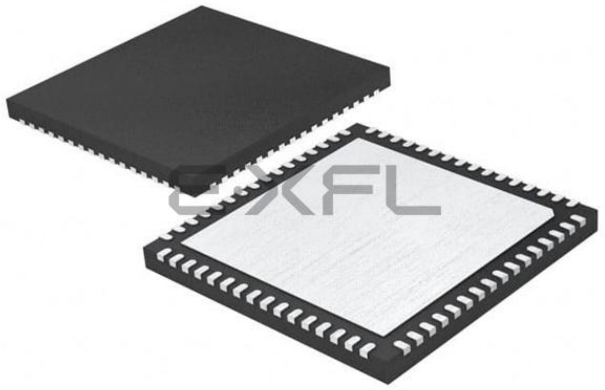


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, GFX, LVD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256da206t-i-mr

Email: info@E-XFL.COM
Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

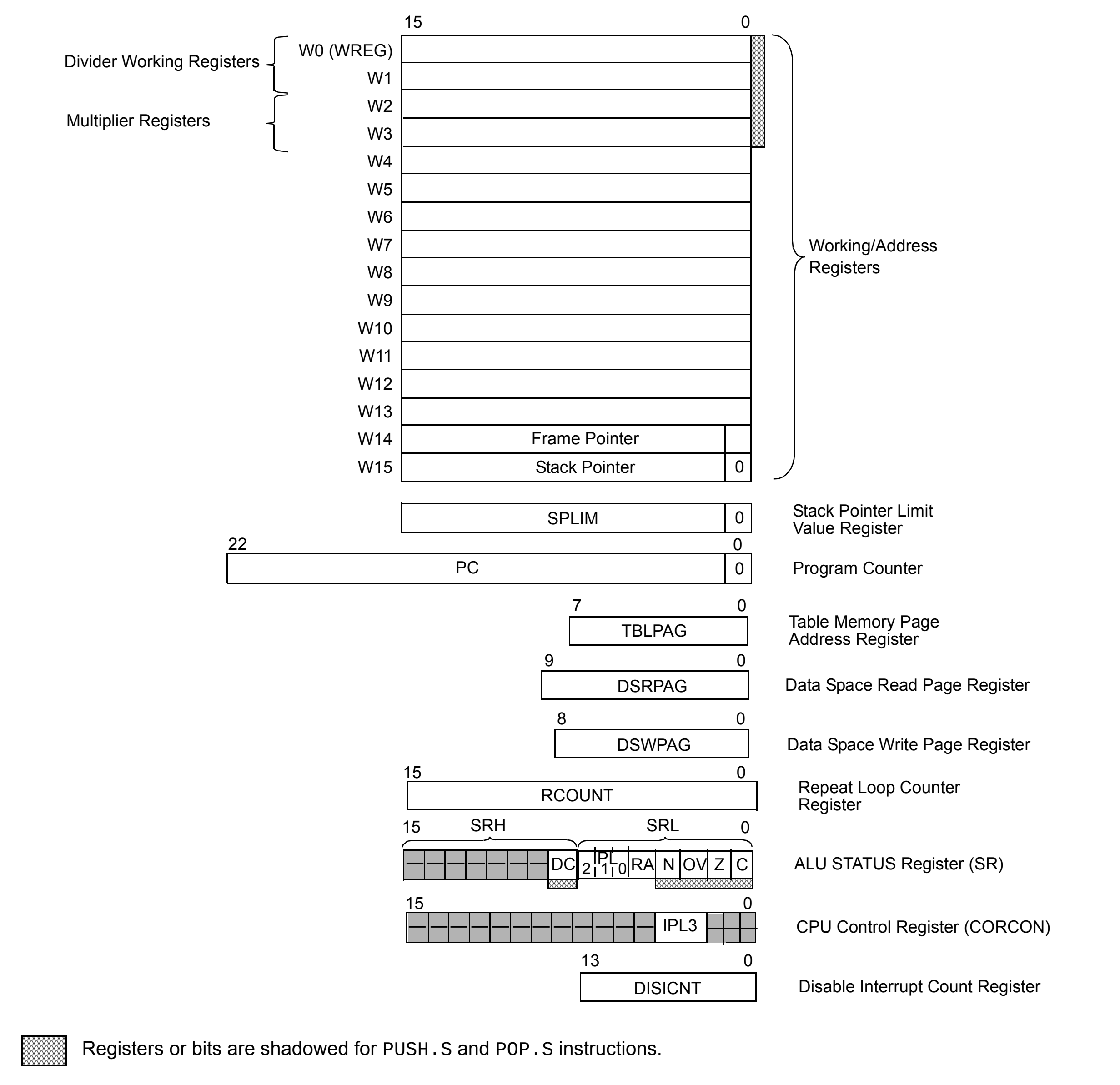
Printed on recycled paper.

