



USB 2.0/3.2 IP

IP Version: v3.2.0

User Guide

FPGA-IPUG-02237-1.5

October 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language [FAQ 6878](#) for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Contents.....	3
Abbreviations in This Document.....	8
1. Introduction	9
1.1. Overview of the IP	9
1.2. Quick Facts	9
1.3. IP Support Summary.....	9
1.4. Features.....	9
1.5. Licensing and Ordering Information.....	9
1.6. Hardware Support	9
1.7. Minimum Device Requirements.....	9
1.8. Naming Conventions	10
1.8.1. Nomenclature.....	10
1.8.2. Signal Names	10
2. Functional Description.....	11
2.1. IP Architecture Overview	11
2.2. Clocking	12
2.2.1. Clocking Overview	12
2.1 User Interfaces	12
2.2 Internal IP Blocks.....	12
2.2.1 USB 2.0/3.2.....	12
2.2.2 AHBL-to-LMMI Converter.....	12
3. IP Parameter Description.....	13
4. Signal Description	14
4.1. Clock Interface.....	14
4.2. Reset Interface	14
4.3. USB Interface.....	14
4.4. LMMI/AHBL Interface – Configuration Path.....	14
4.5. AXI Interface – Data Path	15
4.6. Other Signals	18
5. Register Description.....	19
5.1. USB IP Controller Register Information.....	19
5.1.1 Global Core Control Register (GCTL)	20
5.1.2 Global ID Register (GSNPSID).....	25
5.1.3 Global USB2 PHY Configuration Register (GUSB2PHYCFG(#n)).....	25
5.1.4 Global USB 3.0 PIPE Control Register (GUSB3PIPECTL(#n)).....	30
5.1.5 Global Event Buffer Address Register (GEVNTADDR(n))	33
5.1.6 Global Event Buffer Address (High) Register (GEVNTADRHI(#n)).....	33
5.1.7 Global Event Buffer Size Register (GEVNTSIZ(#n)).....	33
5.1.8 Global Event Buffer Count Register (GEVNTCOUNT(#n))	34
5.1.9 Device Configuration Register (DCFG).....	34
5.1.1. Device Control Register (DCTL).....	36
5.1.10 Device Event Enable Register (DEVTEN)	41
5.1.11 Device Status Register (DSTS).....	42
5.1.12 Device Active USB Endpoint Enable Register (DALEPENA).....	44
5.1.13 Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2)	45
5.1.14 Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)	45
5.1.15 Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)	45
5.1.16 Device Physical Endpoint-n Command Register (DEPCMD)	45
5.1.17 GLOBAL_TX_FIFO_SIZE (GTXFIFOSIZ(#n)).....	46
5.1.18 GLOBAL_RX_FIFO_SIZE (GRXFIFOSIZ(#n))	47
5.2. USB IP 2/3 PHY Control Status Register Information	47

6. Operation Details	54
6.1. Enumeration Sequence	54
6.1.1. Power-On or Soft Reset	54
6.1.2. USB Reset	54
6.1.3. Connect Done	54
6.1.4. SetAddress Device Request	55
6.2. Transfer Flow	55
6.2.1. Control-data/Bulk/Interrupt OUT Transfer Flow	55
6.2.2. Isochronous OUT Transfer Flow	56
6.2.3. Control-data/ Bulk/Interrupt IN Transfer Flow	58
6.2.4. Isochronous IN Transfer Flow	61
7. Designing with the IP	63
7.1. Generating and Instantiating the IP	63
7.1.1. Generated Files and File Structure	65
7.1.2. Design Implementation	66
7.1.3. Timing Constraints	66
8. Example Design	67
8.1. Overview of the Example Design and Features	67
8.2. Example Design Components	67
8.2.1. Description on Data Flow from USB Device to Host	70
8.2.2. Description on Data Flow from USB Host to Device	71
8.2.3. USB 2.0/3.2 Primitive	71
8.2.4. USB23_ED	71
8.3. Running the Example Design	85
8.3.1. Generating the USB23 Benchmark Example Design	86
8.3.2. HAL Driver Compilation	86
8.3.3. Radiant Software Compilation for Bitstream	101
8.3.4. Trim the Memory File	105
8.3.5. Merging .mem with Bitstream File	106
8.3.6. Hardware Testing	108
8.3.7. FPGA Bitstream Programming	108
8.3.8. Performing Streaming Operations	112
8.4. Example Software API	128
8.4.1. USB Enumeration API	128
Appendix A. Resource Utilization	129
References	130
Technical Support Assistance	131
Revision History	132

Figures

Figure 2.1. USB 2.0/3.2 IP Block Diagram	11
Figure 2.2. USB 2.0/3.2 IP Clock Domain Block Diagram	12
Figure 6.1. Device Software Sets Up Data Buffers Before Host Request for Control-data/Bulk/Interrupt Transfer	55
Figure 6.2. Device Software Sets Up Data Buffers After Host Requests for Control-data/Bulk/Interrupt Transfer	56
Figure 6.3. Isochronous OUT Transfer when Device Software Sets Up Data Buffers Before Host Request	57
Figure 6.4. Isochronous OUT Transfer when Device Software Sets Up Data Buffers after Host Request	58
Figure 6.5. Control-data/Bulk/Interrupt IN Transfer when Device Software Sets Up Data Buffers before Host Request	59
Figure 6.6. Control-data/Bulk/Interrupt IN Transfer when Device Software Sets Up Data Buffers after Host Request	60
Figure 6.7. Isochronous IN Transfer when Device Software Sets Up Data Buffers before Host Request	61
Figure 6.8. Isochronous IN Transfer when Device Software Sets Up Data Buffers after Host Request	62
Figure 7.1. Module/IP Block Wizard	63

Figure 7.2. IP Configuration	64
Figure 7.3. Check Generated Result.....	65
Figure 8.1. USB2.0/3.2 IP Example Design Block Diagram	68
Figure 8.2. Clocking Diagram of the Example Design	69
Figure 8.3. IN Data Flow from USB Device to Host	70
Figure 8.4. OUT Data Flow from USB Host to Device.....	71
Figure 8.5. PLL Configuration	72
Figure 8.6. PLL Optional Port Configuration	73
Figure 8.7. RISC-V Configuration.....	74
Figure 8.8. UART Configuration	75
Figure 8.9. Hardware Version GPIO Configuration	76
Figure 8.10. Remote Wakeup GPIO Configuration	77
Figure 8.11. USB2 PHY Reset GPIO Configuration	78
Figure 8.12. System Memory Configuration	79
Figure 8.13. System Memory Port S0 Configuration	79
Figure 8.14. System Memory Port S1 Configuration	80
Figure 8.15. AHBL Interconnect General Configuration	80
Figure 8.16. AHBL Interconnect Main Configuration.....	81
Figure 8.17. AHBL Interconnect Manager Priority Settings	81
Figure 8.18. AHBL Interconnect Max Burst Settings	82
Figure 8.19. APB Interconnect Configuration 1	82
Figure 8.20. APB Interconnect Configuration 2	83
Figure 8.21. AHB-Lite to APB Configuration.....	83
Figure 8.22. Lattice CrosslinkU-NX33 Evaluation Board (Rev B)	85
Figure 8.23. nx33u_cmp_revb_evn_bd_rd_des.zip File	86
Figure 8.24. Create a Propel Project	86
Figure 8.25. Regenerate the Sub-Modules in the Propel Software	87
Figure 8.26. Generate Button in the Propel Software	87
Figure 8.27. Create a Project	88
Figure 8.28. Toolchain Setting	89
Figure 8.29. Copy all C and H Files to the src Folder	90
Figure 8.30. Overwrite File	90
Figure 8.31. Set Build Configuration to Release Mode	91
Figure 8.32. Set Optimization Level	92
Figure 8.33. Linker ID File.....	93
Figure 8.34. Linker ID Tab	93
Figure 8.35. Base Address Update	94
Figure 8.36. System Memory Instance	95
Figure 8.37. utils.h File.....	96
Figure 8.38. CPU_M_Frequency Update	97
Figure 8.39. Build a Project	97
Figure 8.40. Console Message	98
Figure 8.41. Import Project.....	98
Figure 8.42. Project from Folder or Archive	99
Figure 8.43. Select the Project Folder from Directory	99
Figure 8.44 Project Exploration Tab.....	100
Figure 8.45 Build Project.....	100
Figure 8.46 Console Message	100
Figure 8.47 Eval Folder.....	101
Figure 8.48 nx33u_cmp_revb_evn_bd_rd_des Sub Folder	101
Figure 8.49. Radiant Device Setup	101
Figure 8.50. USB IP GUI Option.....	102
Figure 8.51. Open a Radiant Software Project.....	103
Figure 8.52. USB23 Benchmark Example Design Radiant Software Project	103

Figure 8.53. Bitstream Generation	104
Figure 8.54. Place & Route Design Settings	104
Figure 8.55. Placement Iteration	105
Figure 8.56. Trim the Memory File	106
Figure 8.57. ECO Editor	106
Figure 8.58. Memory Initialization	107
Figure 8.59. Memory Initialization Settings	107
Figure 8.60. Save the Settings in the ECO Editor and Generate Bitstream	108
Figure 8.61. Radiant Programmer Getting Started Window	109
Figure 8.62. Program Button	109
Figure 8.63. Select Programming Mode	110
Figure 8.64. SPI Flash Programmer Settings	111
Figure 8.65. USB Enumeration	112
Figure 8.66. USB Device Tree Viewer	113
Figure 8.67. WinUSB Asynchronous CLI Application	113
Figure 8.68. Obtaining Device Information through WinUSB API	114
Figure 8.69. Bulk IN Test	115
Figure 8.70. Bulk OUT Test	116
Figure 8.71. Interrupt IN Test	117
Figure 8.72. Interrupt OUT Test	118
Figure 8.73. Isochronous IN Test	119
Figure 8.74. Isochronous OUT Test	120
Figure 8.75. USB2 Connection Speed	121
Figure 8.76. USB2 Bulk IN Test Result	122
Figure 8.77. USB2 Bulk OUT Test Result	123
Figure 8.78. USB2 Interrupt IN Test Result	124
Figure 8.79. USB2 Interrupt OUT Test	125
Figure 8.80. USB2 Isochronous IN Test Result	126
Figure 8.81. USB2 Isochronous OUT Test Result	127
Figure 8.82. HAL Driver API Document	128

Tables

Table 1.1. Summary of the USB2.0/3.2 IP Core	9
Table 1.2. USB2.0/3.2 IP Support Readiness	9
Table 1.3. Minimum Device Requirements for the USB 2.0/3.2 IP Core	10
Table 3.1. General Attributes of the USB 2.0/3.2 IP Core	13
Table 4.1. Clock Ports	14
Table 4.2. Reset Interface	14
Table 4.3. USB Interface	14
Table 4.4. LMMI Interface - Configuration Path	14
Table 4.5. AHBL Interface - Configuration Path	15
Table 4.6. AXI Interface - Data Path	15
Table 4.7. Other Signals	18
Table 5.1. USB IP Controller Register Information	19
Table 5.2. GLOBAL_CTL	20
Table 5.3. GLOBAL_ID	25
Table 5.4. GLOBAL_USB2_PHY_CFG(#n) (for n = 0)	25
Table 5.5. GLOBAL_USB3_PIPE_CTL(#n) (for n = 0)	30
Table 5.6. GLOBAL_EVNT_ADR(#n) (for n = 0)	33
Table 5.7. GLOBAL_EVNT_ADRI(#n) (for n=0)	33

Table 5.8. GLOBAL_EVNT_SIZ(#n) (for n=0)	33
Table 5.9. GLOBAL_EVNT_COUNT(#n) (for n=0)	34
Table 5.10. DEVICE_CFG (DCFG)	34
Table 5.11. DEVICE_CTL (DCTL)	36
Table 5.12. DEVICE_EVNT_EN	41
Table 5.13. DEVICE_STS(DSTS)	42
Table 5.14. DEVICE_ACT_EP_EN (DALEPENA)	44
Table 5.15. DEPCMDPAR2[0:7]	45
Table 5.16. DEPCMDPAR1[0:7]	45
Table 5.17. DEPCMDPAR0[0:7]	45
Table 5.18. DEPCMD[0:7]	45
Table 5.19. GLOBAL_TX_FIFO_SIZE	46
Table 5.20. GLOBAL_RX_FIFO_SIZE	47
Table 5.21. USB IP 2/3 PHY Control Status Register Information	47
Table 6.1. Maximum Packet Size Based on the Configured Speed	54
Table 7.1. Generated Files	65
Table A.1. Resource Utilization Using LIFCL-33U-7CTG104C	129

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
AHB	Advanced High-Performance Bus
AHB-Lite	Advanced High-Performance Bus Lite version
APB	Advanced Peripheral Bus
API	Application Programming Interface
AXI	Advanced Extensible Interface
CPU	Central Processing Unit
ECC	Error Correction Code
FIFO	First In First Out
FPGA	Field Programmable Gate Array
HAL	Hardware Abstraction Layer
IP	Intellectual Property
LED	Light Emitting Diode
LMMI	Lattice Memory Mapped Interface
MUX	Multiplexer
PRBS	Pseudorandom Binary Sequence
RISC	Reduced Instruction Set Computer
TRB	Transfer Request Block
UART	Universal Asynchronous Receiver/ Transmitter
USB	Universal Serial Bus

1. Introduction

1.1. Overview of the IP

The Lattice Semiconductor USB 2.0/3.2 intellectual property (IP) core provides a solution to interface with the USB host. The design is implemented using Verilog HDL. The IP core is targeted for Lattice CrossLinkU™-NX FPGAs. The IP core is implemented using the Lattice Radiant™ software integrated with the Synopsys® Synplify Pro® synthesis tool.

The Lattice USB 2.0/3.2 IP core supports the USB-IF compliant USB standard.

1.2. Quick Facts

Table 1.1. Summary of the USB2.0/3.2 IP Core

IP Requirements	Supported FPGA Family	CrossLinkU-NX (LIFCL-33U)
	IP Changes	For a list of changes to the IP, refer to the USB 2.0/3.2 IP Release Notes (FPGA-RN-02037) .
Resource Utilization	Performance Grade	–7 , –8 , –9
Design Tool Support	Lattice Implementation	IP core v3.2.0—Lattice Radiant software 2025.1.1
	Synthesis	Synopsys Synplify Pro software for Lattice
	Simulation	Not supported

1.3. IP Support Summary

Table 1.2. USB2.0/3.2 IP Support Readiness

Device Family	Radiant Timing Model	Hardware Validated
CrossLinkU-NX	Final	Yes

1.4. Features

The following lists the key features of the USB 2.0/3.2 IP core:

- Supports High Speed and Full Speed USB 2.0 specification and Super Speed USB 3.2 specification.
- Supports high-speed, high-bandwidth isochronous transactions.
- Supports two control endpoints at EP0/1 (1 IN 1 OUT) and eight individually reconfigurable logical endpoints. Each logical endpoint reflects two physical endpoints: one IN and one OUT. For more information, refer to the [CrossLink-NX-33 and CrossLinkU-NX Data Sheet \(FPGA-DS-02104\)](#).
- Includes an example design as a reference usage model.

1.5. Licensing and Ordering Information

The USB 2.0/3.2 IP is provided at no additional cost with the Lattice Radiant software.

1.6. Hardware Support

Refer to the [Example Design](#) section for more information on the boards used.

1.7. Minimum Device Requirements

The minimum device requirements for the USB 2.0/3.2 IP core are as follows:

Table 1.3. Minimum Device Requirements for the USB 2.0/3.2 IP Core

Devices	Speed Grades
CrossLinkU-NX	-7

1.8. Naming Conventions

1.8.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.8.2. Signal Names

Signals name having extension as:

- `_n` are active low signals (asserted when value is logic 0)
- `_i` are input signals
- `_o` are output signals
- `_io` are bidirectional signals

2. Functional Description

2.1. IP Architecture Overview

The USB 2.0/3.2 IP core includes the following components:

- USB23 primitive
- Lattice Memory Mapped Interface (LMMI)
- AHBL to LMMI converter (Optional)
- Advanced Extensible Interface (AXI)
- USB data lines

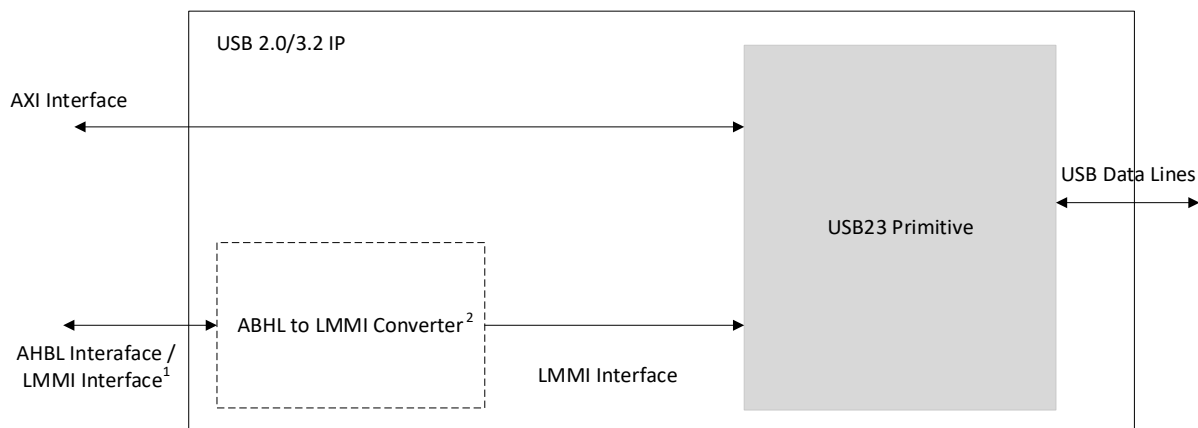


Figure 2.1. USB 2.0/3.2 IP Block Diagram

Notes:

1. The LMMI interface is available if the AHBL interface is not enabled.
2. The AHBL to LMMI converter is optional and can be enabled through the GUI. The default mode is LMMI interface.

2.2. Clocking

2.2.1. Clocking Overview

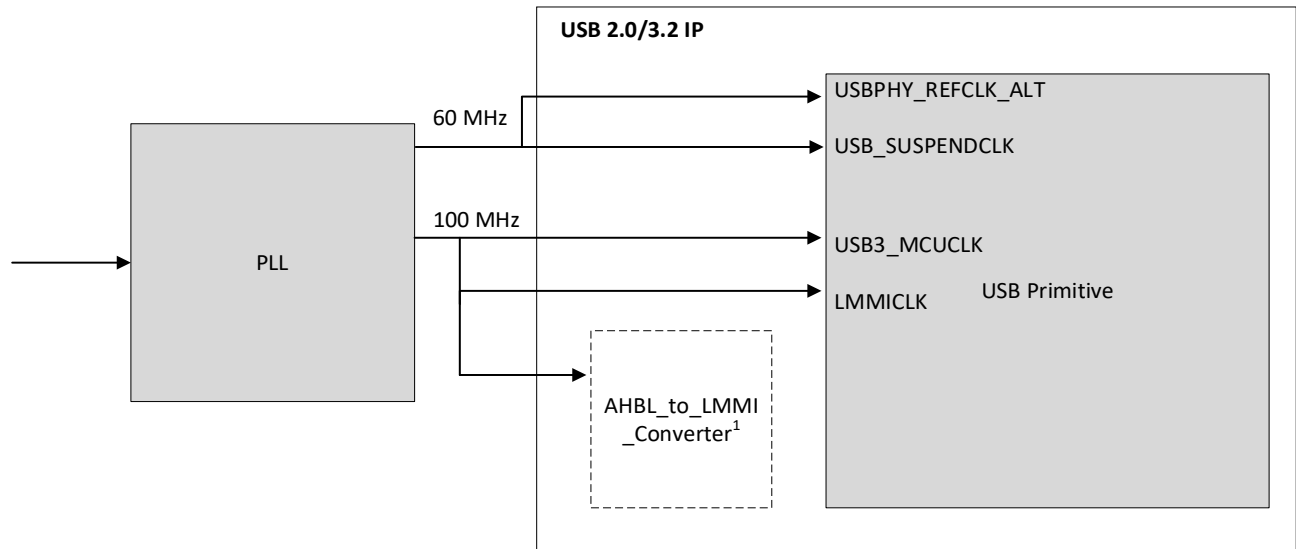


Figure 2.2. USB 2.0/3.2 IP Clock Domain Block Diagram

Note:

1. The AHBL_to_LMMI_Converter is enabled through the GUI by selecting HOST_IF = AHBL. The AHBL clock and LMMI clock must share the same clock source.

2.1 User Interfaces

The LMMI configuration interface and the AXI data interface are available for the user application to connect with the USB 2.0/3.2 IP and perform the transfers. These interfaces use usb23_reference_clk.

2.2 Internal IP Blocks

2.2.1 USB 2.0/3.2

The USB 2.0/3.2 is the hard IP block for the USB device. It is compliant to the USB-IF standards and provides all the functionality.

2.2.2 AHBL-to-LMMI Converter

The AHBL-to-LMMI converter is used to convert the AHBL interface to the LMMI interface. This block is optional and can be enabled through the GUI selection. The default mode is LMMI interface.

3. IP Parameter Description

The configurable attributes of the USB 2.0/3.2 IP core are shown in [Table 3.1](#). You can configure the IP core by setting the attributes accordingly in the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Table 3.1. General Attributes of the USB 2.0/3.2 IP Core

Attribute	Default Value	Description
USB Mode	USB23	USB mode of operation options: • USB2: Supports USB2 mode only • USB3: Supports USB3 mode only • USB23: Supports both USB2 and USB3 mode (recommended)
GSR	Enabled	Global set reset: • Enabled • Disabled
Host_IF	LMMI	CSR interface: • LMMI • AHBL

4. Signal Description

This section describes the USB 2.0/3.2 IP ports.

4.1. Clock Interface

Table 4.1. Clock Ports

Port	Type	Width	Description
USB3_MCUCLK_i	Input	1	Clock for the USB Hard IP and other modules. The frequency is 100 MHz.
USBPHY_REFCLK_ALT_i	Input	1	Internal clock for the USB 2.0 and USB 3.0 (60 MHz).
REFINCLKEXTP_i	Input	1	USB 3.0 PHY external positive differential clock.
REFINCLKEXTM_i	Input	1	USB 3.0 PHY external negative differential clock.
USB_SUSPENDCLK_i	Input	1	To achieve low power during suspend state, the PIPE3/MAC3/MAC2/MAC clock is switched to a lower frequency variable clock (60 MHz).
XOIN18_i	Input	1	24 MHz or 12 MHz crystal input pad.
XOOUT18_o	Output	1	24 MHz or 12 MHz crystal output pad.

4.2. Reset Interface

Table 4.2. Reset Interface

Port	Type	Width	Description
USB_RESETN_i	Input	1	Asynchronous, active low reset signal.
USB2_RESET_i	Input	1	Synchronous reset for USB 2.0 PHY. Resets all blocks in the transceiver. The RESET signal should retain no less than 300 ns.
USB3_SYSRSTN_i	Input	1	Active low reset signal for USB 3.0 PHY.

4.3. USB Interface

Table 4.3. USB Interface

Port	Type	Width	Description
usb23_DP	Inout	1	USB 2.0 positive data line.
usb23_DM	Inout	1	USB 2.0 negative data line.
RXM_i	Input	1	USB 3.0 negative differential data line for the receiver.
RXP_i	Input	1	USB 3.0 positive differential data line for the receiver.
TXM_o	Output	1	USB 3.0 negative differential data line for the transmitter.
TXP_o	Output	1	USB 3.0 positive differential data line for the transmitter.

4.4. LMMI/AHBL Interface – Configuration Path

Table 4.4. LMMI Interface - Configuration Path

Port	Type	Width	Description
lmmi_clk_i	Input	1	Clock signal for the LMMI interface. The clock frequency is 100 MHz.
lmmi_rst_n_i	Input	1	Active low reset signal for the LMMI interface.
lmmi_request_i	Input	1	Start transaction. Asserted during read or write cycle.
lmmi_offset_i	Input	15	LMMI address bus.
lmmi_wr_rdn_i	Input	1	Indicates the direction of the transfer. Indicates a write access when high

Port	Type	Width	Description
			and a read access when low.
lmmi_wdata_i	Input	32	LMMI write data bus.
lmmi_rdata_o	Output	32	LMMI read data bus.
lmmi_rdata_valid_o	Output	1	Indicates that the read transaction is complete and lmmi_rdata_o contains valid data.
lmmi_ready_o	Output	1	Indicates that the previous transaction is complete and the target is ready to start a new transaction.

Table 4.5. AHBL Interface - Configuration Path

Port	Type	Width	Description
ahbl_clk	Input	1	Clock signal for the AHBL interface. The clock frequency is 100 MHz.
ahbl_rst_n	Input	1	Active low reset signal for the AHBL interface.
ahbl_haddr_i	Input	32	Write address.
ahbl_hburst_i	Input	3	Unused.
ahbl_hprot_i	Input	4	Unused.
ahbl_hmastlock_i	Input	1	Unused.
ahbl_hsize_i	Input	1	Unused.
ahbl_hready_i	Input	1	Indicates transfer completion.
ahbl_hsel_i	Input	1	Subordinate select.
ahbl_hwdata_i	Input	32	AHBL write data bus.
ahbl_htrans_i	Input	2	Transfer type of the current transfer (IDLE/BUSY/NSEQ/SEQ).
ahbl_hwrite_i	Input	1	Write = 1, Read = 0.
ahbl_hrdata_o	Output	32	AHBL read data bus.
ahbl_hreadyout_o	Output	1	Indicates that the read transaction is complete and lmmi_rdata_o contains valid data.
ahbl_hresp_o	output	1	Subordinate response (OKAY/ERROR) .

4.5. AXI Interface – Data Path

Table 4.6. AXI Interface - Data Path

Port	Type	Width	Description
xm_awaddr_o	Output	32	AXI write address. Specifies the address of the first transfer in the write burst transaction. Associated control signals determine the address of the remaining transfers in a burst.
xm_awid_o	Output	8	AXI write address ID. Gives identification tag for write address signals.
xm_awvalid_o	Output	1	AXI write address valid. This signal indicates that the channel is signaling valid write address and control information when it is high. Address and control information remains stable until xm_awready_i signal is high.
xm_awsiz_o	Output	3	AXI write burst size. Indicates the size of each transfer in a burst. Byte lane strobes indicate exactly which byte lanes to update.
xm_awlen_o	Output	8	AXI write burst length. Specifies the exact number of transfers in burst, determines the number of data transfers associated with the address.
xm_awburst_o	Output	2	AXI write burst type. Combined with the size information and shows how the address for each transfer within a burst is calculated. 01 – INCR 10 – WRAP 11 – RESERVED 00 – FIXED

Port	Type	Width	Description
xm_awcache_o	Output	4	AXI write memory type. This signal indicates how transactions are required to progress through a system. [0] – Bufferable [1] – Cacheable [2] – Write through [3] – Write back
xm_awprot_o	Output	3	AXI write protection type. This signal indicates the privilege and security level of the transaction and whether the transaction is for data access or instruction access. [0] – Privileged or unprivileged [1] – Secure or non-secure [2] – Instruction or data access
xm_awready_i	Input	1	AXI write address ready. When the signal is high it indicates that the subordinate is ready to accept the address and associated control signals.
xm_awmisc_info_o	Output	4	AXI write request miscellaneous information. Currently not used so leave it unconnected.
xm_awlock_o	Output	2	AXI write lock type. Provides additional information about atomic characteristics of the transfer. 00 – Normal 01 – Exclusive 10 – Locked 11 – Reserved.
xm_wdata_o	Output	64	AXI write data.
xm_wstrb_o	Output	8	AXI write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for every 8 bits of the write data bus.
xm_wvalid_o	Output	1	AXI write valid. This signal indicates that valid write data and strobes are available when it is high.
xm_wlast_o	Output	1	AXI write last. This signal indicates the last transfer in a write burst.
xm_wid_o	Output	8	AXI write data ID. Gives identification tag for write data transfer signals.
xm_wready_i	Input	1	AXI write ready. When this signal is high, it indicates that the subordinate can accept write data.
xm_bid_i	Input	8	AXI response ID tag. Gives identification tag for write response signals.
xm_bvalid_i	Input	1	AXI write response valid. When this signal is high, it indicates that the valid write response is available.
xm_bmisc_info_i	Input	4	AXI write response miscellaneous information. This signal is currently not in use, tie to '0'.
xm_bresp_i	Input	2	AXI write response. Indicates the status of the write transaction. This signal indicates an error if the data phase ID does not match the address phase ID. For example, if xm_awid_o = 0 and xm_wid_o != 0, the controller responds with bresp error. 00 – OKAY (Normal access success) 01 – EXOKAY (Exclusive access okay) 10 – SLVERR (Subordinate error) 11 – DECERR (Decode error)
xm_bready_o	Output	1	AXI response ready. This signal indicates that the manager can accept a write response when it is high.
xm_araddr_o	Output	32	AXI read address. Gives the initial address of the read burst transaction. The control signals issued alongside the address show how the address is calculated for the remaining transfers in a burst.
xm_arvalid_o	Output	1	AXI read address valid. When this signal is high, it indicates that the channel is signaling a valid read address and control information. Address and control information remains stable until xm_arready_i signal is high.
xm_arlen_o	Output	8	AXI read burst length. Specifies the exact number of transfers in burst,

Port	Type	Width	Description
			determines the number of data transfers associated with the address.
xm_arsize_o	Output	3	AXI read burst size. Indicates the size of each transfer in a burst.
xm_arburst_o	Output	2	AXI read burst type. Combined with the size information and shows how the address for each transfer within a burst is calculated. 01 – INCR 10 – WRAP 00 – FIXED 11 – RESERVED
xm_arid_o	Output	8	AXI read address ID. Gives identification tag for read address signals.
xm_arcache_o	Output	4	AXI read memory type. This signal indicates how transactions are required to progress through a system. [0] – Bufferable [1] – Cacheable [2] – Write through [3] – Write back
xm_arprot_o	Output	3	AXI read protection type. This signal indicates the privilege and security level of the transaction, and whether the transaction is for data access or instruction access. [0] – privileged or Unprivileged [1] – secure or non-secure [2] – instruction or data access
xm_armisc_info_o	Output	4	AXI read request miscellaneous information. Currently not in use so leave it unconnected.
xm_arlock_o	Output	2	AXI read lock type. Provides additional information about atomic characteristics of the transfer. 00 – Normal 01 – Exclusive 10 – Locked 11 -- Reserved
xm_arready_i	Input	1	AXI read address ready. When the signal is high, it indicates that the subordinate is ready to accept read address and associated control signals.
xm_rid_i	Input	8	AXI read ID tag. Gives the identification tag for read data transfer signals.
xm_rvalid_i	Input	1	AXI read valid. When this signal is high, it indicates that required read data is available and read transfer can complete.
xm_rlast_i	Input	1	AXI read last. This signal indicates the last transfer in a read burst.
xm_rdata_i	Input	64	AXI read data.
xm_rmisc_info_i	Input	4	AXI read response miscellaneous information. This signal is currently not used, tie it to '0'.
xm_rresp_i	Input	2	AXI read response. Indicates the status of the read transaction. 00 – OKAY (Normal access success) 01 – EXOKAY (Exclusive access okay) 10 – SLVERR (Subordinate error) 11 – DECERR (Decode error)
xm_rready_o	Output	1	AXI read ready. This signal indicates that the manager can accept a read and response when it is high.
interrupt_o	Output	1	Interrupt output signal. This signal indicates that there is an interrupt request.

4.6. Other Signals

Table 4.7. Other Signals

Port	Type	Width	Description
VBUS_i	Input	1	Power signal.
RESEXTUSB2	Inout	1	This is a pad to connect an extra 300 ohm ($\pm 1\%$) resistor to the ground.
RESEXTUSB3	Inout	1	This is a pad to connect an extra 300 ohm ($\pm 1\%$) resistor to the ground.

5. Register Description

This section describes the USB IP register information. The USB IP has the following three types of register interfaces:

- USB IP Controller Register Information
- USB IP Controller Control Status Register Information
- USB IP 2.0 PHY Control Status Register Information

The USB IP register address is in 17-bit format ([0:16] bits) as follows:

- USB IP 2.0 PHY Control Status Register Information: 16th bit = 1 and [15:12] = 2

5.1. USB IP Controller Register Information

Table 5.1 lists the USB IP Controller Register Information.

Table 5.1. USB IP Controller Register Information

Offset	Register Name	Description
0xc110	GLOBAL_CTL	Global Core Control Register.
0xc120	GLOBAL_ID	Global ID Register.
0xc200 + (i * 0x4)	GLOBAL_USB2_PHY_CFG(#n) (for n = 0)	Global USB2 PHY Configuration Register.
0xc2c0 + (i * 0x4)	GLOBAL_USB3_PIPE_CTL(#n) (for n = 0)	Global USB 3.0 PIPE Control Register.
0xc400 + (i * 0x10)	GLOBAL_EVNT_ADR(#n) (for n = 0)	Global Event Buffer Address Register.
0xc404 + (i * 0x10)	GLOBAL_EVNT_ADRHI(#n) (for n = 0)	Global Event Buffer Address (High) Register.
0xc408 + (i * 0x10)	GLOBAL_EVNT_SIZ(#n) (for n = 0)	Global Event Buffer Size Register.
0xc40c + (i * 0x10)	GLOBAL_EVNT_COUNT(#n) (for n = 0)	Global Event Buffer Count Register.
0xc700	DEVICE_CFG	Device Configuration Register.
0xc704	DEVICE_CTL	Device Control Register Note.
0xc708	DEVICE_EVNT_EN	Device Event Enable Register.
0xc70c	DEVICE_STS	Device Status Register.
0xc720	DEVICE_ACT_EP_EN	Device Active USB Endpoint Enable Register.
0xc800 + (i * 0x10)	DEVICE_EP_CMDPAR2[0:7]	Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2n).
0xc804 + (i * 0x10)	DEVICE_EP_CMDPAR1[0:7]	Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1n).
0xc808 + (i * 0x10)	DEVICE_EP_CMDPAR0[0:7]	Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0n).
0xc80c + (i * 0x10)	DEVICE_EP_CMD[0:7]	Device Physical Endpoint-n Command Register (DEPCMD).
0xc300 + (i*0x4)	GLOBAL_TX_FIFO_SIZE	Global Transmit FIFO Size Register.
0xc380 + (n*0x4)	GLOBAL_RX_FIFO_SIZE	Global Receiver FIFO Size Register.

5.1.1 Global Core Control Register (GCTL)

The default value of the GLOBAL_CTL Register is 0x30c12004.

Table 5.2. GLOBAL_CTL

Bit	Label	Access	Description
31:19	PWRDNSCALE	R/W	Power Down Scale (PwrDnScale). The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 controller that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock. The Power Down Scale field specifies how many suspend_clk periods fit into a 16 kHz clock period. When performing the division, round up the remainder. For example, when using an 8-bit/16-bit/32-bit PHY and 25 MHz Suspend clock, $\text{Power Down Scale} = 25000 \text{ kHz} / 16 \text{ kHz} = 13'd1563 \text{ (rounded up)}$ Note: - Minimum suspend clock frequency is 32 kHz - Maximum suspend clock frequency is 125 MHz The LTSSM uses the suspend clock for 12-ms and 100-ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%. 12 ms + 0~+50% accuracy = 18 ms (Range is 12 ms – 18 ms) 100 ms + 0~+50% accuracy = 150 ms (Range is 100 ms - 150 ms). The suspend clock accuracy requirement is: $(12,000/62.5) * (\text{GLOBAL_CTL}[31:19]) * \text{actual suspend_clk_period}$ must be between 12,000 and 18,000 $(100,000/62.5) * (\text{GLOBAL_CTL}[31:19]) * \text{actual suspend_clk_period}$ must be between 100,000 and 150,000 For example, if suspend_clk frequency varies from 7.5 MHz to 10.5 MHz, then the value needs to be programmed is: - Power Down Scale = $10500 / 16 = 657$ (rounded up and fastest frequency used).
18	MASTERFILTBYPASS	R/W	Controller Filter Bypass. When this bit is set to 1'b1, irrespective of the parameter 'USB3_EN_BUS_FILTERS' chosen, all the filters in the USB3_filter module are bypassed. The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.
17	BYPSETADDR	R/W	Bypass SetAddress in Device Mode. When BYPSETADDR bit is set, the device controller uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens. When the xHCI Debug capability is enabled and this bit is set, the Debug Target immediately enters the configured state without requiring the Debug Host to send a SetAddress or SetConfig request. Note: In the actual hardware, this bit must be set to 1'b0.
16	U2RSTECN	R/W	U2RSTECN If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. For each attempt, the device checks receiver termination eight times. From 2.60a release, this bit controls whether to check for

Bit	Label	Access	Description
			Rx.Detect eight times or one time for every attempt. Device controller on USB 2.0 reset checks for receiver termination eight times per attempt, if this bit is set to zero, or only once per attempt if the bit is set to one. Note: This bit is applicable only in device mode.
15:14	FRMSCLDWN	R/W	FRMSCLDWN This field scales down device view of a SOF/USOF/ITP duration. For SS/HS mode: - Value of 2'h3 implements interval to be 15.625 us - Value of 2'h2 implements interval to be 31.25 us - Value of 2'h1 implements interval to be 62.5 us - Value of 2'h0 implements interval to be 125 us For FS mode, the scale-down value is multiplied by 8.
13:12	PRTCAPDIR	R/W	PRTCAPDIR: Port Capability Direction (PrtCapDir) 2'b01: for Host configurations 2'b10: for Device configurations Note: For static host-only or device-only applications, use DRD Host or DRD Device mode. The combination of GLOBAL_CTL.PrtCapDir=2'b11 with SRP and HNP/RSP disabled is not recommended for these applications. The sequence for switching modes in DRD configuration is as follows: Switching from device to host: - Reset the controller using GLOBAL_CTL[11] (CoreSoftReset). - Set GLOBAL_CTL[13:12] (PrtCapDir) to 2'b01 (Host mode). - Reset the host using USBCMD.HCRESET. Switching from host to device: - Reset the controller using GLOBAL_CTL[11] (CoreSoftReset). - Set GLOBAL_CTL[13:12] (PrtCapDir) to 2'b10 (Device mode). - Reset the device by setting Device_CTL[30] (CSftRst). Programming this field with random data causes the controller to keep toggling between the host mode and the device mode. Bit Bash register testing is not recommended.
11	CORESOFTRSET	R/W	Core Soft Reset (CoreSoftReset) 1'b0 - No soft reset 1'b1 - Soft reset to controller Clears the interrupts and all the CSRs except the following registers: - GLOBAL_CTL - GLOBAL_USB2_PHY_CFGn registers - GLOBAL_USB3_PIPE_CTLn registers - DEVICE_CFG - DEVICE_CTL - DEVICE_EVT_EN - DEVICE_STS When you reset PHYs (using GUBS3PHYCFG or GLOBAL_USB3_PIPE_CTL registers), you must keep the controller in reset state until PHY clocks are stable. This controls the bus, RAM, and MAC domain resets. Note: This bit is for debugging purposes only. Use USBCMD.HCRESET in xHCI Mode and Device_CTL[30] (CSftRst) in device mode for soft reset. Programming this field with random data will reset the internal logic of the host controller. Due to this side effect, Bit Bash

Bit	Label	Access	Description
			register testing is not recommended.
10	SOFITPSYNC	R/W	SOFITPSYNC If this bit is set to '0' operating in host mode, the controller keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever there is a SuperSpeed port that is not in Rx.Detect, SS.Disable and U3. If this bit is set to '1' operating in host mode, the controller keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever the other non-SuperSpeed ports are not in a suspended state. This feature is useful because it saves power by suspending UTMI/ULPI when SuperSpeed only is active, and it helps resolve when the PHY does not transmit a host resume unless it is placed in suspend state. This bit must be programmed as a part of initialization at power-on reset, and must not be dynamically changed afterwards. Notes: - USB2PHYCFGn[6].PhySusp eventually decides to put the UTMI/ULPI PHY in to suspend state. In addition, when this bit is set to '1', the controller generates ITP from the ref_clk based counter. Otherwise, ITP and SOF are generated from utmi/ulpi_clk[0] based counter. To program the reference clock period inside the controller, refer to GUCTL[31:22].REFCLKPER. - This feature is valid in Host and DRD configurations and used only in Host mode operation. - If you never use this feature or the GFLADJ.GFLADJ_REFCLK_LPM_SEL, the minimum frequency for the ref_clk can be as low as 32KHz. You can connect the suspend_clk (as low as 32 KHz) to the ref_clk. - If you plan to enable hardware-based LPM or software-based LPM (PORTPMSC. HLE=1), then you cannot use this feature. Turn off this feature by setting this bit to '0' and use the GFLADJ.GFLADJ_REFCLK_LPM_SEL feature. - If you set this bit to '1', the GLOBAL_USB2_PHY_CFG.U2_FRE-ECLK_EXISTS bit and the USB3_FRE-ECLK_USB2_EXIST parameter must be set to '0'. Program this bit to 0 if the controller is intended to be operated in USB 3.0 mode.
9	U1U2TimerScale	R/W	Disable U1/U2 timer Scaledown (U1U2TimerScale).
8	DEBUGATTACH	R/W	Debug Attach when this bit is set, - SS Link proceeds directly to the Polling link state (after RUN/STOP in the Device_CTL register is asserted) without checking remote termination. - Link LFPS polling timeout is infinite. - Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish).
7:6	RAMCLKSEL	R/W	RAM Clock Select (RAMClkSel) 2'b00: bus clock 2'b01: pipe clock (Only used in device mode) 2'b10: In device mode, pipe/2 clock. In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2/U3 ports 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode) In Host mode, controller switches ram_clk between pipe_clk,

Bit	Label	Access	Description
			mac2_clk and bus_clk based on the status of the U2/U3 ports. In device mode, after a USB reset and USB disconnect, the hardware clears these bits to 2'b00. Note: In device mode, if you set RAMClkSel to 2'b11 (mac2_clk), the controller internally switches the ram_clk to bus_clk when the link state changes to Suspend (L2) and), and switches the ram_clk back to mac2_clk when the link state changes to resume or U2. - In host mode, if a value of 2/3 is chosen, then controller switches ram_clk between bus_clk, mac2_clk and pipe_clk, pipe_clk/2, based on the state of the U2/U3 ports. For example, if only the U2 port is active and the U3 ports are suspended, then the ram_clk is switched to mac2_clk. When only the U3 ports are active and the U2 ports are suspended, the controller internally switches the ram_clk to pipe3 clock and when all U2 and U3 ports are suspended, it switches the ram_clk to bus_clk. This allows decoupling the ram_clk from the bus_clk, and depending on the band- width requirement allows the bus_clk to be run at a lower frequency than the ram_clk requirements. The bus_clk frequency still cannot be less than 60 MHz in host mode, and this is not verified. A value of 2 can be chosen only if the pipe data width is 8 or 16 bits. In this case the when the ram_clk is switched to pipe_clk, it uses pipe_clk/2 instead of pipe_clk. If a value of 3 is chosen for RAMClkSel, when ram_clk is switched to pipe_clk, then pipe_clk is used without any divider. - In device mode, when RAMClkSel != 2'b00, the bus_ - clk_early frequency can be a minimum of 1 MHz. This is tested in hardware with Linux, Microsoft Windows 8, and MCCI Windows7 host drivers. Only control and non-periodic transfers are supported when bus_clk is 1 MHz. For periodic applications, the bus_ - clk_early minimum frequency is higher depending on your application and SoC bus. Even though 1 MHz has been tested with standard host drivers, It is recommended to use 5 MHz minimum for ASIC designs to provide a margin or at least have a backup option to increase the bus_clk frequency to 5 MHz if needed. Programming this field with random data will cause side effects. Bit Bash register testing is not recommended.
5:4	SCALEDOWN	R/W	Scale-Down Mode (ScaleDown). When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation. <i>HS/FS/LS Modes</i> 2'b00: Disables all scale-downs. Actual timing values are used. 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include Speed enumeration, HNP/SRP, and Host mode suspend and resume. 2'b10: Enables scale-down of Device mode suspend and resume timing values only. 2'b11: Enables bit 0 and bit 1 scale-down timing values. <i>SS Mode</i> 2'b00: Disables all scale-downs. Actual timing values are used. 2'b01: Enables scaled down SS timing and repeat values including: (1) Number of TxEq training sequences reduce to 8; (2) LFPS polling burst time reduce to 256 nS; (3) LFPS warm reset receive reduce to 30 uS. Refer to the rtl_vip_s-caledown_mapping.xls file under <work- space>/sim/SoC_sim

Bit	Label	Access	Description
			directory for the complete list. 2'b10: No TxEq training sequences are sent. Overrides Bit 4. 2'b11: Enables bit 0 and bit 1 scale-down timing values.
3	DISSCRAMBLE	R/W	Disable Scrambling (DisScramble) Transmit request to Link Partner on next transition to Recovery or Polling.
2	U2EXIT_LFPS	R/W	U2EXIT_LFPS If this bit is, 0: the link treats 248ns LFPS as a valid U2 exit. 1: the link waits for 8us of LFPS before it detects a valid U2 exit. This bit is added to improve interoperability with a third-party host/device controller. This host/device controller in U2 state while performing receiver detection generates an LFPS glitch of about 4 mins duration. This causes the host/device to exit from U2 state because the LFPS filter value is 248 ns. With the new functionality enabled, the host/device can stay in U2 while ignoring this glitch from the host/device controller. This bit is applicable for both the host and the device controller. This bit is added to improve interoperability with a third-party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4 ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248 ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.
1	GblHibernationEn	RO	GblHibernationEn This bit enables hibernation at the global level. If hibernation is not enabled through this bit, the PMU immediately accepts the D0->D3 and D3->D0 power state change requests but does not save or restore any controller state. In addition, the PMUs never drive the PHY interfaces and let the controller continue to drive the PHY interfaces.
0	DSBLCLKGTNG	R/W	Disable Clock Gating (DsblClkGtng) This bit is set to 1 and the controller is in low-power mode, internal clock gating is disabled. You can set this bit to 1'b1 after power on reset.

