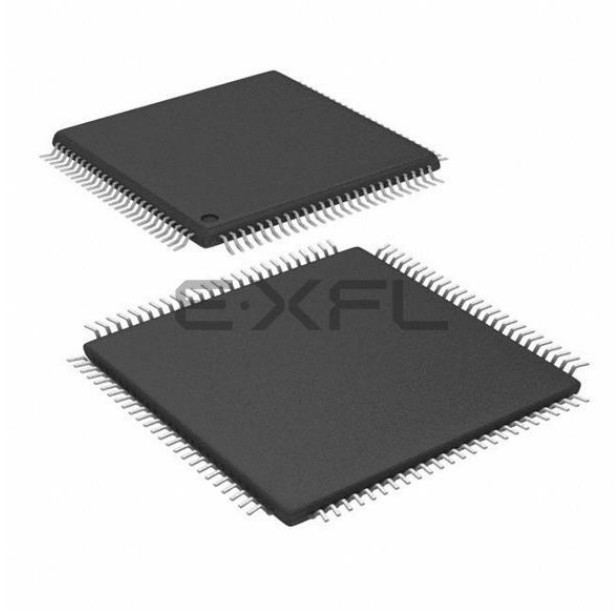
Microchip Technology - PIC24FJ256GA410T-I/PT Datasheet



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, IrDA, LINbus, PMP/PSP, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, LCD, LVD, POR, PWM, WDT
Number of I/O	85
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 24x10/12b; D/A 1x10b
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/pic24fj256ga410t-i-pt

TABLE 1-5: PIC24FJ256GB412 FAMILY PINOUT DESCRIPTION (CONTINUED)

Pin Function	Pin/Pad Number			I/O	Input Buffer	Description
	64-Pin TQFP	100-Pin TQFP	121-Pin TFBGA			
SEG41	—	53	J10	O	ANA	LCD Driver Segment Outputs
SEG42	—	66	E11	O	ANA	
SEG43	—	67	E8	O	ANA	
SEG44	—	79	A9	O	ANA	
SEG45	—	80	D8	O	ANA	
SEG46	—	89	E6	O	ANA	
SEG47	59	88	A6	O	ANA	
SEG48	—	17	G3	O	ANA	
SEG49	—	90	A5	O	ANA	
SEG50	—	1	B2	O	ANA	
SEG51	—	7	E4	O	ANA	
SEG52	—	9	E1	O	ANA	
SEG53	—	39	L6	O	ANA	
SEG54	—	40	K6	O	ANA	
SEG55	—	58	H11	O	ANA	
SEG56	—	59	G10	O	ANA	
SEG57	—	91	C5	O	ANA	
SEG58	—	92	B5	O	ANA	
SEG59	—	95	C4	O	ANA	
SEG60	—	96	C3	O	ANA	
SEG61	—	97	A3	O	ANA	
SEG62	64	100	A1	O	ANA	
SEG63	18	27	J3	O	ANA	
SOSCI	47	73	C10	—	—	Secondary Oscillator/Timer1 Clock Input
SOSCO	48	74	B11	—	—	Secondary Oscillator/Timer1 Clock Output
$\overline{SS4}$/FSYNC4	24	35	K5	I/O	DIG/ST	SPI4 Slave Select/Frame Sync
T1CK	22	33	L4	I	ST	Timer1 Clock
TCK	27	38	J6	I	ST	JTAG Test Clock/Programming Clock Input
TDI	28	60	G11	I	ST	JTAG Test Data/Programming Data Input
TDO	24	61	G9	O	DIG	JTAG Test Data Output
$\overline{TMPR}$	22	33	L4	—	—	Tamper Detect Input
TMS	23	17	G3	I	ST	JTAG Test Mode Select Input
$\overline{U5CTS}$	58	87	B6	I	ST	UART5 Clear-to-Send Output
$\overline{U5RTS}$/U5BCLK	55	84	C7	O	DIG	UART5 Request-to-Send Input
U5RX	54	83	D7	I	ST	UART5 Receive Input
U5TX	49	76	A11	O	DIG	UART5 Transmit Output
$\overline{U6CTS}$	46	72	D9	I	ST	UART6 Clear-to-Send Output
$\overline{U6RTS}$/U6BCLK	42	68	E9	O	DIG	UART6 Request-to-Send Input
U6RX	27	41	J7	I	ST	UART6 Receive Input
U6TX	18	27	J3	O	DIG	UART6 Transmit Output

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

REGISTER 7-2: RCON2: RESET AND SYSTEM CONTROL REGISTER 2

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-0	R/CO-1	R/CO-1	R/CO-1	R/CO-0
—	—	—	—	VDDBOR(1)	VDDPOR(1,2)	VBPOR(1,3)	VBAT(1)
bit 7							bit 0