ISBN: 978-1-60932-235-9

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

FIGURE 3-2: PROGRAMMER'S MODEL



4.2 Data Memory Space

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 45. "Data Memory with Extended Data Space"** (DS39733). The information in this data sheet supersedes the information in the FRM.

The PIC24F core has a 16-bit wide data memory space, addressable as a single linear range.

The data space is accessed using two Address Generation Units (AGUs), one each for read and write operations. The data space memory map is shown in Figure 4-3.

The 16-bit wide data addresses in the data memory space point to bytes within the Data Space (DS). This gives a DS address range of 64 Kbytes or 32K words. The lower 32 Kbytes (0x0000 to 0x7FFF) of DS is compatible with the PIC24F microcontrollers without EDS.

The upper 32 Kbytes of data memory address space (0x8000 - 0xFFFF) are used as an EDS window.

The EDS window is used to access all memory region implemented in EDS, as shown in Figure 4-4.

The EDS includes any additional internal data memory not accessible by the lower 32-Kbyte data address space and any external memory through EPMP. For more details on accessing internal extended data memory, refer to the "*PIC24F Family Reference Manual*", **Section 45. "Data Memory with Extended Data Space (EDS)"** (DS39733**)**. For more details on accessing external memory using EPMP, refer to the "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730). In PIC24F microcontrollers with EDS, the program memory can also be read from EDS. This is called Program Space Visibility (PSV). Table 4-2 lists the total memory accessible by each of the devices in this family.

The EDS is organized as pages, with a single page called an EDS page that equals the EDS window (32 Kbytes). A particular EDS page is selected through the Data Space Read register (DSRPAG) or Data Space Write register (DSWPAG). For PSV, only the DSRPAG register is used. The combination of the DSRPAG register value and the 16-bit wide data address forms a 24-bit Effective Address (EA). For more information on EDS, refer to **Section 4.3.3 "Reading Data from Program Memory Using EDS"**.

TABLE 4-2: TOTAL MEMORY ACCESSIBLE BY THE DEVICE

Devices	Internal RAM	External RAM Access Using EPMP	Program Memory Access Using EDS
PIC24FJXXXDA210	96 Kbytes (30K + 66K(1))	Yes (up to 16 MB)	Yes
PIC24FJXXXDA206	96 Kbytes (30K + 66K(1))	No	Yes
PIC24FJXXXDA110	24 Kbytes	Yes (up to 16 MB)	Yes
PIC24FJXXXDA106	24 Kbytes	No	Yes

Note 1: The internal RAM above 30 Kbytes can be accessed through EDS window.

