Input Capture Channel 1 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 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-25: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

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

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

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

bit 3 **Unimplemented:** Read as ‘0’

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

8.5 Oscillator Modes and USB Operation

Because of the timing requirements imposed by USB, an internal clock of 48 MHz is required at all times while the USB module is enabled. Since this is well beyond the maximum CPU clock speed, a method is provided to internally generate both the USB and system clocks from a single oscillator source. PIC24FJ256GB110 family devices use the same clock structure as other PIC24FJ devices, but include a two-branch PLL system to generate the two clock signals.

The USB PLL block is shown in Figure 8-2. In this system, the input from the Primary Oscillator is divided down by a PLL prescaler to generate a 4 MHz output. This is used to drive an on-chip 96 MHz PLL frequency multiplier to drive the two clock branches. One branch uses a fixed divide-by-2 frequency divider to generate the 48 MHz USB clock. The other branch uses a fixed divide-by-3 frequency divider and configurable PLL prescaler/divider to generate a range of system clock frequencies. The CPDIV bits select the system clock speed; available clock options are listed in Table 8-2.

The USB PLL prescaler does not automatically sense the incoming oscillator frequency. The user must manually configure the PLL divider to generate the required 4 MHz output, using the PLLDIV<2:0> Configuration bits. This limits the choices for Primary Oscillator frequency to a total of 8 possibilities, shown in Table 8-3.

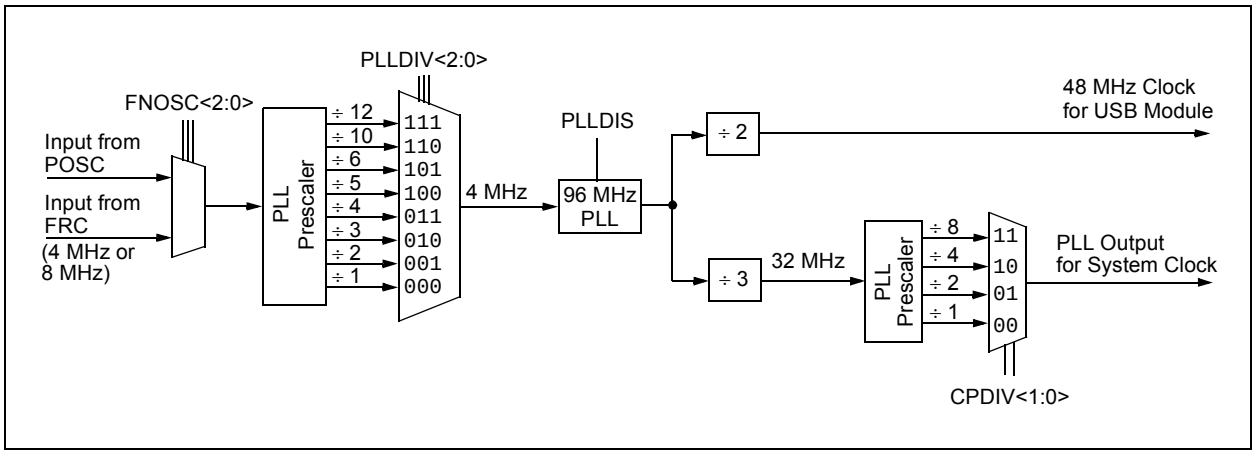
TABLE 8-2: SYSTEM CLOCK OPTIONS DURING USB OPERATION

MCU Clock Division (CPDIV<1:0>)	Microcontroller Clock Frequency
None (00)	32 MHz
÷2 (01)	16 MHz
÷4 (10)	8 MHz
÷8 (11)	4 MHz

TABLE 8-3: VALID PRIMARY OSCILLATOR CONFIGURATIONS FOR USB OPERATIONS

Input Oscillator Frequency	Clock Mode	PLL Division (PLLDIV<2:0>)
48 MHz	ECPLL	÷12 (111)
40 MHz	ECPLL	÷10 (110)
24 MHz	HSPLL, ECPLL	÷6 (101)
20 MHz	HSPLL, ECPLL	÷5 (100)
16 MHz	HSPLL, ECPLL	÷4 (011)
12 MHz	HSPLL, ECPLL	÷3 (010)
8 MHz	ECPLL, XTPLL	÷2 (001)
4 MHz	ECPLL, XTPLL	÷1 (000)

FIGURE 8-2: USB PLL BLOCK



REGISTER 12-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER(3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(1)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1,2)	—
bit 7							bit 0

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

bit 15 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

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

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

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit(1,2)
1 = External clock from pin TyCK (on the rising edge)
0 = Internal clock (FOSC/2)

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

Note 1: When 32-bit operation is enabled (T2CON<3> or T4CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through T2CON and T4CON.