Legend:	CO = Clearable Only bit	r = Reserved 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 **Reserved:** Maintain as '0'

bit 3 **VDDBOR:** VDD Brown-out Reset Flag bit(1)
1 = A VDD Brown-out Reset has occurred (set by hardware)
0 = A VDD Brown-out Reset has not occurred

bit 2 **VDDPOR:** VDD Power-on Reset Flag bit(1,2)
1 = A VDD Power-on Reset has occurred (set by hardware)
0 = A VDD Power-on Reset has not occurred

bit 1 **VBPOR:** VBPOR Flag bit(1,3)
1 = A VBAT POR has occurred (no battery connected to VBAT pin or VBAT power below Deep Sleep Semaphore register retention level is set by hardware)
0 = A VBAT POR has not occurred

bit 0 **VBAT:** VBAT Flag bit(1)
1 = A POR exit has occurred while power was applied to VBAT pin (set by hardware)
0 = A POR exit from VBAT has not occurred

Note 1: This bit is set in hardware only; it can only be cleared in software.
2: This bit indicates a VDD Power-on Reset. Setting the POR bit (RCON<0>) indicates a VCORE Power-on Reset.
3: This bit is set when the device is originally powered up, even if power is present on VBAT.

TABLE 7-1: RESET FLAG BIT OPERATION

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap Conflict Event	POR
IOPUWR (RCON<14>)	Illegal Opcode or Uninitialized W Register Access	POR
CM (RCON<9>)	Configuration Mismatch Reset	POR
EXTR (RCON<7>)	$\overline{MCLR}$ Reset	POR
SWR (RCON<6>)	RESET Instruction	POR
WDTO (RCON<4>)	WDT Time-out	CLRWDT, PWRSAV Instruction, POR
SLEEP (RCON<3>)	PWRSAV #0 Instruction	POR
DPSLP (RCON<10>)	PWRSAV #0 Instruction while DSEN bit is set	POR
IDLE (RCON<2>)	PWRSAV #1 Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—

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

REGISTER 8-7: IFS1: INTERRUPT FLAG STATUS REGISTER 1 (CONTINUED)

bit 2 **CMIF:** Comparator Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **MI2C1IF:** Master I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** Slave I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 8-13: IFS7: INTERRUPT FLAG STATUS REGISTER 7

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
—	—	JTAGIF	U6ERIF	U6TXIF	U6RXIF	U5ERIF	U5TXIF
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 15-6 **Unimplemented:** Read as '0'

bit 5 **JTAGIF:** JTAG Controller Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **U6ERIF:** UART6 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **U6TXIF:** UART6 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **U6RXIF:** UART6 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U5ERIF:** UART5 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **U5TXIF:** UART5 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

8.4 Interrupt Setup Procedures

8.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS (INTCON1<15>) control bit if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits, for all enabled interrupt sources, may be programmed to the same non-zero value.

Note: At a device Reset, the IPCx registers are initialized such that all user interrupt sources are assigned to Priority Level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

8.4.2 INTERRUPT SERVICE ROUTINE (ISR)

The method that is used to declare an Interrupt Service Routine (ISR) and initialize the IVT with the correct vector address will depend on the programming language (i.e., 'C' or assembler), and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles; otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

8.4.3 TRAP SERVICE ROUTINE (TSR)

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

8.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to Priority Level 7 by inclusive ORing the value, 0Eh, with SRL.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (Levels 8-15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of Priority Levels 1-6 for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

REGISTER 11-31: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
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
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
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 **RP17R<5:0>:** RP17 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP17 (see Table 11-12 for peripheral function numbers).

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

bit 5-0 **RP16R<5:0>:** RP16 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP16 (see Table 11-12 for peripheral function numbers).

REGISTER 11-32: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

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

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

bit 13-8 **RP19R<5:0>:** RP19 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP19 (see Table 11-12 for peripheral function numbers).

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

bit 5-0 **RP18R<5:0>:** RP18 Output Pin Mapping bits
Peripheral Output Number n is assigned to pin, RP18 (see Table 11-12 for peripheral function numbers).

REGISTER 17-2: SPIxCON1H: SPIx CONTROL REGISTER 1 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AUDEN(1)	SPISGNEXT	IGNROV	IGNTUR	AUDMONO(2)	URDTEN(3)	AUDMOD1(4)	AUDMOD0(4)
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
FRMEN	FRMSYNC	FRMPOL	MSSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0
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 **AUDEN:** Audio Codec Support Enable bit(1)
- 1 = Audio protocol is enabled; MSTEN controls the direction of both SCKx and frame (a.k.a. LRC), and this module functions as if FRMEN = 1, FRMSYNC = MSTEN, FRMCNT<2:0> = 001 and SMP = 0, regardless of their actual values
- 0 = Audio protocol is disabled

bit 14 **SPISGNEXT:** SPIx Sign-Extend RX FIFO Read Data Enable bit
- 1 = Data from RX FIFO is sign-extended
- 0 = Data from RX FIFO is not sign-extended