TABLE 4-5: ICN 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
CNPD1	0056	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE(1)	CN20PDE(1)	CN19PDE(1)	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	005A	CN47PDE(1)	CN46PDE(1)	CN45PDE(1)	CN44PDE(1)	CN43PDE(1)	CN42PDE(1)	CN41PDE(1)	CN40PDE(1)	CN39PDE(1)	CN38PDE(1)	CN37PDE(1)	CN36PDE(1)	CN35PDE(1)	CN34PDE(1)	CN33PDE(1)	CN32PDE	0000
CNPD4	005C	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE(1)	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE(1)	0000
CNPD5	005E	CN79PDE(1)	CN78PDE(1)	CN77PDE(1)	CN76PDE(1)	CN75PDE(1)	CN74PDE(1)	CN73PDE(1)	—	CN71PDE	CN70PDE(1)	CN69PDE	CN68PDE	CN67PDE(1)	CN66PDE(1)	CN65PDE	CN64PDE	0000
CNPD6	0060	—	—	—	—	—	—	—	—	—	—	—	CN84PDE	CN83PDE	CN82PDE(1)	CN81PDE(1)	CN80PDE(1)	0000
CNEN1	0062	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE(1)	CN20IE(1)	CN19IE(1)	CN18IE	CN17IE	CN16IE	0000
CNEN3	0066	CN47IE(1)	CN46IE(1)	CN45IE(1)	CN44IE(1)	CN43IE(1)	CN42IE(1)	CN41IE(1)	CN40IE(1)	CN39IE(1)	CN38IE(1)	CN37IE(1)	CN36IE(1)	CN35IE(1)	CN34IE(1)	CN33IE(1)	CN32IE	0000
CNEN4	0068	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE(1)	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE(1)	0000
CNEN5	006A	CN79IE(1)	CN78IE(1)	CN77IE(1)	CN76IE(1)	CN75IE(1)	CN74IE(1)	CN73IE(1)	—	CN71IE	CN70IE(1)	CN69IE	CN68IE	CN67IE(1)	CN66IE(1)	CN65IE	CN64IE	0000
CNEN6	006C	—	—	—	—	—	—	—	—	—	—	—	CN84IE	CN83IE	CN82IE(1)	CN81IE(1)	CN80IE(1)	0000
CNPU1	006E	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE(1)	CN20PUE(1)	CN19PUE(1)	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0072	CN47PUE(1)	CN46PUE(1)	CN45PUE(1)	CN44PUE(1)	CN43PUE(1)	CN42PUE(1)	CN41PUE(1)	CN40PUE(1)	CN39PUE(1)	CN38PUE(1)	CN37PUE(1)	CN36PUE(1)	CN35PUE(1)	CN34PUE(1)	CN33PUE(1)	CN32PUE	0000
CNPU4	0074	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE(1)	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE(1)	0000
CNPU5	0076	CN79PUE(1)	CN78PUE(1)	CN77PUE(1)	CN76PUE(1)	CN75PUE(1)	CN74PUE(1)	CN73PUE(1)	—	CN71PUE	CN70PUE(1)	CN69PUE	CN68PUE	CN67PUE(1)	CN66PUE(1)	CN65PUE	CN64PUE	0000
CNPU6	0078	—	—	—	—	—	—	—	—	—	—	—	CN84PUE	CN83PUE	CN82PUE(1)	CN81PUE(1)	CN80PUE(1)	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices; read as '0'.

TABLE 4-6: INTERRUPT CONTROLLER REGISTER MAP (CONTINUED)

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
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0	—	MI2C3IP2	MI2C3IP1	MI2C3IP0	—	SI2C3IP2	SI2C3IP1	SI2C3IP0	4444
IPC22	00D0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC23	00D2	—	—	—	—	—	—	—	—	—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0	0044
IPC25	00D6	—	—	—	—	—	—	—	—	—	—	—	—	—	GFX1IP2	GFX1IP1	GFX1IP0	0004
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

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

Note 1: Unimplemented in 64-pin devices, read as '0'.

2: The Reset value in 64-pin devices are '0004'.

TABLE 4-7: TIMER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 Register																0000
TMR5HLD	0116	Timer5 Holding Register (for 32-bit operations only)																0000
TMR5	0118	Timer5 Register																0000
PR4	011A	Timer4 Period Register																FFFF
PR5	011C	Timer5 Period Register																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

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

REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

R/S-0, HC(1)	R/W-0(1)	R-0, HSC(1)	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0(1)	U-0	U-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)
—	ERASE	—	—	NVMOP3(2)	NVMOP2(2)	NVMOP1(2)	NVMOP0(2)
bit 7							bit 0

