OPB Universal Serial Bus 2.0 Device (v1.00a)

Introduction

The Xilinx Universal Serial Bus 2.0 High Speed Device with On-chip Peripheral Bus (OPB) enables USB connectivity to the user's design with a minimal amount of resources. This interface is suitable for USB-centric, high-performance designs, bridges and legacy port replacement operations.

Features

- Compliant with the USB 2.0 Specification.
- Supports High Speed and Full Speed.
- Has a 32-bit OPB Slave Interface.
- Has a ULPI[1] interface to external USB PHY. Supports positive and negative ULPI clocks.
- Has eight endpoints, including one control endpoint 0. Endpoints 1 - 7 may be bulk, interrupt, or isochronous. Endpoints are individually configurable.
- Uses Block RAM for endpoint buffers. Each endpoint has two ping-pong buffers.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex™-4, Virtex-5, Virtex-2P, Virtex-4XA, Spartan™-3, Spartan-3A, Spartan-3E, Spartan-3XA, Spartan-3EXA	
Version of core	opb_usb2_device	v1.00a
Resources Used		
Slices	1235	1235
LUTs	2340	2340
FFs	610	610
Block RAMs	4	4
Special Features	None	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	UCF	
Verification	VHDL Test bench	
Instantiation Template	VHDL Wrapper	
Reference Designs & application notes	None	
Additional Items	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE™ 9.1 or later	
Verification	ModelSim® SE/EE 6.1e or later	
Simulation	ModelSim SE/EE 6.1e or higher	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

1. USB Transceiver Macrocell Interface (UTMI) with a Low Pin Interface

Functional Description

The USB 2.0 protocol multiplexes many devices over a single, half-duplex, serial bus. The bus runs at 480 Mbps (High Speed) or at 12 Mbps (Full Speed) and is designed to be plug-and-play. The host always controls the bus and sends tokens to each device specifying the required action. Each device has an address on the USB 2.0 bus and has one or more endpoints that are sources or sinks of data. All devices have the system control endpoint (endpoint 0).

The OPB USB 2.0 Device has eight endpoints - one control endpoint (endpoint zero) and seven user endpoints.

Endpoint 0 of the USB 2.0 Device has different requirements than the seven user endpoints. Endpoint 0 handles control transactions only, which start with an 8-byte setup packet and are then followed by zero or more data packets. The setup packet is always stored in a dedicated location in the DPRAM at an address offset of 0x80. When a setup packet is received, the SETUP bit of the Status Register is set. Data packets are a maximum of 64 bytes. These data packets are stored in a single bidirectional data buffer set up by the configuration memory of Endpoint 0 located at the address offset 0x0 in the DPRAM. When a data packet is transmitted or received successfully, the Data Buffer Free and Data Buffer Ready bits of the Status Register are set respectively.

The seven user endpoints of the USB 2.0 Device can be configured as bulk, interrupt, or isochronous. In addition, endpoints can be configured as INPUT (to the host) or OUTPUT (from the host). Each of these endpoints has 2 ping-pong buffers of the same size for endpoint data. Data buffers for user endpoints are unidirectional, as configured by the Endpoint Configuration/Status register of the respective endpoint. The size of the buffers can be configured from 0 to 512 bytes for bulk and interrupt endpoints, and up to 1024 for isochronous endpoints. When the host wants to send data to an endpoint of the device, it sends an OUT token along with the address of the device and the endpoint number, followed by the data. To receive the data, the host sends an IN token with the device address and endpoint number, and waits for data from the device.

The OPB USB 2.0 High Speed Device core with the OPB and ULPI interfaces is shown in Figure 1 and described in the subsequent sections.

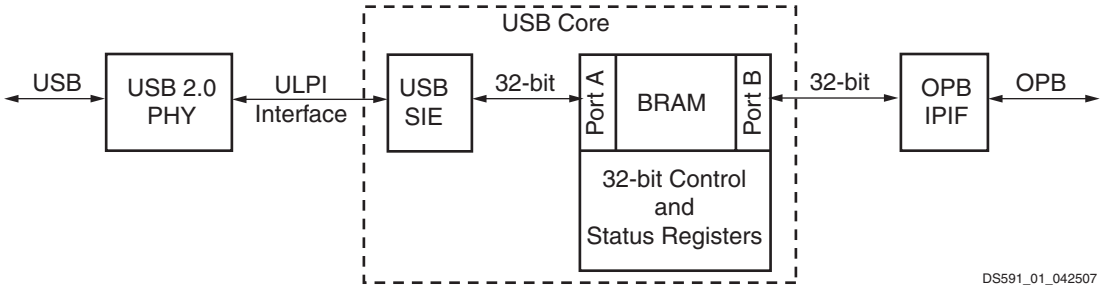

Figure 1: **OPB USB 2.0 Device with OPB and ULPI interfaces**

Control and Status Registers

The USB 2.0 Device includes a few 32-bit registers, which provide control and status information of the core, and are accessed from the OPB bus. An Interrupt Enable Register (IER) allows generation of a OPB interrupt based on specific Status Register bits.