bit 13 **IGNROV:** Ignore Receive Overflow bit
- 1 = A Receive Overflow (ROV) is NOT a critical error; during ROV, data in the FIFO is not overwritten by the receive data
- 0 = A ROV is a critical error that stops SPI operation

bit 12 **IGNTUR:** Ignore Transmit Underrun bit
- 1 = A Transmit Underrun (TUR) is NOT a critical error and data indicated by URDTEN is transmitted until the SPIxTXB is not empty
- 0 = A TUR is a critical error that stops SPI operation

bit 11 **AUDMONO:** Audio Data Format Transmit bit(2)
- 1 = Audio data is mono (i.e., each data word is transmitted on both left and right channels)
- 0 = Audio data is stereo

bit 10 **URDTEN:** Transmit Underrun Data Enable bit(3)
- 1 = Transmits data out of SPIxURDT register during Transmit Underrun conditions
- 0 = Transmits the last received data during Transmit Underrun conditions

bit 9-8 **AUDMOD<1:0>:** Audio Protocol Mode Selection bits(4)
- 11 = PCM/DSP mode
- 10 = Right Justified mode: This module functions as if SPIFE = 1, regardless of its actual value
- 01 = Left Justified mode: This module functions as if SPIFE = 1, regardless of its actual value
- 00 = I^2S mode: This module functions as if SPIFE = 0, regardless of its actual value

bit 7 **FRMEN:** Framed SPIx Support bit
- 1 = Framed SPIx support is enabled ($\overline{SSx}$ pin is used as the FSYNC input/output)
- 0 = Framed SPIx support is disabled

Note 1: AUDEN can only be written when the SPIEN bit = 0.
2: AUDMONO can only be written when the SPIEN bit = 0 and is only valid for AUDEN = 1.
3: URDTEN is only valid when IGNTUR = 1.
4: AUDMOD<1:0> can only be written when the SPIEN bit = 0 and is only valid when AUDEN = 1. When NOT in PCM/DSP mode, this module functions as if FRMSYPW = 1, regardless of its actual value.

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 VUSB3V3 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

$$I_{XCVR} = \frac{40\ mA \bullet V_{USB3V3} \bullet P_{ZERO} \bullet P_{IN} \bullet L_{CABLE}}{3.3V \bullet 5m} + I_{PULLUP}$$

Legend: VUSB3V3 – 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 OTG 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.

20.3 USB Interrupts

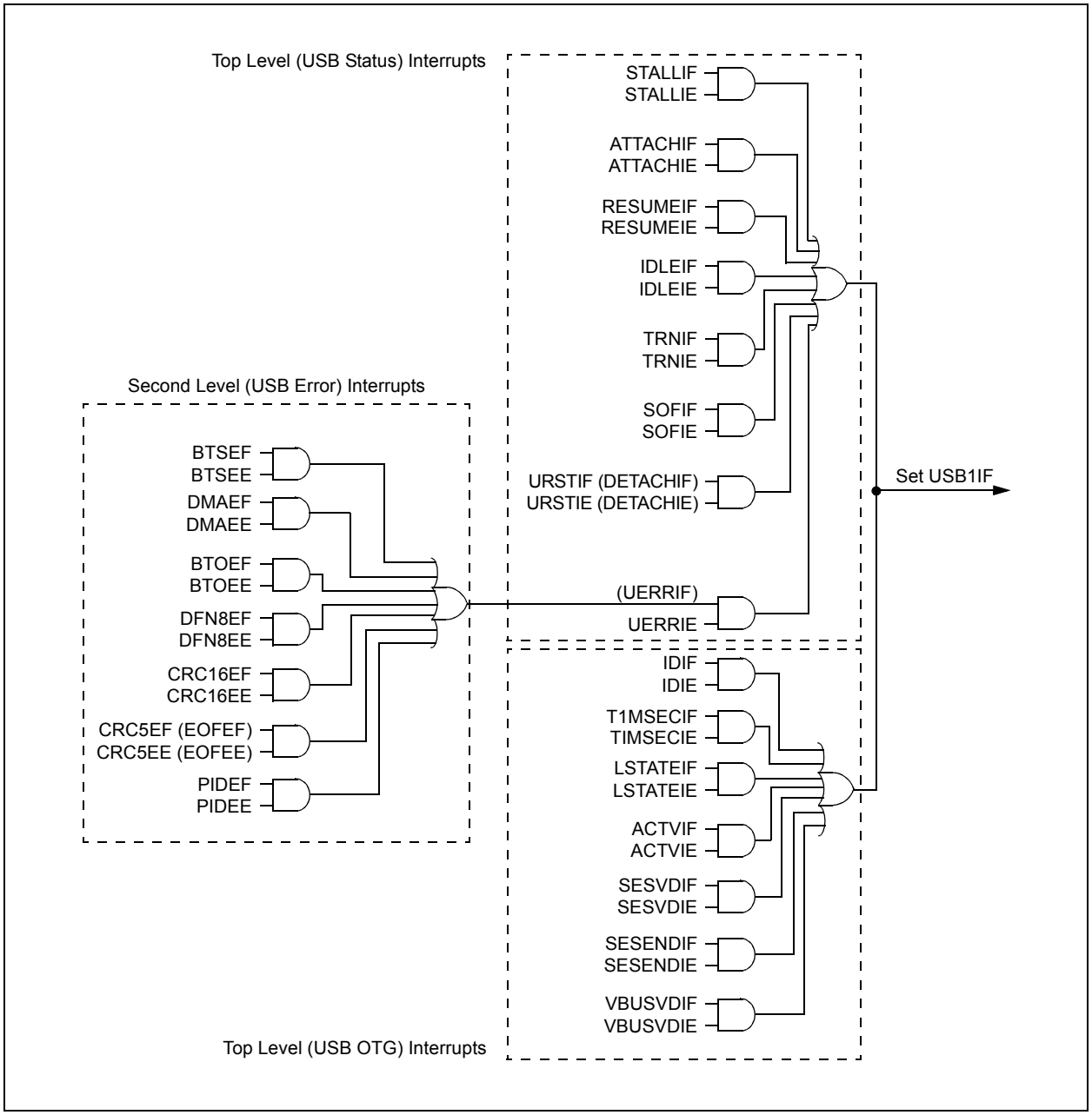
The USB OTG module has many conditions that can be configured to cause an interrupt. All interrupt sources use the same interrupt vector.