5.1.2 Global ID Register (GSNPSID)

This is a read-only register that contains the release number of the USB controller.

Table 5.3. GLOBAL_ID

Bit	Label	Access	Description
31:0	SYNOPSISID	RO	Bit[31:16] indicates core identification number. 0x5533 is ASCII for USB3. Bit[15:0] indicates the release number. Current release is 3.30a.

5.1.3 Global USB2 PHY Configuration Register (GUSB2PHYCFG(#n))

The default value of the Global USB2 PHY Configuration Register is 0x00102400.

Table 5.4. GLOBAL_USB2_PHY_CFG(#n) (for n = 0)

Bit	Label	Access	Description
31	PHYSOFTRST	R/W	UTMI PHY Soft Reset (PHYSoftRst) Asserts the usb2phy_reset signal to to reset a UTMI PHY. Not applicable to ULPI because ULPI PHYs are reset via their FunctionControl.Reset register, and the controller automatically writes to this register when the controller is reset (vcc_reset_n, USBCMD.HCRST, Device_CTL[30] (CSftRst), or GLOBAL_CTL.SoftReset)
30	U2_FREECLK_EXISTS	R/W	U2_FREECLK_EXISTS Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active. If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The controller uses the Port-0 clock for generating the internal mac2 clock. 1'b0: USB 2.0 free clock does not exist 1'b1: USB 2.0 free clock exists Note: When the controller is configured as device-only (USB3_MODE = 0), do not set this bit to 1.
29	ULPI_LPM_WITH_OPMODE_CHK	R/W	ULPI_LPM_WITH_OPMODE_CHK Supports the LPM over ULPI without NOPID token to the ULPI PHY. If this bit is set, the ULPI PHY is expected to qualify the EXT PID with OPMODE=2'b00 for LPM and not treat it as a NOPID. Check with your PHY vendor about your PHY behavior. This bit is valid only when the USB3_HSPHY_INTERFACE parameter is 2 or 3. 1'b0: A NOPID is sent before sending an EXTPID for LPM; 1'b1: An EXTPID is sent without previously sending a NOPID.
28:27	HSIC_CON_WIDTH_ADJ	RO	HSIC_CON_WIDTH_ADJ This bit is used in the HSIC device mode of operation. By default, the connect duration for the HSIC device controller is 3 times the strobe period. You can change this duration to 4, 5, or 6 times the strobe period by setting the value of this field to 1, 2, or 3. This value is added to the default connect duration.
26	INV_SEL_HSIC	RO	INV_SEL_HSIC The application driver uses this bit to control the HSIC enable or disable function. When set to '1', this bit overrides and functionally inverts the <i>if_select_hsic</i> input signal. If

Bit	Label	Access	Description
			{INV_SEL_HSIC, if_select_hsic} is: 00: HSIC Capability is disabled 01: HSIC Capability is enabled 10: HSIC Capability is enabled 11: HSIC Capability is disabled If the controller operates as non-HSIC-capable, it can only connect to non-HSIC-capable PHYs. If it operates as HSIC-capable, it can connect to HSIC-capable PHYs. This bit is reserved if the USB3_ENABLE_HSIC parameter is set to '0'. When selecting the HSIC feature, set the host side to HSIC mode first, then set the device mode side. If the device side is set to HSIC mode first and if the host does not see a connection in HSIC mode, then you must de-select the device HSIC mode and select it again using the if_select_hsic setting or register bit GLOBAL_USB2_PHY_CFGn[26] to ensure that the device can connect to the host.
25	OVRD_FSLDISC_TIME	R/W	Overriding the FS/LS disconnect time to 32 us. - If this value is 0, the FS/LS disconnect time is set to 2.5 us as per the USB specification. - If this value is non-zero, the disconnect detection time is set to 32 us. Normally, this value is set to 0. However, if the USB 2.0 PHYs introduce noise on the UTMI linestate and cause SE0 glitches longer than 2.5 us, then a false disconnect condition may get triggered. To avoid interoperability issues with these PHYs, this bit can be set to 1.
24:22	LSTRD	R/W	LS Turnaround Time (LSTRDTIM) This field indicates the value of the RX-to-TX packet gap for LS devices. The encoding is as follows: 0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times 4: 4 bit times 5: 4.5 bit times 6: 5 bit times 7: 5.5 bit times Note: - This field is applicable only in Host mode. - For normal operation (to work with most LS devices), set the default value of this field to 3'h0 (2-bit times). - The programmable LS device inter-packet gap and turn-around delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3-bit times of inter-packet gap to work correctly. - Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMI/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement.
21:19	LSIPD	R/W	LS Inter-Packet Time (LSIPD) This field indicates the value of TX-to-TX packet gap for LS devices. The encoding is as follows:

Bit	Label	Access	Description
			0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times 4: 4 bit times 5: 4.5 bit times 6: 5 bit times 7: 5.5 bit times Note: - This field is applicable only in Host mode. - For normal operation (to work with most LS devices), set the default value of this field to 3'h2 (3-bit times). - The programmable LS device inter-packet gap and turn-around delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3-bit times of inter-packet gap to work correctly. - Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMI/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement.
18	ULPIEXTVBUSINDIACTOR	R/W	ULPI External VBUS Indicator (ULPIExtVbusIndicator) Indicates the ULPI PHY VBUS over-current indicator. 1'b0: PHY uses an internal VBUS valid comparator 1'b1: PHY uses an external VBUS valid comparator Valid only when RTL parameter USB3_HSPHY_INTERFACE = 2 or 3.
17	ULPIEXTVBUSDRV	R/W	ULPI External VBUS Drive (ULPIExtVbusDrv) Selects supply source to drive 5V on VBUS, in the ULPI PHY. 1'b0: PHY drives VBUS with internal charge pump (default) 1'b1: PHY drives VBUS with an external supply. (Only when RTL parameter USB3_HSPHY_INTERFACE = 2 or 3)
16	reserved_16	RO	Reserved.
15	ULPIAUTORES	R/W	ULPI Auto Resume (ULPIAutoRes) Sets the AutoResume bit in Interface Control register on the ULPI PHY. 1'b0: PHY does not use the AutoResume feature 1'b1: PHY uses the AutoResume feature Valid only when RTL parameter USB3_HSPHY_INTERFACE = 2 or 3.
14	reserved_14	RO	Reserved.
13:10	USBTRDTIM	R/W	USB 2.0 Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). The following are the required values for the minimum SoC bus frequency of 60 MHz. USB turnaround time is a critical certification criterion when using long cables and five hub levels. The required values for this field: 4'h5: When the MAC interface is 16-bit UTMI+ 4'h9: When the MAC interface is 8-bit UTMI+/ULPI If SoC bus clock is less than 60 MHz, and USB turnaround time

Bit	Label	Access	Description
			is not critical, this field can be set to a larger value. Note: This field is valid only in device mode.
9	XCVRDLY	R/W	Transceiver Delay: Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp. When this bit is set to 1, a delay (of approximately 2.5 us) is introduced from the time when the Transceiver Select is set to 2'b00 (HS) to the time the TxValid is driven to 0 for sending the chirp-K. This delay is required for some UTMI/ULPI PHYs. Note: - If you enable the hibernation feature when the device controller comes out of power-off, you must re-initialize this bit with the appropriate value because the controller does not save and restore this bit value during hibernation. - This bit is valid only in device mode.
8	ENBLSLPM	R/W	Enable utmi_sleep_n and utmi_l1_suspend_n (EnbLSlPM) The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state. 1'b0: utmi_sleep_n and utmi_l1_suspend_n assertion from the controller is not transferred to the external PHY 1'b1: utmi_sleep_n and utmi_l1_suspend_n assertion from the controller is transferred to the external PHY Note: In Device mode - Before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. Without disabling this bit, if a command is issued when the device is in L1 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not be completed.
7	PHYSEL	WO	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select: The application uses this bit to select a high-speed PHY or a full-speed transceiver. 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY. This bit is always 0, with Write Only access. 1'b1: USB 1.1 full-speed serial transceiver. This bit is always 1, with Write Only access. If both interface types are selected in coreConsultant (that is, parameters' values are not zero), the application uses this bit to select the active interface is active, with Read-Write bit access. Note: USB 1.1 full-serial transceiver is not supported. This bit always reads as 1'b0.
6	SUSPENDUSB20	R/W	Suspend USB2.0 HS/FS/LS PHY (SusPHY) When set, the USB2.0 PHY enters Suspend mode if Suspend conditions are valid. For DRD configurations, it is recommended that this bit is set to 0 during coreConsultant configuration. If it is set to 1, then the application must clear this bit after power-on reset. The application must set it to 1 after the controller initialization completes. For all other configurations, this bit can be set to 1 during the controller configuration.

Bit	Label	Access	Description
			Note: - In host mode, on reset, this bit is set to 1. Software can override this bit after reset. - In device mode, before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. If you issue a command without disabling this bit when the device is in L2 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not be completed.
5	FSINTF	RO	Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. 1'b0: 6-pin unidirectional full-speed serial interface. This bit is set to 0 with Read Only access. 1'b1: 3-pin bidirectional full-speed serial interface. This bit is set to 0 with Read Only access. Note: USB 1.1 full-speed serial interface is not supported. This bit always reads as 1'b0.
4	ULPI_UTMI_Sel	RO	ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select a UTMI+ or ULPI Interface. 1'b0: UTMI+ Interface 1'b1: ULPI Interface This bit is writable only if UTMI+ and ULPI is specified for High-Speed PHY Interface(s) in coreConsultant configuration (USB3_HSPHY_INTERFACE = 3). Otherwise, this bit is read-only, and the value depends on the interface selected through USB3_HSPHY_INTERFACE.
3	PHYIF	R/W	PHY Interface (PHYIf) If UTMI+ is selected, the application uses this bit to configure the controller to support a UTMI+ PHY with an 8 or 16-bit interface. 1'b0: 8 bits 1'b1: 16 bits - ULPI Mode: 1'b0 Note: - All the enabled 2.0 ports must have the same clock frequency as Port0 clock frequency (utmi_clk[0]). - The UTMI 8-bit and 16-bit modes cannot be used together for different ports at the same time (that is, all the ports must be in 8-bit mode, or all of them must be in 16-bit mode, at a time). - If any of the USB 2.0 ports is selected as ULPI port for operation, then all the USB 2.0 ports must be operating at 60 MHz.
2:0	TOutCal	R/W	HS/FS Timeout Calibration (TOutCal) The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the controller to account for additional delays introduced by the PHY. This may be required because the delay introduced by the PHY in generating the linestate condition may vary among PHYs. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout

Bit	Label	Access	Description
			value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of connection. The number of bit times added per PHY clock are: High-speed operation: - One 30-MHz PHY clock = 16 bit times - One 60-MHz PHY clock = 8 bit times Full-speed operation: - One 30-MHz PHY clock = 0.4 bit times - One 60-MHz PHY clock = 0.2 bit times - One 48-MHz PHY clock = 0.25 bit times

5.1.4 Global USB 3.0 PIPE Control Register (GUSB3PIPECTL(#n))

The default value of Global USB 3.0 PIPE Control Register is 0x010c8002.

Table 5.5. GLOBAL_USB3_PIPE_CTL(#n) (for n = 0)

Bit	Label	Access	Description
31	PHYSoftRst	R/W	USB3 PHY Soft Reset After setting this bit to '1', the software must clear this bit.
30	HstPrtCmpl	R/W	HstPrtCmpl This feature tests the PIPE PHY compliance patterns without having to have a test fixture on the USB 3.0 cable. This bit enables placing the SS port link into a compliance state. By default, this bit must be set to '1'b0. In compliance lab testing, the SS port link enters compliance after failing the first polling sequence after power on. Set this bit to 0, when you run compliance tests. The sequence for using this functionality is as follows: 1. Disconnect any plugged-in devices. 2. Perform USBCMD.HCRST or power-on-chip reset. 3. Set PORTSC.PLS=0xA. 4. Set PORTSC.PP=0. 5. Set GLOBAL_USB3_PIPE_CTL.HstPrtCmpl=1. This places the link into compliance state. To advance the compliance pattern, follow this sequence (toggle the set GLOBAL_USB3_PIPE_CTL.HstPrtCmpl): 1. Set GLOBAL_USB3_PIPE_CTL.HstPrtCmpl=0. 2. Set GLOBAL_USB3_PIPE_CTL.HstPrtCmpl=1. This advances the link to the next compliance pattern. To exit from the compliance state, perform USBCMD.HCRST or power-on-chip reset.
29	U2P3ok	R/W	P3 OK for SSInactive (SSIP3ok) 0: During link state SS.Inactive, put PHY in P2 (Default). 1: During link state SS.Inactive, put PHY in P3.
28	DisRxDetP3	R/W	Disabled receiver detection in P3 (DisRxDetP3) 0: If PHY is in P3 and controller must perform receiver detection, the controller performs receiver detection in P3. (Default). 1: If PHY is in P3 and controller must perform receiver detection, the controller changes the PHY power state to P2 and then performs receiver detection. After receiver detection, the cores change PHY power state to P3.

Bit	Label	Access	Description
27	Ux_exit_in_Px	R/W	Ux Exit in Px (Ux_exit_in_Px) 0: The controller does U1/U2/U3 exit in PHY power state P0 (default behavior). 1: The controller does U1/U2/U3 exit in PHY power state P1/P2/P3 respectively.
26	ping_enhancement_en	R/W	Ping Enhancement Enable (ping_enhancement_en) When set, the Downstream port U1 ping receive timeout becomes 500 ms instead of 300 ms. Minimum Ping.LFPS receive duration is 8 ns (one mac3_clk). This field is valid for the downstream port only.
25	u1u2exitfail_to_recov	R/W	U1U2exitfail to Recovery (u1u2exitfail_to_recov) When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to Recovery instead of SS Inactive. If Recovery fails, then the LTSSM enters SS.Inactive. This is an enhancement only. It prevents interoperability issue if the remote link does not do a proper handshake.
24	request_p1p2p3	R/W	Always Request P1/P2/P3 for U1/U2/U3 (request_p1p2p3) When set, the controller always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition. If this bit is 0, and immediate Ux exit (remotely initiated, or locally initiated) happens, the controller does not request P1/P2/P3 power state change.
23	StartRxDetU3RxDet	WO	Start Receiver Detection in U3/Rx.Detect (StartRxdetU3RxDet) If USB3_GLOBAL_USB3_PIPE_CTL_INIT[22] is set, and the link is in either U3 or Rx.Detect state, the controller starts receiver detection on the rising edge of this bit. This can only be used for Downstream ports. This bit must be set to '0' for Upstream ports. This feature must not be enabled for normal operation.
22	DisRxDetU3RxDet	R/W	Disable Receiver Detection in U3/Rx.Det When set, the controller does not handle receiver detection in either U3 or Rx.Detect states. USB3_GLOBAL_USB3_PIPE_CTL_INIT[23] must be used to start receiver detection manually. This bit can only be used for the downstream port. This bit must be set to "0" for Upstream ports. This feature must not be enabled for normal operation.
21:19	DelayP1P2P3	R/W	Delay P1P2P3 Delay P0 to P1/P2/P3 request when entering U1/U2/U3 until (USB3_GLOBAL_USB3_PIPE_CTL_INIT[21:19]*8) 8B10B error occurs, or Pipe3_RxValid drops to 0. USB3_GLOBAL_USB3_PIPE_CTL_INIT[18] must be 1 to enable this functionality.
18	DELAYP1TRANS	R/W	Delay PHY power change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively. 1'b1: When entering U1/U2/U3, delay the transition to P1/P2/P3 until the pipe3 signals, Pipe3_RxElecIdle is 1 and pipe3_RxValid is 0 1'b0: When entering U1/U2/U3, transition to P1/P2/P3 without checking for Pipe3_RxElecIdle and pipe3_RxValid.
17	SUSPENDENABLE	R/W	Suspend USB3.0 SS PHY (Suspend_en) When set, and if Suspend conditions are valid, the USB 3.0 PHY enters Suspend mode. For DRD configurations, it is recommended that this bit is set

Bit	Label	Access	Description
			to '0' during coreConsultant configuration. If it is set to '1', then the application must clear this bit after power-on reset. Application needs to set it to '1' after the controller initialization is completed. For all other configurations, this bit can be set to '1' during controller configuration.
16:15	DATWIDTH	RO	PIPE Data Width (DatWidth) 2'b00: 32 bits 2'b01: 16 bits One clock after reset, these bits receive the value seen on the pipe3_DataBusWidth. Note: 8-bit data width is not supported.
14	AbortRxDetInU2	R/W	Abort Rx Detect in U2 (AbortRxDetInU2) When set and the link state is U2, the controller will abort receiver detection if it receives U2 exit LFPS from the remote link partner. This bit is for the downstream port only.
13	SkipRxDet	R/W	Skip Rx Detect: When set, the controller skips Rx Detection if pipe3_RxElecIdle is low. Skip is defined as waiting for the appropriate timeout, then repeating the operation.
12	LFPSPOAlign	R/W	LFPS P0 Align: When set, - The controller deasserts LFPS transmission on the clock edge that it requests PHY power state 0 when exiting U1, U2, or U3 low power states. Otherwise, LFPS transmission is asserted one clock earlier. - The controller requests symbol transmission two pipe3_rx-_pclk periods after the PHY asserts PhyStatus as a result of the PHY switching from P1 or P2 state to P0 state.
11	P3P2TranOK	R/W	P3 P2 Transitions OK (P3P2TranOK) When set, the controller transitions directly from PHY power state P2 to P3 or from state P3 to P2. When not set, P0 is always entered as an intermediate state during transitions between P2 and P3, as defined in the PIPE3 Specification. According to the PIPE3 Specification, any direct transition between P3 and P2 is illegal.
10	P3ExSigP2	R/W	P3 Exit Signal in P2 (P3ExSigP2) When this bit is set, the controller always changes the PHY power state to P2, before attempting a U3 exit handshake.
9	LFPSFILTER	R/W	LFPS Filter (LFPSFilt) When set, filter LFPS reception with pipe3_RxValid in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe3_RxelecIdle and pipe3_RxValid are deasserted.
8	RX_DETECT_to_Polling_LFPS_Control	R/W	RX_DETECT to Polling.LFPS Control 1'b0 (Default): Enables a 400 us delay to start Polling LFPS after RX_DETECT. This allows VCM offset to settle to a proper level. 1'b1: Disables the 400 us delay to start Polling LFPS after RX_DETECT. During controller certification with third party PHY it is observed that the PHY is not able to meet the Tx AC common mode voltage active (VTX-CM-ACPP_ACTIVE <100mv) if the link starts polling within 80us from the time rx.detect is

Bit	Label	Access	Description
			performed. To meet this VTX-CM-ACPP_ACTIVE specification, the polling must be delayed further. If the PHY does not have issue, then they can set this bit to 1, which allows polling to start within 80us.
7	SSICEn	R/W	This field is not used.
6	TX_SWING	R/W	Tx Swing (TxSwing) Refer to the <i>PIPE3 Specification</i> .
5:3	TX_MARGIN	R/W	Tx Margin[2:0] (TxMargin) Refer to Table 5-3 of the <i>PIPE3 Specification</i> .
2:1	SS_TX_DE_EMPHASIS	R/W	Tx Deemphasis (TxDeemphasis) The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode. Refer to Table 5-3 of the <i>PIPE3 Specification</i> .
0	ELASTIC_BUFFER_MODE	R/W	Elastic Buffer Mode (ElasticBufferMode) Refer to Table 5-3 of the <i>PIPE3 specification</i> .

5.1.5 Global Event Buffer Address Register (GEVNTADDR(n))

Table 5.6. GLOBAL_EVNT_ADR(#n) (for n = 0)

Bit	Label	Access	Description
63:0	EVNTADR	R/W	EVNTADR DEFAULT VALUE: 0x0

5.1.6 Global Event Buffer Address (High) Register (GEVNTADRHl(#n))

Table 5.7. GLOBAL_EVNT_ADRI(#n) (for n=0)

Bit	Label	Access	Description
31:0	EVNTADRHl	R/W	Event Buffer Address (EvtAdrHi) Holds the higher 32 bits of start address of the external memory for the Event Buffer. During operation, the hardware does not update this address. DEFAULT VALUE: 0x0

5.1.7 Global Event Buffer Size Register (GEVNTSIZ(#n))

Table 5.8. GLOBAL_EVNT_SIZ(#n) (for n=0)

Bit	Label	Access	Description
31	EVNTINTRPTMASK	R/W	Event Interrupt Mask (EvtIntMask). When set to '1', this prevents the interrupt from being generated. However, even when the mask is set, the events are queued. DEFAULT VALUE: 0x0
30:16	reserved_30_16	RO	Reserved. DEFAULT VALUE: 0x0
15:0	EVENTSIZ	R/W	Event Buffer Size in bytes (EVNTSiz) Holds the size of the Event Buffer in bytes; must be a multiple of four. This is programmed by software once during initialization. The minimum size of the event buffer is 32 bytes. DEFAULT VALUE: 0x0

5.1.8 Global Event Buffer Count Register (GEVNTCOUNT(#n))

Table 5.9. GLOBAL_EVNT_COUNT(#n) (for n=0)

Bit	Label	Access	Description
31	EVNT_HANDLER_BUSY	R/W	Event Handler Busy Device software event handler busy indication. The controller sets this bit when the interrupt line is asserted due to pending events. Software clears this bit (with 1'b1) when it has finished processing the events (along with updating the EVNTCOUNT in this register). The controller does not raise the interrupt line for a new event unless this bit is cleared. Note: When Interrupt moderation is disabled (that is, DEVICE_IMODI = 0), this bit is ignored. DEFAULT VALUE: 0x0
30:16	reserved_30_16	RO	Reserved DEFAULT VALUE: 0x0
15:0	EVNTCOUNT	R/W	Event Count (EVNTCount) - When read, returns the number of valid events in the Event Buffer (in bytes). - When written, hardware decrements the count by the value written. - When Interrupt moderation is enabled (that is, DEVICE_IMODI!= 0), the interrupt line gets de-asserted when the first write happens on this register to decrement the count. When Interrupt moderation is disabled (that is, DEVICE_IMODI = 0), the Interrupt line continues to get asserted until the event count becomes zero (no-moderation behavior). DEFAULT VALUE: 0x0

5.1.9 Device Configuration Register (DCFG)

Table 5.10. DEVICE_CFG (DCFG)

Bit	Label	Access	Description
31:25	reserved_31_25	RO	Reserved DEFAULT VALUE: 0x0
24	StopOnDisconnect	RO	Reserved DEFAULT VALUE: 0x0
23	IgnStrmPP	R/W	IgnoreStreamPP This bit only affects stream-capable bulk endpoints. When this bit is set to '0' and the controller receives a Data Packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NumP field set to 0 and PP set to 0 for IN endpoints, the controller attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal. - When the host is setting PP=0 even though it has not finished the stream, or - When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. When this bit is set to '1', the controller ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NumP=0, PP=0). This can enhance the performance when the

Bit	Label	Access	Description
			device system bus bandwidth is low or the host responds to the controller's ERDY transmission very quickly. DEFAULT VALUE: 0x0
22	LPMCAP	R/W	LPM Capable The application uses this bit to control the LPM capabilities of the USB3 controller. If the controller operates as a non-LPM-capable device, it cannot respond to LPM transactions. 1'b0: LPM capability is not enabled 1'b1: LPM capability is enabled DEFAULT VALUE: 0x0
21:17	NUMP	R/W	Number of Receive Buffers. This bit indicates the number of receive buffers to be reported in the ACK TP. The USB3 controller uses this field for non-control endpoints if GRXTHRCFG.UsbRxPktCntSel is set to '0'. The application can program this value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency. For an OUT endpoint, this field controls the number of receive buffers reported in the NumP field of the ACK TP transmitted by the controller. Note: This bit is used in host mode when Debug Capability is enabled. DEFAULT VALUE: 0x4
16:12	INTRNUM	R/W	Interrupt number Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated. DEFAULT VALUE: 0x0
11:10	reserved_10_11	RO	Reserved DEFAULT VALUE: 0x2
9:3	DEVADDR	R/W	Device Address. The application must perform the following: - Program this field after every SetAddress request. - Reset this field to zero after USB reset. DEFAULT VALUE: 0x0
2:0	DEVSPD	R/W	Device Speed. Indicates the speed at which the application requires the controller to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed and is based on the speed of the USB host to which the controller is connected. 3'b100: SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) Values: 0x4 (SuperSpeed): SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) 0x0 (HighSpeed): High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 0x1 (FullSpeed): Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) DEFAULT VALUE: If super speed is enabled then the value is 0x4 otherwise the default value is 0x0.

5.1.1. Device Control Register (DCTL)

Table 5.11. DEVICE_CTL (DCTL)

Bit	Label	Access	Description
31	RUN_STOP	R/W	Run/Stop The software writes 1 to this bit to start the device controller operation. To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DEVICE_STS.DevCtrlHlt bit when the controller is idle and the lower layer finishes the disconnect process. The Run/Stop bit must be used in following cases as specified: - After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. - The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DEVICE_CTL[8:5] to 5 before reasserting the Run/Stop bit. - When the USB or Link is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device controller, it must set this bit to start the device controller. The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect: <i>30 ms:</i> For Super Speed, when the device state is Suspended, Idle, Transmit, or Receive. <i>10 ms:</i> For High Speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions) For Full Speed/Low Speed when the device state is Suspended, Idle, or not Idle/Suspended (performing trans- actions) To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration. DEFAULT VALUE: 0x0
30	CSFTRST	R/W1S	Core Soft Reset Resets all clock domains as follows: - This bit clears the interrupts and all the CSRs except GLOBAL_USB2_PHY_CFGn registers, GLOBAL_USB3_PIPE_CTLn registers, DEVICE_CFG, DEVICE_CTL, DEVICE_EVNT_EN, and DEVICE_STS registers. - All module state machines (except the SoC bus target unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. - Any transactions on the SoC bus controller are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately.

Bit	Label	Access	Description
			The application can write this bit at any time to reset the controller. This is a self-clearing bit; the controller clears this bit after all necessary logic is reset in the controller, which may take several clocks depending on the current state of the controller. Once this bit is cleared, the software must wait at least three PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation. Note: Programming this field with random data causes side effect. Bit Bash register testing is not recommended. DEFAULT VALUE: 0x0
29	reserved_29	RO	Reserved1 DEFAULT VALUE: 0x0
28:24	HIRDTHRES	R/W	HIRD Threshold (HIRD_Thres) The controller asserts output signals utmi_l1_suspend_n and utmi_sleep_n based on this signal: The controller asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true: - HIRD value is greater than or equal to the value in DEVICE_CTL.HIRD_Thres[3:0] - HIRD_Thres[4] is set to 1'b1. The controller asserts utmi_sleep_n on L1 when one of the following is true: - If the HIRD value is less than HIRD_Thres[3:0] or - HIRD_Thres[4] is set to 1'b0. Note: This field must be set to '0' during Super Speed operation mode. DEFAULT VALUE: 0x0
23	APPL1RES	R/W	LPM Response Programmed by ApplicationWhen LPM Errata is disabled: Bit [23]: LPM Response Programmed by Application(AppL1Res) Handshake response to LPM token specified by deviceapplication. Response depends on DEVICE_CFG.LPMCap. - DEVICE_CFG.LPMCap is 1'b0. The controller always responds with Timeout (that is, no response). - DEVICE_CFG.LPMCap is 1'b1. The controller response is based on the value of this bit. 1'b0: The controller responds with an ACK upon a successful LPM transaction, which requires all the following are satisfied. There are no PID/CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL). No data is pending in the Transmit FIFO and OUT endpoints not in flow-controlled state (else NYET). 1'b1: The controller responds with an ACK upon a successful LPM, independent of transmit FIFO status and OUT endpoint flow control state. The LPM transaction is successful if all the following are satisfied. - There are no PID/CRC5 errors in both the EXT token and the

Bit	Label	Access	Description
			LPM token (else ERROR) A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL) Bits [22:20]: Reserved DEFAULT VALUE: 0x0
22			Reserved Field: Yes
23:20	LPM_NYET_thres	R/W	LPM NYET Threshold When LPM Errata is enabled: Bits [23:20]: LPM NYET Response Threshold(LPM_NYET_thres) Handshake response to LPM token specified by device application. Response depends on DEVICE_CFG.LPMCap. - DEVICE_CFG.LPMCap is 1'b0 - The controller always responds with Timeout (that is, no response). - DEVICE_CFG.LPMCap is 1'b1 - The controller responds with an ACK on successful LPM transaction, which requires that all the following are satisfied: - There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). - No data is pending in the TxFIFO and RxFIFO is empty (else NYET). - The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0] DEFAULT VALUE: 0xf
19	KeepConnect	R/W	Keep Connect When '1', this bit enables the save and restore programming model by preventing the controller from disconnecting from the host when DEVICE_CTL.RunStop is set to '0'. It also enables the Hibernation Request Event to be generated when the link goes to U3 or L2. The device controller disconnects from the host when DEVICE_CTL.RunStop is set to '0'. This bit indicates whether to preserve this behavior ('0'), or if the controller must not disconnect when RunStop is set to 0 ('1'). This bit also prevents the LTSSM from automatically going to U0/L0 when the host requests resume from U3/L2. Note: If Hibernation is disabled, that is, GLOBAL_CTL[1].GblHibernationEn = 0, this bit is tied to zero. DEFAULT VALUE: 0x0
18	L1HibernationEn	R/W	L1HibernationEn When this bit is set along with KeepConnect, the device controller generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DEVICE_CTL.HIRD_Thres. The controller does not exit the LPM L1 state until software writes Recovery into the DEVICE_CTL.ULStChngReq field. This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1. Note: If Hibernation is disabled, that is, GLOBAL_CTL[1].GblHibernationEn = 0, this bit is tied to zero. DEFAULT VALUE: 0x0
17	CRS	R/W	Controller Restore State (CRS) This command is similar to the USB_CMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DEVICE_STS.RSS to '1'. When the

Bit	Label	Access	Description
			controller has finished the restore process, it sets DEVICE_STS.RSS to '0'. Note: When read, this field always return to '0'. DEFAULT VALUE: 0x0
16	CSS	R/W	Controller Save State (CSS) This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DEVICE_STS.SSS to '1'. When the controller has finished the save process, it sets DEVICE_STS.SSS to '0'. Note: When read, this field always returns to '0'. DEFAULT VALUE: 0x0
15:13	reserved_15_13	RO	Reserved DEFAULT VALUE: 0x0
12	INITU2ENA	R/W	Initiate U2 Enable 1'b0: May not initiate U2 (default) 1'b1: May initiate U2 On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received. If DEVICE_CTL[11] (AcceptU2Ena) is 0, the link immediately exits U2 state. DEFAULT VALUE: 0x0
11	ACCEPTU2ENA	R/W	Accept U2 Enable 1'b0: Reject U2 except when Force_LinkPM_Accept bit is set (default) 1'b1: Controller accepts transition to U2 state if nothing is pending on the application side On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command. DEFAULT VALUE: 0x0
10	INITU1ENA	R/W	Initiate U1 Enable 1'b0: May not initiate U1 (default); 1'b1: May initiate U1. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE) and clears this bit when ClearFeature(U1_ENABLE) is received. If DEVICE_CTL[9] (AcceptU1Ena) is 0, the link immediately exits U1 state. DEFAULT VALUE: 0x0
9	ACCEPTU1ENA	R/W	Accept U1 Enable 1'b0: Controller rejects U1 except when Force_LinkPM_Accept bit is set (default) 1'b1: Controller accepts transition to U1 state if nothing is pending on the application side On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command. DEFAULT VALUE: 0x0
8:5	ULSTCHNGREQ	WO	ULSTCHNGREQ Software writes this field to issue a USB/Link state change request. A change in this field indicates a new request to the controller. If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB/Link State in

Bit	Label	Access	Description
			DEVICE_STS. These bits are self-cleared on the MAC Layer exiting suspended state. If software is updating other fields of the DEVICE_CTL register and not intending to force any link state change, then it must write a 0 to this field. SS Compliance mode is normally entered and controlled by the remote link partner. Refer to the USB 3.0 specification. Alternatively, you can force the local link directly into compliance mode, by resetting the SS link with the RUN/STOP bit set to zero. If you then write '10' to the USB/Link State Change field and '1' to RUN/STOP, the link goes to compliance mode. Once you are in compliance, you may alternately write zero and '10' to this field to advance the compliance pattern. In SS mode: Value Requested Link State Transition/Action 0 No Action 4 SS.Disabled 5 Rx.Detect 6 SS.Inactive 8 U3 exit request 10 Compliance - Others: Reserved In HS/FS/LS mode: - ValueRequested USB state transition 8 Remote wakeup request - Others: Reserved The Remote wakeup request must be issued 2 us after the device goes into suspend state (DEVICE_STS[21:18] is 3 - refer to Table "Fields for Register: DEVICE_STS"). Note: After coming out of hibernation, software must write 8 (Recovery) into this field to confirm exit from the suspended state. DEFAULT VALUE: 0x0
4:1	TSTCTL	R/W	Test Control 4'b000: Test mode disabled 4'b001: Test_J mode 4'b010: Test_K mode 4'b011: Test_SE0_NAK mode 4'b100: Test_Packet mode 4'b101: Test_Force_Enable - Others: Reserved DEFAULT VALUE: 0x0
0	reserved_0	RO	Reserved DEFAULT VALUE: 0x0

5.1.10 Device Event Enable Register (DEV TEN)

Table 5.12. DEVICE_EVNT_EN

Bit	Label	Access	Description
31:17	reserved_31_17	RO	Reserved DEFAULT VALUE: 0x0
16	ECCERREN	RO	ECC Error Enable. If this bit is set to 1, the controller reports an ECC error to the software when an uncorrectable ECC occurs internally. DEFAULT VALUE: 0x0
15	reserved_15	RO	Reserved DEFAULT VALUE: 0x0
14	L1WKUPEVTEN	R/W	L1 Resume Detected Event Enable. DEFAULT VALUE: 0x0
13	StopOnDisconnectEn	RO	Reserved DEFAULT VALUE: 0x0
12	VENDEVTSTRCDEN	R/W	Vendor Device Test LMP Received Event (VndrDevTstRcvdEn) DEFAULT VALUE: 0x0
11	reserved_11	RO	Reserved DEFAULT VALUE: 0x0
10	reserved_10	RO	Reserved DEFAULT VALUE: 0x0
9	ERRTICERREVTEN	R/W	Erratic Error Event Enable DEFAULT VALUE: 0x0
8	L1SUSPEN	R/W	L1 Suspend Event Enable DEFAULT VALUE: 0x0
7	SOFTEVTEN	R/W	Start of (u)frame DEFAULT VALUE: 0x0
6	U3L2L1SuspEn	R/W	U3/L2 or U3/L2L1 Suspend Event Enable. DEFAULT VALUE: 0x0
5	HibernationReqEvtEn	R/W	This bit enables/disables the generation of the HibernationRequest Event. DEFAULT VALUE: 0x0
4	WKUPEVTEN	R/W	U3/L2 or U3/L2L1 Resume Detected Event Enable. DEFAULT VALUE: 0x0
3	ULSTCNGEN	R/W	USB/Link State Change Event Enable DEFAULT VALUE: 0x0
2	CONNECTDONEEV TEN	R/W	Connection Done Enable DEFAULT VALUE: 0x0
1	USBRSTEV TEN	R/W	USB Reset Enable DEFAULT VALUE: 0x0
0	DISSCONNEVTEN	R/W	Disconnect Detected Event Enable DEFAULT VALUE: 0x0

5.1.11 Device Status Register (DSTS)

Table 5.13. DEVICE_STS(DSTS)

Bit	Label	Access	Description
31:30	reserved_31_30	RO	Reserved DEFAULT VALUE: 0x0
29	DCNRD	RO	Device Controller Not Ready The bit indicates that the controller is in the process of completing the state transitions after exiting from hibernation. To complete the state transitions, it takes 256 bus clock cycles from the time DEVICE_CTL[31].Run/Stop is set. During hibernation, if the UTMI/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DEVICE_CTL[31].Run/Stop to '1' and wait for this bit to be de-asserted to zero before processing DEVICE_STS.USBLnkSt. This bit is valid only when USB3_EN_PWROPT is set to 2 and GLOBAL_CTL[1].GblHibernationEn =1. DEFAULT VALUE: 0x0
28	SRE	R/W1C	Save Restore Error. Currently not supported. DEFAULT VALUE: 0x0
27:26	reserved_27_26	RO	Reserved DEFAULT VALUE: 0x0
25	RSS	RO	RSS Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller finishes the restore process, it completes the command by setting DEVICE_STS.RSS to '0'. DEFAULT VALUE: 0x0
24	SSS	RO	Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DEVICE_STS.SSS to '0'. DEFAULT VALUE: 0x0
23	COREIDLE	RO	Core Idle The bit indicates that the controller finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero. Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the controller and does not hold a static value. DEFAULT VALUE: 0x1
22	DEVCTRLHLT	RO	Device Controller Halted This bit is set to 0 when the Run/Stop bit in the DEVICE_CTL register is set to 1. The controller sets this bit to 1 when, after SW sets Run/Stop to 0, the controller is idle, and the lower layer finishes the disconnect process. When Halted=1, the controller does not generate Device events. Note: - The controller does not set this bit to 1 if GLOBAL_EVNT_COUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GLOBAL_EVNT_COUNTn) while it is waiting for this bit to be set to 1. - When Interrupt Moderation is enabled, there could be delay in raising the interrupt line when the event count is non-zero. Software should read the GLOBAL_EVNT_COUNT register

Bit	Label	Access	Description
			directly and acknowledge them. DEFAULT VALUE: 0x1
21:18	USBLNKST	RO	USBLNKST. USB/Link State <i>In SS mode: LTSSM State</i> 4'h0: U0 4'h1: U1 4'h2: U2 4'h3: U3 4'h4: SS_DIS 4'h5: RX_DET 4'h6: SS_INACT 4'h7: POLL 4'h8: RECOV 4'h9: HRESET 4'ha: CMPPLY 4'hb: LPBK 4'hf: Resume/Reset <i>In HS/FS/LS mode:</i> 4'h0: On state 4'h2: Sleep (L1) state 4'h3: Suspend (L2) state 4'h4: Disconnected state (Default state) 4'h5: Early Suspend state (valid only when Hibernation is disabled, GLOBAL_CTL[1].GblHibernationEn = 0) 4'he: Reset (valid only when Hibernation is enabled, GLOBAL_CTL[1].GblHibernationEn = 1) 4'hf: Resume (valid only when Hibernation is enabled, GLOBAL_CTL[1].GblHibernationEn = 1) The link state Resume/Reset indicates that the controller received a resume or USB reset request from the host while the link was in hibernation. Software must write '8' (Recovery) to the DEVICE_CTL.ULStChngReq field to acknowledge the resume/reset request. When Hibernation is enabled, GLOBAL_CTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when Device_CTL[31].Run/Stop set to '1' and DEVICE_STS[29].DCNRD = 0. The Early Suspend link state is an early indication of device suspend in HS/FS. The link state changes to Early Suspend after detecting bus idle for 3 ms. - In HS operation, this is an indication that the USB bus (that is, LineState) has been idle (SE0) for 3 ms. However, it does not confirm whether the next process is Suspend or Reset. The device checks the bus again after pull up enable delay and if the line state indicates Suspend (full speed J), then the device waits for an additional time (~3 ms) to indicate the actual Suspend state. - In FS operation, this is an indication that the USB bus (that is, LineState) has been in idle (J) for 3 ms. The device waits for an additional time (~3 ms of Idle) to indicate the actual Suspend state. DEFAULT VALUE: 0x4
17	RXFIFOEMPTY	RO	RxFIFO Empty. DEFAULT VALUE: 0x1

Bit	Label	Access	Description
16:3	SOFFN	RO	Frame/Microframe Number of the Received SOF. When the controller is operating at SuperSpeed, [16:3] indicates the uframe/ITP number. When the controller is operating at high-speed, [16:6] indicates the frame number. [5:3] indicates the micro frame number. When the controller is operating at full-speed, [16:14] is not used. Software can ignore these 3 bits. [13:3] indicates the frame number. Note: After power-on reset, the controller generates the microframe number internally for every 125 us if the USB host has not issued SOF/ITP yet. During P3 state, the duration of SOFFN is based on the suspend_clk frequency. DEFAULT VALUE: 0x0
2:0	CONNECTSPD	RO	Connected Speed (ConnectSpd) Indicates the speed at which the USB3 controller has come up after speed detection through a chirp sequence. 3'b100: SuperSpeed (PHY clock is running at 125 or 250 MHz). 3'b000: High-speed (PHY clock is running at 30 or 60 MHz). 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz) Low-speed is not supported for devices using a UTMI+ PHY. Values: 0x4 (SuperSpeed): SuperSpeed (PHY clock is running at 125 or 250 MHz). 0x0 (HighSpeed): High-speed (PHY clock is running at 30 or 60 MHz). 0x1 (FullSpeed): Full-speed (PHY clock is running at 30 or 60 MHz). DEFAULT VALUE: When USB2 is only enabled then the value is 0x1 otherwise it is 0x4

5.1.12 Device Active USB Endpoint Enable Register (DALEPENA)

Table 5.14. DEVICE_ACT_EP_EN (DALEPENA)

Bit	Label	Access	Description
31:0	USBACTEP	R/W	USBACTEP USB Active Endpoints (USBActEP) This field indicates if a USB endpoint is active in the current configuration and interface. It applies to USB IN endpoints 0.15 and OUT endpoints 0.15, with one bit for each of the 32 possible endpoints. Even numbers are for USB OUT endpoints, and odd numbers are for USB IN endpoints, as follows: - Bit[0]: USB EP0-OUT - Bit[1]: USB EP0-IN - Bit[2]: USB EP1-OUT - Bit[3]: USB EP1-IN The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USBReset. Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set

			these bits. DEFAULT VALUE: 0x0
--	--	--	--

5.1.13 Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2)

Table 5.15. DEPCMDPAR2[0:7]

Bit	Label	Access	Description
31:0	PARAMETER	R/W	PARAMETER DEFAULT VALUE: 0x0

5.1.14 Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)

Table 5.16. DEPCMDPAR1[0:7]

Bit	Label	Access	Description
31:0	PARAMETER	R/W	PARAMETER DEFAULT VALUE: 0x0

5.1.15 Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)

Table 5.17. DEPCMDPAR0[0:7]

Bit	Label	Access	Description
31:0	PARAMETER	R/W	PARAMETER DEFAULT VALUE: 0x0

5.1.16 Device Physical Endpoint-n Command Register (DEPCMD)

Table 5.18. DEPCMD[0:7]

Bit	Label	Access	Description
31:16	COMMANDPARAM	R/W	Command Parameters or Event Parameters Command Parameters (CommandParam) when this register is written: For Start Transfer command: [31:16]: StreamID. The USB StreamID assigned to this transfer - For Start Transfer command applied to an isochronous endpoint [31:16]: StartMicroFramNum: Indicates the (micro)frame number to which the first TRB applies. For Update Transfer, End Transfer, and Start New Configuration commands. [22:16]: Transfer Resource Index (XferRscIdx). The hardware-assigned transfer resource index for the transfer, which was returned in response to the Start Transfer command. The application software-assigned transfer resource index for a Start New Configuration command. DEFAULT VALUE: 0x0
15:12	CMDSTATUS	R/W	Command Completion Status (CmdStatus) DEFAULT VALUE: 0x0
11	HIPRI_FORCERM	R/W	HighPriority/ForceRM (HiPri_ForceRM) HighPriority: Only valid for Start Transfer command ForceRM: Only valid for End Transfer command

Bit	Label	Access	Description
			ClearPendIN: Only valid for Clear Stall command. Software sets this bit to clear any pending IN transaction (on that endpoint) stuck at the lower layers when a Clear Stall command is issued. DEFAULT VALUE: 0x0
10	CMDACT	R/W	Command Active (CmdAct) Software sets this bit to 1 to enable the device endpoint controller to execute the generic command. The device controller sets this bit to 0 when the CmdStatus field is valid, and the endpoint is ready to accept another command. This does not imply that all the effects of the previously issued command have taken place. DEFAULT VALUE: 0x0
9	reserved_9	RO	Reserved DEFAULT VALUE: 0x0
8	CMDIOC	R/W	CMDIOC Command Interrupt on Complete (CmdIOC) When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command. Note that this interrupt is mapped to DEPCFG.IntrNum. When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG.IntrNum in the current command. Note: This field must not set to 1 if the DEVICE_CTL.RunStop field is 0. DEFAULT VALUE: 0x0
7:4	reserved_7_4	RO	Reserved DEFAULT VALUE: 0x0
3:0	CMDTYP	R/W	Command Type Specifies the type of command the software driver is requesting the controller to perform. 00h: Reserved 01h: Set Endpoint Configuration - 64 or 96-bit Parameter 02h: Set Endpoint Transfer Resource Configuration - 32-bit Parameter 03h: Get Endpoint State - No Parameter Needed 04h: Set Stall - No Parameter Needed 05h: Clear Stall (see Set Stall) - No Parameter Needed 06h: Start Transfer - 64-bit Parameter 07h: Update Transfer - No Parameter Needed 08h: End Transfer - No Parameter Needed 09h: Start New Configuration - No Parameter Needed DEFAULT VALUE: 0x0

5.1.17 GLOBAL_TX_FIFO_SIZE (GTXFIFOSIZ(#n))

Table 5.19. GLOBAL_TX_FIFO_SIZE

Bit	Label	Access	Description
31:16	TXFSTADDR_N	R/W	Transmit FIFO RAM Start Address This field represents the memory start address for TXFIFO. DEFAULT VALUE: 0x0
15:0	TXFDEP_N	R/W	TXFIFO Depth This field contains the depth of the TxFIFO.