USB 2.0 SIE

The USB 2.0 Serial Interface Engine (SIE) handles the serialization and de-serialization of USB traffic at the byte level and the multiplexing and demultiplexing of USB data to and from the endpoints of the core. The SIE also handles USB 2.0 state transitions, such as suspend, resume, and USB reset.

The SIE interfaces to the PHY using a ULPI interface that requires 12 pins. Data to the FPGA from the USB is received from the PHY, error checked and loaded into the appropriate area of the DPRAM. Data from the FPGA that is to be sent over the USB, is loaded from the DPRAM, protocol wrapped, then when the protocol allows, presented to the PHY, one byte at a time. Details of the ULPI and UTMI interfaces is beyond the scope of this document.

The status of the current USB transactions are signalled by the SIE to the status register. Certain conditions can be enabled through the IER to generate an interrupt on the OPB bus.

Control of the USB SIE comes from 4 sources:

1. The lower 64 bytes of the DPRAM contain the control and status locations for each endpoint.

2. The Control and Status Registers provide overall start and stop, status indication, enabling of interrupts via status register bits, address control on USB, current Start of Frame timing information, and endpoint Buffer Ready indication.

3. The logic of the USB SIE of the core is coded to reflect the requirements of Chapter 8 of the USB 2.0 Specification.

4. The USB is presented to the SIE over the ULPI PHY interface. The SIE natively implements the ULPI protocol.

Dual Port Block RAM (DPRAM)

The DPRAM is the data storage area between the USB SIE and the OPB bus interface. Port A of the DPRAM is used by the SIE and Port B is used by the OPB bus. Both ports are 32-bit wide.

The USB 2.0 Device uses 4 BRAMs implemented as 2K x 8 bits each, with dual asynchronous clock ports. Extra address lines are synthesized in the core so that the 4K byte BRAMs can be used if extra buffer space is required.

Data from the USB 2.0 Device is stored in the appropriate locations in the DPRAM by the SIE through Port A. The firmware or hardware being utilized by the user accesses the data through Port B over the OPB. Data to the USB 2.0 Device is loaded by the user through the OPB to Port B, into appropriate locations in the DPRAM. When the PC requests data from the device, the SIE accesses this data from Port A.

The DPRAM is seen by the SIE as eight endpoint FIFOs and a control register area that defines how the memory is arranged. Each FIFO is double buffered to help support the high throughput possible with USB 2.0. One buffer may be used for a current USB transaction, while the other buffer is available to the user application for processing. The storage areas are treated as FIFOs only from the point of view of the SIE. The firmware or hardware utilized by the user can access the storage as ordinary RAM over the OPB. DPRAM based registers are located in the lower 64 bytes of the DPRAM that control the layout of each FIFO of the endpoints in the DPRAM and also report the status of each FIFO buffer (ready, not ready, count).

OPB Interface

The OPB bus interface connects Port B of the DPRAM and the Control and Status Registers to the OPB bus. This is an OPB slave interface. Byte, half word, and word transfers are supported for the DPRAM interface, but only word transfers are supported for the register interface.

USB 2.0 PHY

The USB PHY can be any ULPI compliant PHY on the market. The primary job of the PHY is to manage the bit level serialization and de-serialization of USB 2.0 traffic. To do this, it must detect and recover the USB bus clock. The clock runs at 480 MHz, a speed that is too fast for practical implementation on the FPGA. Because 480 MHz is also too fast for the USB SIE clock, the PHY interfaces to the SIE on a byte serial basis and generates a 60 MHz clock which runs the SIE side of the USB 2.0 Device.

Test Mode Support

The USB 2.0 Device provides test mode support to facilitate compliance testing.

Three test modes are supported:

- **Test Mode J**: The core transmits a continuous chirp J and remains in this state until the time when it is reset.
- **Test Mode K**: The core transmits a continuous chirp K and remains in this state until the time when it is reset.
- **Test Mode NAK**: The core searches for any IN token with a valid crc5. If crc5 is valid, the core sends a NAK, otherwise it waits for the next valid IN token. The core remains in this state until it is reset.

Test mode packet: As specified by the USB 2.0 Specification, the core transmits a test packet which is composed of a predefined sequence of bytes and is used for analog testing of the USB in the high speed mode. The packet data is loaded into a predefined sequence of locations in the DPRAM. This routine repeats continuously until the core is reset.

Clocking and Reset

Clocking

The clock to the USB 2.0 Device runs at 480 MHz when operating at high speed. Because this frequency is too high for the SIE clock, as well as for the FPGA, the PHY interfaces with the SIE and generates a 60 MHz clock.

