Section 25. USB On-The-Go (OTG)

HIGHLIGHTS

This section of the manual contains the following major topics:

Note: This family reference manual section is meant to serve as a complement to device data sheets. Depending on the device variant, this manual section may not apply to all dsPIC33E/PIC24E devices.

Please consult the note at the beginning of the **"USB On-The-Go (OTG)"** chapter in the current device data sheet to check whether this document supports the device you are using.

Device data sheets and family reference manual sections are available for download from the Microchip Worldwide Web site at: http://www.microchip.com

25.1 INTRODUCTION

The dsPIC33E/PIC24E Universal Serial Bus (USB) On-The-Go (OTG) module includes the following features:

- USB Full-Speed Support for Host and Device
- Low-Speed Host Support
- USB On-The-Go (OTG) Support
- Integrated Signaling Resistors
- Integrated Analog Comparators for VBUS Monitoring
- Integrated USB Transceiver
- Transaction Handshaking Performed by Hardware
- Endpoint Buffering Anywhere in System RAM
- Integrated Bus Master to Access System RAM
- Does not require the dsPIC33E/PIC24E DMA module for its operation
- Supports external USB transceiver and external VBUS comparator

The USB module contains analog and digital components to provide a USB 2.0 full-speed and low-speed embedded host, full-speed device, or OTG implementation with a minimum of external components. This module in Host mode is intended for use as an embedded host and therefore does not implement a UHCI or OHCI controller.

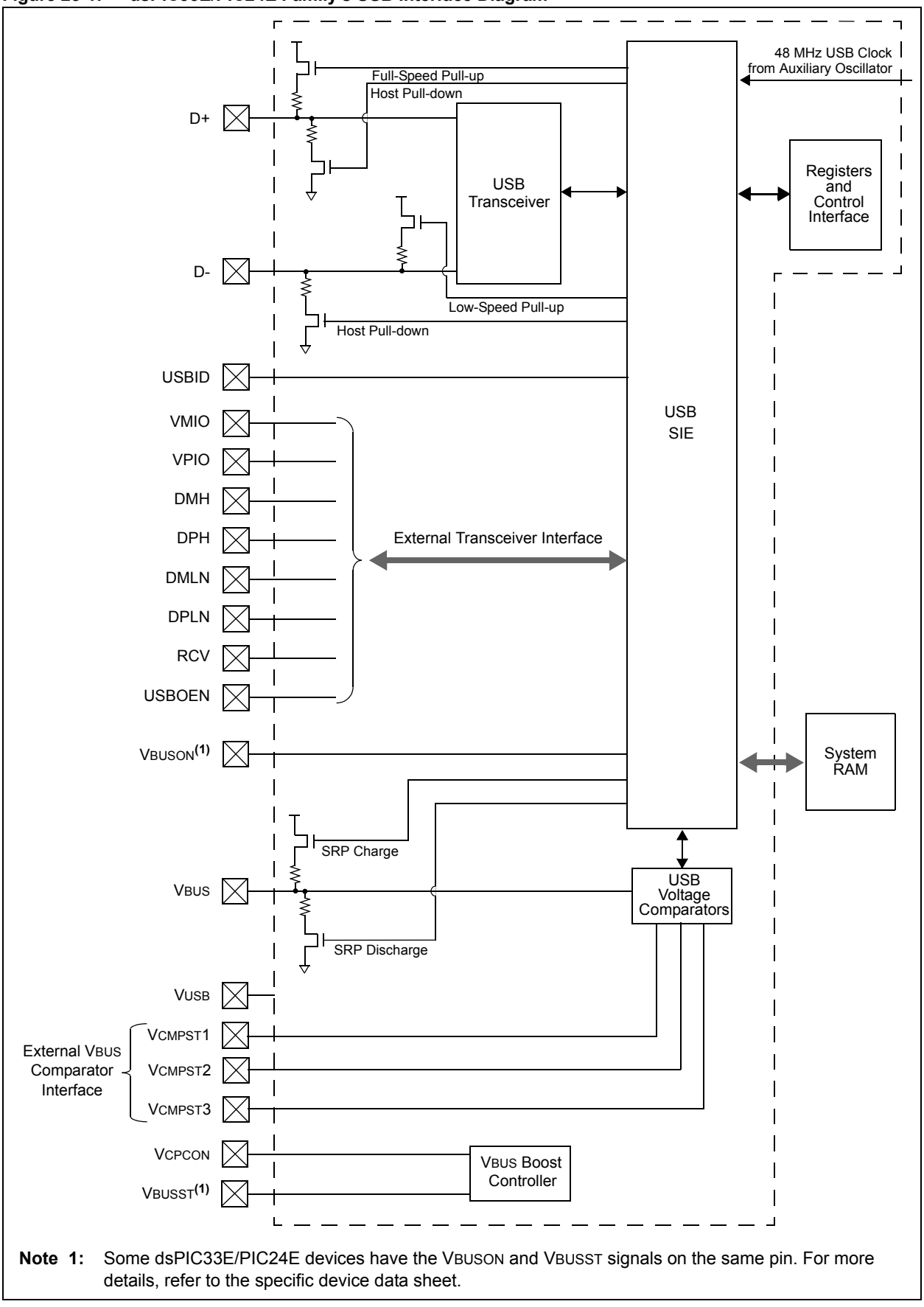
The USB module consists of the clock generator, the USB voltage comparators, the transceiver, the Serial Interface Engine (SIE), a dedicated USB Bus Master, pull-up and pull-down resistors and the register interface. A block diagram of the dsPIC33E/PIC24E USB OTG module is illustrated in Figure 25-1.

The USB module requires a 48 MHz clock for USB full-speed and low-speed communication, which is supplied via a device auxiliary oscillator. The module requires an external supply of +3.3V to be connected to the VUSB pin. The voltage comparators monitor the voltage on the VBUS pin to determine the state of the bus. The application can disable the internal VBUS voltage comparators and use an external VBUS voltage comparator. A flexible, external digital VBUS comparator interface is available for this purpose. The transceiver provides the analog translation between the USB bus and the digital logic. An external USB transceiver interface is available. In applications, where an external transceiver is preferred, the internal transceiver of the USB module can be disabled.

The SIE is a state machine that transfers data to and from the endpoint buffers, and generates the hardware protocol for data transfers. The USB Bus Master transfers data between the data buffers in RAM and the SIE. 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.

Note: The implementation and use of the USB specifications, as well as other third-party specifications or technologies, may require licensing; including, but not limited to, USB Implementers Forum, Inc. (also referred to as USB-IF). The user is fully responsible for investigating and satisfying any applicable licensing obligations.

Figure 25-1: dsPIC33E/PIC24E Family's USB Interface Diagram



Note 1: Some dsPIC33E/PIC24E devices have the VBUSON and VBUSST signals on the same pin. For more details, refer to the specific device data sheet.

25.2 CONTROL REGISTERS

The USB module includes the following control and status registers:

- **UxOTGSTAT: USB OTG Status Register**

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

- **UxOTGCON: USB On-The-Go Control Register**

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

- **UxPWRC: USB Power Control Register**

  This register controls the power-saving modes.

- **UxSTAT: USB Status Register**

  This register is a 16-deep FIFO. It is read-only by the CPU and read/write by the USB module. It is only valid when the TRNIF bit (UxIR<3>) is set.

- **UxCON: USB Control Register (Device mode)** and **UxCON: USB Control Register (Host mode)**

  This register provides various control information for the module.

- **UxADDR: USB Address Register**

  This 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.

- **UxTOK: USB Token Register (Host mode only)**

  This read/write register is 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.

- **UxSOF: USB OTG Start-of-Token Threshold Register (Host mode only)**

  This read/write register contains the count bits of the Start of Frame (SOF) Threshold Value used in Host mode only.

  To prevent collision between a packet data with the Start of Frame token, which is sent every 1 ms, the USB module will not send any new transactions within the last UxSOF byte times. The USB module completes the 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 the SOF token is sent.

- **UxCNFG1: USB Configuration Register 1**

  This read/write register controls the debug and idle behavior of the module. The register must be preprogrammed prior to enabling the module.

- **UxCNFG2: USB Configuration Register 2**

  This read/write register configures the interface signals.

- **UxOTGIR: USB OTG Interrupt Status Register (Host mode only)**

  This register records the 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.

- **UxOTGIE: USB OTG Interrupt Enable Register (Host mode only)**

  This register enables the corresponding interrupt status bits defined in the UxOTGIR register.

- **UxIR: USB Interrupt Status Register (Device mode only)** and **UxIR: USB Interrupt Status Register (Host mode only)**

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

- **UxIE: USB Interrupt Enable Register (Device mode)** and **UxIE: USB Interrupt Enable Register (Host mode)**

  This register's 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 UxIR register.

- **UxEIR: USB Error Interrupt Status Register (Device mode)** and **UxEIR: USB Error Interrupt Status Register (Host mode)**

  This 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.

- **UxEIE: USB Error Interrupt Enable Register (Device mode)** and **UxEIE: USB Error Interrupt Enable Register (Host mode)**

  This register's 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 UxEIR register.

- **UxEPn: USB Endpoint n Control Registers (n = 0 to 15)**

  These registers control the behavior of the corresponding endpoint.

- **UxBDTPx: USB Buffer Description Table Registers** (Register 25-23, Register 25-24 and Register 25-25)

  These read/write registers define the bits (31-9) of the 32-bit base address of the endpoint Buffer Descriptor Table (BDT) in the system memory.

- **UxPWMCON: USB VBUS PWM Generator Control Register**

  This register provides various bits for controlling the VBUS Boost Regulator PWM module.

- **UxPWMRRS: Duty Cycle and PWM Period Register**

  This register specifies the duty cycle and PWM period for the VBUS Boost Regulator PWM module.

- **UxFRMH: USB Frame Number High Register** and **UxFRML: USB Frame Number Low Register**

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

25.2.1 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 has no effect.

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".

25.2.2 Registers Description

Register 25-1: UxOTGSTAT: 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 = Type A plug is 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 = VBUS voltage is above session valid on the A or B device
- 0 = VBUS voltage is below session valid on the A or B device

bit 2 **SESEND:** B-Session End Indicator bit
- 1 = VBUS voltage is below session valid on the B device
- 0 = VBUS voltage is above session valid on the B device

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

bit 0 **VBUSVD:** A-VBUS Valid Indicator bit
- 1 = VBUS voltage is above session valid on the A device
- 0 = VBUS voltage is below session valid on the A device

Register 25-2: UxOTGCON: 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 is enabled; all D+/D- pull-ups and pull-downs bits are enabled
- 0 = USB OTG is disabled; D+/D- pull-ups and pull-downs are controlled in hardware by the settings of the HOSTEN and USBEN bits (UxCON<3,0>)

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

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

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

Register 25-3: UxPWRC: USB Power Control Register

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

HS, HC-x	U-0	U-0	R/W	U-0	U-0	R/W-0, HC	R/W-0
UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR(1)
bit 7							bit 0

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

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

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

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

bit 4 **USLPGRD:** Sleep Guard bit
- 1 = Indicates 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
- 0 = Normal USB OTG operation

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

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

Register 25-4: UxSTAT: 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
ENDPT<3:0>(1)				DIR	PPBI	—	—
bit 7							bit 0

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

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

bit 7-4 **ENDPT<3:0>:** Number of the Last Endpoint Activity bits (represents the number of the endpoint BDT updated by the last USB transfer)(1)

1111 = Endpoint 15
1110 = Endpoint 14
•
•
•
0001 = Endpoint 1
0000 = Endpoint 0

bit 3 **DIR:** Last Buffer Descriptor 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 Buffer Descriptor Pointer Indicator bit

1 = The last transaction was to the ODD buffer descriptor bank
0 = The last transaction was to the EVEN buffer descriptor bank

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

Note 1: 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 25-5: UxCON: USB Control Register (Device mode)

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(1)	RESUME	PPBRST(2)	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(3)
- 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 are disabled; automatically set when a SETUP token is received
- 0 = SIE token and packet processing are enabled

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

bit 3 **HOSTEN:** Host Mode Enable bit(1)
- 1 = USB host capability is enabled; pull-downs on D+ and D- are activated in hardware
- 0 = USB host capability is disabled

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

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit(2)
- 1 = Reset all Ping-Pong Buffer Pointers to the EVEN buffer descriptor banks
- 0 = Ping-Pong Buffer Pointers are not reset

bit 0 **USBEN:** USB Module Enable bit
- 1 = USB module and supporting circuitry are enabled (device attached); D+ pull-up is activated in hardware
- 0 = USB module and supporting circuitry are disabled (device detached)

Note 1: This bit should be '0' in Device mode.

