

## Introduction

The Xilinx Universal Serial Bus 2.0 High Speed Device with Advance Micro controller Bus Architecture Advanced eXtensible Interface (AXI) 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

- AXI Interface based on the AXI4 specification
- Supports burst lengths of 1-256 beats with INCR type transfers
- Compliant with the USB 2.0 Specification
- Supports High Speed and Full Speed
- Supports Universal Transceiver Macrocell Interface (UTMI) + Low Pin Interface (ULPI) to external USB PHY
- Parameterized ULPI PHY Reset
- Resume and Reset detection in low-power mode
- Supports Resuming of Host from Low-power mode with Remote Wake-up signalling.
- Supports up to eight endpoints, including one control endpoint 0. Endpoints 1 - 7 may be bulk, interrupt, or isochronous. Endpoints are individually configurable.
- Supports two ping-pong buffers for each endpoint except for endpoint 0
- USB Error detection, updates Error Count and generates Error interrupt
- DMA mode to increase throughput during the data transfers

LogiCORE™ IP Facts Table	
<b>Core Specifics</b>	
Supported Device Family <sup>(1)</sup>	Spartan®-6, Virtex®-6
Supported User Interfaces	AXI4, ULPI
<b>Resources Used</b>	
LUTs	See <a href="#">Table 25</a> , and <a href="#">Table 26</a> .
DSP Slices	
Block RAMs	
FFs	
<b>Provided with Core</b>	
Documentation	Product Specification
Design Files	VHDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	UCF
Simulation Model	ModelSim/NCprotect encrypted
<b>Tested Design Tools</b>	
Design Entry Tools	ISE® 12.4
Simulation	MentorGraphics ModelSim 6.5c and above
Synthesis Tools	XST
<b>Support</b>	
Provided by Xilinx, Inc.	

### Notes:

1. For a complete listing of supported devices, see the release notes for this core.

## Functional Description

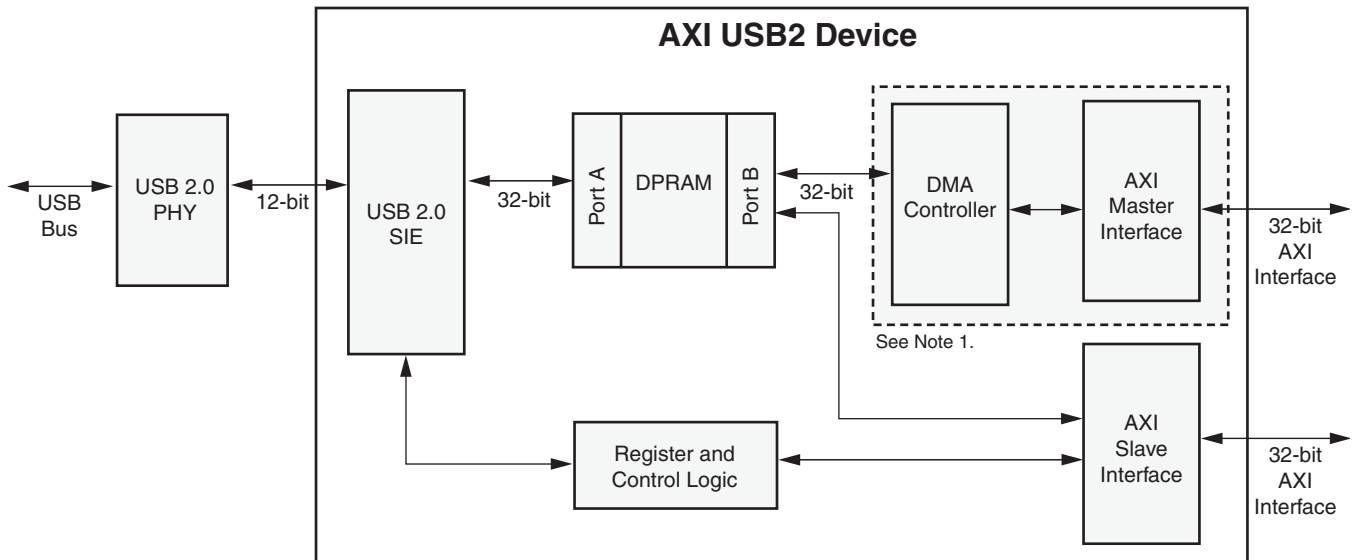
The USB 2.0 protocol multiplexes many devices over a single, half-duplex, serial bus. The bus runs at 480 Mbps (High Speed) or at 12 Mbps (Full Speed) and is designed to be plug-and-play. The host always controls the bus and sends tokens to each device specifying the required action. Each device has 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 AXI USB2 Device has eight endpoints - one control endpoint (endpoint 0) 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 then followed by zero or more data packets. The setup packet is always stored in a dedicated location in the Dual Port Random Access Memory (DPRAM) at an address offset of 0x80. When a setup packet is received, the SETUP bit of the [Interrupt Status Register \(ISR\)](#) 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 [Interrupt Status Register \(ISR\)](#) 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. The user endpoints data buffers are unidirectional, and are configured by the Endpoint Configuration and Status register of the respective endpoint. The size of the buffers can be configured from 0 to 512 bytes for bulk, 64 bytes for interrupt, and up to 1024 bytes for isochronous endpoints.

The AXI USB 2.0 High Speed Device core with the AXI and ULPI interfaces is shown in [Figure 1](#) and described in the subsequent sections.



Note 1:  
The DMA Controller and the AXI Master Interface Module will be included if the parameter C\_INCLUDE\_DMA=1.

DS785\_01

Figure 1: AXI USB2 Device with AXI and ULPI Interfaces

When the host wants to send data to an endpoint of the device, it sends a token which consists of an OUT PID along with the address of the device and the endpoint number, followed by the data. Device uses handshake packets to report the status of the data transaction if, and only if, the token and data are received without any USB errors, such

as Bit Stuff, PID, and CRC errors. Handshake packets indicate successful reception of data, flow control, and halt conditions.

To receive the data, the host sends a token which consists of an IN PID along with the device address and endpoint number, then waits for data from the device. Device responds with the requested data to the host if and only if the token is received without any USB errors and waits for handshake packet.

## Register and Control Logic

The AXI USB 2 Device includes a few 32-bit registers, which provide control and status information of the core, and are accessed from the AXI. An [Interrupt Enable Register \(IER\)](#) allows the generation of an interrupt based on specific [Interrupt Status Register \(ISR\)](#) bits. For more information on the registers refer the section [Register Description](#).

## USB 2.0 Serial Interface Engine (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, USB reset, and remote wake-up signalling (To wake-up the host from suspend mode).

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.

If any packet has USB errors, SIE ignores and discards it. SIE increments the respective USB error count in the [Error Count Register \(ECR\)](#) and sets the respective USB error bits in the [Interrupt Status Register \(ISR\)](#).

The status of the current USB transactions are signalled by the SIE to the [Interrupt Status Register \(ISR\)](#). Certain conditions can be enabled through the IER to generate an interrupt.

Control of the 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 SIE module is coded to reflect the requirements of Chapter 8 of the USB 2.0 Specification.
4. The USB physical bus is connected 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 SIE and AXI interface. Port A of the DPRAM is used by the SIE and Port B is used by the Processor/DMA. Both ports are 32-bit wide.

The AXI USB 2 Device uses two sets of four BRAMs implemented as 64 x 8 bits (DPRAM1) and 2 K x 8 bits each (DPRAM2), with dual asynchronous clock ports.

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 AXI. Data to the USB 2.0

Device is loaded by the user through the AXI to Port B, into appropriate locations in the DPRAM. When the host requests data from the device, the SIE accesses this data from Port A.

The DPRAM is seen by the SIE as DPRAM1 and DPRAM2. DPRAM2 has seven endpoint FIFOs for endpoint 1-7. DPRAM1 has endpoint 0 FIFO and control register area that defines how the memory is arranged.

DPRAM1 consists of endpoint 0 FIFO and control registers of eight endpoints that control the layout of each FIFO of the endpoints in the DPRAM2 and also report the status of each FIFO buffer (ready, not ready, and count).

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 AXI.

## AXI Slave Interface

The AXI Slave interface in the core performs the following operations:

- Responds to AXI transactions to read-from or write-into the 32-bit control registers, status registers, and DPRAM.
- Supports byte, half word, and word transfers for the DPRAM, but only word transfers are supported for the registers.

## AXI Master Interface

The AXI Master interface in the core performs the following operations:

- Performs read and write transactions as a AXI master in DMA mode.
- In DMA mode, interrupts are generated based on DMA done and DMA error conditions.

## DMA Controller

The DMA Controller will be included in the AXI USB2 Device core if `C_INCLUDE_DMA` parameter is set to 1. The DMA Controller provides simple Direct Memory Access services to the DPRAM and external memory device or peripheral on the AXI and vice versa. The DMA controller transfers data from a source address to a destination address without processor intervention for a given length. It provides programmable registers for direction (read-from / write-into DPRAM2), source address, destination address, and transfer length. It supports the setting up of source and destination addresses as incrementing only. It also supports AXI burst transfers. The DMA Controller in the core does the data transfers from/to DPRAM2K to/from external memory only.

## USB 2.0 PHY

The USB PHY can be any ULPI compliant PHY. The primary function of the PHY is to manage the bit level serialization and de-serialization of USB 2.0 traffic. To do so, it must detect and recover the USB 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.

## I/O Signals

Description of the I/O signals for the AXI USB2 Device core is given in [Table 1](#).

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
<b>AXI Global Signals</b>					
P1	M_AXI_ACLK	AXI4	I	-	AXI Master Clock
P2	M_AXI_ARESETN	AXI4	I	-	AXI Master Reset, active LOW
P3	S_AXI_ACLK	AXI4	I	-	AXI Slave Clock
P4	S_AXI_ARESETN	AXI4	I	-	AXI Slave Reset, active LOW
<b>AXI Master Write Address Channel Signals</b>					
P5	M_AXI_AWID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI4	O	0	Master Write address ID. This signal is the identification tag for the write address group of signals.
P6	M_AXI_AWADDR[C_M_AXI_ADDR_WIDTH-1 : 0]	AXI4	O	0	Master Write address. The write address bus gives the address of the first transfer in a write burst transaction
P7	M_AXI_AWLEN[7 : 0]	AXI4	O	0	Master Burst length. This signal gives the exact number of transfers in a burst. "00000000" - "11111111" indicates Burst Length 1 - 256.
P8	M_AXI_AWSIZE[2 : 0]	AXI4	O	0	Master Burst size. This signal indicates the size of each transfer in the write burst.
P9	M_AXI_AWBURST[1 : 0]	AXI4	O	0	Master Burst type. This signal coupled with the size information, details how the address for each write transfer within the burst is calculated
P10	M_AXI_AWCACHE[4 : 0] <sup>(1)</sup>	AXI4	O	0	Master Cache type. This signal provides additional information about the cacheable characteristics of the write transfer.
P11	M_AXI_AWPROT[3 : 0] <sup>(1)</sup>	AXI4	O	2	Master Protection type. This signal indicates the normal, privileged, or secure protection level of the write transaction and whether the transaction is a data access or an instruction access. The default value is normal non secure data access
P12	M_AXI_AWVALID	AXI4	O	0	Master Write address valid. This signal indicates that valid write address and control information are available
P13	M_AXI_AWREADY	AXI4	I	-	Master Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals
P14	M_AXI_AWLOCK <sup>(1)</sup>	AXI4	O	0	Master Lock type: This signal provides additional information about the atomic characteristics of the transfer. "0" - Normal access "1" - Exclusive access
<b>AXI Master Write Channel Signals</b>					
P15	M_AXI_WDATA[C_M_AXI_DATA_WIDTH-1 : 0]	AXI4	O	0	Master Write data bus

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
P16	M_AXI_WSTRB[C_M_AXI_DATA_WIDTH/8-1 : 0]	AXI4	O	0	Master Write strobes. This signal indicates which byte lanes to update in memory
P17	M_AXI_WLAST	AXI4	O	0	Master Write last. This signal indicates the last transfer in a write burst
P18	M_AXI_WVALID	AXI4	O	0	Master Write valid. This signal indicates that valid write data and strobes are available
P19	M_AXI_WREADY	AXI4	I	-	Master Write ready. This signal indicates that the slave can accept the write data
<b>AXI Master Write Response Channel Signals</b>					
P20	M_AXI_BID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI4	I	-	Master Write response ID. This signal is the identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding
P21	M_AXI_BRESP[1 : 0]	AXI4	I	-	Master Write response. This signal indicates the status of the write transaction
P22	M_AXI_BVALID	AXI4	I	-	Master Write response valid. This signal indicates that a valid write response is available
P23	M_AXI_BREADY	AXI4	O	-	Master Response ready. This signal indicates that the master can accept the response information
<b>AXI Master Read Address Channel Signals</b>					
P24	M_AXI_ARID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI4	O	0	Master Read address ID. This signal is the identification tag for the read address group of signals
P25	M_AXI_ARADDR[C_M_AXI_ADDR_WIDTH -1 : 0]	AXI4	O	0	Master Read address. The read address bus gives the initial address of a read burst transaction
P26	M_AXI_ARLEN[7 : 0]	AXI4	O	0	Master Burst length. This signal gives the exact number of transfers in a burst. "00000000" - "11111111" indicates Burst Length 1 - 256.
P27	M_AXI_ARSIZE[2 : 0]	AXI4	O	0	Master Burst size. This signal indicates the size of each transfer in the read burst.
P28	M_AXI_ARBURST[1 : 0]	AXI4	O	0	Master Burst type. The burst type, coupled with the size information, details how the address for each read transfer within the burst is calculated.
P29	M_AXI_ARCACHE[4 : 0] <sup>(1)</sup>	AXI4	O	0	Master Cache type. This signal provides additional information about the cacheable characteristics of the read transfer.
P30	M_AXI_ARPROT[3 : 0] <sup>(1)</sup>	AXI4	O	2	Master Protection type. This signal provides protection unit information for the read transaction. The default value is normal non secure data access
P31	M_AXI_ARVALID	AXI4	O	0	Master Read address valid. This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledgement signal, ARREDY, is high.