The USB 2.0 Device uses two clocks:

- **OPB CLK**: The OPB Bus interface, Port B of the DPRAM and the OPB registers use this clock. The minimum OPB CLK frequency needed to achieve the maximum performance of 480 MHz is 60 MHz.
- **ULPI CLK**: The SIE interface and Port A of the DPRAM operate using this clock. The ULPI clock is generated by the PHY and has a fixed frequency of 60 MHz.

Reset

The OPB USB 2.0 Device core is reset using the OPB_Rst signal that resets all devices that are connected to the OPB Bus in the processor system. The minimum duration of the reset pulse required to reset the logic on the SIE side is 1 ULPI clock period.

Interrupts

The USB 2.0 Device has a single interrupt line (usb_irpt) to indicate an interrupt. Interrupts are indicated by asserting the usb_irpt signal (transition of the usb_irpt from a logic '0' to a logic '1').

The Interrupt Enable Register allows specific bits of the Status Register to generate interrupts. The Master Enable bit of this register allows all interrupts to be disabled simultaneously. The interrupt condition is cleared when the corresponding bit of the Status Register is cleared by writing a ‘1’ to it. During power on, the usb_irpt signal is driven low.

The following two conditions cause the usb_irpt signal to be asserted:

- If a bit in the SR is '1' and the corresponding bit in the IER is '1'.
- Changing an IER bit from a '0' to '1'; when the corresponding bit in the ISR is already '1'.

Two conditions cause the usb_irpt signal to be de-asserted:

- Clearing a bit in the SR that is '1' by writing a '1' to it; provided the corresponding bit in the IER is '1'.
- Changing an IER bit from '1' to '0'; when the corresponding bit in the SR is '1'.

When both de-assertion and assertion conditions occur simultaneously, the usb_irpt signal is de-asserted first, then is reasserted if the assertion condition remains true.

I/O Signals

Description of the I/O signals for the OPB USB 2.0 Device is given in Table 1.

Table 1: **OPB USB 2.0 Device External I/O**

Signal Name	Signal Direction	Default Value	Description
ULPI_Clock	Input	N/A	USB Clock
OPB_Clk	Input	N/A	OPB bus clock
OPB_Rst	Input	N/A	OPB bus reset (Active High)
OPB_ABus(0:31)	Input	N/A	OPB address bus
OPB_BE(0:3)	Input	N/A	OPB Byte Enables
OPB_Data_In(0:31)	Input	N/A	OPB write data bus
OPB_RNW	Input	N/A	OPB read not write
OPB_Select	Input	N/A	Active High Chip Select
OPB_Data_Out(0:31)	Output	X"00000000"	Slave read data bus
OPB_XferAck	Output	’0’	Slave data acknowledge
OPB_seqAddr	IN	N/A	Unconnected to USB internal modules
OPB_ErrAck	OUT	’0’	Unconnected to USB internal modules
OPB_Retry	OUT	’0’	Unconnected to USB internal modules
OPB_TimeOutSup	OUT	’0’	Unconnected to USB internal modules
ULPI_Dir	Input	N/A	Direction of data flow between host and device

Table 1: **OPB USB 2.0 Device External I/O *(Contd)***

Signal Name	Signal Direction	Default Value	Description
ULPI_Next	Input	N/A	Indicator of when the PHY is ready for the next bit
ULPI_Stop	Output	’0’	Indicator that transmission of last byte is complete
ULPI_Reset	Output	’0’	Active High reset to the PHY
ULPI_Data_I(7:0)	Input	N/A	Input data to the core from the host
ULPI_Data_O(7:0)	Output	X"00"	Output data from the core to the host
ULPI_Data_T	Output	’0’	ULPI_Data is a 3-state port, with ULPI_Data_I as the IN port, ULPI_Data_O as the OUT port and ULPI_Data_T as the tristate output.
usb_irpt	Output	’0’	Active High Interrupt Line
vbus_detect	Output	’0’	Indicator that a valid V_{BUS} has been detected
show_currentspeed	Output	’0’	USB 2.0 Device current speed indicator. ’0’ indicates FULL Speed and ’1’ indicates HIGH Speed.
running	Output	’0’	Indicator that the USB 2.0 Device is running. When ’0’, it indicates that the SIE is reset and will not respond to USB traffic.
suspended	Output	’0’	USB 2.0 Device suspend indicator
disconnected	Output	’0’	USB cable disconnected indicator
configured	Output	’0’	Used for test modes for USB 2.0 certification
spare1	Output	’0’	Used for test modes for USB 2.0 certification
spare2	Output	’0’	Used for test modes for USB 2.0 certification

Design Parameters

N/A

Parameter - Port Dependencies

N/A

Register Descriptions

The memory map for the USB 2.0 Device, shown in Table 2, includes endpoint configuration space (offset 0x0000), setup packet storage space (offset 0x0080), RAM for endpoint buffers (offset 0x0088),

and register space for the Control and Status Registers (offset 0x4000). Table 3 lists the mapping for endpoint configuration space. All offsets are byte offsets.