2: Keeping this bit set will force the USB module to always use the EVEN buffer descriptors. The ping-pong feature gets disabled.

3: While this bit indicates the single-ended zero condition on the bus, it is recommended that the application uses the URSTIF bit (UxIR<0>) to detect a RESET condition.

Register 25-6: UxCON: USB Control Register (Host mode)

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(1)	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) is detected on the USB
- 0 = No J state is detected

bit 6 **SE0:** Live Single-Ended Zero Flag bit
- 1 = Single-ended zero is active on the USB bus
- 0 = No single-ended zero is 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 is generated. For Software Reset, the user application must set this bit for 50 ms, and then clear it
- 0 = USB Reset is terminated

bit 3 **HOSTEN:** Host Mode Enable bit
- 1 = USB host capability is enabled; pull-downs on D+ and D- are activated in hardware
- 0 = USB host capability is disabled

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

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit(1)
- 1 = Reset all Ping-Pong Buffer Pointers to the EVEN buffer descriptor banks
- 0 = Ping-Pong Buffer Pointers are not reset

bit 0 **SOFEN:** Start of Frame Enable bit
- 1 = Start of Frame token sent every 1 ms
- 0 = Start of Frame token is disabled

Note 1: Keeping this bit set will force the USB module to always use the EVEN buffer descriptors. The ping-pong feature gets disabled.

Register 25-7: UxADDR: 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)	DEVADDR<6:0>						
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 15-8 **Unimplemented:** Read as '0'

bit 7 **LSPDEN:** Low-Speed Enable Indicator 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.

Register 25-8: UxTOK: 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
PID<3:0>(1)				EP<3:0>			
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 15-8 **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 should not to be used.

Register 25-9: UxSOF: 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
CNT<7:0>							
bit 7							bit 0

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

bit 15-8 **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 25-10: UxCNFG1: 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	U-0	U-0
UTEYE	UOEMON	—	USBSIDL	—	—	—	—
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 is activated
0 = Eye pattern test is deactivated

bit 6 **UOEMON:** USB $\overline{OE}$ Monitor Enable bit
1 = $\overline{OE}$ signal is active; it indicates intervals during which the D+/D- lines are driving
0 = $\overline{OE}$ signal is 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 3-0 **Unimplemented:** Read as '0'

Note 1: When the UTRIS bit (UxCNFG2<0>) is set, the $\overline{OE}$ signal is active regardless of the setting of UOEMON.

Register 25-11: UxCNFG2: 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	PUVBUS	EXTI2CEN	UVBUSDIS(1)	UVCMPDIS(1)	UTRDIS(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 15-6 **Unimplemented:** Read as '0'

bit 5 **UVCMPSEL:** External Comparator Input Mode Select bit
When UVCMPDIS is set:
1 = Use 3-pin input for external comparators
0 = Use 2-pin input for external comparators

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

bit 3 **EXTI2CEN:** I2C™ Interface for External Module Control Enable bit
1 = External transceiver pull-up and pull-down resistors are controlled via I2C. The DMH, DML, DPH and DPL pins operate as a general purpose I/O pins
0 = External transceiver pull-up and pull-down resistors are controlled by DMH, DML, DPH and DPL pins

bit 2 **UVBUSDIS:** On-Chip 5V Boost Regulator Disable bit(1,2)
1 = On-chip boost regulator is disabled
0 = On-chip boost regulator is active

bit 1 **UVCMPDIS:** On-Chip VBUS Comparator Disable bit(1)
1 = On-chip VBUS comparator is disabled
0 = On-chip VBUS comparator is active

bit 0 **UTRDIS:** On-Chip Transceiver Disable bit(1)
1 = On-chip transceiver is disabled
0 = On-chip transceiver is active

Note 1: Do not change this bit while the USBPWR bit is set (UxPWRC<0> = 1).
2: The VBUSON bit should be set (UxOTGCON<3> = 1) to enable the output of the Boost Regulator module.

Register 25-12: UxOTGIR: USB OTG 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	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 was 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) is stable for 1 ms, but different from last time
- 0 = USB line state is not stable for 1 ms

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

bit 3 **SESVDIF:** Session Valid Change Indicator bit

Write a '1' to this bit to clear the interrupt.
- 1 = VBUS voltage has dropped below the session end level
- 0 = VBUS voltage has not dropped below the session end level

bit 2 **SESENDIF:** B-Device VBUS Change Indicator bit

Write a '1' to this bit to clear the interrupt.
- 1 = A change on the session end input was detected
- 0 = No change on the session end input was detected

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

bit 0 **VBUSVDIF:** A-Device VBUS Change Indicator bit

Write a '1' to this bit to clear the interrupt.
- 1 = Change on the session valid input was detected
- 0 = No change on the session valid input was detected

Register 25-13: UxOTGIE: USB OTG Interrupt Enable 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	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 is enabled
0 = Interrupt is disabled

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

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

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

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

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

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

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

Register 25-14: UxIR: 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 the transaction in Device mode
- 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 is observed

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

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of current token is complete; read UxSTAT register for endpoint BDT information
- 0 = Processing of current token is in progress (Clearing this bit causes the UxSTAT FIFO to advance.)

bit 2 **SOFIF:** Start of Frame Token Interrupt bit
- 1 = A Start of Frame token was received by the peripheral
- 0 = A Start of Frame token was not received by the peripheral

bit 1 **UERRIF:** USB Error Condition Interrupt bit (read-only)
- 1 = An unmasked error condition has occurred; only error states enabled in the UxEIE 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 reasserted
- 0 = No USB Reset has occurred

Register 25-15: UxIR: 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(1)	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF(1)
bit 7							bit 0

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

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

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

bit 6 **ATTACHIF:** Peripheral Attach Interrupt bit(1)
- 1 = A peripheral attachment is 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 is 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 is observed

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

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of current token is complete; read USTAT register for endpoint BDT information
- 0 = Processing of current token is not complete

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

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

bit 0 **DETACHIF:** Detach Interrupt bit(1)
- 1 = A peripheral detachment was detected by the module
- 0 = No peripheral detachment was detected

Note 1: These bits are persistent. An attempt to clear these bits while the cause condition is active does not have any effect.

Register 25-16: UxIE: USB Interrupt Enable Register (Device mode)

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
STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE
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 is enabled
0 = Interrupt is disabled

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

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

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

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

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

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

bit 0 **URSTIE:** USB Reset Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Register 25-17: UxIE: USB 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	R/W-0	R/W-0
STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	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 is enabled
0 = Interrupt is disabled

bit 6 **ATTACHIE:** Peripheral Attach Interrupt bit(1)
1 = Interrupt is enabled
0 = Interrupt is disabled

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

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

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

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

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

bit 0 **DETACHIE:** USB Detach Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Note 1: This bit is unimplemented in OTG mode, read as '0'.

Register 25-18: UxEIR: USB Error Interrupt Status Register (Device 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	R/K-0, HS	R/K-0, HS
BTSEF	BUSACCEF	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	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 was detected
- 0 = No bit stuff error was detected

bit 6 **BUSACCEF:** Bus Access Error Flag bit
- 1 = Peripheral tried to access an unimplemented RAM location
- 0 = RAM location access was successful

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

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

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 error checking has failed
- 0 = CRC16 error checking has passed

bit 1 **CRC5EF:** CRC5 Host Error Flag bit
- 1 = Token packet was rejected due to CRC5 error
- 0 = Token packet was accepted (no CRC5 error)

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

Register 25-19: UxEIR: USB Error 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	R/K-0, HS	R/K-0, HS
BTSEF	BUSACCEF	DMAEF	BTOEF	DFN8EF	CRC16EF	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 was detected
- 0 = No bit stuff error was detected

bit 6 **BUSACCEF:** Bus Access Error Flag bit
- 1 = Peripheral tried to access an unimplemented RAM location
- 0 = RAM location access was successful

bit 5 **DMAEF:** DMA Error Flag bit
- 1 = A USB DMA error condition was detected; the data size indicated by the buffer descriptor 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 has occurred

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 error checking has failed
- 0 = CRC16 error checking has passed

bit 1 **EOFEF:** End of Frame Error Flag bit
- 1 = End of Frame error has occurred
- 0 = End of Frame interrupt has not occurred

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

Register 25-20: UxEIE: USB Error Interrupt Enable Register (Device 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	R/W-0	R/W-0
BTSEE	BUSACCEE	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	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 is enabled
0 = Interrupt is disabled

bit 6 **BUSACCEE:** Bus Access Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

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

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

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

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

bit 1 **CRC5EE:** CRC5 Host Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

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

Register 25-21: UxEIE: USB Error 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	R/W-0	R/W-0
BTSEE	BUSACCEE	DMAEE	BTOEE	DFN8EE	CRC16EE	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 is enabled
0 = Interrupt is disabled

bit 6 **BUSACCEE:** Bus Access Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

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

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

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

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

bit 1 **EOFEE:** End of Frame Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

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

Register 25-22: UxEPn: 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)	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)
- 1 = Direct connection to a low-speed device is enabled
- 0 = Direct connection to a low-speed device is disabled

bit 6 **RETRYDIS:** Retry Disable bit (UEP0 only)(1)
- 1 = Retry NAK transactions are disabled
- 0 = Retry NAK transactions are 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 is enabled
- 0 = Endpoint n receive is disabled

bit 2 **EPTXEN:** Endpoint Transmit Enable bit
- 1 = Endpoint n transmit is enabled
- 0 = Endpoint n transmit is 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 is enabled
- 0 = Endpoint handshake is disabled (typically used for isochronous endpoints)

Note 1: These bits are available only for UxEP0, and only in Host mode. For all other UxEPn registers, these bits are always unimplemented and read as '0'.

Register 25-23: UxBDTP1: USB Buffer Description Table 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
BDTPTRL<15:9>							—
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 7-1 **BDTPTRL<15:9>:** Endpoint BDT Start Address bits
Defines bits 15-9 of the 32-bit endpoint buffer descriptor table start address.

bit 0 **Unimplemented:** Read as ‘0’

Register 25-24: UxBDTP2: USB Buffer Description Table Register 2

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
BDTPTRH<23:16>							
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 **BDTPTRH<23:16>:** Endpoint BDT Start Address bits
Defines bits 23-16 of the 32-bit endpoint buffer descriptor table start address.

Register 25-25: UxBDTP3: USB Buffer Description Table Register 3

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
BDTPTRU<31:24>							
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 **BDTPTRU<31:24>:** Endpoint BDT Start Address bits
Defines bits 31-24 of the 32-bit endpoint buffer descriptor table start address.

Register 25-26: UxPWMCON: 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'

Register 25-27: UxPWMRRS: Duty Cycle and PWM Period Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DC<7:0>							
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
PER<7:0>							
bit 7							bit 0

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

bit 15-8 **DC<7:0>:** Duty Cycle bits
These bits select the PWM duty cycle.

bit 7-0 **PER<7:0>:** PWM Period bits
These bits select the PWM period.

Register 25-28: UxFRMH: 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
—	—	—	—	—	FRMH<10:8>		
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 **FRMH<10:8>:** 11-Bit Frame Number Upper 3 bits
The register bits are updated with the current frame number whenever a SOF token is received.

Register 25-29: UxFRML: 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
FRML<7:0>							
bit 7							bit 0

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

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

bit 7-0 **FRML<7:0>:** 11-Bit Frame Number Lower 8 bits
The register bits are updated with the current frame number whenever a SOF token is received.

25.3 OPERATION

This section contains a brief overview of USB operation, followed by dsPIC33E/PIC24E USB module implementation specifics, and module initialization requirements.

Note: A good understanding of USB can be gained from documents that are available on the USB implementers web site. In particular, refer to the *"Universal Serial Bus Specification" (*Revision 2.0), which is available for download from: http://www.usb.org/developers/docs.

25.3.1 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).

25.3.2 Modes of Operation

The following USB implementation modes are described in this overview:

- Host mode
  - USB Standard Host mode – the USB implementation that is typically used for a personal computer
  - Embedded Host mode – the USB implementation that is typically used for a microcontroller
- Device mode – the USB implementation that is typically used for a peripheral such as a thumb drive, keyboard or mouse
- OTG Dual Role mode – the USB implementation in which an application may dynamically switch its role as either host or device

25.3.2.1 HOST MODE

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.