2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select"** for more information.

3: Changing the value of TyCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

13.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own 16-bit timer. To increase resolution, 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 (ICx) provides the Least Significant 16 bits of the 32-bit register pairs, and the even module (ICy) provides the Most Significant 16 bits. Wraparounds of the ICx registers cause an increment of their corresponding ICy registers.

Cascaded operation is configured in hardware by setting the IC32 bits (ICxCON2<8>) for both modules.

13.2 Capture Operations

The input capture module can be configured to capture timer values and generate interrupts on rising edges on ICx, or all transitions on ICx. Captures can be configured to occur on all rising edges, or just some (every 4th or 16th). Interrupts can be independently configured to generate on each event, or a subset of events.

To set up the module for capture operations:

1. Configure the ICx input for one of the available Peripheral Pin Select pins.
2. If Synchronous mode is to be used, disable the sync source before proceeding.
3. Make sure that any previous data has been removed from the FIFO by reading ICxBUF until the ICBNE bit (ICxCON1<3>) is cleared.
4. Set the SYNCSEL bits (ICxCON2<4:0>) to the desired sync/trigger source.
5. Set the ICTSEL bits (ICxCON1<12:10>) for the desired clock source.
6. Set the ICI bits (ICxCON1<6:5>) to the desired interrupt frequency
7. Select Synchronous or Trigger mode operation:
   a) Check that the SYNCSEL bits are not set to '00000'.
   b) For Synchronous mode, clear the ICTRIG bit (ICxCON2<7>).
   c) For Trigger mode, set ICTRIG, and clear the TRIGSTAT bit (ICxCON2<6>).
8. Set the ICM bits (ICxCON1<2:0>) to the desired operational mode.
9. Enable the selected trigger/sync source.

For 32-bit cascaded operations, the setup procedure is slightly different:

1. Set the IC32 bits for both modules (ICyCON2<8> and (ICxCON2<8>), enabling the even numbered module first. This ensures the modules will start functioning in unison.
2. Set the ICTSEL and SYNCSEL bits for both modules to select the same sync/trigger and time base source. Set the even module first, then the odd module. Both modules must use the same ICTSEL and SYNCSEL settings.
3. Clear the ICTRIG bit of the even module (ICyCON2<7>); this forces the module to run in Synchronous mode with the odd module, regardless of its trigger setting.
4. Use the odd module's ICI bits (ICxCON1<6:5>) to the desired interrupt frequency.
5. Use the ICTRIG bit of the odd module (ICxCON2<7>) to configure Trigger or Synchronous mode operation.

Note:	For Synchronous mode operation, enable the sync source as the last step. Both input capture modules are held in Reset until the sync source is enabled.

6. Use the ICM bits of the odd module (ICxCON1<2:0>) to set the desired capture mode.

The module is ready to capture events when the time base and the trigger/sync source are enabled. When the ICBNE bit (ICxCON1<3>) becomes set, at least one capture value is available in the FIFO. Read input capture values from the FIFO until the ICBNE clears to '0'.

For 32-bit operation, read both the ICxBUF and ICyBUF for the full 32-bit timer value (ICxBUF for the lsw, ICyBUF for the msw). At least one capture value is available in the FIFO buffer when the odd module's ICBNE bit (ICxCON1<3>) becomes set. Continue to read the buffer registers until ICBNE is cleared (perform automatically by hardware).

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 Compare with Dedicated Timers"** (DS39723).

Devices in the PIC24FJ256GB110 family all feature 9 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 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 30 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 resolution, 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 module (OCy) provides the Most Significant 16 bits. Wraparounds of the OCx registers cause an increment of their corresponding OCy registers.

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

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

bit 15 **I2CEN:** I2Cx Enable bit
1 = Enables the I2Cx module and configures the SDAx and SCLx pins as serial port pins
0 = Disables I2Cx module. All I^2C pins are controlled by port functions.

