Section 27. USB On-The-Go (OTG)

HIGHLIGHTS

This section of the manual contains the following major topics:

27.1 INTRODUCTION

The PIC24F USB module includes the following features:

- USB Full-Speed Support for Host and Device
- Low-Speed Host Support
- USB On-The-Go Support
- Integrated Signaling Resistors
- Integrated Analog Comparators for VBUS Monitoring
- Integrated USB Transceiver
- Hardware Performs Transaction Handshaking
- Integrated DMA Controller to Access System RAM

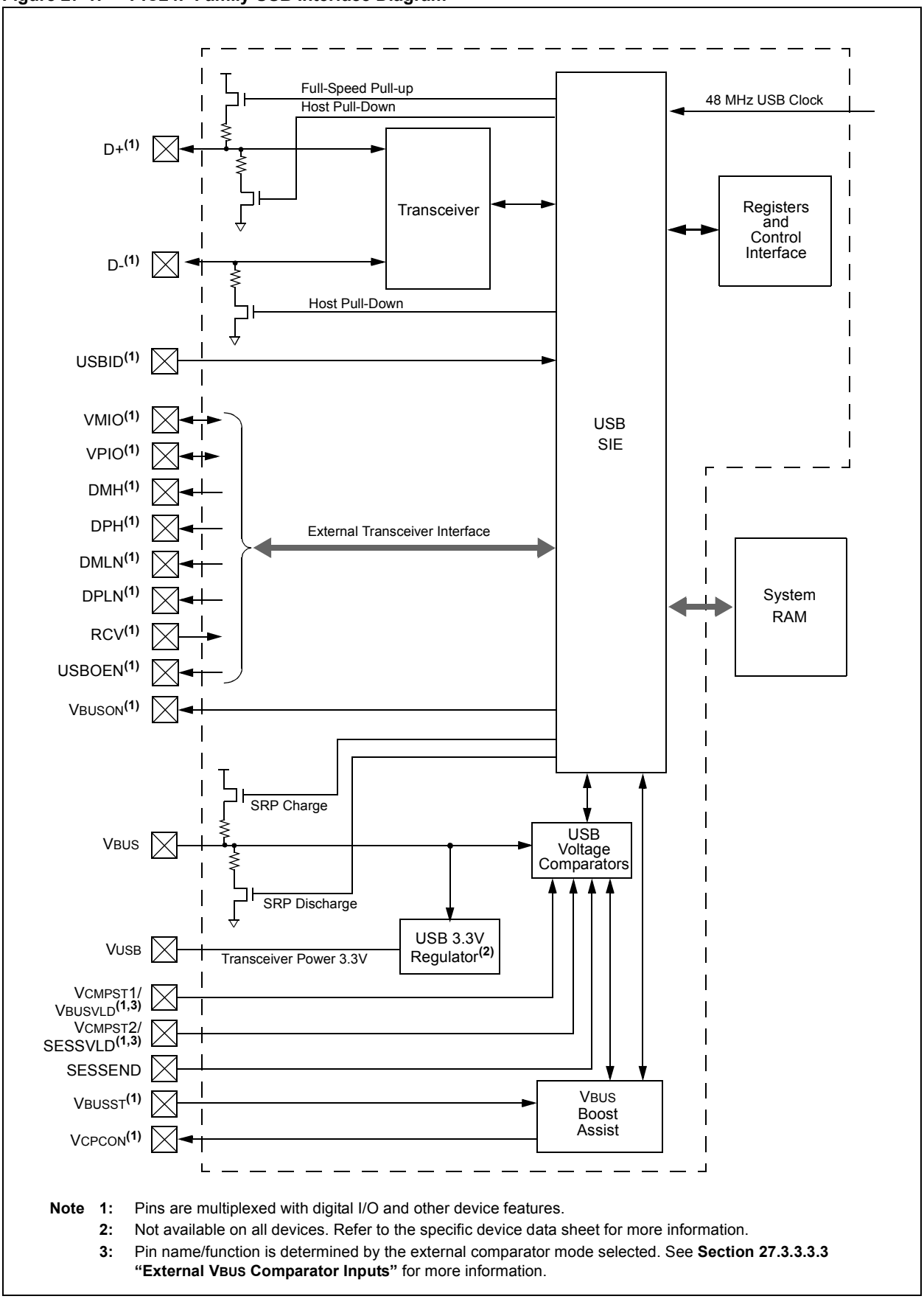
The Universal Serial Bus (USB) module contains the analog and digital components to provide a USB 2.0 full-speed and low-speed embedded host, full-speed device or On-The-Go (OTG) implementation with a minimum of external components.

The USB module consists of the clock generator, the USB voltage comparators, the transceiver, the Serial Interface Engine (SIE), pull-up and pull-down resistors, and the register interface. Figure 27-1 shows the block diagram of the PIC24F USB OTG module.

The clock generator provides the 48 MHz clock required for USB communication. The voltage comparators monitor the voltage on the VBUS pin to determine the state of the bus. The transceiver provides the analog translation between the USB bus and the digital logic. The SIE is a state machine that transfers data to and from the endpoint buffers and generates the protocol for data transfers. The integrated pull-up and pull-down resistors eliminate the need for external signaling components. The register interface allows the CPU to configure and communicate with the module.

The dedicated USB DMA controller allows the USB module to perform direct read and write of USB packet data to the system RAM. This architecture allows the USB module to achieve effective high USB bandwidth with a minimum of microcontroller firmware processing. The dedicated USB DMA controller uses 16-bit address pointers, allowing the USB module to read or write to the first 62 kB of implemented system RAM on the microcontroller. On devices with less than 62 kB of RAM, all of the system RAM can be accessed by the USB DMA controller.

Figure 27-1: PIC24F Family USB Interface Diagram



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

2: Not available on all devices. Refer to the specific device data sheet for more information.

3: Pin name/function is determined by the external comparator mode selected. See **Section 27.3.3.3.3 "External VBUS Comparator Inputs"** for more information.

27.2 CONTROL REGISTERS

The USB module includes the following control and status registers:

- U1OTGIR Register

  The U1OTGIR register records changes of the ID and VBUS pins to enable software to determine the event causing an interrupt. The interrupt bits are cleared by writing a '1' to the respective interrupt.

- U1OTGIE Register

  The U1OTGIE register enables the corresponding interrupt status bits defined in the U1OTGIR register.

- U1OTGSTAT Register

  The U1OTGSTAT register provides access to the status of the VBUS voltage comparators and the debounced status of the ID pin.

- U1OTGCON Register

  The U1OTGCON register controls the operation of the VBUS pin and the pull-up and pull-down resistors.

- U1PWRC Register

  The U1PWRC register controls the power-saving modes.

- U1IR Register

  The U1IR register contains information on pending interrupts. Once an interrupt bit is set, it can be cleared by writing a '1' to the corresponding bit.

- U1IE Register

  The U1IE register values provide gating of the various interrupt signals onto the USB interrupt signal. These values do not interact with the USB module. Setting any of these bits enables the respective interrupt source in the U1IR register.

- U1EIR Register

  The U1EIR register contains information on pending error interrupt values. Once an interrupt bit is set, it can be cleared by writing a '1' to the corresponding bit.

- U1EIE Register

  The U1EIE register values provide gating of the various interrupt signals onto the USB interrupt signal. These values do not interact with the USB module. Setting any of these bits enables the respective interrupt source in the U1EIR register.

- U1STAT Register

  The U1STAT register is a 16-deep FIFO. It is read-only by the CPU and read/write by the USB module. U1STAT is only valid when the U1IR<TRNIF> bit is set.

- U1CON Register

  The U1CON register provides various control information for the module.

- U1ADDR Register

  The U1ADDR register is read/write from the CPU side and read-only from the USB module side. Although the register values affect the settings of the USB module, the content of the registers does not change during access.

  In Device mode, this address defines the USB device address as assigned by the host during the SETUP phase. The firmware writes the address in response to the SETUP request. The address is automatically reset when a USB bus Reset is detected. In Host mode, the module transmits the address provided in this register with the corresponding token packet. This allows the USB module to uniquely address the connected device.

- U1FRMH and U1FRML Registers

  The U1FRMH/U1FRML are read-only registers. The frame number is formed by concatenating the two 8-bit registers. The low-order byte is in the U1FRML register and the high-order byte is in the U1FRMH register.

- U1TOK Register

  The U1TOK is a read/write register required when the module operates as a host. It is used to specify the token type, PID<3:0>, and the endpoint, EP<3:0>, being addressed by the host processor. Writing to this register triggers a host transaction.

- U1SOF Register

  The U1SOF threshold is a read/write register that contains the count bits of the Start-of-Frame (SOF) threshold value used in Host mode only.

  To prevent colliding a packet data with the Start-of-Frame (SOF) token that is sent every 1 ms, the USB module will not send any new transactions within the last U1SOF bit times. The USB module will complete any transactions that are in progress. The SOF interrupt occurs when this threshold is reached, not when the SOF occurs. Transactions started within the SOF threshold are held by the USB module until after the SOF token is sent.

- U1BDTP1

  The U1BDTP1 register is a read/write register that defines the upper 7 bits of the 16-bit base address of the Buffer Descriptor Table (BDT) in the system memory. The BDT is forced to be 512 byte-aligned. This register allows real-time relocation of the BDT.

- U1CNFG1 Register

  The U1CNFG1 register is a read/write register that controls the debug and Idle behavior of the module. The register must be preprogrammed prior to enabling the module.

- U1CNFG2 Register

  The U1CNFG2 register is a read/write register that configures interface signals.

- Endpoint Control Registers

  The Endpoint Control registers control the behavior of the corresponding endpoint.

The following registers are not part of the USB module but are associated with module operation.

- OSCCON: Oscillator Control Register
- IFS1: Interrupt Flag Status Registers
- IEC1: Interrupt Enable Control Registers
- DEVCFG2: Device Configuration Control Register

27.2.1 Clearing USB OTG Interrupts

Unlike device level interrupts, the USB OTG interrupt status flags are not freely writable in software. All USB OTG flag bits are implemented as hardware set only bits. Additionally, these bits can only be cleared in software by writing a '1' to their locations. Writing a '0' to a flag bit (i.e., a BCLR instruction) has no effect.

Issuing a "Bit Set" instruction on one of the interrupt registers is not recommended as the instruction performs a read-modify-write. The result is any interrupt flag that is already set in the register will automatically clear itself, in addition to the bit being set by the instruction. The recommended method is to write the entire register with an appropriate bit mask (Example 27-1).

Note:	Throughout this section, a bit that can only be cleared by writing a '1' to its location is referred to as "Write '1' to clear bit". In register descriptions, this function is indicated by the descriptor, "K".

Example 27-1: Clearing an OTG Interrupt Flag

```
                        // write 1 to bit 0 of U1IR only,
                        // '0' to all other bits
U1IR = 0x0001           // clears only U1IR<0> (URSTIF)
```

27.2.2 USB OTG Module Control Registers

Register 27-1: U1OTGSTAT: USB OTG 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	U-0	R-0, HSC	U-0	R-0, HSC	R-0, HSC	U-0	R-0, HSC
ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD
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 **ID:** ID Pin State Indicator bit
1 = No cable is attached or a type B plug has been plugged into the USB receptacle
0 = A type A plug has been plugged into the USB receptacle

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

bit 5 **LSTATE:** Line State Stable Indicator bit
1 = The USB line state (as defined by SE0 and JSTATE) has been stable for the previous 1 ms
0 = The USB line state has NOT been stable for the previous 1 ms

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

bit 3 **SESVD:** Session Valid Indicator bit
1 = The V_{BUS} voltage is above $V_{A\_SESS\_VLD}$ (as defined in the USB OTG Specification) on the A or B-device
0 = The V_{BUS} voltage is below $V_{A\_SESS\_VLD}$ on the A or B-device

bit 2 **SESEND:** B-Session End Indicator bit
1 = The V_{BUS} voltage is below $V_{B\_SESS\_END}$ (as defined in the USB OTG Specification) on the B-device
0 = The V_{BUS} voltage is above $V_{B\_SESS\_END}$ on the B-device

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

bit 0 **VBUSVD:** A-V_{BUS} Valid Indicator bit
1 = The V_{BUS} voltage is above $V_{A\_VBUS\_VLD}$ (as defined in the USB OTG Specification) on the A-device
0 = The V_{BUS} voltage is below $V_{A\_VBUS\_VLD}$ on the A-device

Register 27-2: U1OTGCON: USB On-The-Go Control Register

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPPULUP	DMPULUP	DPPULDWN(1)	DMPULDWN(1)	VBUSON(1)	OTGEN(1)	VBUSCHG(1)	VBUSDIS(1)
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 15-8 **Unimplemented:** Read as '0'

bit 7 **DPPULUP:** D+ Pull-Up Enable bit
- 1 = D+ data line pull-up resistor is enabled
- 0 = D+ data line pull-up resistor is disabled

bit 6 **DMPULUP:** D- Pull-Up Enable bit
- 1 = D- data line pull-up resistor is enabled
- 0 = D- data line pull-up resistor is disabled

bit 5 **DPPULDWN:** D+ Pull-Down Enable bit(1)
- 1 = D+ data line pull-down resistor is enabled
- 0 = D+ data line pull-down resistor is disabled

bit 4 **DMPULDWN:** D- Pull-Down Enable bit(1)
- 1 = D- data line pull-down resistor is enabled
- 0 = D- data line pull-down resistor is disabled

bit 3 **VBUSON:** VBUS Power-on bit(1)
- 1 = VBUS line is powered
- 0 = VBUS line is not powered

bit 2 **OTGEN:** OTG Features Enable bit(1)
- 1 = USB OTG enabled; all D+/D- pull-ups and pull-downs are controlled by the DPPULxxx/DMPULxxx bits
- 0 = USB OTG disabled; D+/D- pull-ups and pull-downs are controlled in hardware by the settings of the HOSTEN and USBEN bits (U1CON<3,0>)

bit 1 **VBUSCHG:** VBUS Charge Selection bit(1)
- 1 = VBUS line set to charge to 3.3V
- 0 = VBUS line set to charge to 5V

bit 0 **VBUSDIS:** VBUS Discharge Enable bit(1)
- 1 = VBUS line discharged through a resistor
- 0 = VBUS line not discharged

Note 1: These bits are only used in Host mode; do not use in Device mode.

Register 27-3: U1PWRC: USB Power Control Register

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

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

Legend:	HC = Hardware 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 15-8 **Unimplemented:** Read as '0'

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

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

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

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

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

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

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

Register 27-4: 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(2)	ENDPT2(2)	ENDPT1(2)	ENDPT0(2)	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)(2)
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.

2: In Host mode, all transactions are processed through Endpoint 0 and the Endpoint 0 BDTs; therefore, ENDPT<3:0> will always read as '0000'.