Figure 20-8 shows the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the U1IE and U1IR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the U1EIR and U1EIE registers.

An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level. Unlike the device-level interrupt flags in the IFSx registers, USB interrupt flags in the U1IR registers can only be cleared by writing a '1' to the bit position.

Interrupts may be used to trap routine events in a USB transaction. Figure 20-9 provides some common events within a USB frame and their corresponding interrupts.

FIGURE 20-8: USB OTG INTERRUPT FUNNEL



20.4.2 RECEIVING AN IN TOKEN IN DEVICE MODE

1. Attach to a USB host and enumerate as described in Chapter 9 of the *"USB 2.0 Specification"*.
2. Create a data buffer and populate it with the data to send to the host.
3. In the appropriate (even or odd) TX BD for the desired endpoint:
   a) Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address register (BDnADR) with the starting address of the data buffer.
   c) Set the UOWN bit of the status register to '1'.
4. When the USB module receives an IN token, it automatically transmits the data in the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Token Complete Interrupt Flag, TRNIF (U1IR<3>).

20.4.3 RECEIVING AN OUT TOKEN IN DEVICE MODE

1. Attach to a USB host and enumerate as described in Chapter 9 of the *"USB 2.0 Specification"*.
2. Create a data buffer with the amount of data you are expecting from the host.
3. In the appropriate (even or odd) TX BD for the desired endpoint:
   a) Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address register (BDnADR) with the starting address of the data buffer.
   c) Set the UOWN bit of the status register to '1'.
4. When the USB module receives an OUT token, it automatically receives the data sent by the host to the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Token Complete Interrupt Flag, TRNIF (U1IR<3>).

20.5 Host Mode Operation

The following sections describe how to perform common Host mode tasks. In Host mode, USB transfers are invoked explicitly by the host software. The host software is responsible for the Acknowledge portion of the transfer. Also, all transfers are performed using the USB Endpoint 0 Control register (U1EP0) and Buffer Descriptors.

20.5.1 ENABLE HOST MODE AND DISCOVER A CONNECTED DEVICE

1. Enable Host mode by setting the HOSTEN bit (U1CON<3>). This causes the Host mode control bits in other USB OTG registers to become available.
2. Enable the D+ and D- pull-down resistors by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5:4>). Disable the D+ and D- pull-up resistors by clearing the DPPULUP and DMPULUP bits (U1OTGCON<7:6>).
3. At this point, SOF generation begins with the SOF counter loaded with 12,000. Eliminate noise on the USB by clearing the SOFEN bit (U1CON<0>) to disable Start-of-Frame (SOF) packet generation.
4. Enable the device attached interrupt by setting the ATTACHIE bit (U1IE<6>).
5. Wait for the device attached interrupt (U1IR<6> = 1). This is signaled by the USB device changing the state of D+ or D- from '0' to '1' (SE0 to J-state). After it occurs, wait 100 ms for the device power to stabilize.
6. Check the state of the JSTATE and SE0 bits in U1CON. If the JSTATE bit (U1CON<7>) is '0', the connecting device is low speed. If the connecting device is low speed, set the LSPDEN and LSPD bits (U1ADDR<7> and U1EP0<7>) to enable low-speed operation.
7. Reset the USB device by setting the USBRST bit (U1CON<4>) for at least 50 ms, sending Reset signaling on the bus. After 50 ms, terminate the Reset by clearing USBRST.
8. In order to keep the connected device from going into suspend, enable the SOF packet generation by setting the SOFEN bit.
9. Wait 10 ms for the device to recover from Reset.
10. Perform enumeration as described by Chapter 9 of the *"USB 2.0 Specification"*.

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

1. Follow the procedure described in **Section 20.5.1 "Enable Host Mode and Discover a Connected Device"** and **Section 20.5.2 "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 (U1EP0<7>) bit. 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 Token Complete 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 (RETRYDIS) 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 Token Complete Interrupt Flag 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.

20.6 OTG Operation

20.6.1 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 configuring a GPIO pin to disable an external power transistor, or voltage regulator enable signal, which controls the VBUS supply. 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.
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 properly configuring the general purpose I/O port pin 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 reconnect to the USB link by pulling up D+ or D- (via the DPPULUP or DMPULUP bit).

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

21.0 ENHANCED PARALLEL MASTER PORT (EPMP)

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"*, **"Enhanced Parallel Master Port (EPMP)"** (DS39730). The information in this data sheet supersedes the information in the FRM.

The Enhanced Parallel Master Port (EPMP) module provides a parallel, 4-bit (Master mode only), 8-bit (Master and Slave modes) or 16-bit (Master mode only) data bus interface to communicate with off-chip modules, such as memories, FIFOs, LCD controllers and other microcontrollers. This module can serve as either the master or the slave on the communication bus.

For EPMP Master modes, all external addresses are mapped into the internal Extended Data Space (EDS). This is done by allocating a region of the EDS for each Chip Select (CS), and then assigning each Chip Select to a particular external resource, such as a memory or external controller. This region should not be assigned to another device resource, such as RAM or SFRs. To perform a write or read on an external resource, the CPU simply performs a write or read within the address range assigned for the EPMP.

Key features of the EPMP module are:

- Extended Data Space (EDS) Interface allows Direct Access from the CPU
- Up to 23 Programmable Address Lines
- Up to 2 Chip Select Lines
- Up to 2 Acknowledgment Lines (one per Chip Select)
- 4-Bit, 8-Bit or 16-Bit Wide Data Bus
- Programmable Strobe Options (per Chip Select):
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Programmable Address/Data Multiplexing
- Programmable Address Wait States
- Programmable Data Wait States (per Chip Select)
- Programmable Polarity on Control Signals (per Chip Select)
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support:
  - Address Support
  - 4-Byte Deep Auto-Incrementing Buffer

21.1 Specific Package Variations

While all PIC24FJ256GA412/GB412 family devices implement the EPMP, I/O pin constraints place some limits on 16-Bit Master mode operations in some package types. This is reflected in the number of dedicated Chip Select pins implemented and the number of dedicated address lines that are available. The differences are summarized in Table 21-1. All available EPMP pin functions are summarized in Table 21-2.

For 64-pin devices, the dedicated Chip Select pins (PMCS1 and PMS2) are not implemented. In addition, only 16 address lines (PMA<15:0>) are available. If required, PMA14 and PMA15 can be remapped to function as APMCS1 and APMCS2 (Alternate Chip Select 1/2), respectively.

The memory space addressable by the device depends on the number of address lines available, as well as the number of Chip Select signals required for the application. Devices with lower pin counts are more affected by Chip Select requirements, as these take away address lines. Table 21-1 shows the maximum addressable range for each pin count.

TABLE 21-1: EPMP FEATURE DIFFERENCES BY DEVICE PIN COUNT

Device	Dedicated Chip Select		Address Lines	Address Range (bytes)		
	CS1	CS2		No CS	1 CS	2 CS
PIC24FJXXXGX406 (64-pin)(1)	—	—	16	64K	32K	16K
PIC24FJXXXGX410 (100-pin)	X	X	23	16M		
PIC24FJXXXGX412 (121-pin)	X	X	23	16M		

Note 1: The 64-pin devices can use the Alternate Chip Select pins, APMCS1 and APMCS2.

REGISTER 21-4: PMCON4: EPMP CONTROL REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN<13:8>					
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
PTEN<7:3>					PTEN<2:0>		
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 **PTEN15:** PMA15 Port Enable bit
1 = PMA15 functions as either Address Line 15 or Chip Select 2
0 = PMA15 functions as port I/O

bit 14 **PTEN14:** PMA14 Port Enable bit
1 = PMA14 functions as either Address Line 14 or Chip Select 1
0 = PMA14 functions as port I/O

bit 13-3 **PTEN<13:3>:** EPM Address Port Enable bits
1 = PMA<13:3> function as EPM address lines
0 = PMA<13:3> function as port I/Os

bit 2-0 **PTEN<2:0>:** PMALU/PMALH/PMALL Strobe Enable bits
1 = PMA<2:0> function as either address lines or address latch strobes
0 = PMA<2:0> function as port I/Os

REGISTER 22-2: LCDREG: LCD CHARGE PUMP CONTROL REGISTER

RW-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
CPEN	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	RW-0	RW-0
—	—	—	—	—	—	CKSEL1	CKSEL0
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 **CPEN:** 3.6V Charge Pump Enable bit
1 = The regulator generates the highest (3.6V) voltage
0 = Highest voltage in the system is supplied externally (AVDD)

bit 14-2 **Unimplemented:** Read as ‘0’

bit 1-0 **CLKSEL<1:0>:** Regulator Clock Select Control bits
11 = SOSC
10 = 8 MHz FRC
01 = 31 kHz LPRC
00 = Disables regulator and floats regulator voltage output

28.0 10-BIT DIGITAL-TO-ANALOG CONVERTER (DAC)

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"*, **"10-Bit Digital-to-Analog Converter (DAC)"** (DS39615). The information in this data sheet supersedes the information in the FRM.

PIC24FJ256GA412/GB412 family devices include 10-bit Digital-to-Analog Converters (DACs) for generating analog outputs from digital data. A simplified block diagram for a the DAC is shown in Figure 28-1.

The DAC generates an analog output voltage based on the digital input code, according to the formula:

$$V_{DAC} = \frac{V_{DACREF} \times DACxDAT}{1024}$$

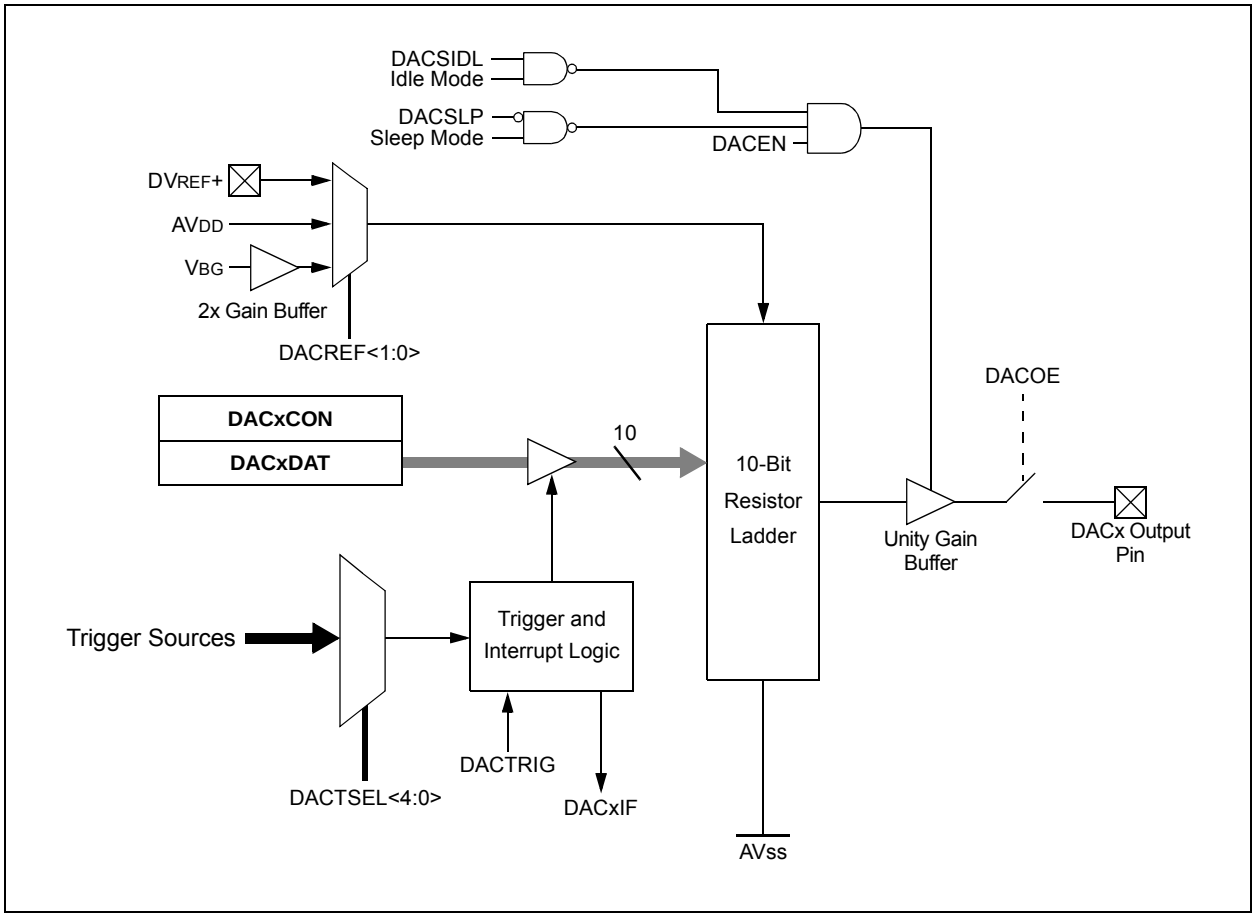
where V_{DAC} is the analog output voltage and V_{DACREF} is the reference voltage selected by DACREF<1:0>.

The DAC includes these features:

- Precision 10-Bit Resistor Ladder for High Accuracy
- Fast Settling Time, Supporting 1 Msps Effective Sampling Rates
- Buffered Output Voltage
- Three User-Selectable Voltage Reference Options
- Multiple Conversion Trigger Options, Plus a Manual Convert-on-Write Option
- Left and Right Justified Input Data Options
- User-Selectable Sleep and Idle mode Operation

When using the DAC, it is required to set the ANSx and TRISx bits for the DACx output pin to configure it as an analog output. See **Section 11.2 "Configuring Analog Port Pins (ANSx)"** for more information.

FIGURE 28-1: DAC SIMPLIFIED BLOCK DIAGRAM



33.0 SPECIAL FEATURES

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 following sections in the *"dsPIC33/PIC24 Reference Manual"*. The information in this data sheet supersedes the information in the FRMs.

- **"Watchdog Timer (WDT)"** (DS39697)
- **"High-Level Device Integration"** (DS39719)
- **"Programming and Diagnostics"** (DS39716)
- **"CodeGuard™ Intermediate Security"** (DS70005182)

PIC24FJ256GA412/GB412 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™
- In-Circuit Emulation

33.1 Configuration Bits

The Flash Configuration Words are stored in the last page location of implemented program memory. Their bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. There are two types of Configuration bits: system operation bits and code-protect bits. The system operation bits determine the power-on settings for system-level components, such as the oscillator and the Watchdog Timer. The code-protect bits prevent program memory from being read and written.

Table 33-1 lists the Configuration register address ranges for each device in Single and Dual Partition Flash modes. A detailed explanation of the various bit functions is provided in Register 33-1 through Register 33-12.

33.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ256GA412/GB412 FAMILY DEVICES

In PIC24FJ256GA412/GB412 family devices, most of the Configuration Words are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration registers during device Resets.

Note: Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The upper byte of all Configuration Words in program memory should always be '0000 0000'. This makes them appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '0's to these locations has no effect on device operation.

Note: Performing a page erase operation on the last page of program memory clears the Flash Configuration Words, enabling code protection as a result. Therefore, users should avoid performing page erase operations on the last page of program memory.

33.1.2 FBOOT

Unlike the Configuration Words, the FBOOT register is not implemented as volatile Flash memory. It is located away from the other Flash Configuration Words, at a constant address for all devices outside of the program memory space. Device Resets do not affect its contents.

Note that the address for FBOOT, 801800h, belongs to the configuration memory space (800000h-FFFFFFh), which can only be accessed using Table Reads and Table Writes.

34.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

34.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

FIGURE 36-16: SPIx MODULE SLAVE MODE TIMING CHARACTERISTICS (CKE = 1)