25.3.2.1.1 USB Standard Host Mode

In USB Standard Host mode, the following features and requirements are relevant:

- Large variety of devices are supported
- Supports all USB transfer types
- USB hubs are supported (allows connection of multiple devices simultaneously)
- Device drivers can be updated to support new devices
- 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 protocols must be supported (high-speed can be supported)

Note: The USB module of dsPIC33E/PIC24E device family does not support high-speed USB.

25.3.2.1.2 Embedded Host Mode

In Embedded Host mode, the following features and requirements are relevant:

- Only supports a specific list of devices, referred to as a Targeted Peripheral List (TPL)
- Only required to support those transfer types that are required by devices in the TPL
- USB hub support is optional
- Device drivers are not required to be updated
- Type 'A' receptacle is used for each port
- Only those speeds required by devices in the TPL 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

25.3.2.2 DEVICE MODE

USB devices accept commands and data from the host and respond to requests for data. USB devices perform peripheral functions, e.g., a mouse or other I/O, or data storage.

The following characteristics generally describe a USB device:

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

25.3.2.3 OTG DUAL ROLE MODE

An 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 which plug type was connected. The plug type connected to the receptacle determines the default role of the host or device. An OTG device performs 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 host and device between the two without disconnecting and reconnecting the cable. 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 device connected to the micro-B plug.

25.3.2.3.1 A-Device, the Default Host

In OTG dual role, operating as a host, the following features and requirements describe an A-device:

- Supports the devices on the TPL (class support is not allowed)
- Required to support those transaction types that are required by devices in the TPL
- USB hub support is optional
- Device drivers are not required to be updated
- A single micro-AB receptacle is used
- Full-speed protocol must be supported (high-speed and/or low-speed protocol can be supported)
- 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 HNP; the host can switch roles to become a device
- Supports at least one form of SRP
- A-device supplies VBUS power when the bus is powered, even if the roles are swapped using HNP

25.3.2.3.2 B-Device, the Default Device

In OTG dual role, operating as a USB device, the following features and requirements describe a B-Device:

- Class-specific 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
- A single micro-AB receptacle is used
- Must support full-speed protocol (support of low-speed and/or high-speed protocol is optional
- Supports control transfers, and supports data transfers as they are required for implementation
- Supports both forms of SRP – VBUS pulsing and data-line pulsing
- Supports HNP
- B-device does not supply VBUS power, even if the roles are swapped using HNP

Note: Dual-role devices that do not support full OTG functionality are possible using multiple USB receptacles. However, there may be special requirements if these devices are to be made USB-compliant. Refer to the USB IF (implementers forum) for details.

25.3.2.4 PROTOCOL

USB communication requires the use of specific protocols. The following subsections provide an overview of communication via USB.

25.3.2.4.1 Bus Transfers

Communication on the USB bus occurs through transfers between a host and a device. Each transfer type has unique features. An embedded or OTG host can implement only the control and the data transfer(s) it will use.

The following four transfer types are possible on the bus:

- Control

  Control transfer is used to identify a device during enumeration and to control it during operation. A percentage of the USB bandwidth is ensured to be available to control transfers. The data is verified by a Cyclic Redundancy Check (CRC) and reception by the target is verified.

- Interrupt

  Interrupt transfer is a scheduled transfer of data in which the host allocates time slots for the transfers as required by the device's configuration. 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 acknowledged.

- Isochronous

  Isochronous transfer is a scheduled transfer of data in which the host allocates time slots for the transactions as required by the device's configuration. Reception of the data is not acknowledged, but the data integrity is verified by the device using a CRC. This transfer type is typically used for audio and video.

- Bulk

  Bulk transfer 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 time that has not been allocated to the other three transfer types. The data is verified by a CRC and reception is acknowledged.

The following transfer speeds are defined in the *"Universal Serial Bus Specification"* (Revision 2. 0):

- 480 Mbps – high-speed
- 12 Mbps – full-speed
- 1.5 Mbps – low-speed

The dsPIC33E/PIC24E OTG devices support full-speed operation in Host and Device modes, and support low-speed operation in Host mode.

Information contrasting the timeliness, data integrity, data size and speed of each transfer or transaction type is shown in Table 25-1.

Table 25-1: Transaction Types (Full-Speed Operation)

Transaction Type	Timeliness Ensured	Data Integrity Ensured	Maximum Packet Size	Maximum Throughput(1)
Control	Yes	Yes	64	0.83 Mbps
Interrupt	Yes	Yes	64	1.22 Mbps
Isochronous	Yes	No	1023	1.28 Mbps
Bulk	No	Yes	64	1.22 Mbps

Note 1: These numbers reflect the theoretical maximum data throughput including the protocol overhead, on an otherwise empty bus. The bit stuffing overhead required by the Non-Return to Zero Inverted (NRZI) encoding is not included in the calculations.

25.3.2.4.2 Bandwidth Allocation

Control transfers, or transactions, are guaranteed at least 10% of the available bandwidth within a given frame. The remainder is available for allocation to Interrupt and Isochronous transfers. Bulk transfers are allocated from any bandwidth not allocated to control, interrupt or isochronous transfers. Bulk transfers are not assured bandwidth. However, in practice, they have the greatest bandwidth since frames are rarely fully allocated.

25.3.2.4.3 Endpoints and USB Descriptors

The 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) or receive (RX) functionality. Each endpoint uses one transaction type. Endpoint 0 is the default control transfer endpoint.

25.3.2.5 PHYSICAL BUS INTERFACE

25.3.2.5.1 Bus Speed Selection

The *"Universal Serial Bus Specification"* (Revision 2.0) defines full-speed operation as 12 Mbps and low-speed operation as 1.5 Mbps. 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.

25.3.2.5.2 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 list describes the VBUS operation:

- Standard host typically supplies continuous power to the bus
- Host may switch off VBUS to save power
- USB device never powers the bus – VBUS pulsing may be supported as part of the SRP
- OTG A-device supplies power to the bus, and typically turns off VBUS to conserve power
- OTG B-device can pulse VBUS for SRP

Note: The dsPIC33E/PIC24E device family does not supply the VBUS power. Refer to the specific device data sheet for VBUS electrical parameters.

25.3.2.5.3 Generating VBUS Externally

The USB module of dsPIC33E/PIC24E device family does not generate the 5V power required for the VBUS operation; it must be supplied externally. The VBUSON output can be used to control an off-chip 5V VBUS regulator. The VBUSON bit (UxOTGCON<3>) controls the VBUSON pin. Figure 25-8 and Figure 25-9 illustrate the VBUSON pin configuration.

In applications, where an external +5V power supply is not available, the USB module features a Voltage Boost Regulator module. This module along with the external boost circuitry can be used to convert +3.3V to 5.0V required for VBUS. The Boost Regulator module is enabled by clearing the UVBUSDIS bit (UxCNFG2<2>). The USBPWR bit (UxPWRC<0>) and the VBUSON bit (UxOTGCON<3>) should be set.

The Boost Regulator module features a PWM generator which is controlled by UxPMCON and UxPWMRRS registers. This module uses the VCPCON and the VBUSST pins to control an external voltage boost circuitry. The VCPCON pin outputs a Pulse Width Modulated signal. The application must set the PWMEN (UxPWMCON<15>) and CNTEN (UxPWMCON<8>) bits to enable the PWM output. The PWM period is controlled by the PER<7:0> bits in the UxPWMRRS register. The PWM duty cycle is controlled by the DC<7:0> bits in the UxPWMRRS register. The VBUSST pin can be used to implement an over-current protection circuit. The Boost Regulator module disables the VCPCON output when the voltage at the VBUSST pin rises above VBUSST_TH.

Note: Refer to the **"Electrical Characteristics"** chapter in the specific device data sheet for information on the VBUSST_TH voltage level.

The schematic of an external boost circuit controlled by the Boost Regulator module is illustrated in Figure 25-6. When S1 and S2 switches are closed, current builds up in the inductor L1. When S1 and S2 switches are open, L1 charges the capacitor C1. The cycle repeats until the VBUS reaches the desired regulated voltage.

The VBUSCHG bit (UxOTGCON<1>) specifies the desired VBUS voltage. Setting this bit specifies the VBUS voltage to be 3.3V. Clearing this bit specifies the VBUS voltage to be 5V. The Boost Regulator module monitors the voltage on the VBUS pin. If the VBUS voltage exceeds the set voltage, the Boost Regulator module disables the PWM output on the VCPCON pin.

Equation 25-1 is used to calculate the PWM period of Boost Regulator module.

Equation 25-1:

$$PWM\ Period = (PER<7:0> + 1) \times \frac{1}{FCY}$$

Note: The PER<7:0> bits should have values greater than '0'. FCY is the CPU clock frequency.

The duty cycle of the PWM signal is specified by the DC<7:0> bits (UxPWMRRS<15:8>) and is calculated by using the Equation 25-2.

Equation 25-2:

$$Duty\ Cycle = \frac{PER<7:0> - DC<7:0>}{(PER<7:0> + 1)}$$

Example 25-1 provides the code example to configure the boost regulator to operate at 4 MHz with a 25% duty cycle.

Example 25-1:

```
/*
The CPU is operating at 60 MHz. The desired PWM frequency is 4 MHz. As
per Equation 25-1, the value of the PER bits in U1PWMRRS register is
calculated to be 0xE. The desired duty cycle is 25%. Using Equation 25-2,
the value of the DC bits in U1PWMRRS register is calculated to be 0xA.
*/


U1PWRCbits.USBPWR = 1;          // USB OTG module enabled
U1OTGCONbits.VBUSON = 1;        // VBUS line powered
U1PWMRRSbits.PER = 0xE;         // Load the Period
U1PWMRRSbits.DC = 0xA;          // Load the Duty Cycle
U1PWMCONbits.CNTEN = 1;         // Counter is enabled
U1CNFG2bits.UVBUSDIS = 0;       // On-chip Boost Controller active
U1OTGCONbits.VBUSCHG = 1;       // VBUS line set to charge to 3.3V
U1PWMCONbits.PWMPOL = 1;        // PWM output is active-low and resets high
U1PWMCONbits.PWMEN = 1;         // PWM generator is enabled
```

25.3.2.5.4 Interfacing an External Transceiver and VBUS Comparator

The USB module of the dsPIC33E/PIC24E device families features an external USB transceiver interface. Setting the UTRDIS bit (UxCNFG2<0>) disables the internal transceiver and enables the external transceiver interface.

The external transceiver interface provides several options to connect to a range of commercially available USB transceivers. The VMIO, VPIO, RCV and USBOEN pins of the external transceiver interface of the USB module can be connected to the similar pins on the external transceiver. In cases, where the external transceivers have discrete pins for controlling the D+ and D- pull-up and pull-down resistors, the external transceiver interface of dsPIC33E/PIC24E device family provides the DMH, DMLN, DPH and DPLN pins.

- The DMH pin goes high when DMPULUP bit (UxOTGCON<6>) is set. This requests the external transceiver to activate the pull-up resistors on its D- line.
- The DMLN pin goes high when DMPULDWN bit (UxOTGCON<4>) is set. This requests the external transceiver to activate the pull-down resistors on its D- line.
- The DPH pin goes high when DPPULUP bit (UxOTGCON<7>) is set. This requests the external transceiver to activate the pull-up resistors on its D+ line.
- The DPLN pin goes high when DPPULDWN bit (UxOTGCON<5>) is set. This requests the external transceiver to activate the pull-down resistors on its D+ line.

In cases, where an external transceiver has an I^2C bus interface for controlling the D+ and D- line pull-up and pull-down resistors, the dsPIC33E/PIC24E application can use the available device I^2C modules to communicate with the external transceiver. While using the I^2C bus, the application can disable the DMH, DPH, DMLN and DPLN pins by setting the EXTI2CEN bit (UxCNFG2<3>). These pins then behave as general purpose I/O.

The application can disable the internal VBUS comparator and use an external VBUS comparator by setting the UVCMPDIS bit (UxCNFG2<1>). Setting the UVCMPDIS bit enables the external VBUS comparator interface. In such a case, the SESVDIF, SESENDIF and VBUSVDIF bits in the UxOTGIR register are affected by logic levels on the external VBUS comparator interface.

A commercially available VBUS comparator may provide two/three status outputs. The USB module of dsPIC33E/PIC24E device family provides an option to connect to either of these. The external VBUS comparator interface is implemented through the VCMPST1, VCMPST2 and VCMPST3 pins. The UVCMPSEL bit (UxCNFG2<5>) configures the external comparator interface to use two or three pins for its operation.