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

bit 13 **I2CSIDL:** Stop in Idle Mode bit
1 = Discontinues module operation when device enters an Idle mode
0 = Continues module operation in Idle mode

bit 12 **SCLREL:** SCLx Release Control bit (when operating as I^2C Slave)
1 = Releases SCLx clock
0 = Holds SCLx clock low (clock stretch)
If STREN = 1:
Bit is R/$\overline{W}$ (i.e., software may write '0' to initiate stretch and write '1' to release clock).
Hardware clear at beginning of slave transmission.
Hardware clear at end of slave reception.
If STREN = 0:
Bit is R/S (i.e., software may only write '1' to release clock).
Hardware clear at beginning of slave transmission.

bit 11 **IPMIEN:** Intelligent Platform Management Interface (IPMI) Enable bit
1 = IPMI Support mode is enabled; all addresses Acknowledged
0 = IPMI mode disabled

bit 10 **A10M:** 10-Bit Slave Addressing bit
1 = I2CxADD is a 10-bit slave address
0 = I2CxADD is a 7-bit slave address

bit 9 **DISSLW:** Disable Slew Rate Control bit
1 = Slew rate control disabled
0 = Slew rate control enabled

bit 8 **SMEN:** SMBus Input Levels bit
1 = Enables I/O pin thresholds compliant with SMBus specification
0 = Disables SMBus input thresholds

bit 7 **GCEN:** General Call Enable bit (when operating as I^2C slave)
1 = Enables interrupt when a general call address is received in the I2CxRSR
(module is enabled for reception)
0 = General call address disabled

bit 6 **STREN:** SCLx Clock Stretch Enable bit (when operating as I^2C slave)
Used in conjunction with SCLREL bit.
1 = Enables software or receive clock stretching
0 = Disables software or receive clock stretching

18.5.2 COMPLETE A CONTROL TRANSACTION TO A CONNECTED DEVICE

1. Follow the procedure described in **Section 18.5.1 "Enable Host Mode and Discover a Connected Device"** to discover a device.
2. Set up the Endpoint Control register for bidirectional control transfers by writing 0Dh to U1EP0 (this sets the EPCONDIS, EPTXEN, and EPHSHK bits).
3. Place a copy of the device framework setup command in a memory buffer. See Chapter 9 of the USB 2.0 specification for information on the device framework command set.
4. Initialize the buffer descriptor (BD) for the current (EVEN or ODD) Tx EP0, to transfer the eight bytes of command data for a device framework command (i.e., a GET DEVICE DESCRIPTOR):
   a) Set the BD data buffer address (BD0ADR) to the starting address of the 8-byte memory buffer containing the command.
   b) Write 8008h to BD0STAT (this sets the UOWN bit, and sets a byte count of 8).
5. Set the USB device address of the target device in the address register (U1ADDR<6:0>). After a USB bus Reset, the device USB address will be zero. After enumeration, it will be set to another value between 1 and 127.
6. Write D0h to U1TOK; this is a SETUP token to Endpoint 0, the target device's default control pipe. This initiates a SETUP token on the bus, followed by a data packet. The device handshake is returned in the PID field of BD0STAT after the packets are complete. When the USB module updates BD0STAT, a transfer done interrupt is asserted (the TRNIF flag is set). This completes the setup phase of the setup transaction as referenced in chapter 9 of the USB specification.
7. To initiate the data phase of the setup transaction (i.e., get the data for the GET DEVICE descriptor command), set up a buffer in memory to store the received data.
8. Initialize the current (EVEN or ODD) Rx or Tx (Rx for IN, Tx for OUT) EP0 BD to transfer the data.
   a) Write C040h to BD0STAT. This sets the UOWN, configures Data Toggle (DTS) to DATA1, and sets the byte count to the length of the data buffer (64 or 40h, in this case).
   b) Set BD0ADR to the starting address of the data buffer.
9. Write the token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 90h to U1TOK for an IN token for a GET DEVICE DESCRIPTOR command). This initiates an IN token on the bus followed by a data packet from the device to the host. When the data packet completes, the BD0STAT is written and a transfer done interrupt is asserted (the TRNIF flag is set). For control transfers with a single packet data phase, this completes the data phase of the setup transaction as referenced in chapter 9 of the USB specification. If more data needs to be transferred, return to step 8.
10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (even or odd) Tx EP0 BD to transfer the status data.:
    a) Set the BDT buffer address field to the start address of the data buffer
    b) Write 8000h to BD0STAT (set UOWN bit, configure DTS to DATA0, and set byte count to 0).
12. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 01h to U1TOK for an OUT token for a GET DEVICE DESCRIPTOR command). This initiates an OUT token on the bus followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device, and a transfer done interrupt is asserted (the TRNIF flag is set). This completes the status phase of the setup transaction as described in Chapter 9 of the USB specification.

