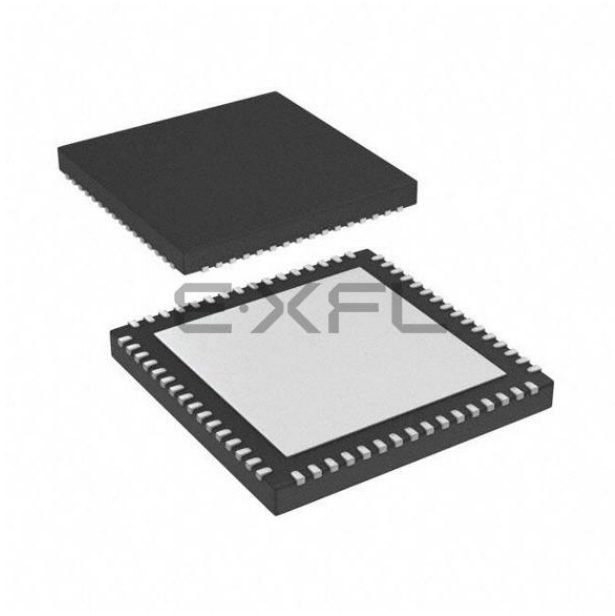
Microchip Technology - PIC24FJ256GB606T-I/MR Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, PMP, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	256KB (85.5K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fj256gb606t-i-mr

TABLE 4: COMPLETE PIN FUNCTION DESCRIPTIONS (PIC24FJXXXGA610 TQFP)

Pin	Function	Pin	Function
1	OCM1C/CTED3/RG15	51	**RP16**/RF3
2	VDD	52	**RP30**/RF2
3	IC4/CTED4/PMD5/RE5	53	**RP15**/RF8
4	SCL3/IC5/PMD6/RE6	54	RF7
5	SDA3/IC6/PMD7/RE7	55	INT0/RF6
6	**RPI38**/OCM1D/RC1	56	SDA1/RG3
7	**RPI39**/OCM2C/RC2	57	SCL1/RG2
8	**RPI40**/OCM2D/RC3	58	PMPCS1/SCL2/RA2
9	AN16/**RPI41**/OCM3C/PMCS2/RC4	59	SDA2/PMA20/RA3
10	AN17/C1IND/**RP21**/ICM1/OCM1A/PMA5/RG6	60	TDI/PMA21/RA4
11	AN18/C1INC/**RP26**/OCM1B/PMA4/RG7	61	TDO/RA5
12	AN19/C2IND/**RP19**/ICM2/OCM2A/PMA3/RG8	62	VDD
13	$\overline{\text{MCLR}}$	63	OSCI/CLKI/RC12
14	AN20/C1INC/C2INC/C3INC/**RP27**/OCM2B/PMA2/PMALU/RG9	64	OSCO/CLKO/RC15
15	VSS	65	VSS
16	VDD	66	**RPI36**/PMA22/RA14
17	TMS/OCM3D/RA0	67	**RPI35**/PMBE1/RA15
18	**RPI33**/PMCS1/RE8	68	CLC4OUT/**RP2**/$\overline{\text{U6RTS}}$/U6BCLK/ICM5/RD8
19	AN21/**RPI34**/PMA19/RE9	69	**RP4**/PMACK2/RD9
20	PGEC3/AN5/C1INA/**RP18**/ICM3/OCM3A/RB5	70	**RP3**/PMA15/PMCS2/RD10
21	PGED3/AN4/C1INB/**RP28**/OCM3B/RB4	71	**RP12**/PMA14/PMCS1/RD11
22	AN3/C2INA/RB3	72	CLC3OUT/**RP11**/$\overline{\text{U6CTS}}$/ICM6/RD0
23	AN2/CTCMP/C2INB/**RP13**/CTED13/RB2	73	SOSCI/C3IND/RC13
24	PGEC1/ALTCVREF-/ALTVREF-/AN1/**RP1**/CTED12/RB1	74	SOSCO/C3INC/**RPI37**/PWRLCLK/RC14
25	PGED1/ALTCVREF+/ALTVREF+/AN0/**RP0**/RB0	75	VSS
26	PGEC2/AN6/**RP6**/RB6	76	**RP24**/U5TX/ICM4/RD1
27	PGED2/AN7/**RP7**/U6TX/RB7	77	**RP23**/PMACK1/RD2
28	CVREF-/VREF-/PMA7/RA9	78	**RP22**/ICM7/PMBE0/RD3
29	CVREF+/VREF+/PMA6/RA10	79	**RPI42**/OCM3E/PMD12/RD12
30	AVDD	80	OCM3F/PMD13/RD13
31	AVSS	81	**RP25**/PMWR/PMENB/RD4
32	AN8/**RP8**/PWRGT/RB8	82	**RP20**/PMRD/$\overline{\text{PMWR}}$/RD5
33	AN9/$\overline{\text{TMPR}}$/**RP9**/T1CK/RB9	83	C3INB/U5RX/OC4/PMD14/RD6
34	CVREF/AN10/PMA13/RB10	84	C3INA/$\overline{\text{U5RTS}}$/U5BCLK/OC5/PMD15/RD7
35	AN11/REFI/PMA12/RB11	85	VCAP
36	VSS	86	N/C
37	VDD	87	$\overline{\text{U5CTS}}$/OC6/PMD11/RF0
38	TCK/RA1	88	PMD10/RF1
39	**RP31**/RF13	89	PMD9/RG1
40	**RPI32**/CTED7/PMA18/RF12	90	PMD8/RG0
41	AN12/U6RX/CTED2/PMA11/RB12	91	AN23/OCM1E/RA6
42	AN13/CTED1/PMA10/RB13	92	AN22/OCM1F/PMA17/RA7
43	AN14/**RP14**/CTED5/CTPLS/PMA1/PMALH/RB14	93	PMD0/RE0
44	AN15/**RP29**/CTED6/PMA0/PMALL/RB15	94	PMD1/RE1
45	VSS	95	CTED11/PMA16/RG14
46	VDD	96	OCM2E/RG12
47	**RPI43**/RD14	97	OCM2F/CTED10/RG13
48	**RP5**/RD15	98	PMD2/RE2
49	**RP10**/PMA9/RF4	99	CTED9/PMD3/RE3
50	**RP17**/PMA8/RF5	100	HLVDIN/CTED8/PMD4/RE4