A three-pin VBUS comparator interface is selected by setting the UVCMPSEL bit (UxCNFG2<5>). Table 25-2 shows the relationship between the logic levels on the VCMPST1, VCMPST2 and VCMPST3 pins and the SESVDIF, SESENDIF and VBUSVDIF bits.

Table 25-2: External VBUS 3-pin Interface Truth Table

External VBUS Comparator Interface Pins			USB Module VBUS Status Bits		
VCMPST3	VCMPST2	VCMPST1	VBUSVDIF	SESENDIF	SESVDIF
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	0	1
1	0	0	0	1	0

A two-pin external interface is selected by clearing the UVCMPSEL bit (UxCNFG2<5>). In this case, only the VCMPST1 and VCMPST2 pins are used. Table 25-3 shows the relationship between the logic levels on the VCMPST1 and VCMPST2 pins and the VBUSVDIF (UxOTGIR<0>), SESENDIF (UxOTGIR<2>) and SESVDIF (UxOTGIR<3>) bits.

Table 25-3: External VBUS 2-pin Interface Truth Table

External VBUS Comparator Interface Pins		USB Module VBUS Status Bits		
VCMPST2	VCMPST1	VBUSVDIF	SESENDIF	SESVDIF
0	0	0	1	0
0	1	0	0	0
1	0	0	0	1
1	1	1	0	1

25.3.3 dsPIC33E/PIC24E USB Implementation Specifics

This section details how the *"Universal Serial Bus Specification"* (Revision 2.0) requirements are implemented in the dsPIC33E/PIC24E USB module.

25.3.3.1 BUS SPEED

The dsPIC33E/PIC24E USB module supports the following bus speeds:

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

25.3.3.2 ENDPOINTS AND DESCRIPTORS

All USB endpoints are implemented as buffers in RAM. The CPU and USB module have access to the buffers. To arbitrate access to these buffers between the USB module and CPU, a semaphore flag system is used. Each endpoint can be configured for TX and/or RX, and each has an ODD and an EVEN buffer, resulting in up to four buffers per endpoint.

Use of the BDT allows the buffers to be located anywhere in RAM, and provides status flags and control bits. The BDT contains the address of each endpoint data buffer and information about each buffer (see Figure 25-5). Each BDT entry is called a Buffer Descriptor (BD) and is 8 bytes long. Four descriptor entries are used for each endpoint. All endpoints, ranging from Endpoint 0 to the highest endpoint in use, must have four descriptor entries. Even if all of the buffers for an endpoint are not used, four descriptor entries are required for each endpoint.

The USB module calculates a buffer's location in memory using the BDT Pointer registers. The base of the BDT is held in registers U1BDTP1 through U1BDTP3. The address of the desired buffer is 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.

Note: The contents of the U1BDTP1-U1BDTP3 registers provide the upper 23 bits of the 32-bit address; therefore, the BDT must be aligned to a 512-byte boundary (see Figure 25-2).

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

Having EVEN and ODD buffers 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 in the buffer descriptor 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 data transfer. This technique is referred to as ping-pong buffering. Figure 25-5 illustrates how the endpoints are mapped in the BDT.

25.3.3.2.1 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 U1EPn register also allows support of control transfers.

25.3.3.2.2 Host Endpoints

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

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

25.3.3.2.3 Device Endpoints

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

25.3.3.3 BUFFER MANAGEMENT

The buffers are shared between the CPU and the USB module, and are implemented in system memory. So, a simple semaphore mechanism is used to distinguish the current ownership of the BD, and associated buffers, in memory. This semaphore mechanism is implemented by the UOWN bit in each BD.

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 CPU – which may modify the descriptor and buffer as necessary.

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

A BD is valid only if the corresponding endpoint has been enabled in the U1EPn register. The BDT is implemented in data memory, and the BDs are not modified when the USB module is reset. Initialize the BDs prior to enabling them through the U1EPn. At a minimum, the UOWN bits must be cleared prior to being enabled.

In Host mode, BDT initialization is required before the U1TOK register is written, which triggers a transfer.

Figure 25-2 illustrates how the USB module generates the address of (and accesses) a buffer descriptor associated with an endpoint. Bits 31 through 9 are specified by the contents of U1BDTP1, U1BDTP2 and U1BDTP3 registers. Bits 8 through 0 are generated by the USB module.

Figure 25-2: BDT Address Generation

BDTBA<22:0>	ENDPOINT<3:0>	DIR	PPBI	FIELD
31:9	8:5	4	3	2:0

bit 31-9 **BDTBA<22:0>:** BDT Base Address bits
The 23-bit value is made up of the contents of the U1BDTP3, U1BDTP2 and U1BDTP1 registers.

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

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**
Used to access fields within the BD.

25.3.3.3.1 Buffer Descriptor Format

The buffer descriptor is used in the following formats:

- Control
- Status

Buffer descriptor control format, in which software writes the descriptor and hands it to hardware, is illustrated in Figure 25-3.

Figure 25-3: Buffer Descriptor Control Format: Software to Hardware

WORD0	15 8 Unimplemented	7 UOWN	6 DATA0/1	5 KEEP	4 NINC	3 DTSEN	2 BSTALL	1 0 Reserved
WORD1	15 10 Unimplemented	9 0 BYTECOUNT<9:0>						
WORD2	15 0 BUFFERADDRESS<15:0>							
WORD3	15 0 BUFFERADDRESS<31:16>							

WORD0 bit 15-8 **Unimplemented**

bit 7 **UOWN:** USB Own bit
- 1 = The USB module owns the endpoint buffer descriptor and its corresponding buffer. The CPU must not modify the buffer descriptor or the buffer
- 0 = The CPU can modify the buffer

bit 6 **DATA0/1:** Data Toggle Packet bit
- 1 = Transmit a Data 1 packet or Check received PID == DATA1, if DTSEN = 1
- 0 = Transmit a Data 0 packet or Check received PID == DATA1, if DTSEN = 1

bit 5 **KEEP:** Buffer Descriptor Keep Enable bit
- 1 = USB will keep the buffer descriptor indefinitely once the UOWN bit is set
- 0 = USB will hand back the buffer descriptor once a token has been processed

bit 4 **NINC:** Address Increment Disable bit
- 1 = Address increment is disabled
- 0 = Address increment is enabled

bit 3 **DTSEN:** Data Toggle Synchronization Enable bit
- 1 = Data Toggle Synchronization is enabled; data packets with incorrect sync value will be ignored
- 0 = No Data Toggle Synchronization is performed

bit 2 **BSTALL:** Buffer STALL Enable bit
- 1 = Buffer STALL enabled; STALL handshake is issued if a token is received that would use the buffer descriptor in the given location (UOWN bit remains set, buffer descriptor value is unchanged). The EPSTALL bit in the corresponding UxEPn register is set on any STALL handshake.
- 0 = Buffer will be consumed by the USB module. A STALL handshake will not be issued.

WORD0 bit 1-0 **Reserved:** Do not modify

WORD1 bit 15-10 **Unimplemented**

bit 9-0 **BYTECOUNT<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.

WORD2 bit 15-0 **BUFFERADDRESS<15:0>:** Buffer Address bits

The starting point address bits through <15:0> of the endpoint packet data buffer.

WORD3 bit 15-0 **BUFFERADDRESS<31:16>:** Buffer Address bits

The starting point address bits through <31:16> of the endpoint packet data buffer.

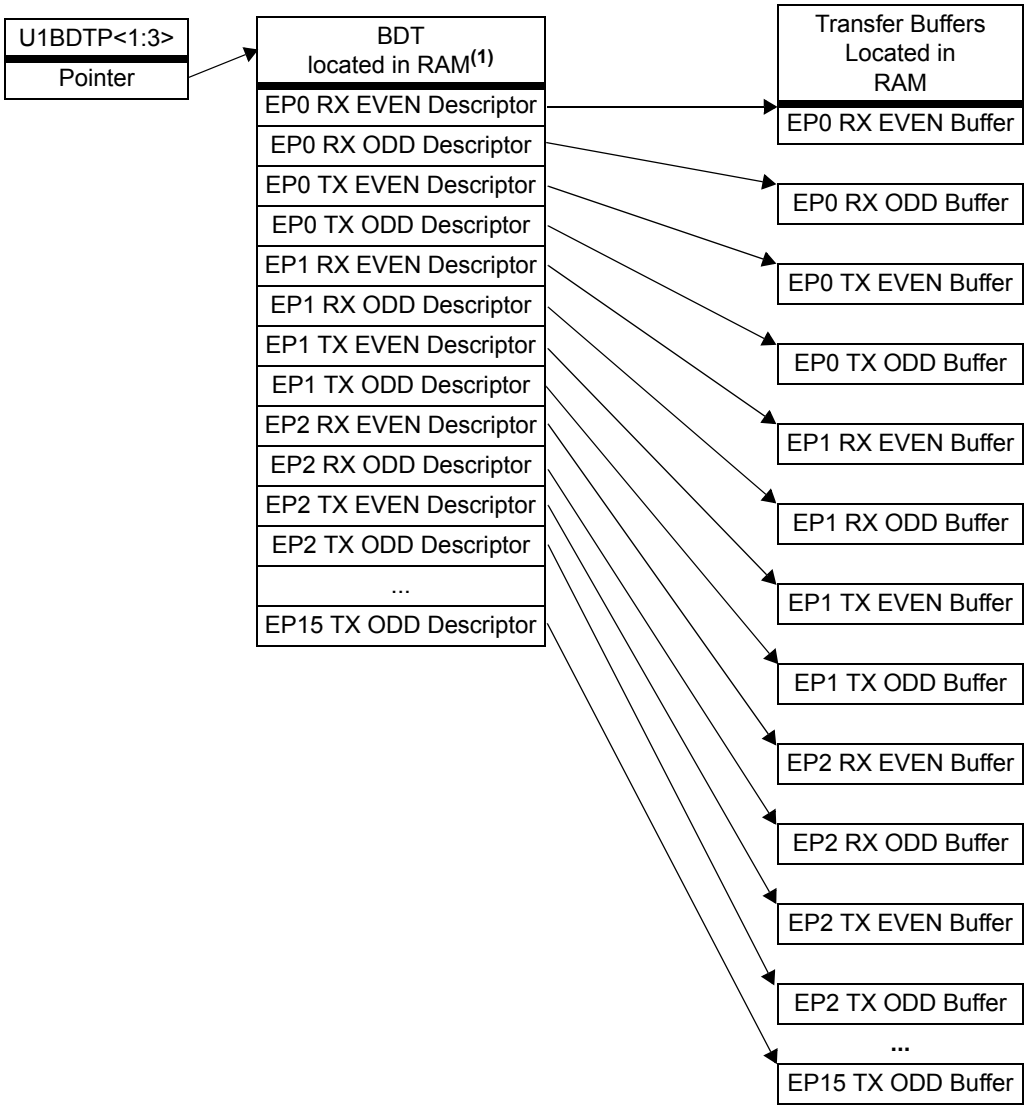
Buffer descriptor status format, in which hardware writes the descriptor and hands it back to software, is illustrated in Figure 25-4.

Figure 25-4: Buffer Descriptor Status Format: Hardware to Software

WORD0	15 Unimplemented 8	7 UOWN	6 DATA0/1	5 PID<3:0> 2	1 Reserved 0
WORD1	15 Unimplemented 10	9 BYTECOUNT<9:0> 0			
WORD2	15 BUFFERADDRESS<15:0> 0				
WORD3	15 BUFFERADDRESS<31:16> 0				

WORD0	bit 15-10	**Unimplemented**
	bit 7	**UOWN:** USB Own bit 1 = The USB owns the endpoint buffer descriptor and its corresponding buffer. The CPU must not modify the buffer descriptor or the buffer 0 = The CPU can modify the buffer
	bit 6	**DATA0/1:** Data Toggle Packet bit 1 = Data 1 packet was received 0 = Data 0 packet was received This bit is unchanged when a packet is transmitted.
	bit 5-2	**PID<3:0>:** Packet Identifier 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 and 0xf Data Error.
WORD0	bit 1-0	**Reserved:** Do not modify
WORD1	bit 15-10	**Unimplemented**
	bit 9-0	**BYTECOUNT<9:0>:** Byte Count Byte count reflects the actual number of bytes received or transmitted.
WORD2	bit 15-0	**BUFFERADDRESS<15:0>:** Buffer Address The starting point address bits through <15:0> of the endpoint packet data buffer.
WORD3	bit 15-0	**BUFFERADDRESS<31:16>:** Buffer Address The starting point address bits through <31:16> of the endpoint packet data buffer.