Table 2: **USB 2.0 Device Address Map**

Address Offset	Memory/Register Space
0x0000	Endpoint Configuration Registers
0x0080	Setup Packet Storage Word 0
0x0084	Setup Packet Storage Word 1
0x0088	RAM for endpoint buffers
0x4000	USB Address Register
0x4004	Control Register
0x4008	Status Register
0x400C	Frame Number Register
0x4010	Interrupt Enable Register
0x4014	Buffer Ready Register
0x4018	Test Mode Register

Endpoint Configuration/Status Registers

The Endpoint Configuration and Status register control the operational characteristics of each endpoint and reports its current condition.

The total endpoint configuration register space is divided between the eight endpoints of the USB 2.0 Device as shown in Table 3.

Table 3: **Endpoint Configuration Registers**

Address Offset	Memory/Register Space
0x00	Endpoint 0
0x10	Endpoint 1
0x20	Endpoint 2
0x30	Endpoint 3
0x40	Endpoint 4
0x50	Endpoint 5
0x60	Endpoint 6
0x70	Endpoint 7

Each endpoint has four 32-bit words that describe the behavior of the endpoint. These words are located sequentially and arranged by endpoint number as shown in Table 4.

Table 4: **Endpoint Configuration Words**

Address Offset	Memory/Register Space
0x00	Endpoint configuration/status register
0x04	Reserved
0x08	Buffer 0 count: 0 to 1024
0x0C	Buffer 1 count: 0 to 1024

The bit description for the Endpoint Configuration/Status Registers is given in Table 5. All the bits of this register can be modified by the firmware. Under normal operation, some of the bits are modified by the USB SIE itself, and only their initial values need to be set by the firmware.

Table 5: **Endpoint Configuration/Status Register**

Bit(s)	Name	Access	Default Value	Description
0	EP_VALID	R/W	0	Master enable bit; '1'=Enable, '0'=Disable
1	EP_STALL	R/W	0	When '0', the endpoint accepts IN's and OUT's; When '1', the endpoint responds to the host only with a STALL.
2	EP_OUT_IN	R/W	0	IN/OUT is with respect to the host. When '0', the core receives data from the host (OUT from host); When '1', the core sends data to the host (IN to host).
3	EP_ISO	R/W	0	When '1', the endpoint behaves as an isochronous endpoint. ISO endpoints do not send or expect ACK or NAK.
4	EP_DATA_TOGGLE	R/W	0	Used as a weak form of synchronization; When '0', the next DATA packet must be a DATA0 packet; When '1', the next packet must be a DATA1 packet. Can be explicitly set in response to a driver command.
5	EP_BUFFER_SELECT	R/W	0	Implements ping-pong buffers; When '0', Buffer 0 is used; When '1', Buffer 1 is used; This bit is toggled by the SIE from buffer to buffer.
6-16	EP_PACKET_SIZE	R/W	0	Endpoint packet size
17-18	Reserved	R/W	0	Reserved for future use
19-31	EP_BASE	R/W	0	Base offset of the buffers in the DPRAM

The VALID, STALL, OUT_IN, and ISO bits are set by the firmware to define how the endpoint operates. For example, to set up the endpoint to receive bulk OUT's from the PC, set EP_VALID='1', EP_OUT_IN='0', EP_STALL='0', and EP_ISO='0'.

The EP_DATA_TOGGLE and EP_BUFFER_SELECT bits are modified by the USB SIE in response to USB operations and only their initial values are set by the firmware.

Buffer Count Register

The Buffer 0 Count and Buffer 1 Count registers, shown in Table 6, indicate the amount of data in the respective buffers. If the endpoint is an OUT endpoint, then the SIE sets the value of this register at the end of a successful reception from the host. If the endpoint is an IN endpoint, then the firmware sets the value of this register before transmission.

These registers are 32-bit wide and have R/W access.

Table 6: **Buffer Count Register**

Bit(s)	Name	Access	Default Value	Description
0-20	Reserved	R/W	0	Reserved for future use
21-31	(In/Out)_Pkt_Count(10:0)	R/W	0	Packet count in the buffer

USB Address Register

The USB Address register, shown in Table 7, contains the host-assigned USB address of the device. There are 128 possible USB devices on the USB. Therefore, the register takes values from 0 to 127. The lower seven bits of the register (6:0) are used to set the address. An address of 0 indicates that the device is un-enumerated. Address 0 is the default address of all USB devices at plug-in time and the address value on hardware reset.

This register is 32-bit wide and has R/W access

Table 7: **USB Address Register**

Bit(s)	Name	Access	Default Value	Description
0-24	Reserved	R/o	0	Reserved for future use
25-31	USB Address	R/W	0	Indicates the USB address of the device.

Control Register

As as shown in Table 8, only bit 31 of this register is used. This bit indicates SIE operation. When clear, the USB SIE is paused and will not respond to any USB activity. When set, the SIE operates normally.