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select (PPS) functions.

Note: Pinouts are subject to change.

TABLE 1-3: PIC24FJ1024GA610/GB610 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Pin Function	Pin Number/Grid Locator						I/O	Input Buffer	Description
	GA606 64-Pin QFN/TQFP/ QFP	GB606 64-Pin QFN/ TQFP/QFP	GA610 100-Pin TQFP/ QFP	GB610 100-Pin TQFP/ QFP	GA612 121-Pin BGA	GB612 121-Pin BGA			
RD0	46	46	72	72	D9	D9	I/O	DIG/ST	PORTD Digital I/Os
RD1	49	49	76	76	A11	A11	I/O	DIG/ST	
RD2	50	50	77	77	A10	A10	I/O	DIG/ST	
RD3	51	51	78	78	B9	B9	I/O	DIG/ST	
RD4	52	52	81	81	C8	C8	I/O	DIG/ST	
RD5	53	53	82	82	B8	B8	I/O	DIG/ST	
RD6	54	54	83	83	D7	D7	I/O	DIG/ST	
RD7	55	55	84	84	C7	C7	I/O	DIG/ST	
RD8	42	42	68	68	E9	E9	I/O	DIG/ST	
RD9	43	43	69	69	E10	E10	I/O	DIG/ST	
RD10	44	44	70	70	D11	D11	I/O	DIG/ST	
RD11	45	45	71	71	C11	C11	I/O	DIG/ST	
RD12	—	—	79	79	A9	A9	I/O	DIG/ST	
RD13	—	—	80	80	D8	D8	I/O	DIG/ST	
RD14	—	—	47	47	L9	L9	I/O	DIG/ST	
RD15	—	—	48	48	K9	K9	I/O	DIG/ST	
RE0	60	60	93	93	A4	A4	I/O	DIG/ST	PORTE Digital I/Os
RE1	61	61	94	94	B4	B4	I/O	DIG/ST	
RE2	62	62	98	98	B3	B3	I/O	DIG/ST	
RE3	63	63	99	99	A2	A2	I/O	DIG/ST	
RE4	64	64	100	100	A1	A1	I/O	DIG/ST	
RE5	1	1	3	3	D3	D3	I/O	DIG/ST	
RE6	2	2	4	4	C1	C1	I/O	DIG/ST	
RE7	3	3	5	5	D2	D2	I/O	DIG/ST	
RE8	—	—	18	18	G1	G1	I/O	DIG/ST	
RE9	—	—	19	19	G2	G2	I/O	DIG/ST	
REFI	24	24	35	35	J5	J5	I	ST	Reference Clock Input

Legend: TTL = TTL input buffer
ANA = Analog level input/output
DIG = Digital input/output
ST = Schmitt Trigger input buffer
I^2C = I^2C/SMBus input buffer
XCVR = Dedicated Transceiver

TABLE 1-3: PIC24FJ1024GA610/GB610 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Pin Function	Pin Number/Grid Locator						I/O	Input Buffer	Description
	GA606 64-Pin QFN/TQFP/ QFP	GB606 64-Pin QFN/ TQFP/QFP	GA610 100-Pin TQFP/ QFP	GB610 100-Pin TQFP/ QFP	GA612 121-Pin BGA	GB612 121-Pin BGA			
U6RX	27	27	41	41	J7	J7	I	ST	UART6 Receive Input
U6TX	18	18	27	27	J3	J3	O	DIG	UART6 Transmit Output
USBID	—	33	—	51	—	K10	I	ST	USB OTG ID Input
$\overline{\text{USBOEN}}$	—	12	—	21	—	H2	O	DIG	USB Output Enable (active-low)
VBUS	—	34	—	54	—	H8	I	—	VBUS Supply Detect
VCAP	56	56	85	85	B7	B7	P	—	External Filter Capacitor Connection (regulator enabled)
VDD	10,26,38	10,26,38	2,16,37, 46,62	2,16,37, 46,62	C2,F8, G5,H6, K8	C2,F8, G5,H6, K8	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins
VREF+	16	16	25,29	25,29	K2,K3	K2,K3	I	ANA	Comparator and A/D Reference Voltage (high) Input
VREF-	15	15	24,28	24,28	K1,L2	K1,L2	I	ANA	Comparator and A/D Reference Voltage (low) Input
VSS	9,25,41	9,25,41	15,36,45, 65,75	15,36,45, 65,75	B10,F5, F10,G6, G7	B10,F5, F10,G6, G7	P	—	Ground Reference for Peripheral Digital Logic and I/O Pins
VUSB3V3	—	35	—	55	—	H9	P	—	3.3V VUSB