Register 27-5: U1CON: USB Control Register (Device Mode)(1)

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

U-0	R-x HSC	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN
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 15-7 **Unimplemented:** Read as '0'

bit 6 **SE0:** Live Single-Ended Zero Flag bit
- 1 = Single-ended zero active on the USB bus
- 0 = No single-ended zero detected

bit 5 **PKTDIS:** Packet Transfer Disable bit
- 1 = SIE token and packet processing disabled; automatically set when a SETUP token is received
- 0 = SIE token and packet processing enabled

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

bit 3 **HOSTEN:** Host Mode Enable bit(1)
- 1 = USB module operates in Host mode when U1PWRC<0> is set; pull-downs on D+ and D- are activated in hardware
- 0 = USB host capability disabled

bit 2 **RESUME:** Resume Signaling Enable bit
- 1 = Resume signaling activated
- 0 = Resume signaling disabled

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit
- 1 = Reset all Ping-Pong Buffer Pointers to the EVEN BD banks
- 0 = Ping-Pong Buffer Pointers not reset

bit 0 **USBEN:** USB Device Mode Enable bit
- 1 = USB module operates in Device mode when U1PWRC<0> is set; D+ pull-up is activated in hardware (device attached)
- 0 = USB Device mode circuitry disabled (device detached)

Note 1: Some USB module register bit definitions/usages depend on the state of the HOSTEN and USBEN bits in the U1CON register. The bit definitions shown in this register description apply when HOSTEN = 0.

Register 27-6: U1CON: USB Control Register (Host Mode Only)(2)

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

R-x, HSC	R-x, HSC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
JSTATE	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN
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 **JSTATE:** Live Differential Receiver J State Flag bit
1 = J state (differential '0' in low speed, differential '1' in full speed) detected on the USB
0 = No J state detected

bit 6 **SE0:** Live Single-Ended Zero Flag bit
1 = Single-ended zero active on the USB bus
0 = No single-ended zero detected

bit 5 **TOKBUSY:** Token Busy Status bit
1 = Token being executed by the USB module in On-The-Go state
0 = No token being executed

bit 4 **USBRST:** Module Reset bit
1 = USB Reset has been generated; for Software Reset, application must set this bit for 50 ms, then clear it(1)
0 = USB Reset terminated

bit 3 **HOSTEN:** Host Mode Enable bit(2)
1 = USB module operates in Host mode when U1PWRC<0> is set; pull-downs on D+ and D- are activated in hardware
0 = USB host mode circuitry disabled

bit 2 **RESUME:** Resume Signaling Enable bit
1 = Resume signaling activated; software must set bit for 10 ms and then clear to enable remote wake-up
0 = Resume signaling disabled

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit
1 = Reset all Ping-Pong Buffer Pointers to the EVEN BD banks
0 = Ping-Pong Buffer Pointers not reset

bit 0 **SOFEN:** Start-of-Frame Enable bit
1 = Start-of-Frame token automatically sent every 1 millisecond
0 = Start-of-Frame token generation disabled

Note 1: May be a continuous interval of at least 50 ms, or several closely spaced intervals of at least 10 ms that total at least 50 ms. See Section 7.1.7.5 of the USB 2.0 Specification for additional details.

2: Some USB module register bit definitions/usages depend on the state of the HOSTEN (and USBEN) bits in the U1CON register. The bit definitions shown in this register description apply when HOSTEN = 1.

Register 27-7: U1ADDR: USB Address Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
LSPDEN(1)	DEVADDR6	DEVADDR5	DEVADDR4	DEVADDR3	DEVADDR2	DEVADDR1	DEVADDR0
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 15-8 **Unimplemented:** Read as '0'

bit 7 **LSPDEN:** Low-Speed Enable bit(1)
1 = USB module operates at low speed
0 = USB module operates at full speed

bit 6-0 **DEVADDR<6:0>:** USB Device Address bits

Note 1: Host mode only. In Device mode, this bit is unimplemented and the module operates at full speed.

Register 27-8: U1TOK: USB Token Register (Host Mode Only)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PID3(1)	PID2(1)	PID1(1)	PID0(1)	EP3	EP2	EP1	EP0
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 15-8 **Unimplemented:** Read as '0'

bit 7-4 **PID<3:0>:** Token Type Identifier bits(1)
1101 = SETUP (TX) token type transaction
1001 = IN (RX) token type transaction
0001 = OUT (TX) token type transaction

bit 3-0 **EP<3:0>:** Token Command Endpoint Address bits
This value must specify a valid endpoint on the attached device.

Note 1: All other combinations are reserved and are not to be used.

Register 27-9: U1SOF: USB OTG Start-of-Token Threshold Register (Host Mode Only)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
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 15-8 **Unimplemented:** Read as '0'

bit 7-0 **CNT<7:0:>** Start-of-Frame Count bits
Value represents 10 + (packet size of n bytes); for example:
0100 1010 = 64-byte packet
0010 1010 = 32-byte packet
0001 0010 = 8-byte packet

Register 27-10: U1CNFG1: USB Configuration Register 1

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

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0
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 15-8 **Unimplemented:** Read as '0'

bit 7 **UTEYE:** USB Eye Pattern Test Enable bit
1 = Eye pattern test enabled
0 = Eye pattern test disabled

bit 6 **UOEMON:** USB $\overline{OE}$ Monitor Enable bit
1 = $\overline{OE}$ signal active; it indicates intervals during which the D+/D- lines are driving
0 = $\overline{OE}$ signal inactive(1)

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

bit 4 **USBSIDL:** USB OTG Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

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

bit 1-0 **PPB<1:0>:** Ping-Pong Buffers Configuration bits
11 = EVEN/ODD ping-pong buffers enabled for Endpoints 1 to 15
10 = EVEN/ODD ping-pong buffers enabled for all endpoints
01 = EVEN/ODD ping-pong buffer enabled for OUT Endpoint 0
00 = EVEN/ODD ping-pong buffers disabled

Note 1: When the UTRDIS bit (U1CNFG2<0>) is set, the $\overline{OE}$ signal is active regardless of the setting of UOEMON.

Register 27-11: U1CNFG2: USB Configuration 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	UVCMPSEL(1)	PUVBUS	EXTI2CEN	UVBUSDIS(2)	UVCMPDIS(2)	UTRDIS(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 15-6 **Unimplemented:** Read as '0'

bit 5 **UVCMPSEL:** External Comparator Input Mode Select bit(1)
1 = Use 3-pin input configuration for VBUS comparators(3)
0 = Use 2-pin input configuration for VBUS comparators(3)

bit 4 **PUVBUS:** VBUS Pull-up Enable bit
1 = Pull-up on VBUS pin enabled
0 = Pull-up on VBUS pin disabled

bit 3 **EXTI2CEN:** I2C™ Interface for External Module Control Enable bit
1 = External module(s) controlled via I2C interface
0 = External module(s) controller via dedicated pins

bit 2 **UVBUSDIS:** On-Chip 5V Boost Regulator Circuit Disable bit(2)
1 = On-chip boost regulator circuit disabled; digital output control interface enabled
0 = On-chip boost regulator circuit active

bit 1 **UVCMPDIS:** On-Chip VBUS Comparator Disable bit(2)
1 = On-chip charge VBUS comparator disabled; digital input status interface enabled
0 = On-chip charge VBUS comparator active

bit 0 **UTRDIS:** On-Chip Transceiver Disable bit(2,4)
1 = On-chip transceiver and VBUS detection disabled; digital transceiver interface enabled
0 = On-chip transceiver and VBUS detection active

Note 1: Unimplemented on some devices; see specific device data sheet for information.

2: Never change these bits while the USBPWR bit is set (U1PWRC<0> = 1).

3: See **Section 27.3.3.3.3 "External VBUS Comparator Inputs"** for information on comparator configurations.

4: The UTRDIS bit is also used on some microcontrollers to control the general purpose (non-USB related) input buffer behavior of D+ and D-. If the USB module will not be used, see the device data sheet for additional details on how to configure these pins as general purpose inputs/outputs.

27.2.3 USB Interrupt Registers

Register 27-12: U1OTGIR: USB OTG Interrupt Status Register (Host Mode)

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

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

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

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

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

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

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

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

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

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

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

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

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

Register 27-13: U1OTGIE: USB OTG Interrupt Enable Register (Host Mode)

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE
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 15-8 **Unimplemented:** Read as '0'

bit 7 **IDIE:** ID Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 6 **T1MSECIE:** 1 Millisecond Timer Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 5 **LSTATEIE:** Line State Stable Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 4 **ACTVIE:** Bus Activity Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 3 **SESVDIE:** Session Valid Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 2 **SESENDIE:** B-Device Session End Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

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

bit 0 **VBUSVDIE:** A-Device VBUS Valid Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

Register 27-14: U1IR: USB Interrupt Status Register (Device Mode Only)

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

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R-0	R/K-0, HS
STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF
bit 7							bit 0

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

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

bit 7 **STALLIF:** STALL Handshake Interrupt bit
- 1 = A STALL handshake was sent by the peripheral during the handshake phase of a transaction
- 0 = A STALL handshake has not been sent

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

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

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

bit 3 **TRNIF:** Transaction Complete Interrupt bit
- 1 = Processing of pending transaction is complete; read U1STAT register for endpoint information
- 0 = Processing of pending transaction is not complete or no transaction is pending (Clearing this bit causes the U1STAT FIFO to advance.)

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

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

bit 0 **URSTIF:** USB Reset Interrupt bit
- 1 = Valid USB Reset has occurred for at least 2.5 μs; Reset state must be cleared before this bit can be re-asserted.
- 0 = No USB Reset has occurred

Register 27-15: U1IR: USB Interrupt Status Register (Host Mode Only)

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

R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R-0	R/K-0, HS
STALLIF	ATTACHIF	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF
bit 7							bit 0

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

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

bit 7 **STALLIF:** STALL Handshake Interrupt bit
1 = A STALL handshake was received from the USB device during the handshake phase of a transaction
0 = A STALL handshake has not been received

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

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

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

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

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

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

bit 0 **DETACHIF:** Detach Interrupt bit
1 = A peripheral detachment has been detected by the module; Reset state must be cleared before this bit can be re-asserted
0 = No peripheral detachment detected

Register 27-16: U1IE: USB Interrupt Enable Register (All USB Modes)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE DETACHIE
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 15-8 **Unimplemented:** Read as '0'

bit 7 **STALLIE:** STALL Handshake Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 6 **ATTACHIE:** Peripheral Attach Interrupt bit (Host mode only)(1)
1 = Interrupt enabled
0 = Interrupt disabled

bit 5 **RESUMEIE:** Resume Interrupt bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 4 **IDLEIE:** Idle Detect Interrupt bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 3 **TRNIE:** Token Processing Complete Interrupt bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 2 **SOFIE:** Start-of-Frame Token Interrupt bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 1 **UERRIE:** USB Error Condition Interrupt bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 0 **URSTIE or DETACHIE:** USB Reset Interrupt (Device mode) or USB Detach Interrupt (Host mode) Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

Note 1: Unimplemented in Device and OTG modes; read as '0'.

Register 27-17: U1EIR: USB Error Interrupt Status Register

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

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF EOFEF	PIDEF
bit 7							bit 0

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

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7 **BTSEF:** Bit Stuff Error Flag bit
1 = Bit stuff error has been detected
0 = No bit stuff error

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **DMAEF:** DMA Error Flag bit
1 = A USB DMA error condition detected; the data size indicated by the BD byte count field is less than the number of received bytes; the received data is truncated.
0 = No DMA error

bit 4 **BTOEF:** Bus Turnaround Time-out Error Flag bit
1 = Bus turnaround time-out has occurred
0 = No bus turnaround time-out

bit 3 **DFN8EF:** Data Field Size Error Flag bit
1 = Data field was not an integral number of bytes
0 = Data field was an integral number of bytes

bit 2 **CRC16EF:** CRC16 Failure Flag bit
1 = CRC16 failed
0 = CRC16 passed

bit 1 For Device mode:
CRC5EF: CRC5 Host Error Flag bit
1 = Token packet rejected due to CRC5 error
0 = Token packet accepted (no CRC5 error)
For Host mode:
EOFEF: End-Of-Frame (EOF) Error Flag bit
1 = End-Of-Frame error has occurred
0 = End-Of-Frame interrupt disabled

bit 0 **PIDEF:** PID Check Failure Flag bit
1 = PID check failed
0 = PID check passed

Register 27-18: U1EIE: USB Error Interrupt Enable Register

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

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE EOFEE	PIDEE
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 15-8 **Unimplemented:** Read as '0'

bit 7 **BTSEE:** Bit Stuff Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

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

bit 5 **DMAEE:** DMA Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 4 **BTOEE:** Bus Turnaround Time-out Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 3 **DFN8EE:** Data Field Size Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 2 **CRC16EE:** CRC16 Failure Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 1 For Device mode:
CRC5EE: CRC5 Host Error Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled
For Host mode:
EOFEE: End-Of-Frame Error interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

bit 0 **PIDEE:** PID Check Failure Interrupt Enable bit
1 = Interrupt enabled
0 = Interrupt disabled

27.2.4 USB Endpoint Management Registers

Register 27-19: U1EPn: USB Endpoint n Control Registers (n = 0 to 15)

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

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LSPD(1,2)	RETRYDIS(1)	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
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 15-8 **Unimplemented:** Read as '0'

bit 7 **LSPD:** Low-Speed Direct Connection Enable bit (UEP0 only)(1,2)
1 = Direct connection to a low-speed device enabled
0 = Direct connection to a low-speed device disabled

bit 6 **RETRYDIS:** Retry Disable bit (UEP0 only)(1)
1 = Retry NAK transactions disabled
0 = Retry NAK transactions enabled; retry done in hardware

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

bit 4 **EPCONDIS:** Bidirectional Endpoint Control bit
If EPTXEN and EPRXEN = 1:
1 = Disable Endpoint n from control transfers; only TX and RX transfers are allowed
0 = Enable Endpoint n for control (SETUP) transfers; TX and RX transfers are also allowed
For all other combinations of EPTXEN and EPRXEN:
This bit is ignored.

bit 3 **EPRXEN:** Endpoint Receive Enable bit
1 = Endpoint n receive enabled
0 = Endpoint n receive disabled

bit 2 **EPTXEN:** Endpoint Transmit Enable bit
1 = Endpoint n transmit enabled
0 = Endpoint n transmit disabled

bit 1 **EPSTALL:** Endpoint Stall Status bit
1 = Endpoint n was stalled
0 = Endpoint n was not stalled

bit 0 **EPHSHK:** Endpoint Handshake Enable bit
1 = Endpoint handshake enabled
0 = Endpoint handshake disabled (typically used for isochronous endpoints)

Note 1: These bits are available only for U1EP0 and only in Host mode. For all other U1EPn registers, these bits are always unimplemented and read as '0'.

2: Clear this bit if the USB module is operating in Host mode and connected to a low-speed device through a hub.

27.2.5 USB VBUS Power Control Register

Register 27-20: U1PWMCON: USB VBUS PWM Generator Control Register

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
PWMEN	—	—	—	—	—	PWMPOL	CNTEN
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-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 **PWMEN:** PWM Enable bit
1 = PWM generator is enabled
0 = PWM generator is disabled; output is held in Reset state specified by PWMPOL

bit 14-10 **Unimplemented:** Read as '0'

bit 9 **PWMPOL:** PWM Polarity bit
1 = PWM output is active-low and resets high
0 = PWM output is active-high and resets low

bit 8 **CNTEN:** PWM Counter Enable bit
1 = Counter is enabled
0 = Counter is disabled

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

27.2.6 USB Frame Registers

Register 27-21: U1FRML: USB Frame Number Low Register

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	
bit 15							bit 8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Frame Count Low Byte							
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 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **FRM0<7:0>:** 11-Bit Frame Number Lower bits
The register bits are updated with the current frame number whenever a SOF token is received.

Register 27-22: U1FRMH: USB Frame Number High 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	U-0	R-0	R-0	R-0
—	—	—	—	—	Frame Count High Byte		
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 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **FRM0<10:8>:** Frame Number Upper 3 bits
The register bits are updated with the current frame number whenever a SOF token is received.