Table 8: **Control Register**

Bit(s)	Name	Access	Default Value	Description
0	MASTER_READY	R/W	0	When '0', the USB SIE is paused and does not respond to any USB activity. When '1', the SIE operates normally.
1-31	Reserved	R/o	0	Reserved for future use

Status Register

The Status Register (SR), shown in Table 9, reports status on the operation of the USB 2.0 Device. Bits of this register get cleared as soon as they are read.

Table 9: **Status Register**

Bit(s)	Name	Access	Default Value	Description
0-7	Reserved	R/o	0	Reserved for future use.
8	USB Reset	R/o	0	Active high reset signal from the core to the host; It remains set for up to 3 ms.
9	USB Suspend	R/o	0	Active high suspend signal; It remains set as long as the core is suspended.
10	USB Disconnect	R/o	0	When '1', indicates that the USB cable is unplugged; When '0', indicates that the USB cable is connected.
11	Data Buffer Ready	R/o	0	When '1', indicates that endpoint 0 packet has been received.
12	Data Buffer Free	R/o	0	When '1', indicates that endpoint 0 packet has been transmitted.
13	SETUP	R/o	0	When '1', indicates that endpoint 0 Setup packet has been received.
14	SOF	R/o	0	When '1', indicates that Start of Frame packet has been received; Start of Frames happen every 125 us in the High Speed mode.
15	High Speed	R/o	0	When '1', indicates that the USB 2.0 Device is running at High Speed; When '0', indicates that the USB 2.0 Device is running at Full Speed.
16	EP-7 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 7 is complete.
17	EP-6 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 6 is complete.
18	EP-5 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 5 is complete.
19	EP-4 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 4 is complete.
20	EP-3 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 3 is complete.
21	EP-2 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 2 is complete.
22	EP-1 2nd Buf Comp	R/o	0	When '1', indicates that the 2nd buffer of endpoint 1 is complete.
23	Not Used	R/o		

Table 9: **Status Register *(Contd)***

Bit(s)	Name	Access	Default Value	Description
24	EP-7 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 7 is complete.
25	EP-6 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 6 is complete.
26	EP-5 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 5 is complete.
27	EP-4 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 4 is complete.
28	EP-3 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 3 is complete.
29	EP-2 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 2 is complete.
30	EP-1 1st Buf Comp	R/o	0	When '1', indicates that the 1st buffer of endpoint 1 is complete.
31	EP-0 Buf Comp	R/o	0	When '1', that the single buffer of endpoint 0 is complete; This buffer is bidirectional.

Frame Number Register

The Frame Number Register (FNR), shown in Table 10, is composed of two fields - Frame and Microframe. Frames are sent once every 1 ms and denote the beginning of a USB frame. All host scheduling starts at the start of Frame Time. The Microframe field is the result of additional Start of Frame tokens, sent once every 125 us. When the USB is operated in the High Speed mode, this can generate a potentially high rate of interrupts; hence, the interrupt enable of Start of Frame should be used with caution.

Frame count values are of 11 bits and Microframe count values are of 3 bits.

Table 10: **Frame Number Register**

Bit(s)	Name	Access	Default Value	Description
0-17	Reserved	R/o	0	Reserved for future use
18-28	Frame number(10:0)	R/o	0	Frame numbers - 0 to 2047
29-31	Microframe number(2:0)	R/o	0	Microframe numbers - 0 to 7

Interrupt Enable Register

The Interrupt Enable Register (IER), shown in Table 11, allows specific bits of the Status Register to generate interrupts. The Master Enable bit of this register allows all interrupts to be disabled simultaneously. The interrupt condition is cleared when the corresponding bit of the Status Register is cleared.

A specific bit of the IER may be cleared to prevent a long duration condition, such as USB Reset, from continuously generating an interrupt.

Table 11: **Interrupt Enable Register**

Bit(s)	Name	Access	Default Value	Description
0	Master Enable	R/W	0	When '1', enables setting of all other interrupts
1-7	Reserved	R/W	0	Reserved for future use
8	USB Reset	R/W	0	When '1', enables USB resets to generate an Interrupt Request (IRQ)
9	USB Suspend	R/W	0	When '1', enables USB Suspends to generate an IRQ
10	USB Disconnect	R/W	0	When '1', enables USB Disconnects to generate an IRQ
11	FIFO Buf Rdy	R/W	0	When '1', enables a received data packet on endpoint 0 to generate an IRQ
12	FIFO Buf Free	R/W	0	When '1', enables a successfully transmitted data packet on endpoint 0 to generate an IRQ
13	Setup Packet	R/W	0	When '1', enables a received setup packet to generate an IRQ
14	SOF Packet	R/W	0	When '1', enables a Start of Frame to generate an IRQ
15	High Speed	R/W	0	When '1', enables an IRQ when the core operates at High Speed
16	EP-7 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 7 is complete
17	EP-6 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 6 is complete
18	EP-5 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 5 is complete
19	EP-4 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 4 is complete
20	EP-3 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 3 is complete
21	EP-2 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 2 is complete
22	EP-1 2nd Buf Comp	R/W	0	When '1', enables an IRQ when 2nd buffer of Endpoint 1 is complete
23	Not Used	R/W	0	
24	EP-7 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 7 is complete
25	EP-6 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 6 is complete