Legend:	S = Settable bit	HSC = Hardware Settable/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
HC = Hardware Clearable bit			

bit 15 **WR:** Write Control bit(1)
- `1` = Initiates a Flash memory program or erase operation; the operation is self-timed and the bit is cleared by hardware once the operation is complete
- `0` = Program or erase operation is complete and inactive

bit 14 **WREN:** Write Enable bit(1)
- `1` = Enable Flash program/erase operations
- `0` = Inhibit Flash program/erase operations

bit 13 **WRERR:** Write Sequence Error Flag bit(1)
- `1` = An improper program or erase sequence attempt or termination has occurred (bit is set automatically on any set attempt of the WR bit)
- `0` = The program or erase operation completed normally

bit 12-7 **Unimplemented:** Read as ‘0’

bit 6 **ERASE:** Erase/Program Enable bit(1)
- `1` = Perform the erase operation specified by NVMOP<3:0> on the next WR command
- `0` = Perform the program operation specified by NVMOP<3:0> on the next WR command

bit 5-4 **Unimplemented:** Read as ‘0’

bit 3-0 **NVMOP<3:0>:** NVM Operation Select bits(1,2)
- `1111` = Memory bulk erase operation (ERASE = `1`) or no operation (ERASE = `0`)(3)
- `0011` = Memory word program operation (ERASE = `0`) or no operation (ERASE = `1`)
- `0010` = Memory page erase operation (ERASE = `1`) or no operation (ERASE = `0`)
- `0001` = Memory row program operation (ERASE = `0`) or no operation (ERASE = `1`)

Note 1: These bits can only be reset on POR.
2: All other combinations of NVMOP<3:0> are unimplemented.
3: Available in ICSP™ mode only; refer to the device programming specification.

5.6.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time. To do this, it is necessary to erase the 8-row erase block containing the desired row. The general process is:

1. Read eight rows of program memory (512 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase the block (see Example 5-1):
   a) Set the NVMOP bits (NVMCON<3:0>) to '0010' to configure for block erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the block to be erased into the TBLPAG and W registers.
   c) Write 55h to NVMKEY.
   d) Write AAh to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 64 instructions from data RAM into the program memory buffers (see Example 5-3).
5. Write the program block to Flash memory:
   a) Set the NVMOP bits to '0001' to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 55h to NVMKEY.
   c) Write AAh to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.
6. Repeat steps 4 and 5, using the next available 64 instructions from the block in data RAM by incrementing the value in TBLPAG, until all 512 instructions are written back to Flash memory.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as shown in Example 5-4.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY BLOCK (ASSEMBLY LANGUAGE CODE)

```
; Set up NVMCON for block erase operation
    MOV #0x4042, W0 ;
    MOV W0, NVMCON                    ; Initialize NVMCON
; Init pointer to row to be ERASED
    MOV #tblpage(PROG_ADDR), W0       ;
    MOV W0, TBLPAG                    ; Initialize Program Memory (PM) Page Boundary SFR
    MOV #tbloffset(PROG_ADDR), W0     ; Initialize in-page EA<15:0> pointer
    TBLWTL W0, [W0]                   ; Set base address of erase block
    DISI #5                           ; Block all interrupts with priority <7
                                      ; for next 5 instructions
    MOV.B #0x55, W0
    MOV W0, NVMKEY                    ; Write the 0x55 key
    MOV.B #0xAA, W1 ;
    MOV W1, NVMKEY                    ; Write the 0xAA key
    BSET NVMCON, #WR                  ; Start the erase sequence
    NOP                               ; Insert two NOPs after the erase
    NOP                               ; command is asserted
```


TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
Graphics Controller	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C3 Master Event	85	0000BEh	0001BEh	IFS5<5>	IEC5<5>	IPC21<6:4>
I2C3 Slave Event	84	0000BCh	0001BCh	IFS5<4>	IEC5<4>	IPC21<2:0>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Capture 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
Input Capture 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
Input Capture 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
Input Capture 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
Input Change Notification (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
Low-Voltage Detect (LVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Output Compare 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
Output Compare 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
Output Compare 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
Output Compare 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>
Enhanced Parallel Master Port (EPMP)(1)	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock and Calendar (RTCC)	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI3 Error	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 Event	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

REGISTER 7-2: CORCON: CPU CONTROL REGISTER

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

U-0	U-0	U-0	U-0	R/C-0, HSC	r-1	U-0	U-0
—	—	—	—	IPL3(1)	r	—	—
bit 7							bit 0

Legend:	r = Reserved bit	C = Clearable bit	HSC = Hardware Settable/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 15-4 **Unimplemented:** Read as ‘0’

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

bit 2 **Reserved:** Read as ‘1’

bit 1-0 **Unimplemented:** Read as ‘0’

Note 1: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level; see Register 3-2 for bit description.