Figure 25-5: Buffer Management Overview



Note 1: 512-byte aligned.

25.3.3.4 BUFFER DESCRIPTOR CONFIGURATION

The UOWN, DTS and BSTALL bits in each BDT entry control the data transfer for the associated buffer and endpoint.

Setting the DTS bit enables the USB module to perform data toggle synchronization. When DTS is enabled: if a packet arrives with an incorrect DTS, it will be ignored. The buffer remains unchanged, and the packet will be negatively acknowledged (NAK'd).

Setting the BSTALL bit causes the USB to issue a STALL handshake, if a token is received by the SIE that would use the BD in this location – the corresponding EPSTALL bit is set and a STALLIF interrupt is generated. When the BSTALL bit is set, the BD is not consumed by the USB module (the UOWN bit remains set and the rest of the BD values are unchanged). If a SETUP token is sent to the stalled endpoint, the module automatically clears the corresponding BSTALL bit.

The byte count represents the total number of bytes that are transmitted or received. Valid byte counts range 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, after the transfer is completed. If number of bytes received exceeds the corresponding byte count value written by the firmware, the overflow bit is set and the data is truncated to fit the size of the buffer (as given in the BDT).

25.3.4 Hardware Interface

25.3.4.1 POWER SUPPLY REQUIREMENTS

Power supply requirements for USB implementation vary with the type of application, and are outlined as follows.

- Device:

  Operation as a device requires a power supply for the dsPIC33E/PIC24E and the USB transceiver, see Figure 25-7 for an overview of USB implementation as a device.

- Embedded Host:

  Operation as a host requires a power supply for the dsPIC33E/PIC24E, 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 the devices in the TPL. The application dictates whether the VBUS power supply can be disabled or disconnected from the bus by the dsPIC33E/PIC24E application. Figure 25-8 illustrates an overview of USB implementation as a host.

- OTG Dual Role:

  Operation as an OTG dual role requires a power supply for the dsPIC33E/PIC24E, the USB transceiver, and a switchable 5V nominal supply for the USB VBUS. An overview of USB implementation as OTG is illustrated in Figure 25-9.

  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 devices in the TPL.

  When acting as a B-device, power must not be supplied to VBUS. VBUS pulsing can be performed by the USB module or by a capable power supply.

25.3.4.2 VBUS REGULATOR INTERFACE

The VBUSON output can be used to control an off-chip 5V VBUS regulator. The VBUSON pin is controlled by the VBUSON bit (U1OTGCON<3>). The VBUSON pin configuration is illustrated in Figure 25-8 and Figure 25-9.

Figure 25-6 illustrates a conceptual schematic of an external boost circuit controlled by the VBUS boost regulator.

Figure 25-6: Conceptual Circuit for Obtaining VBUS from a 3.3V Power Supply Using the USB OTG Module

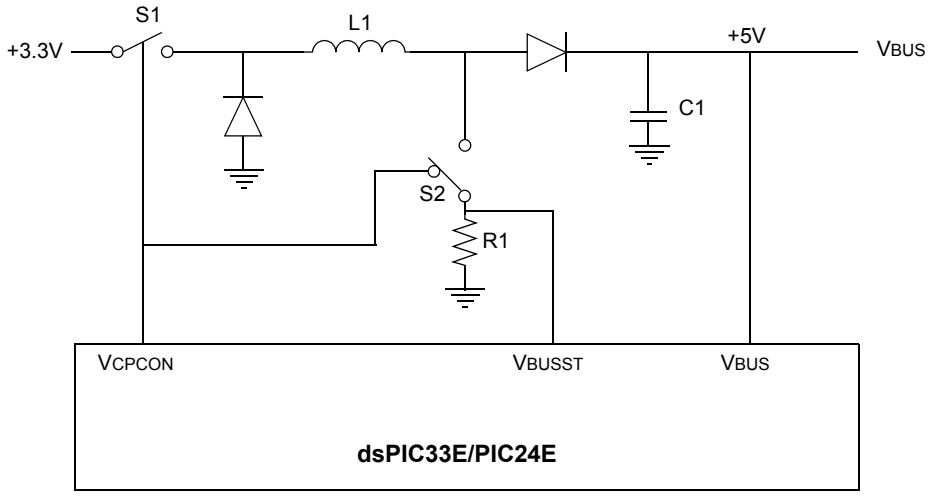
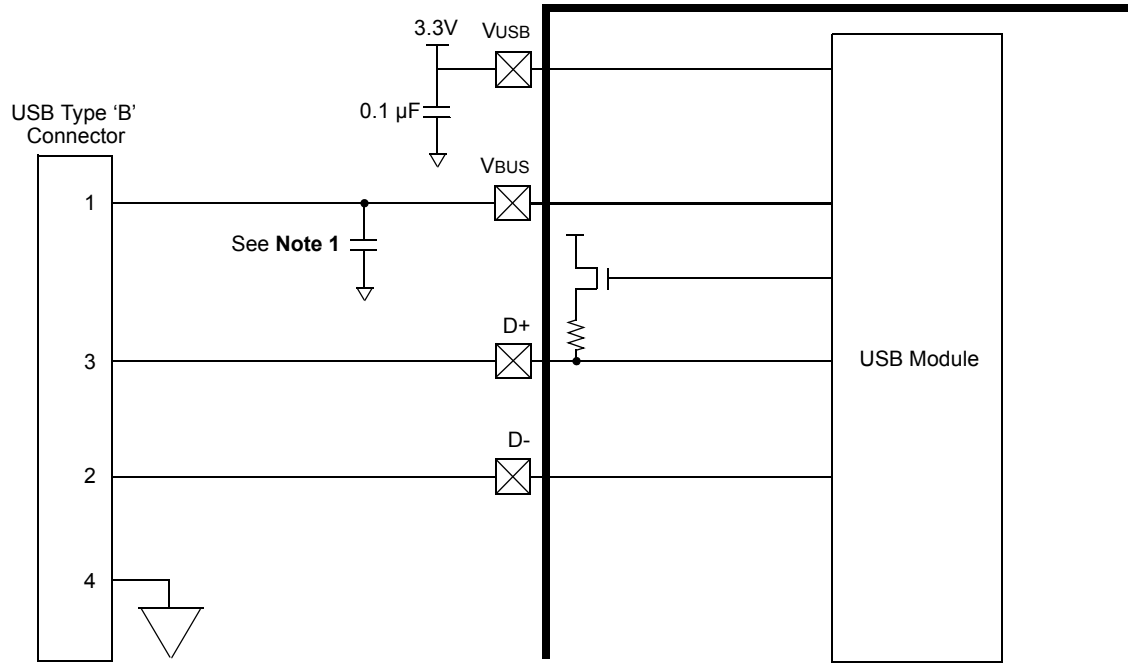
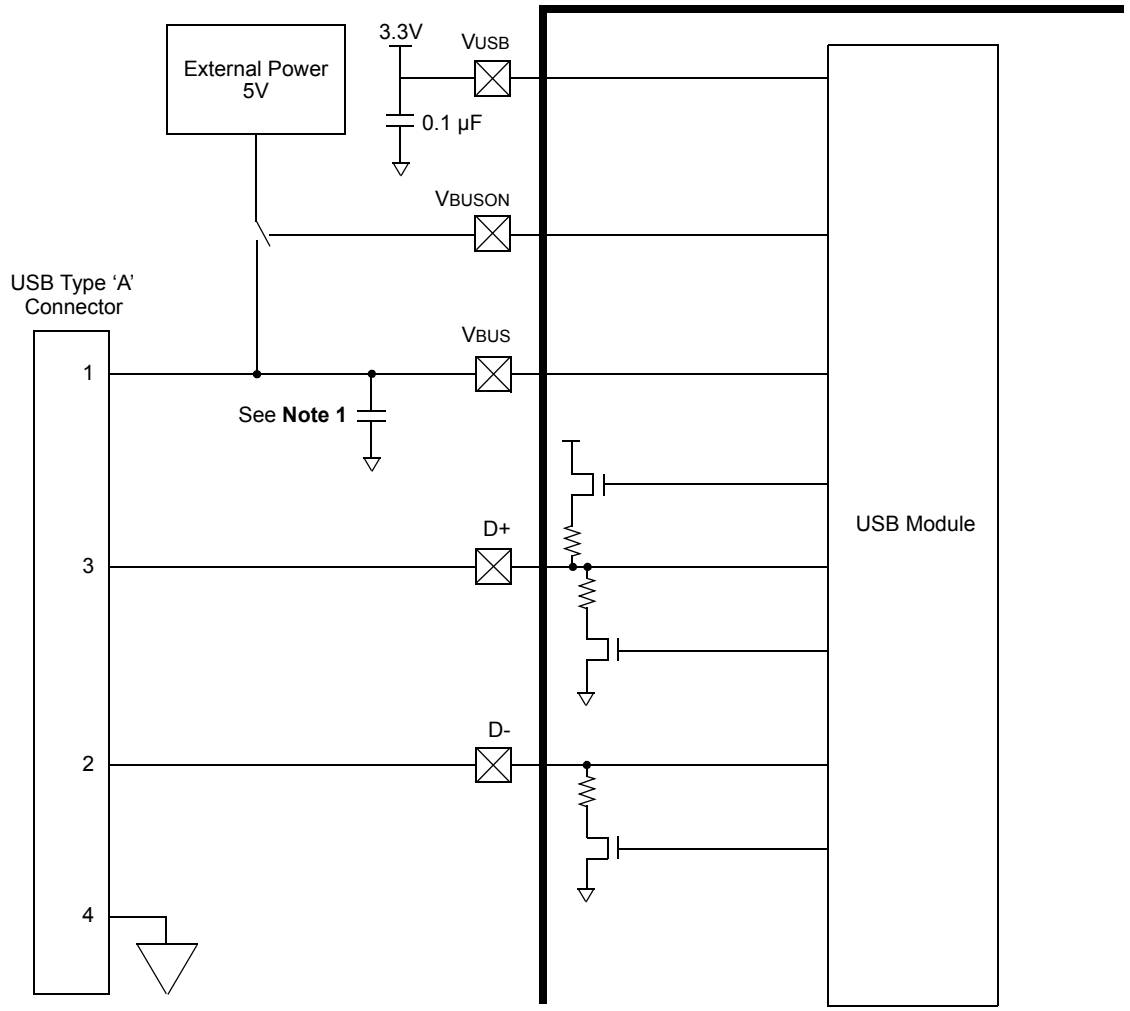


Figure 25-7: Overview of USB Implementation as a Device



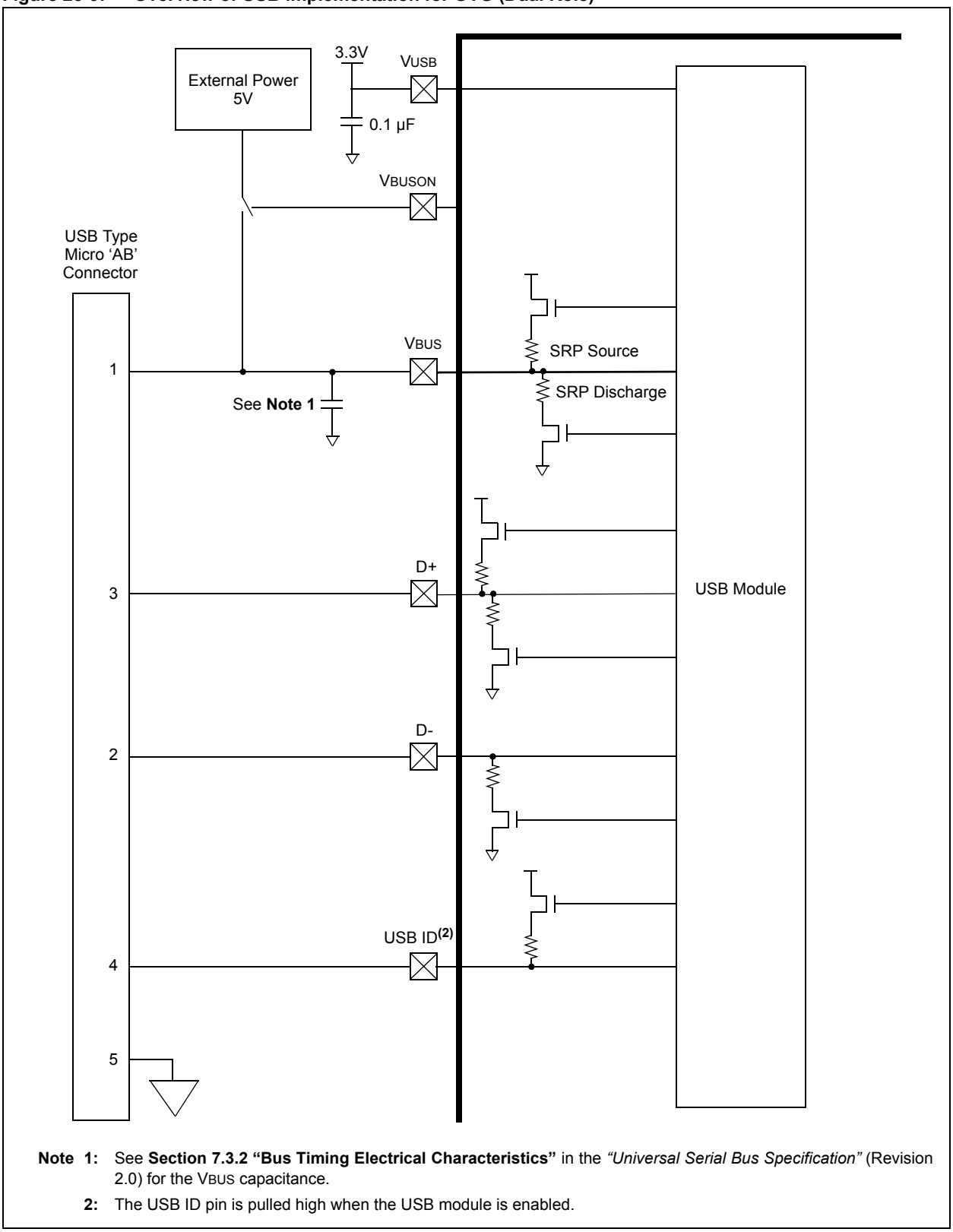
Note 1: See **Section 7.3 "Physical Layer"** in the *"Universal Serial Bus Specification"* (Revision 2.0) for the VBUS capacitance value in a device implementation.