Table 11: **Interrupt Enable Register *(Contd)***

Bit(s)	Name	Access	Default Value	Description
26	EP-5 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 5 is complete
27	EP-4 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 4 is complete
28	EP-3 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 3 is complete
29	EP-2 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 2 is complete
30	EP-1 1st Buf Comp	R/W	0	When '1', enables an IRQ when 1st buffer of Endpoint 1 is complete
31	EP-0 Buf Comp	R/W	0	When '1', enables an IRQ when the single bidirectional buffer of Endpoint 0 is complete

Buffer Ready Register

The Buffer Ready Register (BFR) has a buffer ready bit corresponding to each buffer of each endpoint, as shown in Table 12. The firmware sets each bit when that buffer is ready for either USB IN or USB OUT traffic. Until that bit is set, an attempted IN or OUT to/from the buffer will result in a NAK to the host. The ability of the buffer to handle an IN or OUT is determined by the EP_OUT_IN bit in the corresponding endpoint's Configuration/Status register. It should be noted that as per the USB 2.0 Specification, endpoint 0 has only one buffer that handles IN or OUT.

Table 12: **Buffer Ready Register**

Bit(s)	Name	Access	Default Value	Description
0-15	Reserved	R/o	0	Reserved for future use
16	EP-7 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 7 is available for IN or OUT
17	EP-6 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 6 is available for IN or OUT
18	EP-5 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 5 is available for IN or OUT
19	EP-4 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 4 is available for IN or OUT
20	EP-3 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 3 is available for IN or OUT
21	EP-2 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 2 is available for IN or OUT
22	EP-1 2nd Buffer Ready	R/W	0	When '1', the 2nd buffer of endpoint 1 is available for IN or OUT
23	Not Used	R/W		Endpoint 0 has only one buffer

Table 12: **Buffer Ready Register *(Contd)***

Bit(s)	Name	Access	Default Value	Description
24	EP-7 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 7 is available for IN or OUT
25	EP-6 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 6 is available for IN or OUT
26	EP-5 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 5 is available for IN or OUT
27	EP-4 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 4 is available for IN or OUT
28	EP-3 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 3 is available for IN or OUT
29	EP-2 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 2 is available for IN or OUT
30	EP-1 1st Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 1 is available for IN or OUT
31	EP-0 Buffer Ready	R/W	0	When '1', the 1st buffer of endpoint 0 is available for IN or OUT

Test Mode Register

The Test Mode Register (TMR) shown in Table 13 defines the different test modes in which the USB 2.0 Device operates (see Test Mode Support for more details). The USB Implementor's Forum, the organization that controls USB logo certification, requires all USB 2.0 devices that operate at High Speed support these test modes.

Table 13: **Test Mode Register**

Bit(s)	Name	Access	Default Value	Description
0-28	Reserved	R/o	0	Reserved for future use
29-31	Test Mode(2:0)	R/W	0	Value defines the test mode. 0 - Normal Mode 1 - Test Mode J 2 - Test Mode K 3 - Test Mode NAK 4 - Test Mode Packet

Programming the Core

This section describes how to program the USB 2.0 Device for various operations. For applications that use the Xilinx driver, users are expected to change the Vendor ID before use of the OPB USB 2.0 Device.

Initialization

Configuring an Endpoint

Program the individual Endpoint Configuration/Status Registers to configure the respective endpoints:

- A specific endpoint can be enabled by writing a '1' to the EP_VALID bit of the endpoint's configuration/status register.
- The direction of an endpoint can be set to IN by writing '1', or to OUT by writing '0' to the EP_OUT_IN bit of the register.
- The endpoint can be configured as an isochronous endpoint by writing a '1' , or as a bulk endpoint by writing '0' to the EP_ISO bit of the register.
- The packet size for the endpoint can be set by writing to the EP_PACKET_SIZE bits of the register.
- The base offset of the endpoint buffers in the DPRAM can be set by writing to the EP_BASE bits of the register.

Operating in the Interrupt Mode

If desired, configure the device to operate in the interrupt mode after enabling the desired interrupts in the Interrupt Enable Register.

- Interrupts for USB Reset, Suspend and Disconnect can be enabled by writing to those specific bits of the IER.
- To generate an interrupt when the core in operating in High Speed, the High Speed bit of the IER should be set.
- To generate an interrupt when the core in operating in High Speed, the High Speed bit of the IER should be set.
- To generate an interrupt when a startup packet or SOF packet is received, those bits of the IER must be set.
- For endpoint 0, the Fifo Buffer Ready and Fifo Buffer Free bits of the IER must be set to generate interrupts when packets are received or transmitted respectively.
- For all other endpoints, the respective Buffer Complete bit of the IER must be set to generate interrupts when that endpoint buffer is complete.