REGISTER 7-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5 (CONTINUED)

bit 1 **U3ERIF:** UART3 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 7-11: IFS6: INTERRUPT FLAG STATUS REGISTER 6

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-0, HS	U-0	U-0	U-0	U-0
—	—	—	GFX1IF	—	—	—	—
bit 7							bit 0

Legend:	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 15-5 **Unimplemented:** Read as '0'

bit 4 **GFX1IF:** Graphics 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

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

REGISTER 7-19: 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 10-43: RPOR14: PERIPHERAL PIN SELECT OUTPUT REGISTER 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
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
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
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 **RP29R<5:0>:** RP29 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP29 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP28R<5:0>:** RP28 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP28 (see Table 10-4 for peripheral function numbers).

REGISTER 10-44: RPOR15: PERIPHERAL PIN SELECT OUTPUT REGISTER 15(1)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP31R5	RP31R4	RP31R3	RP31R2	RP31R1	RP31R0
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
—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0
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 **RP31R<5:0>:** RP31 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP31 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP30R<5:0>:** RP30 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP30 (see Table 10-4 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as ‘0’.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK(1)	DISSDO(2)	MODE16	SMP	CKE(3)
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
SSEN(4)	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
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 **DISSCK:** Disable SCKx Pin bit (SPI Master modes only)(1)
- 1 = Internal SPI clock is disabled; pin functions as I/O
- 0 = Internal SPI clock is enabled

bit 11 **DISSDO:** Disable SDOx Pin bit(2)
- 1 = SDOx pin is not used by the module; pin functions as I/O
- 0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
- 1 = Communication is word-wide (16 bits)
- 0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPIx Data Input Sample Phase bit

<u>Master mode:</u>
- 1 = Input data sampled at the end of data output time
- 0 = Input data sampled at the middle of data output time

<u>Slave mode:</u>
SMP must be cleared when SPIx is used in Slave mode.

bit 8 **CKE:** SPIx Clock Edge Select bit(3)
- 1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
- 0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

bit 7 **SSEN:** Slave Select Enable (Slave mode) bit(4)
- 1 = $\overline{\text{SSx}}$ pin is used for Slave mode
- 0 = $\overline{\text{SSx}}$ pin is not used by the module; pin is controlled by the port function

bit 6 **CKP:** Clock Polarity Select bit
- 1 = Idle state for the clock is a high level; active state is a low level
- 0 = Idle state for the clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
- 1 = Master mode
- 0 = Slave mode

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

2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** 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, $\overline{\text{SSx}}$ must be configured to an available RPn/PRIn pin. See **Section 10.4 “Peripheral Pin Select (PPS)”** for more information.

NOTES:

18.0 UNIVERSAL SERIAL BUS WITH ON-THE-GO SUPPORT (USB OTG)

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 27. "USB On-The-Go (OTG)"** (DS39721). The information in this data sheet supersedes the information in the FRM.

PIC24FJ256DA210 family devices contain a full-speed and low-speed compatible, On-The-Go (OTG) USB Serial Interface Engine (SIE). The OTG capability allows the device to act either as a USB peripheral device or as a USB embedded host with limited host capabilities. The OTG capability allows the device to dynamically switch from device to host operation using OTG's Host Negotiation Protocol (HNP).

For more details on OTG operation, refer to the "*On-The-Go Supplement to the USB 2.0 Specification*", published by the USB-IF. For more details on USB operation, refer to the "*Universal Serial Bus Specification*", v2.0.

The USB OTG module offers these features:

- USB functionality in Device and Host modes, and OTG capabilities for application-controlled mode switching
- Software-selectable module speeds of full speed (12 Mbps) or low speed (1.5 Mbps, available in Host mode only)
- Support for all four USB transfer types: control, interrupt, bulk and isochronous
- 16 bidirectional endpoints for a total of 32 unique endpoints
- DMA interface for data RAM access
- Queues up to sixteen unique endpoint transfers without servicing
- Integrated, on-chip USB transceiver with support for off-chip transceivers via a digital interface
- Integrated VBUS generation with on-chip comparators and boost generation, and support of external VBUS comparators and regulators through a digital interface
- Configurations for on-chip bus pull-up and pull-down resistors

A simplified block diagram of the USB OTG module is shown in Figure 18-1.