Legend: TTL = TTL input buffer
ANA = Analog level input/output
DIG = Digital input/output
ST = Schmitt Trigger input buffer
I^2C = I^2C/SMBus input buffer
XCVR = Dedicated Transceiver

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC24FJ1024GA610/GB610 family of 16-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins
  (see **Section 2.2 "Power Supply Pins"**)
- The USB transceiver supply, VUSB3V3, regardless of whether or not the USB module is used
  (see **Section 2.2 "Power Supply Pins"**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used
  (see **Section 2.2 "Power Supply Pins"**)
- $\overline{\text{MCLR}}$ pin
  (see **Section 2.3 "Master Clear (MCLR) Pin"**)
- VCAP pin (PIC24F J devices only)
  (see **Section 2.4 "Voltage Regulator Pin (VCAP)"**)

These pins must also be connected if they are being used in the end application:

- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
  (see **Section 2.5 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used
  (see **Section 2.6 "External Oscillator Pins"**)

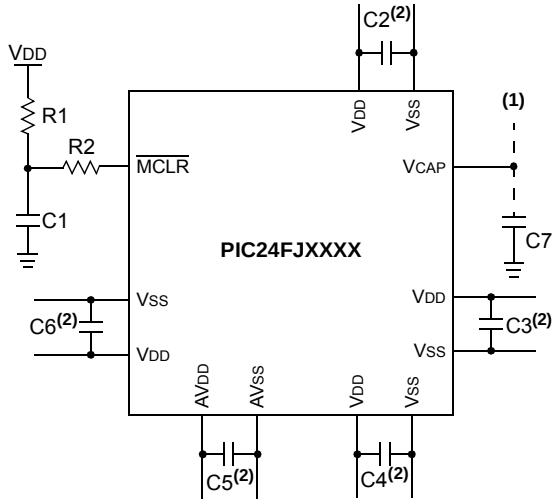
Additionally, the following pins may be required:

- VREF+/VREF- pins used when external voltage reference for analog modules is implemented

Note:	The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1 through C6: 0.1 μF, 50V ceramic

C7: 10 μF, 16V or greater, ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: See **Section 2.4 "Voltage Regulator Pin (VCAP)"** for an explanation of voltage regulator pin connections.

2: The example shown is for a PIC24F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

3.2 CPU Control Registers

REGISTER 3-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0(1)	R/W-0(1)	R/W-0(1)	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2(2)	IPL1(2)	IPL0(2)	RA	N	OV	Z	C
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 15-9 **Unimplemented:** Read as '0'

bit 8 **DC:** ALU Half Carry/Borrow bit
1 = A carry out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry out from the 4th or 8th low-order bit of the result has occurred

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits(1,2)
111 = CPU Interrupt Priority Level is 7 (15); user interrupts are disabled
110 = CPU Interrupt Priority Level is 6 (14)
101 = CPU Interrupt Priority Level is 5 (13)
100 = CPU Interrupt Priority Level is 4 (12)
011 = CPU Interrupt Priority Level is 3 (11)
010 = CPU Interrupt Priority Level is 2 (10)
001 = CPU Interrupt Priority Level is 1 (9)
000 = CPU Interrupt Priority Level is 0 (8)

bit 4 **RA:** REPEAT Loop Active bit
1 = REPEAT loop in progress
0 = REPEAT loop not in progress

bit 3 **N:** ALU Negative bit
1 = Result was negative
0 = Result was not negative (zero or positive)

bit 2 **OV:** ALU Overflow bit
1 = Overflow occurred for signed (2's complement) arithmetic in this arithmetic operation
0 = No overflow has occurred

bit 1 **Z:** ALU Zero bit
1 = An operation, which affects the Z bit, has set it at some time in the past
0 = The most recent operation, which affects the Z bit, has cleared it (i.e., a non-zero result)

bit 0 **C:** ALU Carry/$\overline{\text{Borrow}}$ bit
1 = A carry out from the Most Significant bit (MSb) of the result occurred
0 = No carry out from the Most Significant bit of the result occurred

Note 1: The IPLx Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

2: The IPLx Status bits are concatenated with the IPL3 Status bit (CORCON<3>) to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

6.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"PIC24F Flash Program Memory"** (DS30009715), which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ1024GA610/GB610 family of devices contains internal Flash program memory for storing and executing application code. The program memory is readable, writable and erasable. The Flash memory can be programmed in four ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- JTAG
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24FJ1024GA610/GB610 family device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (named PGECx and PGEDx, respectively), and three other lines for power (V_{DD}), ground (V_{SS}) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

RTSP is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 128 instructions (384 bytes) at a time and erase program memory in blocks of 1024 instructions (3072 bytes) at a time.