Figure 25-8: Overview of USB Implementation as a Host



Note 1: See **Section 7.3 "Physical Layer"** in the *"Universal Serial Bus Specification"* (Revision 2.0) for the VBUS capacitance value in a device implementation.

Figure 25-9: Overview of USB Implementation for OTG (Dual Role)



Note 1: See **Section 7.3.2 "Bus Timing Electrical Characteristics"** in the *"Universal Serial Bus Specification"* (Revision 2.0) for the VBUS capacitance.

2: The USB ID pin is pulled high when the USB module is enabled.

25.3.5 Module Initialization

This section describes the steps to be performed to properly initialize the USB OTG module.

25.3.5.1 ENABLING THE USB HARDWARE

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

The USBPWR bit is used to initiate the following actions:

- Start the USB clock
- Allow the USB interrupt to be activated
- Select USB as the owner of the necessary I/O pins
- Enable the USB transceiver
- Enable the USB comparators

The USB module and internal registers are reset when USBPWR bit is cleared. Consequently, the appropriate initialization process must be performed whenever the USB module is enabled, as described in the following subsections. Otherwise, any configuration packet sent to the USB module will be NAK'd, by hardware, until the module is configured.

25.3.5.2 INITIALIZING THE BDT

All descriptors for a given endpoint and direction must be initialized prior to enabling the endpoint (for that direction). After a reset, all endpoints are disabled and start with the EVEN buffer for transmit and receive directions.

Transmit descriptors must be written with the UOWN bit cleared to '0' (owned by software). All other transmit descriptor setup may be performed anytime prior to setting the UOWN bit to '1'.

Receive descriptors must be fully initialized to receive data. This means that memory must be reserved for received packet data. The pointer to that memory (Physical Address) and the size reserved in bytes, must be written to the descriptor. The receive descriptor UOWN bit should be initialized to '1' (owned by Hardware). The DTS and STALL bits should also be configured appropriately.

If a transaction is received and the descriptor's UOWN bit is '0' (owned by software), the USB module returns a NAK handshake to the host. Usually, this causes the host to retry the transaction.

25.3.5.3 USB ENABLE/MODE BITS

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

- **OTGEN:** Selects whether the dsPIC33E/PIC24E is to act as an OTG part (OTGEN = 1) or not. OTG devices support SRP and HNP in hardware with Firmware management having direct control over the data-line pull-up and pull-down resistors.
- **HOSTEN:** Controls whether the part is acting in the role of USB Host (HOSTEN = 1) or USB Device (HOSTEN = 0). Note that this role may change dynamically in an OTG application.
- **USBEN/SOFEN:** Controls the connection to USB by enabling the D+ pull-up resistor when the USB module is not configured as a host.

  If the USB module is configured as a host, the SOFEN bit controls whether the host is active on the USB link. Setting the SOFEN bit makes the HOST send out the SOF tokens every 1 ms.

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

25.3.6 Device Operation

All communication on the USB is initiated by the host. Therefore, in device mode, when USB is enabled, USBEN (U1CON<0>) = 1, Endpoint 0 must be ready to receive control transfers. Initialization of the remaining endpoints, descriptors and buffers can be delayed until the host selects a configuration for the device. Refer to **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0) for more information on this subject.

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

1. Software preinitializes the appropriate BDs and sets the UOWN bits to '1' to be ready for a transaction.
2. Hardware receives a TOKEN PID (IN, OUT, SETUP) from the USB host and checks the appropriate BD.
3. If the transaction is transmitted (IN), the module reads packet data from data memory.
4. Hardware receives a DATA PID (DATA0/1) and sends or receives the packet data.
5. If a transaction is received (SETUP, OUT), the module writes packet data to data memory.
6. The module issues or waits for a handshake PID (ACK, NAK, STALL), unless the endpoint is setup as an isochronous endpoint (EPHSHK bit UEPMx<0> is cleared).
7. The module updates the BD and writes the UOWN bit to '0' (software-owned).
8. The module updates the U1STAT register and sets the TRNIF interrupt.
9. Software reads the U1STAT register, and determines the endpoint and direction for the transaction.
10. Software reads the appropriate BD, completes all necessary processing, and clears the TRNIF interrupt.

Note: For transmitted (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'.

25.3.6.1 RECEIVING AN IN TOKEN IN DEVICE MODE

Perform the following steps when an IN token is received in Device mode:

1. Attach to a USB host and enumerate as described in **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0).
2. Populate the data buffer with the data to send to the host.
3. In the appropriate (EVEN or ODD) transmit buffer descriptor for the desired endpoint:
   a) Set up the control bit fields with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address bit field with the starting address of the data buffer.
   c) Set the UOWN bit field to '1'.
4. When the USB module receives an IN token, it automatically transmits the data in the buffer. On completion, the module updates the status bit fields and sets the transfer complete interrupt (U1IR<TRNIF>).

25.3.6.2 RECEIVING AN OUT TOKEN IN DEVICE MODE

Perform the following steps when an OUT token is received in Device mode:

1. Attach to a USB host and enumerate as described in **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0).
2. Create a data buffer with the amount of data you are expecting from the host.
3. In the appropriate (EVEN or ODD) transmit buffer descriptor for the desired endpoint:
   a) Set up the control bit fields with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address bit field with the starting address of the data buffer.
   c) Set the UOWN bit to '1'.
4. When the USB module receives an OUT token, it automatically transfers the data the host sent into the buffer. On completion, the module updates the status bit fields and sets the transfer complete interrupt (U1IR<TRNIF>).

25.4 HOST MODE OPERATION

In Host mode, only Endpoint 0 is used (all other endpoints should be disabled). As the host initiates all transfers, the BD does not require immediate initialization. However, the BDs must be configured before a transfer is initiated, which is done by writing to 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 initiating the setup, data and status stages of all control transfers. The acknowledge (ACK or NAK) is generated automatically by the hardware, based on the CRC. Host software is also responsible for scheduling packets so that they do not violate USB protocol. All transfers are performed using the Endpoint 0 Control register (U1EP0) and BDs.

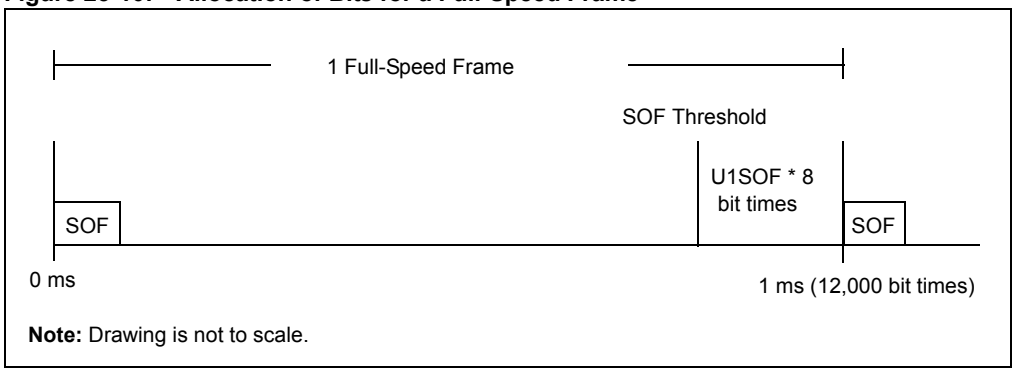
25.4.1 Configuring the SOF Threshold

The module counts the number of bits that could be transmitted within the current USB full-speed frame. As 12,000 bits can be transmitted during the 1 ms frame time, a counter (not visible to software) is loaded with the value '12,000' at the start of each frame. The counter decrements once for each bit time in the frame. When the counter reaches zero, the next frame's SOF packet is transmitted (see Figure 25-10).

The SOF threshold register (U1SOF) is used to ensure that no new tokens are started too close to the end of a frame. This prevents a conflict with the next frame's SOF packet. When the counter reaches the threshold value of the U1SOF register (the value in the U1SOF register is in terms of bytes), no new tokens are started until after the SOF has been transmitted. Thus, the USB module attempts to ensure that the USB link is idle when the SOF token needs to be transmitted.

This implies that the value programmed into the U1SOF register must reserve enough time to ensure the completion of the worst-case transaction. Typically, the worst-case transaction is an IN token followed by a maximum-sized data packet from the target, followed by the response from the host. If the host is targeting a low-speed device that is bridging through a full-speed hub, the transaction will also include the special PRE token packets.

Figure 25-10: Allocation of Bits for a Full-Speed Frame



Note: Drawing is not to scale.

Table 25-4 and Table 25-5 show examples of calculating worst-case bit times.

Note 1: While the U1SOF register value is described in terms of bytes, these examples show the result in terms of bits.

2: In the second table, the IN, DATA and HANDSHAKE packets are transmitted at low-speed (8 times slower than full-speed).

3: These calculations do not take the possibility that the packet data needs to be bit-stuffed for NRZI encoding into account.

Table 25-4: Example of SOF Threshold Calculation: Full-Speed

Packet	Fields	Bits
IN	SYNC, PID, ADDR, ENDP, CRC5, EOP	35
Turnaround(1)	—	8
DATA	SYNC, PID, DATA(2), CRC16, EOP	547
Turnaround	—	2
HANDSHAKE	SYNC, PID, EOP	19
Inter-packet	—	2
Total Bit Times		613

Note 1: Inter-packet delay of 2. An additional 5.5 bit times of latency is added to represent a worst-case propagation delay through 5 hubs.

2: Using 64 bytes maximum packet size for this example calculation.

Table 25-5: Example of SOF Threshold Calculation: Low-Speed Via Hub

Packet	Fields	Bits	FS Bits
PRE	SYNC, PID	16	16
Hub Setup	—	4	4
IN	SYNC, PID, ADDR, ENDP, CRC5, EOP	35	280
Turnaround(1)	—	8	8
DATA	SYNC, PID, DATA(2), CRC16, EOP	99	792
Turnaround	—	2	2
PRE	SYNC, PID	16	16
HANDSHAKE	SYNC, PID, EOP	19	152
Inter-packet	—	2	2
Total Bit Times			1272

Note 1: Inter-packet delay of 2. An additional 5.5 bit times of latency is added to represent a worst-case propagation delay through 5 hubs.

2: Packets limited to 8 bytes maximum in Low-Speed mode.

Note: Refer to **Section 5.11.3 "Calculating Bus Transaction Times"** in the *"Universal Serial Bus Specification"* (Revision 2.0) for details on calculating bus transaction time.