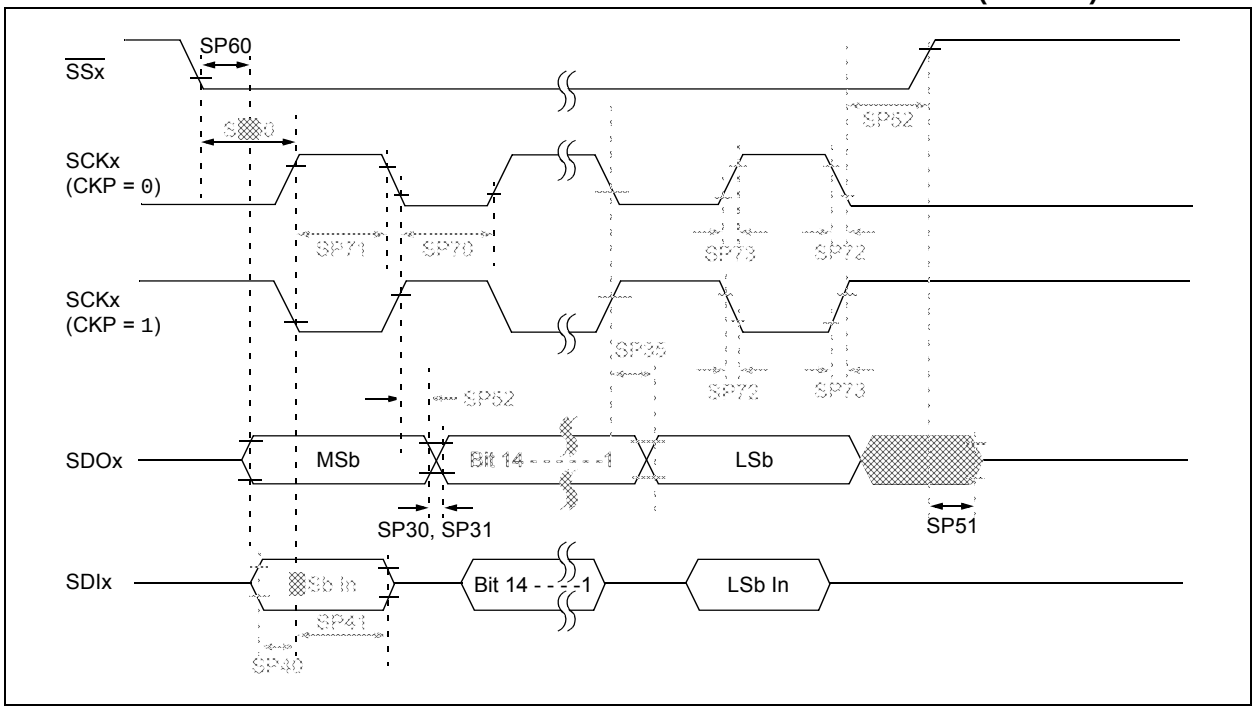


TABLE 36-37: SPIx MODULE SLAVE MODE TIMING REQUIREMENTS (CKE = 1)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
SP70	TscL	SCKx Input Low Time	30	—	—	ns	
SP71	TscH	SCKx Input High Time	30	—	—	ns	
SP72	TscF	SCKx Input Fall Time(2)	—	10	25	ns	
SP73	TscR	SCKx Input Rise Time(2)	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time(2)	—	10	25	ns	
SP31	TdoR	SDOx Data Output Rise Time(2)	—	10	25	ns	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP50	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input	120	—	—	ns	
SP51	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance(3)	10	—	50	ns	
SP52	TscH2ssH TscL2ssH	$\overline{SSx}$ ↑ After SCKx Edge	1.5 TCY + 40	—	—	ns	
SP60	TssL2doV	SDOx Data Output Valid After $\overline{SSx}$ Edge	—	—	50	ns	

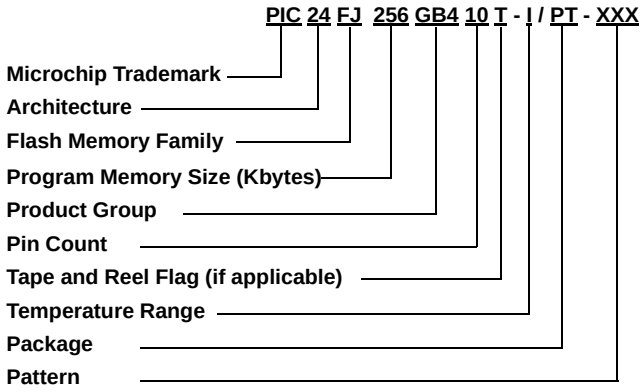
Note 1: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

3: Assumes 50 pF load on all SPIx pins.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture	24	= 16-Bit Modified Harvard without DSP
Flash Memory Family	FJ	= Flash Program Memory
Product Group	GA4	= General Purpose Microcontrollers with Dual Partition Flash Program Memory and XLP Technology
	GB4	= USB OTG Microcontrollers with Dual Partition Flash Program Memory and XLP Technology
Pin Count	06	= 64-pin (TQFP, QFN)
	10	= 100-pin (TQFP)
	12	= 121-pin (TFBGA)
Temperature Range	I	= -40°C to +85°C (Industrial)
Package	BG	= 121-ball (10x10x1.1 mm) TFBGA (Ball Grid Array)
	PT	= 100-lead (12x12x1 mm) TQFP (Thin Quad Flatpack)
	PT	= 64-lead (10x10x1 mm) TQFP (Thin Quad Flatpack)
	MR	= 64-lead (9x9x0.9 mm) QFN (Quad Flatpack, No Lead)
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise)	
	ES	= Engineering Sample

Examples:

a) PIC24FJ64GA406-I/MR:
PIC24F General Purpose Device with Dual Partition Flash Program Memory and XLP Technology, 64-Kbyte Program Memory, 64-Pin, Industrial Temp., QFN Package.

b) PIC24FJ128GB410-I/PT:
PIC24F USB OTG Device with Dual Partition Flash Program Memory and XLP Technology, 128-Kbyte Program Memory, 100-Pin, Industrial Temp., TQFP Package.

c) PIC24FJ256GB412-I/BG:
PIC24F USB OTG Device with Dual Partition Flash Program Memory and XLP Technology, 256-Kbyte Program Memory, 121-Pin, Industrial Temp., TFBGA Package.