The device implements a 7-bit Error Correcting Code (ECC). The NVM block contains a logic to write and read ECC bits to and from the Flash memory. The Flash is programmed at the same time as the corresponding ECC parity bits. The ECC provides improved resistance to Flash errors. ECC single bit errors can be transparently corrected. ECC Double-Bit Errors (ECCDBE) result in a trap.

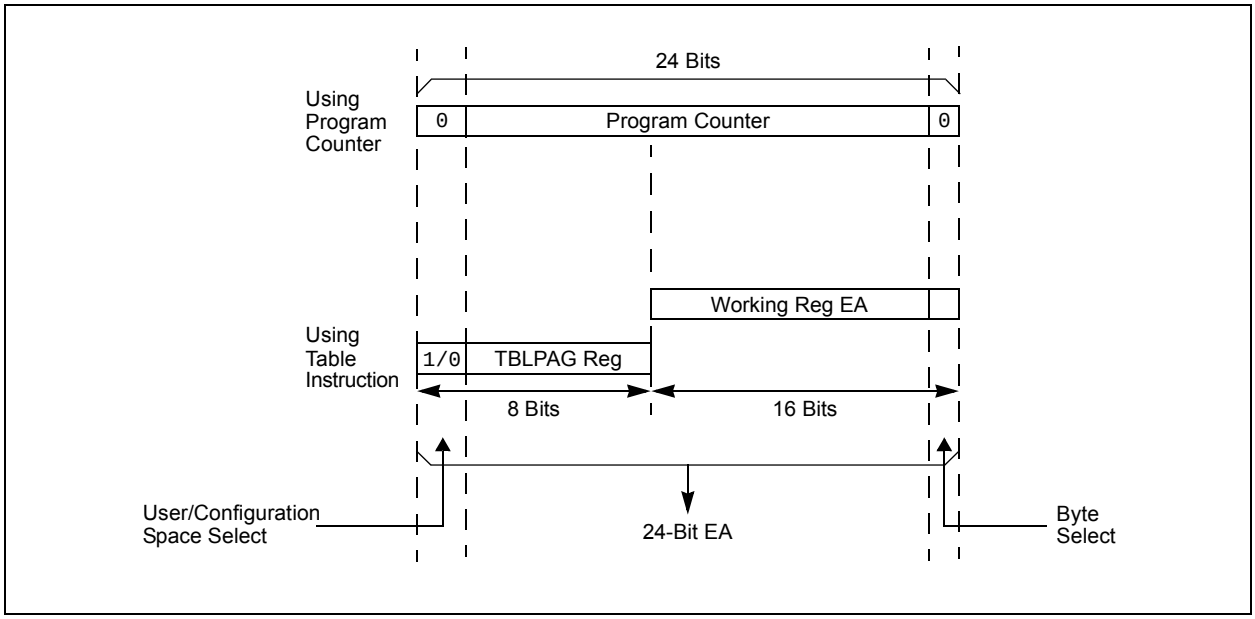
6.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 6-1.

The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 6-1: ADDRESSING FOR TABLE REGISTERS



6.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 128 instructions or 384 bytes. RTSP allows the user to erase blocks of eight rows (1024 instructions) at a time and to program one row at a time. It is also possible to program two instruction word blocks.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 3072 bytes and 384 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, 128 TBLWT instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused address 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 the Table Pointer to point to the programming latches, do a series of TBLWT instructions to load the buffers and set the NVMADRU/NVMADR registers to point to the destination. 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.

6.3 JTAG Operation

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

6.4 Enhanced In-Circuit Serial Programming

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the Program Executive (PE), 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.

6.5 Control Registers

There are four SFRs used to read and write the program Flash memory: NVMCON, NVMADRU, NVMADR and NVMKEY.

The NVMCON register (Register 6-1) 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 6.6 "Programming Operations"** for further details.

The NVMADRU/NVMADR registers contain the upper byte and lower word of the destination of the NVM write or erase operation. Some operations (chip erase, Inactive Partition erase) operate on fixed locations and do not require an address value.

6.6 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.

In Dual Partition mode, programming or erasing the Inactive Partition will not stall the processor; the code in the Active Partition will still execute during the programming operation.

It is important to mask interrupts for a minimum of 5 instruction cycles during Flash programming. This can be done in Assembly using the DISI instruction (see Example 6-1).