27.3 OPERATION

This section contains a brief overview of USB operation followed by the PIC24F USB module implementation specifics, and module initialization requirements.

27.3.1 USB 2.0 Operation Overview

USB is an asynchronous serial interface with a tiered star configuration. USB is implemented as a master/slave configuration. On a given bus, there can be multiple (up to 127) slaves (devices), but there is only one master (host). There are three possible module implementations: host, device and OTG dual role.

The user should have an understanding of the USB documents available on the USB implementers web site (www.usb.org). This section provides an overview. It does not contain all the information required to implement a USB compliant interface.

27.3.1.1 MODES OF OPERATION

There are two modes of USB implementations covered in this overview: host and device. Support for a dual role OTG implementation, where an application may dynamically switch its role as either host or device.

27.3.1.1.1 Host

The host is the master in a USB system and is responsible for identifying all devices connected to it (enumeration), initiating all transfers, allocating bus bandwidth and supplying power to any bus-powered USB devices connected directly to it. There are two types of hosts.

USB Standard Host:

- A large variety of devices are supported.
- This host supports all USB transfer types.
- USB hubs are supported to allow connection of multiple devices simultaneously.
- Device drivers can be updated to support new devices.
- A type ‘A’ receptacle is used for each port.
- Each port must be able to deliver a minimum of 100 mA for a configured or unconfigured device, and optionally, up to 500 mA for a configured device.
- Full speed and low speed must be supported. High speed can be supported.
- This is a typical personal computer implementation.

Embedded Host:

- Only supports a specific list of devices, referred to as a Targeted Peripheral List (TPL).
- This type of host is only required to support transfer types required by devices in the TPL.
- USB hub support is optional.
- Device drivers are not required to be pocketable.
- A type ‘A’ receptacle is used for each port.
- Only speeds required by devices in the TLP must be supported.
- Each port must be able to deliver a minimum of 100 mA for a configured or unconfigured device, and optionally, up to 500 mA for a configured device.
- This is a typical implementation for a microcontroller.

27.3.1.1.2 Device

The USB device accepts data from the host and responds to requests for data. It performs some peripheral functions, such as a mouse or data storage device.

- Functionality may be class or vendor-specific.
- Draws 100 mA or less before configuration.
- Can draw up to 500 mA after successful negotiation with the host.
- Can support low-speed, full-speed or high-speed protocol. High-speed support requires implementation of full speed.
- Supports control transfers. Supports data transfers required for implementation.
- Optionally supports Session Request Protocol (SRP).
- Can be bus-powered or self-powered.

27.3.1.1.3 OTG Dual Role

The OTG dual role device supports both USB host and device functionality. OTG dual role devices use a micro-AB receptacle. This allows a micro-A or a micro-B plug to be attached. Both the micro-A and micro-B plugs have an additional pin, the ID pin, to signify the connect type. The plug type, micro-A or micro-B, determines the default role of the OTG device, host or USB device. An OTG device will perform the role of a host when a micro-A plug is detected. When a micro-B plug is detected, the role of a USB device is performed.

When an OTG device is directly connected to another OTG device using an OTG cable, micro-A to micro-B, Host Negotiation Protocol (HNP) can be used to swap the roles of the host and USB device between the two without disconnecting and reconnecting cabling. To differentiate between the two OTG devices, the term, “A-device”, is used to refer to the device connected to the micro-A plug and “B-device” is used to refer to the OTG device connected to the micro-B plug.

OTG dual role operating as a host (A-device):

- Only supports a specific list of devices, referred to as a Targeted Peripheral List (TPL). Generic class support is not allowed.
- Only required to support transaction types required by devices in the TPL.
- USB hub support is optional.
- Device drivers are not required to be pocketable.
- Only a single micro-AB receptacle is used.
- Only full speed must be supported. High speed and/or low speed can be supported.
- The USB port must be able to deliver a minimum of 8 mA for a configured or unconfigured device, and optionally, up to 500 mA for a configured device.
- Supports Host Negotiation Protocol (HNP) and Session Request Protocol (SRP). The host can switch roles to become a device. The initial role as a host or device is determined by the plug type, micro-A or micro-B, inserted into the micro-AB receptacle.
- The A-device supplies VBUS power, when the bus is powered, even if the roles are swapped using HNP.

OTG dual role operating as a USB device (B-device):

- Class or vendor-specific functionality.
- Draws 8 mA or less before configuration.
- Is typically self-powered due to low-current requirements, but can draw up to 500 mA after successful negotiation with the host.
- Only a single micro-AB receptacle is used.
- Must support full speed. Support of low speed and/or high speed is optional.
- Supports control transactions. Supports data transactions required for implementation.
- Supports Session Request Protocol (SRP) and/or Host Negotiation Protocol (HNP).
- The A-device supplies VBUS power, when the bus is powered, even if the roles are swapped using HNP.

27.3.1.2 PROTOCOL

USB communication requires the use of a specific protocol. The following subsections provide an overview of the required protocol.

27.3.1.2.1 Bus Transfers

Communication on the USB bus takes place between a host and a device in transfers. There are four transfer types possible on the bus. Each transfer type has unique features. An embedded or OTG host can implement only the control and the data transfer(s) it will use. There are three transfer speeds as defined by the USB 2.0 Specification: 480 Mbps (high speed), 12 Mbps (full speed) and 1.5 Mbps (low speed). PIC24F devices support full-speed and low-speed Host mode transfers.

- Control Transfer: This is used to determine a device's type during enumeration and to control the device. A percentage of the USB bandwidth is ensured to control transfers. The data is verified by a CRC and reception by the target is verified.
- Interrupt Transfer: This is a scheduled transfer of data where the host allocates time slots for the transfers. This time slot allocation results in the device being polled in a periodic manner. The data is verified by a CRC and reception by the target is verified.
- Isochronous Transfer: This is a scheduled transfer of data where the host allocates time slots for the transactions. Reception of the data is not verified, but the data integrity is verified by the device using a CRC. This transfer type is typically used for audio and video.
- Bulk Transfer: This is used to move large amounts of data where the time of the transaction is not ensured. Time for this type of transfer is allocated from any time not allocated to the other three transfer types. The data is verified by a CRC and reception is verified.

Table 27-1 shows maximum data size, percentage of bandwidth, timeliness and integrity information for each of the transfer types.

Table 27-1: Transaction Types (Full-Speed Operation)

Transaction Type	Timeliness Ensured	Data Arrival Ensured	Maximum Packet Size	Maximum Percentage of Bandwidth
Control	No	Yes	64	10%(1)
Interrupt	Yes	Yes	64	90%
Isochronous	Yes	No	1023	69%
Bulk	No	Yes	64	0-100%

Note 1: Data stage only. By USB Specification, control transfers are assured 10% of the total bandwidth. Refer to the USB Specification 2.0, Section 5, for more information.

27.3.1.2.2 Endpoints and USB Descriptors

All data transferred on the bus is sent or received through endpoints. USB supports devices with up to 16 endpoints. Each endpoint can have transmit (TX) and/or receive (RX) functions. Each endpoint utilizes one transaction type. Endpoint 0 utilizes only control transfers.

27.3.1.3 PHYSICAL BUS INTERFACE

27.3.1.3.1 Bus Speed Selection

The USB Specification defines full-speed operation as a host and a device, and optionally, low speed and/or high speed. A data line pull-up resistor is used to identify a device as full speed or low speed. For full-speed operation, the D+ line is pulled up; for low-speed operation, the D- line is pulled up.

27.3.1.3.2 VBUS Control

VBUS is the 5V USB power supplied by the host or a hub to operate bus-powered devices. The need for VBUS control depends on the role of the application. If VBUS power must be enabled and disabled, the control must be managed by firmware. The following lists the VBUS requirements. Refer to the specific device data sheet for VBUS electrical parameters.

- A standard or embedded host typically supplies power to the bus at all times.
- A USB device never supplies power to the bus.
- An OTG dual role implementation must be able to control VBUS to comply with host and device requirements, as well as support OTG functions. An OTG A-device supplies power to the bus, and can turn off VBUS to conserve power.

Note:	A device can pulse VBUS as part of Session Request Protocol (SRP).

27.3.2 PIC24F Implementation Specifics

This section details how the USB Specification requirements are implemented in the PIC24F USB module.

27.3.2.1 BUS SPEED

The PIC24F USB module supports the following speeds:

- Full-speed operation as a host and a device.
- Low-speed operation as a host.

27.3.2.2 ENDPOINTS AND DESCRIPTORS

All USB endpoints are implemented as buffers in RAM. The CPU and the USB module both have access to the buffers. The USB module can read and write USB data packets directly into (or from) these buffers using a dedicated Direct Memory Access (DMA) interface. To arbitrate access to these buffers between the USB module and the CPU, a semaphore flag system is used. Each endpoint can be configured for transmit (TX) and/or receive (RX), and each may have an ODD and an EVEN buffer.

The Buffer Descriptor Table (BDT) is a table located in RAM that contains the addresses of each of the endpoint data buffers and information about each buffer (see Figure 27-2, Figure 27-3 and Figure 27-4). The addresses and additional information contained within the BDT are used to control the operation of the dedicated USB DMA interface. Each BDT entry within the table is called a Buffer Descriptor (BD) and is 4 bytes long. Two BDT entries are used for each endpoint, one for RX and one for TX operations. However, when “ping-pong buffering” (described later) is enabled on an endpoint, four BDT entries are used for each endpoint, two for RX and two for TX. Therefore, 8 or 16 bytes must be allocated for an endpoint even if all the buffers for that endpoint will not be used.

The BDT can be thought of as an array of “hardware registers” that control the behavior of the USB DMA interface, independently, for each endpoint number and direction. However, these registers are not statically allocated in the hardware design. They are implemented using standard microcontroller RAM, allowing the entire BDT to be dynamically moved at run time. Most firmware designs will not need to do this, however, and will only wish to write to the U1BDTP1 register once during the initialization of the USB module.

The U1BDTP1 register determines where the BDT will be located in RAM. The BDT must be located in an implemented RAM location starting at a 512-byte boundary. For microcontrollers with more than 62 kB of implemented RAM, the BDT must be located entirely within the first 16 bits of address space. Refer to Equation 27-1 to determine the proper value to load into the U1BDTP1 register.

Equation 27-1: Location of the BDT

BDT Base Address = U1BDTP1 * 256

and

U1BDTP1 = (Desired BDT Base Address)/256

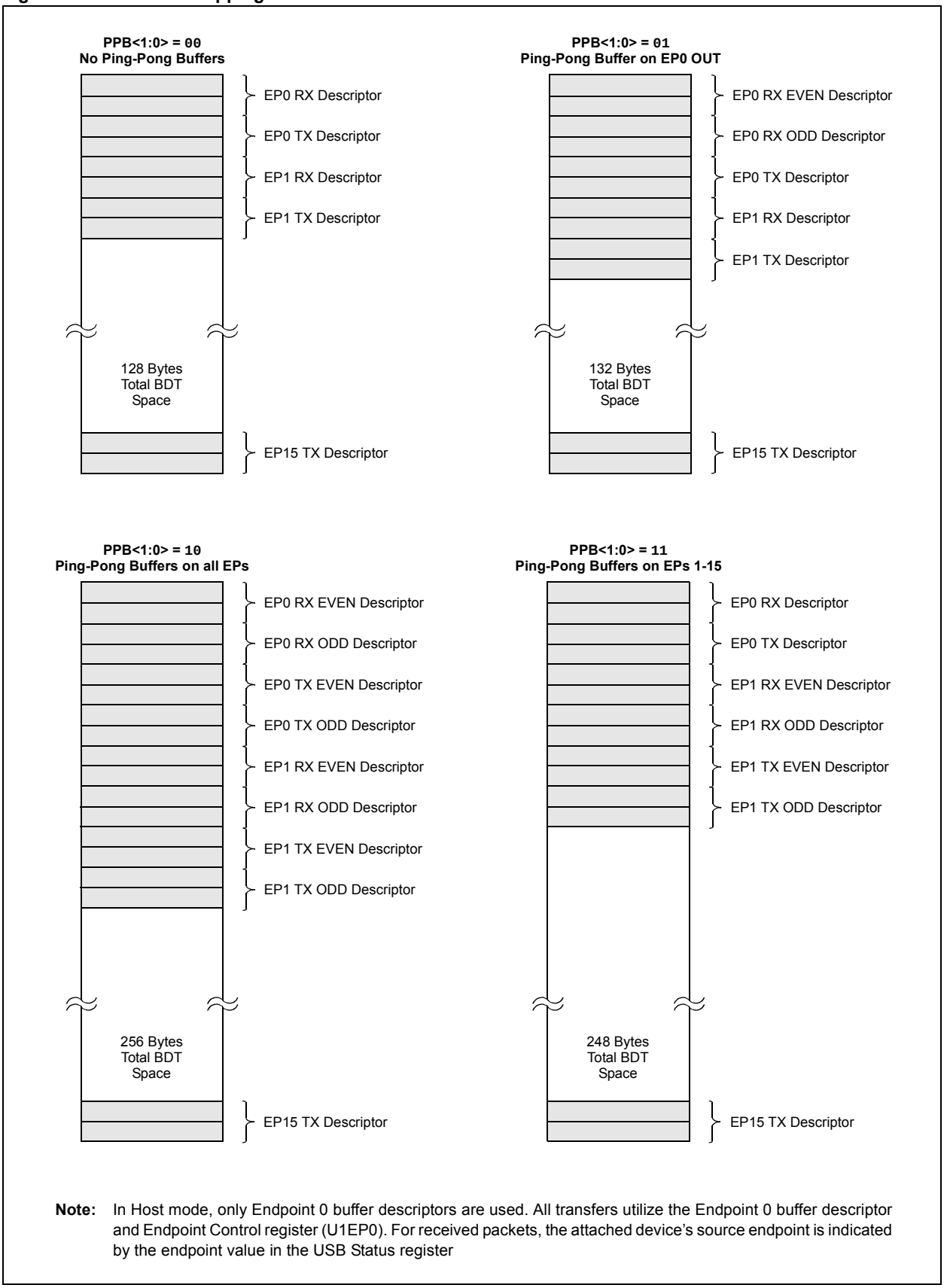
Note: The BDT Base Address must be 512-byte boundary aligned, therefore U1BDTP1 should always be even.

The USB module calculates a buffer’s location in RAM using the BDT Pointer registers. The base of the BDT is held in the U1BDTP1 register. The address of the desired buffer is then found by using the endpoint number, the type (RX/TX) and the ODD/EVEN bit to index into the BDT. The address held by this entry is the address of the desired data buffer.

Each of the 16 endpoints owns up to two descriptor pairs: two for packets to transmit and two for packets received. Each pair manages one or two buffers, an EVEN and an ODD, requiring a maximum of 64 descriptors (16 * 2 * 2).

An EVEN and ODD buffer for each direction allows the CPU to access data in one buffer while the USB module transfers data to or from the other buffer. The USB module alternates between buffers, clearing the UOWN bit automatically when the transaction for that buffer is complete. The use of alternating buffers maximizes data throughput by allowing CPU data access in parallel with the data transfer. This technique is referred to as ping-pong buffering. Ping-pong buffering is optional and must be configured with the PPB<1:0> bits in the U1CNFG1 register. Figure 27-2 illustrates how the endpoints are mapped in the BDT.