Enabling the USB 2.0 Device

Set the MASTER_READY bit of the Control Register to '1' to enable the configured endpoints for operation.

Setting the USB 2.0 Device Address

Set the device to the unenumerated state by writing an address of 0 to the USB Address Register

Handling a Control Packet

1. Poll the SETUP bit of the Status Register for '1' to detect the reception of the setup packet.
2. Read the setup packet from Endpoint 0's Buffer location in the DPRAM, which will cause the SETUP bit of the Status Register to be cleared.
3. Process the received Chapter 9 (as detailed in the USB 2.0 Specification) command and prepare a buffer for response that must be sent for the subsequent IN packet.

Handling Bulk/Isochronous IN Transactions

For a Bulk or Isochronous IN transaction, the following steps are performed:

1. Write the IN packet into the selected endpoint buffer location in the DPRAM.
2. Write the packet count into the specific endpoint buffer's Buffer Count Register.

3. Set the Buffer Ready bit for the selected endpoint buffer in the Buffer Ready Register.
4. Poll the Buffer Ready bit of the selected endpoint buffer in the Buffer Ready Register (this bit must be cleared) to ensure that the transmitted data has been received by the host and the buffer is available for the next write.

Handling Bulk/Isochronous OUT Transactions

When the host sends an OUT packet to an endpoint buffer on the device while the buffer is in use, the device core sends a NAK to the host. Once the buffer is free, the packet is written into the buffer and an ACK is automatically sent to the host.

On reception of the OUT packet, the following steps are performed:

1. Poll the Buffer Complete bit of the selected endpoint buffer in the Status Register to detect the reception of a packet.
2. Check that the received packet count matches with the specified packet count in the Buffer Count Register.
3. Read the OUT packet from the selected endpoint buffer location in the DPRAM.
4. Set the Buffer Ready bits of the selected endpoint buffer to prepare it for the next transaction.

Test Mode Operation

- The default mode of operation for the USB 2.0 Device is the normal mode, for which all the bits of the Test Mode Register are set to ’0’.
- To put the core in Test Mode operation, program the bits [29:31] of the Test Mode Register for different test modes:
  - To put the core in the Test mode J, program TMR[29:31]="001". In this mode, Chirp J sequences must be seen on the bus.
  - To put the core in the Test mode K, program TMR[29:31]="010". In this mode, Chirp K sequences must be seen on the bus.
  - To put the core in the Test mode NAK, program TMR[29:31]="011". In this mode, NAK must be seen on the bus.
  - To put the core in the Test mode packet, program TMR[29:31]="100". In this mode, the test packet specified by the USB 2.0 Specification must be seen on the bus.

Timing Diagrams

The USB 2.0 Device supports two modes of transfer:

- Single read
- Single write

Single Read Transaction

When the read transfer is enabled (OPB_select = '1' and OPB_RNW = '1'), the controller samples the address on the OPB_ABus and returns the corresponding read data on the OPB_Data_Out. Read data is returned on a successive clock rising edge. OPB_xferAck is asserted when the data is ready on the OPB_Data_Out pins. The address indicated by the OPB_ABus is assumed to be valid for the entire duration during which OPB_select is asserted.

OPB_xferAck is asserted for all read transactions. Successive single read operations require that the OPB_select be deasserted and reasserted. The timing diagram for a single read transaction is shown in Figure 2.

For single read transactions:

- Reads from address locations defined as reserved will return all 0s on the OPB_Data_Out bus.
- Reads from write only address locations will return all 0's on the OPB_Data_Out bus.

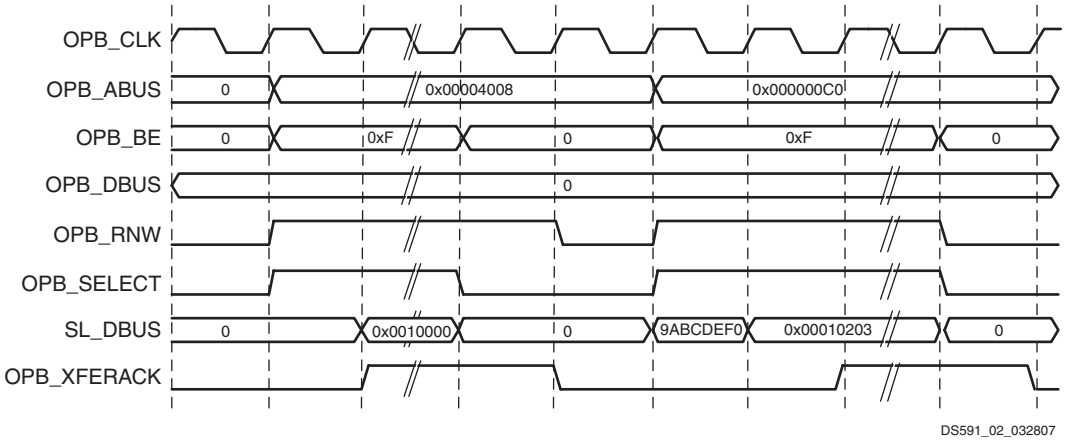

Figure 2: **OPB Single Read Transaction Timing Diagram**

Single Writes

When the write transfer is enabled (OPB_select = '1' and OPB_RNW = '0'), the controller samples both the address on the OPB_ABus and the data on the OPB_Data_In. The write operation is completed on a successive clock rising edge. The completion of the write operation is indicated by the assertion of the OPB_xferAck. The address indicated by the OPB_ABus and the data indicated by the OPB_Data_In is assumed to be valid for the entire duration during which OPB_select is asserted.

OPB_xferAck is asserted for all write transactions. Successive write operations require that OPB_select be deasserted and reasserted. The timing diagram for a single read transaction is shown in Figure 3.

For single write transactions:

- Writes to address locations and bits defined as reserved will not have any effect.

- Writes to address locations defined as read only will not have any effect.

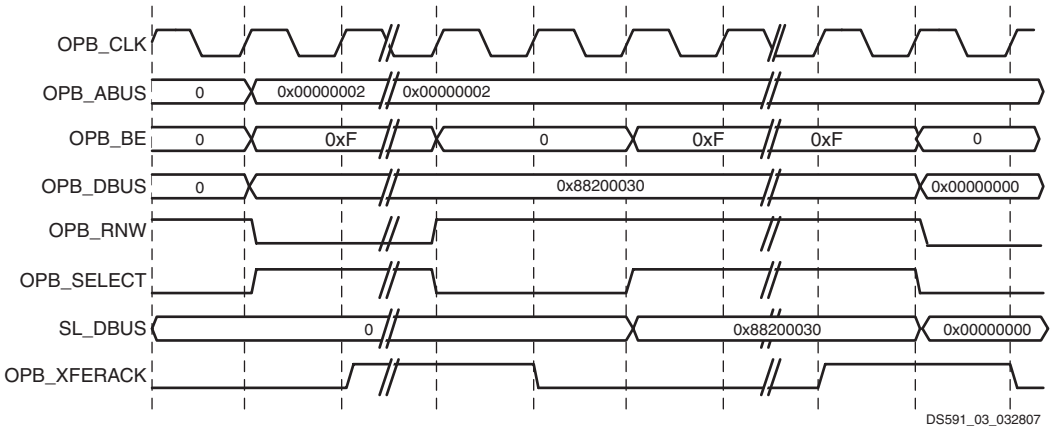

Figure 3: **OPB Single Write Transaction Timing Diagram**

Design Constraints

Location Constraints

The ULPI pins of the core should be connected to the corresponding pins of the ULPI PHY.

Timing Constraints

The core has 2 different clock domains: OPB_CLK and ULPI_Clock. A timing ignore constraint should be added to isolate these two clock domains. The constraints given below can be used with the USB 2.0 Device.

PERIOD Constraints for Clock Nets

ULPI_CLK

The ULPI clock input is generated through the ULPI PHY and has a fixed frequency of 60 MHz.

```
# Set the ULPI_CLK constraints
NET "ULPI_CLK" TNM_NET = "ULPI_CLK";
TIMESPEC "TS_ULPI_CLK" = PERIOD "ULPI_CLK" 16667 ps HIGH 50%;
```

OPB_CLK

The clock provided to OPB_CLK must be constrained for a clock frequency of 60 MHz - 100 MHz.

```
# Set the OPB_CLK constraints; This can be relaxed based on the actual frequency
NET "OPB_CLK" TNM_NET = "OPB_CLK";
TIMESPEC "TS_OPB_CLK" = PERIOD "OPB_CLK" 10 ns HIGH 50%;
```

Design Implementation

Target Technology

The USB 2.0 device can be implemented in Spartan 3, Spartan-3 XA, Spartan 3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Virtex-II Pro, Virtex-4, and Virtex-4 XA devices. The device used must have the following attributes:

- Large enough to accommodate the core.
- Contains a sufficient number of IOBs.

The intended technology for the following section is the Virtex-4 FPGA.

Device Utilization and Performance Benchmarks

Because the USB 2.0 Device core does not have any parameters, the core has a fixed count for the device resources as shown in Table 14.

Table 14: **Performance and Resource Utilization Benchmarks**

Device Resources			f_{MAX} (MHz)
Slices	Slice Flip- Flops	4-input LUTs	OPB f_{MAX}
1171	561	2221	79.334

Reference Document(s)

Universal Serial Bus Specification, Revision 2.0

UTMI++ Low Pin Interface (ULPI) Specification, Revision 1.1

Revision History

Date	Version	Revision
5/10/2007	1.0	Initial Xilinx release.