25.4.2 Enabling Host Mode and Discovering a Connected Device

To enable Host mode, perform the following steps:

1. Enable Host mode (U1CON<HOSTEN> = 1).
   This enables the D+ and D- pull-down resistors, and disables the D+ and D- pull-up resistors. To reduce noise on the bus, disable the SOF packet generation by writing the SOF enable bit to '0' (U1CON<SOFEN> = 0).
2. Enable the device attach interrupt (U1IE<ATTACHIE> = 1).
3. Wait for the device attach interrupt (U1IR<ATTACHIF>).
   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 is minimum, 100 ms is recommended).
4. Check the state of the JSTATE and SE0 bits in the control register U1CON.
   If U1CON<JSTATE> is '0', the connecting device is low-speed; otherwise, the device is full-speed.
5. If the connecting device is low-speed, set the low-speed enable bit in the address register (U1ADDR<LSPDEN>= 1), and the low-speed bit in the Endpoint 0 Control register (U1EP0<LSPD> = 1). But, if the device is full-speed, clear these bits.
6. Reset the USB device by sending the Reset signaling for at least 50 ms (U1CON<USBRST> = 1). After 50 ms, terminate the Reset (U1CON<USBRST> = 0).
7. Enable SOF packet generation to keep the connected device from going into Suspend (U1CON<SOFEN> = 1).
8. Wait 10 ms for the device to recover from Reset.
9. Perform enumeration as described in **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0).

25.4.2.1 HOST TRANSACTIONS

When acting as a host, a transaction consists of the following:

1. Software configures the appropriate BD and sets the UOWN bit to '1' (hardware-owned).
2. Software checks the state of TOKBUSY (U1CON<5>) to verify that any previous transaction has completed.
3. Software writes the address of the target device in the U1ADDR register.
4. Software writes the endpoint number and the desired TOKEN PID (IN, OUT or SETUP) to the U1TOK register.
5. Hardware reads the BD to determine the appropriate action and to obtain the pointer to data memory.
6. Hardware issues the correct TOKEN PID (IN, OUT, SETUP) on the USB link.
7. If the transaction is a transmit transaction (OUT, SETUP), the USB module reads the packet data out of data memory. Then the module follows with the desired DATA PID (DATA0/DATA1) and packet data.
8. If the transaction is a receive transaction (IN), the USB module waits to receive the DATA PID and packet data. Hardware writes the packet data to memory.
9. Hardware issues or waits for a Handshake PID (ACK, NAK or STALL), unless the endpoint is set up as an Isochronous Endpoint (EPHSHK bit U1EPx<0> is cleared).
10. Hardware updates the BD and writes the UOWN bit to '0' (software-owned).
11. Hardware updates the U1STAT register and sets the TRNIF (U1IR<3>) interrupt.
12. Hardware reads the next BD (EVEN or ODD) to see whether it is owned by the USB module. If it is, hardware begins the next transaction.
13. Software should read the U1STAT register and then clear the TRNIF interrupt.

If Software does not set the UOWN bit to '1' in the appropriate BD prior to writing the U1TOK register, the module just reads the descriptor and does not perform any action.

25.4.3 Completing a Control Transaction to a Connected Device

To discover a connected device, perform the following steps:

1. Set up the Endpoint Control register for bidirectional control transfers, U1EP0<4:0> = 0x0D.
2. Place an 8-byte device setup packet in the appropriate memory buffer. See **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0) for information on the device framework command set.
3. Initialize the current (EVEN or ODD) TX EP0 BD to transfer the 8-byte device framework command (for example, a `GET DEVICE DESCRIPTOR` command).
   a) Set the BD control offset 0 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, if it is not already initialized.
4. Set the USB 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 must be set to another value, between 1 and 127, by the host software.
5. Write the token register with a `SETUP` command 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 BD status after the packets complete. When the module updates BD status, a transfer done interrupt will be asserted (U1IR<TRNIF>). This completes the setup stage of the setup transfer as described in **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0).
6. To initiate the data stage of the setup transaction (for example, get the data for the `GET DEVICE DESCRIPTOR` command), set up a buffer in memory to store the received data.
7. 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 control UOWN bit to '`1`', data toggle (DTS) to DATA1 and byte count to the length of the data buffer.
   b) Set the BD data buffer address (BD0ADR) to the starting address of the data buffer, if it is not already initialized.
8. Write the Token register with the appropriate IN or OUT token to Endpoint 0 (the target device's default control pipe). For example, an IN token for a `GET DEVICE DESCRIPTOR` command (U1TOK = 0x90). This initiates an IN token on the bus, followed by a data packet from the device to the host. When the data packet completes, the BD status 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. If more data needs to be transferred, return to step 6.
9. To initiate the status stage of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
10. Initialize the current (EVEN or ODD) TX EP0 BD to transfer the status data.
    a) Set the BD control 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.
11. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe. For example, an OUT token for a `GET DEVICE DESCRIPTOR` command (U1TOK = 0x10). This initiates a token on the bus, followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device, and a transfer done interrupt is asserted (U1IR<TRNIF>). This completes the status phase of the setup transaction.

Note: Some devices can only effectively respond to one transaction per frame.

25.4.4 Data Transfer with a Target Device

To discover and configure a connected device, perform the following steps:.

1. Write the EP0 Control register (U1EPn) to enable transmit and receive transfers as appropriate with handshaking enabled (unless isochronous transfers are to be used). If the target device is a low-speed device, then set the Low-Speed Enable bit (U1EPn<LSPDEN>). If you want the hardware to automatically retry indefinitely if the target device asserts a NAK on the transfer, clear the Retry Disable bit (U1EPn<RETRYDIS>).

Note:	Use of automatic indefinite retries can lead to a deadlock condition if the device never responds.

2. Set up the current Buffer Descriptor (EVEN or ODD) in the appropriate direction to transfer the desired number of bytes.
3. Set the address of the target device in the address register (U1ADDR<6:0>).
4. Write the Token register (U1TOK) with an IN or OUT token as appropriate for the desired endpoint. This triggers the module's transmit state machines to begin transmitting the token and the data.
5. Wait for the transfer done interrupt (U1IR<TRNIF>). This indicates that the BD has been released back to the microprocessor and the transfer has completed. If the retry disable bit is set, the handshake (ACK, NAK, STALL or ERROR (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<DETACHIF>).
6. Once the transfer done interrupt (U1IR<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 setup phase. It is not recommended to change these settings while the module is enabled.

25.4.4.1 USB LINK STATES

Three possible link states are described in the following subsections:

- Reset
- Idle and Suspend
- Resume Signaling
  - Driving Resume Signaling
  - Receiving Resume Signaling

25.4.4.1.1 Reset

As a host, software is required to drive Reset signaling. It may do this by setting USBRST (U1CON<4>). As per the *"Universal Serial Bus Specification"* (Revision 2.0), the host must drive the Reset for at least 50 ms. (This does not have to be continuous Reset signaling. For more information, refer to **Chapter 9. "USB Device Framework"** of the *"Universal Serial Bus Specification"* (Revision 2.0). Following Reset, the host must not initiate any downstream traffic for another 10 ms.

As a device, the USB module asserts 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.

25.4.4.1.2 Idle and Suspend

The Idle state of the USB is a constant J state. When the USB has been Idle for 3 ms, a device should go into Suspend state. During active operation, the USB host sends 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 bit (U1CON<0>).

As a USB device, hardware sets 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.

When a Suspend condition is detected, the software can place the USB hardware in a Suspend mode by setting the USUSPEND bit (U1PWRC<1>). The hardware Suspend mode gates the USB module's 48 MHz clock and places the USB transceiver in a low-power mode.

Additionally, the user can put the dsPIC33E/PIC24E device into Sleep mode while the link is suspended.

25.4.4.1.3 Driving Resume Signaling

If software wants to wake the USB from Suspend state, it should set the RESUME bit (U1CON<2>). This causes the hardware to generate the proper resume signaling (including finishing with a low-speed EOP if in host mode).

A USB device 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 device, or greater 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 "Resume"**, **11.4.4 "Generate Resume (GResume)"** and **11.9 "Suspend and Resume"** in the *"Universal Serial Bus Specification"* (Revision 2.0).

Writing RESUME automatically clears the special hardware Suspend (low-power) state.

If the part is acting as a USB host, software should, at minimum, set the SOFEN bit (U1CON<0>) after driving its resume signaling. Otherwise, the USB link would return right back to the Suspend state after 3 ms of inactivity. Also, software must not initiate any downstream traffic for 10 ms following the end of resume signaling.

25.4.4.1.4 Receiving Resume Signaling

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

A device 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 dsPIC33E/PIC24E is in Sleep mode will cause the ACTVIF (U1OTGIR<4>) interrupt to be set. This will cause wake-up from Sleep.

25.4.4.2 SRP SUPPORT

SRP support is not required by non-OTG applications. SRP may only be initiated at full-speed. Refer to the *"On-The-Go and Embedded Host Supplement to the USB Revision 2.0 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.

An OTG A-device or embedded host may repower the VBUS supply at any time to initiate a new session. An OTG B-device may also request that the OTG A-device repower the VBUS supply to initiate a new session. This is the purpose of the 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 end 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 can use the LSTATEIF bit (U1OTGIR<5>) and the 1 ms timer to identify 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 the VBUSDIS bit (U1OTGCON<0>).

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

After data line pulsing, the B-device should complete SRP signaling by pulsing the VBUS supply. This should be done in software by setting the VBUSCHG bit (U1OTGCON<1>).

When 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 the VBUSON bit (U1OTGCON<3>).

The B-device should not monitor the state of the VBUS supply while performing VBUS supply pulsing. Afterwards, if the B-device does detect that the VBUS supply has been restored (via the SESVDIF (U1OTGIR<3>) interrupt), it must reconnect to the USB link by pulling up D+. The A-device must complete the SRP by enabling VBUS and driving Reset signaling.

For more details, refer to the *"On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification"*.

25.4.4.3 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. 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 the DPPULUP bit (U1OTGCON<7>).

When the A-device detects the disconnect condition (via the URSTIF (U1IR<0>) interrupt), it may allow the B-device to take over as host. The A-device does this by signaling connect as a full-speed device. Software may accomplish this by disabling host operation, HOSTEN (U1CON<3>) = 0, and connecting as a device (USBEN = 1). If the A-device responds with resume signaling, the A-device will remain as host.

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

When the B-device has finished its role as host, it stops all bus activity and turns on its D+ pull-up resistor by disabling host operations (HOSTEN = 0) and reconnects as a device (USBEN = 1).

The A-device then detects a Suspend condition (Idle for 3 ms) and turns off its D+ pull-up resistor. Alternately, the A-device may also power-down the VBUS supply to end the session; otherwise, it continues to provide the VBUS supply throughout this process.

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

For more details of HNP, refer to the *"On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification"*.

25.4.4.4 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 the clock source for the SIE. The control registers are clocked at the same speed as the CPU.

The dsPIC33E/PIC24E device family features an auxiliary oscillator and PLL which can be configured to provide a 48 MHz clock required for USB module operation. Example 25-2 shows how the auxiliary PLL can be configured to provide a 48 MHz clock when a 8 MHz crystal is connected to the primary oscillator of the device.

Example 25-2:

```
/*
In this case, an 8 MHz crystal is connected to the primary oscillator of the
device. The Auxiliary PLL should be configured to use the primary oscillator
to generate the 48 MHz clock needed for the USB module. To do this, the
8 MHz clock is first divided by 2, then multiplied by 24, and then divided
by 2 again.
*/

ACLKCON3 = 0;
ACLKCON3bits.APLLPRE = 0x1;       // Divide PLL input by 2
ACLKCON3bits.APLLPOST = 0x6;      // Divide PLL output by 2
ACLKCON3bits.ASRCSEL = 1;         // Aux PLL source is primary oscillator
ACLKCON3bits.SELACLK = 1;         // Input to PLL divider is PLL output

ACLKDIV3 = 0x7;                   // PLL multiply by 24
ACLKCON3bits.ENAPLL = 1;          // Enable the Aux PLL
while(ACLKCON3bits.APLLCK != 1);  // Wait for it lock
```

For more details on oscillator configuration, refer to **Section 7. "Oscillator"** (DS70580) of "*dsPIC33E/PIC24E Family Reference Manual*".

25.5 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. Software must be able to respond to these interrupts in a timely manner.

25.5.1 Interrupt Control

Each interrupt source in the USB module has an interrupt flag bit and a corresponding enable bit. In addition, the read-only UERRIF bit (U1IR<1>) is a logical OR of all the enabled error flags. This bit can be used to check the USB module for events while in an Interrupt Service Routine (ISR).