Figure 27-2: Buffer Mapping



Note: In Host mode, only Endpoint 0 buffer descriptors are used. All transfers utilize the Endpoint 0 buffer descriptor and Endpoint Control register (U1EP0). For received packets, the attached device's source endpoint is indicated by the endpoint value in the USB Status register

27.3.2.2.1 Endpoint Control

Each endpoint is controlled by an Endpoint Control register, U1EPn, that configures the transfer direction, the handshake and the stalling properties of the endpoint. The Endpoint Control register also allows support of control transfers.

Note:	In Host mode, Endpoint 0 has additional bits for auto-retry and hub support.

27.3.2.2.2 Host Endpoints

The host performs all transactions though a single endpoint (Endpoint 0). All other endpoints should be disabled and other endpoint buffers are not be used.

27.3.2.2.3 Device Endpoints:

Endpoint 0 must be implemented to allow a USB device to be enumerated. Devices typically implement additional endpoints to transfer data.

27.3.2.3 BUFFER MANAGEMENT

Because the buffers are shared between the PIC24F and the USB module, a simple semaphore mechanism is used to distinguish current ownership of the descriptor and associated buffers in memory. This semaphore mechanism is implemented by the UOWN bit in each buffer descriptor.

The USB module clears the UOWN bit automatically when the transaction for that buffer is complete. When the UOWN bit is clear, the descriptor is owned by the PIC24F. The PIC24F may modify the descriptor and buffer at its discretion.

Software must configure the BDT entry for the next transaction, then set the UOWN bit to return control to the USB module.

A buffer descriptor is only valid if the corresponding endpoint has been enabled using the U1EPn register. The BDT is implemented in data memory and the buffer descriptors are not reset to a known state. The user needs to initialize the buffer descriptors prior to enabling them through the U1EPn. As a minimum, the UOWN bits must be cleared prior to being enabled.

In Host mode, BDT initialization is not required until the U1TOK register is written, triggering a transfer.

27.3.2.3.1 Buffer Descriptor Format

The buffer descriptor has two formats.

The buffer descriptor format when the software writes the descriptor and hands it to hardware is shown in Table 27-3.

The buffer descriptor format when the hardware writes the descriptor and hands it back to software is shown in Table 27-4.

Table 27-2: BDT Address Generation

BDTBA<15:9>	ENDPOINT<3:0>	DIR	PPBI	FSOTG
15:9	8:5	4	3	2:0

bit 15-9 **BDTBA<15:9>:** BDT Base Address bits
The 7-bit value is made up of the contents of the U1BDTP1 register.

bit 8-5 **ENDPOINT<3:0>:** Transfer Endpoint Number bits
0000 = Endpoint 0
0001 = Endpoint 1
....
1110 = Endpoint 14
1111 = Endpoint 15

bit 4 **DIR:** Transfer Direction bit
1 = Transmit: SETUP/OUT for host, IN for function
0 = Receive: IN for host, SETUP/OUT for function

bit 3 **PPBI:** Ping-Pong Pointer bit
1 = ODD buffer
0 = EVEN buffer

bit 2-0 **Manipulated by the USB Module**

Table 27-3: USB Buffer Descriptor Format: Software ≥ Hardware

31	30	29	28	27	26	25									16
UOWN	DTS(1)	—		DTSEN	BSTALL	BYTE_COUNT<9:0>									
15															0
BUFFER_ADDRESS<15:0>															

bit 31 **UOWN:** USB Own bit

1 = The USB module owns the BD and its corresponding buffer; the CPU must not modify the BD or the buffer

0 = The CPU owns the BD and its corresponding buffer; the USB module ignores all other fields in the BD

bit 30 **DTS:** Data Toggle Packet bit(1)

1 = Transmit a DATA1 packet or check received, PID = DATA1 if DTSEN = 1

0 = Transmit a DATA0 packet or check received, PID = DATA0 if DTSEN = 1

bit 27 **DTSEN:** Data Toggle Synchronization Enable bit(2)

1 = Data toggle synchronization is enabled – data packets with incorrect sync value will be ignored

0 = No data toggle synchronization is performed

bit 26 **BSTALL:** Buffer Stall Enable bit

1 = Buffer STALL enabled; STALL handshake issued if a token is received that would use the BD in the given location (UOWN bit remains set, BD value is unchanged). The corresponding EPSTALL bit will get set on any STALL handshake.

0 = Buffer STALL disabled

bit 25-16 **BYTE_COUNT<9:0>:** Byte Count bits

The byte count represents the number of bytes to be transmitted or the maximum number of bytes to be received during a transfer.

bit 15-0 **BUFFER_ADDRESS:** Buffer Address bit

The starting point address of the endpoint packet data buffer (see Table 27-2).

Note 1: This bit is ignored unless BDnST<DTSEN> = 1.

2: The expected value of DATA PID (DATA0/DATA1) is specified in the DATA0/1 field.

Table 27-4: USB Buffer Descriptor Format: Hardware -> Software

31	30	29	28	27	26	25									16
UOWN	DTS(1)	PID<3:0>				BYTE_COUNT<9:0>									
15															**0**
BUFFER_ADDRESS<15:0>															

bit 31 **UOWN:** USB Own bit

1 = The USB module owns the BD and its corresponding buffer; the CPU must not modify the BD or the buffer

0 = The CPU owns the BD and its corresponding buffer; the USB module ignores all other fields in the BD

bit 30 **DTS:** Data Toggle Packet bit(1)

1 = DATA1 packet received

0 = DATA0 packet received

bit 29-26 **PID<3:0>:** Packet Identifier bits

The current token PID when a transfer completes. The values written back are the token PID values from the USB Specification: 0x1 for an OUT token, 0x9 for an IN token or 0xd for a SETUP token.

In Host mode, this field is used to report the last returned PID or a transfer status indication. The possible values returned are: 0x3 DATA0, 0xb DATA1, 0x2 ACK, 0xe STALL, 0xa NAK, 0x0 Bus Time-out, 0xf Data Error.

bit 25-16 **BYTE_COUNT<9:0>:** Byte Count bits

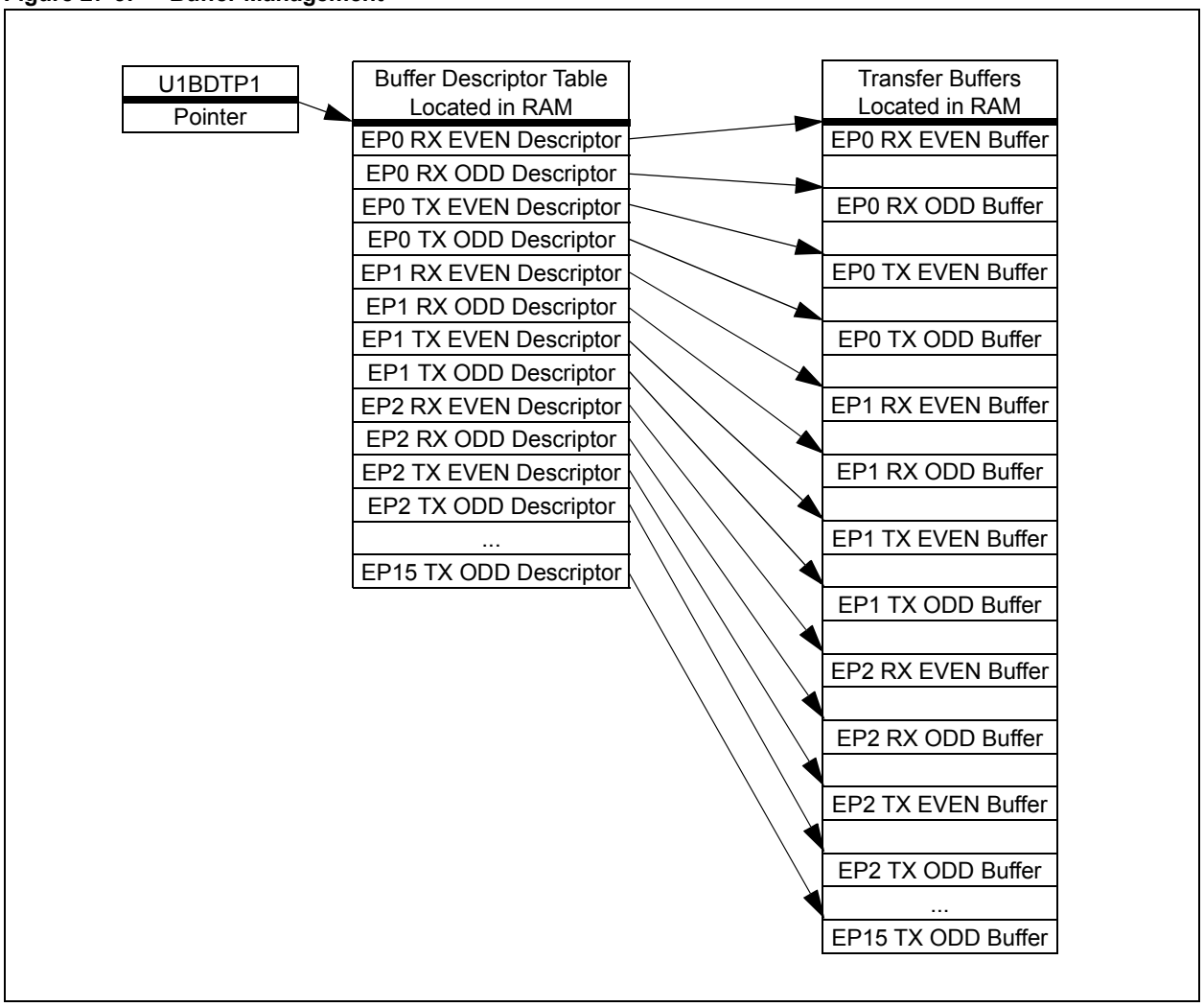
The byte count reflects the actual number of bytes received or transmitted.

bit 15-0 **BUFFER_ADDRESS<15:0>:** Buffer Address bits

The starting point address of the endpoint packet data buffer.

Note 1: This bit is unchanged on an outgoing packet.

Figure 27-3: Buffer Management



27.3.2.4 BUFFER DESCRIPTOR CONFIGURATION

The UOWN, DTSEN and BSTALL bits in each BDT entry control the behavior of the endpoint.

Setting the DTSEN bit will enable the USB module to perform data toggle synchronization. If a packet arrives with an incorrect DTS, it will be ignored and the buffer will remain unchanged. Note that when DTSEN is enabled, and the DATA bit is mismatched, the device will NAK the host to resynchronize.

Setting the BSTALL bit will cause the USB to issue a STALL handshake if a token is received by the SIE that would use the Buffer Descriptor (BD) in this location. The corresponding EPSTALL bit gets set and a STALLIF interrupt is issued. The BD is not consumed by the USB module (the UOWN bit remains set and the rest of the BD values are unchanged) when the BSTALL bit is set. A SETUP token to the stalled endpoint automatically clears the corresponding BSTALL bit.

The byte count represents the total number of bytes that will be transmitted or received. Valid byte counts are from 0 to 1023. For all endpoint transfers, the byte count is updated by the USB module with the actual number of bytes transmitted or received once the transfer is completed. If the number of bytes received exceeds the corresponding byte count value written by the firmware, the overflow bit will be set and the data will be truncated to fit the size of the buffer as given in the BTD.

27.3.3 Hardware Interface

27.3.3.1 PULL-UP AND PULL-DOWN RESISTORS

To indicate if a USB application is operating as a device or a host, the USB Specification requires the use of pull-up and pull-down resistors on the D+ and D- data lines. The combination of pull-ups and pull-downs is used to signal both the operating mode and device's speed. The specification does not distinguish between on-chip or external resistors. Since many applications are either Device or Host only, the use of fixed external resistors may be a practical solution.

PIC24F family devices have on-chip pull-up and pull-down resistors on the D+/D- line pair. They can be enabled or disabled under software control, making it convenient to change between Host and Device mode under software control.

The built-in, 1.5 kΩ pull-up resistors are available when the microcontroller is operating in Device mode. They are engaged by setting the DPPULUP or DMPULUP bits, respectively. (U1OTGCON<7,6>). Enabling the pull-up on D+ is used to signal the bus that the PIC® MCU-based application is operating in Full-Speed Device mode. Similarly, Low-Speed Device mode would be selected by enabling the pull-up on D-; however, this configuration is not used, since Low-Speed Device mode is not available for these devices.

The built-in, 15 kΩ pull-down resistors are engaged by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5,4>). These are used together to signal to the bus that the microcontroller is operating in Host mode.

27.3.3.2 POWER SUPPLY REQUIREMENTS

In general terms, PIC24F family devices support three broad power categories based on the USB mode:

- Device (functions as a B-device)
- Host (functions as an A-device)
- OTG Dual Role (dynamically switches between A and B-device roles)

Many USB applications will likely have several different sets of power requirements and configuration. For example, an OTG application may function at different times as either a host or a device; while functioning as a device, the application may be able to switch between being bus-powered and self-powered. Each mode has distinct power supply requirements and requires different power connections.

27.3.3.2.1 Device Power Modes

When operating in Device mode, there are three power modes that are most commonly encountered:

- Bus Power Only
- Self-Power Only
- Dual Power with Self-Power Dominance

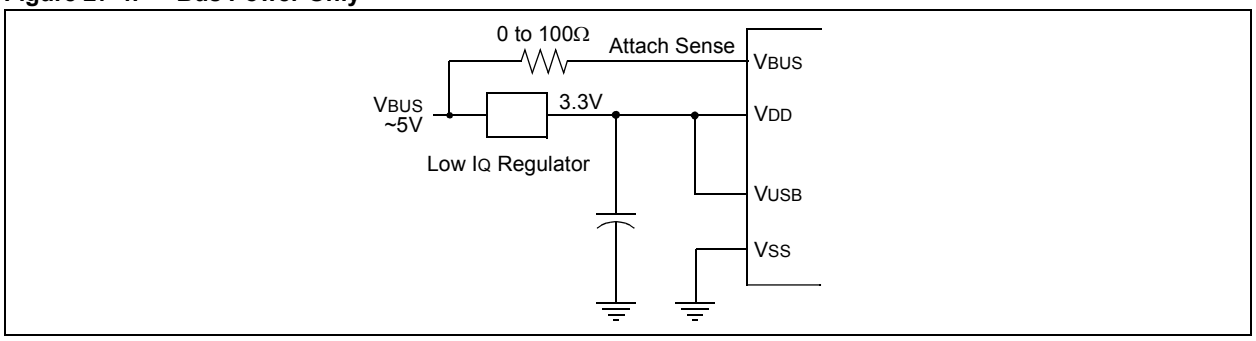
While these modes are discussed here, keep in mind that other device power modes may be used.

In Bus Power Only mode, all power for the application is drawn from the USB (Figure 27-4). This is effectively the simplest power method for the device.

To meet the inrush current requirements of the USB 2.0 Specification, the total effective capacitance appearing across VBUS and ground must be greater than 1 µF, but no more than 10 µF. If not, some kind of inrush limiting is required. For more details, see Section 7.2.4 of the USB 2.0 Specification, as well as all Engineering Change Notice (ECN) updates to the specification.

According to the USB 2.0 Specification and ECNs, all USB devices must also support a Low-Power Suspend mode. In the USB Suspend mode, devices must consume no more than 2.5 mA from the 5V VBUS line of the USB cable. The host signals the USB device to enter the Suspend mode by stopping all USB traffic to that device for more than 3 ms. This condition will cause the IDLEIF bit in the U1IR register to become set. During the USB Suspend mode, the D+ pull-up resistor must remain active in order to remain "attached" to the host. This will consume some of the allowed suspend current budget.