REGISTER 8-3: INTCON1: INTERRUPT CONTROL REGISTER 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
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 **NSTDIS:** Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14-5 **Unimplemented:** Read as ‘0’

bit 4 **MATHERR:** Math Error Status bit
1 = Math error trap has occurred
0 = Math error trap has not occurred

bit 3 **ADDRERR:** Address Error Trap Status bit
1 = Address error trap has occurred
0 = Address error trap has not occurred

bit 2 **STKERR:** Stack Error Trap Status bit
1 = Stack error trap has occurred
0 = Stack error trap has not occurred

bit 1 **OSCFAIL:** Oscillator Failure Trap Status bit
1 = Oscillator failure trap has occurred
0 = Oscillator failure trap has not occurred

bit 0 **Unimplemented:** Read as ‘0’

9.6 PLL Oscillator Modes and USB Operation

The PLL block, shown in Figure 9-2, can generate a wide range of clocks used for both parts with USB functionality (PIC24FJ1024GB610 family) and non-USB functionality (PIC24FJ1024GA610 family). All of the available PLL modes are available for both families whether or nor USB is enabled or present.

The PLL input clock source (FRC or POSC) is controlled by the COSC<2:0> bits (OSCCON<14:12>) if the PLL output is used as a system clock. When COSC<2:0> = 001 (FRCPLL), the PLL is clocked from FRC, and when COSC<2:0> = 011 (PRIPLL), the Primary Oscillator (POSC) is connected to the PLL. The default COSC<2:0> value is selected by the FNOSC<2:0> Configuration bits (FOSCSEL<2:0>). Also, REFO can use the PLL when it is not selected for the system clock (COSC<2:0> bits (OSCCON<14:12>) are not '001' or '011'). In this case, the PLL clock source is selected by the PLLSS Configuration bit (FOSC<4>). If PLLSS is cleared ('0'), the PLL is fed by the FRC Oscillator. If the PLLSS Configuration bit is not programmed ('1'), the PLL is clocked from the Primary Oscillator.

When used in a USB application, the 48 MHz internal clock must be running at all times which requires the VCO of the PLL to run at 96 MHz. This, in turn, forces the system clock (that drives the CPU and peripherals) to route the 96 MHz through a fixed divide-by-3 block (generating 32 MHz) and then through a selection of four fixed divisors ('postscaler'). The postscaler output becomes the system clock.

The input to the PLL must be 4 MHz when used in a USB application, which restricts the frequency input sources to be used with a small set of fixed frequency dividers (see Figure 9-2). For example, if a 12 MHz crystal is used, the PLLMODE<3:0> Configuration bits must be set for divide-by-3 to generate the required 4 MHz. A popular baud rate crystal is 11.0592 MHz, but this value cannot be used for USB operation as there are no divisors available to generate 4 MHz exactly. See Table 9-3 for the possible combinations of input clock and PLLMODE<3:0> bits settings for USB operation.

Non-USB operation allows a wider range of PLL input frequencies. The multiplier ratios can be selected as 4x, 6x or 8x and there is no clock prescaler. The postscaler (CPDIV<1:0>) is available and can be used to reduce the system clock to meet the 32 MHz maximum specification. Note that the minimum input frequency to the PLL is 2 MHz, but the range is 2 MHz to 8 MHz. Therefore, it is possible to select a multiplier ratio that exceeds the 32 MHz maximum specification for the system clock. This allows the system clock to be any frequency between 8 MHz (2 MHz input clock with 4x multiplier ratio) and 32 MHz (4 MHz input clock with 8x multiplier ratio). For example, a common crystal frequency is 3.58 MHz ('color burst') and this can be used with the 6x multiplier to generate a system clock of 21.48 MHz. The VCO frequency becomes the system clock.

Note 1: The maximum operating frequency of the system clock is 32 MHz. It is up to the user to select the proper multiplier ratio with the selected clock source frequency.

The PLL block is shown in Figure 9-2. In this system, the PLL input 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<1:0> bits select the system clock speed. The available clock options are listed in Table 9-2.

The USB PLL prescaler must be configured to generate the required 4 MHz VCO input using the PLLMODE<3:0> Configuration bits. This limits the choices for the PLL source frequency to a total of 8 possibilities, as shown in Table 9-3.

TABLE 9-2: SYSTEM CLOCK OPTIONS DURING USB OPERATION

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

Note 1: System clock frequencies below 16 MHz are too slow to allow USB operation. The USB module must be disabled to use this option. See **Section 9.6.1 "Considerations for USB Operation"**.

TABLE 9-3: VALID PRIMARY OSCILLATOR CONFIGURATIONS FOR USB OPERATIONS(1)

PLL Input Frequency	Clock Mode	PLL Mode (PLLMODE<3:0>)
48 MHz	EC	÷12 (0111)
32 MHz	HS, EC	÷8 (0110)
24 MHz	HS, EC	÷6 (0101)
20 MHz	HS, EC	÷5 (0100)
16 MHz	HS, EC	÷4 (0011)
12 MHz	HS, EC	÷3 (0010)
8 MHz	EC, XT, FRC(2)	÷2 (0001)
4 MHz	EC, XT	÷1 (0000)

Note 1: USB operation restricts the VCO input frequency to be 4 MHz.

2: This requires the use of the FRC self-tune feature to maintain the required clock accuracy.