Note:	Only one control transaction can be performed per frame.

18.5.3 SEND A FULL-SPEED BULK DATA TRANSFER TO A TARGET DEVICE

1. Follow the procedure described in **Section 18.5.1 "Enable Host Mode and Discover a Connected Device"** and **Section 18.5.2 "Complete a Control Transaction to a Connected Device"** to discover and configure a device.
2. To enable transmit and receive transfers with handshaking enabled, write 1Dh to U1EP0. If the target device is a low-speed device, also set the LSPD bit (U1EP0<7>). If you want the hardware to automatically retry indefinitely if the target device asserts a NAK on the transfer, clear the Retry Disable bit, RETRYDIS (U1EP0<6>).
3. Set up the BD for the current (EVEN or ODD) Tx EP0 to transfer up to 64 bytes.
4. Set the USB device address of the target device in the address register (U1ADDR<6:0>).
5. Write an OUT token to the desired endpoint to U1TOK. This triggers the module's transmit state machines to begin transmitting the token and the data.
6. Wait for the Transfer Done Interrupt Flag, TRNIF. This indicates that the BD has been released back to the microprocessor, and the transfer has completed. If the retry disable bit is set, the handshake (ACK, NAK, STALL or ERROR (0Fh)) is returned in the BD PID field. If a STALL interrupt occurs, the pending packet must be dequeued and the error condition in the target device cleared. If a detach interrupt occurs (SE0 for more than 2.5 µs), then the target has detached (U1IR<0> is set).
7. Once the transfer done interrupt occurs (TRNIF is set), the BD can be examined and the next data packet queued by returning to step 2.

Note:	USB speed, transceiver and pull-ups should only be configured during the module setup phase. It is not recommended to change these settings while the module is enabled.

18.6 OTG Operation

18.6.1 SESSION REQUEST PROTOCOL (SRP)

An OTG A-device may decide to power down the VBUS supply when it is not using the USB link through the Session Request Protocol (SRP). Software may do this by clearing VBUSON (U1OTGCON<3>). When the VBUS supply is powered down, the A-device is said to have ended a USB session.

An OTG A-device or Embedded Host may repower the VBUS supply at any time (initiate a new session). An OTG B-device may also request that the OTG A-device repower the VBUS supply (initiate a new session). This is accomplished via Session Request Protocol (SRP).

Prior to requesting a new session, the B-device must first check that the previous session has definitely ended. To do this, the B-device must check for two conditions:

1. VBUS supply is below the Session Valid voltage and

2. Both D+ and D- have been low for at least 2 ms.

The B-device will be notified of condition 1 by the SESENDIF (U1OTGIR<2>) interrupt. Software will have to manually check for condition 2.

Note:	When the A-device powers down the VBUS supply, the B-device must disconnect its pull-up resistor from power. If the device is self-powered, it can do this by clearing DPPULUP (U1OTGCON<7>) and DMPULUP (U1OTGCON<6>).

The B-device may aid in achieving condition 1 by discharging the VBUS supply through a resistor. Software may do this by setting VBUSDIS (U1OTGCON<0>).

After these initial conditions are met, the B-device may begin requesting the new session. The B-device begins by pulsing the D+ data line. Software should do this by setting DPPULUP (U1OTGCON<7>). The data line should be held high for 5 to 10 ms.

The B-device then proceeds by pulsing the VBUS supply. Software should do this by setting PUVBUS (U1CNFG2<4>). When an A-device detects SRP signaling (either via the ATTACHIF (U1IR<6>) interrupt or via the SESVDIF (U1OTGIR<3>) interrupt), the A-device must restore the VBUS supply by either setting VBUSON (U1OTGCON<3>), or by setting the I/O port controlling the external power source.