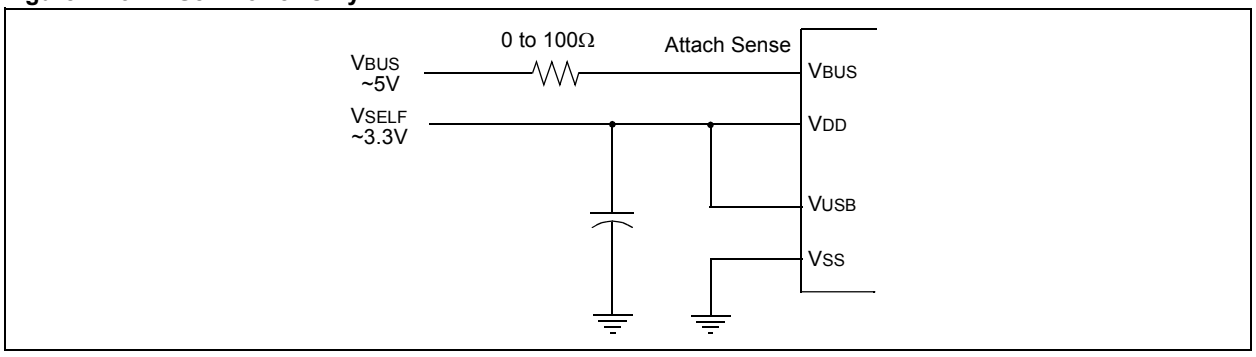
Figure 27-4: Bus Power Only



In Self-Power Only mode, the USB application provides its own power, with very little power being pulled from the bus (Figure 27-5). In this example, an attach indication is added to indicate when the USB has been connected and the host is actively powering VBUS.

To meet compliance specifications, the D+ pull up resistor (signalling device attachment) should not be enabled until the host actively drives VBUS high. The internal VBUS comparators (such as SESVD), general purpose internal comparators, or even general purpose analog or digital I/O pins can be used for VBUS sensing. The application should never source any current onto the 5V VBUS pin of the USB cable.

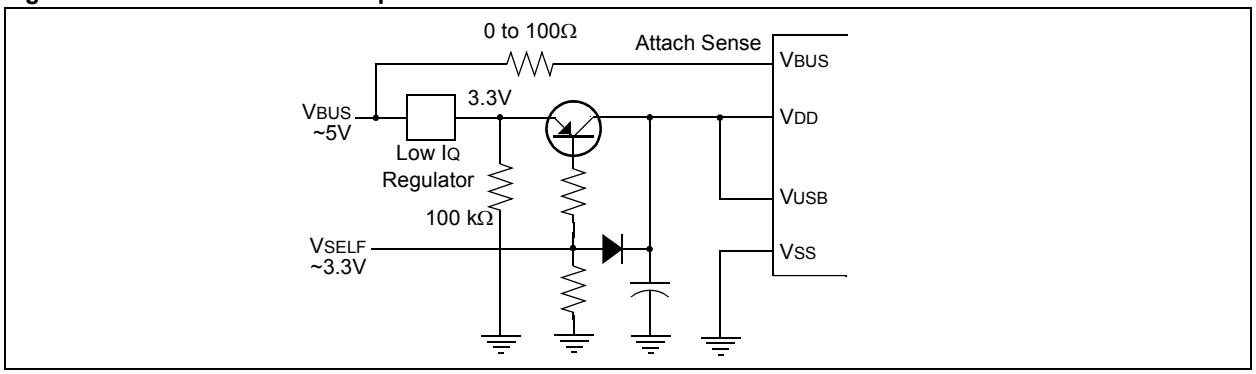
Figure 27-5: Self-Power Only



Some applications may require a dual power option, allowing the application to use internal power primarily and switch to power from the USB when no internal power is available. Figure 27-6 shows a simple implementation of Dual Power with Self-Power Dominance mode, which automatically switches between Self-Power Only and USB Bus Power Only modes.

Dual power devices must meet all of the special requirements for inrush current and Suspend mode current previously described, and must not enable the USB module until VBUS is driven high. Additionally, dual power devices must never source current onto the 5V VBUS pin of the USB cable.

Figure 27-6: Dual Power Example



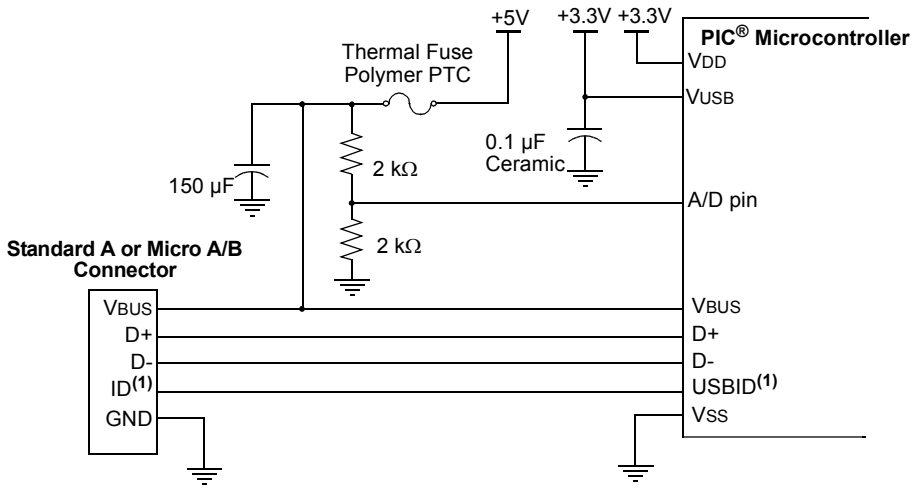
Note: USB Specification 2.0 limits the power drawn by devices from the USB VBUS supply from the host. A USB device should consume no more than 100 mA, unless it has requested more in the USB configuration descriptor, and has been configured. A device may request up to 500 mA maximum.

27.3.3.2.2 Host and OTG Power Modes

In Host mode, as well as Host mode in On-The-Go operation, the USB 2.0 Specification requires that the host application supply power on VBUS. Since the microcontroller is running below VBUS and is not able to source sufficient current, a separate power supply must be provided. Operation as a host requires a power supply for the PIC24F, the USB transceiver and a 5V nominal supply for the USB VBUS. The power supply must be able to deliver 100 mA, or up to 500 mA, depending on the requirements of devices in the TPL. The application dictates if the VBUS power supply can be disabled or disconnected from the bus by the PIC24F application.

When the application is always operating in Host mode, a simple circuit can be used to supply VBUS and regulate current on the bus (Figure 27-7).

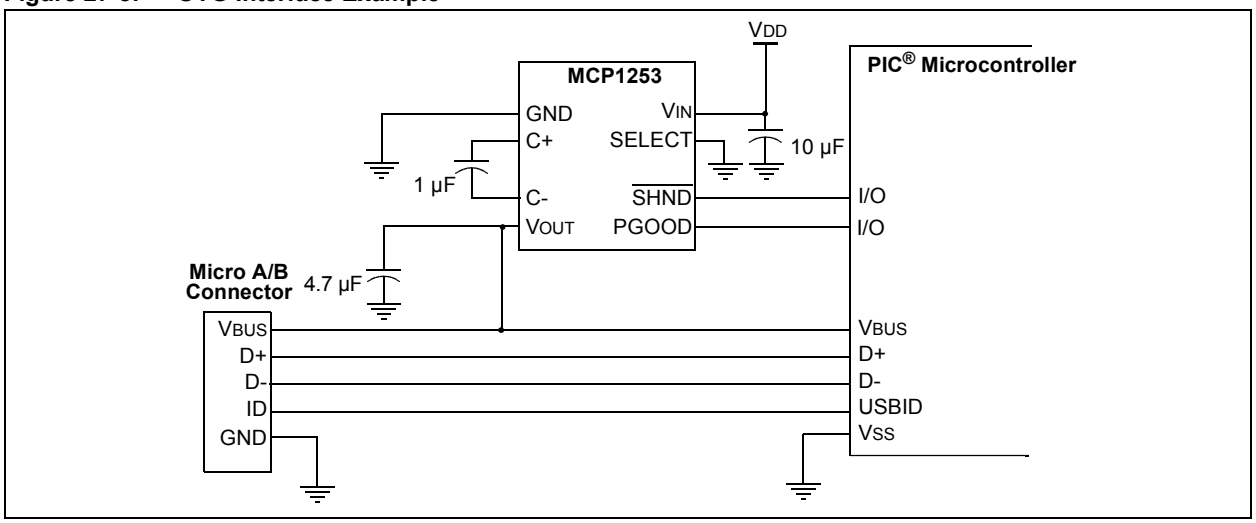
Figure 27-7: Host Interface Example



Note 1: The ID/USBID signal is only used in OTG modes. In non-OTG Host and Device modes, the ID/USBID signal is not used and may be left unconnected.

Operation as an OTG dual role requires a power supply for the PIC24F, the USB transceiver and a switchable 5V nominal supply for the USB VBUS. When acting as an A-device, power must be supplied to VBUS. The power supply must be able to deliver 8 mA, 100 mA, or up to 500 mA, depending on the requirements of the TPL. It is also necessary to be able to turn VBUS on or off as needed, as the microcontroller switches between Device and Host modes. A typical example using an external charge pump is shown in Figure 27-8.

Figure 27-8: OTG Interface Example



27.3.3.3 EXTERNAL INTERFACE SUPPORT

In some applications, it is required to isolate the USB interface from the rest of the system. PIC24F family devices provide a complete interface of 14 pins to control external transceiver and bus power supplies; module support is built-in to facilitate these type of designs.

27.3.3.3.1 External Transceiver Control

External transceivers are designed to provide a complete bus interface between the application controller and the serial bus. The transceiver converts logic level differential data from the SIE to bus level differential signals and convert differential bus signals back to data. Transceivers also include built-in pull-up and pull-down resistors to signal the device's mode and speed to the bus.

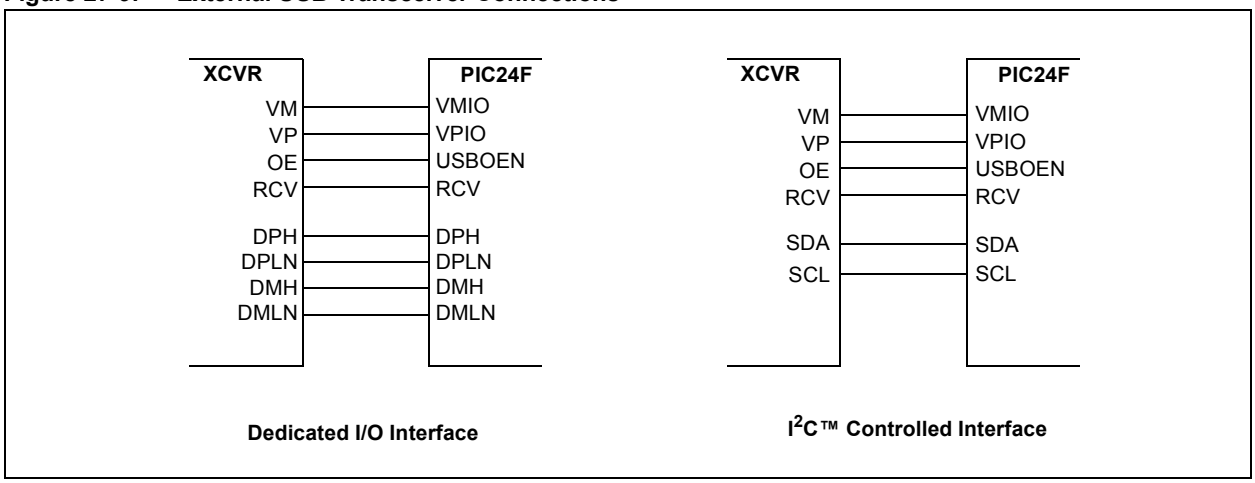
USB OTG transceivers are available with several different types of interface. PIC24F family devices support two of the most common: Dedicated I/O (transceiver functions are controlled by dedicated controller pins) and I^{2}C™ controlled (functions are controlled over an I^{2}C serial bus).

External transceiver support is enabled by setting the UTRDIS bit (U1CNFG2<0> = 1). The EXTI2CEN bit (U1CNFG2<3>) selects between dedicated I/O interfaces (EXTI2CEN = 0) and I^{2}C controlled interfaces (EXTI2CEN = 1).

In the Dedicated I/O mode, setting or clearing the DPPULUP, DPPULDWN, DMPULUP, or DMPULDWN pins (U1OTGCON<7:4>) directly controls the DPH, DPLN, DMH, and DMLN pins respectively. These, in turn, are used to control the pull-up/pull-down resistors in the interface. In I^{2}C controlled mode, the I^{2}C module controls the transceiver, including the pull-up/pull-down resistors. This mode can also be used to interface with any other transceiver interface.

Differential data and output enable connections are identical between the two modes (Figure 27-9).

Figure 27-9: External USB Transceiver Connections



27.3.3.3.2 VBUS Boost Regulator Interface

The VBUSON output can be used to control an off-chip 5V VBUS boost regulator. The VBUSON pin is controlled by the VBUSON bit (U1OTGCON<3>).

When operating as a USB host, either as an A-device in an OTG configuration or as an embedded host, VBUS must be supplied to the attached device. PIC24F family devices have an internal VBUS boost assist to help generate the required 5V VBUS from the available voltages on the board. This is comprised of a simplified PWM output to control a Switch mode power supply circuit, with built-in comparators for current-limiting and output voltage monitoring. The VCPCON pin is the PWM output. The VBUSST pin is the current-limiting monitor, while VBUS is the output voltage monitor.

To use the boost assist interface, connect the VBUS pin to the final output of the switching system and the VBUS line of the USB connector. This pin measures the output voltage, which will turn the PWM on and off to maintain the desired voltage. When the VBUSST pin reaches 190 mV (typical), the output of the PWM will stop switching until it has dropped below that point again.

To enable voltage generation:

1. Verify that the USB module is powered (U1PWRC<0> = 1) and that the VBUS discharge is disabled (U1OTGCON<0> = 0).
2. Set the PWM period (U1PWMRRS<7:0>) and duty cycle (U1PWMRRS<15:8>) as required.
3. Select the required polarity of the output signal based on the configuration of the external circuit with the PWMPOL bit (U1PWMCON<9>).
4. Select the desired target voltage using the VBUSCHG bit (U1OTGCON<1>).
5. Enable the PWM counter by setting the CNTEN bit to '1' (U1PWMCON<8>).
6. Enable the PWM module by setting the PWMEN bit to '1' (U1PWMCON<15>).
7. Enable the VBUS generation circuit (U1OTGCON<3> = 1).

27.3.3.3.3 External VBUS Comparator Inputs

The external comparator interface allows an external circuit with the appropriate logic to monitor the bus for signalling conditions. The use of external comparators is enabled by setting the UVCMPDIS bit (U1CNFG2<1>). This disables the internal VBUS comparators and removes the need to connect the VBUS voltage to the VBUS pin of the microcontroller.