This reference clock output is controlled by the REFOCONL, REFOCONH and REFOTRIML registers. Setting the ROEN bit (REFOCONL<15>) makes the clock signal available on the REFO pin. The RODIV<14:0> bits (REFOCONH<14:0>) enable the selection of different clock divider options. The ROTRIM<0:8> bits (REFOTRIML<7:15>) allow the user to provide a fractional addition to the RODIV<14:0> value. The formula for determining the final frequency output is shown in Equation 9-1. The ROSWEN bit (REFOCONL<9>) indicates that the clock divider has been successfully switched. In order to switch the divider or trim the REFO frequency, the user should ensure that this bit reads as '0'. Write the updated values to the ROTRIM<0:8> and RODIV<14:0> bits, set the ROSWEN bit and then wait until it is cleared before assuming that the REFO clock is valid.

EQUATION 9-1: CALCULATING FREQUENCY OUTPUT

$$F_{REFOUT} = \frac{F_{REFIN}}{2 \cdot \left(N + \frac{M}{512}\right)}$$

Where: F_{REFOUT} = Output Frequency
F_{REFIN} = Input Frequency
N = Value of RODIV<14:0>
M = Value of ROTRIM<8:0>
When N = 0, the initial clock is the same as the input clock

For example, for an input frequency of 10 MHz, an N of 5 and an M of 256, the resulting frequency would be:

$$F_{REFOUT} = \frac{10MHz}{2 \cdot \left(5 + \frac{256}{512}\right)} \cong 909.1kHz$$

The ROSEL<3:0> bits (REFOCONL<3:0>) determine which clock source is used for the reference clock output. The ROSLP bit (REFOCONL<11>) determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSLP bit must be set and the clock selected by the ROSEL<3:0> bits must be enabled for operation during Sleep mode, if possible. Clearing the ROSEL<3:0> bits allows the reference output frequency to change as the system clock changes during any clock switches. The ROOUT bit enables/disables the reference clock output on the REFO pin.

The ROACTIVE bit (REFOCONL<8>) indicates that the module is active; it can be cleared by disabling the module (setting ROEN to '0'). The user must not change the reference clock source, or adjust the trim or divider when the ROACTIVE bit indicates that the module is active. To avoid glitches, the user should not disable the module until the ROACTIVE bit is '1'.

The REFO can use the PLL when it is not selected for the system clock (COSC<2:0> bits (OSCCON<14:12>) are not '001' or '011'). In this case, the PLL clock source is selected by the PLLSS Configuration bit (FOSC<4>). If PLLSS is cleared ('0'), the PLL is fed by the FRC Oscillator. If the PLLSS Configuration bit is not programmed ('1'), the PLL is clocked from the Primary Oscillator.

REGISTER 9-9: REFOCONH: REFERENCE OSCILLATOR CONTROL REGISTER HIGH

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV<14:8>						
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
RODIV<7:0>							
bit 7							bit 0

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

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

bit 14-0 **RODIV<14:0>:** Reference Clock Divider bits

Specifies 1/2 period of the reference clock in the source clocks
(ex: Period of Output = [Reference Source * 2] * RODIV<14:0>; this equation does not apply to RODIV<14:0> = 0).

111111111111111 = REFO clock is the base clock frequency divided by 65,534 (32,767 * 2)
111111111111110 = REFO clock is the base clock frequency divided by 65,532 (32,766 * 2)
•
•
•
000000000000011 = REFO clock is the base clock frequency divided by 6 (3 * 2)
000000000000010 = REFO clock is the base clock frequency divided by 4 (2 * 2)
000000000000001 = REFO clock is the base clock frequency divided by 2 (1 * 2)
000000000000000 = REFO clock is the same frequency as the base clock (no divider)

REGISTER 11-7: PADCON: PORT CONFIGURATION REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
IOCON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
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 **IOCON:** Interrupt-on-Change Enable bit
1 = Interrupt-on-Change functionality is enabled
0 = Interrupt-on-Change functionality is disabled

bit 14-1 **Unimplemented:** Read as ‘0’

bit 0 **PMPTTL:** PMP Port Type bit
1 = TTL levels on PMP port pins
0 = Schmitt Triggers on PMP port pins

13.0 TIMER2/3 AND TIMER4/5

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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Timers"** (DS39704), which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

The Timer2/3 and Timer4/5 modules are 32-bit timers, which can also be configured as four independent, 16-bit timers with selectable operating modes.

As 32-bit timers, Timer2/3 and Timer4/5 can each operate in three modes:

- Two Independent 16-Bit Timers with All 16-Bit Operating modes (except Asynchronous Counter mode)
- Single 32-Bit Timer
- Single 32-Bit Synchronous Counter

They also support these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- Interrupt on a 32-Bit Period Register Match
- A/D Event Trigger (only on Timer2/3 in 32-bit mode and Timer3 in 16-bit mode)

Individually, all four of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the A/D Event Trigger. This Trigger is implemented only on Timer2/3 in 32-bit mode and Timer3 in 16-bit mode. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON and T5CON registers. T2CON and T4CON are shown in generic form in Register 13-1; T3CON and T5CON are shown in Register 13-2.

For 32-bit timer/counter operation, Timer2 and Timer4 are the least significant word; Timer3 and Timer5 are the most significant word of the 32-bit timers.