25.5.2 USB Module Interrupt Request Generation

The USB module can generate interrupt requests from a variety of events. 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 (see Figure 25-12). The USB 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 enabled via the U1EIE register bits. In cases, where the USB module accesses unimplemented RAM, the UAE bit (INTCON3<6>) is set.

25.5.3 Interrupt Timing

Interrupts for transfers are generated at the end of the transfer. Figure 25-11 illustrates 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.

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

25.5.4 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' to the appropriate bit position to clear the interrupt. The USB Interrupt, USBIF (IFS1<25>), must be cleared before the end of the ISR.

Figure 25-11: Typical Events for USB Interrupts

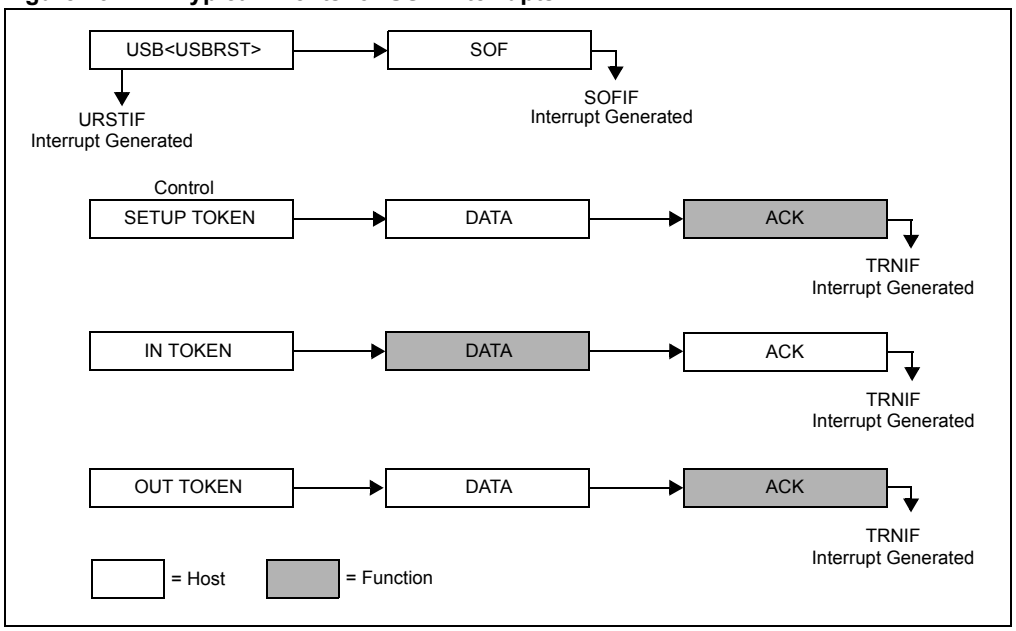
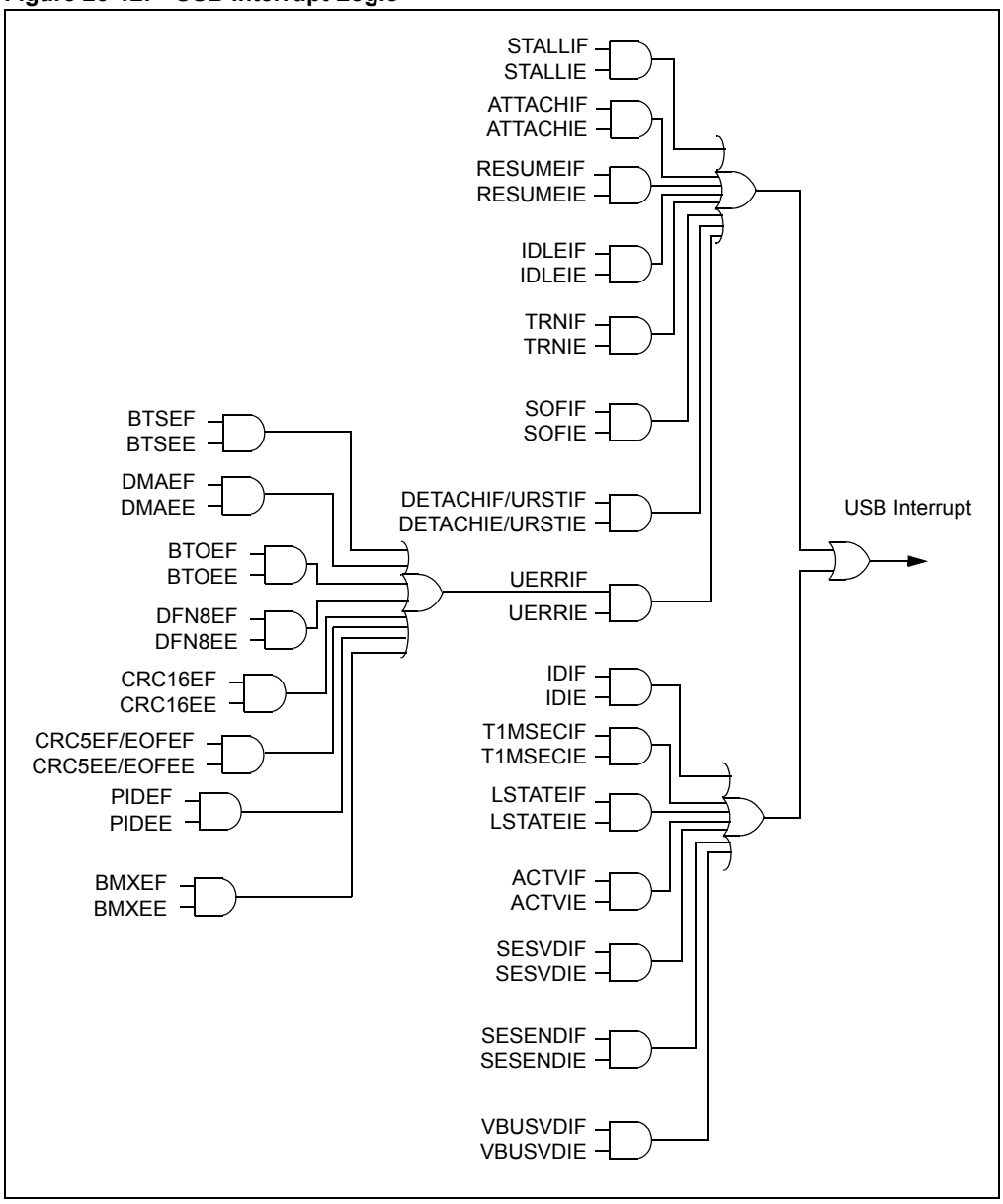


Figure 25-12: USB Interrupt Logic



25.6 OPERATION IN DEBUG AND POWER-SAVING MODES

25.6.1 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 violating USB protocol.

To further reduce power consumption, the USB module can be placed in Suspend mode. This can be done prior to placing the CPU in Sleep using the USUSPEND bit (U1PWRC<1>).

The USB activity interrupt can be enabled when the device enters Sleep mode. The activity interrupt can be used to wake the device from Sleep. It occurs when there is bus activity on the USB bus. The oscillator start-up time and PLL lock time must be taken into account when the activity interrupt is used to wake the device.

25.6.1.1 BUS ACTIVITY COINCIDENT WITH ENTERING SLEEP MODE

Software is unable to predict bus activity, therefore, even when software has determined that the USB link is in a state safe for entering Sleep, bus activity can still occur, potentially placing USB in a non-safe link state. The USLPGRD (U1PWRC<4>) and UACTPND (U1PWRC<7>) bits can be used to prevent this. Before entering the sensitive code region, software can set the USLPGRD bit to indicate the USB module that the device will be placed in Sleep mode. The UACTPND bit should be polled to ensure no interrupt is pending before attempting to enter Sleep.

25.6.2 Operation in Idle Mode

When the device enters Idle mode, the behavior of the USB module is determined by the USBSIDL bit (UxCNFG1<4>).

25.6.2.1 IDLE OPERATION WITH USBSIDL CLEARED

When the bit is clear, the clock to the CPU is gated off but the clock to the USB module is maintained when in Idle mode. The USB module can therefore continue operation while the CPU is Idle. When enabled USB interrupts are generated they will bring the CPU out of Idle.

25.6.2.2 IDLE OPERATION WITH USBSIDL SET

When the USBSIDL bit is set, the clock to the CPU and the clock to the USB module are both gated off. In this mode, the USB module does not continue normal operation and has lower power consumption. Any USB activity interrupt can be used to generate an interrupt to bring the CPU out of Idle. This requires the USB activity interrupt to be enabled.

25.6.3 Operation in Debug Modes

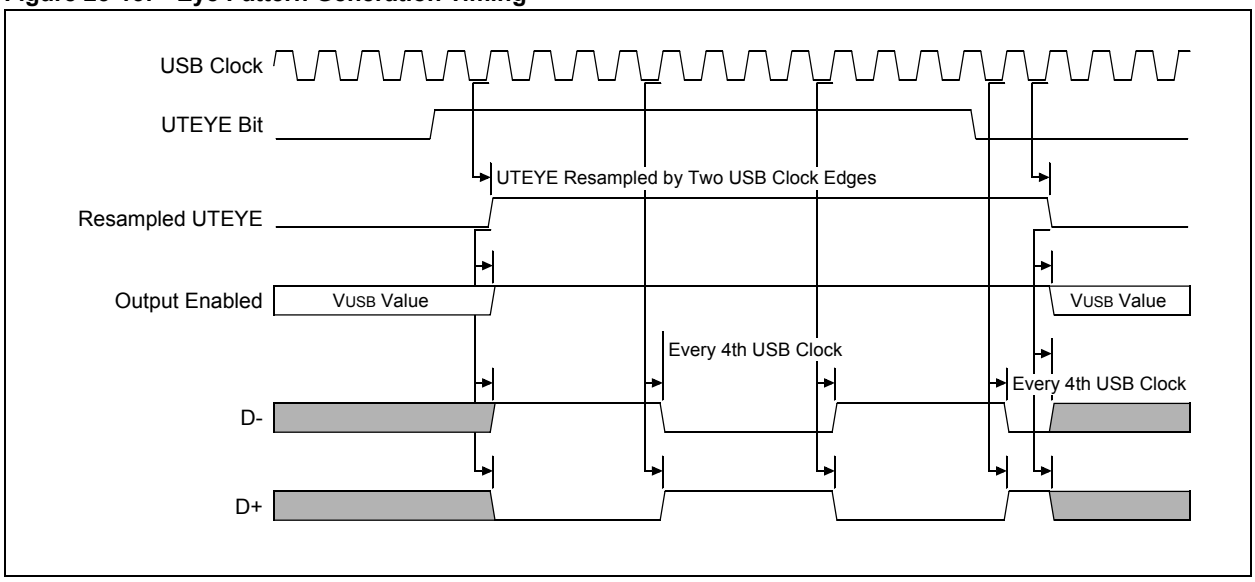
25.6.3.1 EYE PATTERN

To assist with USB hardware debugging and testing, an eye pattern test generator is incorporated into 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 should not be in Freeze mode.

Once the UTEYE bit is set, the module starts 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 25-13).

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.

Figure 25-13: Eye Pattern Generation Timing



25.7 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.

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

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

25.7.2 Power-on Reset (POR)

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

25.7.3 Watchdog Timer Reset (WDT)

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

25.8 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 dsPIC33E/PIC24E product 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 #
USB Embedded Host Stack	AN1140
USB Embedded Host Stack Programmer's Guide	AN1141
USB Mass Storage Class on an Embedded Host	AN1142
Using a USB Flash Drive with an Embedded Host	AN1145
USB HID Class on an Embedded Device	AN1163
USB CDC Class on an Embedded Device	AN1164
USB Generic Function on an Embedded Device	AN1166
USB Mass Storage Class on an Embedded Device	AN1169

Note: Please visit the Microchip web site (http://www.microchip.com) for additional Application Notes and code examples for the dsPIC33E/PIC24E family of devices.

25.9 REVISION HISTORY

Revision A (January 2009)

This is the initial released version of this document.

Revision B (October 2010)

This revision includes the following changes:

- Changed the document name from dsPIC33E Family Reference Manual to dsPIC33E/PIC24E Family Reference Manual
- Added a Note 2 to Figure 25-9.
- Additional minor updates to text and formatting have been incorporated throughout the document

NOTES:

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

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

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

Trademarks

Printed on recycled paper.

ISBN: 978-1-60932-620-3

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

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

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/04/10