Most devices provide an option to select either a 2-pin or a 3-pin interface. The 2-pin interface uses the VCMPST1 and VCMPST2 digital inputs. The 3-pin interface uses the VBUSVLD, SESSVLD and SESSEND digital inputs. (Note that the VCMPST1/VBUSVLD functions are multiplexed to the same pins, as are the VCMPST2/SESSVLD functions.) The interface configuration is determined by the UVCMPSET bit (U1CNFG2<5>).The default configuration (UVCMPSET = 0) is the 2-pin interface; setting UVCMPSET selects the 3-pin interface.

To function properly, the digital comparator outputs must conform to the truth table shown in Table 27-5.

Note:	Certain devices only implement the 2-pin interface. In these cases, the UVCMPSET bit is unimplemented. Refer to the specific device data sheet for information.

Table 27-5: Bus Conditions for External Comparator Interface

3-Pin Interface (UVCMPSEL = 1)			
VBUSVLD	**SESSVLD**	**SESSEND**	**Bus Condition**
0	0	1	VBUS < VB_SESS_END
0	0	0	VB_SESS_END < VBUS < VA_SESS_VLD
0	1	0	VA_SESS_VLD < VBUS < VA_VBUS_VLD
1	1	0	VBUS > VA_VBUS_VLD
2-Pin Interface (UVCMPSEL = 0)			
VCMPST1	**VCMPST2**		**Bus Condition**
0	0		VBUS < VB_SESS_END
1	0		VB_SESS_END < VBUS < VA_SESS_VLD
0	1		VA_SESS_VLD < VBUS < VA_VBUS_VLD
1	1		VBUS > VA_VBUS_VLD

27.3.3.4 USB TRANSCEIVER CURRENT CONSUMPTION

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.

When operating the microcontroller in USB Device mode, data patterns that consist of “IN” traffic consume far more current than “OUT” traffic. IN traffic requires the USB device to drive the USB cable, whereas OUT traffic requires that the host drive the USB cable.

The data that is sent across the USB cable is NRZI encoded. In the NRZI encoding scheme, bits set to ‘0’ cause a toggling of the output state of the transceiver (either from a “J” state to a “K” state, or vise versa). With the exception of the effects of bit stuffing, NRZI encoded ‘1’ bits do not cause the output state of the transceiver to change. Therefore, IN traffic consisting of data bits of value, ‘0’, cause the most current consumption, as the transceiver must charge/discharge the USB cable in order to change states.

More details about NRZI encoding and bit stuffing can be found in Section 7.1 of the USB 2.0 Specification. However, knowledge of such details is not required to make USB applications. Among other things, the SIE handles bit stuffing/unstuffing, NRZI encoding/decoding and CRC generation/checking in hardware.

The total transceiver current consumption will be application-specific.To help estimate how much current actually may be required in full-speed applications, Equation 27-2 can be used. Example 27-2 shows how this equation can be used for a theoretical application.

Note:	The calculated value should be considered an approximation and additional guard band or application-specific product testing is recommended. The transceiver current is in addition to the rest of the current consumed by the microcontroller during normal operation.

Equation 27-2: Estimating USB Transceiver Current Consumption

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

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

PZERO: Percentage (in decimal) of the IN traffic bits sent by the PIC® MCU 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. On the host or hub end of the USB cable, 15 kΩ nominal resistors (14.25 kΩ to 24.8 kΩ) are present which pull both the D+ and D- lines to ground. During bus Idle conditions (such as between packets or during USB Suspend mode), this results in up to 218 μA of quiescent current drawn at 3.3V. IPULLUP is also dependent on bus traffic conditions and can be as high as 2.2 mA when the USB bandwidth is fully utilized (either IN or OUT traffic) for data that drives the lines to the “K” state most of the time.

Example 27-2: Calculating USB Transceiver Current

For this example, the following assumptions are made about the application:

- 3.3V will be applied to V_{USB} and V_{DD} with the core voltage regulator enabled.
- This is a full-speed application that uses one interrupt IN endpoint that can send one packet of 64 bytes every 1 ms, with no restrictions on the values of the bytes being sent. The application may or may not have additional traffic on OUT endpoints.
- A regular USB "B" or "mini-B" connector will be used on the application circuit board.

In this case, P_{ZERO} = 100% = 1, because there should be no restriction on the value of the data moving through the IN endpoint. All 64 kBps of data could potentially be bytes of value, 00h. Since '0' bits cause toggling of the output state of the transceiver, they cause the USB transceiver to consume extra current charging/discharging the cable. In this case, 100% of the data bits sent can be of value '0'. This should be considered the "max" value, as normal data will consist of a fair mix of '1's and '0's.

This application uses 64 kBps for IN traffic out of the total bus bandwidth of 1.5 MBps (12 Mbps), therefore:

$$Pin = \frac{64\ kBps}{1.5\ MBps} = 4.3\% = 0.043$$

Since a regular "B" or "mini-B" connector is used in this application, the end user may plug in any type of cable up to the maximum allowed 5m length. Therefore, we use the worst-case length:

L_{CABLE} = 5 meters

Assume I_{PULLUP} = 2.2 mA. The actual value of I_{PULLUP} will likely be closer to 218 μA, but allow for the worst case. USB bandwidth is shared between all the devices which are plugged into the root port (via hubs). If the application is plugged into a USB 1.1 hub that has other devices plugged into it, your device may see host to device traffic on the bus, even if it is not addressed to your device. Since any traffic, regardless of source, can increase the I_{PULLUP} current above the base 218 μA, it is safest to allow for the worst case of 2.2 mA.

Therefore:

$$I_{XCVR} = \frac{(40\ mA \bullet 3.3V \bullet 1 \bullet 0.043 \bullet 5m)}{(3.3V \bullet 5m)} + 2.2\ mA = 3.9\ mA$$

27.3.3.5 CLOCK REQUIREMENTS

For proper USB operation, the USB module must be clocked with a 48 MHz clock. This clock source is used to generate the timing for USB transfers; it is not the clock source for the SIE. The SIE is clocked from the same source as the CPU.

The USB module clock is derived from the Primary Oscillator (POSC) for USB operation. A USB PLL and input prescalers are provided to allow 48 MHz clock generation from a wide variety of input frequencies. The USB PLL allows the CPU and the USB module to operate at different frequencies while using POSC as a clock source.

27.3.4 Module Initialization

This section describes the steps that must be taken to properly initialize the OTG USB module.

27.3.4.1 ENABLING THE USB HARDWARE

In order to use the USB peripheral, software must set the USBPWR (U1PWRC<0>) to '1'. This may be done in start-up boot sequence.

USBPWR is used to:

- Start the USB clock.
- Hold the USB module in its inactive state when USBPWR= 0.
- Select USB as the owner of the necessary I/O pins.
- Enable the USB transceiver.
- Enable the USB comparators.

The USB core logic and registers will be reset when the USBPWR bit is cleared. This requires the USB module initialization process to be performed whenever the USB module is enabled. Any configuration accesses targeting the USB logic will be stalled until the Reset is complete.

27.3.4.2 INITIALIZING THE BUFFER DESCRIPTOR TABLE

All descriptors for a given endpoint and direction must be initialized prior to enabling the endpoint. After a Reset, all endpoints start transfers with the EVEN buffer for transmit and receive directions. The buffers are reset when the module power is disabled (U1PWRC<USBPWR>) or after a device Reset.

TX descriptors must be written with the UOWN bit cleared to '0' (owned by software). All other TX descriptor setups may be performed any time prior to setting the UOWN bit.

RX descriptors must be fully initialized to receive data. This means that memory must be reserved for received packet data. The pointer to that memory and the size reserved in bytes must be written to the descriptor. The RX descriptor UOWN bit should be initialized to '1' (owned by hardware). The DTSEN and BSTALL bits should also be configured appropriately.

If an RX transaction is received and the RX descriptor UOWN bit is '0' (owned by software), then the USB module will return a NAK handshake to the host. This will likely cause the host to retry the transaction. In Host mode, the BDT does not need to be initialized until the host begins a transfer.

27.4 DEVICE MODE OPERATION

The following section describes how to perform a common device mode task. In Device mode, USB transfers are performed at the transfer level. The USB module automatically performs the status phase of the transfer.

27.4.1 Enabling Device Mode

1. Reset the Ping-Pong Buffer Pointers by setting, and then clearing, the PPBRST bit (U1CON<1>).
2. Disable all USB interrupts (U1IE = 0 and U1EIE = 0).
3. Clear any existing interrupt flags (U1IR = 0xFF and U1EIR = 0xFF).
4. Verify that VBUS is present (non OTG devices only).
5. Enable the USB module by setting USBEN (U1CON<0>).
6. Set OTGEN (U1OTGCON<2>) to '1'.
7. Enable the Endpoint 0 buffer to receive the first setup packet by setting the EPRXEN and EPHSHK bits (U1EP0<3,0>).
8. Power-up the USB module by setting the USBPWR bit (U1PWRC<0>).
9. Enable the D+ pull-up resistor to signal an attach by setting the DPPULUP bit (U1OTGCON<7> = 1).

27.4.2 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. 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 will automatically transmit the data in the buffer. Upon completion, the module will update the status register (BDnSTAT) and set the transfer complete interrupt bit, TRNIF (U1IR<3>).

27.4.3 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 will automatically receive the data the host sent into the buffer. Upon completion, the module will update the status register (BDnSTAT) and set the Transfer Complete Interrupt bit, TRNIF (U1IR<3>).

27.5 HOST MODE OPERATION

In Host mode, only Endpoint 0 is used. Since the host initiates all transfers, the buffer descriptor does not require initialization. The buffer descriptors must be configured before a transfer is initiated, which is done by writing the U1TOK register.

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 Endpoint 0 Control register (U1EP0) and buffer descriptors.

27.5.1 Enable Host Mode and Discover a Connected Device

1. Enable Host mode by setting the HOSTEN bit (U1CON<3> = 1).
2. Enable the D+ and D- pull-down resistors by setting DPPULDNW and DMPULDWN (U1OTGCON<5:4> = 1). Disable the D+ and D- pull-up resistors by clearing DPPULUP and DMPULUP (U1OTGCON<7:6> = 0).
3. SOF generation will begin. The SOF counter is loaded with 12,000. Disable the Start-of-Frame packet generation by writing '0' to the SOFEN bit (U1CON<0>).
4. Enable the device attach interrupt by setting ATTACHIE (U1IE<6>).
5. Wait for the device Attach Interrupt, ATTACHIF (U1IR<6>). This is signaled by the USB device changing the state of D+ or D- from '0' to '1' (SE0 to JSTATE). After it occurs, wait for the device power to stabilize (10 ms minimum, 100 ms recommended).
6. Check the state of the JSTATE and SE0 bits in the control register (U1CON). If JSTATE (U1CON<7>) is '0', then the connecting device is low speed; otherwise, the device is full speed.
7. If the connecting device is low speed, then set the Low-Speed Enable bit, LSPDEN, in the address register (U1ADDR<7>) and the Low-Speed bit, LSPD, in the Endpoint 0 Control register (U1EP0<7>). If the device is full speed, clear these bits.
8. Reset the USB device by sending the Reset signaling for at least 50 ms (U1CON<4> = 1). After 50 ms, terminate the Reset (U1CON<4> = 0).
9. Enable SOF packet generation to keep the connected device from going into suspend by setting SOFEN (U1CON<0>).
10. Wait 10 ms for the device to recover from Reset.
11. Perform enumeration as described by Chapter 9 of the USB 2.0 Specification.

27.5.2 Complete a Control Transaction to a Connected Device

1. Complete all steps to discover a connected device.
2. Set up the Endpoint Control register for bidirectional control transfers, U1EP0<4:0> = 0x0D.
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 current (EVEN or ODD) TX EP0 Buffer Descriptor (BD) to transfer the 8 bytes of command data for a device framework command (i.e., a `GET DEVICE DESCRIPTOR`).
   a) Set the BD status (BD0STAT) to 0x8008 – UOWN bit set, byte count of 8.
   b) Set the BD data buffer address (BD0ADR) to the starting address of the 8-byte memory buffer containing the command.
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 the token register with a SETUP to Endpoint 0, the target device's default control pipe (U1TOK = 0xD0). This will initiate a SETUP token on the bus followed by a data packet. The device handshake will be returned in the PID field of BD0STAT after the packets complete. When the module updates BD0STAT, a transfer done interrupt will be asserted (U1IR<3>). This completes the setup phase of the setup transaction as referenced in Chapter 9 of the USB Specification.
7. To initiate the data phase of the setup transaction (i.e., get the data for the `GET DEVICE DESCRIPTOR` command), set up a buffer in memory to store the received data.
8. Initialize the current (EVEN or ODD) RX or TX (RX for IN, TX for OUT) EP0 BD to transfer the data.
   a) Set the BD status (BD0STAT) to 0xC040 – UOWN bit to '1', data toggle (DTS) to DATA1 and byte count to the length of the data buffer (in this case 64 or 0x40).
   b) Set the BD data buffer address (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., an IN token for a `GET DEVICE DESCRIPTOR` command (U1TOK = 0x90)). This will initiate an IN token on the bus followed by a data packet from the device to the host. When the data packet completes, the BD0STAT is written and a transfer done interrupt will be asserted (U1IR<TRNIF>). For control transfers with a single packet data phase, this completes the data phase of the setup transaction as referenced in Chapter 9 of the USB Specification. If more data needs to be transferred, return to step 8.
10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (EVEN or ODD) TX EP0 BD to transfer the status data:
    a) Set the BD status (BD0STAT) to 0x8000 – UOWN bit to '1', data toggle (DTS) to DATA0 and byte count to '0'.
    b) Set the BDT buffer address field to the start address of the data buffer.
12. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., an OUT token for a `GET DEVICE DESCRIPTOR` command, (U1TOK = 0x10). This will initiate an OUT token on the bus, followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device and a transfer done interrupt will be asserted (U1IR<3>). This completes the status phase of the setup transaction as described in Chapter 9 of the USB Specification.

27.5.3 Send a Full-Speed Bulk Data Transfer to a Target Device

1. Complete all steps to discover and configure a connected device.
2. Write the EP0 Control register (U1EP0) to 0x1D to enable transmit and receive transfers with handshaking enabled. If the target device is a low-speed device, also set the Low-Speed Enable bit, LSPD (U1EP0<7>). If you want the hardware to automatically retry indefinitely, if the target device asserts a NAK on the transfer, clear the Retry Disable bit, RETRYDIS (U1EP0<7>).
3. Set up the current (EVEN or ODD) TX EP0 BD 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 the Token register (U1TOK) with an OUT token to the desired endpoint. This triggers the module's transmit state machines to begin transmitting the token and the data.
6. Wait for the Transfer Done Interrupt, TRNIF (U1IR<3>). This will indicate that the BD has been released back to the microprocessor and the transfer has completed. If the retry disable bit is set, the handshake (ACK, NAK, STALL or ERROR (0xF)) will be returned in the BD PID field. If a STALL interrupt occurs, then 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> = 1).
7. Once the Transfer Done Interrupt, TRNIF, occurs, 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 set-up phase. It is not recommended to change these settings while the module is enabled.

27.5.3.1 USB ENABLE/MODE BITS

The USB mode of operation is controlled by three enable bits: OTGEN (U1OTGCON<2>), HOSTEN (U1CON<3>) and USBEN/SOFEN (U1CON<0>).

OTGEN selects whether the peripheral is to act as an On-The-Go part (OTGEN = 1) or not. On-The-Go devices support Session Request Protocol (SRP) and Host Negotiation Protocol (HNP) in hardware with firmware management.

HOSTEN controls whether the part is acting in the role of USB host or USB device. Note that this role may change dynamically in an OTG application.