The USB OTG module can function as a USB peripheral device or as a USB host, and may dynamically switch between Device and Host modes under software control. In either mode, the same data paths and Buffer Descriptors (BDs) are used for the transmission and reception of data.

In discussing USB operation, this section will use a controller-centric nomenclature for describing the direction of the data transfer between the microcontroller and the USB. RX (Receive) will be used to describe transfers that move data from the USB to the microcontroller and TX (Transmit) will be used to describe transfers that move data from the microcontroller to the USB. Table 18-1 shows the relationship between data direction in this nomenclature and the USB tokens exchanged.

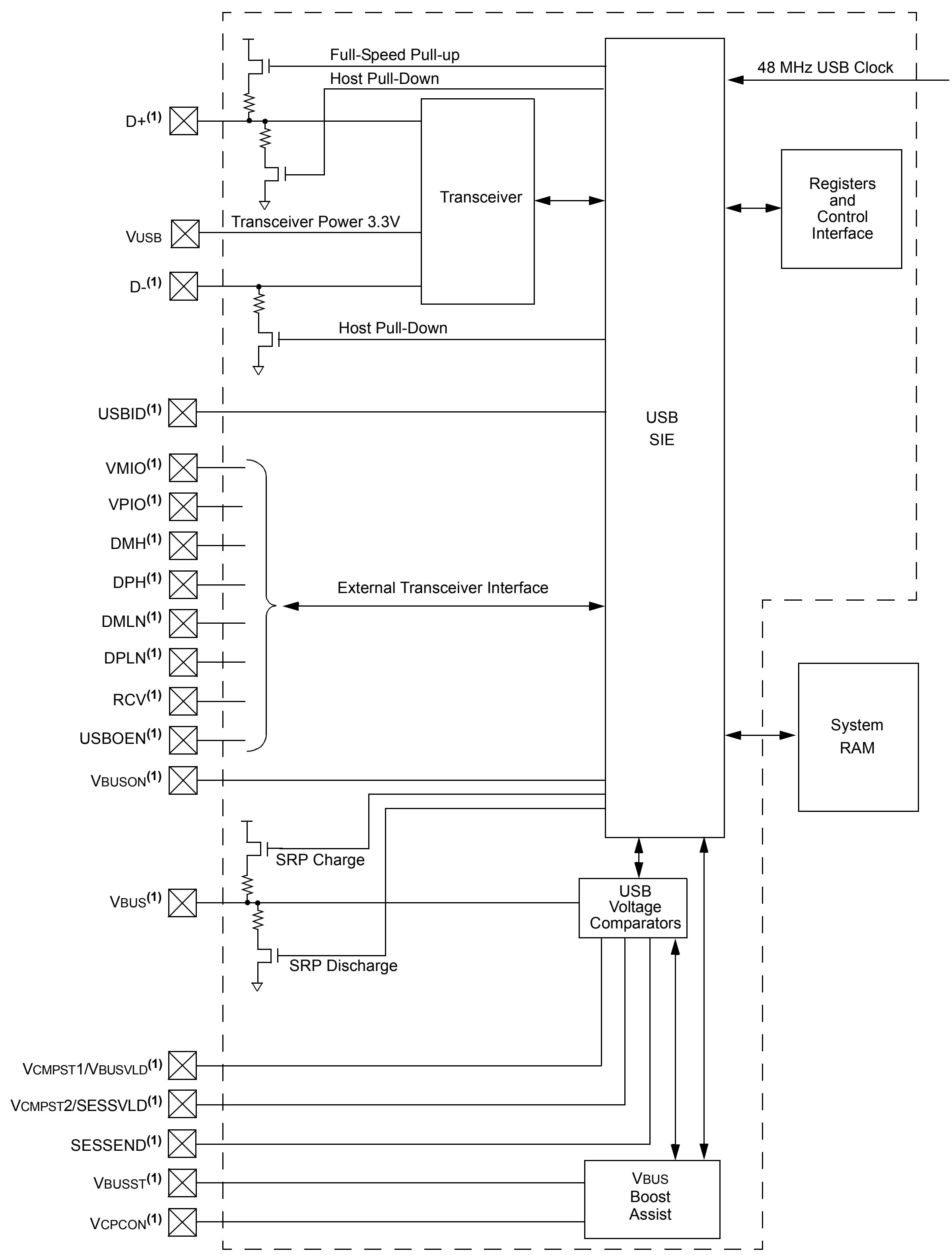
TABLE 18-1: CONTROLLER-CENTRIC DATA DIRECTION FOR USB HOST OR TARGET

USB Mode	**Direction**	
	RX	**TX**
Device	OUT or SETUP	IN
Host	IN	OUT or SETUP

This chapter presents the most basic operations needed to implement USB OTG functionality in an application. A complete and detailed discussion of the USB protocol and its OTG supplement are beyond the scope of this data sheet. It is assumed that the user already has a basic understanding of USB architecture and the latest version of the protocol.

Not all steps for proper USB operation (such as device enumeration) are presented here. It is recommended that application developers use an appropriate device driver to implement all of the necessary features. Microchip provides a number of application-specific resources, such as USB firmware and driver support. Refer to www.microchip.com/usb for the latest firmware and driver support.

FIGURE 18-1: USB OTG MODULE BLOCK DIAGRAM



Note 1: Pins are multiplexed with digital I/O and other device features.

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.

REGISTER 20-6: WKDYHR: WEEKDAY AND HOURS VALUE REGISTER[1]

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

U-0	U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/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 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

bit 7-6 **Unimplemented:** Read as ‘0’

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

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

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

REGISTER 20-7: MINSEC: MINUTES AND SECONDS VALUE REGISTER

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

U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/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 **Unimplemented:** Read as ‘0’

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute’s Tens Digit bits
Contains a value from 0 to 5.

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

bit 7 **Unimplemented:** Read as ‘0’

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

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

NOTES:

REGISTER 27-6: DEVREV: DEVICE REVISION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

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	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend: R = Readable bit	U = Unimplemented bit

bit 23-4	**Unimplemented:** Read as '0'
bit 3-0	**REV<3:0>:** Device revision identifier bits

27.2 On-Chip Voltage Regulator

All PIC24FJ256DA210 family devices power their core digital logic at a nominal 1.8V. This may create an issue for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the PIC24FJ256DA210 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the ENVREG pin. Tying VDD to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR capacitor (such as ceramic) must be connected to the VCAP pin (Figure 27-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor (CEFC) is provided in **Section 30.1 "DC Characteristics"**.

27.2.1 VOLTAGE REGULATOR LOW-VOLTAGE DETECTION

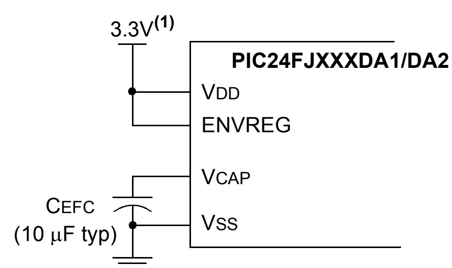
When the on-chip regulator is enabled, it provides a constant voltage of 1.8V nominal to the digital core logic.

The regulator can provide this level from a VDD of about 2.1V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels. In order to prevent "brown-out" conditions when the voltage drops too low for the regulator, the Brown-out Reset occurs. Then the regulator output follows VDD with a typical voltage drop of 300 mV.

To provide information about when the regulator voltage starts reducing, the on-chip regulator includes a simple Low-Voltage Detect circuit, which sets the Low-Voltage Detect Interrupt Flag, LVDIF (IFS4<8>). This can be used to generate an interrupt to trigger an orderly shutdown.

FIGURE 27-1: CONNECTIONS FOR THE ON-CHIP REGULATOR

Regulator Enabled (ENVREG tied to VDD):



Note 1: This is a typical operating voltage. Refer to **Section 30.1 "DC Characteristics"** for the full operating ranges of VDD.

27.2.2 ON-CHIP REGULATOR AND POR

When the voltage regulator is enabled, it takes approximately 10 µs for it to generate output. During this time, designated as TVREG, code execution is disabled. TVREG is applied every time the device resumes operation after any power-down, including Sleep mode. TVREG is determined by the status of the VREGS bit (RCON<8>) and the WUTSEL Configuration bits (CW3<11:10>). Refer to **Section 30.0 "Electrical Characteristics"** for more information on TVREG.