The B-device should not monitor the state of the VBUS supply while performing VBUS supply pulsing. When the B-device does detect that the VBUS supply has been restored (via the SESVDIF (U1OTGIR<3>) interrupt), the B-device must re-connect to the USB link by pulling up D+ or D- (via the DPPULUP or DMPULUP).

The A-device must complete the SRP by driving USB Reset signaling.

REGISTER 18-5: U1PWRC: USB POWER CONTROL REGISTER

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

R/W-0, HS	U-0	U-0	R/W-0	U-0	U-0	R/W-0, HC	R/W-0
UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR
bit 7							bit 0

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

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

bit 7 **UACTPND:** USB Activity Pending bit
1 = Module should not be suspended at the moment (requires USLPGRD bit to be set)
0 = Module may be suspended or powered down

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **USLPGRD:** Sleep/Suspend Guard bit
1 = Indicate to the USB module that it is about to be suspended or powered down
0 = No suspend

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **USUSPND:** USB Suspend Mode Enable bit
1 = USB OTG module is in Suspend mode; USB clock is gated and the transceiver is placed in a low-power state
0 = Normal USB OTG operation

bit 0 **USBPWR:** USB Operation Enable bit
1 = USB OTG module is enabled
0 = USB OTG module is disabled(1)

Note 1: Do not clear this bit unless the HOSTEN, USBEN and OTGEN bits (U1CON<3,0> and U1OTGCON<2>) are all cleared.

REGISTER 18-17: U1IR: USB 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	R-0	R/K-0, HS
STALLIF	ATTACHIF	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF
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 **STALLIF:** STALL Handshake Interrupt bit
- 1 = A STALL handshake was sent by the peripheral device during the handshake phase of the transaction in Device mode
- 0 = A STALL handshake has not been sent

bit 6 **ATTACHIF:** Peripheral Attach Interrupt bit
- 1 = A peripheral attachment has been detected by the module; set if the bus state is not SE0 and there has been no bus activity for 2.5 μs
- 0 = No peripheral attachement detected

bit 5 **RESUMEIF:** Resume Interrupt bit
- 1 = A K-state is observed on the D+ or D- pin for 2.5 μs (differential '1' for low speed, differential '0' for full speed)
- 0 = No K-state observed

bit 4 **IDLEIF:** Idle Detect Interrupt bit
- 1 = Idle condition detected (constant Idle state of 3 ms or more)
- 0 = No Idle condition detected

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of current token is complete; read U1STAT register for endpoint information
- 0 = Processing of current token not complete; clear U1STAT register or load next token from U1STAT

bit 2 **SOFIF:** Start-Of-Frame Token Interrupt bit
- 1 = A Start-Of-Frame token received by the peripheral or the Start-Of-Frame threshold reached by the host
- 0 = No Start-Of-Frame token received or threshold reached

bit 1 **UERRIF:** USB Error Condition Interrupt bit
- 1 = An unmasked error condition has occurred; only error states enabled in the U1EIE register can set this bit
- 0 = No unmasked error condition has occurred

bit 0 **DETACHIF:** Detach Interrupt bit
- 1 = A peripheral detachment has been detected by the module; Reset state must be cleared before this bit can be reasserted
- 0 = No peripheral detachment detected. 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 bit-wise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

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 19-5: MASTER MODE, PARTIALLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, TWO CHIP SELECTS)

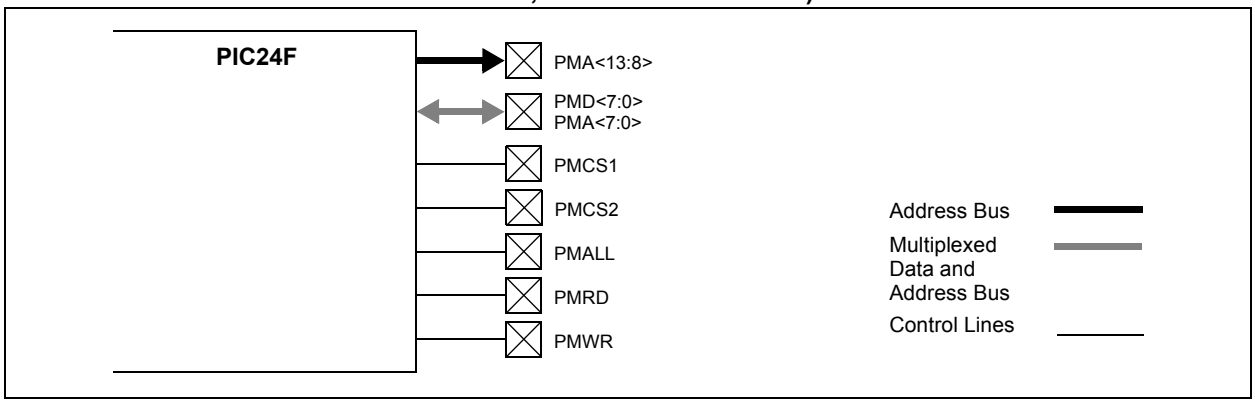


FIGURE 19-6: MASTER MODE, FULLY MULTIPLEXED ADDRESSING (SEPARATE READ AND WRITE STROBES, TWO CHIP SELECTS)

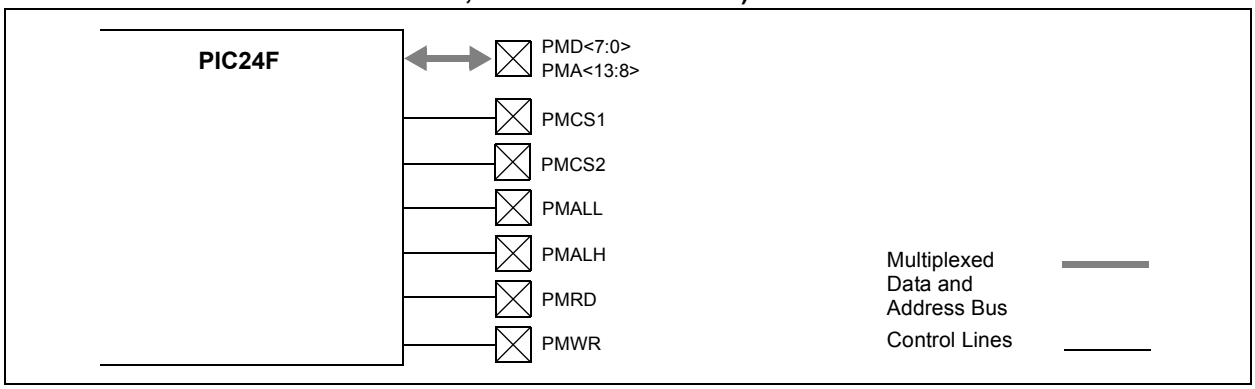


FIGURE 19-7: EXAMPLE OF A MULTIPLEXED ADDRESSING APPLICATION

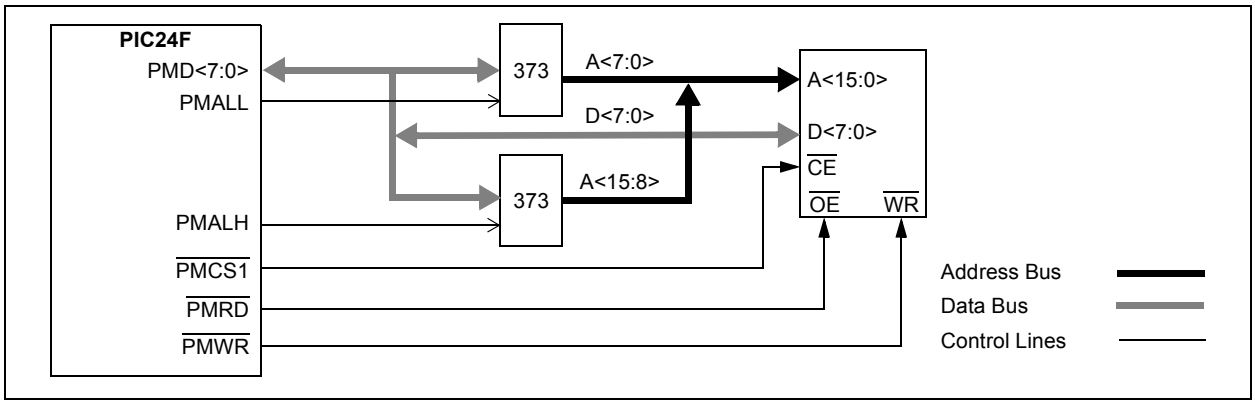
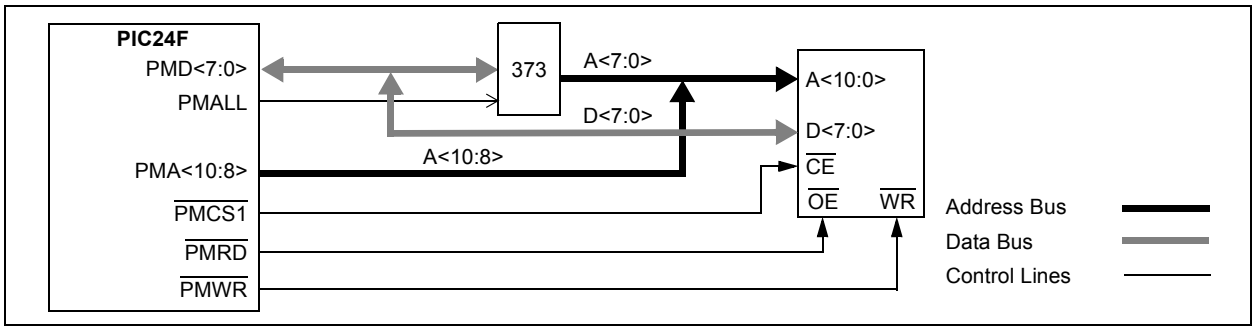


FIGURE 19-8: EXAMPLE OF A PARTIALLY MULTIPLEXED ADDRESSING APPLICATION



REGISTER 26-1: CW1: FLASH CONFIGURATION WORD 1

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
r	JTAGEN(1)	GCP	GWRP	$\overline{\text{DEBUG}}$	r	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	—	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:	r = Reserved bit	
R = Readable bit	PO = Program Once bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed	'1' = Bit is set	'0' = Bit is cleared

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **Reserved:** The value is unknown; program as '0'

bit 14 **JTAGEN:** JTAG Port Enable bit(1)
1 = JTAG port is enabled
0 = JTAG port is disabled

bit 13 **GCP:** General Segment Program Memory Code Protection bit
1 = Code protection is disabled
0 = Code protection is enabled for the entire program memory space

bit 12 **GWRP:** General Segment Code Flash Write Protection bit
1 = Writes to program memory are allowed
0 = Writes to program memory are disabled

bit 11 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Device resets into Operational mode
0 = Device resets into Debug mode

bit 10 **Reserved:** Always maintain as '1'

bit 9-8 **ICS1:ICS0:** Emulator Pin Placement Select bits
11 = Emulator functions are shared with PGEC1/PGED1
10 = Emulator functions are shared with PGEC2/PGED2
01 = Emulator functions are shared with PGEC3/PGED3
00 = Reserved; do not use

bit 7 **FWDTEN:** Watchdog Timer Enable bit
1 = Watchdog Timer is enabled
0 = Watchdog Timer is disabled

bit 6 **WINDIS:** Windowed Watchdog Timer Disable bit
1 = Standard Watchdog Timer enabled
0 = Windowed Watchdog Timer enabled; FWDTEN must be '1'

bit 5 **Unimplemented:** Read as '1'

bit 4 **FWPSA:** WDT Prescaler Ratio Select bit
1 = Prescaler ratio of 1:128
0 = Prescaler ratio of 1:32

Note 1: The JTAGEN bit can only be modified using In-Circuit Serial Programming™ (ICSP™). It cannot be modified while programming the device through the JTAG interface.

The size and type of protection for the segmented code range are configured by the WPFPx, WPEND, WPCFG and WPDIS bits in Configuration Word 3. Code segment protection is enabled by programming the WPDIS bit (= 0). The WPFP bits specify the size of the segment to be protected, by specifying the 512-word code page that is the start or end of the protected segment. The specified region is inclusive, therefore, this page will also be protected.

The WPEND bit determines if the protected segment uses the top or bottom of the program space as a boundary. Programming WPEND (= 0) sets the bottom of program memory (000000h) as the lower boundary of the protected segment. Leaving WPEND unprogrammed (= 1) protects the specified page through the last page of implemented program memory, including the Configuration Word locations.

A separate bit, WPCFG, is used to independently protect the last page of program space, including the Flash Configuration Words. Programming WPCFG (= 0) protects the last page regardless of the other bit settings. This may be useful in circumstances where write protection is needed for both a code segment in the bottom of memory, as well as the Flash Configuration Words.

The various options for segment code protection are shown in Table 26-2.

26.4.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against inadvertent or unwanted changes or reads in two ways. The primary protection method is the same as that of the RP registers – shadow registers contain a complimentary value which is constantly compared with the actual value.

To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset.

The data for the Configuration registers is derived from the Flash Configuration Words in program memory. When the GCP bit is set, the source data for device configuration is also protected as a consequence. Even if General Segment protection is not enabled, the device configuration can be protected by using the appropriate code cement protection setting.

TABLE 26-2: SEGMENT CODE PROTECTION CONFIGURATION OPTIONS

Segment Configuration Bits			Write/Erase Protection of Code Segment
WPDIS	WPEND	WPCFG	
1	X	x	No additional protection enabled; all program memory protection configured by GCP and GWRP
0	1	x	Addresses from first address of code page defined by WPFP<7:0> through end of implemented program memory (inclusive) write/erase protected, including Flash Configuration Words
0	0	1	Address 000000h through last address of code page defined by WPFP<7:0> (inclusive) write/erase protected
0	0	0	Address 000000h through last address of code page defined by WPFP<7:0> (inclusive) write/erase protected, and the last page is also write/erase protected.

APPENDIX A: REVISION HISTORY

Revision A (October 2007)

Original data sheet for the PIC24FJ256GB110 family of devices.

Revision B (March 2008)

Changes to **Section 29.0 "Electrical Characteristics"** and minor edits to text throughout document.

Revision C (December 2009)

Updates all Pin Diagrams to reflect the correct order of priority for multiplexed peripherals.

Adds packaging information for the new 64-pin QFN package to **Section 30.0 "Packaging Information"** and the Product Information System.

Updates **Section 5.0 "Flash Program Memory"** with revised code examples in assembler, and new code examples in C.

Updates **Section 6.2 "Device Reset Times"** with revised information, particularly Table 6-3.

Adds the INTTREG register to **Section 4.0 "Memory Organization"** and **Section 7.0 "Interrupt Controller"**.

Makes several additions and changes to **Section 10.0 "I/O Ports"**, including:

- revision of **Section 10.4.2.1 "Peripheral Pin Select Function Priority"**
- revisions to Table 10-3, "Selectable Output Sources"

Makes several changes and additions to **Section 18.0 "Universal Serial Bus with On-The-Go Support (USB OTG)"**, including:

- changes the name of the bit U1CON<x> from RESET to USBRST
- replaces the former Section 18.3 with **Section 18.1 "Hardware Configuration"**, including an expanded discussion of how to interface the microcontroller to application in different USB modes

Updates **Section 21.0 "Programmable Cyclic Redundancy Check (CRC) Generator"** with new illustrations, and a revised **Section 21.1 "User Interface"**.

Updates **Section 22.0 "10-Bit High-Speed A/D Converter"** by changing all references to AD1CHS0, to AD1CHS (as well as other locations in the document). Also revises bit field descriptions in registers, AD1CON3 (bits 7:0) and AD1CHS (bits 12:8).

Makes minor text edits to bit descriptions in **Section 23.0 "Triple Comparator Module"** (Register 23-1) and **Section 25.0 "Charge Time Measurement Unit (CTMU)"** (Register 25-1).

Updates **Section 26.0 "Special Features"** with revised text on the operation of the regulator during POR and Standby mode.

Updates **Section 26.5 "JTAG Interface"** to remove references to programming via the interface.

Makes multiple additions and changes to **Section 29.0 "Electrical Characteristics"**, including:

- Addition of I_{PD} specifications for operation at 60°C
- New DC characteristics of V_{BOR}, V_{BG}, T_{BG} and I_{CNPD}
- Addition of new V_{PEW} specification for V_{DDCORE}
- New AC characteristics for internal oscillator start-up time (T_{LPRC})
- Combination of all Internal RC accuracy information into a single table

Makes other minor typographic corrections throughout the text.