Note: For 32-bit operation, T3CON and T5CON control bits are ignored. Only T2CON and T4CON control bits are used for setup and control. Timer2 and Timer4 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3 or Timer5 interrupt flags.

To configure Timer2/3 or Timer4/5 for 32-bit operation:

1. Set the T32 or T45 bit (T2CON<3> or T4CON<3> = 1).
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. If TCS is set to an external clock, RPINRx (TyCK) must be configured to an available RPn/RPIn pin. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)"**.
4. Load the timer period value. PR3 (or PR5) will contain the most significant word (msw) of the value, while PR2 (or PR4) contains the least significant word (lsw).
5. If interrupts are required, set the interrupt enable bit, T3IE or T5IE. Use the priority bits, T3IP<2:0> or T5IP<2:0>, to set the interrupt priority. Note that while Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
6. Set the TON bit (= 1).

The timer value, at any point, is stored in the register pair, TMR<3:2> (or TMR<5:4>). TMR3 (TMR5) always contains the most significant word of the count, while TMR2 (TMR4) contains the least significant word.

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit corresponding to that timer (T2CON<3> for Timer2 and Timer3 or T4CON<3> for Timer4 and Timer5).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. See **Section 11.4 "Peripheral Pin Select (PPS)"** for more information.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the interrupt enable bit, TxIE. Use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON (TxCON<15> = 1) bit.

REGISTER 13-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	TECS1(2)	TECS0(2)
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32(3,4)	—	TCS(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:** Timerx On bit
When TxCON<3> = 1:
1 = Starts 32-bit Timerx/y
0 = Stops 32-bit Timerx/y
When TxCON<3> = 0:
1 = Starts 16-bit Timerx
0 = Stops 16-bit Timerx

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

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

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **TECS<1:0>:** Timerx Extended Clock Source Select bits (selected when TCS = 1)(2)
When TCS = 1:
11 = Generic timer (TxCK) external input
10 = LPRC Oscillator
01 = TyCK external clock input
00 = SOSC
When TCS = 0:
These bits are ignored; the timer is clocked from the internal system clock (FOSC/2).

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

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

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

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

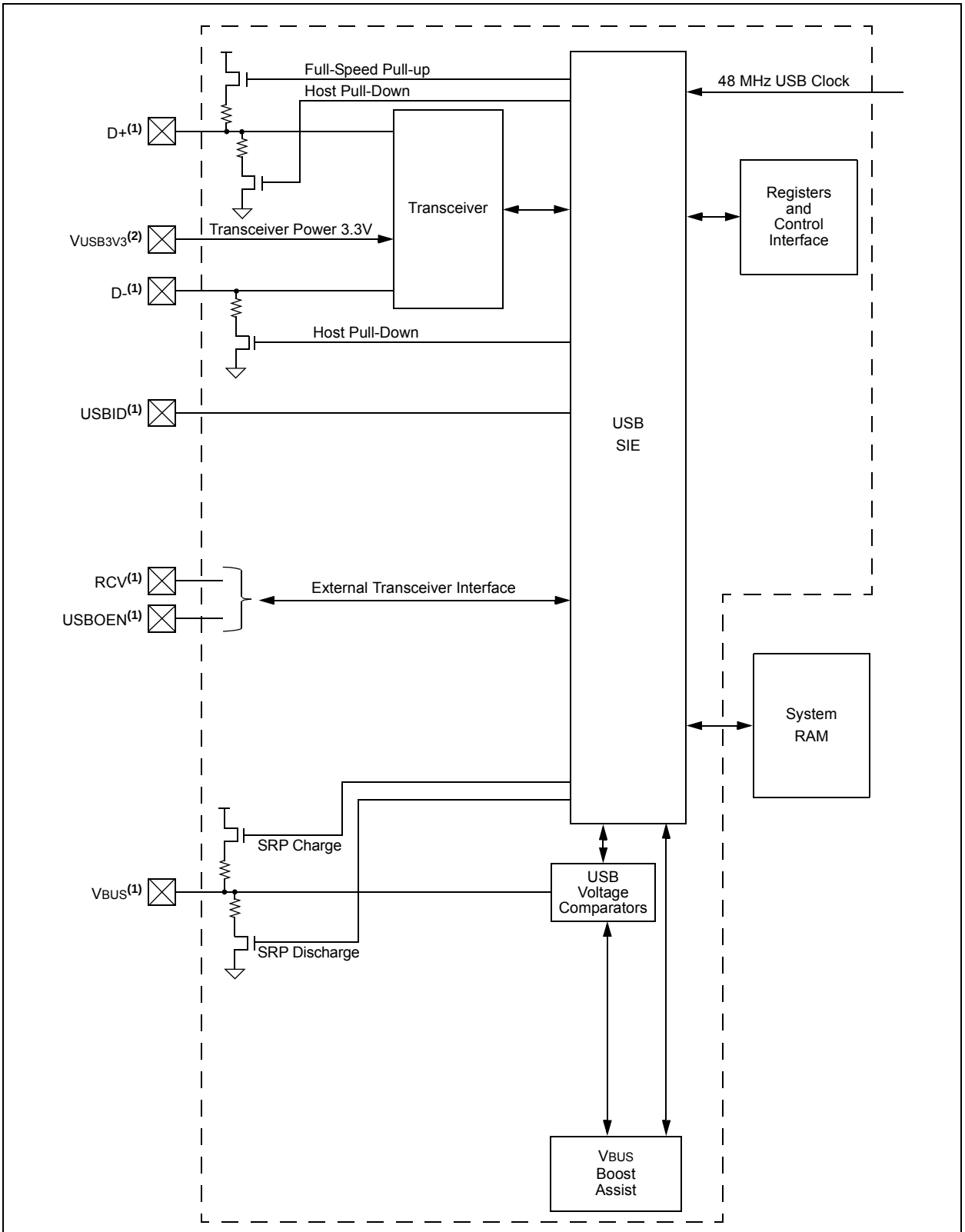
2: If TCS = 1 and TECS<1:0> = x1, the selected external timer input (TxCK or TyCK) must be configured to an available RPn/RPIn pin. For more information, see **Section 11.4 "Peripheral Pin Select (PPS)"**.

3: In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.

4: This bit is labeled T45 in the T4CON register.

NOTES:

FIGURE 20-1: USB OTG MODULE BLOCK DIAGRAM



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

2: Connecting V_{BUS3V3} to V_{DD} is highly recommended, as floating this input can cause increased I_{PD} currents. The pin should be tied to V_{DD} when the USB functions are not used.

20.1.3 CALCULATING TRANSCEIVER POWER REQUIREMENTS

The USB transceiver consumes a variable amount of current depending on the characteristic impedance of the USB cable, the length of the cable, the VUSB supply voltage and the actual data patterns moving across the USB cable. Longer cables have larger capacitances and consume more total energy when switching output states. The total transceiver current consumption will be application-specific. Equation 20-1 can help estimate how much current actually may be required in full-speed applications.

Refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"USB On-The-Go (OTG)"** (DS39721) for a complete discussion on transceiver power consumption.

EQUATION 20-1: ESTIMATING USB TRANSCEIVER CURRENT CONSUMPTION

$$IXCVR = \frac{40\ mA \bullet V_{USB} \bullet P_{ZERO} \bullet P_{IN} \bullet L_{CABLE}}{3.3V \bullet 5m} + I_{PULLUP}$$

Legend: VUSB – Voltage applied to the VUSB3V3 pin in volts (3.0V to 3.6V).

PZERO – Percentage (in decimal) of the IN traffic bits sent by the PIC® microcontroller that are a value of '0'.

PIN – Percentage (in decimal) of total bus bandwidth that is used for IN traffic.

LCABLE – Length (in meters) of the USB cable. The *"USB 2.0 Specification"* requires that full-speed applications use cables no longer than 5m.

IPULLUP – Current which the nominal, 1.5 kΩ pull-up resistor (when enabled) must supply to the USB cable.

20.5.2 COMPLETE A CONTROL TRANSACTION TO A CONNECTED DEVICE

1. Follow the procedure described in **Section 20.5.1 "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., `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 Token Complete Interrupt Flag is asserted (the TRNIF flag is set). This completes the setup phase of the setup transaction, as referenced in Chapter 9 of the *"USB 2.0 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 the Data Toggle bit (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 Token Complete Interrupt Flag 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 2.0 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 Token Complete Interrupt Flag is asserted (the TRNIF flag is set). This completes the status phase of the setup transaction, as described in Chapter 9 of the *"USB 2.0 Specification"*.

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

22.3.2 RTCVAL REGISTER MAPPINGS

REGISTER 22-4: RTCCON2H: RTCC CONTROL REGISTER 2 (HIGH)(1)

R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
DIV<15:8>							
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
DIV<7:0>							
bit 7							bit 0

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

bit 15-0 **DIV<15:0>:** Clock Divide bits
Sets the period of the clock divider counter; value should cause a nominal 1/2 second underflow.

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

REGISTER 25-6: AD1CHS: A/D SAMPLE SELECT 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
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
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
CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
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 **CH0NB<2:0>:** Sample B Channel 0 Negative Input Select bits
1xx = Unimplemented
01x = Unimplemented
001 = Unimplemented
000 = AVSS

bit 12-8 **CH0SB<4:0>:** Sample B Channel 0 Positive Input Select bits
11110 = AVDD **(1)**
11101 = AVSS **(1)**
11100 = Band Gap Reference (VBG) **(1)**
11011 = Reserved
11010 = Reserved
11001 = No channels connected (used for CTMU)
11000 = No channels connected (used for CTMU temperature sensor)
10111 = AN23
10110 = AN22
10101 = AN21
10100 = AN20
10011 = AN19
10010 = AN18
10001 = AN17
10000 = AN16
01111 = AN15
01110 = AN14
01101 = AN13
01100 = AN12
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 7-5 **CH0NA<2:0>:** Sample A Channel 0 Negative Input Select bits
Same definitions as for CHONB<2:0>.

bit 4-0 **CH0SA<4:0>:** Sample A Channel 0 Positive Input Select bits
Same definitions as for CHOSB<4:0>.

Note 1: These input channels do not have corresponding memory-mapped result buffers.