5.1.18 GLOBAL_RX_FIFO_SIZE (GRXFIFOSIZ(#n))

Table 5.20. GLOBAL_RX_FIFO_SIZE

Bit	Label	Access	Description
31:16	RXFSTADDR_N	R/W	Receive FIFO RAM Start Address This field represents the memory start address for RXFIFO. DEFAULT VALUE: 0x0
15:0	RXFDEP_N	R/W	RXFIFO Depth This field contains the depth of the RxFIFO.

5.2. USB IP 2/3 PHY Control Status Register Information

The following table lists the USB IP 2.0 PHY Control Status Register Information.

Table 5.21. USB IP 2/3 PHY Control Status Register Information

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value																		
0x10040	u2phy_csr_ctrl0_reg	RW	[6:0]	PLL_MULTIPLIER	USB3 PHY register. Output clock divide ratio. Reserved for test and debug.	7'b0110010																		
		RO	[7]	Reserved	Reserved.	1'b0																		
		RW	[10:8]	Reserved	Reserved.	3'b000																		
		RW	[13:11]	PLL_SSCRANGE	USB3 PHY register. PLL Spread Spectrum Clock (SSC) range	3'b000																		
					<table><tr><th>Value</th><th>Description</th></tr><tr><td>000</td><td>-5100 ppm</td></tr><tr><td>001</td><td>-4700 ppm</td></tr><tr><td>010</td><td>-4200 ppm</td></tr><tr><td>011</td><td>-1200 ppm</td></tr><tr><td>100</td><td>-1200 ppm</td></tr><tr><td>101</td><td>-1200 ppm</td></tr><tr><td>110</td><td>-1200 ppm</td></tr><tr><td>111</td><td>-880 ppm</td></tr></table>	Value	Description	000	-5100 ppm	001	-4700 ppm	010	-4200 ppm	011	-1200 ppm	100	-1200 ppm	101	-1200 ppm	110	-1200 ppm	111	-880 ppm	
		Value	Description																					
000	-5100 ppm																							
001	-4700 ppm																							
010	-4200 ppm																							
011	-1200 ppm																							
100	-1200 ppm																							
101	-1200 ppm																							
110	-1200 ppm																							
111	-880 ppm																							
RW	[15:14]	PLL_CLKREF_SEL	USB3 PHY register for PLL Reference Clock Selection.	2'b11																				
			<table><tr><th>Value</th><th>Description</th></tr><tr><td>00</td><td>Choose the single-ended off chip clock PLL_CLKREF_EXT as reference</td></tr><tr><td>01</td><td>Reserved.</td></tr><tr><td>10</td><td>Choose the off-chip differential clock REFIN_CLK_EXT_P/M or on chip differential clock REFIN_CLK_INT_P/M as reference.</td></tr><tr><td>11</td><td>Choose the single-ended on chip clock PLL_CLKREF_INT as reference.</td></tr></table>	Value	Description	00	Choose the single-ended off chip clock PLL_CLKREF_EXT as reference	01	Reserved.	10	Choose the off-chip differential clock REFIN_CLK_EXT_P/M or on chip differential clock REFIN_CLK_INT_P/M as reference.	11	Choose the single-ended on chip clock PLL_CLKREF_INT as reference.											
Value	Description																							
00	Choose the single-ended off chip clock PLL_CLKREF_EXT as reference																							
01	Reserved.																							
10	Choose the off-chip differential clock REFIN_CLK_EXT_P/M or on chip differential clock REFIN_CLK_INT_P/M as reference.																							
11	Choose the single-ended on chip clock PLL_CLKREF_INT as reference.																							

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value						
		RW	[16]	PLL_CLK_WORD_QUARTER_EN	USB3 PHY register for quarter word clock output enable. Use this register for test and debug purposes.	1'b1						
		RW	[17]	PLL_CLK_WORD_HALF_EN	USB3 PHY register for half word clock output enable. Use this register for test and debug purposes.	1'b1						
		RW	[18]	PLL_CLK_WORD_EN	USB3 PHY register word clock output enable. Use this register for test and debug purposes.	1'b1						
		RW	[19]	PLL_CLKREF_DIV3_EN	USB3 PHY register for input ref divided by 3 enable. This register must be set to 1'b0 if the reference clock is 60 MHz.	1'b0						
		RW	[20]	PLL_CLKREF_SSC_EN	Reserved.	1'b0						
		RW	[21]	PLL_CLKREF_EXT_EN	Reference clock source selection <table><tr><td>1</td><td>Select off chip reference clock</td></tr><tr><td>0</td><td>Select on chip ref clock</td></tr></table>	1	Select off chip reference clock	0	Select on chip ref clock	1'b1		
		1	Select off chip reference clock									
		0	Select on chip ref clock									
		RW	[22]	PLL_SSC_EN	PLL SSC enable control <table><tr><td>1</td><td>Enable</td></tr><tr><td>0</td><td>Disable</td></tr></table>	1	Enable	0	Disable	1'b1		
		1	Enable									
		0	Disable									
		RW	[23]	PLL_CLKREF_SSP_EN	Reserved.	1'b1						
RW	[24]	PLL_PI_CLK_SEL	PI output clock enable For test and debug purposes.	1'b1								
RW	[25]	PLL_OUTPUT_R_EN	'PLL_CLK_I(Q)_R_P' output enable For test and debug purposes.	1'b1								
RW	[26]	PLL_OUTPUT_L_EN	'PLL_CLK_I(Q)_L_P' output enable For test and debug purposes.	1'b1								
RW	[27]	PLL_OUTPUT_EN	'PLL_CLKREF_SSC_OUT','PLL_CLK_I_L(R)_P', 'PLL_CLK_Q_L(R)_P' output enable For test and debug purposes.	1'b1								
0x1008c	u2phy_csr_ctrl1_reg	RW	[31:28]	PLL_FSEL	PLL start up timing selection: <table><tr><td>0011</td><td>Minimum start up time</td></tr><tr><td>0111</td><td>Medium start up time</td></tr><tr><td>1111</td><td>Largest start up time</td></tr></table>	0011	Minimum start up time	0111	Medium start up time	1111	Largest start up time	4'b0111
		0011	Minimum start up time									
		0111	Medium start up time									
1111	Largest start up time											
		RW	[7:0]	pipew_beacon_min_freq	Used by the detect beacon function. Bits [7:0] of the configuration value for the detect beacon filter. The register value to program = maximum beacon period (ns) / 2*miphy_clk_osc period (ns).	8'b00101100						
		RW	[11:8]	pipew_beacon_min_freq	Used by the detect beacon function. Bits [11:8] of the configuration value for the detect beacon filter. The register value to program is equal to: maximum beacon period (ns) / 2*miphy_clk_osc period (ns).	0x1						
		RW	[15:12]	Reserved	Reserved.	0x0						

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value
		RW	[16]	detect_status_invert	Detection polarity 1 Invert polarity 0 Normal polarity	1'b0
		RW	[17]	txdetectrx_pol	Detect polarity. 1 Positive 0 Negative	1'b0
		RW	[21:18]	refclk_div_pipe_clk	refclk_div_pipe_clk is equal to the reference clock period/pipe_clk period(250MHz), upper integer.	4'b0101
		RW	[22]	force_detect	Force RxStatus code to 'Present' state.	1'b0
		RW	[23]	use_new_txdetectrx	TX detect RX method selection 1 Use new TX detection. 0 Use old TX detection.	1'b1
		RW	[31:24]	pipew_time_cst_ns_force[7:0]	Protocol time constant for receiver detection in ns (lower bytes).	8'b10010000
0x10090	u2phy_csr_ctrl2_reg	RW	[7:0]	pipew_time_cst_ns_force[15:8]	Protocol time constant for receiver detection in ns (middle bytes).	8'b11000000
		RW	[11:8]	pipew_time_cst_ns_force[19:16]	Protocol time constant for receiver detection in ns, (upper bytes).	4'b0011
		RW	[15:12]	Reserved	Reserved.	4'b0000
		RW	[23:16]	pipew_time_cst_ns_threshold[7:0]	New TX detect RX threshold time constant in ns (lower bytes).	8'h01101010
		RW	[31:24]	pipew_time_cst_ns_threshold[15:8]	New TX detect RX threshold time constant in ns (middle bytes).	8'h00011000
0x10094	u2phy_csr_ctrl3_reg	RW	[3:0]	pipew_time_cst_ns_threshold[19:16]	New TX detect RX threshold time constant in ns (upper bytes).	4'b0000
		RW	[7:4]	Reserved	Reserved.	4'b0000
		RW	[15:8]	pipew_time_cst_ns_timeout[7:0]	New TX detect RX threshold timeout in ns (lower bytes).	8'b00011100
		RW	[23:16]	pipew_time_cst_ns_timeout[15:8]	New TX detect RX threshold timeout in ns (middle bytes).	8'b00011001
		RW	[27:24]	pipew_time_cst_ns_timeout[19:16]	New TX detect RX threshold timeout in ns (upper bytes).	4'b0010
		RW	[31:28]	Reserved	Reserved.	4'b0000
0x100AC	u2phy_csr_ctrl4_reg	RW	[3:0]	pipew_detect_p3_usb_rise_value_threshold_2	Used by USB3 TX detect RX function. Bits [3:0] of wait value defining a time interval before detecting that a signal is present on the partner chip RX inputs. The register value to program = wait_time(ns)/ pipe_clk period (ns).	4'b0000
		RW	[7:4]	Reserved	Reserved.	4'b0000
		RW	[8]	rxelecidle_ctrl	1 Pipe_rxvalid asserts rxelecidle regardless of sigdet (signal detected).	1'b1
		RW	[9]	ignore_analog_sigdet	1 Ignore analog sigdet if RX10b-8b decode is error free.	1'b0

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value
		RW	[10]	use_new_pipew_pwm_ctrl	1 Use the new power mode control logic.	1'b1
		RW	[15:11]	count_8b10b_decerr_deassert_rxvalid	If 8b10b decode error occurs in count_8b10b_decerr_deassert_rxvalid sequential cycles, pipe_rxvalid deasserts.	5'b01000
		RW	[16]	pcie_p2_pclk_en	Not used for USB.	1'b0
		RW	[17]	Reserved	Reserved.	1'b1
		RW	[18]	txalign_ready_enable	Not used for USB.	1'b1
		RW	[19]	pipew_phystatus_reach_for ce_en	Not used for USB. Used to force phystatus ready time. The time depends on pipew_time_power_reach_0 and pipew_time_power_reach_1.	1'b0
		RW	[23:20]	Reserve	Reserved.	4'b0000
		RW	[24]	force_sync	Not used for USB. Force clear the txebuff.	1'b0
		RW	[25]	sync_enable_txelecidle	1 Txelecidle falling edge clears the TX elastic buffer (txebuff).	1'b1
		RW	[26]	sync_enable_rate	1 Rate edge clears the TX elastic buffer (txebuff).	1'b0
		RW	[31:27]	Reserve	Reserved.	5'b00000
0x100C8	u2phy_csr_ctrl5_reg	RW	[7:0]	pipew_time_power_reach_0	Not used for USB. If power mode is changed, power reach is asserted after pipew_time_power_reach timeout. Because the PMA cannot assert power reach, the PCS generates power reach by itself.	8'b10010010
		RW	[15:8]	pipew_time_power_reach_1	Not used for USB. If power mode is changed, power reach is asserted after pipew_time_power_reach timeout. Because the PMA cannot assert power reach, the PCS generates power reach by itself.	8'b00111011
		RW	[16]	bist_en	Software starts bist test. bist_en is cleared by software if no bist error occurs. However, if bist error occurs, bist_en is cleared automatically.	1'b0
		RW	[17]	force_scram_off	During bist mode, the scramble and de-scramble feature is forced to turn off as the bist mode is only used for debugging purposes.	1'b0
		RW	[18]	clear_fsm	It is used to clear the BIST FSM for re-test purposes.	1'b0
		RW	[19]	bist_done	This bit is used to indicate bist done.	1'b0
		RW	[20]	bist_fail	This bit is used to indicate bist failure.	1'b0
		RW	[21]	pat_gen_data	This bit is used to indicate FSM pat_gen stays in PAT_GEN_DATA, only if pat_gen_data asserts. Software can clear bist_en.	1'b0

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value								
		RW	[22]	bist_mode	<table><tr><td>0</td><td>Normal function mode.</td></tr><tr><td>1</td><td>To enable Bist test mode.</td></tr></table>	0	Normal function mode.	1	To enable Bist test mode.	1'b0				
		0	Normal function mode.											
		1	To enable Bist test mode.											
		RW	[23]	error_inject	<table><tr><td>0</td><td>Indicates no error is injected. The bist error count must be 0 when bist test is completed.</td></tr><tr><td>1</td><td>Indicates 1 error is injected. The bist error count must be 1 when bist test is completed.</td></tr></table>	0	Indicates no error is injected. The bist error count must be 0 when bist test is completed.	1	Indicates 1 error is injected. The bist error count must be 1 when bist test is completed.	1'b0				
0	Indicates no error is injected. The bist error count must be 0 when bist test is completed.													
1	Indicates 1 error is injected. The bist error count must be 1 when bist test is completed.													
RW	[31:24]	pipew_bist_tseq_num_l	Indicates the lower 8 bits of TSEQ number in bist mode.	8'b00000000										
0x14010	u3phy_pma_debug_sel_misc_reg	RW	[7:0]	PMA_DEBUG_SELO	Indicates PMA signals debug flag select.	8'b00000000								
		RW	[15:8]	PMA_DEBUG_SEL1	Indicates PMA signals debug flag select.	8'b00000000								
		RW	[16]	USB3_PHY_CLK_SEL	This bit indicates PHY clock selection for USB3.	1'b0								
		RW	[17]	usb3phy_internal_rstn	This bit indicates internal reset for USB3.	1'b0								
		RW	[18]	usb3phy_refclk_sel	This bit indicates reference clock selection for USB3.	1'b0								
		RO	[19]	BIST_DONE_U3PHY	This bit indicates that the BIST is done for USB3.	1'b0								
		RO	[20]	BIST_FAIL_U3PHY	This bit indicates BIST failure for USB3.	1'b0								
		RO	[31:21]	Reserved	Reserved.	11'b0								
0x18000	u2phy_misc_ctrl0_reg	RW	[1 : 0]	LDO_1P0_ADJ	Indicates internal LDO Configure with different VDD voltage application as follows: <table><tr><td>00</td><td>1V VDD application</td></tr><tr><td>01</td><td>0.9V VDD application</td></tr><tr><td>10</td><td>0.8V VDD application</td></tr><tr><td>11</td><td>0.7V VDD application</td></tr></table>	00	1V VDD application	01	0.9V VDD application	10	0.8V VDD application	11	0.7V VDD application	2'b00
		00	1V VDD application											
		01	0.9V VDD application											
		10	0.8V VDD application											
		11	0.7V VDD application											
		RW	[4 : 2]	LS_CROSS_ADJ	Indicates low-speed signal cross point adjustment.	3'b000								
		RW	[6 : 5]	HS_RISE_TUNE	Indicates TX high-speed signal Rise/Fall time adjustment.	2'b00								
		RW	[7]	CDR_RST_SEL	Indicates CDR reset configuration pin.	1'b0								
		RW	[15 : 8]	GLOBAL_CONFIG	Indicates global configure control.	8'b00000000								
		RW	[18 : 16]	FS_CROSS_ADJ	Indicates full speed signal cross point adjustment.	3'b000								
		RW	[21 : 19]	REG14_ADJ	Indicates internal regulator output adjustment.	3'b000								
		RW	[23 : 22]	HS_REGOP8_ADJ	Indicates internal regulator output adjustment.	2'b00								
RW	[27 : 24]	SQL_SP_ADJ	Indicates RX squelch output adjustment.	4'b0000										
RW	[31 : 28]	VBUS_VLD_ADJ	Indicates VBUS valid detector adjustment.	4'b0000										
0x18004	u2phy	RW	[0]	CALIB_ONCE_EN	Indicates high-speed calibration	1'b0								

Offset	Register Name	Field Access	Bit Width	Field Name	Description	Reset Value
	misc_ctrl1_reg				configuration pin.	
		RW	[2 : 1]	HS_EMP_ADJ	Indicates high-speed pre-emphasis adjustment.	2'b00
		RW	[4 : 3]	SQL_CUR_ADJ	Indicates RX squelch current adjustment.	2'b00
		RW	[7 : 5]	PLLBW_SEL	Indicates PLL bandwidth configuration pins.	3'b000
		RW	[8]	BIST_EN_N	Indicates the pin that provides the BIST enabling signal.	1'b1
					0 BIST enabled, internal BIST test	
					1 Normal operating mode	
		RW	[9]	SCP_EN	Indicates DP /DM short protect enable.	1'b0
		RW	[10]	SEL_12_24M	The pins are used to select the input reference clock /Crystal OSC frequency.	1'b0
					0 The input internal clock /Crystal OSC frequency must be 12 MHz.	
					1 The input internal clock /Crystal OSC frequency must be 24 MHz.	
		RW	[11]	HS_LP_MODE_EN	Indicates high-speed low-power mode enable pin.	1'b0
		RW	[15 : 12]	SQL_VTH_ADJ	Indicates RX squelch threshold voltage adjustment.	4'b0000
		RW	[16]	HS_EMP_EN	Indicates high-speed pre-emphasis enable pin.	1'b0
		RW	[17]	RSTN_BYPASS	Indicates RESET Bypass configuration pin.	1'b0
		RW	[19 : 18]	CDR_BW_SEL	Indicates CDR bandwidth configuration pins.	2'b00
		RW	[23 : 20]	CDR_TIMING_SEL	Indicates CDR timing configuration pins.	4'b0000
		RW	[24]	SEL_INTERNALCLK	Indicates the pin is used to select the 12 MHz/24 MHz input clock source.	1'b0
					0 The clock is from crystal OSC.	
					1 The clock is from INTERNALCLK.	
		RW	[27 : 25]	XOSC_CUR_ADJ	Indicates XOSC current adjustment.	3'b000
		RW	[31 : 28]	DISC_ADJ	Indicates RX Disconnect threshold voltage adjustment.	4'b0000
0x18008	u2phy_misc_ctrl2_reset_status_reg	RW	[0]	XCLK12MOUTEN	Indicates internal crystal oscillator enable pin.	1'b0
					1 The internal crystal oscillator is always powered on even in USB Suspend mode.	
					0 The internal crystal oscillator will power done in USB Suspend mode.	
		RW	[3 : 1]	TEST_LOOP	Indicates the pins provide internal loop test for the PHY IP:	3'b000
					000 Loop 1 (Digital blocks + Analog blocks) (HS).	
					001 Loop 2 (Digital blocks + Analog blocks - Xmit/Rcv/Squelch)(HS).	
					010 Loop 3 (Digital blocks + Analog blocks - Xmit/Rcv/Squelch – CDR) (HS).	

Offset	Register Name	Field Access	Bit Width	Field Name	Description		Reset Value
					011	Loop 4 (Digital blocks + analog blocks - Xmit/Rcv/Squelch -CDR - EB) (HS).	
					100	Loop 1 (Digital blocks + analog blocks) (FS).	
					101	Loop 2 (Digital blocks + analog blocks - Xmit/Rcv/Dpll) (FS).	
					110	Reserved (FS).	
					111	No loop. Normal operating status.	
		RW	[4]	RX_LP_EN	Indicates RX low-power mode enable pin.		1'b0
		RW	[7 : 5]	REG20_ADJ	Indicates internal regulator output adjustment.		3'b000
		RW	[8]	USB2_PHY_CLK_SEL	This bit indicates PHY clock selection for USB2.		1'b1
		RW	[9]	usb2phy_internal_rst	This bit indicates reference clock selection for USB2.		1'b1
		RW	[10]	usb2phy_refclk_sel	This bit indicates reference clock selection for USB2.		1'b0
		RO	[11]	BIST_DONE_U2PHY	This bit indicates BIST done for USB2.		1'b0
		RO	[12]	BIST_ERR_U2PHY	This bit indicates BIST failure for USB2.		1'b0
		RO	[31 : 13]	Reserved	Reserved.		19'b0

6. Operation Details

6.1. Enumeration Sequence

The following lists the steps involved in the enumeration sequence:

- Power-on or soft reset
- USB reset event
- Connect done event
- SetAddress device request
- SetConfiguration device request

6.1.1. Power-On or Soft Reset

When the controller is powered on, the software must perform the following initialization:

1. Set DCTL.CSftRst field to 1. Wait until 0 is returned on reading. Device controller is reset.
2. Software must read the GSNPSID register to find the controller version and configure the driver for any version-specific features.
3. Program device speed bits and periodic frame interval using register DCFG.
4. Create a TRB for setup packet using register DEPCMD.
5. Enable physical endpoints 0 and 1 using register DALEPENA.
6. Set DCTL.Runstop to 1 to allow the device to attach to the host.

6.1.2. USB Reset

To initialize the controller as the device, you must perform the following steps:

1. Complete the control transfer and get the controller into the *setup a control setup TRB/start transfer* stage if the control transfer is still in progress.
2. Issue a DEPENDXFER command (08h) for any active transfer through the DEPCMD register.
3. Issue a DEPCSTALL (ClearStall)(04h) command for any endpoint in the STALL mode through the DEPCMD register.
4. Set DevAddr to 0 through the DCFG register.

6.1.3. Connect Done

Software is required to perform the following steps when this event is received:

1. Read DSTS register to obtain the connection speed.
2. Program the GCTL.RAMCLKSEL field to select the correct clock for the RAM clock domain. The GCTL.RAMCLKSEL field is reset to 0 after USB reset. You must reprogram this register each time on Connect Done.
3. Issue a DEPCMD.DEPCFG command for physical endpoints#0 and endpoints#1 from power-on reset.

Table 6.1. Maximum Packet Size Based on the Configured Speed

Mode	MaxPacketSize
SuperSpeed	512
High-Speed	64
Full-Speed	8/16/32/64
Low-Speed	8

4. Optional: You may choose to re-allocate the TX FIFO size by writing to the GTXFIFOSIZ(#n) register. Refer to the table above for the maximum packet size.

6.1.4. SetAddress Device Request

If the software receives a SetAddress request during the SETUP packet, perform the following steps:

1. Configure the DCFG register with the device address received as part of the SetAddress request when the SETUP packet is decoded.
2. Acknowledge the DEPCMD.CMDSTATUS by issuing a DEPCMD.CMDTY (Start Transfer) command pointing to a status TRB after receiving the XferNotReady (status) event.

6.2. Transfer Flow

6.2.1. Control-data/Bulk/Interrupt OUT Transfer Flow

This section describes the typical data OUT transfer flow for **Control-data/Bulk/Interrupt**. There are two scenarios that can happen:

1. Device software sets up data buffers before host request.
2. Device software sets up data buffers after host request.

The following figure shows how the device software sets up data buffers before the host requests for Control-data/bulk/interrupt transfer flow.

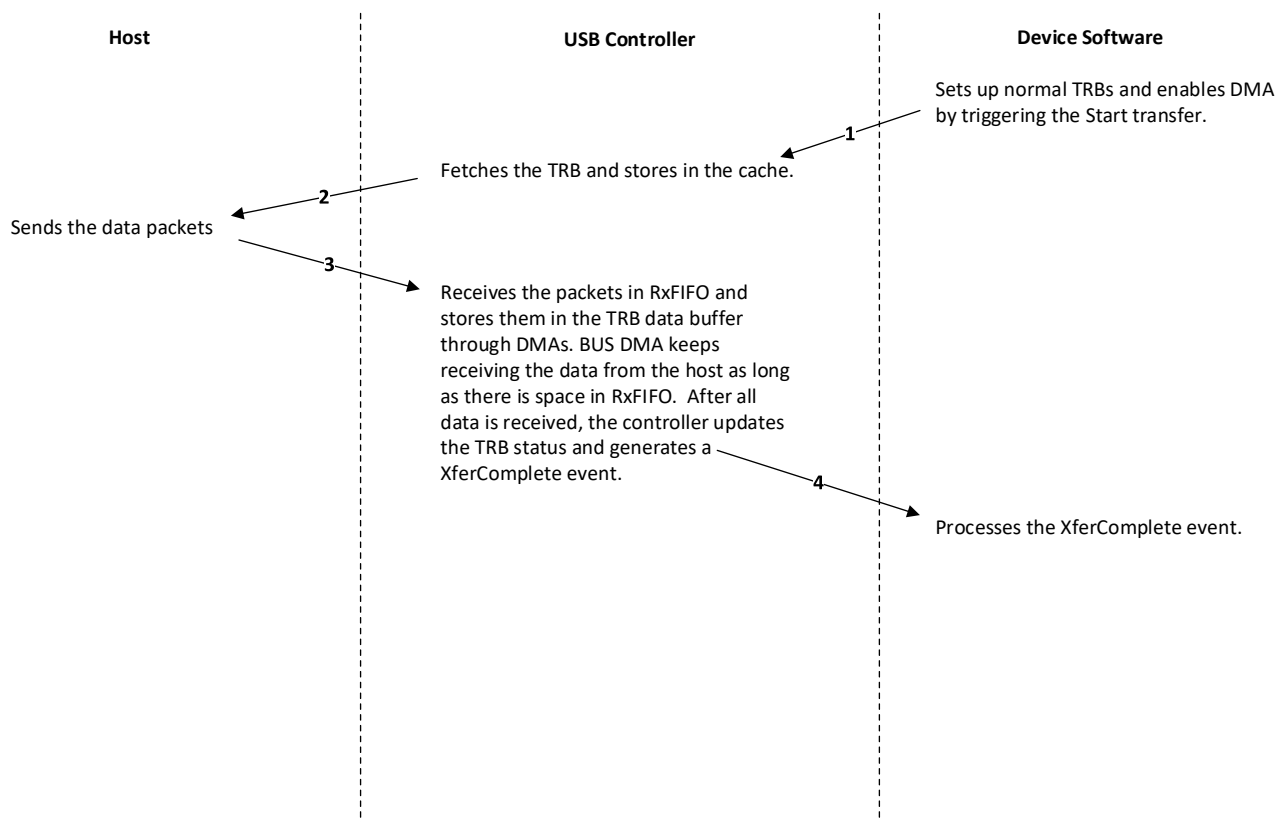


Figure 6.1. Device Software Sets Up Data Buffers Before Host Request for Control-data/Bulk/Interrupt Transfer

If the device software sets up data buffer **after the host request**, refer to the following figure for the transfer flow.

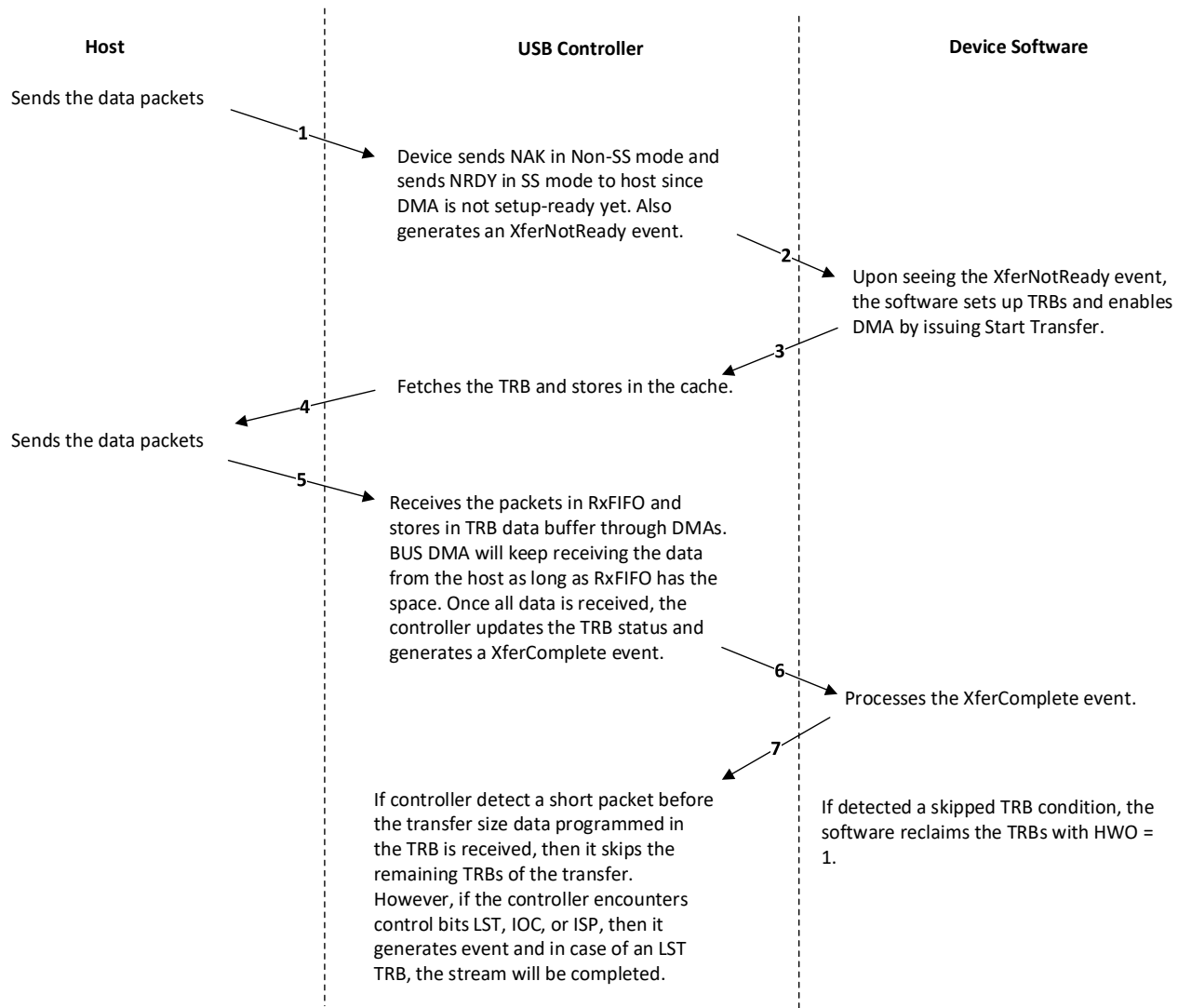


Figure 6.2. Device Software Sets Up Data Buffers After Host Requests for Control-data/Bulk/Interrupt Transfer

6.2.2. Isochronous OUT Transfer Flow

This section describes the Isochronous OUT data transfer flow. There are two scenarios that can happen during the Isochronous OUT transfer:

1. Device software sets up data buffers before host request.
2. Device software sets up data buffers after host request.

The following figure shows the Isochronous OUT transfer when the device software sets up data buffers before host request.

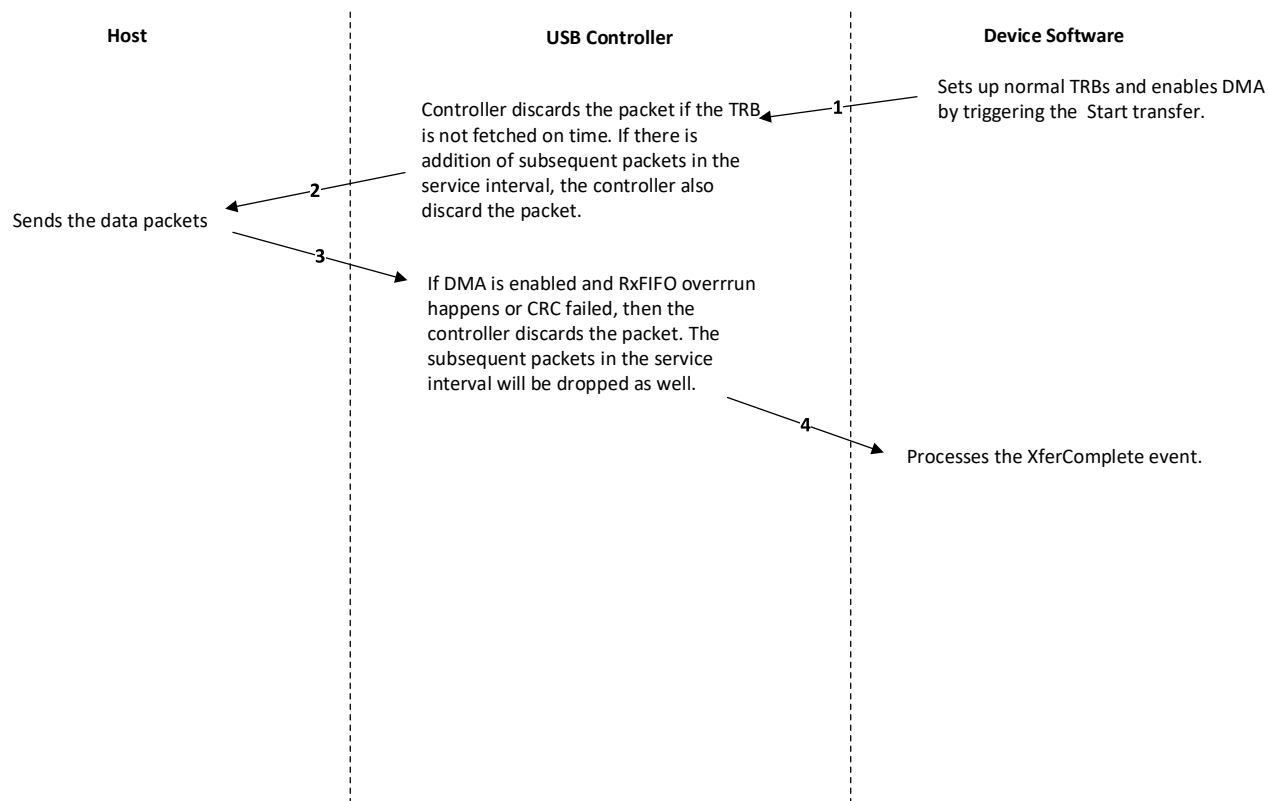


Figure 6.3. Isochronous OUT Transfer when Device Software Sets Up Data Buffers Before Host Request

However, if the device software sets up data buffer after the host request, refer to the following figure.

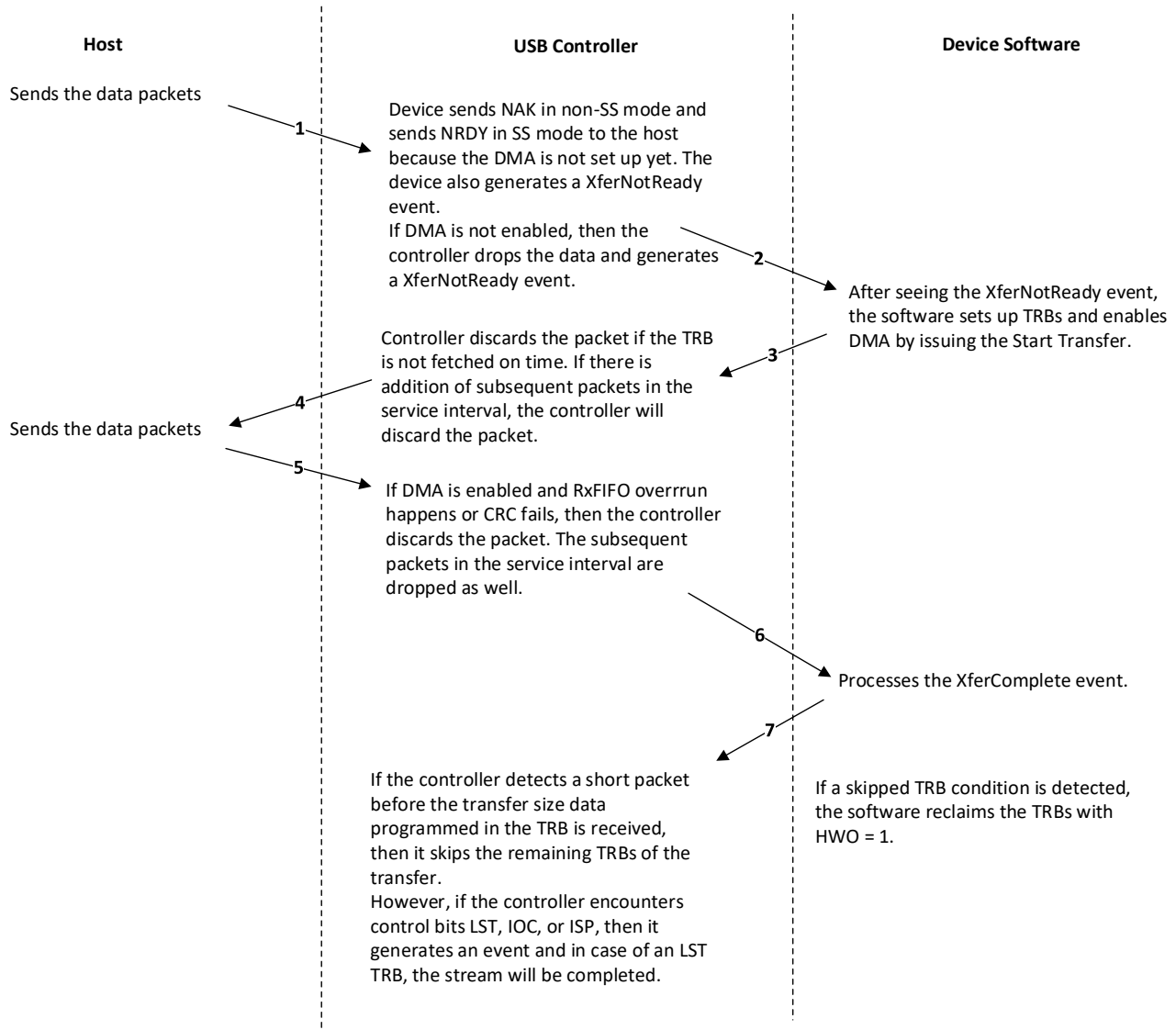


Figure 6.4. Isochronous OUT Transfer when Device Software Sets Up Data Buffers after Host Request

6.2.3. Control-data/ Bulk/Interrupt IN Transfer Flow

This section describes the data transfer flow for Control-data/Bulk and Interrupt. There are two possible scenarios that can happen:

1. The software sets up data buffers before a host request.
2. The software sets up data buffers after a host request.

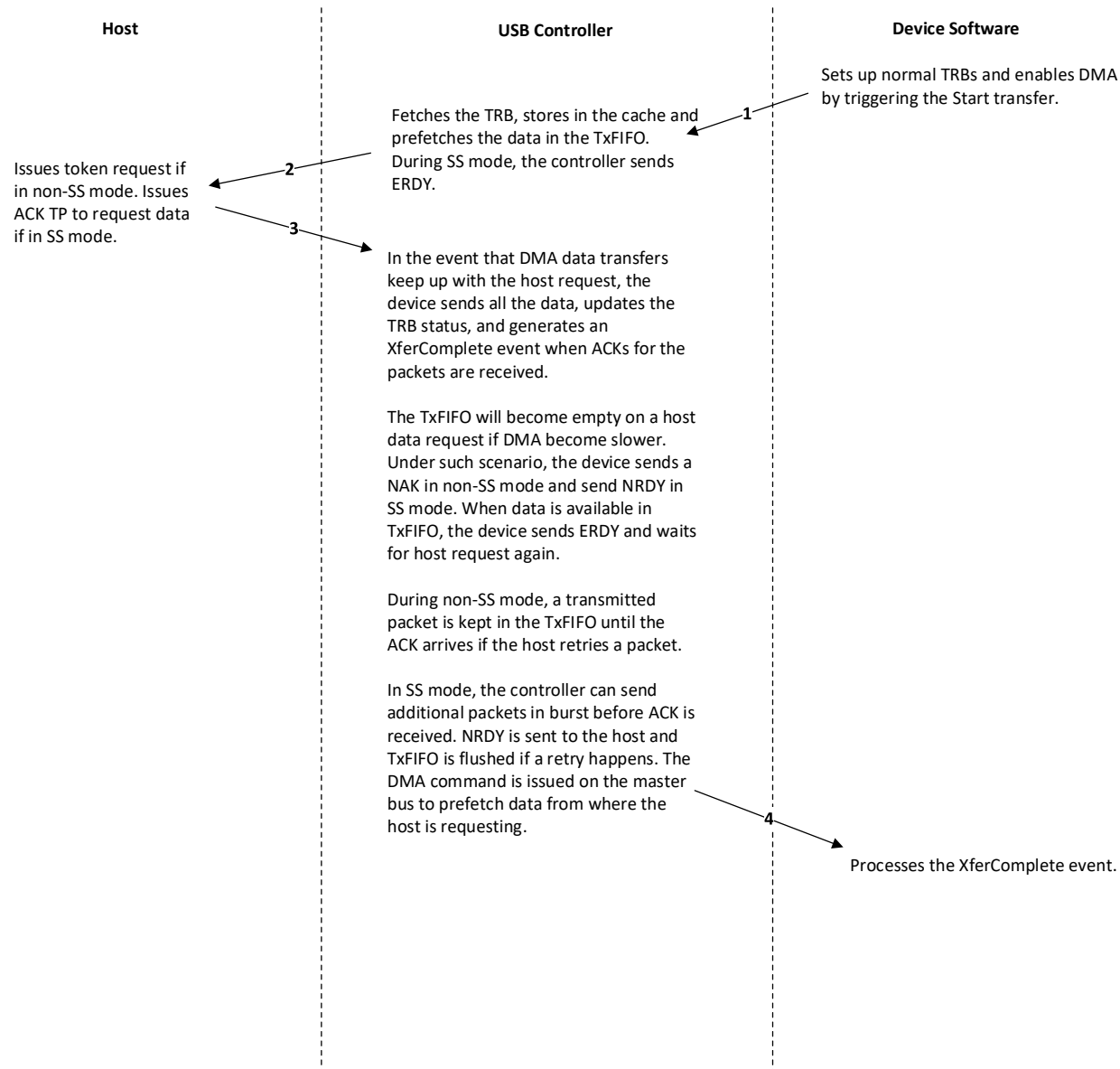


Figure 6.5. Control-data/Bulk/Interrupt IN Transfer when Device Software Sets Up Data Buffers before Host Request

However, if the device software sets up data buffers after host request, refer to the following figure.

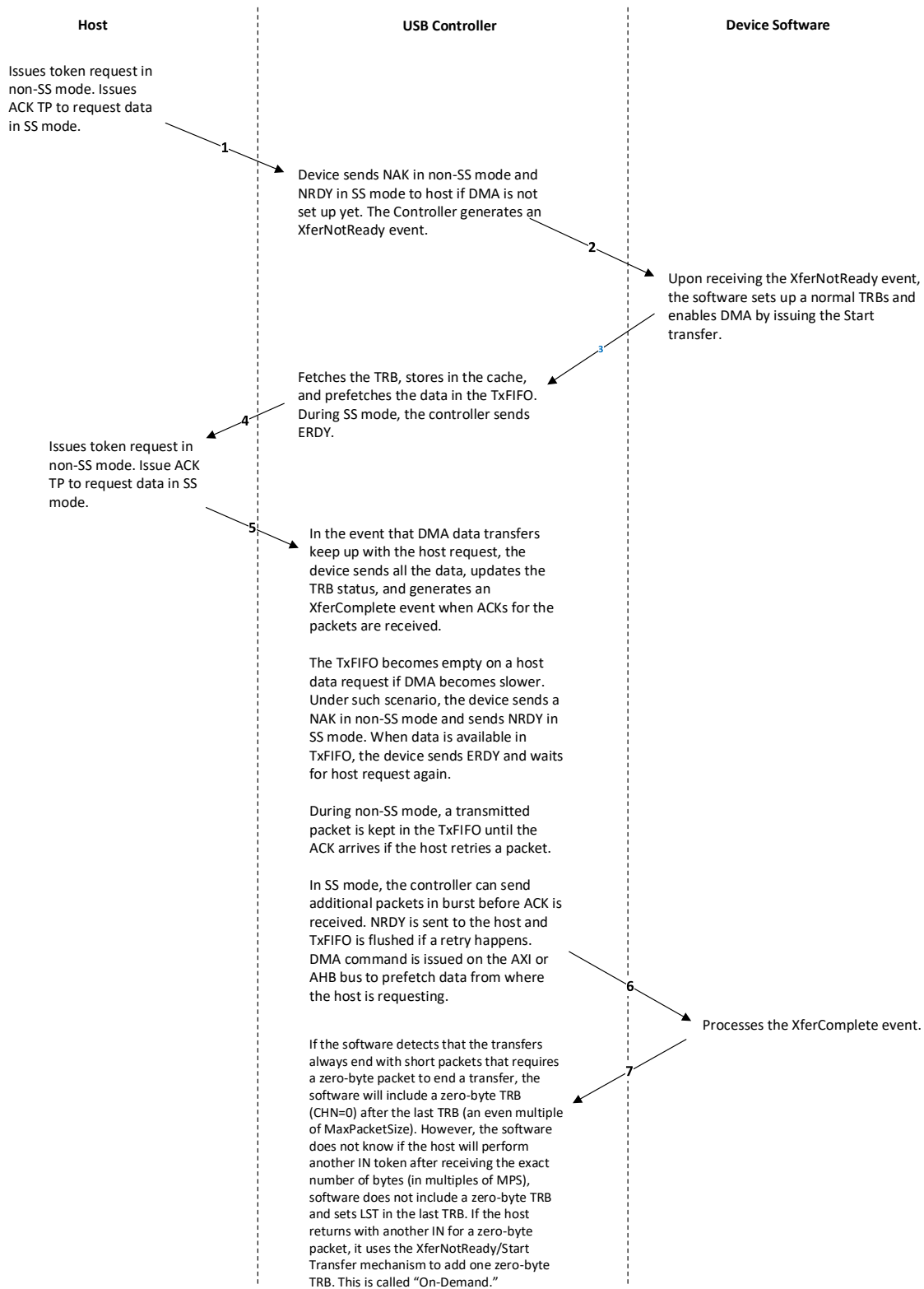


Figure 6.6. Control-data/Bulk/Interrupt IN Transfer when Device Software Sets Up Data Buffers after Host Request

6.2.4. Isochronous IN Transfer Flow

This section describes the Isochronous IN transfer flow. The following lists the difference between isochronous and non-isochronous IN transfer flow:

- When the software sets up data buffers before host request, the controller returns a zero-length packet in both non-SS and SS modes.
- When the software sets up data buffers after host request, the device returns zero length packet and generates an XferNotReady event.

The following figure shows the Isochronous IN transfer when the device software sets up data buffers before host request.

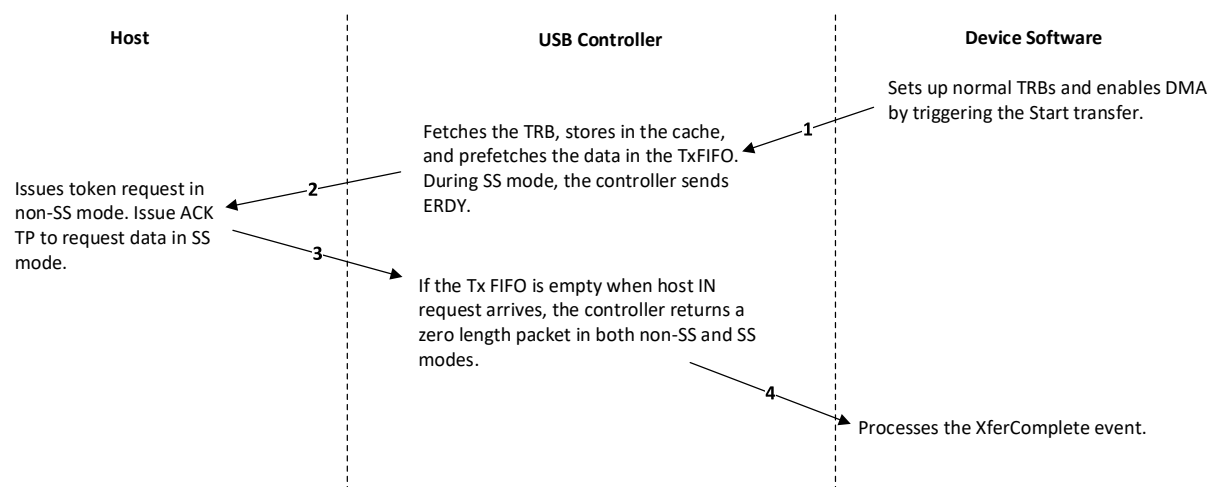


Figure 6.7. Isochronous IN Transfer when Device Software Sets Up Data Buffers before Host Request

For Isochronous IN transfer when the device software sets up data buffers after host request, refer to the following figure.

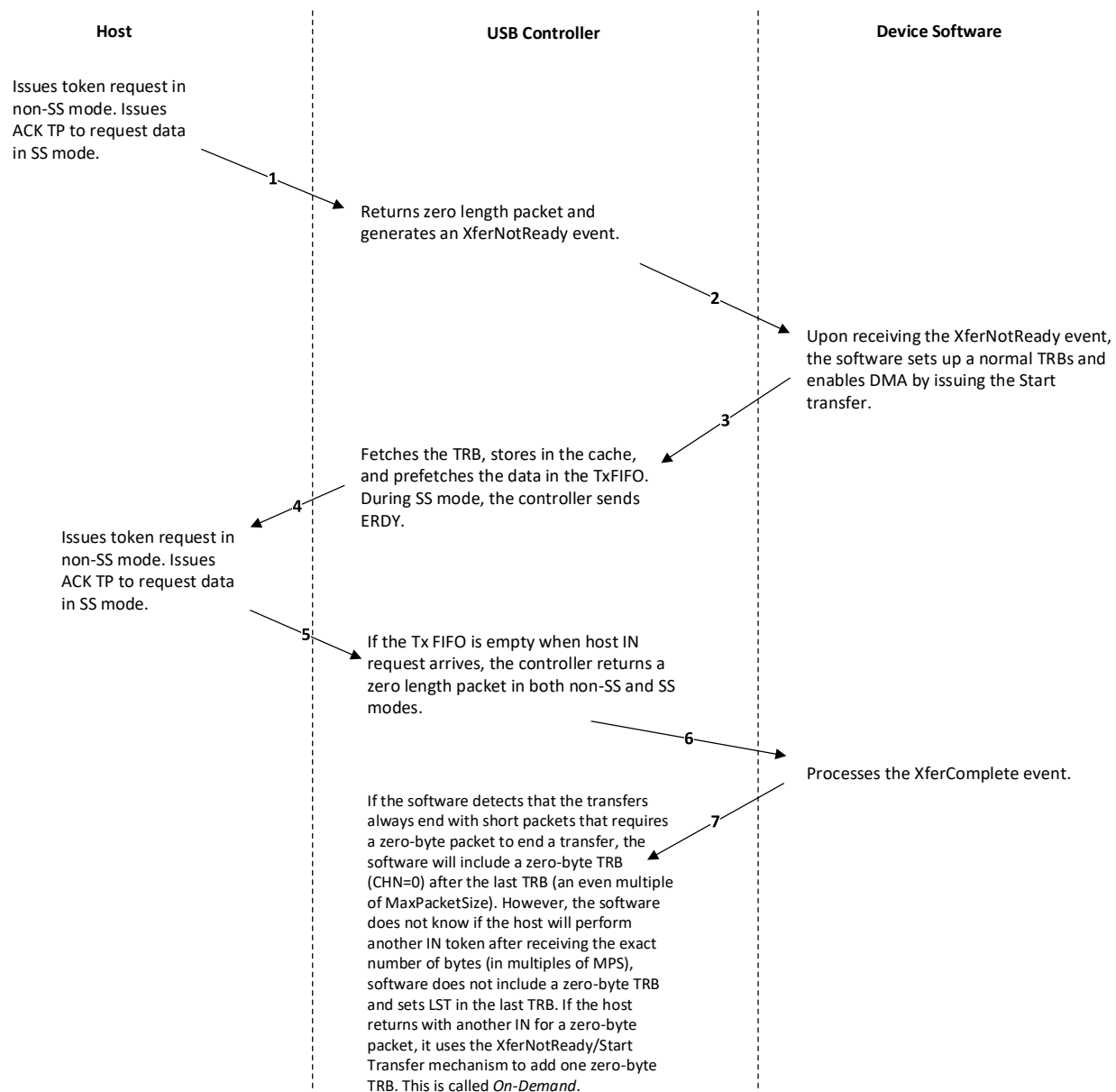


Figure 6.8. Isochronous IN Transfer when Device Software Sets Up Data Buffers after Host Request

7. Designing with the IP

This section provides information on how to generate the IP core using the Lattice Radiant software and how to run a synthesis. For more details on the Lattice Radiant software, refer to the [Lattice Radiant Software User Guide](#).

7.1. Generating and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into the device's architecture. The following steps describe how to generate the USB2.0/3.2 IP in the Lattice Radiant software:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **USB23** under **IP, USB23** category. The **Module/IP Block Wizard** opens, as shown in [Figure 7.1](#). Enter values in the **Component name** and the **Create in** fields, and click **Next**.

Module/IP Block Wizard

Generate Component from IP USB23 Version 3.1.1
This wizard will guide you through the configuration, generation and instantiation of this Module/IP. Enter the following information to get started.

Component name:

Create in:

This will automatically create a folder for USB inside the C:/home/

Figure 7.1. Module/IP Block Wizard

3. In the next **Module/IP Block Wizard** window, customize the selected USB 2.0/3.2 IP using the drop-down lists and check boxes. [Figure 7.2](#) shows a configuration example of the USB 2.0/3.2 IP core. For details on the configuration options, refer to the *Attributes summary* section.

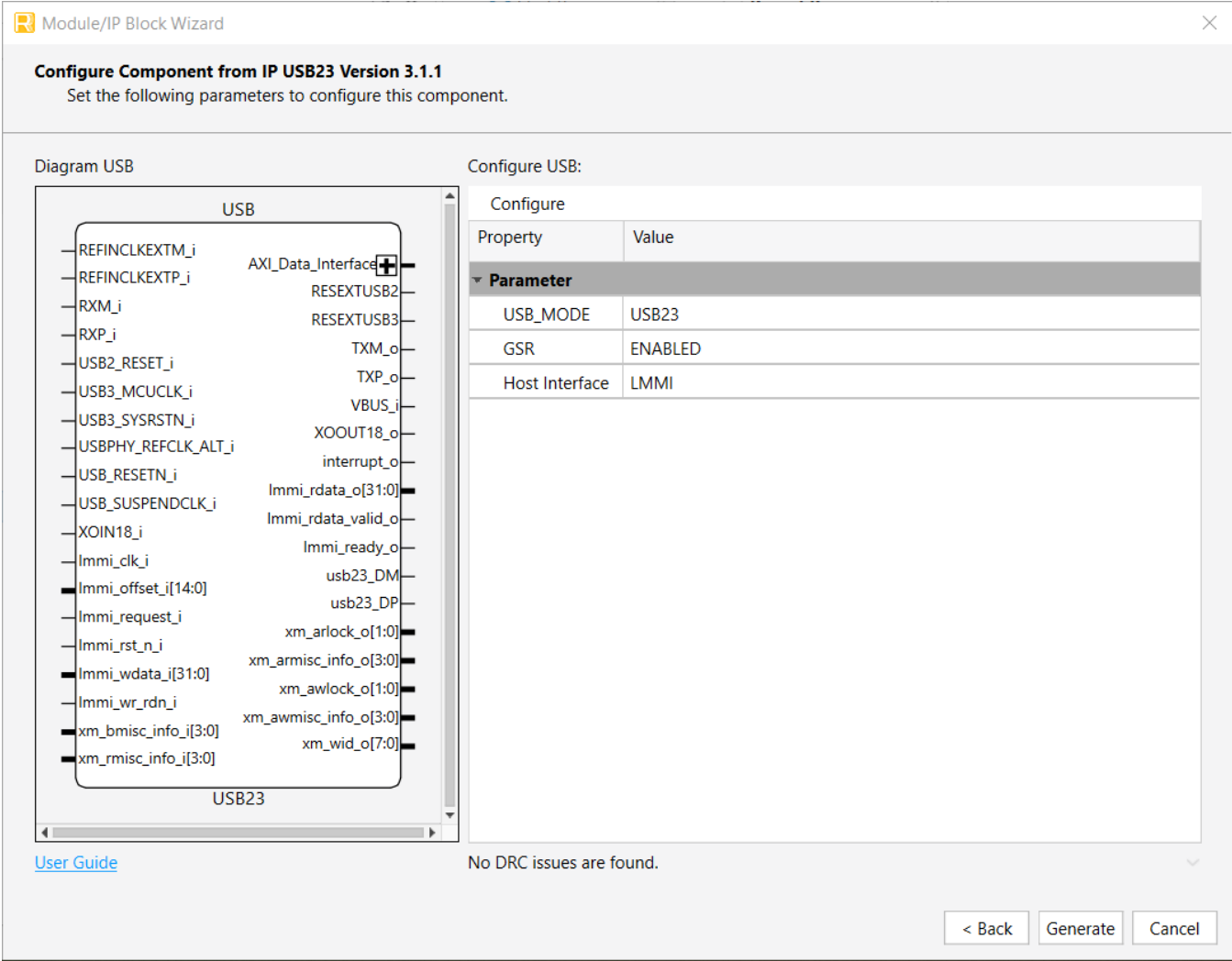


Figure 7.2. IP Configuration

4. Click **Generate**. The **Check Generated Result** dialog box opens with design block messages and results, as shown in Figure 7.3.

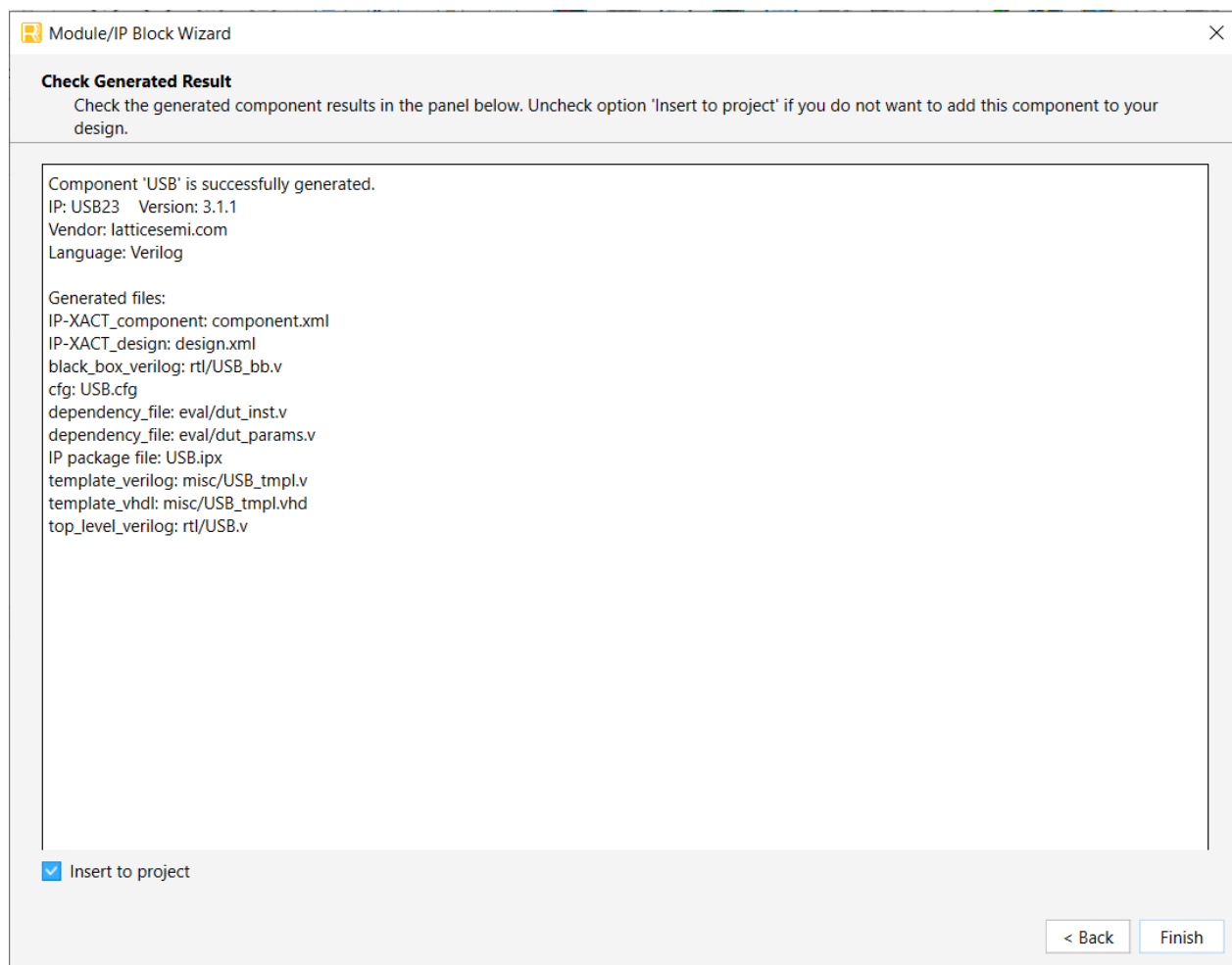


Figure 7.3. Check Generated Result

- Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields, as shown in [Figure 7.1](#).

7.1.1. Generated Files and File Structure

The generated USB2.0/3.2 module package includes the black box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An RTL example of the top-level reference source file (<Component name>.v) that can be used as an instantiation template for the module is also provided. You may also use this example as the starting template for your top-level design. The following table lists the generated files.

Table 7.1. Generated Files

Generated Files	Description
<Component name>.ipx	This file contains information on the files associated with the generated IP.
<Component name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	This file contains the ipxact:component information of the IP.
design.xml	This file documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an RTL example of the top file that instantiates the module.

Generated Files	Description
rtl/<Component name>_bb.v	This file provides the synthesis closed box.
misc/<Component name>_tmpl.v misc/<Component name>_tmpl.vhd	These files provide instance templates for the module.
eval/dut_inst.v eval/dut_params.v eval/nx33u_cmp_revb_evn_bd_rd_des.zip eval/application/winusb_async_app_rel_v_1_6.zip eval/software/usb23hal_6eps_v2_2.zip	This folder includes the following files required for the USB23 example design: <i>nx33u_cmp_revb_evn_bd_rd_des.zip</i>—Contains the Radiant and Propel™ software design file. <i>application/winusb_async_app_rel_v_1_6.zip</i>—Windows Host application package. <i>software/usb23hal_6eps_v2_2.zip</i>—Contains all the C code for HAL driver.

7.1.2. Design Implementation

The USB23 Benchmark design that is provided already has the correct pin assignments. However, you can add or modify the design to include additional steps that specify analog properties, pin assignments, and timing constraints. You can add and edit the constraints using the Device Constraint Editor or by creating a PDC file manually.

Device Constraint Editor

For more information on how to use the Device Constraint Editor, refer to the [Lattice Radiant Software 2025.1 User Guide](#).

Manual PDC File Creation

The USB23 Benchmark design already has the *.pdc* (post synthesis constraint file) file added in the Radiant software and defines the I/O pins according to the schematic design. If you wish to add other I/O pin assignments, you can define different types of constraints such as pins, clocks, and other timing paths.

7.1.3. Timing Constraints

See the [Clock Interface](#) section for details about the clock ports of the USB2.0/3.2 IP.

For more information on timing constraints, refer to the [Lattice Radiant Timing Constraints Methodology Application Note \(FPGA-AN-02059\)](#).

8. Example Design

This section describes the USB23 Benchmark example design that demonstrates the functionality of the USB 2.0/3.2 IP. All the functionality of the USB 2.0/3.2 IP is supported in the Eval folder example design.

The USB23 Benchmark example design allows you to compile, simulate, and test the USB 2.0/3.2 IP on the following Lattice evaluation board:

- CrossLinkU-NX Evaluation Board RevB (LIFCL-33U-EVN)

8.1. Overview of the Example Design and Features

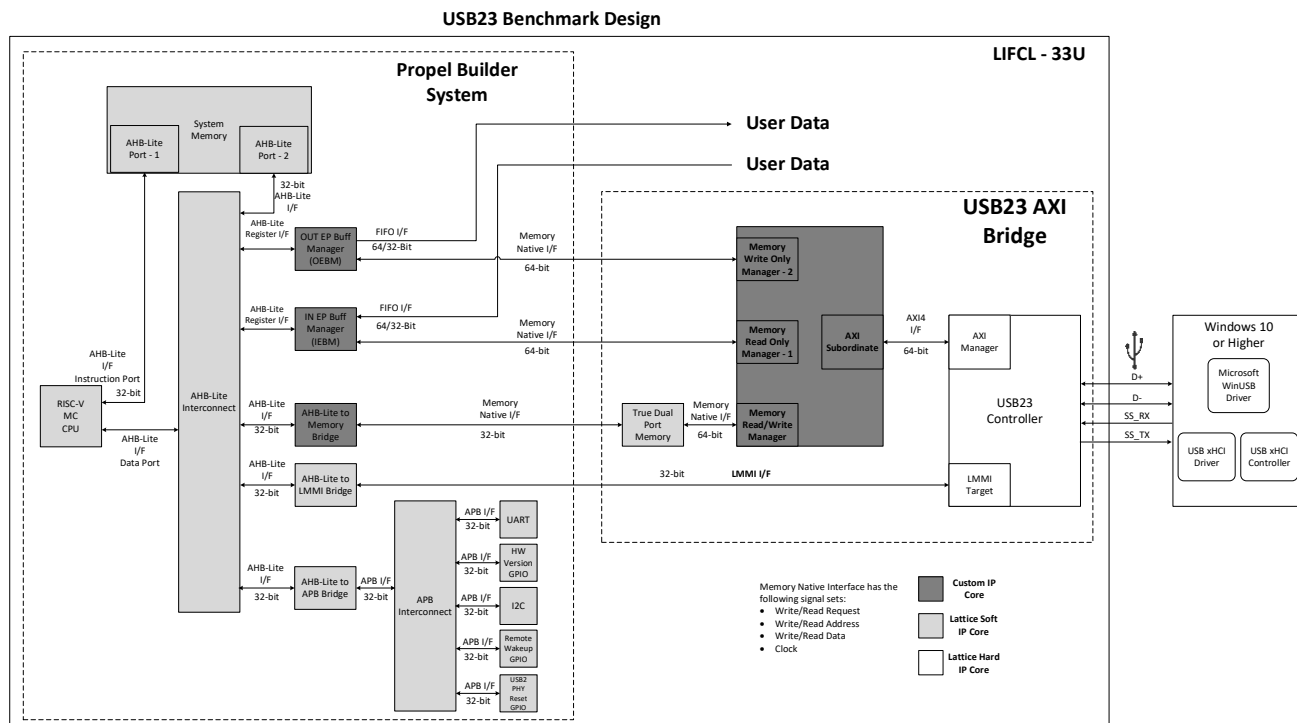
The following lists the key features of the example design:

- Supports USB device enumeration¹
- Supports USB transfers such as Bulk, Interrupt, Control, and Isochronous
- Provides UART for debugging purposes
- Provides an example design for hardware testing, debugging (through UART), or release modes

8.2. Example Design Components

The Eval_Top example design includes the following blocks:

- USB23 controller
- RISC-V MC microcontroller processor
- AHBL-to-LMMI converter
- UART
- GPIO
- System memory
- PLL
- AHBL interconnect
- APB interconnect
- AHBL-to-APB converter
- AHBL-to-memory bridge
- In FIFO controller
- IN endpoint buffer manager (IEBM)
- OUT endpoint buffer manager (OEBM)
- Out FIFO controller
- USB23 to AXI bridge



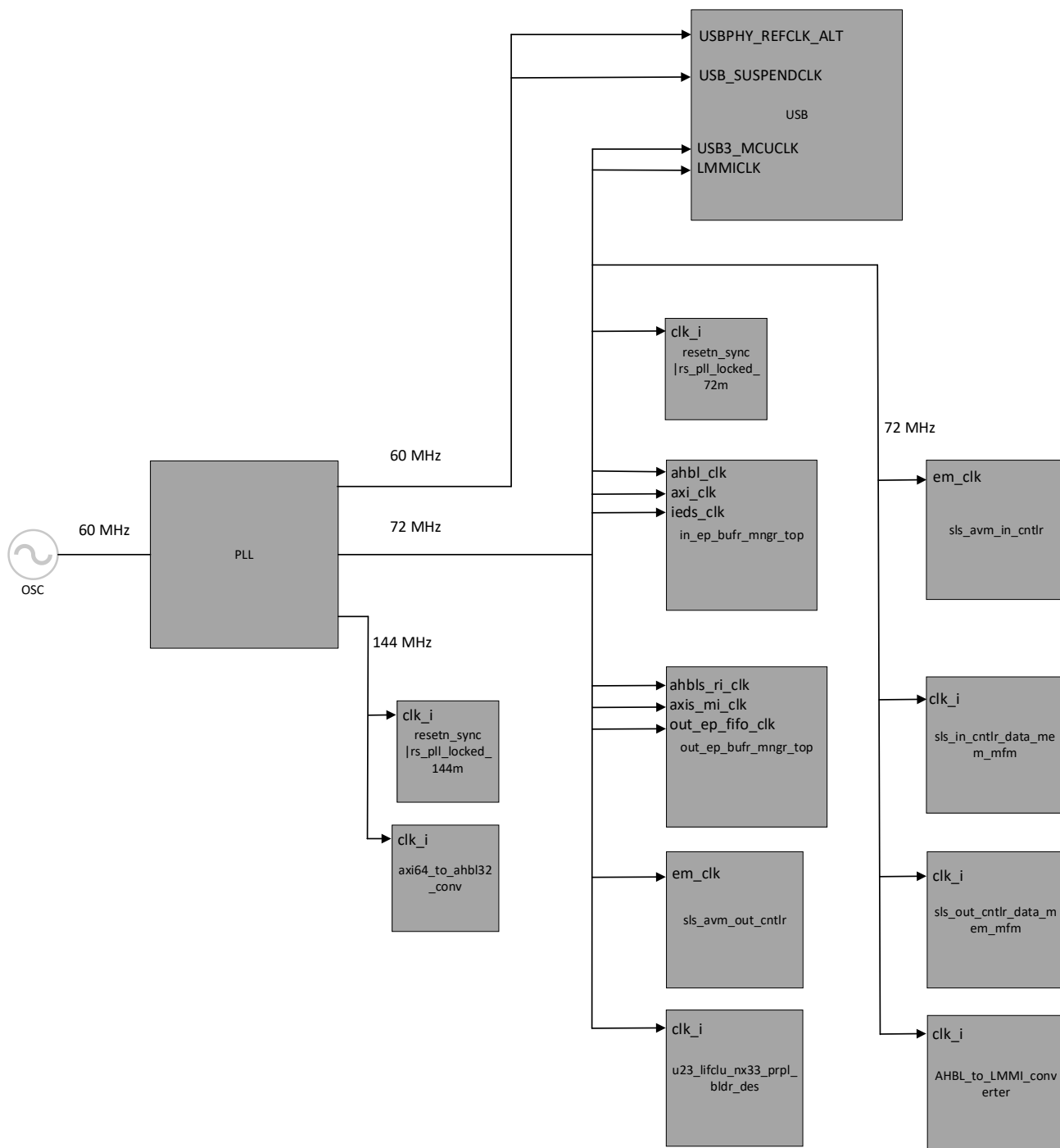


Figure 8.2. Clocking Diagram of the Example Design

8.2.1. Description on Data Flow from USB Device to Host

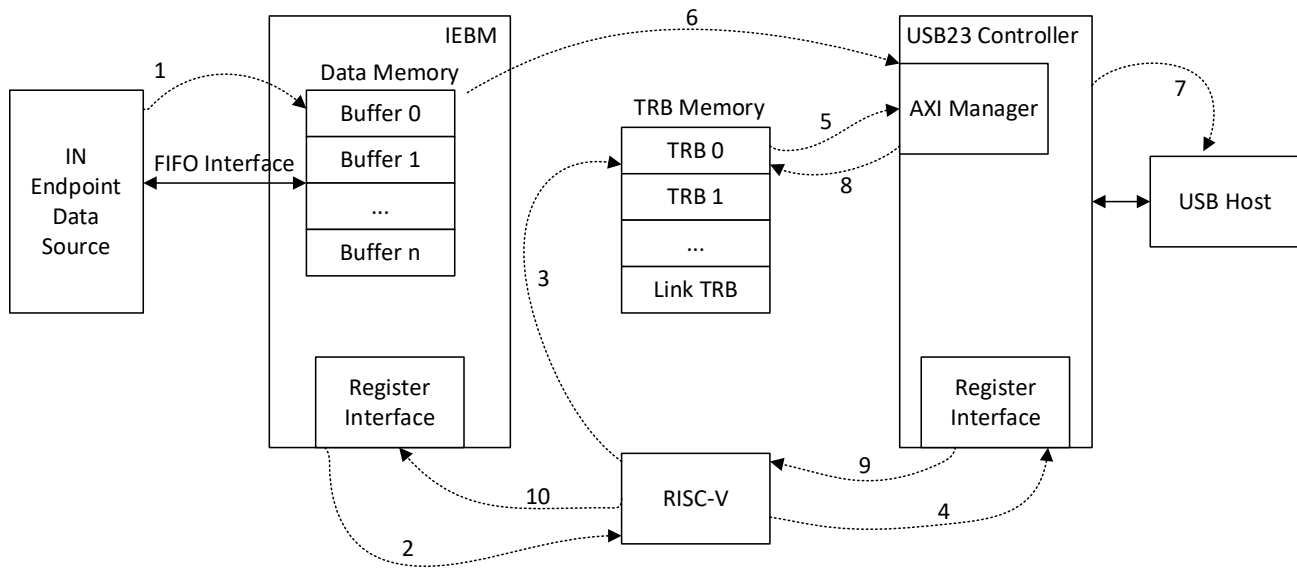


Figure 8.3. IN Data Flow from USB Device to Host

The following lists the IN data flow from USB device to host:

1. The *in_cntlr* (IN Endpoint Data Source) module generates dummy data and writes it into the IEBM buffer.
2. When a single buffer in the IEBM is filled, the IEBM module generates an interrupt signal to notify the RISC-V processor that the buffer is ready for further processing or data transfer operations.
3. Based on the buffer size, the RISC-V firmware prepares the Transfer Request Block (TRB).
The TRB is stored in True Dual Port Memory, which is part of the USB23 AXI Bridge. This memory is shared between the RISC-V CPU and the USB23 controller. For more information, refer to [Figure 8.1](#).
4. After the RISC-V firmware has filled the TRB, it notifies the USB23 controller by writing to its registers.
5. The USB23 controller fetches and processes the TRB.
6. The USB23 controller reads the data from the IEBM payload buffer according to the TRB and sends it serially to the USB host.
7. The USB23 controller transmits the data to the USB host. The application on the USB host system receives the data.
8. After transferring the number of bytes specified in the TRB, the USB23 controller resets the corresponding flag field in the TRB and notifies the RISC-V CPU via interrupt.
9. Upon receiving the interrupt, the RISC-V CPU reads the relevant USB23 controller registers.
10. The RISC-V CPU updates the IEBM register to free up the corresponding buffer for new data.

8.2.2. Description on Data Flow from USB Host to Device

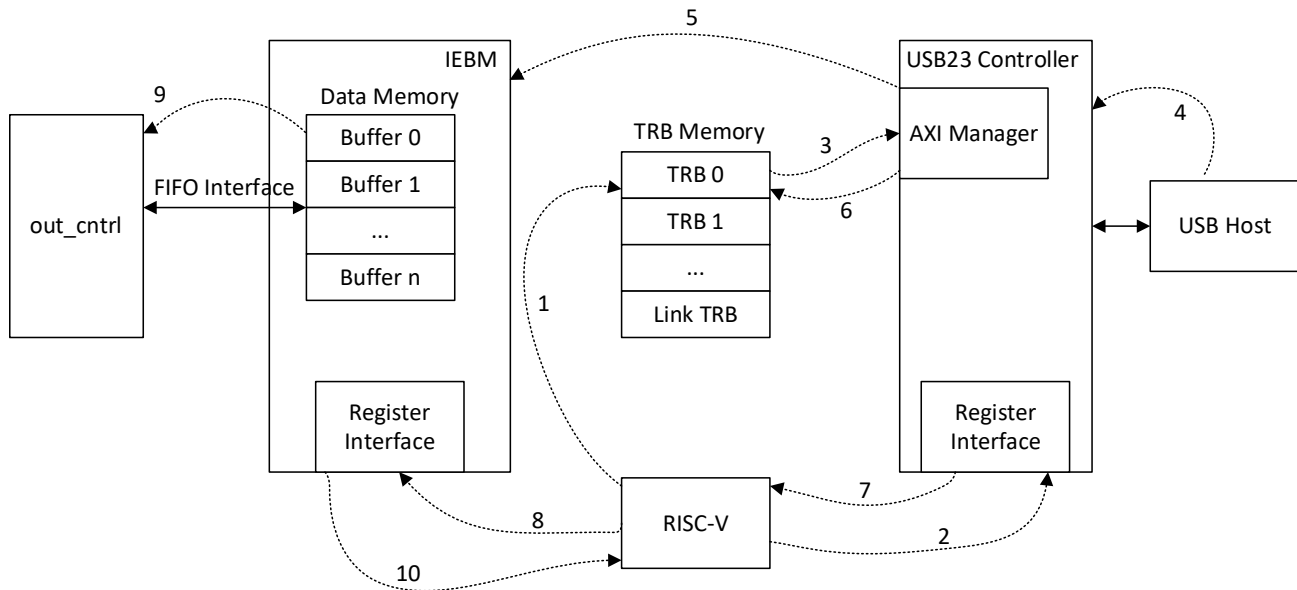


Figure 8.4. OUT Data Flow from USB Host to Device

The following lists the OUT data flow from USB host to device:

1. The RISC-V CPU first fills the TRB for the OUT endpoint.
2. After the TRB is filled, the RISC-V CPU notifies the USB23 controller through the register interface.
3. After the appropriate register bits are set by the RISC-V CPU, the USB23 controller fetches and processes the TRB.
4. When the application on the USB Host system sends data to the USB23 controller, the controller receives the data.
5. The AXI manager of the USB23 controller transfers the received data to the OEBM buffer.
6. Upon successfully transferring the specified number of bytes (as defined in the TRB), the USB23 controller generates an event interrupt for the RISC-V CPU.
7. When the RISC-V CPU receives the event interrupt from the USB23 controller, it reads the controller register to decode the event.
8. After receiving the transfer complete event, the RISC-V CPU updates the OEBM register to notify OEBM that a data buffer containing valid data is ready for output.
9. The OEBM sets its FIFO interface signal, prompting the OUT controller (out_cntrl) to begin reading data from the OEBM buffer.
10. When the OEBM buffer is empty, it generates an interrupt for the RISC-V CPU. Upon receiving this interrupt, the RISC-V CPU performs the necessary actions to make the buffer available for the next TRB.

8.2.3. USB 2.0/3.2 Primitive

This IP contains USB 2.0/3.2 primitive. For more information, refer to the [Operation Details](#) section.

8.2.4. USB23_ED

This block handles the bulk, control, and interrupt transfers requested by the USB 2.0/3.2 primitive. The following sub-blocks are present in this block.

8.2.4.1. PLL

This phase-locked loop (PLL) IP generates 60 MHz, 72 MHz, and 144 MHz clock signals from an onboard oscillator, which are routed to other submodules for further processing.

Module/IP Block Wizard

Configure Component from Module pll Version 1.9.0
Set the following parameters to configure this component.

Diagram pll_60m_i_60m_o

pll_60m_i_60m_o

clkop_o

clkos2_o

clkos_o

lock_o

clk_i

pll

Document

No DRC issues are found.

Calculate

Configure IP

General

Optional Ports

Property	Value
General	
Configuration Mode	Frequency
Set Parameter Optimization Target	Minimum Jitter (Higher VCO)
Enable Fractional-N Divider	☒
Enable Spread Spectrum Clock Generation	☐
Enable Internal Path Switching	☐
VCO Frequency [800 - 1600]	1440
Reference Clock	
CLKI: Frequency (MHz) [18 - 800]	60
CLKI: Divider Actual Value [1 - 44]	1
Phase Detector Frequency (MHz) [18 - 100]	60
Enable Reference Clock Monitor	☐
Feedback	
Primary Clock Output	
CLKOP: Frequency Desired Value (MHz) [6.25 - 800]	60
CLKOP: Divider Actual Value [1 - 128]	24
CLKOP Tolerance (%)	0.0
CLKOP: ERROR (PPM)	0
CLKOP: Enable Trim for CLKOP	☐
Secondary Clock Output	
CLKOS: Enable	☒
CLKOS: Frequency Desired Value (MHz) [6.25 - 800]	72
CLKOS: Divider Actual Value [1 - 128]	20
CLKOS Tolerance (%)	0.0
CLKOS: ERROR (PPM)	0
CLKOS: Static Phase Shift (Degrees)	0
CLKOS: Enable Trim for CLKOS	☐
Secondary Clock Output (2)	
CLKOS2: Enable	☒
CLKOS2: Frequency Desired Value (MHz) [6.25 - 800]	144
CLKOS2: Divider Actual Value [1 - 128]	10
CLKOS2 Tolerance (%)	0.0
CLKOS2: ERROR (PPM)	0

Figure 8.5. PLL Configuration

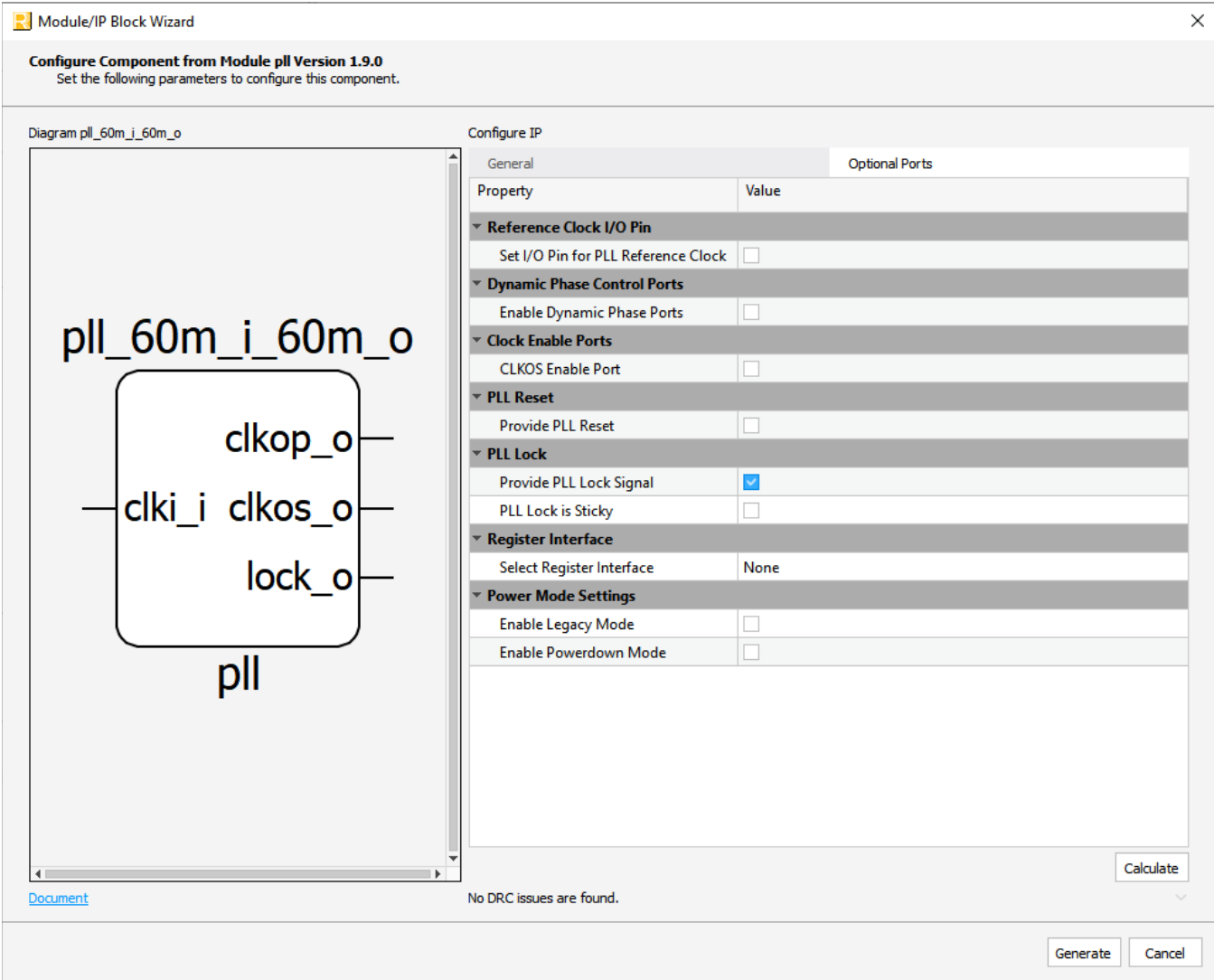


Figure 8.6. PLL Optional Port Configuration

Note: If you are using the LIFCL-33U-EVN-REVA board, then incoming clock to the FPGA is different compared to the LIFCL-33U-CMP-EVN_REVB board. In this case, the PLL configuration must be changed.

8.2.4.2. USB23 Controller

This is the USB23 Device Controller hardened IP core. This IP communicates with the system memory via the AXI4-to-AHBL converter module. The AXI DMA manager of the USB23 Controller is connected with two different memory via the AXI interconnect bridge:

- For non-control IN transfer, the USB controller fetches data from IEBM modules data RAM. For non-control OUT transfer, the USB controller writes data into OEBM modules data RAM. These RAMs are not shared with any other host and/or RISC-V processor.
- For all other operations, the USB controller communicates with the system memory, which is shared with the RISC-V processor. This type of architecture helps to achieve higher throughput.

8.2.4.3. AHBL-to-LMMI Converter

The USB23 controller register interface can be configured through its LMMI interface. Because the RISC-V MC supports only AHB-Lite interface, the AHBL-to-LMMI converter is needed.

8.2.4.4. RISC-V MC Processor

All the serial peripheral blocks are connected to the RISC-V soft processor via the AHBL interface. The RISC-V MC CPU IP processes data and instructions while monitoring external interrupts coming from the USB23 controller, and IN endpoint buffer manager (IEBM) core. The RISC-V processor configuration is shown in the following figure.

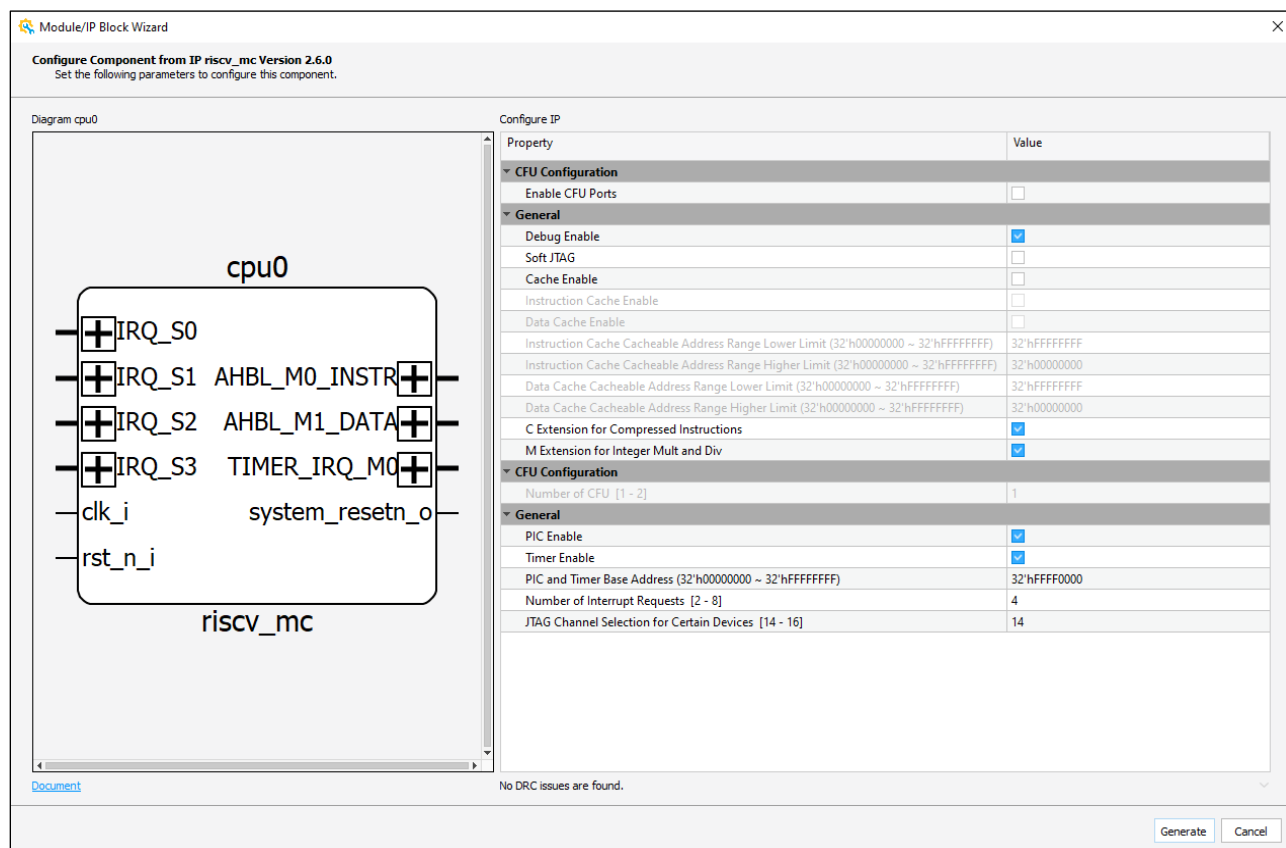


Figure 8.7. RISC-V Configuration

8.2.4.5. UART and GPIO

The RISC-V processor communicates with UART and GPIO core via its APB interface. Hence, the UART and GPIO core are connected to RISC-V soft processor via AHBL-to-APB bridge. Generally, UART IP core performs serial-to-parallel conversion on data characters received from a peripheral UART device and parallel-to-serial conversion on data characters received from the host (RISC-V processor) located inside the FPGA through an APB Interface. The UART IP core configuration is shown in [Figure 8.8](#). Additionally, use a 32-bit input GPIO (hardware version of the GPIO) to monitor the FPGA design version for a particular release. The hardware version of the GPIO configuration is shown in [Figure 8.9](#)[Error! Reference source not found.](#)

Module/IP Block Wizard

Configure Component from IP **uart** Version 1.3.0
Set the following parameters to configure this component.

Diagram **uart0**

uart0

APB_S0

clk_i

rst_n_i

rx_d_i

INT_M0

tx_d_o

uart

Configure IP

Property	Value
General	
Enable APB	☒
System Clock Frequency (MHz) [2 - 200]	72
Serial Data Width	8
Stop Bits	1
Parity Enable	☐
ODD Parity	☐
Enable Stick Parity	☐
Baud Rate	
Baud Rate Type	Standard
UART Standard Baud Rate	115200
UART Custom Baud Rate [2400 - 1000000]	115200
UART Feature Enables	
FIFO Enable	☐
MODEM Enable	☐
Rx Ready Enable	☐
Tx Ready Enable	☐

[Document](#)

No DRC issues are found.

Generate

Cancel

Figure 8.8. UART Configuration

Hardware Version GPIO

Additionally, we're using a 32-bit input GPIO to monitor the FPGA design version for a particular release. Hardware version GPIO configuration is shown in [Figure 8.9](#).

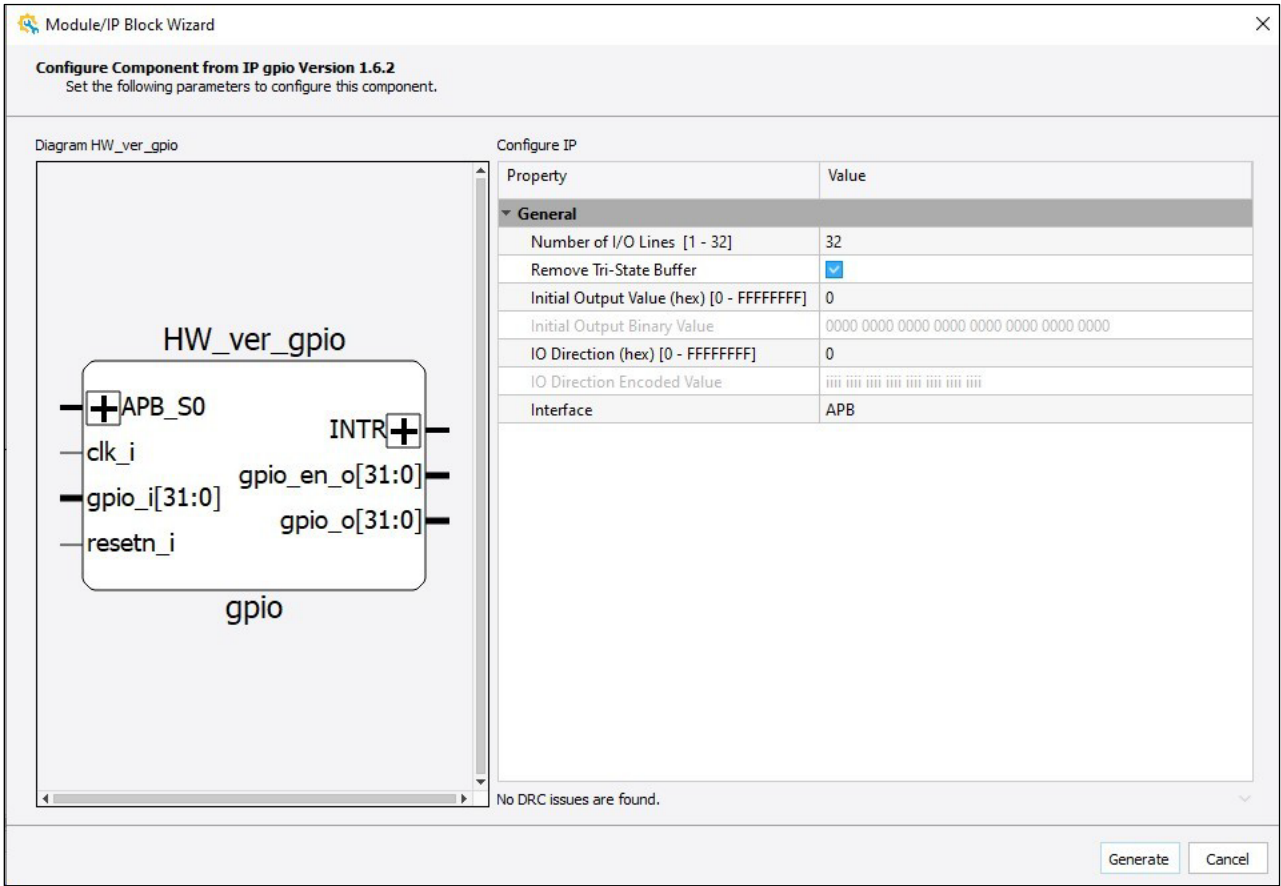


Figure 8.9. Hardware Version GPIO Configuration

Remove Wakeup GPIO

A 1-bit input GPIO (Remote wakeup GPIO) is used to wake up the USB device using the remote wakeup. While Performing the Remote Wakeup - Enabled Test (TD 9.12) from the Chapter 9 compliance test, You must press the Pushbutton2 (SW2) on the CrossLinkU-NX REVB Evaluation Board. Remote wakeup GPIO configuration is shown in [Figure 8.10](#).

Module/IP Block Wizard

Configure Component from IP gpio Version 1.6.2
Set the following parameters to configure this component.

Diagram rw_gpio

Configure IP

Property	Value
General	
Number of I/O Lines [1 - 32]	1
Remove Tri-State Buffer	☒
Initial Output Value (hex) [0 - FFFFFFFF]	0
Initial Output Binary Value	0
IO Direction (hex) [0 - FFFFFFFF]	0
IO Direction Encoded Value	i
Interface	APB

[Document](#)

No DRC issues are found.

[Generate](#) [Cancel](#)

Figure 8.10. Remote Wakeup GPIO Configuration

USB2 PHY Reset GPIO

A 1-bit output GPIO is used to facilitate firmware reset USB2.0 PHY. This facilitates RISC V firmware to reset USB2.0 PHY based on requirements. The configuration is shown in [Figure 8.11](#).

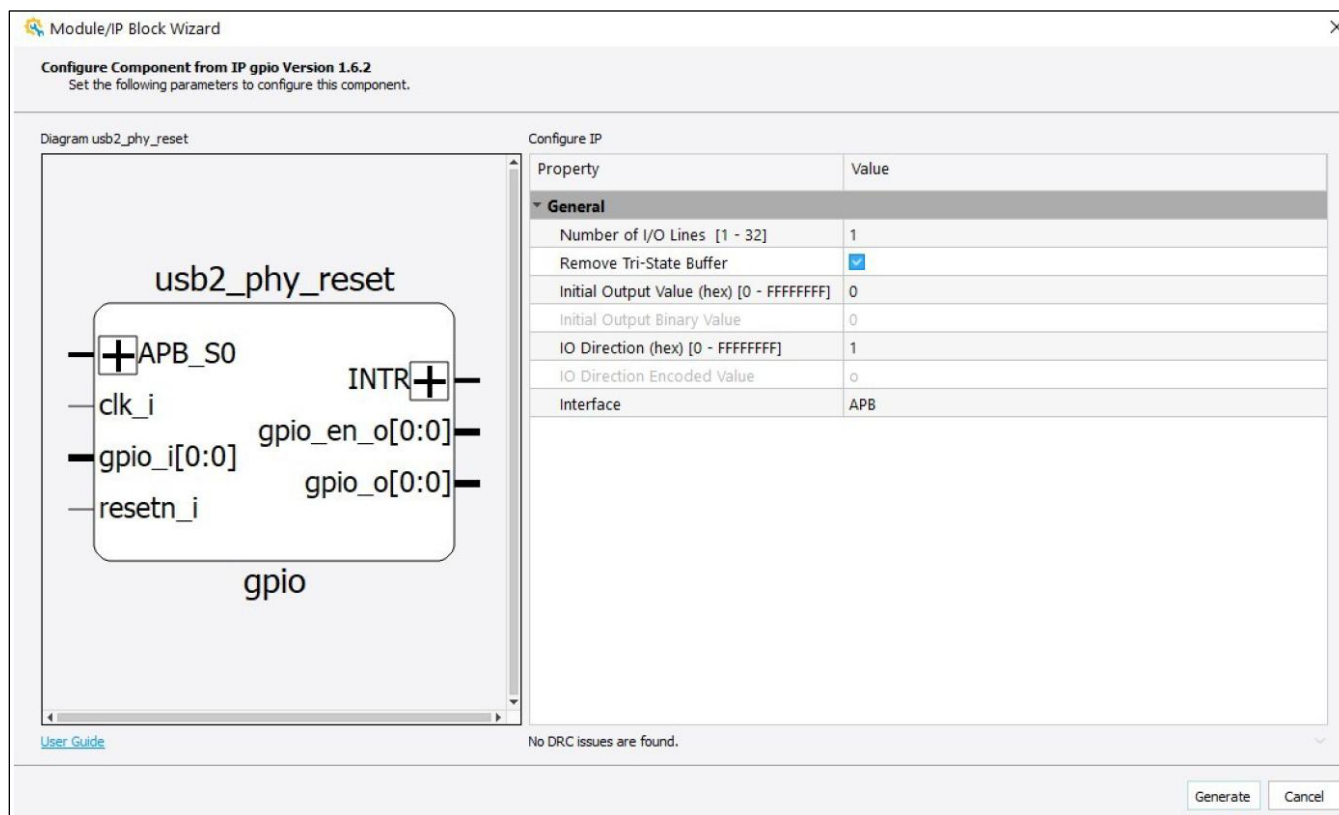


Figure 8.11. USB2 PHY Reset GPIO Configuration

8.2.4.6. System Memory

This system memory IP core is a soft IP that is used by the RISC-V processor. The RISC-V processor reads the instructions stored in this memory via the AHBL interface and carry out the instructions accordingly.

This system memory IP core has the following two interfaces that the manager communicates with:

- AHBL
- AXI4

The system memory IP core configuration is shown in the following figure.

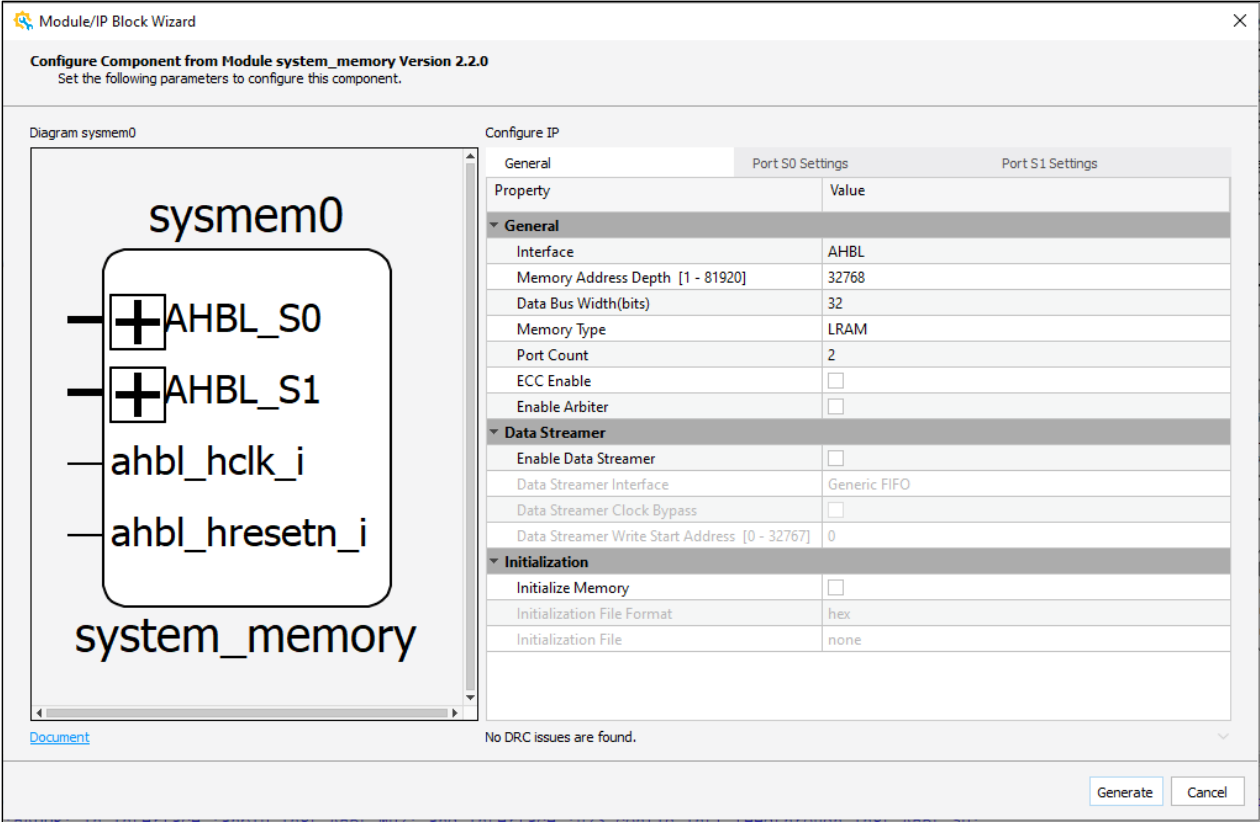


Figure 8.12. System Memory Configuration

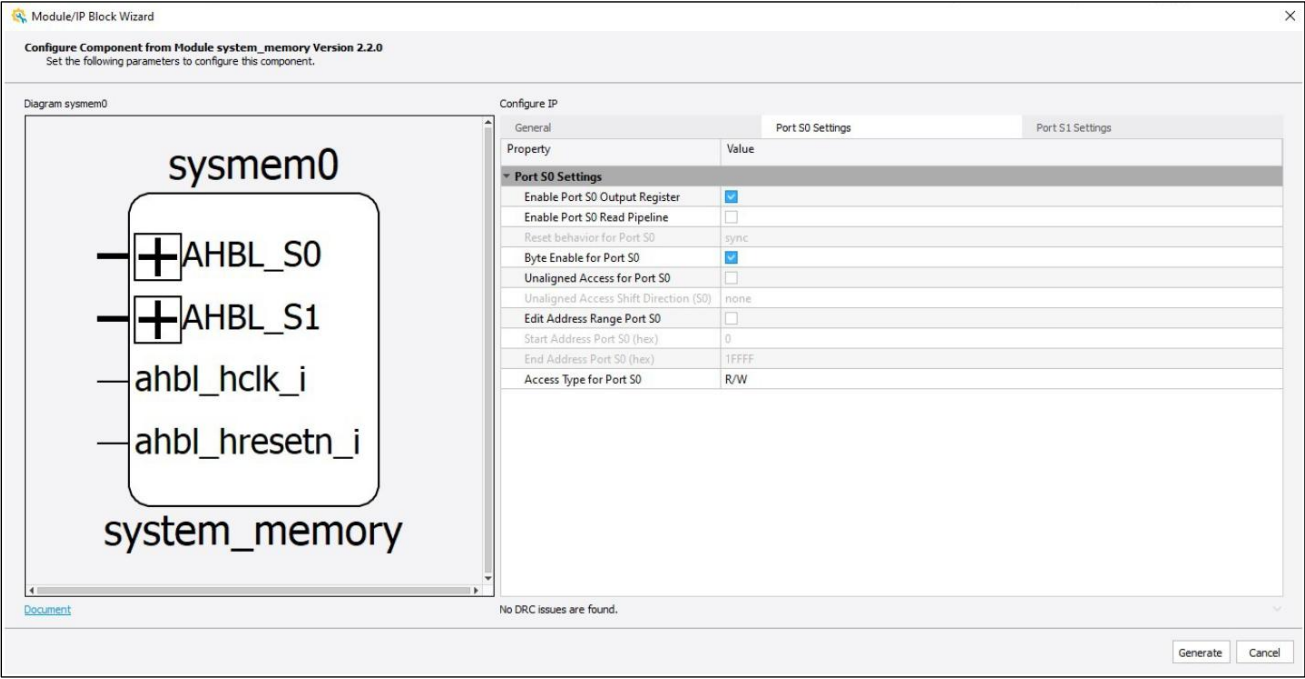


Figure 8.13. System Memory Port S0 Configuration

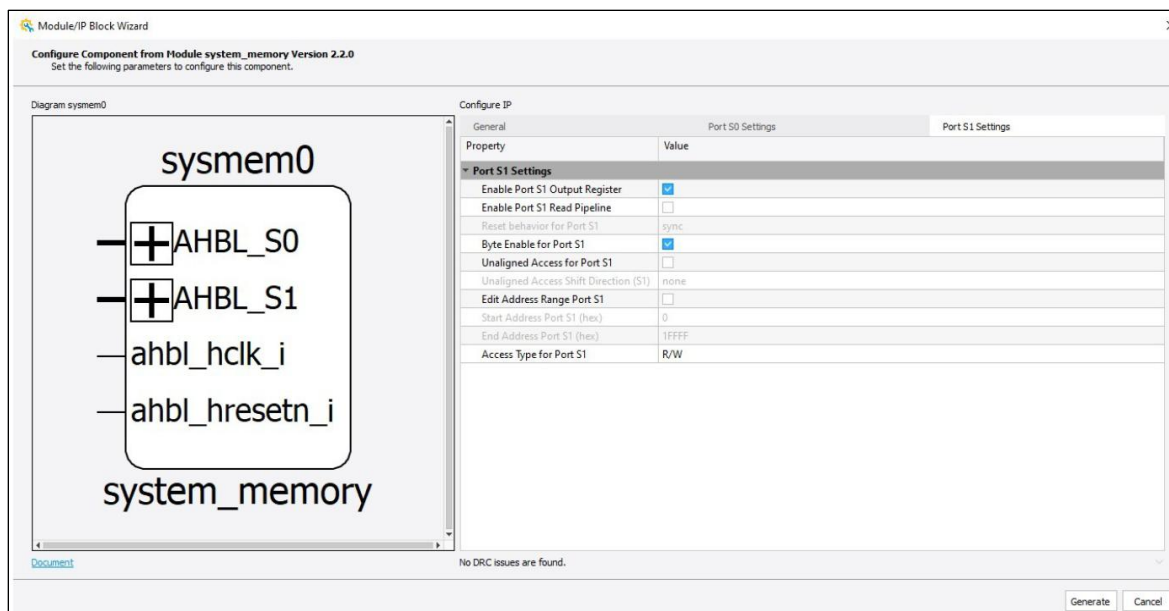


Figure 8.14. System Memory Port S1 Configuration

8.2.4.7. AHBL Interconnect

This IP provides the communication interface between RISC-V soft processor and peripherals (system memory, AHBL to LMMI bridge etc.) which supports AHBL interface. AHBL interconnect configuration is shown in the following figures.

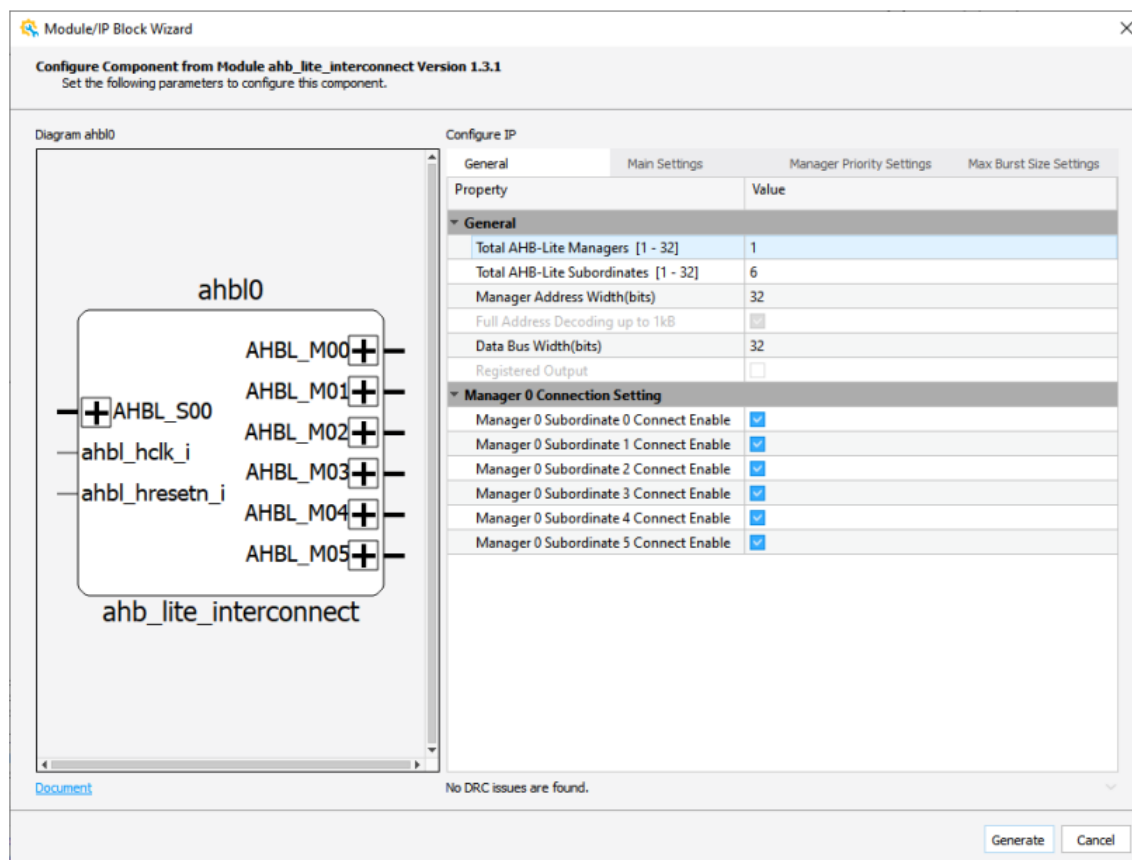


Figure 8.15. AHBL Interconnect General Configuration

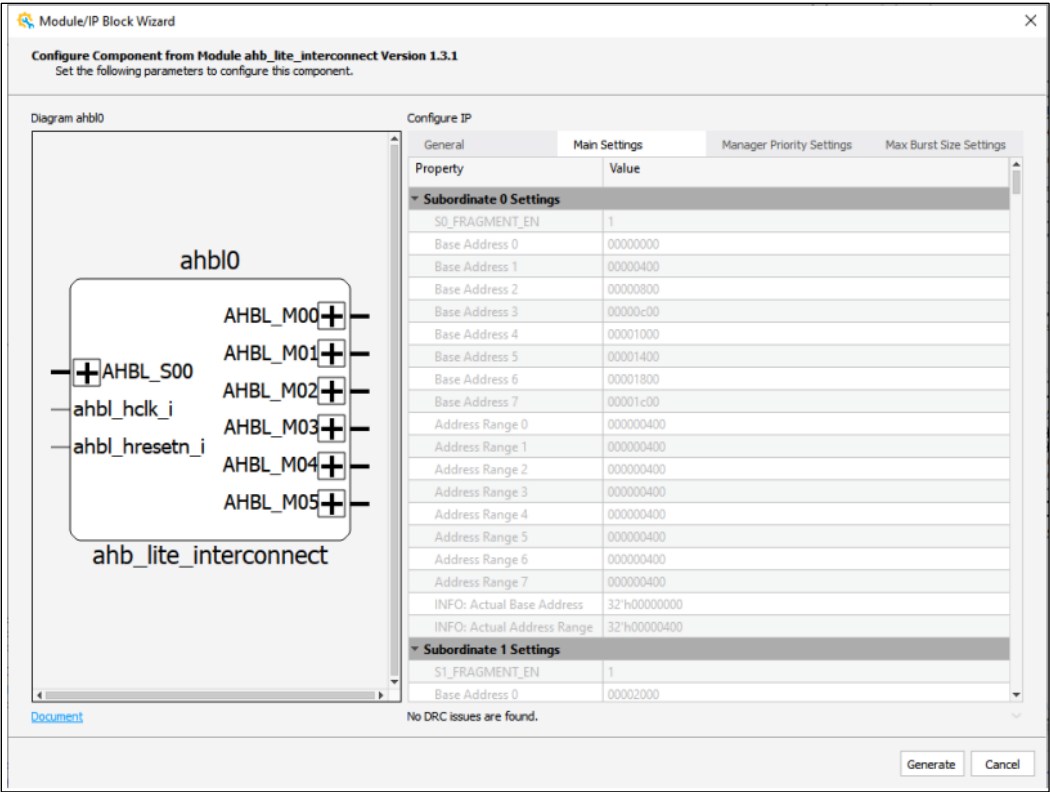


Figure 8.16. AHBL Interconnect Main Configuration

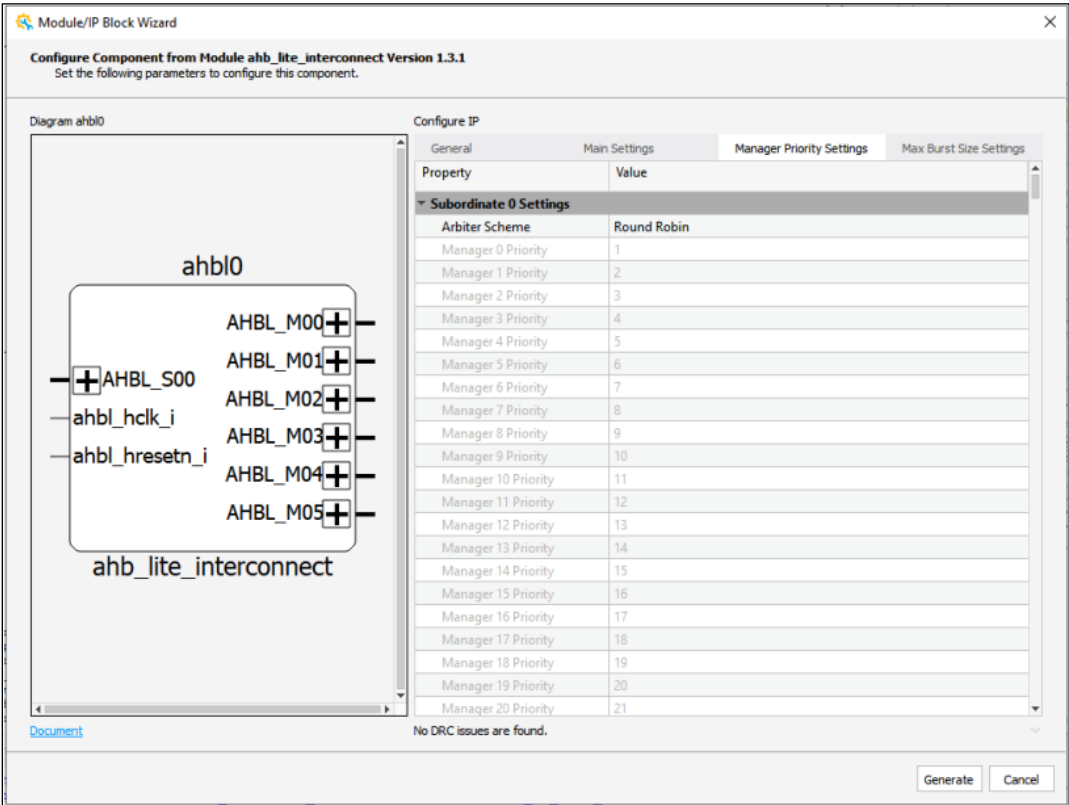


Figure 8.17. AHBL Interconnect Manager Priority Settings

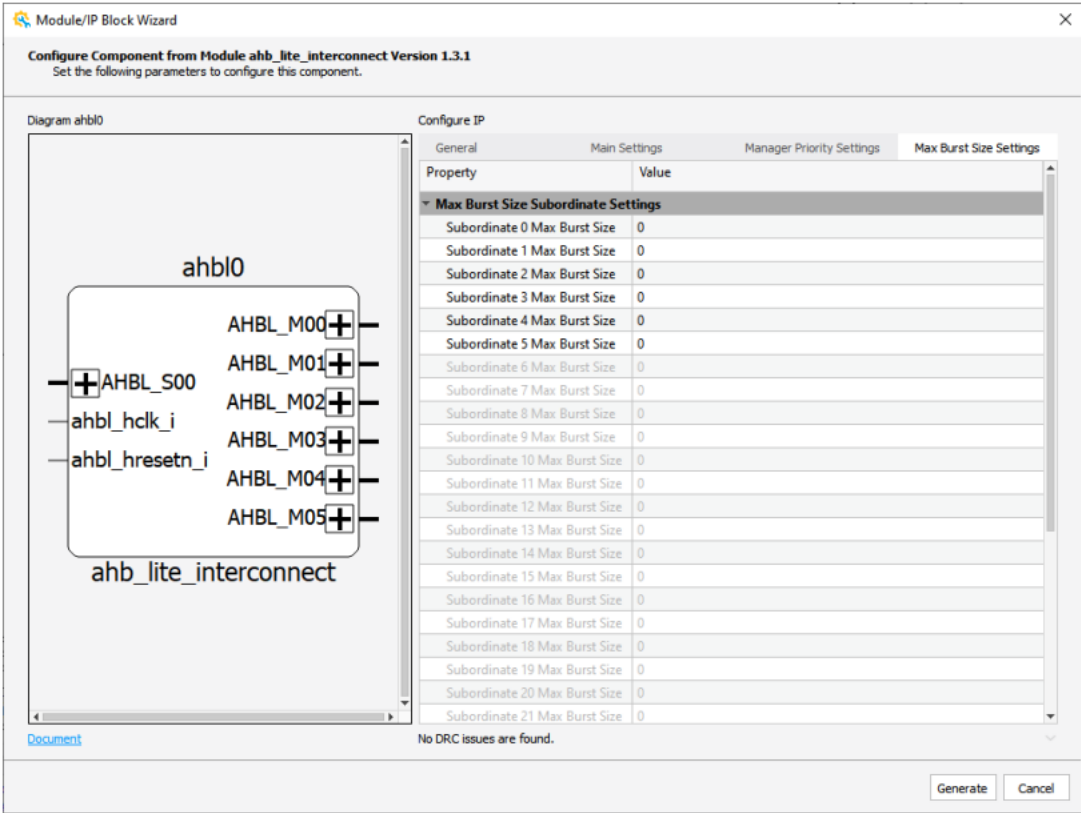


Figure 8.18. AHBL Interconnect Max Burst Settings

8.2.4.8. APB Interconnect

The APB interconnect provides the communication interface between the RISC-V soft processor and other slow peripherals (UART, GPIO, and so on) that supports the APB interface. The interconnect configuration is shown in the following figures.

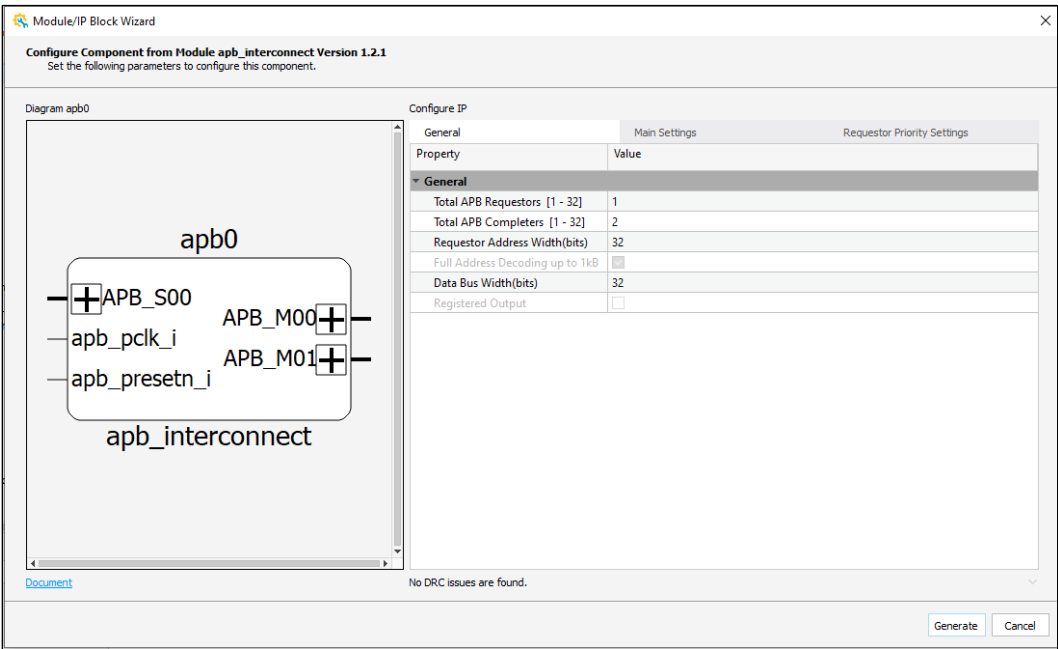


Figure 8.19. APB Interconnect Configuration 1

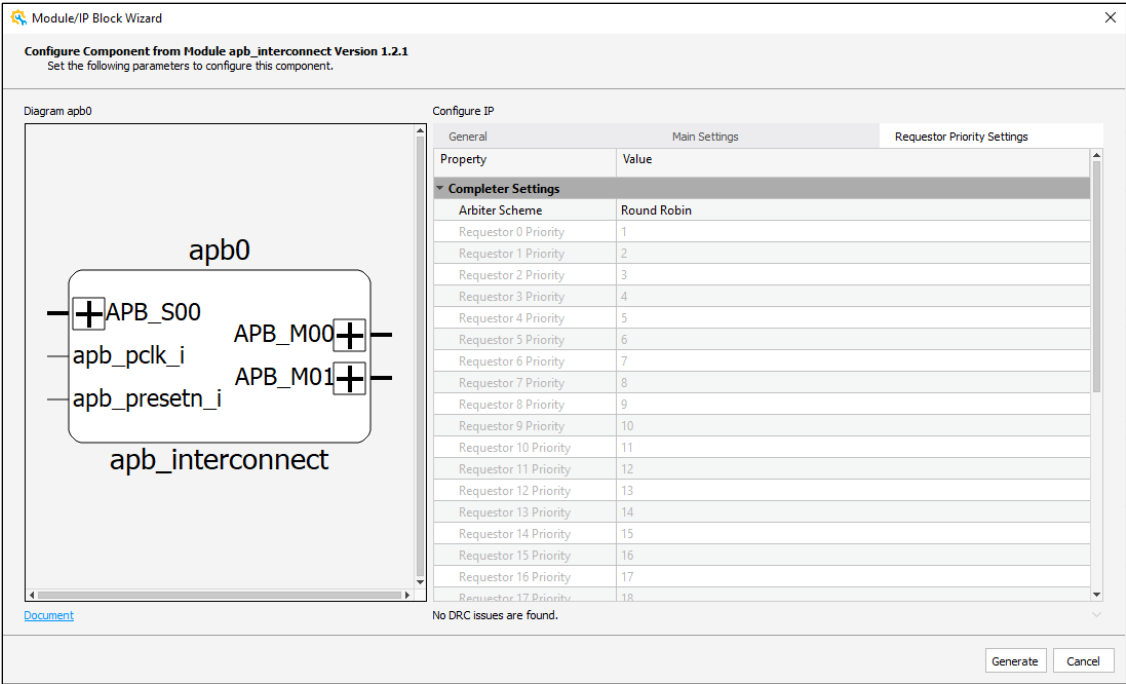


Figure 8.20. APB Interconnect Configuration 2

8.2.4.9. AHBL-to-APB Converter

The AHB-Lite-to-APB bridge provides an interface between the AHB-Lite manager and APB completer. Read and write transfers on the AHB are converted into equivalent transfers on the APB. The AHBL-to-APB Converter IP core configuration is shown in the following figure.

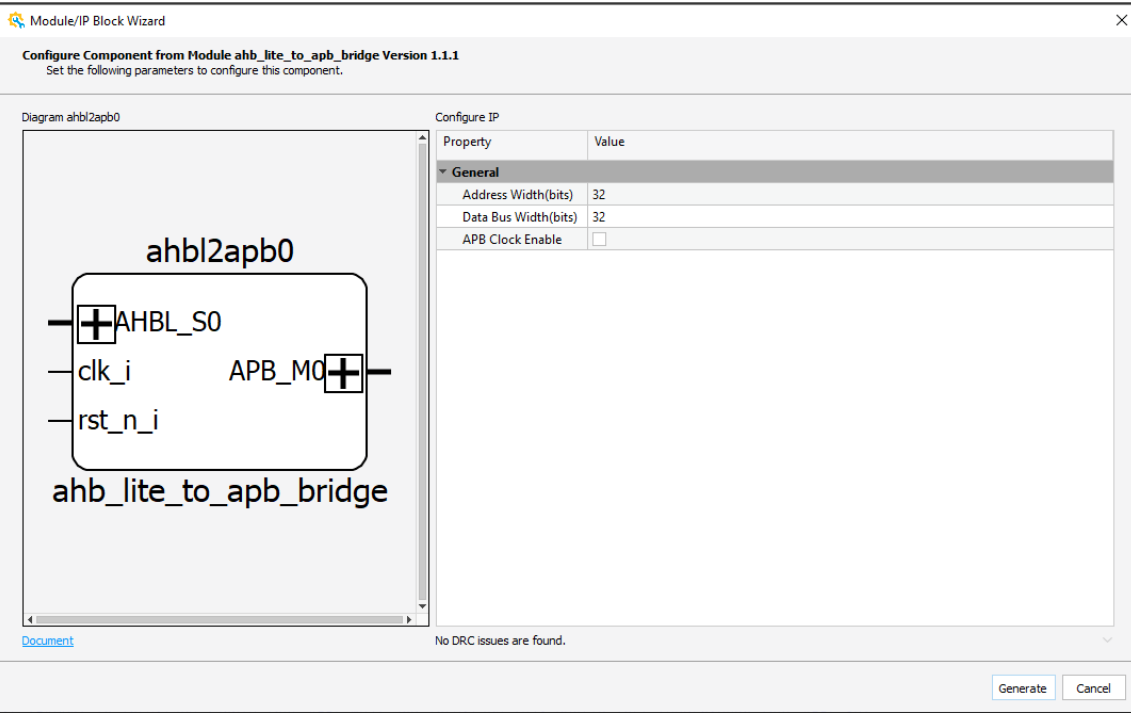


Figure 8.21. AHB-Lite to APB Configuration

8.2.4.10. AHBL to Memory Bridge

This block helps RISC-V data manager to access True Dual Port memory that is dedicated for USB operations. This block converts AHB-Lite subordinate signals into native memory manager interface signals.

8.2.4.11. IN FIFO Controller

When the IEBM block grants permission to write the data, the IN FIFO controller module initiates the process. It reads the data from the FIFO memory, where the incremental data is temporarily stored and it writes through the FIFO interface to the IN endpoint buffers.

8.2.4.12. IN Endpoint Buffer Manager (IEBM)

The IEBM is responsible for managing the IN endpoint buffers for USB23 Controller. The data is written into different buffers sequentially through the First-In-First-Out (FIFO) interface. Once the data is available in these buffers, the USB23 controller reads it through the AXI interface. This entire process, from managing the FIFO writes to facilitating the AXI reads is managed by the IEBM module.

For more information, refer to the document located at the following IP package installation path:

`<installed_IP_package>\nx33u_cmp_revb_evn_bd_rd_des\documents\nx33u_iebm_architecture.pdf`

Endpoint IN Buffer Configuration

The default buffer size for the IN endpoint is set to 8192 bytes. This can be modified in the following ways:

1. Modify the static buffer size in `rtl/include/user_cfg_params.v`:
`localparam IEBM_SINGLE_BUFR_SIZE = 8192;`
Note: Changing this value requires Radiant software recompilation.
2. Override the buffer size in firmware:
 - a. In `lsc_in_ep_buf_mgmt.c`, update `lsc_iebm_init()`:
 Replace:
`LSC_IEBM_FW_SINGLE_BUF_DEPTH`
 with:
`in_user_ep_cnfg[usb_ep_num].ebm_fw_single_buf_depth`
 - b. In `lsc_user_cnfg.c`, update the `bulk_in_user_ep_cnfg` structure:
 Set `.ebm_fw_single_buf_depth = 4096` (or desired size).

Partial Buffer Handling

To handle variable-length IN transfers, use the `ieds_hand_over_partial_filled_fifo_pl_i` signal. This signal indicates that the data is less than the configured buffer size. In the example design, this is controlled by the `in_cntrl` module using the `NO_OF_BYTES_TO_XFER` parameter. Adjusting this value requires Radiant software re-compilation.

Interrupt Generation

The USB23 IP firmware generates interrupts on every transaction. This is configured in `lsc_endpoint.c` within the `lsc_usb_set_ep_cfg()` function. The `PARAM1` register is set with the following bits to enable endpoint-specific events:

- Bit 10: XferNotReady Enable (XferNRdyEn)
- Bit 9: XferInProgress Enable (XferInProgEn)
- Bit 8: XferComplete Enable (XferCmplEn)

8.2.4.13. OUT Endpoint Buffer Manager (OEBM)

The OEBM is responsible for managing the OUT endpoint buffers for the USB23 Controller. The USB23 controller writes the data sequentially into the different buffers through the AXI interface. When the data is available in these buffers, the OUT FIFO controller reads the data through the FIFO interface. This entire process, from managing the AXI writes to facilitating the FIFO reads is managed by the OEBM module.

For more information, refer to the document located at the following IP package installation path:

`<installed_IP_package>\nx33u_cmp_revb_evn_bd_rd_des\documents\nx33u_oebm_architecture.pdf`

8.2.4.14. OUT FIFO Controller

When the OEBM block grants permission to read the data, the OUT FIFO controller module initiates the process. It reads the data from the OUT endpoint buffers and writes through the FIFO interface to the IN endpoint buffers.

8.2.4.15. USB23 to AXI Bridge

This block helps the USB23 controller to access various memories. The USB23 controller has an AXI manager interface. This block has an AXI subordinate interface. This block converts AXI subordinate signals into native memory manager interface signals. This block also acts as a decoder to select appropriate memory based on input address.

8.3. Running the Example Design

The USB 2.0/3.2 IP is hardware validated. The USB 2.0/3.2 IP comes with a hardware validation design that can be used to validate the IP on board. The Lattice NX33U FPGA contains hardened USB controller (USB23 controller) that supports USB3.2 Gen1 mode and USB2.0. This document describes the procedure to test data streaming through the USB controller using the WinUSB driver-based CLI application.

The following lists the hardware requirements:

- Lattice CrosslinkU-NX33 Evaluation Board (Rev B). This board has a USB Type C connector as shown in the following figure.

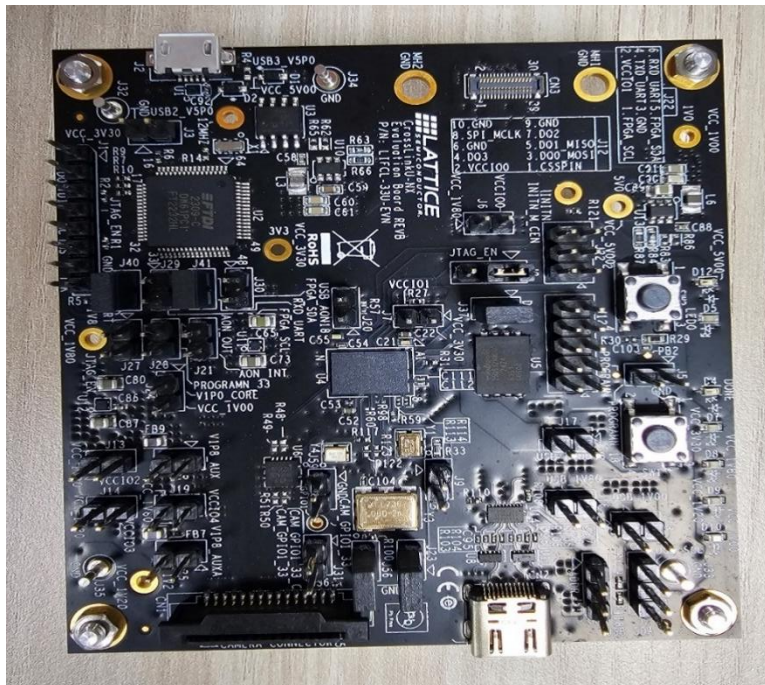


Figure 8.22. Lattice CrosslinkU-NX33 Evaluation Board (Rev B)

- USB2.0 Type A to Micro Type B cable for FPGA programming.
- Type C to Type C cable for USB streaming test.

The following lists the software requirements:

- Radiant Programmer v2024.2 or latest
- WinUSB Asynchronous CLI utility

8.3.1. Generating the USB23 Benchmark Example Design

This section describes the steps to generate the USB23 Benchmark example design. There are two main compilation flows:

- Propel software compilation for HAL driver
- Radiant software compilation for bitstream

8.3.2. HAL Driver Compilation

The following lists two methods to run the project:

- Creating the Project
- Import Project from Directory

8.3.2.1. Creating the Project

To create a new project in the Lattice Propel 202.1 (or latest) Software Build Tools for Eclipse, follow the steps:

1. Make sure you unzip the `nx33u_cmp_revb_evn_bd_rd_des.zip` file.

`application dut_inst.v dut_params.v nx33u_cmp_revb_evn_bd_rd_des.zip software`

Figure 8.23. nx33u_cmp_revb_evn_bd_rd_des.zip File

2. Open the Lattice Propel 2024.1 (or latest) Software Build Tools for Eclipse. Select a workspace folder to create the software project, and click **Launch**.
3. Select **File > New > Lattice C/C++ Project**, as shown in the following figure.

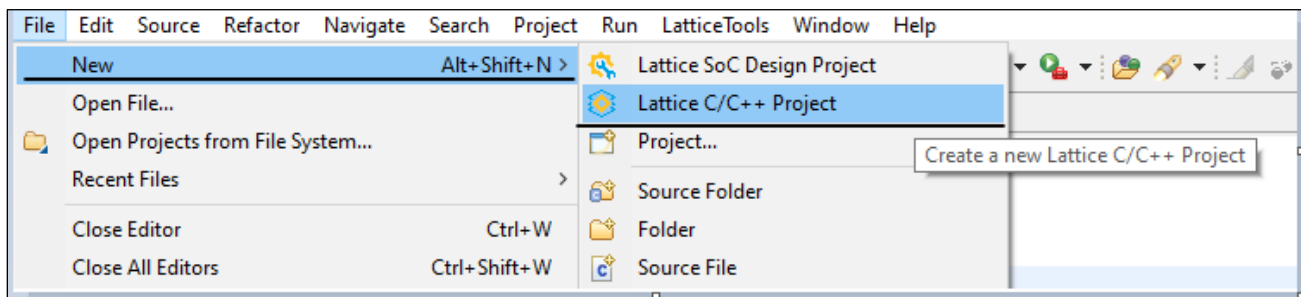


Figure 8.24. Create a Propel Project

4. Browse the system environment file and BSP package, and select the `sys_env.xml` file.

The `sys_env.xml` file can be found in the following folder path:

`<Generated_USB_IP>/eval/nx33u_cmp_revb_evn_bd_rd_des/u23_lifclu_nx33u_prpl_bldr/sge/sys_env.xml`

5. Click **Next**.

Note:

- The `sys_env.xml` file must be within the `sge` folder generated by the Lattice Propel Builder v2024.2 or latest. Otherwise, no BSP package will be generated in the project.
- If the example design is modified in the Propel Builder, you are required to re-initialize the `sys_env.xml` through the Propel Builder v2024.2 or the latest.
- Regenerate all the sub-blocks as shown in the following figure by loading this `.sbx` file :
`<Generated_USB_IP>/eval/nx33u_revBbd_u23toaxi_rd_des/u23_lifcl_nx33u_prpl_bldr/u23_lifcl_nx33u_prpl_bldr/u23_lifcl_nx33u_prpl_bldr/u23_lifcl_nx33u_prpl_bldr.u23_lifcl_nx33u_prpl_bldr.sbx`

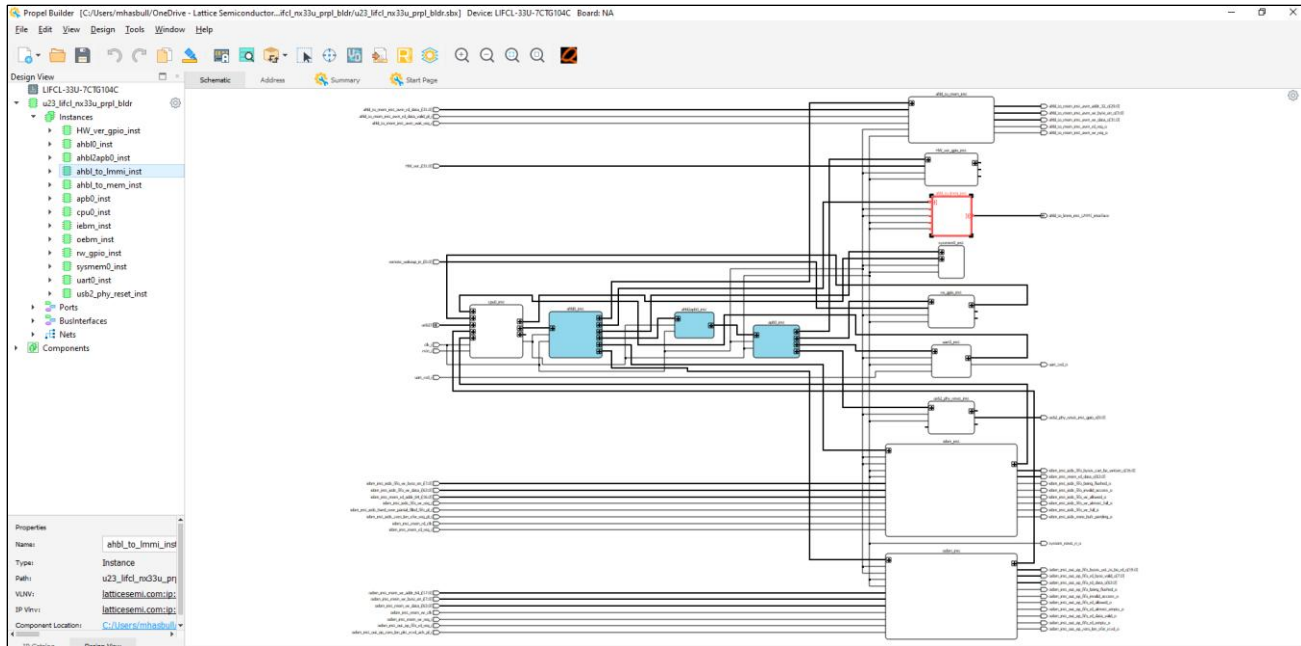


Figure 8.25. Regenerate the Sub-Modules in the Propel Software

- Click on the **Generate** button.

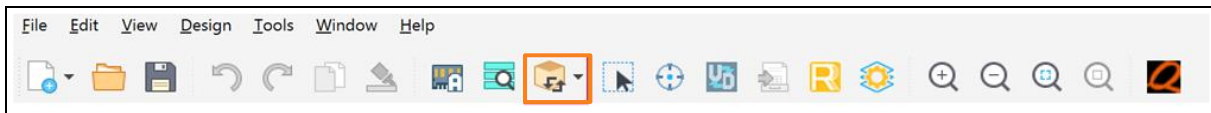


Figure 8.26. Generate Button in the Propel Software

- A new `sys_env.xml` will be generated.
- Browse to the `sys_env.xml` file, provide a name for the project, and click **Next**, as shown in the following figure.

C/C++ Project

Load System and BSP
Load lattice system environment file and BSP package

Select system environment file and BSP package

System env:

Select processor core to create C/C++ Project

Core selected:

Project type:

System information

Device Family	CPU Name	Instance Name
LIFCL	riscv_mc	cpu0_inst

Select Example Application

Hello World Project

Example-HelloWorld-blink-uart

1.Led blink
The corresponding SoC project should include gpio instance.

2.Uart print
There are two ways to support uart print:

Project name:

☒ Use default location

Location:

Choose file system:

☐ Build the project

☒ Create a debug launch configuration for

Figure 8.27. Create a Project

- In the **Lattice Toolchain Setting** window, follow the settings shown in the following figure, and click **Finish**.

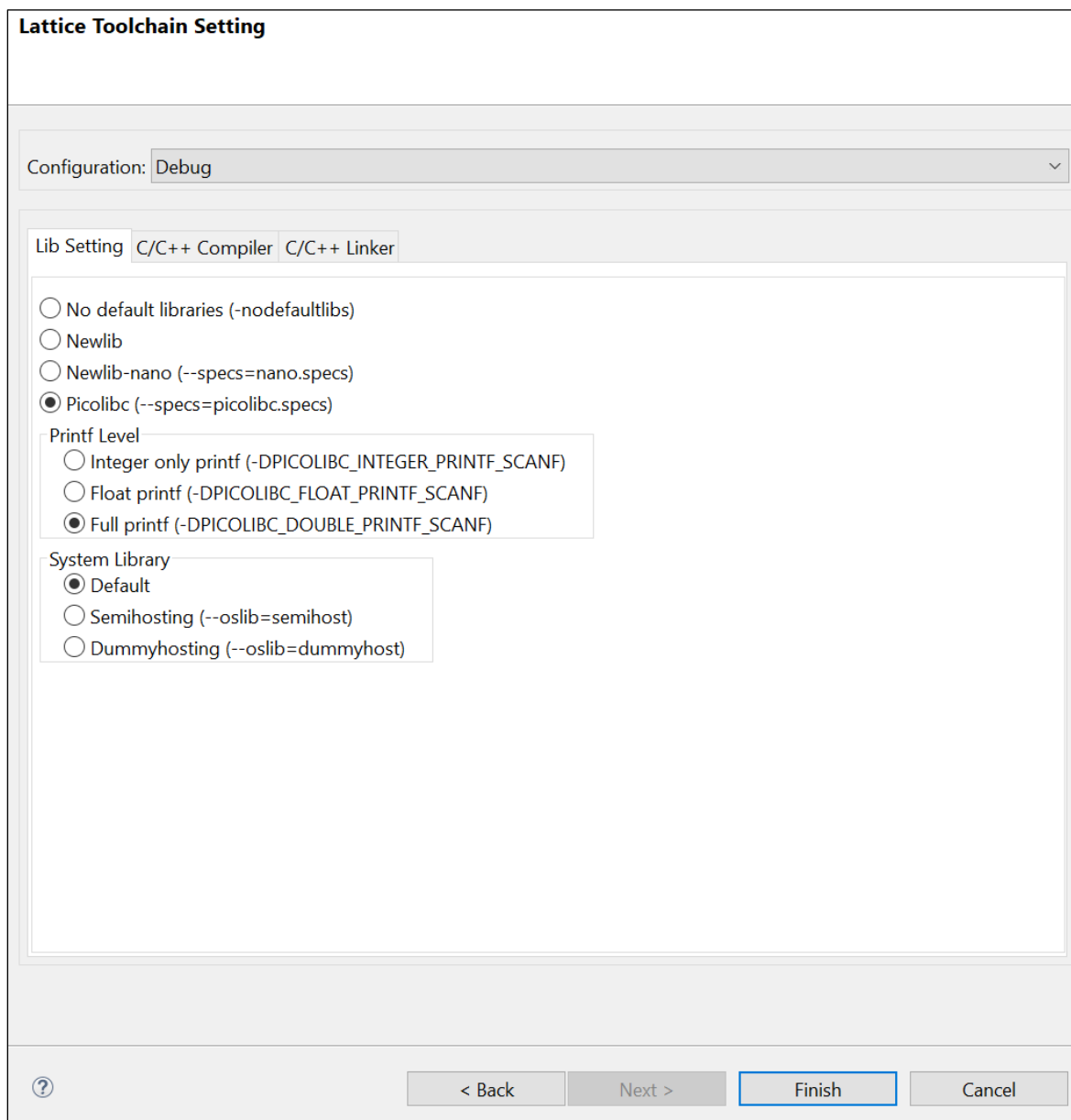


Figure 8.28. Toolchain Setting

8.3.2.2. Building the Project

To build a project, follow these steps:

1. Obtain the firmware files from the following path and make sure to unzip the file first—
<Generated_USB_IP>/eval/software/usb23hal_6eps_v2_2.zip.
2. Copy all .c and .h files including the **class folder** from the usb23hal_6eps_v2_2 folder and paste them into the src folder of the SDK project.

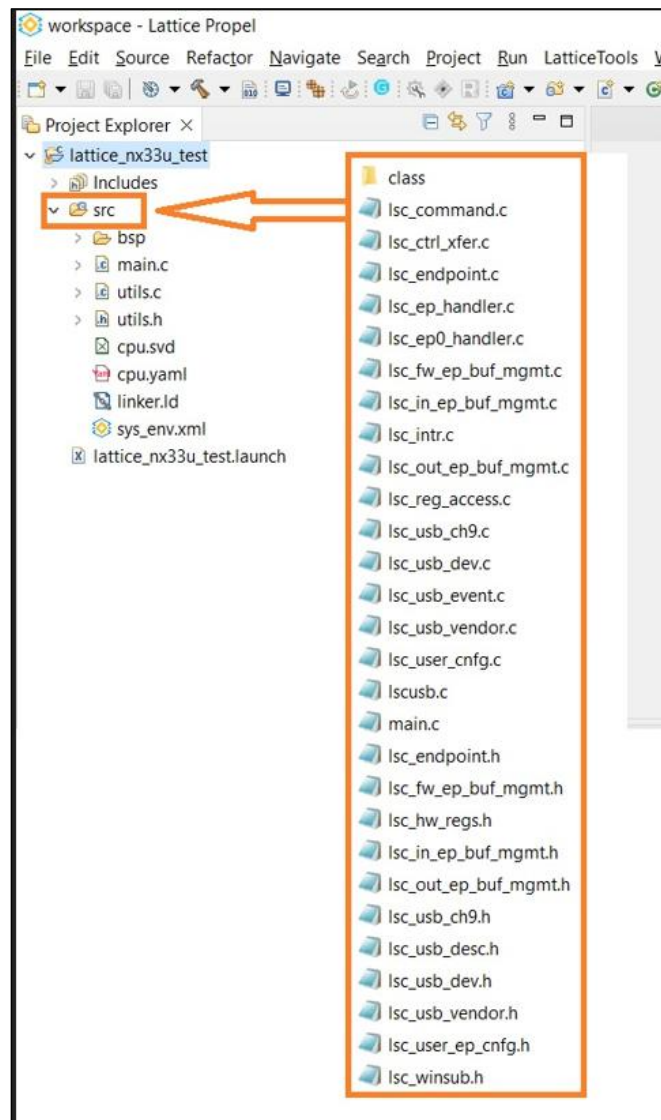


Figure 8.29. Copy all C and H Files to the src Folder

3. The following dialog box appears prompting you to overwrite the main.c file. Click **Overwrite** to overwrite the file.

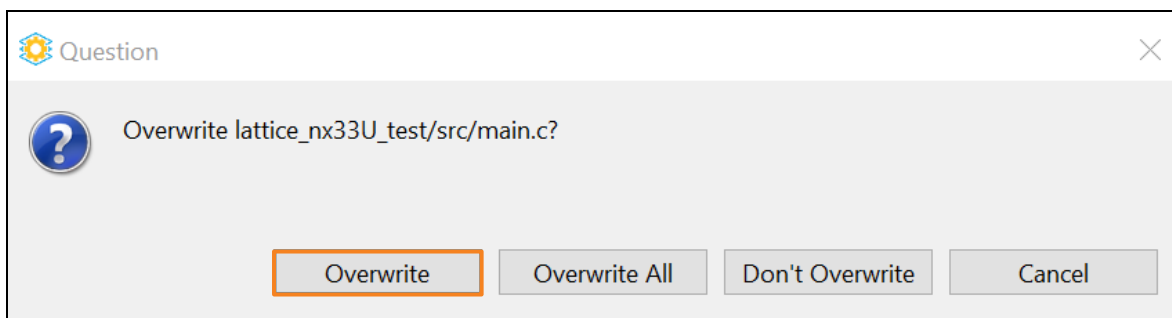


Figure 8.30. Overwrite File

4. Set the build configuration to Release mode. Select **Build configuration > Set Active > 2 Release**, as shown in the following figure.

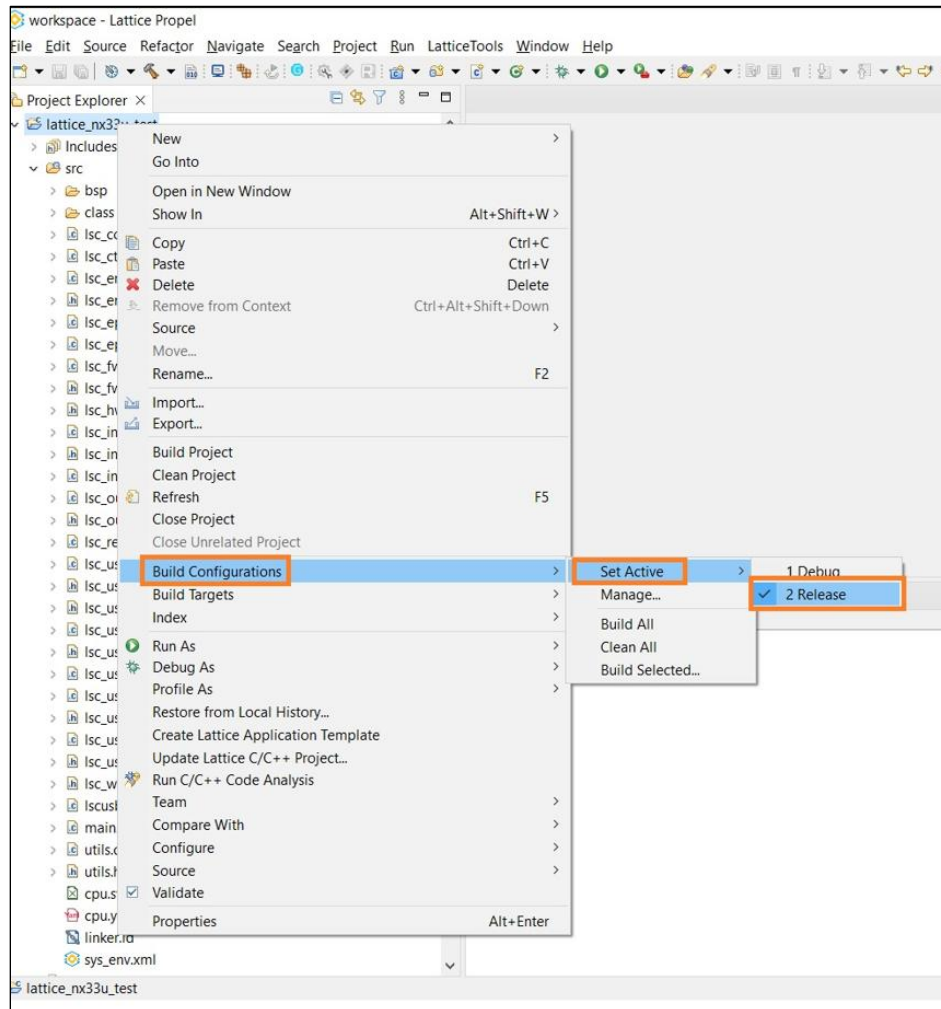


Figure 8.31. Set Build Configuration to Release Mode

- To set the optimization level to O2, right-click on **Project -> Properties-> Set optimization level to O2**.

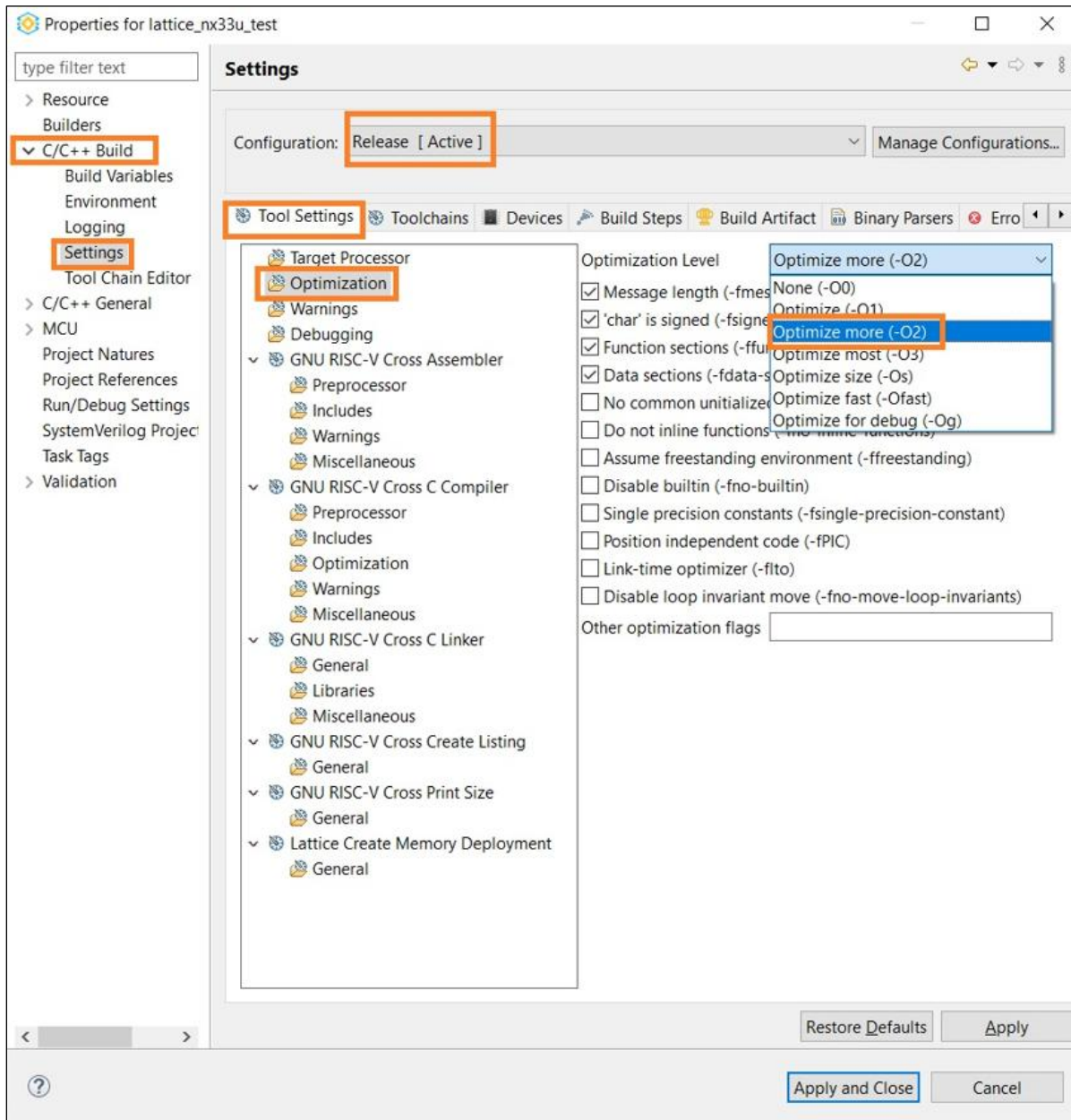


Figure 8.32. Set Optimization Level

- Open the *linker.ld* file from the src folder as shown in the following figure.

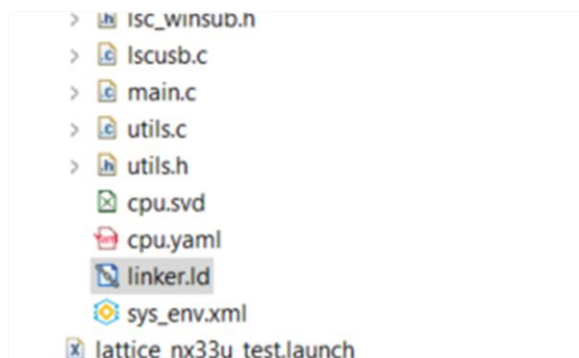


Figure 8.33. Linker ID File

- Click on the *linker.ld* tab as shown in the following figure.

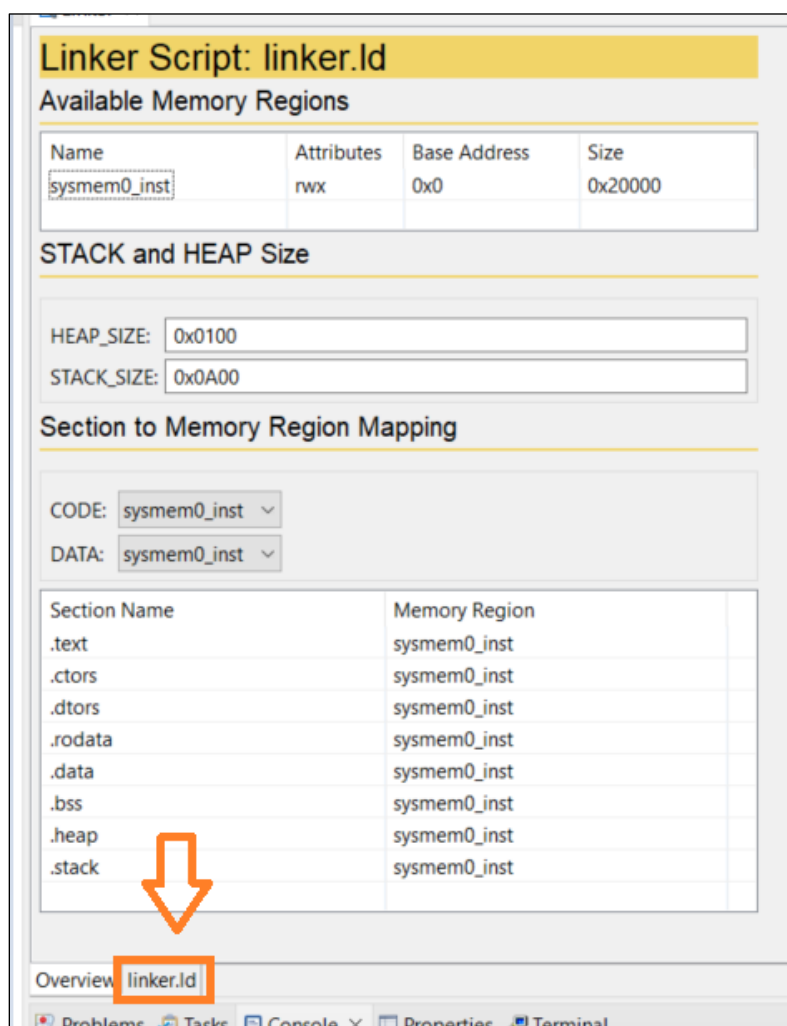


Figure 8.34. Linker ID Tab

8. The following lists the different memory section used in the design and the respective base addresses:
(Note: Verify this with the design in which you are using and update if necessary.)
- mem0: TDP—This memory is used for event buffer, control endpoint data, TRB structures etc.) Base address: 0x100000.
 - mem1: IEBM—This section is used to IEBM - IN EP buffer management) Base address: 0x140000.
 - mem2: OEBM—This section is used for OEBM - OUT EP buffer management) Base address: 0x180000.
9. Add the memory sections in the linker script according to the memory addresses shown above. Copy the following lines and paste them in the linker script as shown in the following figure:

```
system1_inst (rwx) : org = 0x100000, len = 0x18000
system2_inst (rwx) : org = 0x140000, len = 0x20000
system3_inst (rwx) : org = 0x180000, len = 0x20000
```

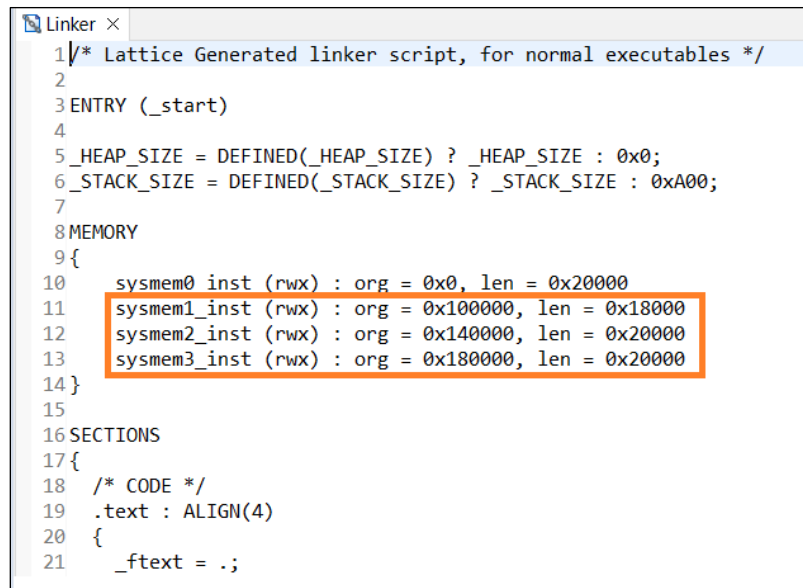


Figure 8.35. Base Address Update

10. Copy the following lines and paste them after the line stated “ }>system0_inst” as shown in the following figure:

```
/* placing my named section at given address: */
.myBufBlock 0x100000 :
{
KEEP(*(.myBufSection)) /* keep my variable even if not referenced */
} >system1_inst

/* placing my named section at given address: */
.myBufBlock_2 0x140000 :
{
KEEP(*(.myBufSection_2)) /* keep my variable even if not referenced */
} >system2_inst

/* placing my named section at given address: */
.myBufBlock_3 0x180000 :
{
KEEP(*(.myBufSection_3)) /* keep my variable even if not referenced */
} >system3_inst
```

```

109     . = ALIGN(16);
110     PROVIDE (_stack_start = .);
111     PROVIDE (_end = .);
112     PROVIDE (end = .);
113 } >system0_inst
114
115 /* placing my named section at given address: */
116 .myBufBlock 0x100000 :
117 {
118 KEEP(*(.myBufSection)) /* keep my variable even if not referenced */
119 } >system1_inst
120
121 /* placing my named section at given address: */
122 .myBufBlock_2 0x140000 :
123 {
124 KEEP(*(.myBufSection_2)) /* keep my variable even if not referenced */
125 } >system2_inst
126
127 /* placing my named section at given address: */
128 .myBufBlock_3 0x180000 :
129 {
130 KEEP(*(.myBufSection_3)) /* keep my variable even if not referenced */
131 } >system3_inst
132
133 /* END */
134 }
135

```

Figure 8.36. System Memory Instance

11. Save and close the linker script.
12. Open the *utils.h* file from the src folder.



Figure 8.37. utils.h File

13. Uncomment line no. 65 and update the value in the bracket to 72 as shown in the following figure.

Note: This step is required because UART print messages uses custom function to print timestamp as well as the actual frequency of the CPU, which is 72 MHz in this design.


```

54 #endif
55
56 typedef struct timer_reg_s {
57     volatile uint32_t cnt_l;
58     volatile uint32_t cnt_h;
59     volatile uint32_t rsvd[2];
60     volatile uint32_t cmp_l;
61     volatile uint32_t cmp_h;
62 } timer_reg_t, *timer_reg_p;
63
64 #define CPU_M_FREQUENCY (72) /* Use this line if got the true cpu frequency */
65 #if CPU0_INST_NANO_CPU
66 #define CONFIG_DEFAULT_M_HZ (3)
67 #else
68 #define CONFIG_DEFAULT_M_HZ (100)
69 #endif
70
71 #ifdef CPU_M_FREQUENCY
72 #define SYSCLK_KHZ (CPU_M_FREQUENCY * 1000)
73 #define TIMER0_INST_BASE_ADDR (CPU0_INST_BASE_ADDR + 0x400)
74 void delayMS(uint32_t ms){

```

Figure 8.38. CPU_M_Frequency Update

14. Save and close the *utils.h* file.
15. Click on the **Project Explorer** tab. Select **Project > Build Project**.

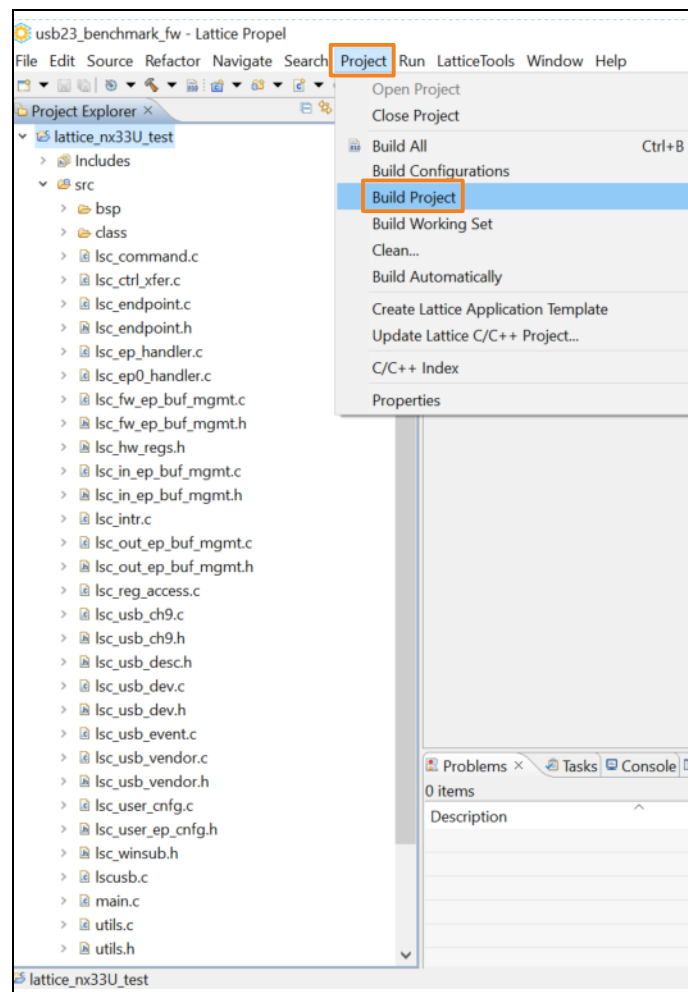


Figure 8.39. Build a Project

```

CDT Build Console [lattice_nx33U_test]
riscv-none-embed-size --format=berkeley "lattice_nx33U_test.elf"
text    data    bss    dec    hex    filename
30196   1568   39424   71188  11614  lattice_nx33U_test.elf
Finished building: lattice_nx33U_test.siz

Invoking: Lattice Create Memory Deployment
riscv-none-embed-objcopy -O binary --gap-fill 0 "lattice_nx33U_test.elf" "lattice_nx33U_test.bin"; srec_cat "lattice_nx33U_test.bin" -Binary -byte-swap 4 -DISable Header -Output
"lattice_nx33U_test.mem" -MEM 32
Finished building: lattice_nx33U_test.mem

10:39:49 Build Finished. 0 errors, 0 warnings. (took 4s.483ms)

```

Figure 8.40. Console Message

Note that the generated `.mem` is very important for merging with bitstream file.

8.3.2.3. Import Project from Directory

Note: You can skip this step if the previous steps in the [Creating the Project](#) section has been completed.

To import a project in the Lattice Propel 2024.1 (or latest) Software Build Tools for Eclipse, follow these steps:

1. Launch the Lattice Propel 2024.1 Software Build Tools for Eclipse. Select a workspace folder to import the software project and click **Launch**.
2. Select **File > Import**, as shown in the following figure.

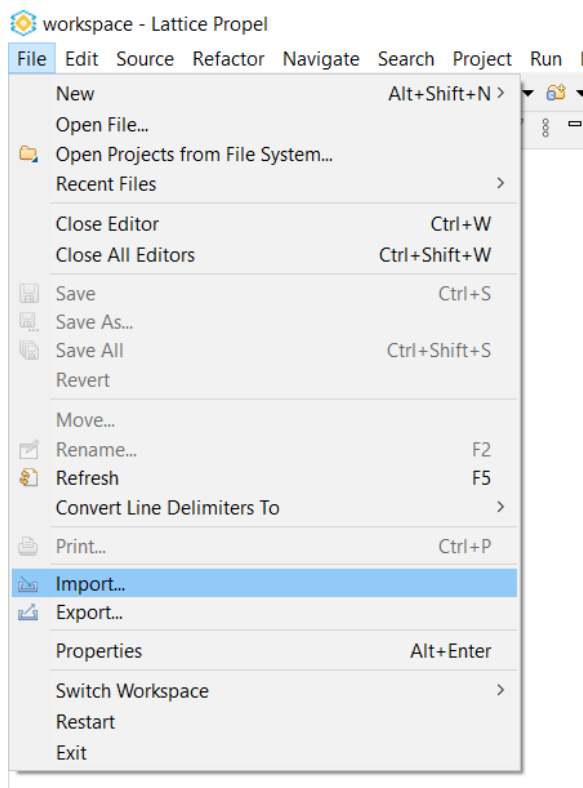


Figure 8.41. Import Project

3. A new Import pop-up window as shown in the following figure displays. Select the **Projects from Folder or Archive** option and click **Next**.

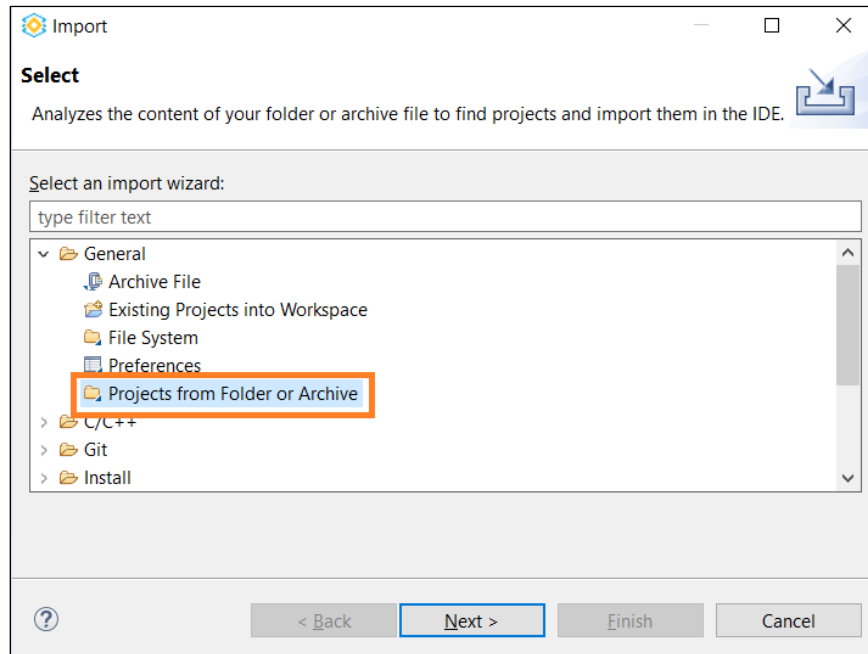


Figure 8.42. Project from Folder or Archive

4. Select the project folder path by clicking on the **Directory** button. Click **Finish**. The following lists the common path for all designs:

<installed directory>\ nx33u_cmp_revb_evn_bd_rd_des\software\usb23_validator.

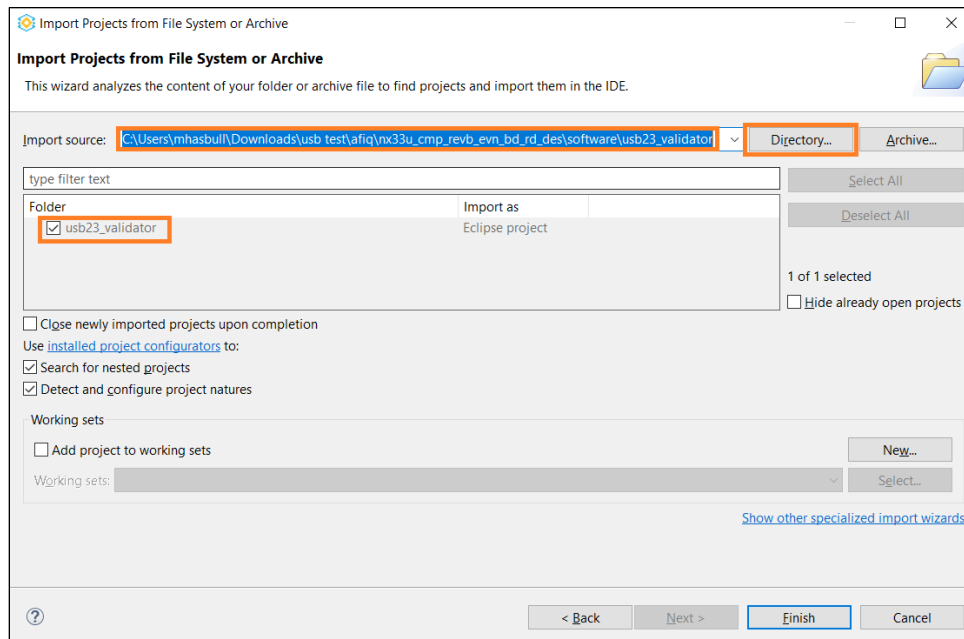


Figure 8.43. Select the Project Folder from Directory

5. Now, you can see the imported SDK project in the **Project Explorer** tab.

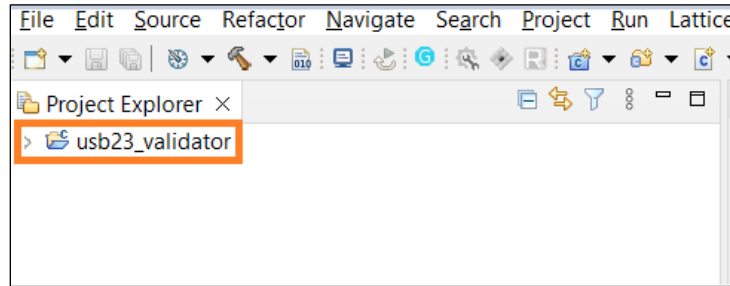


Figure 8.44 Project Exploration Tab

6. Click on the **Project Explorer** tab. Select **Project > Build Project**.

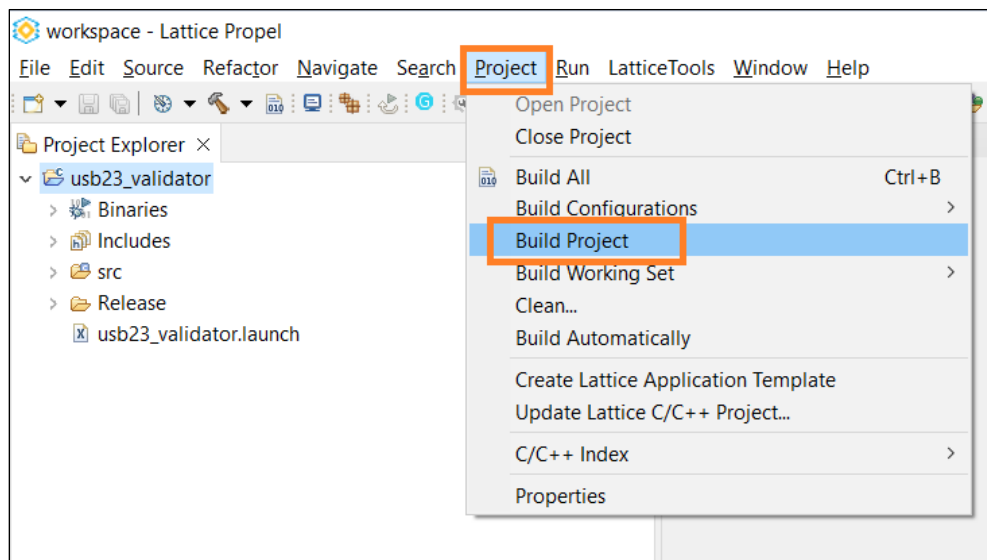


Figure 8.45 Build Project

7. After the build process is successfully completed, the console displays the following message. Note that the generated **.mem file** is very important for merging with the bitstream file.

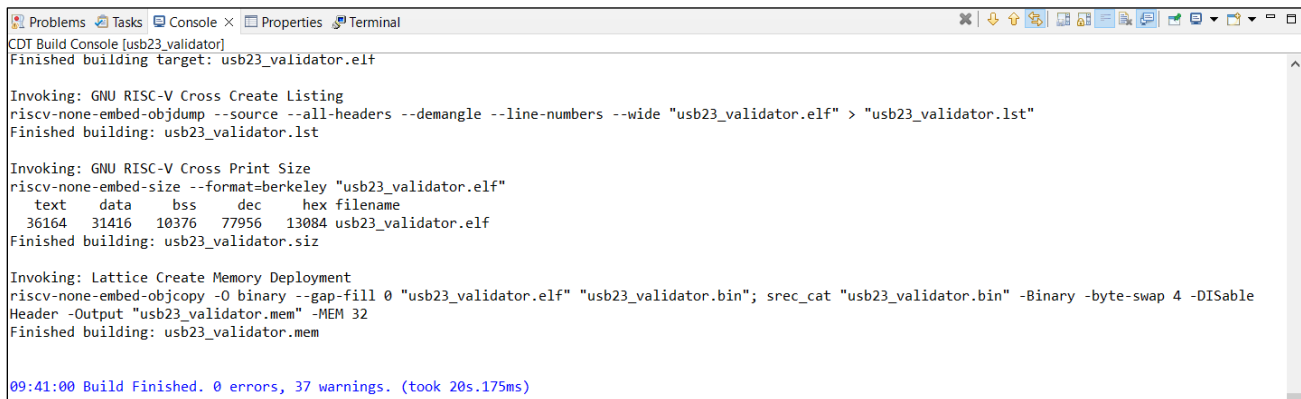


Figure 8.46 Console Message

8.3.3. Radiant Software Compilation for Bitstream

To generate the USB23 Benchmark example design, follow these steps:

1. Go to the generated IP folder (<generated_ip_folder/eval/>) containing the following files:
 - nx33u_cmp_revb_evn_bd_rd_des.zip1—Radiant software project
 - software/usb23hal_6eps_v2_2.zip1—HAL driver C code
 - application/winusb_async_app_rel_v_1_6.zip—Windows host application

Note: Ensure that the project is stored in a shorter path name or directory structure to avoid getting a File Path Too Long error.

```
application  dut_inst.v  dut_params.v  nx33u_cmp_revb_evn_bd_rd_des.zip  software
```

Figure 8.47 Eval Folder

2. Unzip the nx33u_cmp_revb_evn_bd_rd_des.zip file to obtain the files shown in the following figure.

```
axi64_to_ahbl32_conv  nx33u_cmp_revb_evn_bd_rd_des.rdf  remote_files  u23_lifclu_nx33_prpl_bldr
documents             nx33u_cmp_revb_evn_bd_rd_des_tcr.dir  reportview.xml  usb23_ip
impl_1                pdc  rtl
mipi_uvc_nx33u_evalbd_des.sty  pll_60m_i_60m_o  source
```

Figure 8.48 nx33u_cmp_revb_evn_bd_rd_des Sub Folder

3. Create a Radiant software project based on the CrossLink-NX device and generate the USB IP through the IP catalog, as shown in Figure 8.49 and Figure 8.50.

Figure 8.49. Radiant Device Setup

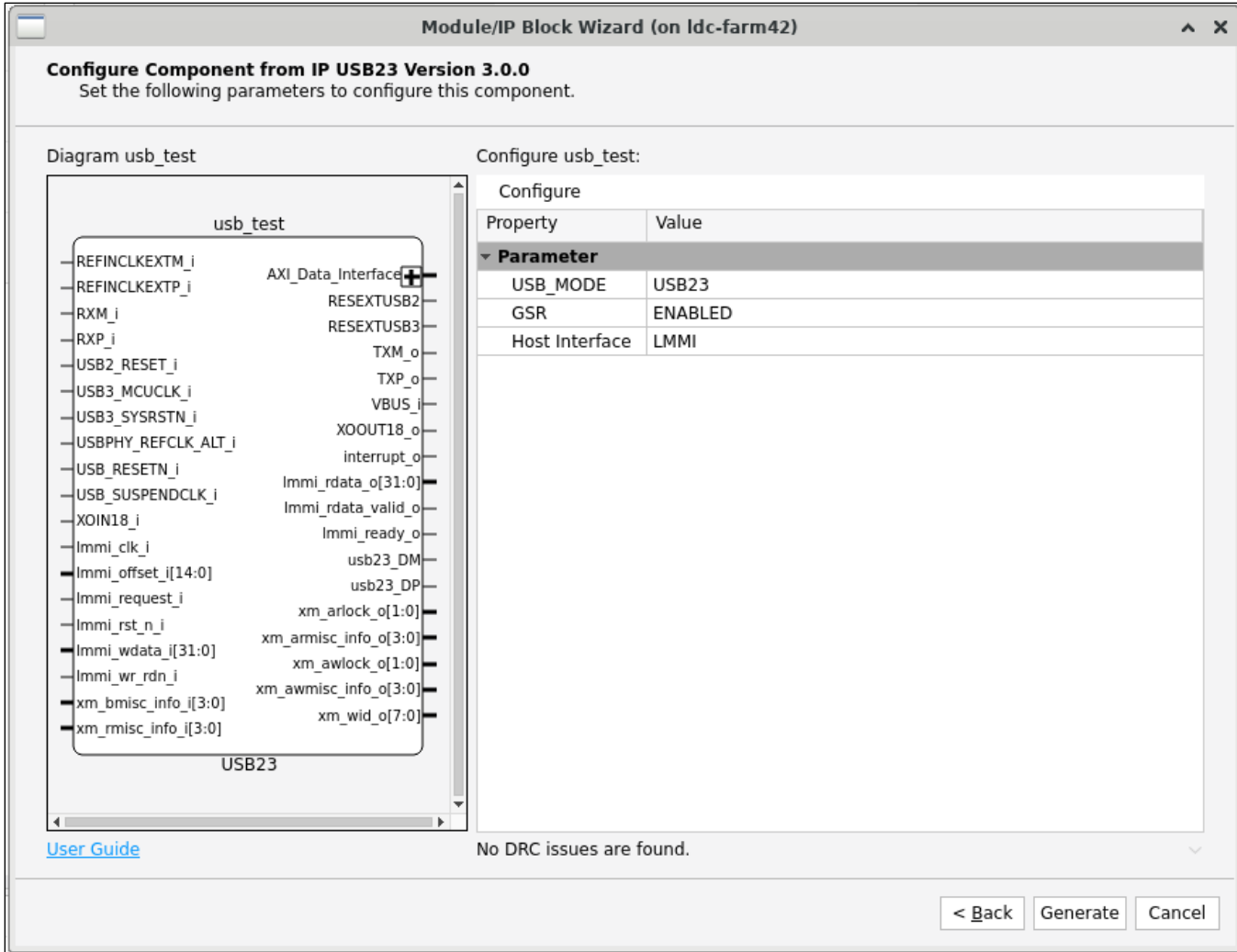


Figure 8.50. USB IP GUI Option

4. To compile the Radiant software project, open the `<Generated_USB_IP>/eval/nx33u_cmp_revb_evn_bd_rd_des/nx33u_cmp_revb_evn_bd_rd_des.rdf` file with the latest Radiant software v2024.2 version. To open a Radiant software project, go to **File -> Open -> Project** as shown in the following figure.

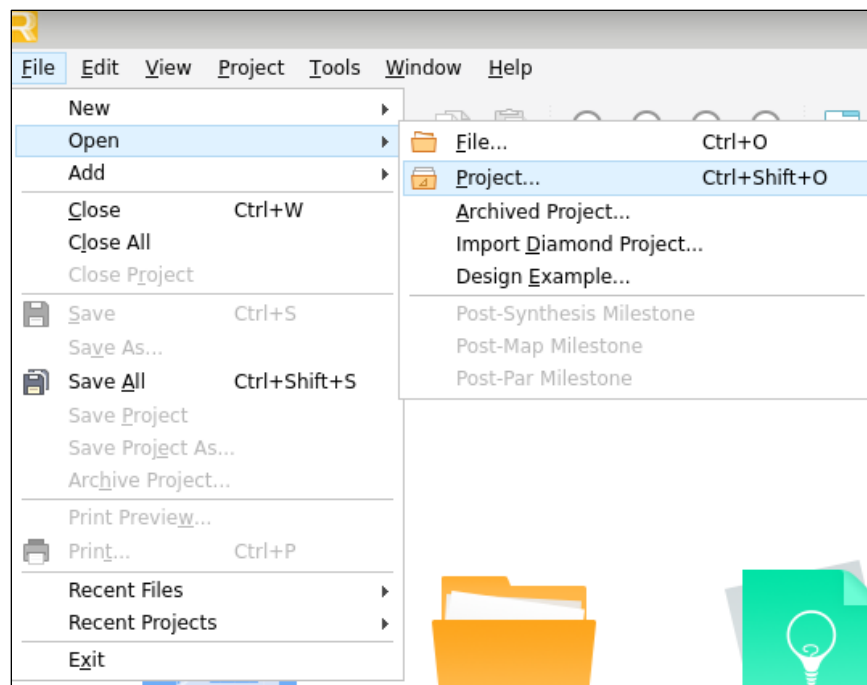


Figure 8.51. Open a Radiant Software Project

5. The following figure shows a project summary of the `nx33u_cmp_revb_evn_bd_rd_des.rdf` file in the Radiant software.

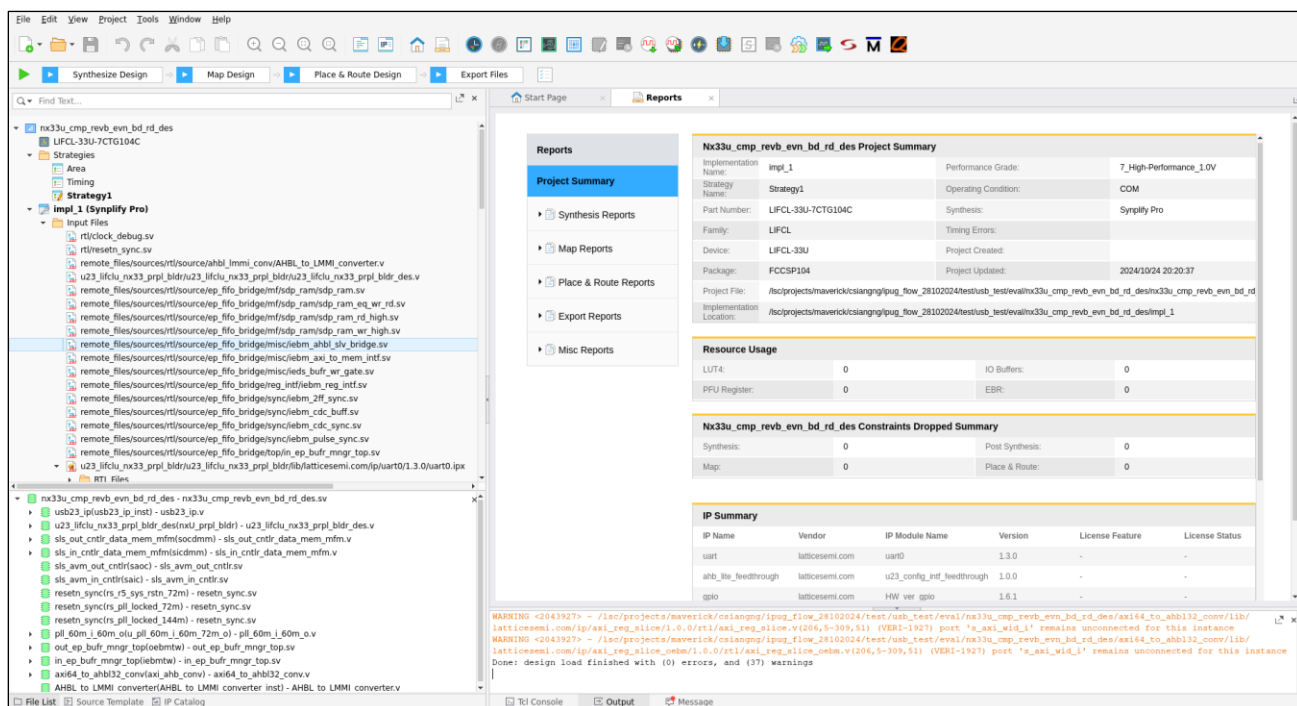


Figure 8.52. USB23 Benchmark Example Design Radiant Software Project

6. To generate the bitstream, click on the **Export Files** icon as shown in the following figure.

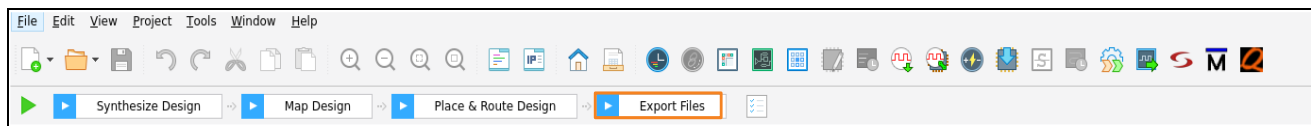


Figure 8.53. Bitstream Generation

Note: It is advisable that you run more iterations if you encounter timing violations. To increase the number of iterations for Place and Route, go to **Project -> Active Strategy -> Place & Route Design Settings** as shown in the following figure.

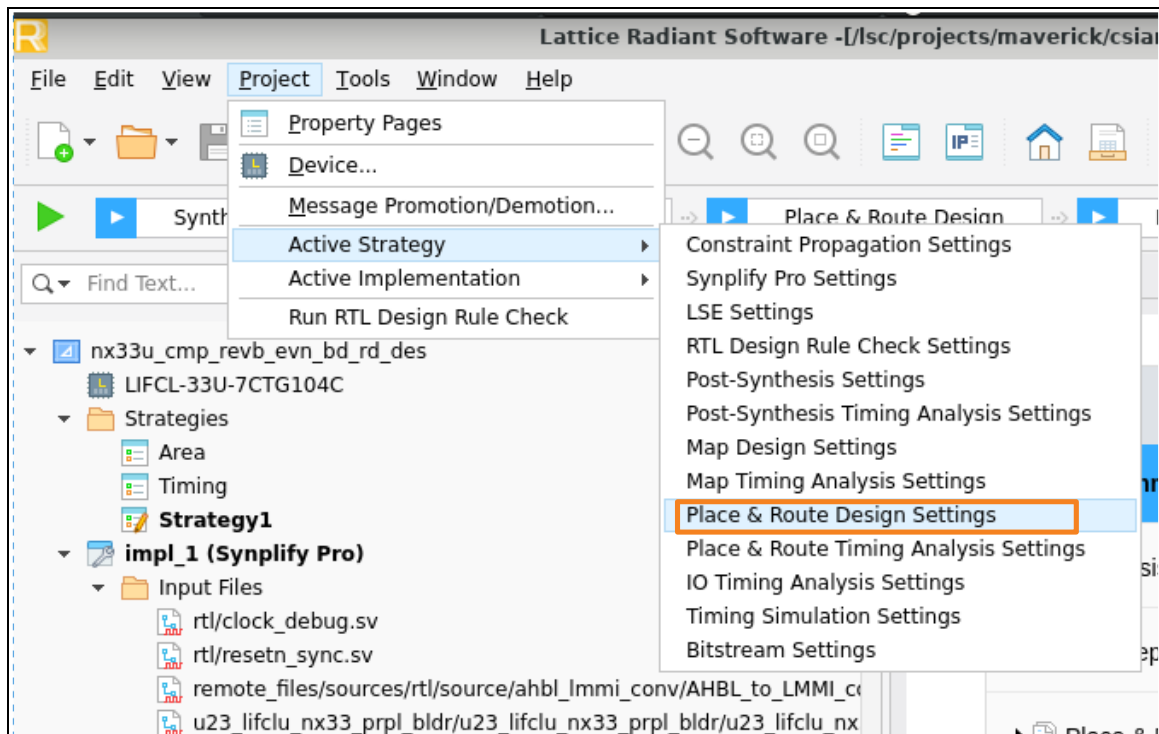


Figure 8.54. Place & Route Design Settings

The following figure shows that you can increase the number of Placement Iteration Start Point and Placement Iterations to obtain a better timing result.

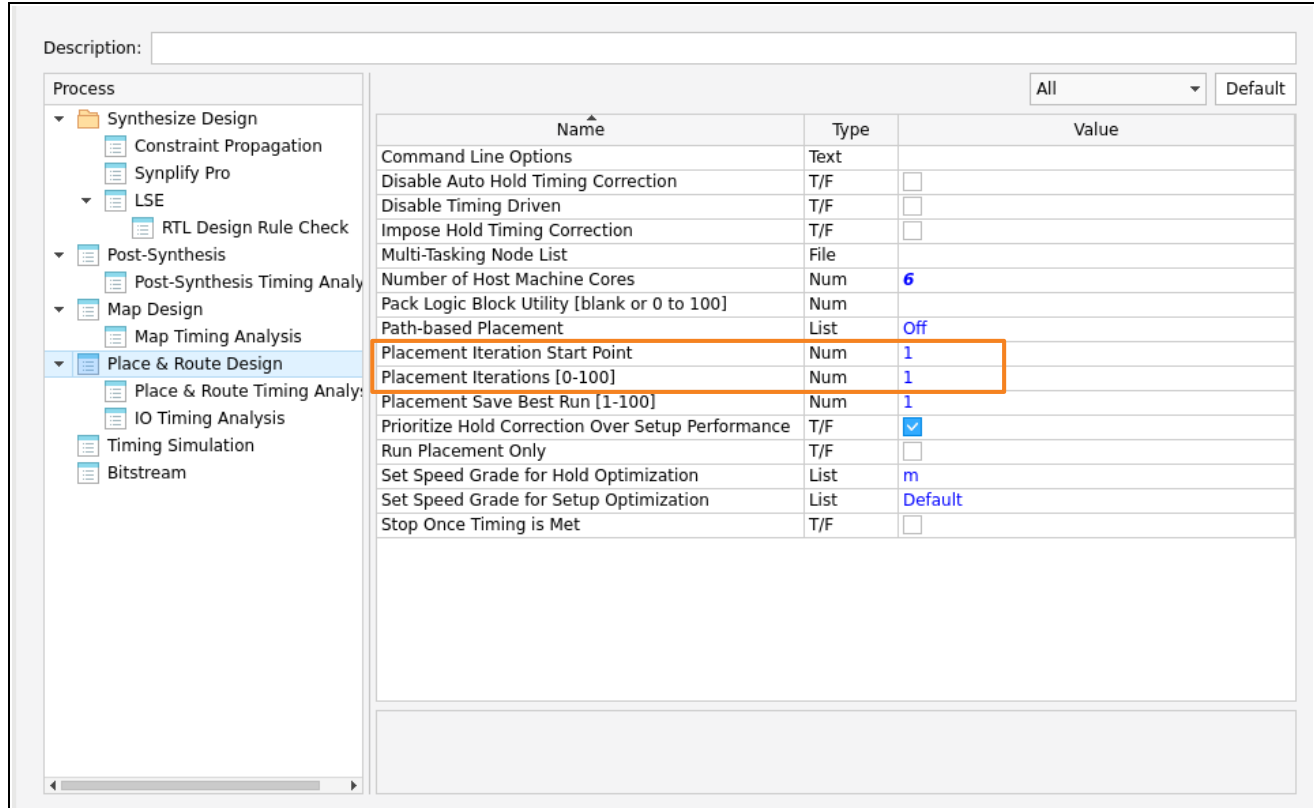


Figure 8.55. Placement Iteration

- After generating the bitstream successfully, find the `nx33u_cmp_revb_evn_bd_rd_des_impl_1.bit` file in the `impl_1` folder.

8.3.4. Trim the Memory File

The following steps describe how to trim the `.mem` file.

We are using two separate memory regions by manually adding a second memory to the linker script. As a result, the `.mem` file locations exceed 32K, while our system memory depth is limited to 32K, as shown in Figure 8.58. Follow these steps to trim the memory file and use the truncated `.mem` file in the ECO editor tool.

To trim the memory file, follow these steps:

- Launch the Windows command prompt.
- Change directory to where your baremetal project's Release folder is located.
- Locate where your Propel 2024.2 SDK is installed, and you must invoke the tools inside.
- Make sure the command is copied correctly because white space causes an error. You must also ensure that the `.elf`, `.bin`, and `.mem` filename in both commands follows the project name.
- In the command prompt, type:
 - `C:\lscc\propel\2024.2\sdk\riscv-none-embed-gcc\riscv-none-embed\bin\objcopy.exe -O binary --remove-section=.myBufBlock RiscVMC_Test.elf RiscVMC_Test.bin`
 - `C:\lscc\propel\2024.2\sdk\tools\bin\srec_cat.exe RiscVMC_Test.bin -Binary -byte-swap 4 -Disable Header -Output RiscVMC_Test.mem -MEM 32`

Make sure the terminal return nothing (no error) when executing both commands as shown in Figure 8.56. The memory file locations are now trimmed.

Note: After the 1st command is successfully executed, the *.bin* file in the Release folder is updated and after the 2nd command is successfully executed, the *.mem* file is updated in the Release folder.



```

C:\Users\mhasbull>cd C:\lsc\propel\2024.2\workspace\RiscVMC_Test\Release

C:\lsc\propel\2024.2\workspace\RiscVMC_Test\Release>"C:\lsc\propel\2024.2\sdk\riscv-none-embed-
gcc\riscv-none-embed\bin\objcopy.exe" -O binary --gap-fill 0 --remove-section=.myBufBlock RiscVMC
_Test.elf RiscVMC_Test.bin

C:\lsc\propel\2024.2\workspace\RiscVMC_Test\Release>"C:\lsc\propel\2024.2\sdk\tools\bin\srec_cat
t.exe" RiscVMC_Test.bin -Binary -byte-swap 4 -DISable Header -Output RiscVMC_Test.mem -MEM 32

C:\lsc\propel\2024.2\workspace\RiscVMC_Test\Release>
  
```

Figure 8.56. Trim the Memory File

8.3.5. Merging *.mem* with Bitstream File

The following steps describe how to add the *.mem* file that had been trimmed and merge it into the bitstream file from the Radiant software:

1. To merge the *.mem* file with the bitstream file, make sure you have completed the flow listed in the [Building the Project](#) section to generate the *.mem* file.
2. To merge the software application file (*.mem*) with the programming file (*.bit*), use the ECO Editor within the Radiant software tool. Make sure you have completed the necessary flows listed in the [Generating the USB23 Benchmark Example Design](#) section, [HAL Driver Compilation](#) section, and [Radiant Software Compilation for Bitstream](#) section.

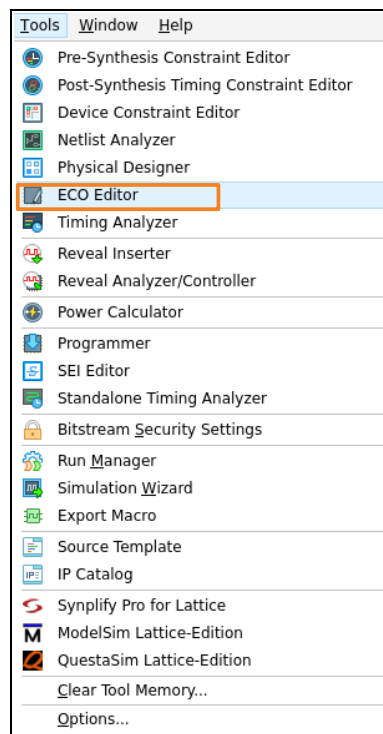


Figure 8.57. ECO Editor

3. Select **Tools>ECO Editor** to open the ECO Editor. Click on the **Memory Initialization** tab and select the first memory initialization as shown in the following figure.

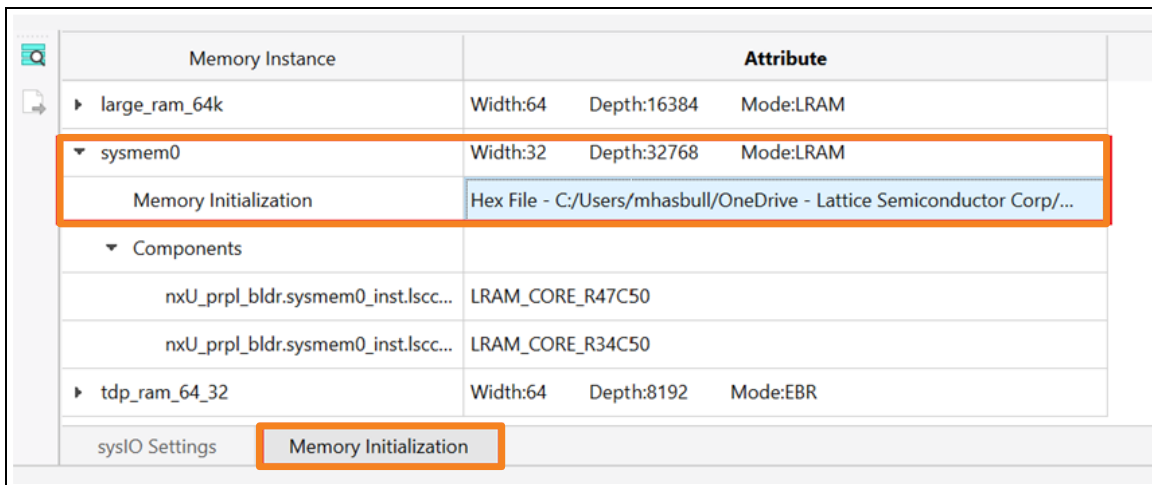


Figure 8.58. Memory Initialization

4. Select the **.mem** file that you have generated from the steps in the [Building the Project](#) section as shown in the following figure.

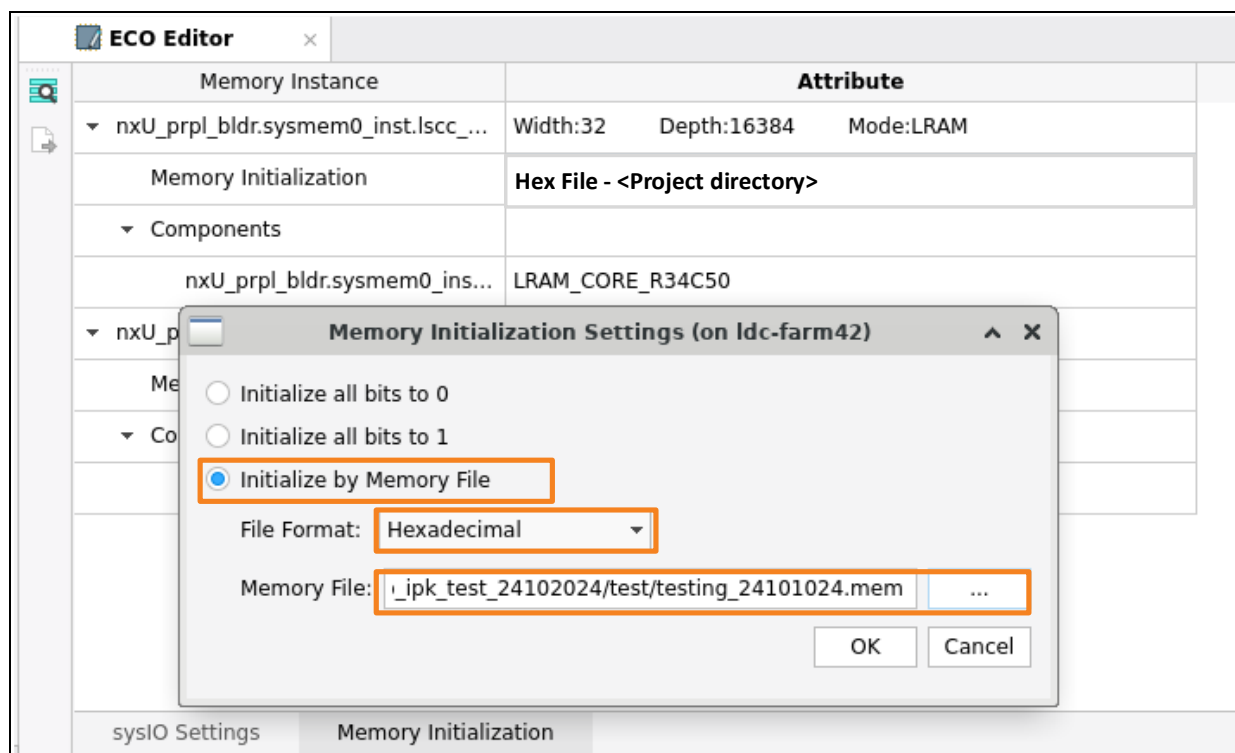


Figure 8.59. Memory Initialization Settings

5. Save the settings in the ECO Editor and click **Export Files** to generate the bitstream file.

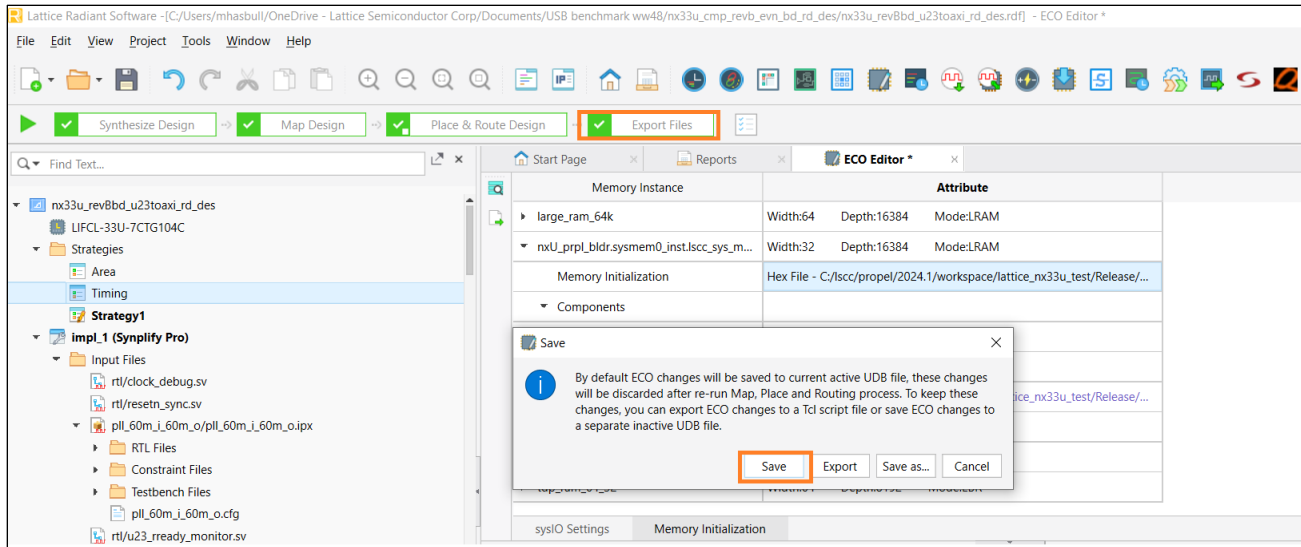


Figure 8.60. Save the Settings in the ECO Editor and Generate Bitstream

8.3.6. Hardware Testing

This section describes the steps to perform hardware testing or data streaming.

The following jumpers must be set accordingly and in the CLOSED position:

- J8 (Short pins 1-2)
- J23
- J40
- J37
- J41
- J56

The rest of the jumpers must be in the OPEN position.

8.3.7. FPGA Bitstream Programming

This test process uses the **default programming file** (bit file) provided with the with the example design. Therefore, you have extracted the **nx33u_cmp_revb_evn_bd_rd_des** example design project. If you have changed the design or code, make sure to regenerate the bitstream.

Follow these steps to program the FPGA bitstream:

1. Power-on the board by connecting the USB cable (Type A to micro B) to J2 of the CrossLinkU-NX REVB Evaluation Board.
2. The board is programmed using the Lattice Radiant Programmer software, which can be launched as a stand-alone tool or from a Lattice Radiant software project.
3. Start a programming project by launching the tool and initiating a board scan.
4. When the board has been scanned successfully, the **Radiant Programmer - Getting Started** window shown in the following figure displays. Enter the project name, and click **OK**.

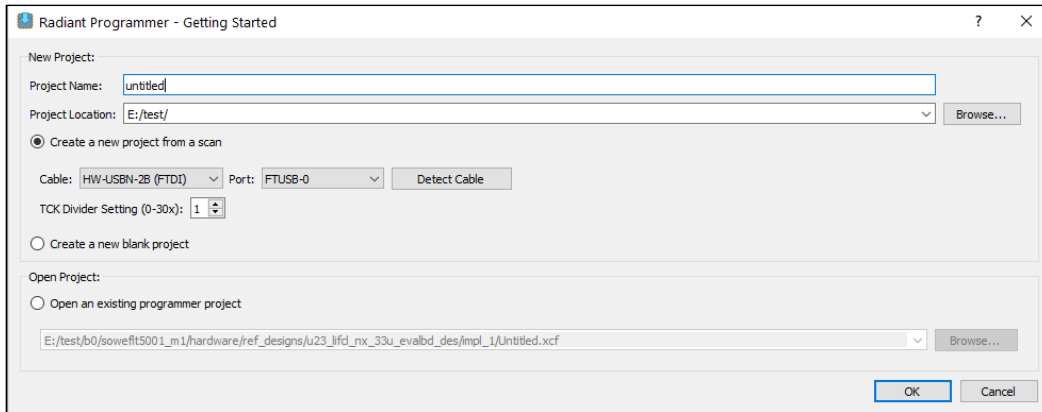


Figure 8.61. Radiant Programmer Getting Started Window

5. Double-click on the **Operation** field and select the appropriate programming mode. The following lists two programming options:
 - Fast Configuration of the SRAM array
 - External SPI Flash memory

For **Fast Configuration of the SRAM array** option through JTAG, follow these steps:

- a. Browse programming file (bit file) from following location: *<Extracted example design project directory>\impl_1*.
- b. Click **OK**. Click the Program button to configure the LIFCL-33U on the board as shown in the following figure.

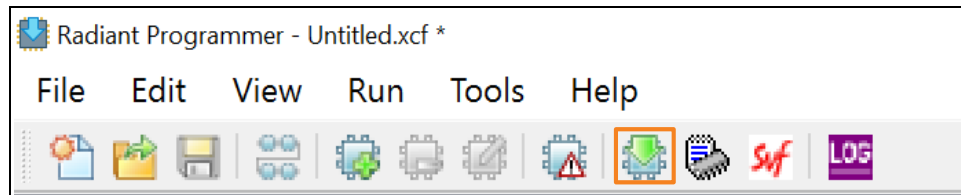


Figure 8.62. Program Button

- c. Check the Output console at the bottom of Lattice Radiant Programmer for the result of the programming.

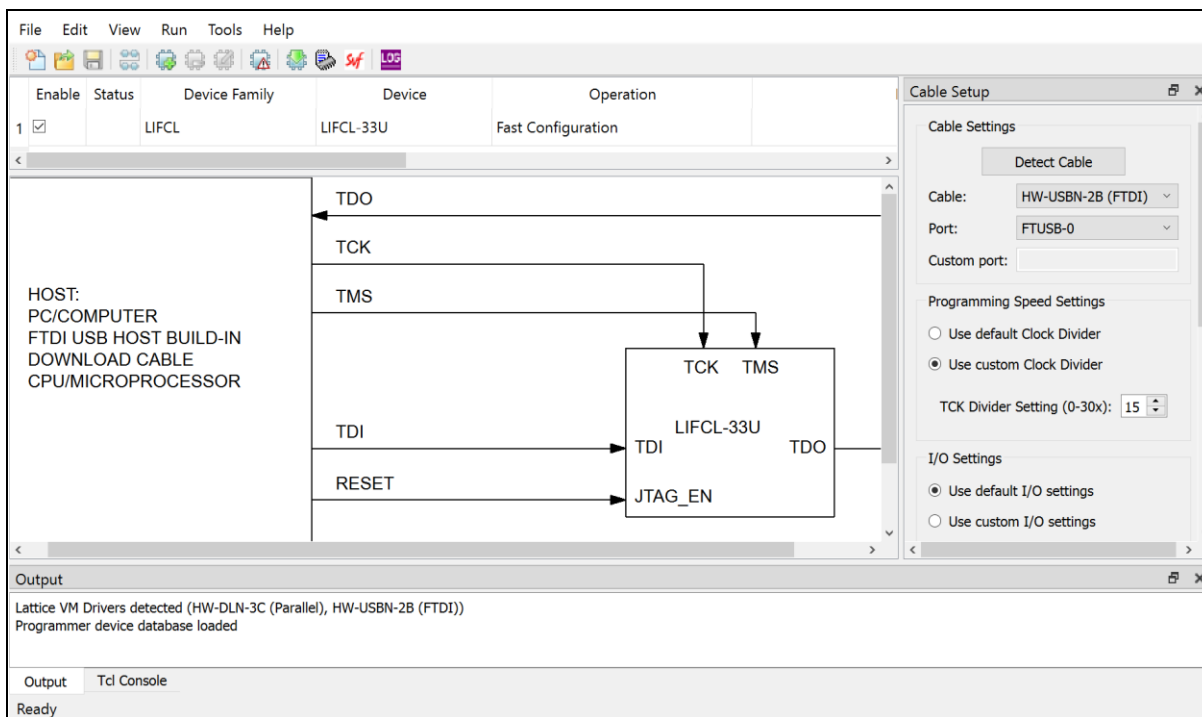


Figure 8.63. Select Programming Mode

For External SPI Flash memory option, follow these steps:

- Make sure that the board is turned off.
- Short jumper J5 on the board.
- Select the appropriate settings in the Radiant programmer as shown in the following figure.

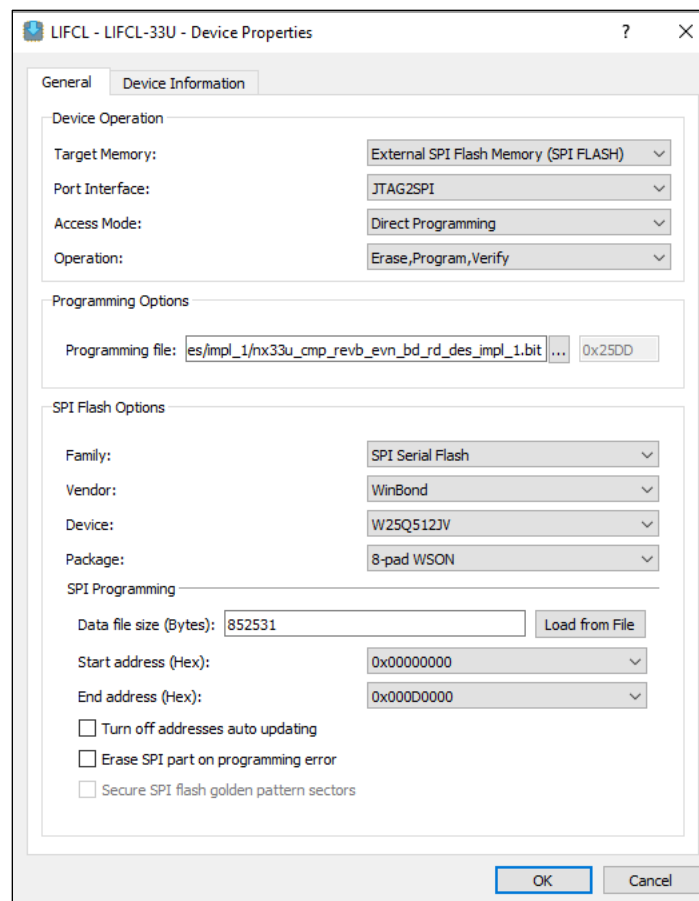


Figure 8.64. SPI Flash Programmer Settings

6. Browse to the programming file (bit file) at the following location: *<Extracted example design project directory>\impl_1*.
7. Click **OK**. Click the Program button to configure the LIFCL-33U on the board.
8. Check the **Output** console at the bottom of Lattice Radiant Programmer for the result of the programming.
9. After the flash programming is done, turn off the board by unplugging the USB cables.
10. Remove jumper J5 connection on the board
11. Turn on the board by plugging Type-A to Micro Type-B cable.

8.3.8. Performing Streaming Operations

This section describes the steps to perform streaming operations over USB via different types of USB transfers such as Bulk, Interrupt, and Isochronous. Note that the Bulk/Interrupt and Isochronous hardware results may vary across different platforms or setups.

To perform streaming operations, follow these steps:

1. Connect USB Type C cable to the board's USB Type C connector and USB Type C to your PC or laptop.
2. Open the Device manager from your PC or laptop.
3. The USB device is enumerated as shown in the following figure. If the Lattice USB23 is not detected, this means that the enumeration process has failed.

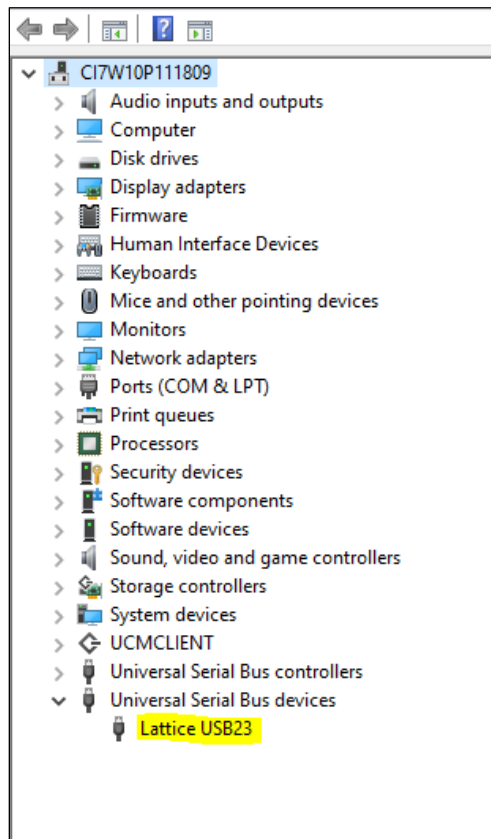


Figure 8.65. USB Enumeration

- To obtain the device connection speed, use the USB Device Tree Viewer tool.

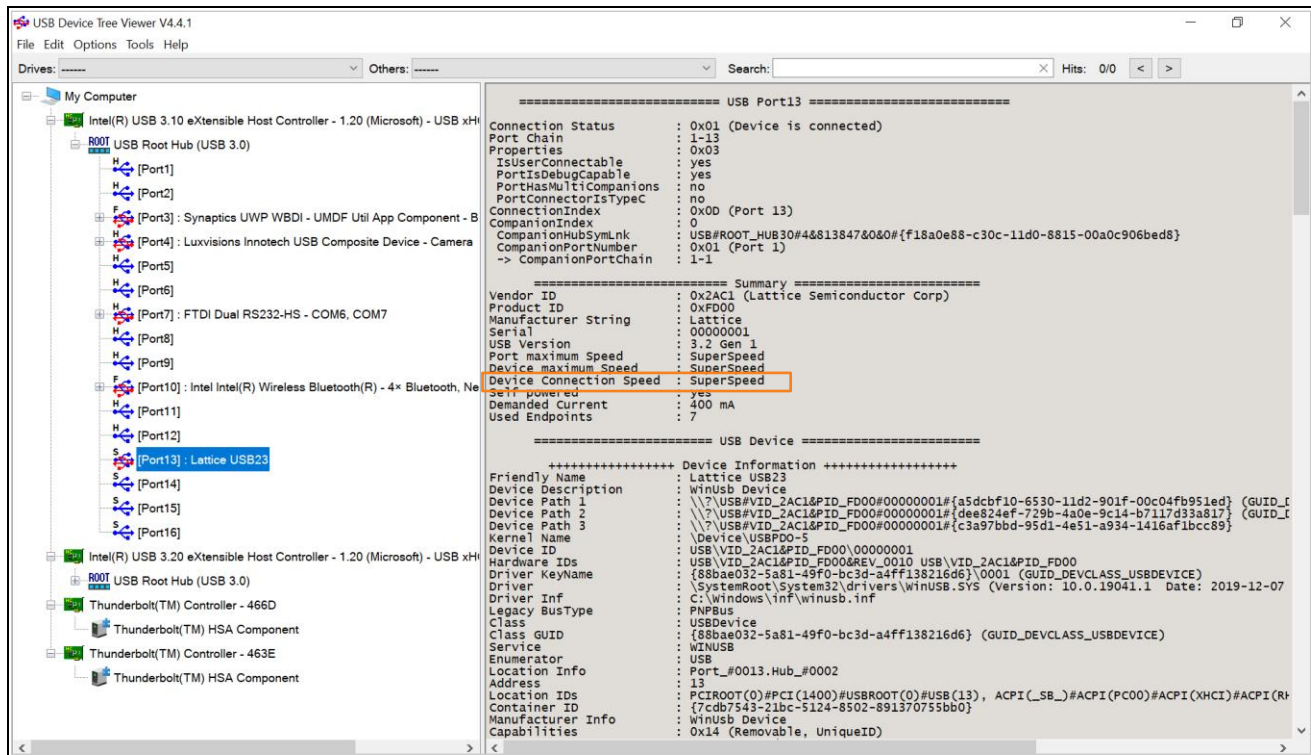


Figure 8.66. USB Device Tree Viewer

- Unzip the *winusb_async_app_rel_v_1_6.zip* file to obtain the WinUSB Asynchronous CLI application.
- Double-click on the *winusb_async_app.exe* file.

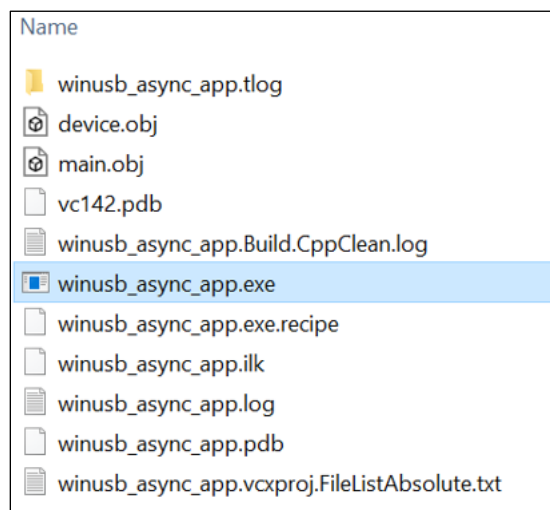


Figure 8.67. WinUSB Asynchronous CLI Application

8.3.8.1. Obtaining Device Information

To obtain the device information, follow these steps:

1. At the command prompt, select option 1.
2. The WinUSB API returns the following options:
 - 0x3: High-speed or higher operating speed.
 - 0x1: Full-speed or lower operating speed.

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endpoints: 0
Alt_set: 1      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...1
Device found: VID_2AC1&PID_FD00; bcdUsb 0320
Device speed: 3 (High speed or Higher operating speed).

EXIT Y:N
    
```

Figure 8.68. Obtaining Device Information through WinUSB API

8.3.8.2. USB3 Bulk IN/OUT Test

To perform the USB3 Bulk IN/Bulk OUT test, follow these steps:

1. To perform the Bulk IN test, select option 2.
2. Enter the following information:
 - Alternate setting number: 2
 - Endpoint number with direction (in hex): 82
 - Number of times
 - Number of bytes to read.

Follow these rules when selecting the number of bytes to read:

Number of bytes to read must be in multiple of endpoint maximum packet size. This is because, the current hardware design for streaming always sends packets with size equals to the endpoint's maximum packet size for an IN endpoint. This applies to the Interrupt IN and Bulk IN transfers.

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...2
Enter Alternate Setting number: 2
Enter Endpoint No with Direction (in hex): 82
Enter Number of times (Enter 0 for Infinite loop): 100
Number of bytes to be read shall be in multiple of Endpoint's maximum packet size and <= 32 MB.
Enter Number of Bytes to Read: 1048576
#      #      #      #      #      #      #      #      #      #      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
Average Speed(IN Mbits/second) for iteration 100 = 3361.344
Speed = 3347.280 Mbits/second
EXIT Y:N _

```

Figure 8.69. Bulk IN Test

3. To perform the Bulk OUT test, select option 3.
4. Enter the following information:
 - Alternate setting number: 2
 - Endpoint number with direction (in hex): 2
 - Number of times (Enter 0 for Infinite loop).
 - Number of bytes to write (Valid range: 1 to 32 MB)

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endpoints: 0
Alt_set: 1      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...3
Enter Alternate Setting number: 2
Enter Endpoint No with Direction (in hex) : 2
Enter Number of times (Enter 0 for Infinite loop): 100
Enter Number of Bytes to Write (Valid Range: 1 to 32 MB ): 1048576
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
Average Speed(IN Mbits/second) for iteration 100 = 3292.181
*
Speed = 3278.688 Mbits/second
EXIT Y:N

```

Figure 8.70. Bulk OUT Test

8.3.8.3. USB3 Interrupt IN/OUT Test

To perform the Interrupt IN/OUT test, follow these steps:

1. To perform the Interrupt IN test, select option 2.
2. Enter the following information:
 - Alternate setting number: 3
 - Endpoint number with direction (in hex): 83
 - Number of times (Enter 0 for Infinite loop).

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...2
Enter Alternate Setting number: 3
Enter Endpoint No with Direction (in hex): 83
Enter Number of times (Enter 0 for Infinite loop): 100
Number of bytes to be read shall be in multiple of Endpoint's maximum packet size and <= 32 MB.
Enter Number of Bytes to Read: 1048576
# # # # # # # # # # * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Average Speed(IN Mbits/second) for iteration 100 = 187.091
Speed = 187.091 Mbits/second
EXIT Y:N

```

Figure 8.71. Interrupt IN Test

3. To perform the Interrupt OUT test, select option 3.
4. Enter the following information:
 - Alternate setting number: 3
 - Endpoint number with direction (in hex): 3
 - Number of times (Enter 0 for Infinite loop).
 - Number of bytes to write (Valid range: 1 to 32 MB)

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...3
Enter Alternate Setting number: 3
Enter Endpoint No with Direction (in hex) : 3
Enter Number of times (Enter 0 for Infinite loop): 100
Enter Number of Bytes to Write (Valid Range: 1 to 32 MB ): 1048576
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
Average Speed(IN Mbits/second) for iteration 100 = 185.658
*
Speed = 185.572 Mbits/second
EXIT Y:N _

```

Figure 8.72. Interrupt OUT Test

8.3.8.4. USB3 ISOC IN/OUT Test

To perform the Isochronous IN/OUT test, follow these steps:

1. To perform the Isochronous IN test, select option 5.
2. Alternate setting number: 1.
3. Enter the number of times (range: 1 to 50).

Based on the number of times entered, the **Isochronous IN transfer size** and **maximum transfer size** are automatically calculated.

4. The Isochronous IN transfer is initiated.

```
Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endpoints: 0
Alt_set: 1      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----

Enter your Selection...5
Enter Alternate Setting number: 1
Enter Number of times (Range: 1 to 50): 5
totalTransferSize: 307200
szBuffer: 307200
IsochInPacketCount: 20
isochPackets: 1200
IsochInTransferSize: 61440
Requested 61440 bytes in 20 packets per transfer.
Transfer 1 completed. Read 55296 bytes.

Requested 61440 bytes in 20 packets per transfer.
Transfer 2 completed. Read 61440 bytes.

Requested 61440 bytes in 20 packets per transfer.
Transfer 3 completed. Read 61440 bytes.

Requested 61440 bytes in 20 packets per transfer.
Transfer 4 completed. Read 61440 bytes.

Requested 61440 bytes in 20 packets per transfer.
Transfer 5 completed. Read 61440 bytes.

EXIT Y:N
```

Figure 8.73. Isochronous IN Test

5. To perform the Isochronous OUT test, select option 4.
6. Enter the following information:
 - Alternate setting number: 1.
 - Number of times (Range: 1 to 50).
Based on the number of times entered, the **Isochronous IN transfer size** and **maximum transfer size** are automatically calculated.
 - Number of bytes to be written (Valid range: 1 to 61440). This indicates the number of bytes that can be transmitted in single service interval.
7. The Isochronous OUT transfer is initiated.

```

Application Version: 1.7
bMaxPacketSize0: 9
Alt_set: 0      Number of Endpoints: 0
Alt_set: 1      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 3072
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 1024.
O_MaximumBytesPerInterval: 3072
Alt_set: 2      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 1024.
Alt_set: 3      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...4
Enter Alternate Setting number: 1
Enter Number of times (Range: 1 to 50): 10
IsochOutTransferSize: 61440
Total Transfer Size: 614400
Enter Number of Bytes to be written (Valid range : 1 to 61440) :61440
Write transfer 0 completed.
Write transfer 1 completed.
Write transfer 2 completed.
Write transfer 3 completed.
Write transfer 4 completed.
Write transfer 5 completed.
Write transfer 6 completed.
Write transfer 7 completed.
Write transfer 8 completed.
Write transfer 9 completed.

EXIT Y:N _

```

Figure 8.74. Isochronous OUT Test

Important note: For Isochronous IN and OUT transfers, the firmware follows *on-demand transfer* flow as per USB23 IP core's programming guide. In this (on-demand transfer) case, the firmware initiates requests after receiving notification that Isochronous IN/OUT transfers have been initiated by the host. The notification is received via the *Transfer not ready* event. Because of this, the first few (3 to 4) requests are dropped. In case of Isochronous IN, the USB23 IP automatically send 0 length data while for Isochronous OUT, data will be lost. This is an expected behavior.

8.3.8.5. USB2 Bulk IN/OUT TEST

To obtain the speed of the connected device, use the USB device tree tool to determine. The High-Speed mode appears if USB2 connection is successfully connected.

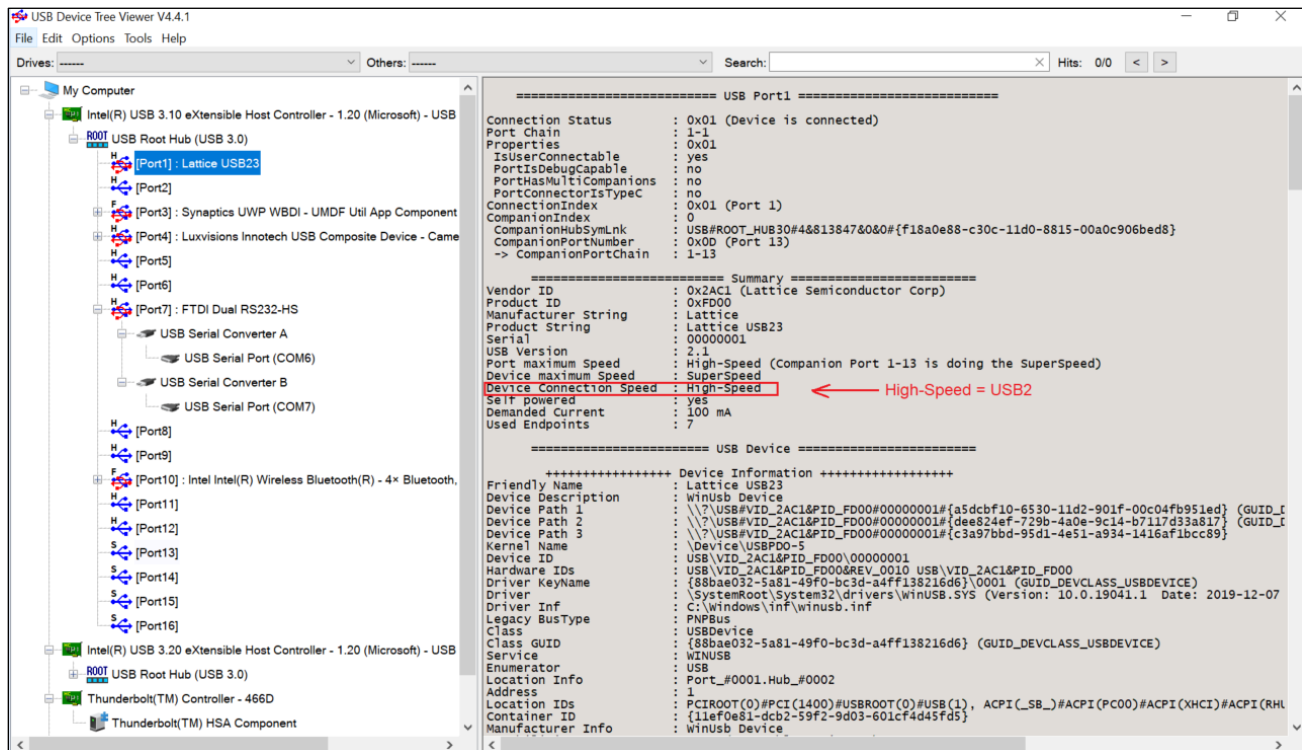


Figure 8.75. USB2 Connection Speed

To perform the USB2 Bulk IN/Bulk OUT test, follow these steps:

1. To perform the Bulk IN test, select option 2.
2. Enter the following information:
 - Alternate setting number: 2.
 - Endpoint number with direction (in hex): 82
 - Number of times.
 - Number of bytes to read.

Follow these rules when selecting the number of bytes to read:

The number of bytes to read must be in multiple of endpoint maximum packet size. This is because the current hardware design for streaming always sends packets with size equals to the endpoint's maximum packet size for an IN endpoint. This applies to the Interrupt IN and Bulk IN transfers.

```
Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512.
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...2
Enter Alternate Setting number: 2
Enter Endpoint No with Direction (in hex): 82
Enter Number of times (Enter 0 for Infinite loop): 100
Number of bytes to be read shall be in multiple of Endpoint's maximum packet size and <= 32 MB.
Enter Number of Bytes to Read: 1048576
# # # # # # # # # #
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
Avrage Speed(IN Mbits/second) for iteration 100 = 343.643
Speed = 343.495 Mbits/second
EXIT Y:N
```

Figure 8.76. USB2 Bulk IN Test Result

3. To perform the Bulk OUT test, select option 3.
4. Enter the following information:
 - Alternate setting number: 2.
 - Endpoint number with direction (in hex): 2
 - Number of times (Enter 0 for Infinite loop)
 - Number of bytes to write (Valid range: 1 to 32 MB)

```

Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512.
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...3
Enter Alternate Setting number: 2
Enter Endpoint No with Direction (in hex) : 2
Enter Number of times (Enter 0 for Infinite loop): 100
Enter Number of Bytes to Write (Valid Range: 1 to 32 MB ): 1048576
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
*      *      *      *      *      *      *      *      *      *      *      *      *      *      *
Average Speed(IN Mbits/second) for iteration 100 = 339.559
*
Speed = 339.271 Mbits/second
EXIT Y:N

```

Figure 8.77. USB2 Bulk OUT Test Result

8.3.8.6. USB2 Interrupt IN/OUT Test

To perform the USB2 Interrupt IN/OUT test, follow these steps:

1. To perform the Interrupt IN test, select option 2.
2. Enter the following information:
 - Alternate setting number: 3.
 - Endpoint number with direction (in hex): 83
 - Number of times
 - Number of bytes to read.

Follow this rule when selecting the number of bytes to read:
The number of bytes to read must be in multiple of endpoint maximum packet size.

```
Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512.
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...2
Enter Alternate Setting number: 3
Enter Endpoint No with Direction (in hex): 83
Enter Number of times (Enter 0 for Infinite loop): 100
Number of bytes to be read shall be in multiple of Endpoint's maximum packet size and <= 32 MB.
Enter Number of Bytes to Read: 1048576
# # # # # # # # # #
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
* * * * * * * * * *
Average Speed(IN Mbits/second) for iteration 100 = 62.495
Speed = 62.495 Mbits/second
EXIT Y:N
```

Figure 8.78. USB2 Interrupt IN Test Result

3. To perform the Interrupt OUT test, select option 3.
4. Enter the following information:
 - Alternate setting number: 3.
 - Endpoint number with direction (in hex): 3
 - Number of times (Enter 0 for Infinite loop).
 - Number of bytes to write (Valid range: 1 to 32 MB)

```
Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...3
Enter Alternate Setting number: 3
Enter Endpoint No with Direction (in hex) : 3
Enter Number of times (Enter 0 for Infinite loop): 100
Enter Number of Bytes to Write (Valid Range: 1 to 32 MB ): 1049576
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Avrage Speed(IN Mbits/second) for iteration 100 = 62.340
*
Speed = 62.331 Mbits/second
EXIT Y:N
```

Figure 8.79. USB2 Interrupt OUT Test

8.3.8.7. USB2 Isochronous IN/OUT Test

To perform the USB2 Isochronous IN/OUT test, follow these steps:

1. To perform the Isochronous IN test, select option 5.
2. Alternate setting number: 1.
3. Enter the number of times (range: 1 to 50).

Based on the number of times entered, the **Isochronous IN transfer size** and **maximum transfer size** are automatically calculated.

4. The Isochronous IN transfer is initiated.

```

Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endpoints: 0
Alt_set: 1      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512.
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endpoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...5
Enter Alternate Setting number: 1
Enter Number of times (Range: 1 to 50): 5
totalTransferSize: 102400
szBuffer: 102400
IsochInPacketCount: 20
isochPackets: 1200
IsochInTransferSize: 20480
Requested 20480 bytes in 20 packets per transfer.
Transfer 1 completed. Read 18432 bytes.

Requested 20480 bytes in 20 packets per transfer.
Transfer 2 completed. Read 20480 bytes.

Requested 20480 bytes in 20 packets per transfer.
Transfer 3 completed. Read 20480 bytes.

Requested 20480 bytes in 20 packets per transfer.
Transfer 4 completed. Read 20480 bytes.

Requested 20480 bytes in 20 packets per transfer.
Transfer 5 completed. Read 20480 bytes.

EXIT Y:N _

```

Figure 8.80. USB2 Isochronous IN Test Result

5. To perform the Isochronous OUT test, select option 4.
6. Enter the following information:
 - Alternate setting number: 1.
 - Number of times (Range: 1 to 50).

Based on the number of times entered, the **Isochronous IN transfer size** and **maximum transfer size** are automatically calculated.

- Number of bytes to be written (Valid range: 1 to 10240). This indicates the number of bytes that can be transmitted in single service interval.

7. The Isochronous OUT transfer is initiated.

```

Application Version: 1.7
bMaxPacketSize0: 40
Alt_set: 0      Number of Endoints: 0
Alt_set: 1      Number of Endoints: 2
Endpoint index: 0      Pipe type: Isochronous   Pipe ID: 81      MaxPktSz: 1024.
I_MaximumBytesPerInterval: 1024
Endpoint index: 1      Pipe type: Isochronous   Pipe ID: 1       MaxPktSz: 512.
O_MaximumBytesPerInterval: 512
Alt_set: 2      Number of Endoints: 2
Endpoint index: 0      Pipe type: Bulk          Pipe ID: 82      MaxPktSz: 512.
Endpoint index: 1      Pipe type: Bulk          Pipe ID: 2       MaxPktSz: 512.
Alt_set: 3      Number of Endoints: 2
Endpoint index: 0      Pipe type: Interrupt     Pipe ID: 83      MaxPktSz: 1024.
Endpoint index: 1      Pipe type: Interrupt     Pipe ID: 3       MaxPktSz: 1024.
1: Get Device Information
2: Read Data from Device (Bulk/Interrupt IN)
3: Write Data to Device (Bulk/Interrupt OUT)
4: Write Data to Device (Isochronous OUT)
5: Read Data from Device (Isochronous IN)
-----
Enter your Selection...4
Enter Alternate Setting number: 1
Enter Number of times (Range: 1 to 50): 10
IsochOutTransferSize: 10240
Total Transfer Size: 102400
Enter Number of Bytes to be written (Valid range : 1 to 10240) :10240
Write transfer 0 completed.
Write transfer 1 completed.
Write transfer 2 completed.
Write transfer 3 completed.
Write transfer 4 completed.
Write transfer 5 completed.
Write transfer 6 completed.
Write transfer 7 completed.
Write transfer 8 completed.
Write transfer 9 completed.

EXIT Y:N _

```

Figure 8.81. USB2 Isochronous OUT Test Result

Important note: For Isochronous IN and OUT transfers, the firmware follows the *on-demand transfer* flow. In the *on-demand transfer* flow, the firmware initiates requests after receiving notification that Isochronous IN/OUT transfers have been initiated by the host. The notification is received via the *Transfer not ready* event. Because of this, the first few (3 to 4) requests will be dropped. In the case of Isochronous IN, the USB23 IP automatically send 0 length data while for Isochronous OUT, data will be lost. This is an expected behavior.

8.4. Example Software API

8.4.1. USB Enumeration API

In the USB23 Benchmark example design, there are some useful software API that you can refer to. To obtain the software API, follow these steps:

1. Unzip the `<IP_INSTALLED_DIRECTORY>\software\usb23hal_6eps_v2_2.zip`.
2. Open `<IP_INSTALLED_DIRECTORY>\software\usb23hal_6eps_v2_2\doc\html\index.html`.

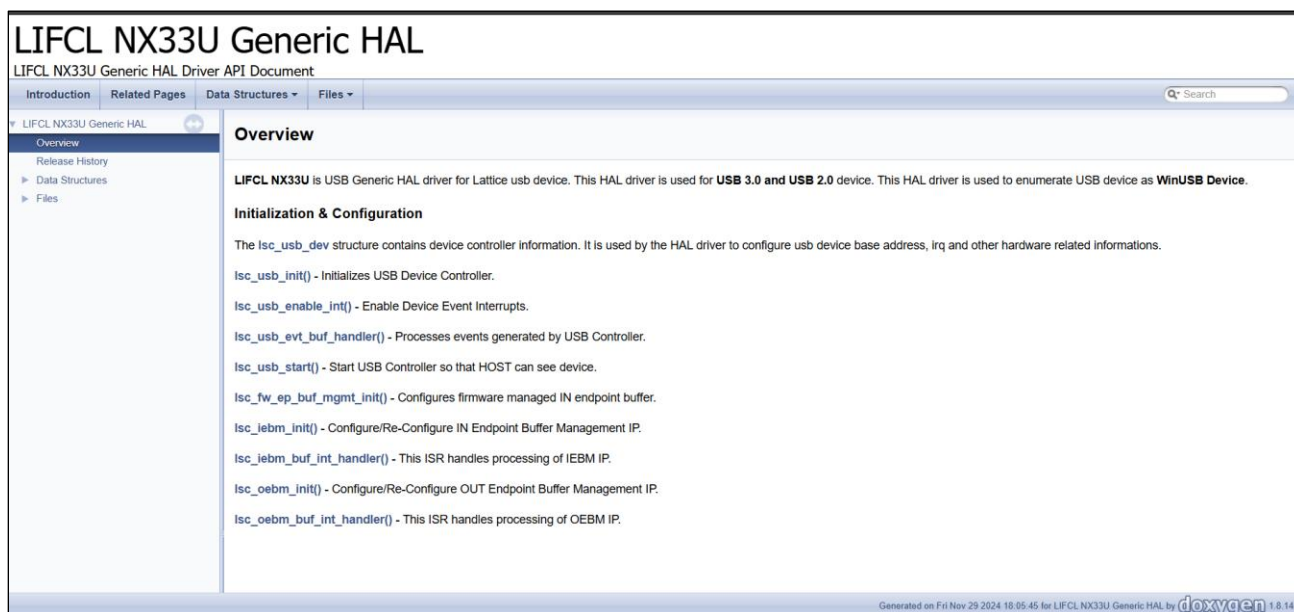


Figure 8.82. HAL Driver API Document

Appendix A. Resource Utilization

The following table shows the resource utilization of the USB 2.0/3.2 IP core using the Synplify Pro of the Lattice Radiant software.

Table A.1. Resource Utilization Using LIFCL-33U-7CTG104C

Configuration	USB23	Registers	LUTs	EBRs
USB23 IP	1	0	0	0
USB23 IP and USB23 example design	1	3,764	6,661	36

References

- [USB 2.0/3.2 IP Release Notes \(FPGA-RN-02037\)](#)
- [CrossLink-NX-33 and CrossLinkU-NX Data Sheet \(FPGA-DS-02104\)](#)
- [Lattice Radiant Timing Constraints Methodology Application Note \(FPGA-AN-02059\)](#)
- [A Step-by-Step Approach to Lattice Propel Application Note \(FPGA-AN-02052\)](#)
- [USB 2.0/3.2 IP Core](#) web page
- [CrossLink-NX FPGA](#) web page
- [Lattice Radiant Software FPGA](#) web page
- [Lattice Insights](#) for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.5, IP v3.2.0, October 2025

Section	Change Summary
All	Updated IP version on the cover page.
Abbreviations in This Document	Added an abbreviation definition for HAL.
Introduction	- Updated Table 1.1. Summary of the USB2.0/3.2 IP Core. - Updated the Features section.
Operation Details	- Removed the Get Device Descriptor without Setup section. - Updated the Connect Done section.
Designing with the IP	Updated the Generating and Instantiating the IP section.
Example Design	- Removed Note 1 in the Overview of the Example Design and Features section. - Updated the Description on Data Flow from USB Device to Host section. - Added Figure 8.6. PLL Optional Port Configuration. - Added the following sections: - Hardware Version GPIO section. - Remove Wakeup GPIO section. - USB2 PHY Reset GPIO section. - Endpoint IN Buffer Configuration section. - Partial Buffer Handling section. - Interrupt Generation section. - Added Figure 8.13. System Memory Port S0 Configuration. - Added Figure 8.14. System Memory Port S1 Configuration. - Updated the following sections: - Creating the Project section. - Building the Project section. - Trim the Memory File section. - Merging .mem with Bitstream File section. - FPGA Bitstream Programming section. - Obtaining Device Information section. - USB3 Bulk IN/OUT Test section. - USB3 Interrupt IN/OUT Test section. - USB3 ISOC IN/OUT Test section. - USB2 Bulk IN/OUT TEST section. - USB2 Interrupt IN/OUT Test section. - USB2 Isochronous IN/OUT Test section. - Removed the Known Issue in Streaming Operation section.
Appendix A. Resource Utilization	Updated Table A.1. Resource Utilization.

Revision 1.4, IP v3.1.1, June 2025

Section	Change Summary
All	Updated IP version on the cover page.
Introduction	- Updated the software version in Table 1.1. Summary of the USB2.0/3.2 IP Core. - Updated the Licensing and Ordering Information section. - Removed the Ordering Part Number section.
Running the Example Design	- Updated Figure 8.52. Memory Initialization. - Updated the Trim the Memory File section.
Appendix A. Resource Utilization	Updated the software version to <i>Lattice Radiant software 2025.1</i> .

Revision 1.3, IP v3.1.0, March 2025

Section	Change Summary
All	Added IP version on the cover page.
Introduction	Updated the IP version in Table 1.1. Summary of the USB2.0/3.2 IP Core.
Example Design	- Updated the notes in the Overview of the Example Design and Features section. - Updated the content and figures in the Example Design Components section. - Updated the content and the figures in the Running the Example Design section. - Updated the content and figure in the Example Software API section.
Resource Utilization	Updated Table A.1. Resource Utilization.
Appendix B. iob.c	Removed this section.

Revision 1.2, IP v3.0.0, December 2024

Section	Change Summary
All	Added IP version on the cover page.
Abbreviations in This Document	Updated the title of this section.
Introduction	- Updated the Quick Facts section. - Added the IP Support Summary section. - Updated the following key feature of the USB 2.0/3.2 IP core in the Features section: - Supports two control endpoints at EP0/1 (1 IN 1 OUT) and eight individually reconfigurable logical endpoints. Each logical endpoint reflect two physical endpoints, one IN and one OUT endpoint. For more information, refer to the CrossLink-NX-33 and CrossLinkU-NX Data Sheet (FPGA-DS-02104). - Renamed the section title IP Validation Summary to Hardware Support and updated the content in the Hardware Support section. - Updated the Licensing and Ordering Information section. - Added the Minimum Device Requirements section.
Functional Description	- Updated the IP Architecture Overview section. - Updated the Clocking Overview section. - Updated the AHBL-to-LMMI Converter section.
IP Parameter Description	Updated Table 3.1. General Attributes of the USB 2.0/3.2 IP Core.
Signal Description	- Updated the title for the LMMI/AHBL Interface – Configuration Path section. - Added the interrupt_o port to Table 4.6. AXI Interface - Data Path.
Register Description	- Updated the introductory paragraph in this section. - Updated Table 5.1. USB IP Controller Register Information. - Updated the title for all the sub-sections in the USB IP Controller Register Information section. - Updated Table 5.2. GLOBAL_CTL. - Added Table 5.3. GLOBAL_ID. - Updated the description for PHYSOFTIRST in Table 5.4. GLOBAL_USB2_PHY_CFG(#n) (for n = 0). - Updated the description for DATWIDTH in Table 5.5. GLOBAL_USB3_PIPE_CTL(#n) (for n = 0) section. - Added the GLOBAL_TX_FIFO_SIZE (GTXFIFOSIZ(#n)) section. - Added the GLOBAL_TX_FIFO_SIZE (GTXFIFOSIZ(#n)) section. - Added Table 5.20. GLOBAL_RX_FIFO_SIZE. - Updated Table 5.21. USB IP 2/3 PHY Control Status Register Information.
Operation Details	- Updated the Enumeration Sequence section. - Added the Transfer Flow section.
Designing with the IP	- Replaced the following figures: - Figure 7.2. IP Configuration - Figure 7.3. Check Generated Result

Section	Change Summary
	- Updated Table 7.1. Generated Files. - Updated the following sections: - Design Implementation - Timing Constraints - Updated the introductory paragraph of the Designing with the IP section. - Removed the <i>Running the Simulation</i> section.
Example Design	Replaced the entire Example Design section with new content.
Appendix A. Resource Utilization	Updated the resource utilization information.
Appendix B. iob.c	Added this section.
References	Added the following references: - USB 2.0/3.2 IP Release Notes (FPGA-RN-02037) - CrossLink-NX-33 and CrossLinkU-NX Data Sheet (FPGA-DS-02104) - Lattice Radiant Timing Constraints Methodology Application Note (FPGA-AN-02059) - A Step-by-Step Approach to Lattice Propel Application Note (FPGA-AN-02052) - USB 2.0.3.2 IP Core web page

Revision 1.1, April 2024

Section	Change Summary
All	Replaced all CrossLink-NX with CrossLinkU-NX.
Introduction	- Updated the Overview of the IP section. - Updated Table 1.1. Summary of the USB2.0/3.2 IP Core. - Updated the Features section.
Functional Description	- Updated the IP Architecture Overview section. - Updated the Clocking Overview section - Updated the User Interfaces section. - Updated the Internal IP Blocks section. - Replaced USB23_reference_clk with usb23_reference_clk in the User Interfaces section.
IP Parameter Description	Updated Table 3.1. General Attributes of the USB 2.0/3.2 IP Core.
Signal Description	Updated the following tables in this section: - Table 4.1. Clock Ports - Table 4.2. Reset Interface - Table 4.3. USB Interface - Table 4.4. LMMI Interface - Configuration Path - Table 4.6. AXI Interface - Data Path. - Table 4.7 Other Signals.
Register Description	Updated the tables in these sections: - USB IP Controller Register Information - USB IP 2.0 PHY Control Status Register Information
Operation Details	- Updated Figure 6.2. Flow Diagram for the Control OUT Transfer. - Replaced all USB23_Device IP with USB 2.0/3.2 IP. - Replaced USB 2.0/3.2 in step 2 to USB23. - Updated the following figures: - Figure 7.1. Module/IP Block Wizard - Figure 7.2. IP Configuration - Figure 7.3. Check Generated Result - Updated Table 7.1. Generated Files. - Updated a reference to the Lattice Radiant Software 2023.2 User Guide. - Replaced all USB23_Device IP with USB 2.0/3.2 IP.
Designing with the IP	Replaced Figure 7.4 Testbench Directory Structure.

Section	Change Summary
	- Added the Running the Simulation section. - Added the Using RISC-V Drivers section. - Added a reference to the Lattice Radiant Timing Constraints Methodology Application Note (FPGA-AN-02059) in the Timing Constraints section.
Example Design	In the Example Design Components section: - Updated Figure 8.1. USB2.0/3.2 IP Example Design Block Diagram. - Added Figure 8.2. Clocking Diagram of the Example Design. - Updated USB 2.0/3.2 Primitive section. - Updated USB23_ED section. - Added Table 8.1. Configurable Attributes of the USB23_ED. - Updated the End Address default value for the Interrupt Handler in Table 8.3. MUX for CSR, Interrupt Handler, and Pattern Generator/Checker. - Added Figure 8.3. USB23_ED Block Diagram. - Added Running the Example Design section. - Added Loading an Example Design from the Propel Builder Tool section. - Added USB Example Design with RISC-V Driver section.
Usage	Added the following statement in this section: <i>All waveform diagrams have the following notation: xm_* are signals to/from the AXI manager interface and xs_* are signals to/from the AXI subordinate interface.</i>
Appendix A. Resource Utilization	Updated Table A.1. Resource Utilization.

Revision 1.0, October 2023

Section	Change Summary
All	Initial release.



www.latticesemi.com