When the USB module is not configured as a host (HOSTEN = 0), USBEN controls the connection to USB. If the USB module is configured as a host (HOSTEN = 1), SOFEN controls whether the host is active on the USB link and sends SOF tokens every 1 ms.

Note that the other USB module control registers should be properly initialized before enabling USB via these bits.

27.5.4 Module Operation

27.5.4.1 FUNCTION OPERATION

The following steps are taken to respond to a USB transaction:

1. Software pre-initializes the appropriate buffer descriptor (Endpoint n, DIR, PPBI) and sets the OWN bit to '1'.
2. Hardware receives a TOKEN PID (IN, OUT, SETUP) from the USB host and checks the appropriate buffer descriptor.
3. If the transaction is TX (IN), the module reads the packet data from data memory.
4. Hardware receives a data PID (DATA0/1/2, MDATA) and sends or receives the packet data.
5. If the transaction is RX (SETUP, OUT), the module writes the packet data to data memory.
6. The module will issue or wait for a HANDSHAKE PID (ACK, NAK, STALL) unless the endpoint is set up as an isochronous endpoint (EPHSHK bit, UEPMx<0>, is cleared).
7. The module will update the buffer descriptor and write the OWN bit to '0' (SW owned).
8. The module will update the U1STAT register and set the TRNIF interrupt.
9. Software reads the U1STAT register and determines the endpoint and direction for the transaction.
10. Software reads the appropriate buffer descriptor, completes all necessary processing and then clears the TRNIF interrupt.

Note that for IN transactions (host reading data from the device), the read data must be ready when the host begins USB signaling. Otherwise, the USB module will send a NAK handshake if UOWN is '0'.

27.5.4.2 USB LINK STATES

27.5.4.2.1 Reset

As a host, software is required to drive Reset signaling. Software may do this by setting USBRST (U1CON<4>). As per the USB Specification, the host must drive the Reset for at least 50 ms. (This does not have to be continuous Reset signaling. Refer to the USB 2.0 Specification for more information.) Following Reset, the host must not initiate any downstream traffic for another 10 ms.

As a device, the USB module will assert the URSTIF (U1IR<0>) interrupt when it has detected Reset signaling for 2.5 μs. Software must perform any Reset initialization processing at this time. This includes setting the Address register to 0x00 and enabling Endpoint 0. The URSTIF interrupt will not be set again until the Reset signaling has gone away and then has been detected again for 2.5 μs.

27.5.4.2.2 Idle and Suspend

The Idle state of the USB bus is in a constant J state. When the USB bus has been Idle for 3 ms, a function should go into suspend state. During active operation, the USB host will send a SOF token every 1 ms, preventing a device from going into suspend state.

Once the USB link is in the suspend state, a USB host or device must drive resume signaling prior to initiating any bus activity. (The USB link may also be disconnected.)

As a USB host, software should consider the link in suspend state as soon as software clears the SOFEN (U1CON<0>).

As a USB function, hardware will set the IDLEIF (U1IR<4>) interrupt when it detects a constant Idle on the bus for 3 ms. Software should consider the link in suspend state when the IDLEIF interrupt is set.

Once a suspend condition has been detected, the firmware may wish to place the USB hardware in a Suspend mode by setting SUSPEND (U1PWRC<0>). The hardware Suspend mode gates the 48 MHz USB peripheral clock and places the USB transceiver in a Low-Power mode.

Additionally, the user may put the PIC24F into Sleep mode while the link is suspended.

27.5.4.2.3 Driving Resume Signaling

If software wants to wake the USB bus from suspend state, it may do so by setting RESUME (U1CON<5>). This will cause the hardware to generate the proper resume signaling (including finishing with a low-speed End-Of-Packet (EOP) if a host).

A USB function should not drive resume signaling unless the Idle state has persisted for at least 5 ms. The USB host also must have enabled the function for remote wake-up.

Software must set RESUME for 1-15 ms if a USB function, or for more than 20 ms if a USB host, then clear it to enable remote wake-up. For more information on RESUME signaling, see Section 7.1.7.7, 11.9 and 11.4.4 in the USB 2.0 Specification.

Writing RESUME will automatically clear the special hardware suspend (low-power) state.

If the part is a USB host, software should, at minimum, set the SOFEN (U1CON<0>) after driving its resume signaling. Otherwise, the USB link would return right back to the suspend state. Also, software must not initiate any downstream traffic for 10 ms following the end of resume signaling.

27.5.4.2.4 Receiving Resume Signaling

When the USB logic detects resume signaling on the USB bus for 2.5 μs, hardware will set the RESUMEIF (U1IR<5>) interrupt.

A function receiving resume signaling must prepare itself to receive normal USB activity. A host receiving resume signaling must immediately start driving resume signaling of its own. The special hardware suspend (low-power) state is automatically cleared upon receiving any activity on the USB link.

Reception of any activity on the USB link (this may be due to resume signaling or a link disconnect) while the PIC24F is in Sleep mode will cause the ACTVIF (U1OTGIR<4>) interrupt to be set; this will cause wake-up from Sleep.

27.5.4.2.5 Session Request Protocol (SRP)

SRP support is not required by non OTG applications. SRP may only be initiated at full speed. Refer to the On-The-Go Supplement Specification for more information regarding SRP.

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

Note:	When the A-device powers down the VBUS supply, the B-device must disconnect its pull-up resistor. Software must do this by clearing DPPULUP (U1OTGCON<7>) and DMPULUP (U1OTGCON<6>).

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 that:

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.

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 VBUS supply. Software should do this by setting VBUSCHG (UTOGCTRL<1>).

The B-device then proceeds by pulsing the D+ data line. Software should do this by setting DPPULUP (U1OTGCON<7>). The data line should be held high for 5-10 ms.

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 setting VBUSON (U1OTGCON<3>).

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

27.5.4.2.6 Host Negotiation Protocol (HNP)

An OTG application with a micro-AB receptacle must support HNP. HNP allows an OTG B-device to temporarily become the USB host. The A-device must first enable HNP in the B-device. Refer to the On-The-Go supplement for more information regarding HNP. HNP may only be initiated at full-speed.

After being enabled for HNP by the A-device, the B-device can request to become the host any time that the USB link is in suspend state by simply indicating a disconnect. Software may accomplish this by clearing DPPULUP.

When the A-device detects the disconnect condition (via the URSTIF (U1IR<0>) interrupt), the A-device may allow the B-device to take over as host. The A-device does this by signaling connect as a full-speed function. Software may accomplish this by setting DPPULUP. If the A-device responds instead with resume signaling, the A-device will remain as host.

When the B-device detects the connect condition (via ATTACHIF (U1IR<6>), the B-device becomes host. The B-device drives Reset signaling prior to using the bus.

When the B-device has finished in its role as host, it stops all bus activity and turns on its D+ pull-up resistor, DPPULUP.

When the A-device detects a suspend condition (Idle for 3 ms), the A-device turns off its D+ pull-up. The A-device may also power-down the VBUS supply to end the session.

When the A-device detects the connect condition (via ATTACHIF), the A-device resumes host operation and drives Reset signaling.

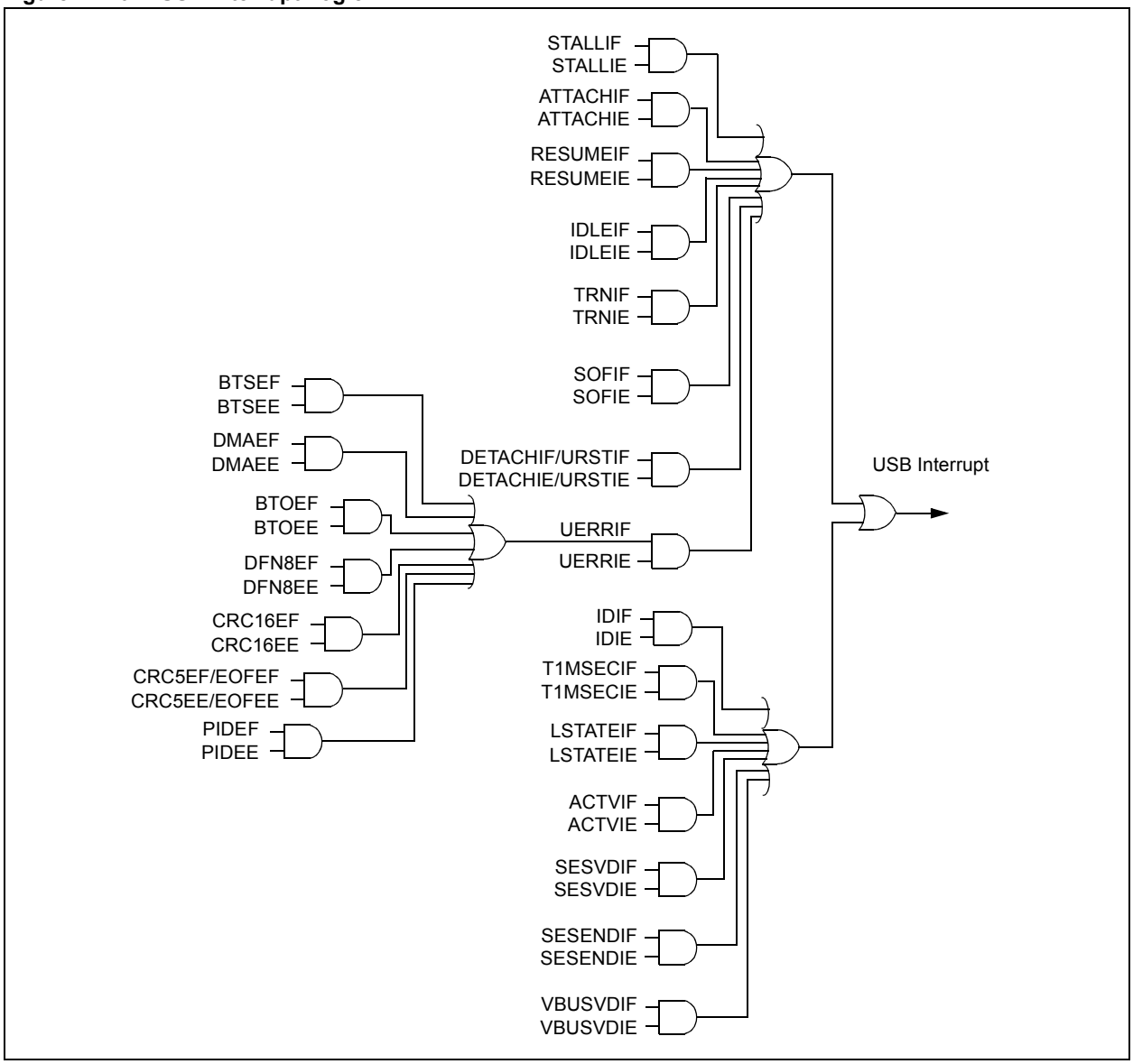
27.6 INTERRUPTS

The USB module uses interrupts to signal USB events, such as a change in status, data received and buffer empty events, to the CPU. Firmware must be able to respond to these interrupts in a timely manner.

The USB module can generate interrupt requests from a variety of events. Each interrupt source in the USB module has an interrupt bit and a corresponding enable bit. To interface these interrupts to the CPU, the USB interrupts are combined such that any enabled USB interrupt will cause a generic USB interrupt (if the USB interrupt is enabled) to the interrupt controller (Figure 27-10). The UERRIF bit (U1IR<1>) is a reflection of all enabled error flags and is read-only. This bit can be used to poll the USB module for events while in an ISR. The USB Interrupt Service Routine (ISR) must then determine which USB event(s) caused the CPU interrupt and service them appropriately.

There are two layers of Interrupt registers in the USB module. The top level of bits consists of overall USB status interrupts in the U1OTGIR and U1IR registers. The U1OTGIR and U1IR bits are individually enabled through the corresponding bits in the U1OTGIE and U1IE registers. In addition, the USB Error Condition bit (UERRIF) passes through any interrupt conditions in the U1EIR register, which is enabled via the U1EIE register bits.

Figure 27-10: USB Interrupt Logic



27.6.1 Interrupt Timing

Interrupts for transactions are generated at the end a successfully completed transaction. Figure 27-11 shows some typical event sequences that can generate a USB interrupt and when that interrupt is generated. There is no mechanism by which software can manually set an interrupt bit in the USB module.

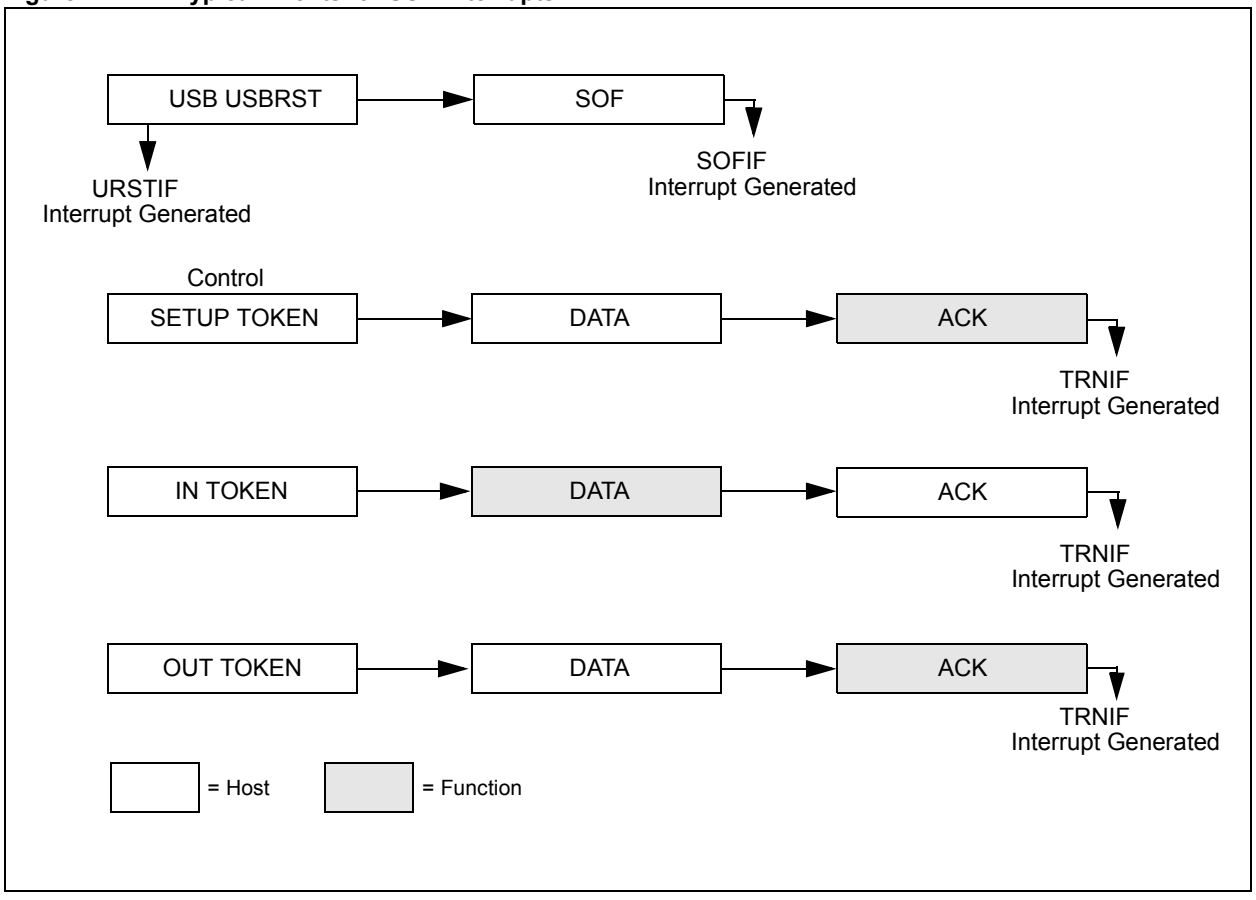
The values in the Interrupt Enable registers (U1IE, U1EIE, U1OTGIE) only affect the propagation of an interrupt condition to the CPU's interrupt controller. Even though an interrupt is not enabled, interrupt conditions can still be polled and serviced.