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
P32	M_AXI_ARREADY	AXI4	I	-	Master Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
P33	M_AXI_ARLOCK <sup>(1)</sup>	AXI4	O	0	Master Lock type: This signal provides additional information about the atomic characteristics of the transfer. "0" - Normal access "1" - Exclusive access
<b>AXI Master Read Data Channel Signals</b>					
P34	M_AXI_RID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	AXI4	I	-	Master Read ID tag. This signal is the ID tag of the read data group of signals. The RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding.
P35	M_AXI_RDATA[C_M_AXI_DATA_WIDTH - 1 : 0]	AXI4	I	-	Master Read data bus
P36	M_AXI_RRESP[1 : 0]	AXI4	I	-	Master Read response. This signal indicates the status of the read transfer.
P37	M_AXI_RLAST	AXI4	I	-	Master Read last. This signal indicates the last transfer in a read burst
P38	M_AXI_RVALID	AXI4	I	-	Master Read valid. This signal indicates that the required read data is available and the read transfer can complete
P39	M_AXI_RREADY	AXI4	O	0	Master Read ready. This signal indicates that the master can accept the read data and response information
<b>AXI Slave Write Address Channel Signals</b>					
P40	S_AXI_AWID[(C_S_AXI_ID_WIDTH-1) : 0]	AXI4	I	-	Slave Write address ID: This signal is the identification tag for the write address group of signals.
P41	S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH-1) : 0]	AXI4	I	-	Slave Write address: The write address bus gives the address of the first transfer in a write burst transaction.
P42	S_AXI_AWLEN[7 : 0]	AXI4	I	-	Slave Burst length: This signal gives the exact number of transfers in a burst. "00000000" - "11111111" indicates Burst Length 1 - 256.
P43	S_AXI_AWSIZE[2 : 0]	AXI4	I	-	Slave Burst size: This signal indicates the size of each transfer in the burst. "000" - 1 byte "001" - 2 byte (half word) "010" - 4 byte (word) "011" - 8 byte (double word) others - NA

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
P44	S_AXI_AWBURST[1 : 0]	AXI4	I	-	Slave Burst type: This signal coupled with the size information, details how the address for each transfer within the burst is calculated. "00" - FIXED "01" - INCR "10" - WRAP "11" - Reserved
P45	S_AXI_AWLOCK <sup>(1)</sup>	AXI4	I	-	Slave Lock type: This signal provides additional information about the atomic characteristics of the transfer. "0" - Normal access "1" - Exclusive access
P46	S_AXI_AWCACHE[3 : 0] <sup>(1)</sup>	AXI4	I	-	Slave Cache type: This signal indicates the bufferable, cacheable, write-through, write-back and allocate attributes of the transaction Bit-0 : Bufferable (B) Bit-1 : Cacheable (C) Bit-2 : Read Allocate (RA) Bit-3 : Write Allocate (WA) The combination where C=0 and WA/RA=1 are reserved.
P47	S_AXI_AWPROT[2 : 0] <sup>(1)</sup>	AXI4	I	-	Slave Protection type: This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access. Bit-0 : 0=Normal access, 1=Privileged access Bit-1 : 0=Secure access, 1=Non-secure access Bit- 2 : 0=Data access; 1=Instruction access
P48	S_AXI_AWVALID	AXI4	I	-	Slave Write address valid: This signal indicates that valid write address and control information are available.
P49	S_AXI_AWREADY	AXI4	O	0	Slave Write address ready: This signal indicates that the slave is ready to accept an address and associated control signals.
<b>AXI Slave Write Channel Signals</b>					
P50	S_AXI_WDATA[(C_S_AXI_DATA_WIDTH-1) : 0]	AXI4	I	-	Slave Write data bus.
P51	S_AXI_WSTRB[((C_S_AXI_DATA_WIDTH/8)-1) : 0]	AXI4	I	-	Slave Write strobes: This signal indicates which byte lanes in S_AXI_WDATA are/is valid.
P52	S_AXI_WLAST	AXI4	I	-	Slave Write last: This signal indicates the last transfer in a write burst.
P53	S_AXI_WVALID	AXI4	I	-	Slave Write valid: This signal indicates that valid write data and strobes are available.
P54	S_AXI_WREADY	AXI4	O	0	Slave Write ready: This signal indicates that the slave can accept the write data.
<b>AXI Slave Write Response Channel Signals</b>					

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
P55	S_AXI_BID[(C_S_AXI_ID_WIDTH-1) : 0]	AXI4	O	0	Slave Write response ID: This signal is the identification tag of the write response. The BID value must match the AWID value of the write transaction to which the slave is responding.
P56	S_AXI_BRESP[1 : 0]	AXI4	O	0	Slave Write response: This signal indicates the status of the write transaction. "00" - OKAY "01" - EXOKAY - NA "10" - SLVERR "11" - DECERR - NA
P57	S_AXI_BVALID	AXI4	O	0	Slave Write response valid: This signal indicates that a valid write response is available.
P58	S_AXI_BREADY	AXI4	I	-	Slave Response ready: This signal indicates that the master can accept the response information.
<b>AXI Slave Read Address Channel Signals</b>					
P59	S_AXI_ARID[(C_S_AXI_ID_WIDTH-1) : 0]	AXI4	I	-	Slave Read address ID: This signal is the identification tag for the read address group of signals.
P60	S_AXI_ARADDR[(C_S_AXI_ADDR_WIDTH -1) : 0]	AXI4	I	-	Slave Read address: The read address bus gives the initial address of a read burst transaction.
P61	S_AXI_ARLEN[7 : 0]	AXI4	I	-	Slave Burst length: This signal gives the exact number of transfers in a burst. "00000000" - "11111111" indicates Burst Length 1 - 256.
P62	S_AXI_ARSIZE[2 : 0]	AXI4	I	-	Slave Burst size: This signal indicates the size of each transfer in the burst. "000" - 1 byte "001" - 2 byte (Half word) "010" - 4 byte (word) "011" - 8 byte (double word) others - NA
P63	S_AXI_ARBURST[1 : 0]	AXI4	I	-	Slave Burst type: The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated. "00" - FIXED "01" - INCR "10" - WRAP "11" - Reserved
P64	S_AXI_ARLOCK <sup>(1)</sup>	AXI4	I	-	Slave Lock type: This signal provides additional information about the atomic characteristics of the transfer. "0" - Normal access "1" - Exclusive access

Table 1: I/O Signals

Port	Signal Name	Interface	I/O	Initial State	Description
P65	S_AXI_ARCACHE[3 : 0] <sup>(1)</sup>	AXI4	I	-	Slave Cache type: This signal provides additional information about the cacheable characteristics of the transfer. Bit-0 : Bufferable (B) Bit-1 : Cacheable (C) Bit-2 : Read Allocate (RA) Bit-3 : Write Allocate (WA) The combination where C=0 and WA/RA=1 are reserved.
P66	S_AXI_ARPROT[2 : 0] <sup>(1)</sup>	AXI4	I	-	Slave Protection type: This signal provides protection unit information for the transaction.
P67	S_AXI_ARVALID	AXI4	I	-	Slave Read address valid: This signal indicates, when HIGH, that the read address and control information is valid and will remain stable until the address acknowledgement signal, S_AXI_ARREADY, is high.
P68	S_AXI_ARREADY	AXI4	O	0	Slave Read address ready: This signal indicates that the slave is ready to accept an address and associated control signals.
<b>AXI Slave Read Data Channel Signals</b>					
P69	S_AXI_RID[(C_S_AXI_ID_WIDTH - 1) : 0]	AXI4	O	0	Slave Read ID tag: This signal is the ID tag of the read data group of signals. The S_AXI_RID value is generated by the slave and must match the ARID value of the read transaction to which it is responding.
P70	S_AXI_RDATA[(C_S_AXI_DATA_WIDTH - 1) : 0]	AXI4	O	0	Slave Read data bus.
P71	S_AXI_RRESP[1 : 0]	AXI4	O	0	Slave Read response: This signal indicates the status of the read transfer. "00" - OKAY "01" - EXOKAY - NA "10" - SLVERR "11" - DECERR - NA
P72	S_AXI_RLAST	AXI4	O	0	Slave Read last: This signal indicates the last transfer in a read burst.
P73	S_AXI_RVALID	AXI4	O	0	Slave Read valid: This signal indicates that the required read data is available and the read transfer can complete.
P74	S_AXI_RREADY	AXI4	I	-	Slave Read ready: This signal indicates that the master can accept the read data and response information.
<b>System Signals</b>					
P75	USB_Irpt	System	O	0	USB Interrupt
<b>USB Specific Signals</b>					
P76	ULPI_Clock	USB	I	-	All USB protocol interface signals are synchronous to this clock
P77	ULPI_Dir	USB	I	-	Direction of Data flow between PHY and SIE

**Table 1: I/O Signals**

Port	Signal Name	Interface	I/O	Initial State	Description
P78	ULPI_Next	USB	I	-	Indicator of when the PHY is ready for the next bit
P79	ULPI_Stop	USB	O	0	Indicator that transmission of last byte is complete
P80	ULPI_Reset	USB	O	0	Active High Reset to the PHY
P81	ULPI_Data_I(7:0)	USB	I	-	Input Data from PHY to SIE
P82	ULPI_Data_O(7:0)	USB	O	0	Output Data from SIE to PHY
P83	ULPI_Data_T	USB	O	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 tri-state output
<b>Optional Ports Used for Debug Purpose</b>					
P84	Configured	USB	O	0	Used for USB2.0 Certification - Test mode 2
P85	Spare1	USB	O	0	Used for USB2.0 Certification -Test mode 0
P86	Spare2	USB	O	0	Used for USB2.0 Certification - Test mode 1
P87	Vbus_detect	USB	O	0	'0' = Indicates valid V <sub>BUS</sub> has not been detected '1' = Indicates Valid V <sub>BUS</sub> has been detected
P88	Show_currentspeed	USB	O	0	'0' = indicates Full-speed '1' = indicates High-speed
P89	Running	USB	O	0	'0' = indicates that the SIE in reset state and will not respond to USB traffic '1' = indicates that SIE finished USB reset and ready to respond to USB traffic
P90	Suspended	USB	O	0	'0' = Indicates AXI USB2 Device has been suspended '1' = Indicates AXI USB2 Device has not been suspended
P91	Disconnected	USB	O	0	'0' = Indicates AXI USB2 Device connected '1' = Indicates AXI USB2 Device disconnected

**Notes:**

1. The design will not respond for any specific combination to these signals.

## Design Parameters

To obtain an AXI USB2 Device that is uniquely tailored to the user system requirements, certain features can be parameterized in the AXI USB2 Device design. The features that can be parameterized in the Xilinx AXI USB2 Device design are shown in [Table 2](#).

## Inferred Parameters

In addition to the parameters listed in [Table 2](#), there are also parameters that are inferred for each AXI interface in the EDK tools. Through the design, these EDK-inferred parameters control the behavior of the AXI Interconnect. For a complete list of the interconnect settings related to the AXI interface, see DS768, *AXI Interconnect IP Data Sheet*.

Table 2: Design Parameters

Generic	Feature/Description	Parameter Name	Allowable Values	Reset Value	VHDL Type
<b>System Parameter</b>					
G1	Device family	C_FAMILY	spartan6, virtex6	virtex6	string
G2	AXI USB2 Device Base Address	C_BASEADDR	Valid Address <sup>(1),(2)</sup>	None	std_logic_vector
G3	AXI USB2 Device High Address	C_HIGHADDR	Valid Address <sup>(1),(2)</sup>	None	std_logic_vector
<b>AXI4 Master Interface Parameters</b>					
G4	AXI Master address bus width	C_M_AXI_ADDR_WIDTH	32	32	integer
G5	AXI Master data bus width	C_M_AXI_DATA_WIDTH	32	32	integer
G6	AXI Master Identification tag width	C_M_AXI_THREAD_ID_WIDTH	1-2	1	integer
<b>AXI4 Slave Interface Parameters</b>					
G7	AXI Slave address bus width	C_S_AXI_ADDR_WIDTH	32	32	integer
G8	AXI Slave data bus width	C_S_AXI_DATA_WIDTH	32	32	integer
G9	AXI Slave Identification tag width	C_S_AXI_ID_WIDTH	1- 16	4	integer
<b>USB2.0 Core Specific Parameters</b>					
G10	Implementation of DMA	C_INCLUDE_DMA	0-1 <sup>(3)</sup>	1	integer

**Table 2: Design Parameters**

G11	Type of ULPI PHY Reset	C_PHY_RESET_TYPE	0-1 <sup>(4)</sup>	1	integer
G12	Enables USB Error Counters	C_INCLUDE_USBERR_LOGIC	0-1 <sup>(5)</sup>	0	integer

**Notes:**

1. Address range specified by C\_BASEADDR and C\_HIGHADDR must be at least 0x8000 and must be power of 2. C\_BASEADDR must be multiple of the range, where the range is C\_HIGHADDR - C\_BASEADDR + 1
2. No default value will be specified to insure that the actual value is set, i.e if the value is not set, a compiler error will be generated. The address range must be at least 0x7FFF. For example, C\_BASEADDR = 0x80000000, C\_HIGHADDR = 0x80007FFF.
3. If C\_INCLUDE\_DMA = 1, the DMA controller logic will be included in the core
4. If C\_PHY\_RESET\_TYPE = 1, then core puts active HIGH reset condition on ULPI Reset port. Otherwise core drives active LOW hard reset condition.
5. If C\_INCLUDE\_USBERR\_LOGIC = 1 includes USB ERROR interrupt generation logic, otherwise core excludes the USB ERROR interrupt generation logic from the core. This parameter is added to reduce the resources used for the generation of USB ERROR interrupt and the respective counters from the core

## Allowable Parameter Combinations

The address-range size specified by C\_BASEADDR and C\_HIGHADDR must be a power of 2, and must be at least 0x7FFF. For example, if C\_BASEADDR = 0x80000000, C\_HIGHADDR must be at least = 0x80007FFF.

## Parameter - Port Dependencies

The width of some of the AXI USB2 Device signals depends on parameters selected in the design. The dependencies between the AXI USB2 Device design parameters and I/O signals are shown in [Table 3](#).

**Table 3: Parameter Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
<b>Design Parameters</b>				
G4	C_M_AXI_ADDR_WIDTH		-	Affects number of bits in the AXI Master address bus
G5	C_M_AXI_DATA_WIDTH		-	Affects number of bits in the AXI Master data bus
G6	C_M_AXI_THREAD_ID_WIDTH		-	Affects number of bits in the AXI Master Identification tag
G7	C_S_AXI_ADDR_WIDTH		-	Affects number of bits in the AXI Slave address bus
G8	C_S_AXI_DATA_WIDTH		-	Affects number of bits in the AXI Slave address bus
G9	C_S_AXI_ID_WIDTH		-	Affects number of bits in the AXI Slave Identification tag
<b>I/O Signals</b>				
P5	M_AXI_AWID[C_M_AXI_THR_EAD_ID_WIDTH-1 : 0]	-	G6	Width varies depending on the size of the AXI Master write address ID
P6	M_AXI_AWADDR[C_M_AXI_ADDR_WIDTH-1 : 0]	-	G4	Width varies depending on the size of the AXI Master write address bus
P15	M_AXI_WDATA[C_M_AXI_DATA_WIDTH-1 : 0]	-	G5	Width varies depending on the size of the AXI Master write data bus
P16	M_AXI_WSTRB[C_M_AXI_DATA_WIDTH/8-1 : 0]	-	G5	Width varies depending on the size of the AXI Master write strobes

Table 3: Parameter Port Dependencies

Generic or Port	Name	Affects	Depends	Relationship Description
P20	M_AXI_BID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	-	G6	Width varies depending on the AXI Master response ID
P24	M_AXI_ARID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	-	G6	Width varies depending on the AXI Master read address ID
P25	M_AXI_ARADDR[C_M_AXI_ADDR_WIDTH -1 : 0]	-	G4	Width varies depending on the AXI Master read address bus
P34	M_AXI_RID[C_M_AXI_THREAD_ID_WIDTH-1 : 0]	-	G6	Width varies depending on the AXI Master read ID tag
P35	M_AXI_RDATA[C_M_AXI_DATA_WIDTH -1 : 0]	-	G5	Width varies depending on the AXI Master read data bus
P40	S_AXI_AWID[(C_S_AXI_ID_WIDTH-1) : 0]	-	G9	Width varies depending on the AXI Slave write address ID
P41	S_AXI_AWADDR[(C_S_AXI_ADDR_WIDTH-1) : 0]	-	G7	Width varies depending on the AXI Slave write address bus
P50	S_AXI_WDATA[(C_S_AXI_DATA_WIDTH-1) : 0]	-	G8	Width varies depending on the AXI Slave write data bus
P51	S_AXI_WSTRB[((C_S_AXI_DATA_WIDTH/8)-1) : 0]	-	G8	Width varies depending on the AXI Slave write strobes
P55	S_AXI_BID[(C_S_AXI_ID_WIDTH-1) : 0]	-	G9	Width varies depending on the AXI Slave response ID
P69	S_AXI_RID[(C_S_AXI_ID_WIDTH - 1) : 0]	-	G9	Width varies depending on the AXI Slave read ID tag
P70	S_AXI_RDATA[(C_S_AXI_DATA_WIDTH -1) : 0]	-	G8	Width varies depending on the AXI Slave read data bus

## Register Bit Ordering

All registers use little-endian bit ordering where bit-31 is MSB and bit-0 is LSB. Table 4 shows the bit ordering.

Table 4: Register Bit Ordering

<b>31</b>	<b>30</b>	<b>29</b>	.....	<b>2</b>	<b>1</b>	<b>0</b>
MSB			.....			LSB

## Register Description

The memory map for the AXI USB2 Device core is shown in Table 5 which includes endpoint configuration space (offset 0x0000), setup packet storage space (offset 0x0080), RAM for endpoint 0 buffer (offset 0x0088), register space for the USB registers (offset 0x0100), and RAM for endpoint 1 - 7 buffers (offset 0x4000). Table 6 lists the mapping for endpoint configuration space. All offsets are word offsets.

Table 5: Register Address Map <sup>(1),(2)</sup>

Register Name	Base Address + Offset (hex)	Reset Value (hex)	Access
Endpoint Configuration and Status Registers	C_BASEADDR + 0x0000	0x00000000	R/W
Setup Packet Storage Word 0 <sup>(2)</sup>	C_BASEADDR + 0x0080	0x00000000	R
Setup Packet Storage Word 1 <sup>(2)</sup>	C_BASEADDR + 0x0084	0x00000000	R
RAM for endpoint 0 buffer <sup>(6)</sup>	C_BASEADDR + 0x0088	0x00000000	R/W
USB Address Register	C_BASEADDR + 0x0100	0x00000000	R/W
Control Register	C_BASEADDR + 0x0104	0x00000000	R/W
Interrupt Status Register	C_BASEADDR + 0x0108	0x00000000	R
Frame Number Register	C_BASEADDR + 0x010C	0x00000000	R
Interrupt Enable Register	C_BASEADDR + 0x0110	0x00000000	R/W
Buffer Ready Register	C_BASEADDR + 0x0114	0x00000000	R/W
Test Mode Register	C_BASEADDR + 0x0118	0x00000000	R/W
Error Count Register <sup>(7)</sup>	C_BASEADDR + 0x011C	0x00000000	R
DMA Software Reset Register <sup>(3)(4)</sup>	C_BASEADDR + 0x0200	0x00000000	W
DMA Control Register <sup>(4)</sup>	C_BASEADDR + 0x0204	0x00000000	R/W
DMA Source Address Register <sup>(4)</sup>	C_BASEADDR + 0x0208	0x00000000	R/W
DMA Destination Address Register <sup>(4)</sup>	C_BASEADDR + 0x020C	0x00000000	R/W
DMA Length Register <sup>(4)</sup>	C_BASEADDR + 0x0210	0x00000000	R/W
DMA Status Register <sup>(4)</sup>	C_BASEADDR + 0x0214	0x00000000	R/W
RAM for endpoint 1-7 buffers <sup>(6)</sup>	C_BASEADDR + 0x4000	0x00000000	R/W

**Notes:**

1. R/W - Read and Write access
2. R - Read Only access. Write into these registers does not have any effect
3. W - Write Only access. Reading of these registers returns zero
4. This Register will not be included in the design if the parameter C\_INCLUDE\_DMA = 0
5. RAM for endpoint 0 buffer should be 64 bytes, EP\_BASE\_ADDRESS (endpoint 0)+ 0x40 should not exceed 0x00FF
6. RAM for endpoint 1 - 7 buffers range (C\_BASEADDR + 0x4000) to (C\_BASEADDR + 0x4000) + 0x1FFF
7. This Register will not be included in the design if the parameter C\_INCLUDE\_USBERR\_LOGIC = 0. But SIE performs usb error checks as part of the USB2.0 Specification ignores the error packets if received

## Endpoint Configuration and 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 6](#).

**Table 6: Endpoint Configuration Registers ( $C\_BASEADDR + \text{Address Offset}$ )**

Address Offset	Memory/Register Space
0x0000	Endpoint 0
0x0010	Endpoint 1
0x0020	Endpoint 2
0x0030	Endpoint 3
0x0040	Endpoint 4
0x0050	Endpoint 5
0x0060	Endpoint 6
0x0070	Endpoint 7

Each endpoint has four 32-bit registers that describe the behavior of the endpoint. These registers are located sequentially and arranged by endpoint number as shown in [Table 7](#).

**Table 7: Endpoint Configuration Register**

Address Offset	Memory/Register Space
0x0000	Endpoint configuration and status register
0x0004	Reserved
0x0008	Buffer 0 count: 0 to 1024
0x000C	Buffer 1 count: 0 to 1024

The bit description for the Endpoint Configuration and Status Registers is given in [Table 8](#). 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 8: Endpoint Configuration and Status Register ( $C\_BASEADDR + 0x0000$ ) (6),(7)**

Bit(s)	Name	Access	Reset Value	Description
31	EP_VALID <sup>(1)</sup>	R/W	0	'0' = disables the endpoint '1' = enables the endpoint
30	EP_STALL	R/W	0	'0' = endpoint accepts IN's and OUT's '1' = endpoint responds to the host only with a STALL
29	EP_OUT_IN	R/W	0	'0' = core receives data from the host (OUT from host) '1' = core sends data to the host (IN to host)
28	EP_ISO <sup>(2)</sup>	R/W	0	'0' = not an isochronous endpoint '1' = is an isochronous endpoint
27	EP_DATA_TOGGLE <sup>(3)</sup>	R/W	0	'0' = next DATA packet must be a DATA0 packet '1' = next DATA packet must be a DATA1 packet

Table 8: Endpoint Configuration and Status Register ( $C\_BASEADDR + 0x0000$ ) (Cont'd) (6),(7)

Bit(s)	Name	Access	Reset Value	Description
26	EP_BUFFER_SELECT <sup>(4)</sup>	R/W	0	'0' = Buffer 0 is used '1' = Buffer 1 is used
25-15	EP_PACKET_SIZE <sup>(5)</sup>	R/W	0	Endpoint packet size
14-13	Reserved	NA	-	Reserved for future use
12-0	EP_BASE <sup>(8)</sup>	R/W	0	Base offset of the buffers in the DPRAM

**Notes:**

1. EP\_VALID is a Master Enable bit for the respective endpoint.
2. EP\_ISO enables endpoint as an isochronous endpoint.
3. SIE uses EP\_DATA\_TOGGLE to detect DATA PID toggle errors during the data transfers and its weak form of synchronization technique. This bit can be explicitly set in response to a firmware command.
4. Used to support ping-pong buffers during data transfers. If this bit is set then SIE toggles the buffer access during data transfers one at each time.
5. EP\_PACKET\_SIZE refers to maximum packet size limited to the respective endpoint.
6. 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 host, set EP\_VALID = '1', EP\_OUT\_IN = '0', EP\_STALL = '0', and EP\_ISO = '0'.
7. The EP\_DATA\_TOGGLE and EP\_BUFFER\_SELECT bits are modified by the SIE in response to the USB operations and only their initial values are set by the firmware.
8. EP\_BASE is word addressable, user should always right shift the endpoint buffer address by 2 before writing EP\_BASE.

### Buffer Count Register0 (BCR0)

The Buffer 0 Count Registers shown in Table 9, indicate the size of data in bytes. 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 9: Buffer Count Register0 ( $C\_BASEADDR + 0x0008$ )

Bit(s)	Name	Access	Reset Value	Description
31-11	Reserved	R/W	0	Reserved for future use
10-0	(In/Out)_Pkt_Count(10:0)	R/W	0	Packet count in the buffer

### Buffer Count Register1 (BCR1)

The Buffer 1 Count Registers shown in Table 10 indicate the size of data in bytes. If the endpoint is an OUT endpoint, 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, the firmware sets the value of this register before transmission.

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

Table 10: Buffer Count Register1 ( $C\_BASEADDR + 0x000c$ )

Bit(s)	Name	Access	Reset Value	Description
31-11	Reserved	R/W	0	Reserved for future use
10-0	(In/Out)_Pkt_Count(10:0)	R/W	0	Packet count in the buffer

## USB Address Register (UAR)

The USB Address register, shown in Table 11 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 11: USB Address Register (C\_BASEADDR + 0x0100)

Bit(s)	Name	Access	Reset Value	Description
31-7	Reserved	NA	-	Reserved for future use
6-0	USB Address	R/W	0	Indicates the USB address of the device.

## Control Register (CR)

As shown in Table 12, the MASTER\_READY 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. The Remote\_Wakeup bit will initiate remote wake-up signalling to the host when the device has been suspended by the host. The Remote\_Wakeup bit should be set by the Firmware only when AXI USB2 Device is in suspend mode. If the firmware set Remote\_Wakeup bit when AXI USB2 Device is not in suspend mode, AXI USB2 Device will not issue remote wake-up signalling to the host. The core generates USB Resume interrupt after the successful completion of remote wake-up signalling with host and clears Remote\_Wakeup bit.

Table 12: Control Register (C\_BASEADDR + 0x0104)

Bit(s)	Name	Access	Reset Value	Description
31	MASTER_READY	R/W	0	'0' = SIE is paused and does not respond to any USB activity '1' = SIE operates normally
30	Remote_Wakeup <sup>(1)</sup>	R/W	0	'0' = SIE will do nothing '1' = SIE will send remote wake-up signalling to host
29-0	Reserved	NA	-	Reserved for future use

### Notes:

1. If processor set the Remote\_Wakeup bit when SIE in normal state, the AXI USB2 Device will not send the Remote\_Wakeup signalling to host.

## Interrupt Status Register (ISR)

The Interrupt Status Register (ISR) shown in Table 13, reports status on the operation of the AXI USB2 Device core. Bits in this register get cleared as soon as they are read.

Table 13: Interrupt Status Register (C\_BASEADDR + 0x0108)

Bit(s)	Name	Access	Reset Value	Description
31-30	Reserved	NA	-	Reserved for future use.
29	Bit Stuff Error <sup>(2)</sup>	R	0	'0' = Bit Stuff error has not occurred '1' = Bit Stuff error has occurred
28	PID Error <sup>(2)</sup>	R	0	'0' = PID error has not occurred '1' = PID error has occurred

**Table 13: Interrupt Status Register (C\_BASEADDR + 0x0108) (Cont'd)**

Bit(s)	Name	Access	Reset Value	Description
27	CRC Error <sup>(2)</sup>	R	0	'0' = CRC error has not occurred '1' = CRC error has occurred
26	DMA Done <sup>(1)</sup>	R	0	'0' = DMA operation is not done '1' = DMA operation is done
25	DMA Error <sup>(1)</sup>	R	0	'0' = DMA error has not occurred '1' = DMA error has occurred
24	USB Resume <sup>(3)</sup>	R	0	'0' = Core has not received resume signalling from host '1' = Core received resume signalling from host
23	USB Reset <sup>(3)</sup>	R	0	'0' = Core has not received USB reset from host '1' = Core received USB reset from host
22	USB Suspend <sup>(3)</sup>	R	0	'0' = Core not in suspend state '1' = Core in suspend state
21	USB Disconnect <sup>(3)</sup>	R	0	'0' = Core connected to host '1' = Core is disconnected from host
20	FIFO Buf Rdy	R	0	'0' = Endpoint 0 packet has not been received by core '1' = Endpoint 0 packet has been received by core
19	FIFO Buf Free	R	0	'0' = Endpoint 0 packet has been transmitted by core '1' = Endpoint 0 packet has not been transmitted by core
18	SETUP Packet	R	0	'0' = Endpoint 0 Setup packet has not been received by core '1' = Endpoint 0 Setup packet has been received
17	SOF Packet	R	0	'0' = Start of Frame packet has not been received by core '1' = Start of Frame packet has been received by core
16	High Speed	R	0	'0' = Core is running at Full-speed '1' = core is running at High-speed
15	EP-7 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 7 '1' = Core has been processed the 2nd buffer of endpoint 7
14	EP-6 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 6 '1' = Core has been processed the 2nd buffer of endpoint 6
13	EP-5 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 5 '1' = Core has been processed the 2nd buffer of endpoint 5
12	EP-4 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 4 '1' = Core has been processed the 2nd buffer of endpoint 4
11	EP-3 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 3 '1' = Core has been processed the 2nd buffer of endpoint 3
10	EP-2 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 2 '1' = Core has been processed the 2nd buffer of endpoint 2
9	EP-1 2nd Buf Comp	R	0	'0' = Core has not been processed the 2nd buffer of endpoint 1 '1' = Core has been processed the 2nd buffer of endpoint 1
8	Reserved	NA	-	NA
7	EP-7 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 7 '1' = Core has been processed the 1st buffer of endpoint 7

Table 13: Interrupt Status Register ( $C\_BASEADDR + 0x0108$ ) (Cont'd)

Bit(s)	Name	Access	Reset Value	Description
6	EP-6 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 6 '1' = Core has been processed the 1st buffer of endpoint 6
5	EP-5 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 5 '1' = Core has been processed the 1st buffer of endpoint 5
4	EP-4 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 4 '1' = Core has been processed the 1st buffer of endpoint 4
3	EP-3 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 3 '1' = Core has been processed the 1st buffer of endpoint 3
2	EP-2 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 2 '1' = Core has been processed the 1st buffer of endpoint 2
1	EP-1 1st Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 1 '1' = Core has been processed the 1st buffer of endpoint 1
0	EP-0 Buf Comp	R	0	'0' = Core has not been processed the 1st buffer of endpoint 0 '1' = Core has been processed the 1st buffer of endpoint 0

**Notes:**

1. This bit will be undefined if the parameter `C_INCLUDE_DMA = 0`
2. This bit will be undefined if the parameter `C_INCLUDE_USBERR_LOGIC = 0`
3. This bit indicate the current status of the AXI USB2 Device core.

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 \(IER\)](#) allows specific bits of the [Interrupt Status Register \(ISR\)](#) 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 [Interrupt Status Register \(ISR\)](#) 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 ISR 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 ISR, that is, by reading the ISR, provided that the corresponding bit in the IER is '1'.
- Changing an IER bit from '1' to '0', when the corresponding bit in the ISR 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.

## Frame Number Register (FNR)

The [Frame Number Register \(FNR\)](#) shown in [Table 14](#) 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  $\mu$ s. When the USB is operated in the High Speed mode, this can generate a potentially high rate of interrupts. Therefore, 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 14: Frame Number Register (C\_BASEADDR + 0x010C)**

Bit(s)	Name	Access	Reset Value	Description
31-14	Reserved	NA	-	Reserved for future use
13-3	Frame number(10:0)	R	0	Frame numbers - 0 to 2047
2-0	Microframe number(2:0)	R	0	Microframe numbers - 0 to 7

## Interrupt Enable Register (IER)

The **Interrupt Enable Register (IER)** shown in [Table 15](#) allows specific bits of the **Interrupt Status Register (ISR)** 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 **Interrupt Status Register (ISR)** 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 15: Interrupt Enable Register (C\_BASEADDR + 0x0110)**

Bit(s)	Name	Access	Reset Value	Description
31	Master Enable	R/W	0	'0' = Disables the setting of all other interrupts '1' = Enables setting of all other interrupts
30	Reserved	NA	-	Reserved for future use
29	Bit Stuff Error <sup>(2)</sup>	R/W	0	'0' = Disables Bit Stuff error interrupt '1' = Enable Bit Stuff error interrupt
28	PID Error <sup>(2)</sup>	R/W	0	'0' = Disables PID error interrupt '1' = Enables PID error interrupt
27	CRC Error <sup>(2)</sup>	R/W	0	'0' = Disables CRC error interrupt '1' = Enables CRC error interrupt
26	DMA Done <sup>(1)</sup>	R/W	0	'0' = Disables DMA Done interrupt '1' = Enables DMA Done interrupt
25	DMA Error <sup>(1)</sup>	R/W	0	'0' = Disables DMA Error interrupt '1' = Enables DMA Error interrupt
24	USB Resume	R/W	0	'0' = Disables USB Resume interrupt '1' = Enables USB Resume interrupt
23	USB Reset	R/W	0	'0' = Disables USB Reset interrupt '1' = Enables USB Reset interrupt
22	USB Suspend	R/W	0	'0' = Disables USB Suspend interrupt '1' = Enables USB Suspend interrupt
21	USB Disconnect	R/W	0	'0' = Disables USB Disconnect interrupt '1' = Enables USB Disconnect interrupt
20	FIFO Buf Rdy	R/W	0	'0' = Disables FIFO Buf Rdy interrupt '1' = Enables FIFO Buf Rdy interrupt
19	FIFO Buf Free	R/W	0	'0' = Disables FIFO Buf Free interrupt '1' = Enables FIFO Buf Free interrupt

**Table 15: Interrupt Enable Register ( $C\_BASEADDR + 0x0110$ ) (Cont'd)**

Bit(s)	Name	Access	Reset Value	Description
18	Setup Packet	R/W	0	'0' = Disables Setup Packet received interrupt '1' = Enables Setup Packet received interrupt
17	SOF Packet	R/W	0	'0' = Disables Start of Frame received interrupt '1' = Enables Start of Frame received interrupt
16	High Speed	R/W	0	'0' = Disables core operates in High Speed interrupt '1' = Enables core operates in High Speed interrupt
15	EP-7 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 7 is complete interrupt '1' = Enables 2nd buffer of Endpoint 7 is complete interrupt
14	EP-6 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 6 is complete interrupt '1' = Enables 2nd buffer of Endpoint 6 is complete interrupt
13	EP-5 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 5 is complete interrupt '1' = Enables 2nd buffer of Endpoint 5 is complete interrupt
12	EP-4 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 4 is complete interrupt '1' = Enables 2nd buffer of Endpoint 4 is complete interrupt
11	EP-3 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 3 is complete interrupt '1' = Enables 2nd buffer of Endpoint 3 is complete interrupt
10	EP-2 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 2 is complete interrupt '1' = Enables 2nd buffer of Endpoint 2 is complete interrupt
9	EP-1 2nd Buf Comp	R/W	0	'0' = Disables 2nd buffer of Endpoint 1 is complete interrupt '1' = Enables 2nd buffer of Endpoint 1 is complete interrupt
8	Reserved	R/W	0	Not used
7	EP-7 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 7 is complete interrupt '1' = Enables 1st buffer of Endpoint 7 is complete interrupt
6	EP-6 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 6 is complete interrupt '1' = Enables 1st buffer of Endpoint 6 is complete interrupt
5	EP-5 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 5 is complete interrupt '1' = Enables 1st buffer of Endpoint 5 is complete interrupt
4	EP-4 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 4 is complete interrupt '1' = Enables 1st buffer of Endpoint 4 is complete interrupt
3	EP-3 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 3 is complete interrupt '1' = Enables 1st buffer of Endpoint 3 is complete interrupt
2	EP-2 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 2 is complete interrupt '1' = Enables 1st buffer of Endpoint 2 is complete interrupt
1	EP-1 1st Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 1 is complete interrupt '1' = Enables 1st buffer of Endpoint 1 is complete interrupt
0	EP-0 Buf Comp	R/W	0	'0' = Disables 1st buffer of Endpoint 0 is complete interrupt '1' = Enables 1st buffer of Endpoint 0 is complete interrupt

**Notes:**

- This bit will be undefined if the parameter `C_INCLUDE_DMA = 0`
- This bit will be undefined if the parameter `C_INCLUDE_USBERR_LOGIC = 0`

## Buffer Ready Register (BRR)

The [Buffer Ready Register \(BRR\)](#) has a buffer ready bit corresponding to each buffer of each endpoint, as shown in [Table 16](#). 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 Configuration and Status register of the corresponding endpoint. It should be noted that per the USB 2.0 Specification, endpoint 0 has only one buffer that handles IN or OUT.

Table 16: Buffer Ready Register ( $C\_BASEADDR + 0x0114$ )

Bit(s)	Name	Access	Reset Value	Description
31-16	Reserved	R	0	Reserved for future use
15	EP-7 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 7 is not available for IN or OUT interrupt '1' = Enables 2nd buffer of Endpoint 7 is available for IN or OUT interrupt
14	EP-6 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 6 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 6 is available for IN or OUT interrupt
13	EP-5 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 5 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 5 is available for IN or OUT interrupt
12	EP-4 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 4 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 4 is available for IN or OUT interrupt
11	EP-3 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 3 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 3 is available for IN or OUT interrupt
10	EP-2 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 2 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 2 is available for IN or OUT interrupt
9	EP-1 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 1 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 1 is available for IN or OUT interrupt
8	Not Used	NA	-	Endpoint 0 has only one buffer
7	EP-7 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 7 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 7 is available for IN or OUT interrupt
6	EP-6 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 6 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 6 is available for IN or OUT interrupt
5	EP-5 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 5 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 5 is available for IN or OUT interrupt
4	EP-4 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 4 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 4 is available for IN or OUT interrupt
3	EP-3 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 3 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 3 is available for IN or OUT interrupt
2	EP-2 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 2 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 2 is available for IN or OUT interrupt
1	EP-1 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 1 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 1 is available for IN or OUT interrupt
0	EP-0 Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 0 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 0 is available for IN or OUT interrupt

## Test Mode Register (TMR)

The [Test Mode Register \(TMR\)](#) described in [Table 17](#) defines the different test modes in which the AXI USB2 Device operates. The USB Implementers Forum, the organization that controls USB logo certification, requires all USB 2.0 devices that operate at High Speed support these test modes.

Table 17: Test Mode Register ( $C\_BASEADDR + 0x0118$ )

Bit(s)	Name	Access	Reset Value	Description
31-3	Reserved	R	0	Reserved for future use
2-0	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

The AXI USB2 Device provides test mode support to facilitate compliance testing.

Four 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.

## Error Count Register (ECR)

The [Error Count Register \(ECR\)](#) (ECR) is described in [Table 18](#). This register contains three counters each of 8-bit width. They are Bit Stuff error count, PID error count, and CRC error count. When USB Reset or read to this register is requested, all these counters will be cleared and assigned to reset values.

When PHY detects seven consecutive ones (1s) on the bus (*bit stuff error condition*), SIE increments Bit Stuff error count by one and Bit Stuff error bit of [Interrupt Status Register \(ISR\)](#) is set.

Whenever four PID check bits are not complement to their respective packet identifier bits while receiving the packet, SIE increments PID error count by one and PID error bit of [Interrupt Status Register \(ISR\)](#) is set.

Whenever the received CRC doesn't match with the CRC calculated on the received packet (i.e., for CRC5 while receiving token and for CRC16 while receiving data), SIE increments CRC error count by one and CRC error bit of [Interrupt Status Register \(ISR\)](#) is set.

Table 18: Error Count Register (C\_BASEADDR + 0x011C) (1)(2)

Bit(s)	Name	Access	Reset Value	Description
31-24	Bit Stuff error count	R	0x0	Bit Stuff Error Counter
23-16	PID error count	R	0x0	PID Error Counter
15-8	CRC error count	R	0x0	CRC Error Counter
7-0	Reserved	R	0x0	Reserved for future use

**Notes:**

1. This register is read-only.
2. When any of the counter reaches 255, it will roll back and start counting from 0.

### DMA Software Reset Register (DSRR)

The DMA Software Reset Register (DSRR) shown in Table 19 defines reset to the DMA modules by AXI USB2 Device core when C\_INCLUDE\_DMA set to 1.

Table 19: DMA Software Reset Register (C\_BASEADDR + 0x0200)

Bit(s)	Name	Access	Reset Value	Description
31-0	RST	W	N/A	A write of 0x0000000A causes the reset to the DMA modules

### DMA Control Register (DMACR)

The DMA Control Register (DMACR) described in Table 20 defines the Direction of the transfer as either Read or Write to DPRAM. The Endpoint Buffer Select bit determines whether the SIE side Buffer Ready status is updated by Hardware once the DMA is complete or by firmware through the BRR register. If Endpoint Buffer Select is set to 1, the DMA Controller updates Buffer Ready status to SIE based on bits [16:31] of the register at the end of a successful DMA transfer only.

Table 20: DMA Control Register (C\_BASEADDR + 0x0204)

Bit(s)	Name	Access	Reset Value	Description
31	Direction <sup>(1)</sup> <sup>(2)</sup>	R/W	0	'0' = Write data into DPRAM '1' = Read data from DPRAM
30	Endpoint Buffer Select <sup>(3)</sup>	R/W	0	'0' = Buffer Ready set by Firmware '1' = Buffer Ready set by DMA Controller
29-16	Reserved	NA	-	Reserved for future use
15	EP-7 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 7 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 7 will be available for IN or OUT transfer after the current DMA transfer
14	EP-6 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 6 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 6 will be available for IN or OUT transfer after the current DMA transfer

**Table 20: DMA Control Register ( $C\_BASEADDR + 0x0204$ ) (Cont'd)**

Bit(s)	Name	Access	Reset Value	Description
13	EP-5 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 5 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 5 will be available for IN or OUT transfer after the current DMA transfer
12	EP-4 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 4 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 4 will be available for IN or OUT transfer after the current DMA transfer
11	EP-3 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 3 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 3 will be available for IN or OUT transfer after the current DMA transfer
10	EP-2 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 2 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 2 will be available for IN or OUT transfer after the current DMA transfer
9	EP-1 2nd Buffer Ready	R/W	0	'0' = 2nd buffer of Endpoint 1 is not available for IN or OUT interrupt '1' = 2nd buffer of Endpoint 1 will be available for IN or OUT transfer after the current DMA transfer
8	Reserved	NA	-	Reserved
7	EP-7 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 7 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 7 will be available for IN or OUT transfer after the current DMA transfer
6	EP-6 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 6 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 6 will be available for IN or OUT transfer after the current DMA transfer
5	EP-5 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 5 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 5 will be available for IN or OUT transfer after the current DMA transfer
4	EP-4 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 4 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 4 will be available for IN or OUT transfer after the current DMA transfer
3	EP-3 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 3 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 3 will be available for IN or OUT transfer after the current DMA transfer
2	EP-2 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 2 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 2 will be available for IN or OUT transfer after the current DMA transfer

Table 20: DMA Control Register ( $C\_BASEADDR + 0x0204$ ) (Cont'd)

Bit(s)	Name	Access	Reset Value	Description
1	EP-1 1st Buffer Ready	R/W	0	'0' = 1st buffer of Endpoint 1 is not available for IN or OUT interrupt '1' = 1st buffer of Endpoint 1 will be available for IN or OUT transfer after the current DMA transfer
0	Reserved	NA	-	Reserved

**Notes:**

1. Write data into DPRAM means that the DMA controller writes data to DPRAM by reading the data from the external memory
2. Read data from DPRAM means that the DMA controller reads data from DPRAM and writes the data to the external memory
3. In DMA mode ( $C\_INCLUDE\_DMA = 1$ ), the Buffer Ready status can be updated by either Firmware through the BRR register or Hardware. The setting is controlled by Endpoint Buffer Select bit of the DMA Control register.

**DMA Source Address Register (DSAR)**

The DMA Source Address Register described in Table 21 defines the source address for the current DMA transfer. When data is moved from the source address, this register updates to track the current source address. If the Direction bit of the DMA Control Register (DMACR) is set to 0, the DMA Source Address of the current DMA transfer should be external memory address. If the Direction bit of the DMA Control Register (DMACR) is set to 1, the DMA Source Address of the current DMA transfer should be DPRAM address.

Table 21: DMA Source Address Register ( $C\_BASEADDR + 0x0208$ )

Bit(s)	Name	Access	Reset Value	Description
31-0	Source Address	R/W	0	Source Address for the current DMA transfer.

**DMA Destination Address Register (DDAR)**

The DMA Destination Address Register described in Table 22 defines the destination address for the current DMA transfer. When data is moved to the destination address, this register updates to track the current destination address. If Direction bit of the DMA Control Register (DMACR) is set to 0, the DMA Destination Address of the current DMA transfer should be DPRAM address. If Direction bit of the DMA Control Register (DMACR) is set to 1, the DMA Destination Address of the current DMA transfer should be external memory address.

Table 22: DMA Destination Address Register ( $C\_BASEADDR + 0x020C$ )

Bit(s)	Name	Access	Reset Value	Description
31-0	Destination Address	R/W	0	Destination Address for the current DMA transfer.

## DMA Length Register (DMALR)

The DMA Length register shown in [Table 23](#) defines the total number of bytes to be transferred from source address to destination address. This register is written only after configuring the [DMA Control Register \(DMACR\)](#), the DMA Source Address register, and the DMA Destination Address register, with their values and any other setup is complete. As bytes are successfully written to the destination, the [DMA Length Register \(DMALR\)](#) decrements to reflect the number of bytes remaining to be transferred. The [DMA Length Register \(DMALR\)](#) will be zero after a successful DMA operation.

Table 23: DMA Length Register ( $C\_BASEADDR + 0x0210$ )

Bit(s)	Name	Access	Reset Value	Description
31-0	DMA Length	R/W	0	Number of bytes to be transferred from source to destination

## DMA Status Register (DMASR)

The DMA Status Register (DMASR) described in [Table 24](#) defines the status of DMA controller as DMA Busy or DMA Error. When DMA operation is in progress, the DMA Busy will be set to 1 until the DMA operation has been finished. If the DMA operation encounters any error condition, the DMA Error will be set to 1

Table 24: DMA Status Register ( $C\_BASEADDR + 0x0214$ )

Bit(s)	Name	Access	Reset Value	Description
31	DMA Busy	R	0	'0' = DMA operation is not initiated '1' = DMA operation is in progress
30	DMA Error	R	0	'0' = DMA Error has not occurred '1' = DMA Error has occurred
29-0	Reserved	NA	-	Reserved for future use

## Programming the Core

This section describes how to program the AXI USB2 Device for various operations. For applications that use the Xilinx driver, users are expected to change the Vendor ID before using the AXI USB2 Device.

### Initialization Sequence

- At first, the AXI USB2 Device core issues ULPI\_reset to PHY during power-on reset
- Firmware enables the USB Disconnect, USB Suspend, and USB Reset bits of the [Interrupt Enable Register \(IER\)](#)
- Firmware enables SIE by setting the MASTER\_READY bit of the [Control Register \(CR\)](#)
- When the MASTER\_READY bit of the [Control Register \(CR\)](#) set to 1 by Firmware, the SIE configures the PHYs to device mode by writing OTG [Control Register \(CR\)](#) to 0x00, Function Control register to 0x41 (Sets TermSelect to 0b, XcvtSelect to 01b (HS) and, Opmode to 00b (Normal Operation))
- If AXI USB2 Device core was not connected to Host, the PHY updates Session End (VBUS not present) RXCMD to SIE. At that point, SIE waits for connection with Host
- When AXI USB2 Device core connects to Host, the PHY updates Vbus Valid (VBUS present) RXCMD to SIE
- SIE pulls up the D+ line of the PHY by writing 0x45 (Sets TermSelect to 1b, XcvtSelect to 01b (FS), by writing Opmode to 00b (Normal Operation)) to the Function Control register of the PHY, and moves to Full-speed mode. SIE updates the USB Disconnect bit of the [Interrupt Status Register \(ISR\)](#) to 1

- When PHY pulls up the D+ line, the host detects the AXI USB2 Device connection and starts issuing a USB Reset to bring the device into the default unconfigured state
- SIE detects USB Reset from the Host and updates the USB Reset bit of the [Interrupt Status Register \(ISR\)](#) to 1 until USB Reset signalling is finished between the Host and the AXI USB2 Device
- After the successful completion of the USB Reset by the Host, SIE updates the HSEn bit (0 - Full-speed, 1 - High-speed) and starts receiving SOFs every 1 ms in Full-speed and every 125  $\mu$ s in High-speed
- AXI USB2 Device responds to transfers from the Host based on the endpoint configuration and status registers programmed

## Operating in Interrupt Mode

If desired, configure the device to operate in the interrupt mode after enabling the desired interrupts in the [Interrupt Enable Register \(IER\)](#)

- Interrupts for USB Reset, USB Suspend, USB Resume (Remote-wakeup), and USB Disconnect can be enabled by writing to those specific bits of the IER
- To generate an interrupt when the core is 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, set the Fifo Buffer Ready, and Fifo Buffer Free bits of the IER to generate interrupts when packets are received or transmitted respectively
- For all other endpoints, set the respective Buffer Complete bit of the IER to generate interrupts when that endpoint buffer is complete

## 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

## Configuring an Endpoint

Program the individual Endpoint Configuration and 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 and status register
- The direction of an endpoint can be set to IN by writing a '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

## Handling a Control Packet

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

## Handling Bulk/Isochronous IN Transactions

For a Bulk or Isochronous IN transaction in No DMA Mode, perform the following steps:

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 \(BRR\)](#)
4. Wait for the Buffer Ready interrupt of the selected endpoint buffer in the [Buffer Ready Register \(BRR\)](#) (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

For a Bulk or Isochronous IN transaction in DMA-Mode, perform the following steps:

1. Write the packet count into the specific endpoint buffer's Buffer Count Register
2. Configure DMA by writing into Direction bit of [DMA Control Register \(DMACR\)](#) to 0, DMA Destination Address (same as EP\_BASE of the endpoint configuration register of the respective endpoint), DMA Destination Address and DMA Length registers
3. DMA generates DMA Done interrupt after successfully writing the configured length of data into the DPRAM
4. Set the Buffer Ready bit for the selected endpoint buffer in the [Buffer Ready Register \(BRR\)](#)
5. Wait for the Buffer Complete interrupt of the selected endpoint buffer in the [Interrupt Status Register \(ISR\)](#) 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 in No DMA Mode, perform the following steps:

1. Poll the Buffer Complete bit of the selected endpoint buffer in the [Interrupt Status Register \(ISR\)](#) 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

On reception of the OUT packet in DMA-Mode, perform the following steps:

1. Poll the Buffer Complete bit of the selected endpoint buffer in the [Interrupt Status Register \(ISR\)](#) to detect the reception of a packet
2. Check the received packet count matches with the specified packet count in the Buffer Count Register
3. Read the OUT packet by configure DMA by writing into Direction bit of [DMA Control Register \(DMACR\)](#) to 1, DMA Source Address (same as EP\_BASE of the endpoint configuration register of the respective endpoint), DMA Destination Address, and DMA Length registers
4. DMA generates DMA Done interrupt after successfully reading the configured length of data into the DPRAM
5. Set the Buffer Ready bits of the selected endpoint buffer to prepare it for the next transaction

## USB Reset Signalling

When the host wants to start communicating with a device it will start by applying a 'Reset' condition which sets the device to its default unconfigured state. This 'Reset' should not be confused with a micro-controller power-on type reset. It is a USB protocol reset to ensure that the device USB signalling starts from a known state. High-speed capable AXI USB2 Device can be reset while the device in the Powered, Default, Address, Configured or Suspended states. AXI USB2 Device can be successfully reset by any host (even USB1.x host).

The Host performs USB Reset Signalling High-speed capable AXI USB2 Device as follows

1. Host drives SE0, the start of SE0 is referred to as time T0.
2. The SIE detects assertion of SE0.
  - a. If SIE detects SE0 from USB Suspend state, then SIE begins the high-speed handshake detection after the detection of SE0 for no less than 2.5 us
  - b. If SIE detects SE0 from a non-suspended full-speed mode, the SIE begins a high-speed handshake detection after the detection of SE0 for no less than 2.5 us and no more than 3.0 ms
  - c. If SIE detects SE0 from a non-suspended high-speed mode, the SIE wait for 3.0 ms before reverting to Full-speed mode. After reverting to Full-speed, SIE checks the line state update from PHY for SE0 (to check whether Host issued suspend or USB Reset), no less than 1 ms from the point SIE reverted to Full-speed. If SIE detects SE0 by the RXCMD updating from PHY, SIE begins the high-speed handshake detection.
3. Because the AXI USB2 Device is high-speed capable device, SIE follows High-speed Handshake Detection protocol.

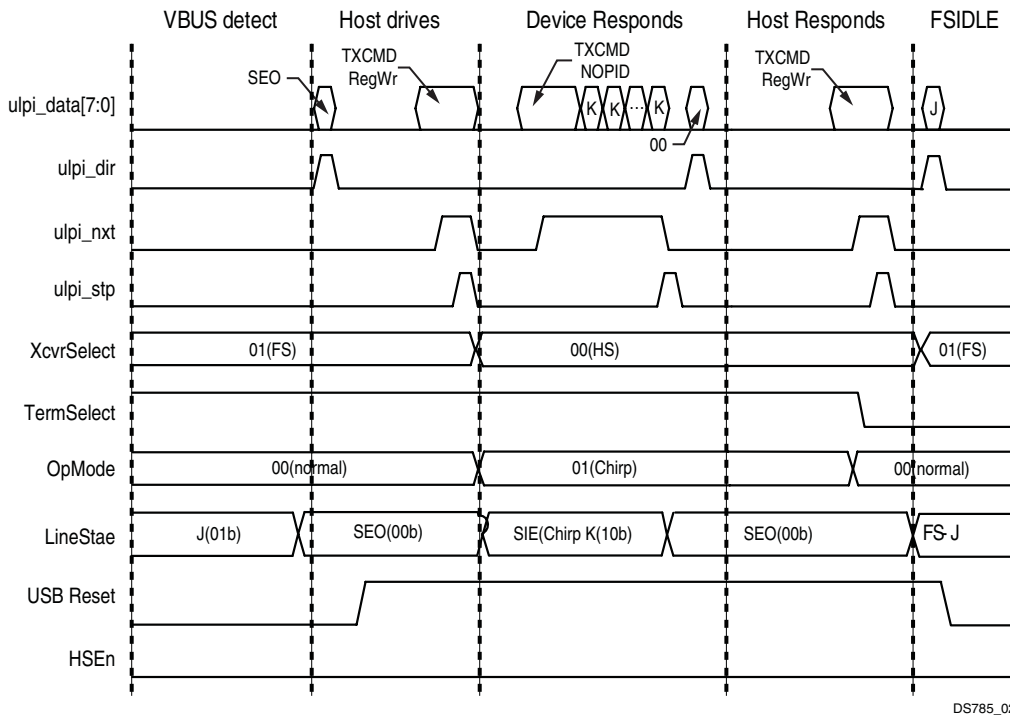


Figure 2: USB Reset Signalling with USB1.x Host - Full Speed

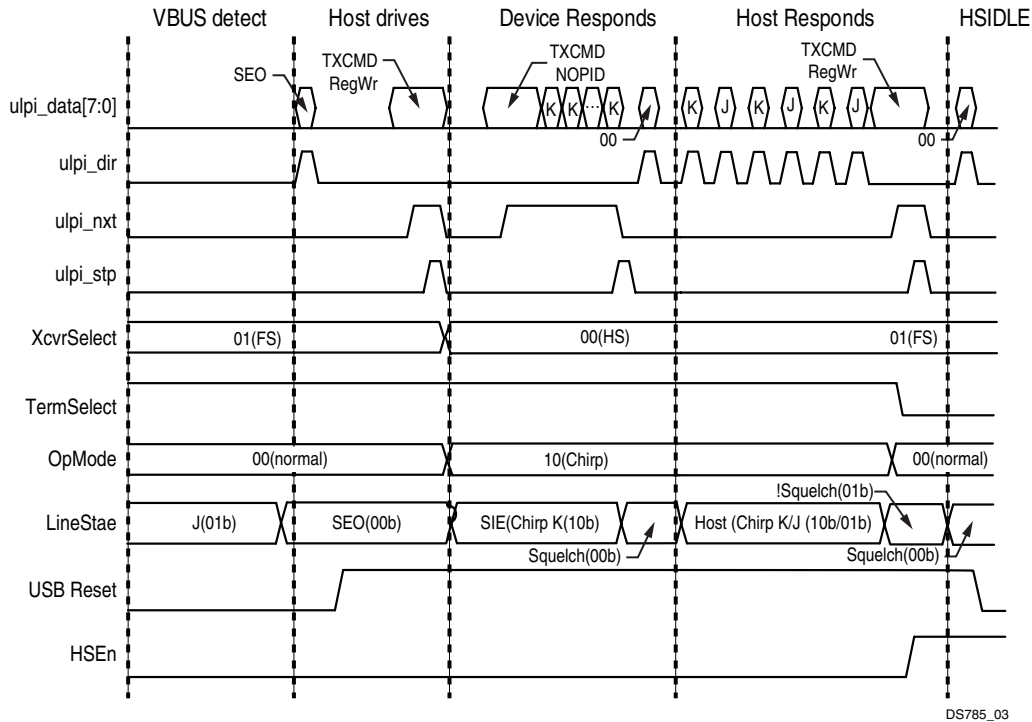


Figure 3: USB Reset Signalling with USB2.0 Host - High Speed Handshaking

### High-speed Handshake Detection Protocol

1. SIE writes 0x54 (sets XcvrSelect to 00b, TermSelect to 1b and Opmode to 10b (Chirping)) to the Function Control register of the PHY
2. Immediately after the above write, SIE puts TXCMD as 0x40 (NOPID) and starts transmitting a chirpK for 2 ms
3. After 2 ms of ChirpK transmission, PHY updates RXCMD with line state as SEO
4. SIE detects SEO and waits for 100 us to look for high-speed Host chirp, for example, an alternating sequence of ChirpKs and ChirpJs
5. If SIE does not observe any high-speed Host chirp, the SIE configures PHY to Full-speed mode by writing 0x45 (sets XcvrSelect to 01b, TermSelect to 1b, and Opmode to 00b (Normal Operation)) to the Function Control register of PHY. At this point, SIE is in Full-speed mode and waits for Full-speed USB traffic from Host.
6. If SIE observes the high-speed Host chirp, the SIE checks for a minimum of chirp K-J-K-J-K-J
7. SIE also checks each individual chirpK and chirpJ for at least 2.5 us
8. After detecting the minimum sequence, SIE configures PHY to high-speed mode by writing 0x40 (Sets TermSelect to 0b, XcvrSelect to 00b (HS) and Opmode to 00b (Normal Operation)) to the Function Control register of PHY and updates the HSEn bit of the [Interrupt Status Register \(ISR\)](#) to 1.
9. At this point, SIE is in High-speed mode and sees line-state as !squelch (linestate != 00) on the RXCMD update.
10. If SIE observes squelch as linestate update from PHY (for example, RXCMD[1:0]), the SIE treats that Host has completed the chirp and stops updating the status as USB Reset in the [Interrupt Status Register \(ISR\)](#).
11. Now SIE waits for High-speed USB traffic from Host.

## Suspend Signalling

When the Host does not want to continue any further communication with the connected device, it keeps the AXI USB2 Device in the suspend mode by ceasing all of its transmissions (including SOFs). The AXI USB2 Device recognizes the suspend condition in High-speed mode as follows:

1. Initially, Host and the AXI USB2 Device are either in Full-speed or High-speed mode.
2. When SIE sees no bus activity for 3 ms, it enters Full-speed mode by writing 0x45 (Sets TermSelect to 1b, XcvtSelect to 01b (FS), and Opmode to 00b (Normal Operation)) to the Function Control register.
3. After 1 ms, SIE samples the line state from the RXCMD update of PHY.
4. If the line state is Full-speed J, SIE enters into the Suspend state after 7 ms and updates the status as suspended by asserting the USB Suspend bit of the [Interrupt Status Register \(ISR\)](#).
5. SIE is now SIE in the Suspended state.
6. Resume or USB Reset can bring SIE out from suspend state.

## Resume Signalling

When AXI USB2 Device in the suspend state, host wakeup the device with a resume signalling. The AXI USB2 Device detects Resume signalling as listed in the steps below:

1. When both Host and AXI USB2 Device are in Low Power Mode
2. If SIE observes Full-speed K as line state (i.e RXCMD[1:0]) update from PHY then SIE treat it as Resume signalling from PHY and update the status as Resume by setting USB Resume bit of the [Interrupt Status Register \(ISR\)](#).
3. SIE wait to detect the End of Resume i.e SE0 as line state update from PHY
4. If SIE detects SE0 linestate update from PHY:
  - a. SIE moves to Full-speed if SIE was in Full-speed prior to entering into suspend state
  - b. SIE moves to High-speed by writing 0x40 (Sets TermSelect to 0b, XcvtSelect to 00b (HS), and Opmode to 00b (Normal Operation)) to the Function Control register of the PHY if SIE was in High-speed prior to entering into the suspend state
5. SIE updates the HSEn bit of [Interrupt Status Register \(ISR\)](#) accordingly.

The Suspend signalling followed by Resuming signalling from Host is shown in [Figure 4](#) (Full-speed) and [Figure 5](#) (High-speed).

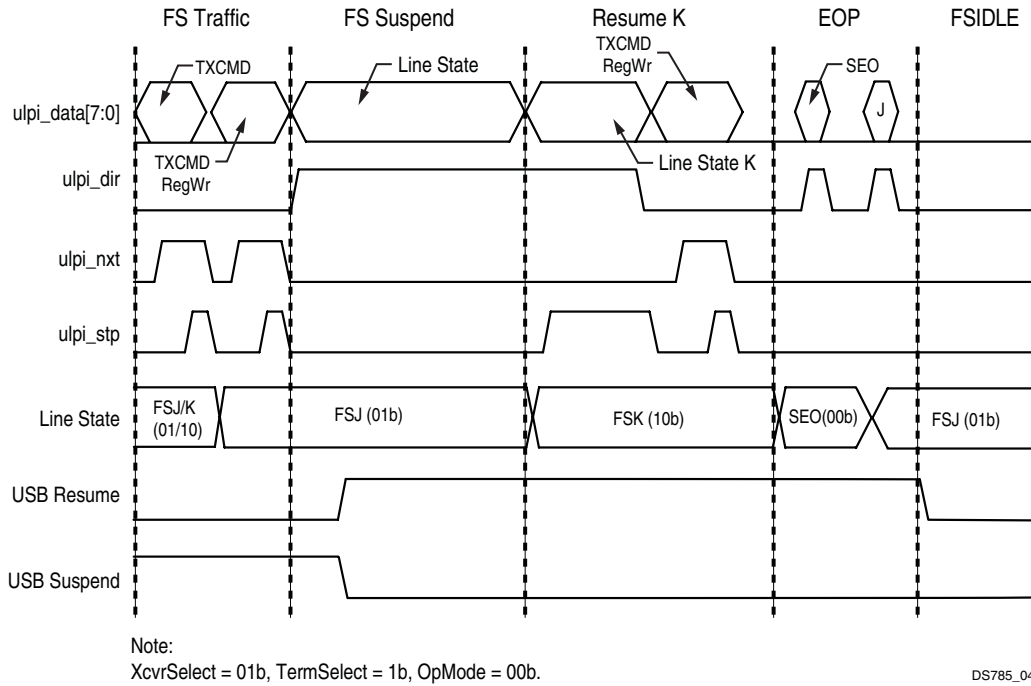


Figure 4: Suspend Signalling Followed with Resume Signalling from Host - Full Speed

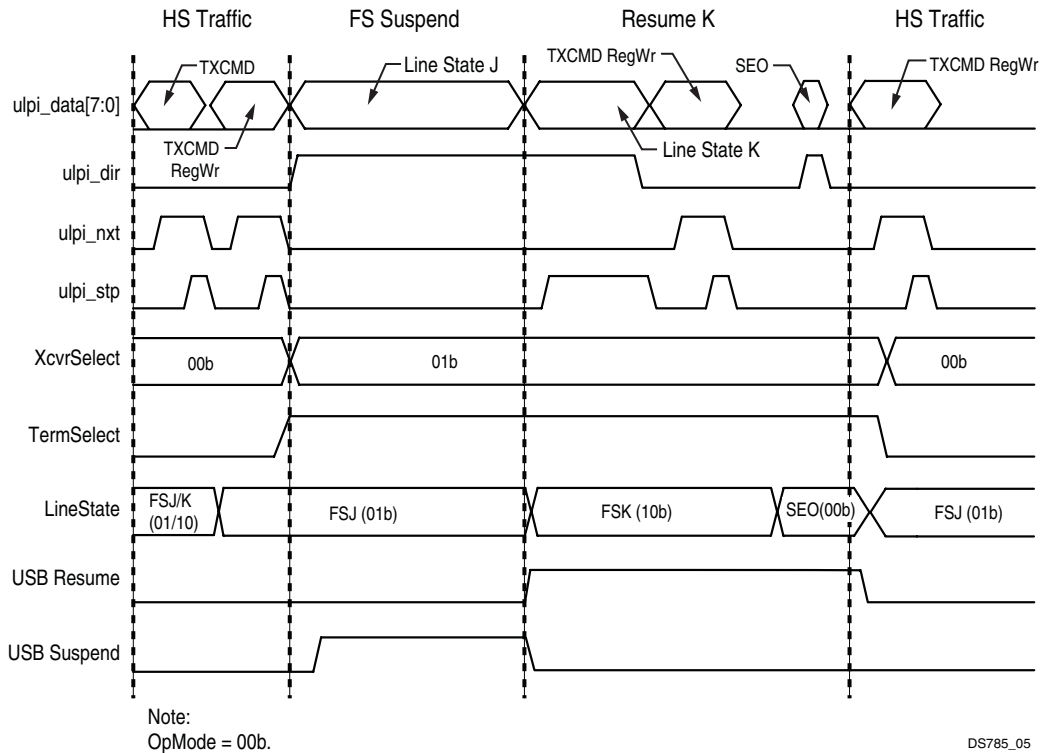


Figure 5: Suspend Signalling Followed with Resume Signalling from Host - High Speed

## Remote Wakeup Signalling

When AXI USB2 Device is suspended by the host, it supports the Remote Wakeup feature and initiates a resume itself. The AXI USB2 Device initiates the Remote Wakeup signalling as follows. The Suspend signalling is followed by Remote wakeup signalling from the AXI USB2 Device, followed by a Resume signalling from the Host as shown in [Figure 6](#) (Full-speed) and [Figure 7](#) (High-speed):

1. When both the Host and the AXI USB2 Device are in Low Power Mode.
2. If REMOTE\_WAKEUP bit of the [Control Register \(CR\)](#) is set to 1, then SIE begins Remote Wake-up signalling by writing 0x54 (sets XcvrSelect to 00b, TermSelect to 1b, and Opmode to 10b (Chirping)) to the Function Control register of the PHY.
3. Immediately after the above write, SIE puts TXCMD as 0x40 (NOPID) following a Full-speed K for 3 ms and updates the status as Resume by setting the USB Resume bit of the [Interrupt Status Register \(ISR\)](#).
4. The Host takes over driving the Resume K within 1 ms after detecting the Remote Wakeup from SIE.
5. SIE wait to detect the End of Resume, for example, SE0 as line state update from PHY.
6. If SIE detects SE0 linestate update from PHY,
  - a. SIE moves to Full-speed by writing 0x45 (Sets TermSelect to 1b, XcvrSelect to 01b (FS), and Opmode to 00b (Normal Operation)) to the Function Control register of the PHY, if SIE was in Full-speed prior to entering into the suspend state.
  - b. SIE moves to High-speed by writing 0x40 (Sets TermSelect to 0b, XcvrSelect to 00b (HS) and Opmode to 00b (Normal Operation)) to the Function Control register of the PHY, if SIE was in High-speed prior to entering into the suspend state.
7. SIE updates the HSEn bit of [Interrupt Status Register \(ISR\)](#) accordingly.

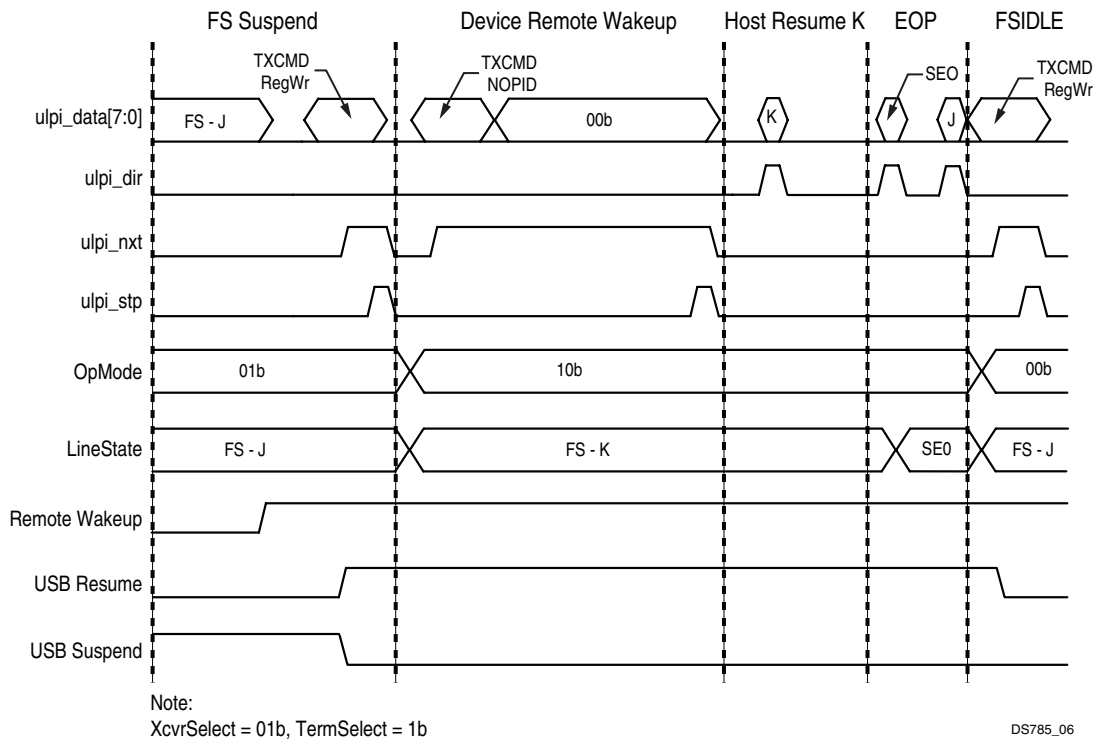


Figure 6: Suspend Signalling Followed by Remote Wakeup Signalling from SIE - Full Speed

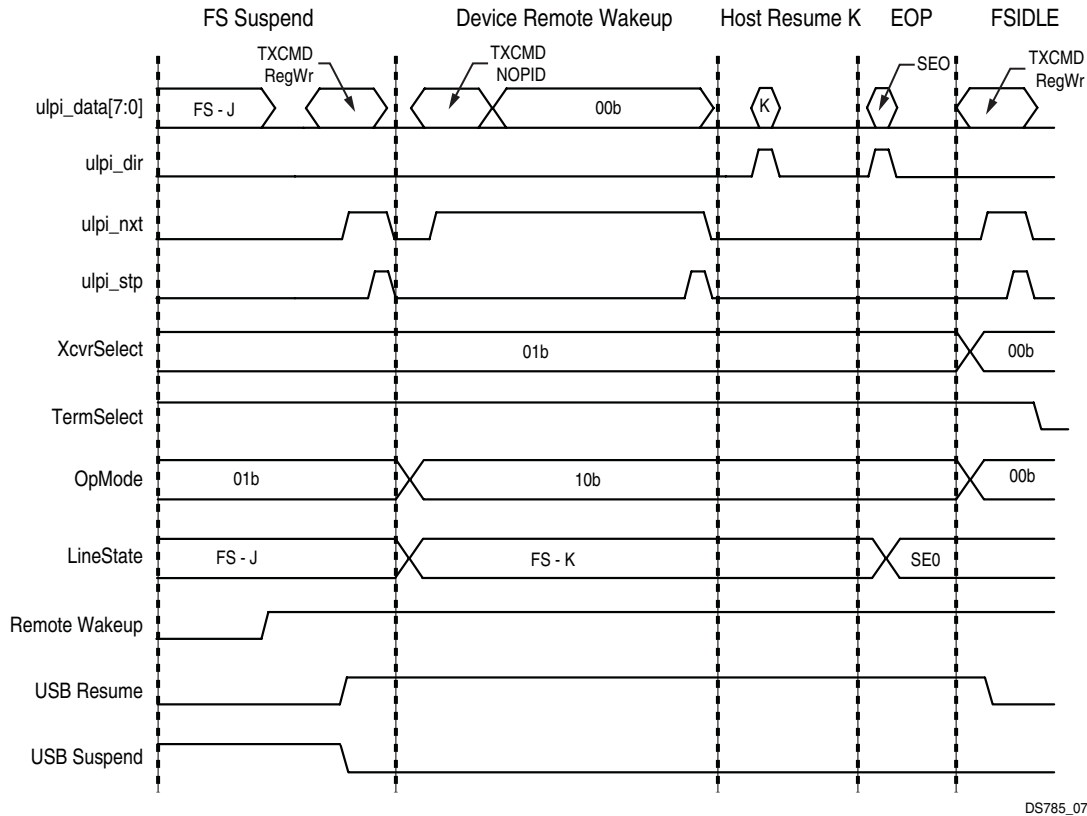


Figure 7: Suspend Signalling Followed by Remote Wakeup Signalling from SIE - High Speed

## 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 \(TMR\)](#) are set to '0'. To put the core in Test Mode operation, program the bits [2:0] of the [Test Mode Register \(TMR\)](#) for different test modes:

- To put the core in Test mode J, program TMR[2:0] = "001". In this mode, Chirp J sequences must be seen on the bus.
- To put the core in Test mode K, program TMR[2:0] = "010". In this mode, Chirp K sequences must be seen on the bus.
- To put the core in Test mode NAK, program TMR[2:0] = "011". In this mode, NAK must be seen on the bus.
- To put the core in Test mode packet, program TMR[2:0] = "100". In this mode, the test packet specified by the USB 2.0 Specification must be seen on the bus.
- For endpoint 0, set the Fifo Buffer Ready, and Fifo Buffer Free bits of the IER to generate interrupts when packets are received or transmitted respectively
- For all other endpoints, set the respective Buffer Complete bit of the IER to generate interrupts when that endpoint buffer is complete

## 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 AXI USB2 Device uses two clocks:

- **S\_AXI\_ACLK:** The AXI Bus interface, Port B of the DPRAM, and the control and status registers use this clock. The minimum S\_AXI\_ACLK 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 AXI USB2 Device core is reset using the S\_AXI\_ARESETN signal which resets all devices that are connected to the AXI 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.

## 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 two different clock domains: S\_AXI\_ACLK and ULPI\_Clock. A timing ignore constraint should be added to isolate these two clock domains. The constraints listed below can be used with the AXI USB2 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%;
```

##### *S\_AXI\_ACLK*

The clock provided to S\_AXI\_ACLK must be constrained for a clock frequency of 60 MHz - 200 MHz.

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

## ULPI Interface constraints

```
# Set the OFFSET IN delay as 9.00 ns with respect to ULPI_CLK
OFFSET = IN 8.50 ns VALID 11.50 ns BEFORE ULPI_CLK RISING;
# Set the OFFSET OUT delay as 10.50 ns with respect to ULPI_CLK
OFFSET = OUT 10.50 ns AFTER ULPI_CLK RISING;
```

## Design Implementation

### Target Technology

The target technology is an FPGA listed in the Supported Device Families field of the [LogiCORE™ IP Facts Table](#). The FPGA device used must have the following attributes:

- Large enough to accommodate the core.
- Contain a sufficient number of IOBs/BRAM.

### Device Utilization and Performance Benchmarks

Because the AXI USB2 Device core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the AXI USB2 Device core will vary from the results reported here.

The AXI USB2 Device core resource utilization for various parameter combinations measured with the Virtex-6 FPGA as the target device are detailed in [Table 25](#).

*Table 25: Performance and Resource Utilization Benchmarks on the Virtex-6 FPGA (xc6vlx130t-1-ff1156)*

Parameter Values		Device Resources			Performance
C_INCLUDE_DMA	C_INCLUDE_USBERR_LOGIC	Slices	Slice Flip-Flops	LUTs	f <sub>Max</sub> (MHz)
0	0	733	647	2,037	160
0	1	667	685	2,097	160
1	0	966	1,231	2,613	160
1	1	1,034	1,267	2,699	160

The AXI USB2 Device core resource utilization for various parameter combinations measured with the Spartan-6 FPGA as the target device are detailed in [Table 26](#).

**Table 26: Performance and Resource Utilization Benchmarks on the Spartan-6 FPGA (xc6slx150t-2-fgg900)**

Parameter Values		Device Resources			Performance
		Slices	Slice Flip-Flops	LUTs	$f_{Max}$ (MHz)
0	0	664	814	2,031	110
0	1	662	846	2,003	110
1	0	949	1,606	2,655	110
1	1	962	1,643	2,666	110

## System Performance

To measure the system performance ( $F_{MAX}$ ) of the AXI USB2 Device core, it was added as the Device Under Test (DUT) to a Virtex-6 FPGA system as shown in [Figure 8](#), and a Spartan-6 FPGA system as shown in [Figure 9](#).

Because the AXI USB2 Device core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When the AXI USB2 Device core is combined with other designs in the system, the utilization of FPGA resources and timing will vary from the results reported here.

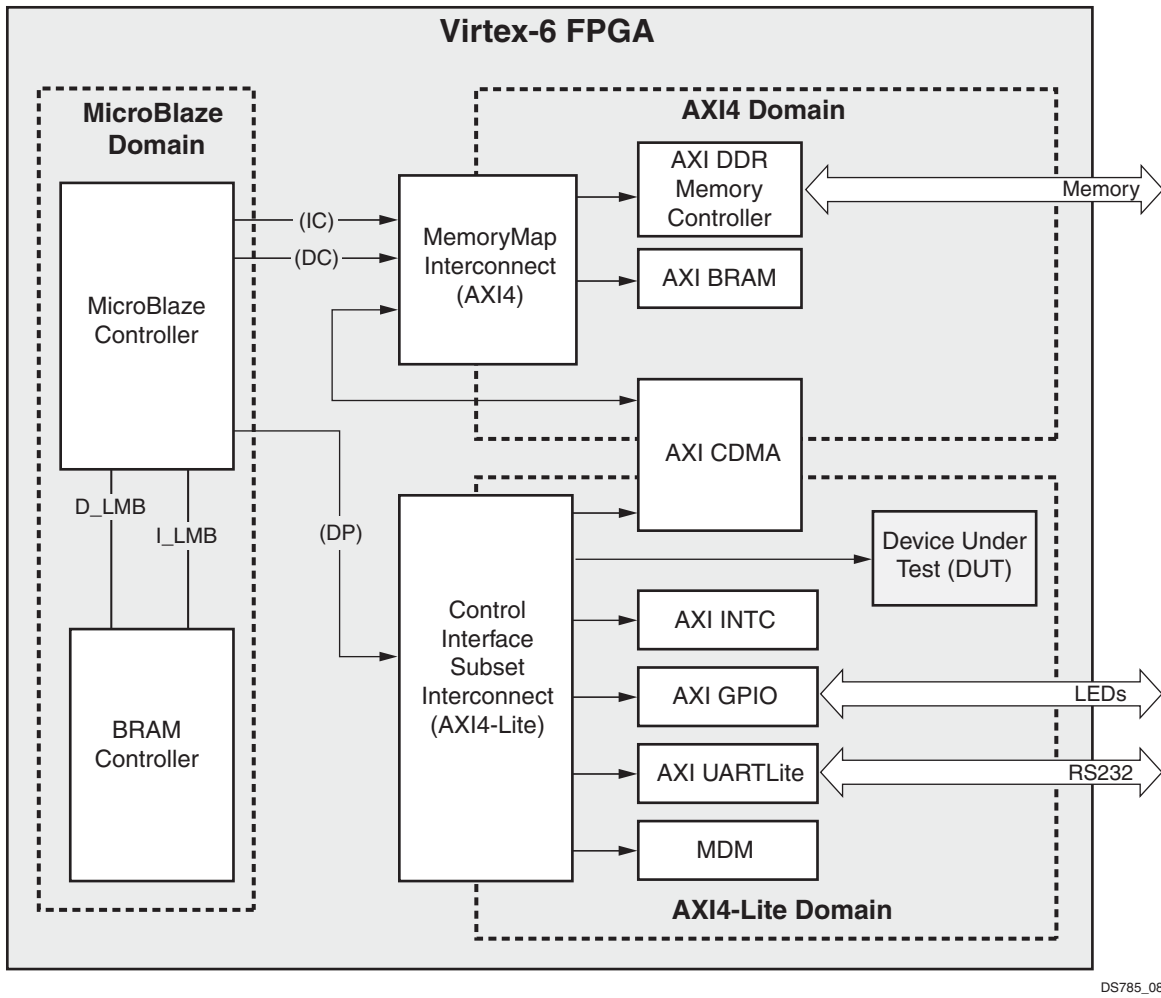
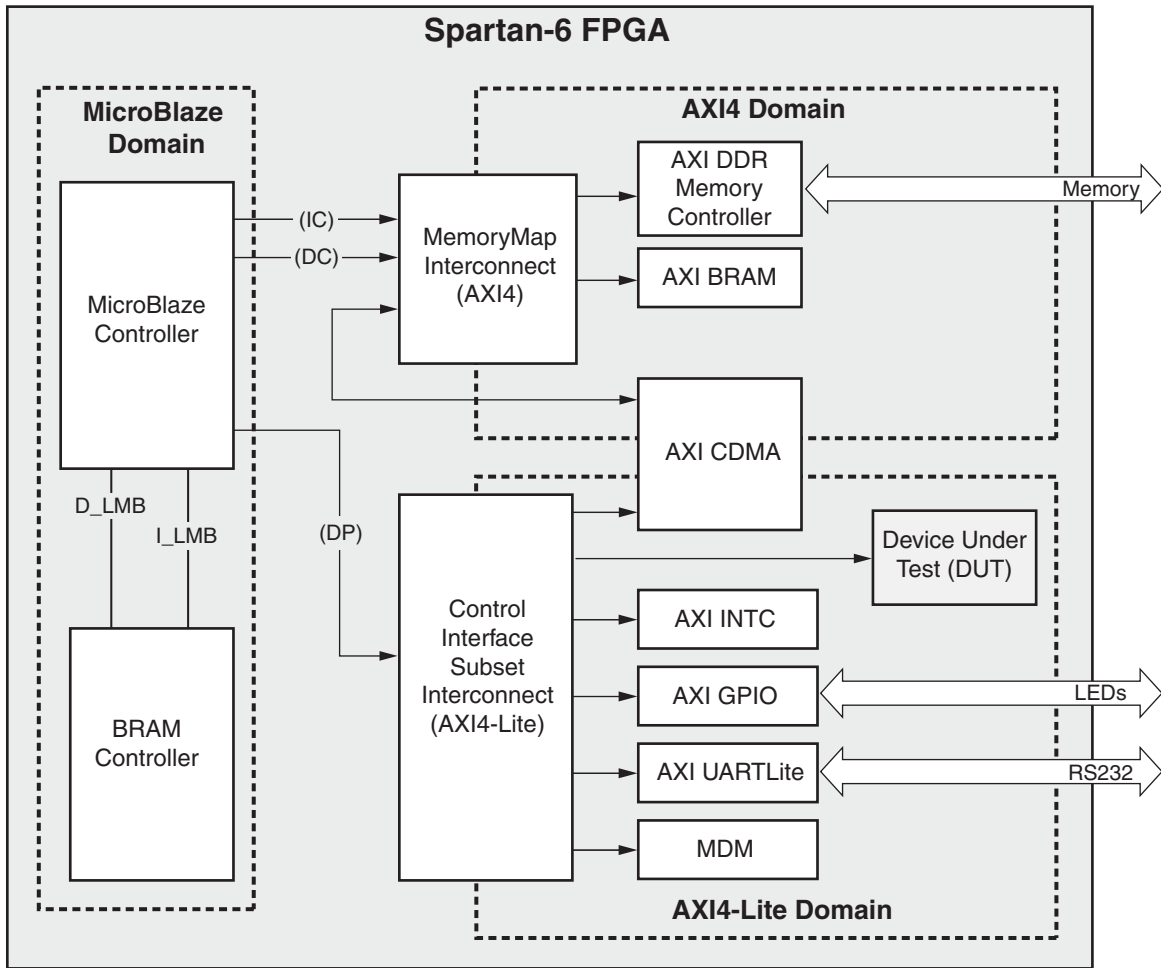


Figure 8: Virtex-6 FPGA System with the AXI USB2 Device as the DUT



DS785\_09

Figure 9: Spartan-6 FPGA System with the AXI USB2 Device as the DUT

The target FPGA was then filled with logic to drive the LUT and BRAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target  $F_{MAX}$  numbers are shown in Table 27.

Table 27: AXI USB2 Device System Performance

Target FPGA	Estimated $F_{MAX}$ (MHz)
S6LX45t -2	110
V6LX130t -1	180

The target  $F_{MAX}$  is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

## Throughput

To measure the system throughput of the AXI USB2 Device core, it was used on a Virtex-6 system similar to [Figure 8](#) and tested on a Xilinx ML605 board.

For this test, the firmware running on the ML605 evaluation board behaves as a mass storage device. A Windows application running on a Windows Desktop PC set up the mass storage device for the throughput test on its BULK IN endpoint. The windows application allowed an option of selecting the amount of traffic being sent on the bus for the throughput test. The selection options were 1MB/10MB/1000MB. Once the option was selected, the Windows application set up the mass storage device to send the selected amount of data. Upon completion of the data transfer, the result was displayed on the Windows screen in mega bytes per second (MBps). The throughput was measured using the amount of data sent over the bus and the time taken for transmission. The LeCroy USB analyzer was used to capture the transactions on the bus. The throughput results (See [Table 28.](#)) were further provided based on the number of USB data packets received between two SOF packets.

The throughput numbers are shown in the [Table 28](#)

**Table 28: AXI Central DMA System Performance**

Throughput in the Functional Simulations	Throughput on the ML Board
12 packets per micro frame = 49,152,000 bytes per second	11 packets per micro frame = 45,056,000 bytes per second

**Note:** In BULK IN mode, the performance of the device depends on the number of data transfer packets that the Host application can initiate between the two SOFs.

## Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

## Ordering Information

This Xilinx LogiCORE IP module is provided under the terms of the [Xilinx Core Site License](#). The core is generated using the Xilinx ISE Embedded Edition software (EDK). For full access to all core functionality in simulation and in hardware, you must purchase a license for the core. Please contact your local Xilinx sales representative for information on pricing and availability of Xilinx LogiCORE IP.

For more information, please visit the [AXI USB2 Device](#) product web page.

Information about this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information on pricing and availability of other Xilinx LogiCORE modules and software, please contact your [local Xilinx sales representative](#).

This Embedded IP module is provided under the terms of the [Xilinx Core Site License](#). A free evaluation version of the core is included with the ISE Design Suite Embedded Edition software. Use the Xilinx Platform Studio application(XPS) included with the Embedded Edition software to instantiate and use this core.

For full access to all core functionality in simulation and in hardware, you must purchase a license for the core. Please contact your local Xilinx sales representative for information on pricing and availability of Xilinx LogiCORE IP.

## Reference Documents

1. Advance Micro controller Bus Architecture Advanced eXtensible Interface Specification (v2.0).
2. Universal Serial Bus Specification, Revision 2.0
3. UTMI+ Low Pin Interface (ULPI) Specification, Revision 1.1

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
9/21/10	1.0	Xilinx Initial Release

## Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.