The top level USB Interrupt flag, USB1IF, cannot be cleared unless all of the enabled lower level USB interrupts (i.e., those with their corresponding IE bit set) are also clear. To clear USB1IF, the application should first clear (or disable) the lower level interrupt flag that triggered USB1IF to become set.

27.6.2 Interrupt Servicing

Once an interrupt bit has been set by the USB module (in U1IR, U1EIR or U1OTGIR), it must be cleared by software by writing a '1'. The USB Interrupt Flag, USBIF (IFS5<6>), must be cleared at the end of the ISR.

Figure 27-11: Typical Events for USB Interrupts



27.7 I/O PINS

Table 27-6 summarizes the use of pins relating to the USB module.

Table 27-6: Pins Associated with the USB Module

Mode	Pin Name	Module Control	Controlling Bit Field(1)	Required TRIS Bit Setting	Pin Type	Description
Host						
	D+	USBEN	—	—	U, I/O	Data line +
	D-	USBEN	—	—	U, I/O	Data line -
	VBUS(2)	USBEN	—	—	P	Input for USB power, connects to OTG comparators
	VBUSON	USBEN	VBUSON	—	D, O	Output to control supply for VBUS(2)
	VUSB(2)	—	—	—	P	Power in for USB transceiver; output of internal USB voltage regulator (if enabled)
Device						
	D+	USBEN	—	—	U, I/O	Data line +
	D-	USBEN	—	—	U, I/O	Data line -
	ID	USBEN	—	—	D, I	OTG mode host/device select
	VBUS(2)	USBEN	VBUSCHG, VBUSDIS	—	P	Input for USB power; connects to OTG comparators
	VUSB(2)	—	—	—	P	Power in for USB transceiver
On-The-Go (OTG)						
	D+	USBEN	—	—	U, I/O	Data line +
	D-	USBEN	—	—	U, I/O	Data line -
	ID	USBEN	—	—	D, I	OTG mode host/device select
	VBUS(2)	USBEN	—	—	A, I/O, P	Analog input for USB power; connects to OTG comparators
	VBUSON	USBEN	VBUSCHG, VBUSDIS, VBUSON	—	D, O	Output to control supply for VBUS(2)
	VUSB(2)	—	—	—	P	Power in for USB transceiver

Legend: I = Input, O = Output, A = Analog, D = Digital, U = USB, P = Power, I/O = Input/Output

Note 1: All pins are subject to the device pin priority control. See the specific device data sheet for further information.

2: JTAG boundary scan is not available on all USB pins.

27.8 OPERATION IN DEBUG AND POWER-SAVING MODES

27.8.1 Operation in Sleep

Use of Sleep mode is only recommended in two cases:

- The USB module is disabled.
- The USB module is in a suspend state.

Placing the USB module in Sleep mode while the bus is active can result in improper USB device behavior.

When the device enters Sleep mode, the clock to the USB module is maintained. The effect on the CPU clock source is dependent on the USB and CPU clock configuration.

- If the CPU and USB were using the Primary Oscillator (POSC) source, the CPU is disconnected from the clock source when entering Sleep and the oscillator is left in an enabled state for the USB module.
- If the CPU was using a different clock source, that clock source is disabled upon entering Sleep. The USB clock source is left enabled.

To further reduce power consumption, the USB module can be placed in Suspend mode prior to placing the CPU in Sleep. The effect on the CPU clock source is dependent on the USB and CPU clock configuration.

- If the CPU and USB were using the Primary Oscillator (POSC) source, the oscillator is disabled when the CPU enters Sleep.
- If the CPU was not sharing POSC with the USB module, POSC will be disabled when the USB module enters suspend. The CPU clock source will be disabled when the CPU enters Sleep.

The USB activity interrupt is enabled when the device enters Sleep mode. The activity interrupt should be used to awaken the device from Sleep when the device wake clock sources are disabled by Sleep and suspend will be re-enabled. Oscillator start-up times and PLL lock times must be taken into account when the activity interrupt is to be used to wake the device.

27.8.2 Operation in Idle Mode

When the device enters Idle mode, the clock sources for the USB module and CPU are maintained.

The USB activity interrupt is enabled when the device enters Idle mode. The activity interrupt should be used to awaken the device from Idle when the device wake clock sources, disabled by Idle and Suspend, will be re-enabled. Oscillator start-up times and PLL lock times should be taken into account when the activity interrupt is to be used to wake the device. Refer to **Section 10. “Power-Saving Features”** for more information.

27.8.3 Operation in Debug Modes

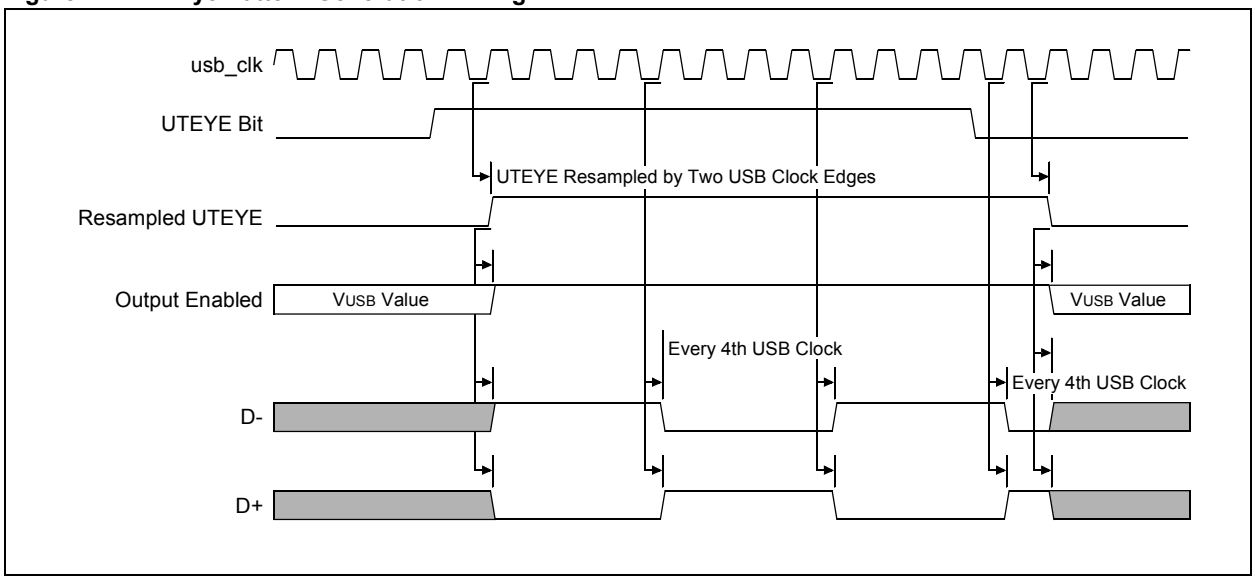
27.8.3.1 EYE PATTERN

To assist in USB hardware debugging and testing, an eye pattern test generator is incorporated in the module. This pattern is generated by the module when the UTEYE bit (U1CNFG1<7>) is set. The USB module must be enabled, USBPWR (PWRC<0> = 1), the USB 48 MHz clock must be enabled, SUSPEND (U1PWRC<1>) = 0, and the module is not in Freeze mode.

Once the UTEYE bit is set, the module will simply emulate a switch from a receive to transmit state and will start transmitting a **J-K-J-K** bit sequence. The bit sequence will be repeated indefinitely while the Eye Pattern Test mode is enabled (see Figure 27-12).

Note:	The UTEYE bit should never be set while the module is connected to an actual USB system. The mode is intended for board verification to aid with USB certification tests. This test does not properly test the transition from a receive to a transmit state. The test is intended to show a system developer the noise integrity of the USB signals, which can be affected by board traces, impedance mismatches and proximity to other system components.

Figure 27-12: Eye Pattern Generation Timing



27.8.3.2 USB $\overline{OE}$ MONITOR

The USB $\overline{OE}$ monitor indicates whether the USB is listening to the bus or actively driving the bus. This debug feature is enabled when U1CNFG1<UOEMON> = 1.

The $\overline{OE}$ monitoring is useful for initial system debugging, as well as scope triggering, during eye pattern generation tests.

27.9 EFFECTS OF A RESET

All forms of Reset force the USB module registers to the default state.

Note:	The USB module cannot ensure the state of the BDT, nor that of the packet data buffers contained in RAM following a Reset.

27.9.1 Device Reset ($\overline{\text{MCLR}}$)

A device Reset forces all USB module registers to their Reset state. This turns the USB module off.

27.9.2 Power-on Reset (POR)

A POR Reset forces all USB module registers to their Reset state. This turns the USB module off.

27.9.3 Watchdog Timer Reset (WDT)

A WDT Reset forces all USB module registers to their Reset state. This turns the USB module off.

27.10 ELECTRICAL SPECIFICATIONS

Table 27-7: OTG Timing Requirements

Param No.	Symbol	Characteristics	Min	Typ	Max	Units	Conditions
USB313	VUSB	USB Voltage	3.0	—	3.6	V	Voltage on VUSB must be in this range for proper USB operation
USB315	VILUSB	Input Low Voltage for USB Buffer	—	—	0.8	V	
USB316	VIHUSB	Input High Voltage for USB Buffer	2.0	—	—	V	
USB318	VDIFS	Differential Input Sensitivity	—	—	0.2	V	The difference between D+ and D- must exceed this value while VCM is met
USB319	VCM	Differential Common Mode Range	0.8	—	2.5	V	
USB320	ZOUT	Driver Output Impedance	28.0	—	44.0	Ω	
USB321	VOL	Voltage Output Low	0.0	—	0.3	V	1.5 kΩ load connected to 3.6V
USB322	VOH	Voltage Output High	2.8	—	3.6	V	1.5 kΩ load connected to ground

27.11 REGISTER MAP

A summary of the Special Function Registers associated with the PIC24F USB On-The-Go (OTG) is provided in Table 27-8.

Table 27-8: USB OTG Register Map

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1OTGIR	0480	—	—	—	—	—	—	—	—	IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF	0000
U1OTGIE	0482	—	—	—	—	—	—	—	—	IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE	0000
U1OTGSTAT	0484	—	—	—	—	—	—	—	—	ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD	0000
U1OTGCON	0486	—	—	—	—	—	—	—	—	DPPULUP	DMPULUP	DPPULDWN(2)	DMPULDWN(2)	VBUSON(2)	OTGEN(2)	VBUSCHG(2)	VBUSDIS(2)	0000
U1PWRC	0488	—	—	—	—	—	—	—	—	UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR	0000
U1IR	048A (1)	—	—	—	—	—	—	—	—	STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF	0000
		—	—	—	—	—	—	—	—	STALLIF	ATTACHIF(1)	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF(1)	0000
U1IE	048C (1)	—	—	—	—	—	—	—	—	STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE	0000
		—	—	—	—	—	—	—	—	STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	DETACHIE(1)	0000
U1EIR	048E (1)	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0000
		—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	EOFEF(1)	PIDEF	0000
U1EIE	0490 (1)	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0000
		—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	EOFEE(1)	PIDEE	0000
U1STAT	0492	—	—	—	—	—	—	—	—	ENDPT3(3)	ENDPT2(3)	ENDPT1(3)	ENDPT0(3)	DIR	PPBI	—	—	0000
U1CON	0494 (1)	—	—	—	—	—	—	—	—	—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN	0000
		—	—	—	—	—	—	—	—	JSTATE	SE0	TOKBUSY	USBRST(1)	HOSTEN	RESUME	PPBRST	SOFEN(1)	0000
U1ADDR	0496	—	—	—	—	—	—	—	—	LSPDEN(1)	USB Device Address (DEVADDR)							0000
U1BDTP1	0498	—	—	—	—	—	—	—	—	Buffer Descriptor Table Base Address							—	0000
U1FRML	049A	—	—	—	—	—	—	—	—	Frame Count Low Byte								0000
U1FRMH	049C	—	—	—	—	—	—	—	—	—	—	—	—	—	Frame Count High Byte			0000
U1TOK(2)	049E	—	—	—	—	—	—	—	—	PID3	PID2	PID1	PID0	EP3	EP2	EP1	EP0	0000
U1SOF(2)	04A0	—	—	—	—	—	—	—	—	Start-of-Frame Threshold								0000
U1CNFG1	04A6	—	—	—	—	—	—	—	—	UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0	0000
U1CNFG2	04A8	—	—	—	—	—	—	—	—	—	—	UVCMPSEL(4)	PUVBUS	EXTI2CEN	UVBUSDIS	UVCMPDIS	UTRDIS	0000

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

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

2: This register is meaningful in Host or OTG modes only.

3: Device mode only. These bits always read as '0' in Host mode.

4: Unimplemented in some devices; read as '0'.

Table 27-8: USB OTG 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
U1EP0	04AA	—	—	—	—	—	—	—	—	LSPD(1)	RETRYDIS(1)	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP1	04AC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP2	04AE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP3	04B0	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP4	04B2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP5	04B4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP6	04B6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP7	04B8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP8	04BA	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP9	04BC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP10	04BE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP11	04C0	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP12	04C2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP13	04C4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP14	04C6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP15	04C8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1PWMRRS	04CC	USB Power Supply PWM Duty Cycle								USB Power Supply PWM Period								0000
U1PWMCON	04CE	PWMEN	—	—	—	—	—	PWMPOL	CNTEN	—	—	—	—	—	—	—	—	0000

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

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

2: This register is meaningful in Host or OTG modes only.

3: Device mode only. These bits always read as '0' in Host mode.

4: Unimplemented in some devices; read as '0'.

27.12 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC24F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the USB On-The-Go (OTG) module are:

Title	Application Note #

No related application notes at this time.

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC24F family of devices.

27.13 REVISION HISTORY

Revision A (December 2007)

This is the initial released revision of this document.

Revision B (January 2010)

Changed the name of bit U1CON<4> from “RESET” to “USBRST” in all occurrences.

Revised **Section 27.3.3 “Hardware Interface”** with entirely new material on pull-up and pull-down resistors, power requirements by USB mode, external USB transceiver interfaces and transceiver power requirements. In the process, removed the existing Figures 27-4, 27-5 and 27-6 entirely. Also updated **Section 27.2 “Control Registers”** and **Section 27.11 “Register Map”** to reflect bit level changes from these revisions.

Revised the main block diagram (Figure 27-1) to reflect changes in the external comparator interface from previous versions.

Removed existing Figures 27-7 and 27-8 from **Section 27.4 “Device Mode Operation”** entirely, as redundant or obsoleted material.

Updated text throughout **Section 27.4 “Device Mode Operation”** and **Section 27.5 “Host Mode Operation”** to reflect standard register and bit name nomenclature.

Other minor typographic fixes throughout.

NOTES: