CoreSRIO v2.0 User Guide

Introduction

CoreSRIO is the Serial Rapid IO (SRIO) endpoint solution comprises a Serial Rapid IO Physical Layer, Logical Layer, and Transport Layer. The SRIO endpoint uses AXI4-Stream interfaces for high-throughput data transfer and AXI4-Lite interfaces for the register configuration interface.

The AXI4 Lite, AXI4 Stream, and AXI4 protocol standard uses the terminology **Master** and **Slave**. The equivalent Microchip terminology used in this document is **Initiator** and **Target** respectively.

Summary

Core Version	This document applies to CoreSRIO v2.0.
Supported Device Families	• PolarFire® SoC • PolarFire • RT PolarFire • RTG4™ • IGLOO® 2 • SmartFusion® 2
Supported Tool Flow	Requires Libero® SoC v12.0 or later releases.
Supported Interfaces	• **AXI4 Lite:** Dedicated to access configuration, control, and status registers inside the core. • **AXI4 Stream:** Dedicated for the core to receive stream data from an external AXI4-Stream initiator interface. Similarly, AXI4-Stream initiator interfaces are dedicated for the core to transmit data to an external AXI4-Stream target interface.
Licensing	CoreSRIO is available in the following license versions: Evaluation, Obfuscated, and RTL. • Evaluation version supports obfuscated and encrypted RTL code with self-destruct feature. Evaluation version is available for free and supports four hours of the functionality on silicon. • Obfuscated version is available with obfuscated and encrypted RTL source, license locked, and supports unlimited functionality on silicon. You must purchase this license separately. • RTL version is available with clear RTL source code and supports unlimited functionality on silicon. You must purchase this license separately.

Features

CoreSRIO has the following key features:

Layer Name	Description
Logical Layer	• Provides AXI4-Lite interface to access register space (capability registers, command and status registers, and extended feature registers). • Provides dedicated host side AXI4-Stream interface ports for Rapid IO Input Output packets, Maintenance packets • Supports optional endian conversion for all the host side interface • Supports 34-bit address • Supports 8-bit, 16-bit, and 32-bit Device ID • Supports the following Rapid IO packets: – NREAD: Read request packets – NWRITE: Write-without-Response request packets – NWRITE_R: Write-with-Response request packets – SWRITE: Streaming-Write request packets – ATOMIC: Set, Clear, Increment, Decrement, Swap, Test-and-Swap, Compare-and-Swap request packets – MAINTENANCE: Read request, write request, Read response, Write response packets – RESPONSE: Response with data payload, Response without data payload, Message response packets
Physical Layer	• Baud Rate Class 1 • 1x lane operation • Idle Sequence: IDLE1 • Control Symbol: control symbol 24 • CRC: CRC-5 and CRC-16 • Single Virtual Channel supported • 8B/10B encoding and decoding • Configurable 10-bit and 20-bit PMA/PCS interface based on data rates
Buffer Layer	• A round robin outgoing scheduler • Receiver-Controlled Flow Control • Transmit and Receive buffer depths of 32 packets
Features Not Supported in this Release	• Doorbell and Data Message operations • Transmitter-Controlled Flow Control • Baud rate discovery • Priority, Critical request flow • Multiple Virtual Channel • Multiple Port configuration • Error Recovery • Canceling packets

Installation Instructions

CoreSRIO must be installed to the IP Catalog of Libero SoC software automatically through the IP Catalog update function in Libero SoC software, or it can be manually downloaded from the catalog. Once the IP core is installed in Libero SoC software IP Catalog, it can be configured, generated, and instantiated within SmartDesign for inclusion in the Libero project.

Device Utilization and Performance

The following table lists CoreSRIO macro that is implemented in the families.

Table 1. CoreSRIO Utilization

Device Details		Resources			Performance (MHz)	RAMs	
Family	Device	LUTs	DFF	Logic Elements		LSRAM	µSRAM
PolarFire® SoC	MPFS250T	6105	4965	11070	SYS_CLK - 100 ACLK - 100 LANE0_TX_CLK - 250 LANE0_RX_CLK - 250	14	8
PolarFire	MPF300T	6105	4965	11070	SYS_CLK - 100 ACLK - 100 LANE0_TX_CLK - 250 LANE0_RX_CLK - 250	14	8
SmartFusion® 2	M2S150TS	6243	5429	11852	SYS_CLK -100 ACLK - 100 LANE0_TX_CLK - 125 LANE0_RX_CLK - 125	16	2
IGLOO® 2	M2GL150TS	6243	5429	11852	SYS_CLK - 100 ACLK - 100 LANE0_TX_CLK - 125 LANE0_RX_CLK - 125	16	2
RTG4™	RT4G150	6267	5256	11523	SYS_CLK - 100 ACLK - 100 LANE0_TX_CLK - 125 LANE0_RX_CLK - 125	16	2

Important: The data in the table is achieved using Libero SoC v2022.1.

1. The data in table is achieved using typical synthesis and layout settings.
2. Clock constraints used to achieve the performance numbers are:
   - SYS_CLK: 100
   - ACLK: 100
   - LANE0_TX_CLK: 250
   - LANE0_RX_CLK: 250
3. For RTG4, family speed grade used is −1.

Table of Contents

Introduction....1
Summary....1
Features....1
Installation Instructions....2
Device Utilization and Performance....3

1. CoreSRIO Overview....6

2. Functional Description....7
2.1. Register Block....7
2.2. Logical Layer....7
2.3. Buffer Layer....7
2.3.1. Transmit Buffer....7
2.3.1.1. Receiver Controlled Flow Control....7
2.3.2. Receive Buffer....7
2.4. Physical Layer....7
2.4.1. PMA Interface Block....8
2.4.1.1. Word Aligner....8
2.4.1.2. 10b/8b Decoder....8
2.4.1.3. Elastic Buffer....8
2.4.1.4. 8b/10b Encoder....8
2.4.2. TX/RX Packet/Control Symbol Block....8
2.4.2.1. IDLE1/Clock Compensation Sequence Generation....8
2.4.2.2. RX Symbol Decoder....8
2.4.2.3. TX Symbol Encoder....8
2.4.2.4. Flow Control Block....8
2.4.2.5. CRC (Gen and Check Block)....8
2.4.2.6. RX Packet or Control Symbol De-assembly and Idle Sequence Deletion....9
2.4.2.7. TX Packet or Control Symbol Assembling and Idle Sequence Insertion....9
2.5. Logical Layer AXI4-Stream Serial RapidIO Interface....9

3. IP Core Parameters and Interface Signals....10
3.1. CoreSRIO Configurator Settings....10
3.2. Input and Output Signals....14

4. Timing Diagrams....20
4.1. AXI4 Lite Target Write Transaction....20
4.2. AXI4 Lite Target Read Transaction....20
4.3. AXI4 Stream Interface....21

5. Implementation of IP Core in Libero Design Suite....22
5.1. SmartDesign....22
5.1.1. Configuring the CoreSRIO....22
5.1.2. Synthesis in Libero SoC....23
5.1.3. Place and Route in Libero SoC....24
5.1.4. System Integration....24

6. Register Map and Descriptions....25
6.1. SRC_OPR_CAR....28
6.2. PE_LL_CTRL_CSR....29
6.3. PE_FEATURES_CAR....30
6.4. LP_SRB_LANE_HDR....31
6.5. LP_SRB_HDR....32
6.6. LP_PORT_GEN_CTRL....33
6.7. LP_PORT_0_ERR_STAT....34
6.8. LP_PORT_0_CTRL2....36
6.9. LP_PORT_0_CTRL....38
6.10. HOST_BDID_LOCK_CSR....39
6.11. LP_LN0_STATUS_0....40
6.12. LCSBA1_CSR....42
6.13. DEV_INFO_CAR....43
6.14. DEV32_BDID_CSR....44
6.15. DEV_ID_CAR....45
6.16. DEST_OPR_CAR....46
6.17. COMP_TAG_CSR....47
6.18. BDID_CSR....48
6.19. ASSY_INFO_CAR....49
6.20. ASSY_ID_CAR....50
7. Testing of CoreSRIO....51
8. Additional References....52
8.1. Known Issues and Workarounds....52
8.2. Discontinued Features and Devices....52
8.3. Ordering Codes....52
9. Glossary....53
10. Revision History....54
Microchip FPGA Support....55
Microchip Information....55
The Microchip Website....55
Product Change Notification Service....55
Customer Support....55
Microchip Devices Code Protection Feature....55
Legal Notice....56
Trademarks....56
Quality Management System....57
Worldwide Sales and Service....58

1. CoreSRIO Overview

CoreSRIO is a packet-switched interconnect intended primarily as an intra-system interface for chip-to-chip and board-to-board communications at Gigabyte-per-second performance levels.

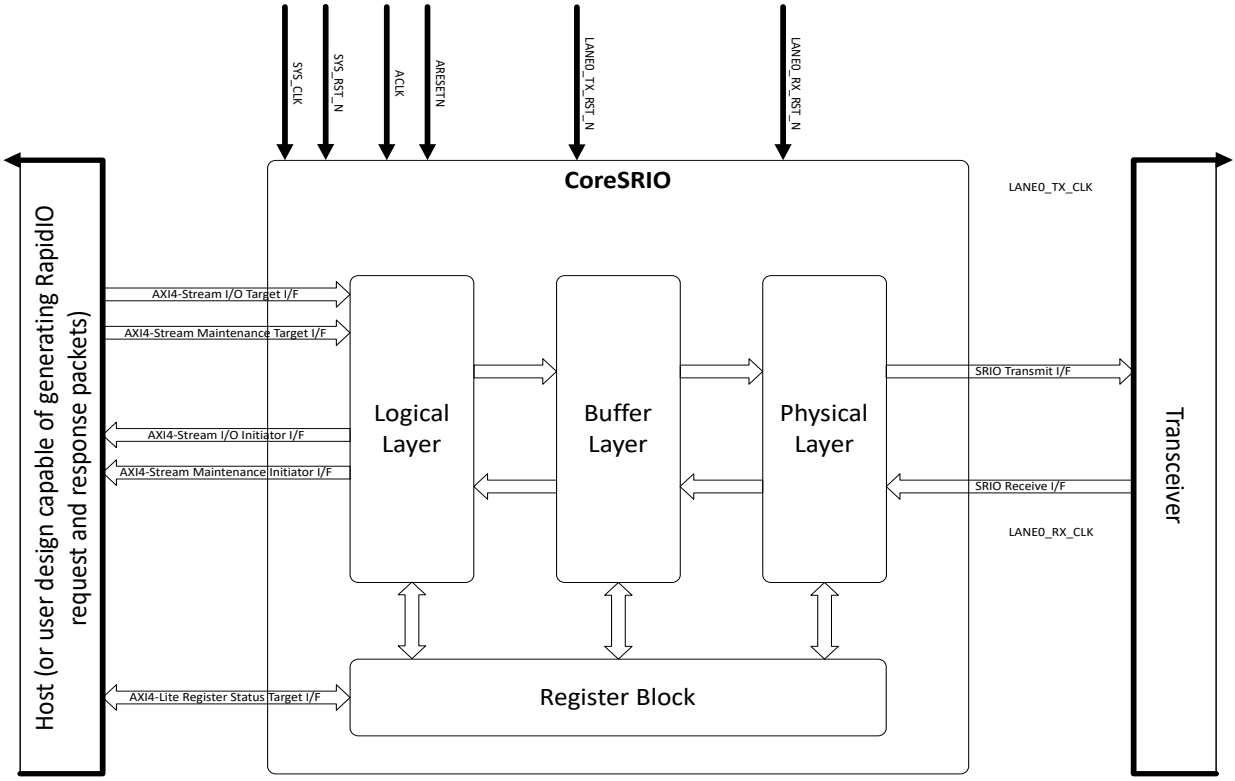
This version of the CoreSRIO is compliant to the following RapidIO v2.0 Specifications:

- Compliant with RapidIO v2.0 Part 1. Input-Output Logical Specification
- Compliant with RapidIO v2.0 Part 3. Common Transport Specification
- Compliant with RapidIO v2.0 Part 6. LP-Serial Physical Specification
- Compliant with AXI4-Lite and AXI4-Stream Specifications

The I/O operation protocols work using request/response transaction pairs through the interconnect fabric. A processing element sends a request transaction to another processing element if it requires an activity to be carried out. The receiving processing element responds with a response transaction when the request has been completed or if an error condition is encountered. Each transaction is sent as a packet through the interconnect fabric.

CoreSRIO supports all input and output operations as mentioned in the specifications to carry different Read, Write and Read-Modify-Write operations and Maintenance operations to support system exploration, initialization, and maintenance.

Figure 1-1. CoreSRIO System-Level Block Diagram



2. Functional Description

This section describes the functional description of CoreSRIO.

2.1 Register Block

Register block contains all the RapidIO CARs, CSRs, and other control and status registers implemented in the core. All these registers are accessed by a host through an AXI4-Lite interface.

2.2 Logical Layer

The Logical Layer block accepts the RapidIO request and responds to packet information from the host, which are required to be transmitted on the link, concatenates the received packet information according to the required packet format, and provides them to the Buffer Layer.

The Logical Layer block accepts the RapidIO request and responds to packets from the buffer, which are received from the link, parses the packets based on the packet type and provides them to the host.

2.3 Buffer Layer

Buffer Layer has two major blocks that are: Transmit Buffer and Receive Buffer modules.

2.3.1 Transmit Buffer

Transmit Buffer module major functionalities are:

- Scheduling the Logic Layer packets
- Storing of packets received from the Logic Layer
- Pass to Physical Layer
- Keep in the buffer, if the packet_accepted is not received from Link Partner for the same packet and flow control

A transmit asynchronous buffer is implemented to hold the input and output and maintenance packets. A round robin outgoing scheduler implementation chooses packets from the input and output and maintenance Logical Layer interfaces for transmiting to buffer. Transmit buffer works in Store-and-forward mode.

The buffer must hold each packet until it has been successfully received by the Link Partner device and acknowledged. Once the packet acknowledgment is received the corresponding packet is discarded from the buffers. This block supports receiver-controlled flow control.

2.3.1.1 Receiver Controlled Flow Control

When the buffer using receiver-controlled flow control, the buffer module depends on PHY retry protocol to control the flow to link partner.

The transmit buffer provides CDC between the Physical Layer PHY_CLK clock domain and the Logical Layer SYS_CLK clock domain.

2.3.2 Receive Buffer

The receive buffer stores the packets coming from the Physical Layer and passes to the Logical Layer in first-in-first-out sequence. The receive buffer provides clock domain crossing between the Physical Layer PHY_CLK clock and the Logical Layer SYS_CLK clock domain.

2.4 Physical Layer

The Physical Layer comprises two major blocks:

- PMA Interface Block
- TX/RX Packet/Control Symbol Block

2.4.1 PMA Interface Block

This block is used to compensate clock offset between the recovered clock received from PolarFire or PolarFire SoC transceiver and PHY_CLK, this block also does PCS functionality of encoding and decoding the data symbols.

2.4.1.1 Word Aligner

Performs detection for the word boundaries. Receives unaligned or aligned data from the PMA, detects the code group boundary with K28.5 and outputs the aligned data and aligned status signal.

2.4.1.2 10b/8b Decoder

Performs 8B10B decoding on properly aligned 10-bit code groups to convert them to the 8-bit characters. Outputs 8-bit decoded data along with the decode and disparity error status.

2.4.1.3 Elastic Buffer

This block compensates up to a + or - 200 ppm difference between the input bit rate and output bit rate. Both rates have a + or -100 ppm tolerance. Receivers receive clock compensation sequence on each of its active input lanes at least once every 4096 characters are received.

2.4.1.4 8b/10b Encoder

Performs 8B10B encoding on the 8-bit data to convert it to 10-bit code groups. Outputs 10 bits encoded data along with invalid K status signal.

2.4.2 TX/RX Packet/Control Symbol Block

This block transmits and receives the data to a PMA interface block and performs the following tasks.

- Port initialization
- Packet and symbol assembling and idle character insertion
- The packet or symbol delineation and idle character extraction
- ACKID insertion and checking for flow control
- CRC encoding and checking on packet and control symbol data

2.4.2.1 IDLE1/Clock Compensation Sequence Generation

During initialization and packet transmission when there is no packet and control symbol available to transmit, this block generates idle sequence according to the control symbol supported and sends to the PMA interface block. This block also generates a clock compensation sequence to send in lane once in every 4096 transmit code group.

2.4.2.2 RX Symbol Decoder

This block decodes the short control symbols received and sends the control signals to a different block for appropriate action. For example, Start Of Packet indication to packet decoder block. ACKID with packet_accepted for successful transmission or packet_retry for retransmission.

2.4.2.3 TX Symbol Encoder

This block receives command from the flow control block and encodes the control symbols for transmission.

2.4.2.4 Flow Control Block

Keeps track of the ACKID and flow control status from receive buffer block and generates appropriate control signals to the TX_symbol generators block to send control symbol, if required. Also, sends the control signals to buffer module for retransmission or packet freeing from queue.

2.4.2.5 CRC (Gen and Check Block)

The CRC generation block generates a CRC over the entire packet header and data payload, except for the first 6 bits of the first packet, which are covered by protocol.

2.4.2.5.1 CRC-16

The ITU polynomial `X16+X12+X5+1` is used to generate the 16-bit CRC for packets.

2.4.2.5.2 CRC-5

The ITU polynomial `x5+x4+x2+1` is used to generate the 5-bit CRC for Control Symbol 24.

2.4.2.6 RX Packet or Control Symbol De-assembly and Idle Sequence Deletion

This block disassembly the input data into two data streams. One goes into the packet buffer, and the other goes into the control symbol decoder. This block also extracts idle sequence from the data stream.

2.4.2.7 TX Packet or Control Symbol Assembling and Idle Sequence Insertion

The packet or symbol assembling, and idle character insertion block assembles packet data and control symbol into a proper output format.

2.5 Logical Layer AXI4-Stream Serial RapidIO Interface

The host interface ports are built from AXI4-Stream interfaces. The AXI4-Stream protocol uses a ready/valid handshake to transfer information from a initiator to a target component. It uses a last beat indicator to designate packet boundaries. Finally, there are data and user signals which carry the actual packet information.

On the target interfaces, the host provides the header information in the first beat of the data transfer. If data payload is available, then host provides it in the following beats.

On the initiator interfaces, the core provides the header information on the first beat of the data transfer. If data payload is available, provide it in the following beats.

The header format for the supported RapidIO packet types are provided on AXI4-Stream data (TDATA) and user (TUSER) signals as shown in the following figures.

Table 2-1. Header Format When Endian Conversion is Disabled (EN_LITTLE2BIG_CNV = 0)

AXI4S_IO/ MT_TARG_TDATA	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
Type 2 or 5 (10)	{prio, tt, ftype}	{transaction, rdsize/wrsize}	srcTID	address[29:22]	address[21:14]	address[13:6]	{address[5:0],wdptr,xambos}	reserved
Type 6 (10)	{prio, tt, ftype}	reserved		address[29:22]	address[21:14]	address[13:6]	{address[5:0],wdptr,xambos}	reserved
Type 13 (RESP)	{prio, tt, ftype}	{transaction, status}	target_info/ target TID	reserved				
Type 8 (MT)	{prio, tt, ftype}	{transaction, rdsize/wrsize/ status}	srcTID/target TID	config_offset[20:13]/ reserved	config_offset[12:5]/ reserved	{config_offset[4:0], wdptr, 2'b00}/reserved	reserved	

AXI4S_IO/ MT_TARG_TUSER	104:96	95:88	87:80	79:72	71:64	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
Type (2, 5, 6, or 13) IO	{Reserved, packet CRC error}	Extended address[31:24]/ 8'b0	Extended address[23:16]/ 8'b0	Extended address[15:8]/ 8'b0	Extended address[7:0]/ 8'b0	Destination ID[31:24]/ 8'b0	Destination ID[23:16]/ 8'b0	Destination ID[15:8]/ 8'b0	Destination ID[7:0]	Source ID[31:24]/ 8'b0	Source ID[23:16]/ 8'b0	Source ID[15:8]/ 8'b0	Source ID[7:0]
Type (8) Maintenance				{Reserved, packet CRC error}	Hope Limit[7:0]	Destination ID[31:24]/ 8'b0	Destination ID[23:16]/ 8'b0	Destination ID[15:8]/ 8'b0	Destination ID[7:0]	Source ID[31:24]/ 8'b0	Source ID[23:16]/ 8'b0	Source ID[15:8]/ 8'b0	Source ID[7:0]

Table 2-2. Header Format When Endian Conversion is Enabled (EN_LITTLE2BIG_CNV = 1)

AXI4S_IO/ MT_INIT_TDATA	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
Type 2 or 5 (10)	reserved	{address[5:0],wdptr,xambcs}	address[13:6]	address[21:14]	address[29:22]	srcTID	{transaction,rdsize/ wrsize}	{prio,tt,ftype}
Type 6 (10)	reserved	{address[5:0],wdptr,xambcs}	address[13:6]	address[21:14]	address[29:22]	reserved		{prio,tt,ftype}
Type 13 (RESP)	reserved					target_info/ target TID	{transaction, status}	{prio,tt,ftype}
Type 8 (MT)	reserved		{config_offset[4:0], wdptr, 2'b00}/reserved	config_offset[12:5]/ reserved	config_offset[20:13]/ reserved	srcTID/target TID	{transaction, rdsize/ wrsize/status}	{prio,tt,ftype}

AXI4S_IO/ MT_INIT_TUSER	104:96	95:88	87:80	79:72	71:64	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
Type (2, 5, 6, or 13) IO	Source ID[31:24]/ 8'b0	Source ID[23:16]/ 8'b0	Source ID[15:8]/ 8'b0	Source ID[7:0]	Destination ID[31:24]/ 8'b0	Destination ID[23:16]/ 8'b0	Destination ID[15:8]/ 8'b0	Destination ID[7:0]	address[31:24]/ 8'b0	address[23:16]/ 8'b0	address[15:8]/ 8'b0	address[7:0]/ 8'b0	packet CRC
Type (8) Maintenance				Source ID[31:24]/ 8'b0	Source ID[23:16]/ 8'b0	Source ID[15:8]/ 8'b0	Source ID[7:0]	Destination ID[31:24]/ 8'b0	Destination ID[23:16]/8'b0	Destination ID[15:8]/8'b0	Destination ID[7:0]	Hop Limit[7:0]	{Reservd, packetCRC error}

3. IP Core Parameters and Interface Signals

This section discusses the parameters in the CoreSRIO Configurator settings and I/O signals.

3.1 CoreSRIO Configurator Settings

The following table lists the CoreSRIO for configuring the core.

Important: The Name column in the following table shows the actual parameter names used in RTL. The Description column starts with the parameter names as they appear in the CoreSRIO Configurator.

These parameters are user configurable and is available in the configuration window.

Table 3-1. Configurable Parameters

Name	Valid Values	Default Value	Description
RESET_TYPE	Asynchronous Synchronous	Asynchronous	**Reset Type** This Parameter is used to select reset type.
Assembly Identity Parameters			
DEV_ID	16'h0000–16'hFFFF	16'h0000	**Device Identifier** Uniquely identifies the type of device from the vendor.
DEV_VEN_ID	16'h0000–16'hFFFF	16'h0000	**Device Vendor Identifier** Identifies the vendor that manufactured the device containing the processing element.
Device Information Parameters			
DEV_REV	32'h00000000– 32'hFFFFFFFF	32'h00000000	**Device Revision Level** Identifies the revision level of the device.
Assembly Identity Parameters			
ASSY_ID	16'h0000–16'hFFFF	16'h0000	**Assembly Identifier** Uniquely identify the type of assembly from the vendor.
ASSY_VEN_ID	16'h0000–16'hFFFF	16'h0000	**Assembly Vendor Identifier** Identifies the vendor that manufactured the assembly or subsystem containing the device.
Assembly Information Parameters			
ASSY_REV	16'h0000–16'hFFFF	16'h0000	**Assembly Revision Level** Identifies the revision level of the assembly.
EXTD_FEATURES_PTR	16'h0100–16'hFFFC	16'h0100	**Extended Features Pointer** Pointer to first entry in Extended Features List.
Processing Element (PE) Feature Parameters			
BRIDGE	0-1	1	**Bridge** Indicator for PE capability of bridge to another interface.

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
MEMORY	0-1	1	**Memory** Indicator for PE physically addressable local address space, that can be accessed as an end point through non-maintenance (that is, non-coherent read and write) operations. This local address space may be limited to local configuration registers, or could be on-chip SRAM, and so on.
PROCESSOR	0-1	1	**Processor** Indicator for PE physical local processor or similar device that executes code. A device that bridges to an interface that connects to a processor does not count (see bit 31 above).
EXTD_FEATURES	0–1	1	**Extended Feature** Indicator for the PE extended features. If set to 1, then the extended features pointer is valid.
DEV_ID_WIDTH	8 bit 16 bit 32 bit	8 bit	**Device ID Width** Device ID width for PE.
EXTD_ADDR_SPRT	34 bit	34 bit	**Extended Address Support** Indicates the number of address bits supported by the PE both as a source and target of an operation.
Source Operation Parameters			
SRC_NREAD	0-1	1	**Source NREAD** Enable source read operation.
SRC_NWRITE	0-1	1	**Source NWRITE** Enable source write operation.
SRC_SWRITE	0-1	1	**Source SWRITE** Enable source streaming-write operation.
SRC_NWRITE_R	0-1	1	**Source NWRITE_R** Enable source write-with-response operation.
SRC_ACSWP	0-1	1	**Source Atomic Comp-Swap** Enable source atomic compare-and-swap operation.
SRC_ATSWP	0-1	1	**Source Atomic Test-Swap** Enable source atomic test-and-swap operation.
SRC_AINC	0-1	1	**Source Atomic Increment** Enable source atomic increment operation.
SRC_ADEC	0-1	1	**Source Atomic Decrement** Enable source atomic decrement operation.
SRC_ASET	0-1	1	**Source Atomic Set** Enable source atomic set operation.

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
SRC_ACLR	0-1	1	**Source Atomic Clear** Enable source atomic clear operation.
SRC_ASWP	0-1	1	**Source Atomic Swap** Enable source atomic swap operation.
Destination Operations Parameters			
DEST_NREAD	0-1	1	**Destination NREAD** Enable destination read operation.
DEST_NWRITE	0-1	1	**Destination NWRITE** Enable destination write operation.
DEST_SWRITE	0-1	1	**Destination SWRITE** Enable destination streaming-write operation.
DEST_NWRITE_R	0-1	1	**Destination NWRITE_R** Enable destination write-with-response operation.
DEST_ACSWP	0-1	1	**Destination Atomic Comp-Swap** Enable destination atomic compare-and-swap operation.
DEST_ATSWP	0-1	1	**Destination Atomic Test-Swap** Enable destination atomic test-and-swap operation.
DEST_AINC	0-1	1	**Destination Atomic Increment** Enable destination atomic increment operation.
DEST_ADEC	0-1	1	**Destination Atomic Decrement** Enable destination atomic decrement operation.
DEST_ASET	0-1	1	**Destination Atomic Set** Enable destination atomic set operation.
DEST_ACLR	0-1	1	**Destination Atomic Clear** Enable destination atomic clear operation.
DEST_ASWP	0-1	1	**Destination Atomic Swap** Enable destination atomic swap operation.
Local Configuration Space Base Address Parameters			
LCSBA	32'h00000000–32'hFFFFFFFF	32'h00000000	**Default Base Address** Default value for bits [33:3] of a 34-bit local physical address. (Valid range for 34-bit address mode is 32'h00000000–32'h7FFFFFFF).
Base Device ID Parameters			

.....continued

Name	Valid Values	Default Value	Description
DEV8_BASE_DEV_ID	8'h00–8'hFF	8'h00	**8-bit Mode Device ID** Device ID value when Device ID Width is selected for 8-bit mode. Available only when DEV_ID_WIDTH = 8 bit
DEV16_BASE_DEV_ID	16'h0000– 16'hFFFF	16'h0000	**16-bit Mode Device ID** Device ID value when Device ID Width is selected for 16-bit mode. Available only when DEV_ID_WIDTH = 16 bit
DEV32_BASE_DEV_ID	32'h00000000–32'hFFFFFFFF	32'h00000000	**32-bit Mode Device ID** Device ID value when Device ID Width is selected for 32-bit mode. Available only when DEV_ID_WIDTH = 32 bit
General Parameters			
EN_LITTLE2BIG_CNV	0-1	1	**Enable Little to Big Endian Conversion** 0 – Disable Endian Conversion. This option must be selected when data input on the host side interface is in big-endian format. 1 – Enable Endian Conversion. This option must be selected when data input on the host side interface is in little-endian format.
NUM_STAGE	2-5	2	**Number of Synchronizer Stages** Number of synchronizer stages can be configured for the signals crossing the clock domain.
Physical Layer Configuration			
BAUD_RATE	1.25 Gbps 2.5 Gbps 3.125 Gbps 5.0 Gbps	2.5 Gbps	**Baud Rate (SERDES line rate)**
HOST	0-1	1	**Host** 0 – agent or target device 1 – host device
MASTER_EN	0-1	1	**Initiater Enable** 0 – Initiater disabled 1 – Initiater Enabled
DISCOVERED	0-1	0	**Discovered** 0 – PE not discovered previously 1 – PE discovered previously
Transceiver Type Configuration			

.....continued			
Name	**Valid Values**	**Default Value**	**Description**
LN0_TX_LANE_TYPE	Short run	Short Run	**LN0_Transmitter_type** Indicates the lane type 0 - short run
LN0_RX_LANE_TYPE	Short run	Short Run	**LN0_Receiver_type** Indicates the lane type: 0 - short run

These are generated parameters from Configurable parameters and are not available to the user.

Table 3-2. Generate Parameters

Parameter	Range	Default Value	Description
EPCS_DATA_WIDTH	10 – 20	20	10 bits PMA-PCS interface, when BAUD_RATE = 1.25 Gbps 20 bits PMA-PCS interface, when BAUD_RATE > 1.25 Gbps

3.2 Input and Output Signals

The following table lists the clock ports for CoreSRIO.

Table 3-3. Clocks and Reset

Port Name	Width	Direction	Description
Clock and Reset for AXI4-Lite interface			
ACLK	1	Input	AXI4-Lite Clock
ARESETN	1	Input	AXI4-Lite reset Active low, asynchronous reset input. This reset can be asserted asynchronously, but the de-assertion is synchronous to ACLK clock. This reset is used as asynchronous reset in non-RTG4 device families. This reset is used as synchronous reset in RTG4™ device family. **Note:** Reset synchronizer must be implemented external to the IP Core.
Clock and Reset for all Logical Layer Interfaces			
SYS_CLK	1	Input	System Clock/Logical Layer Clock This clock is used for all AXI4 Stream interface.
SYS_RST_N	1	Input	System Reset/Logical Layer Reset Active low, asynchronous reset input. This reset can be asserted asynchronously, but the de-assertion is synchronous to SYS_CLK clock. This reset is used as asynchronous reset in non-RTG4 device families. This reset is used as synchronous reset in RTG4 device family. **Note:** Reset synchronizer must be implemented external to the IP Core.
Clock and Reset for SRIO Transmit Interface			

.....continued			
Port Name	**Width**	**Direction**	**Description**
LANE0_TX_CLK	1	Input	Lane 0 Transmit Clock **Note**: PHY_CLK is same as this LANE0_TX_CLK
LANE0_TX_RST_N	1	Input	Lane 0 Transmit Reset Active low, asynchronous reset input. This reset can be asserted asynchronously, but the de-assertion is synchronous to LANE0_TX_CLK clock. This reset is used as asynchronous reset in non-RTG4 device families. This reset is used as synchronous reset in RTG4 device family. **Note:** Reset synchronizer must be implemented external to the IP Core.
Clock and Reset for SRIO Receive Interface			
LANE0_RX_CLK	1	Input	Lane 0 Receive Clock
LANE0_RX_RST_N	1	Input	Lane 0 Receive Reset Active low, asynchronous reset input. This reset can be asserted asynchronously, but the de-assertion is synchronous to LANE0_RX_CLK clock. This reset is used as asynchronous reset in non-RTG4 device families. This reset is used as synchronous reset in RTG4 device family. **Note:** Reset synchronizer must be implemented external to the IP Core.

The following table provides all input and output ports of CoreSRIO. The Clock Domain column in the following table indicates the input ports and the output ports expected to be synchronized in the respective clock domain.

Table 3-4. AXI4-Lite Register Target Interface

Port	Width	Direction	Clock Domain	Description
Write Address Channel Ports				
AXI4L_CS_TARG_AWVALID	1	Input	ACLK	AXI4-Lite Write Address Valid
AXI4L_CS_TARG_AWREADY	1	Output	ACLK	AXI4-Lite Write Address Ready
AXI4L_CS_TARG_AWADDR	24	Input	ACLK	AXI4-Lite Write Address
Write Data Channel Ports				
AXI4L_CS_TARG_WVALID	1	Input	ACLK	AXI4-Lite Write Data Valid
AXI4L_CS_TARG_WREADY	1	Output	ACLK	AXI4-Lite Write Data Ready
AXI4L_CS_TARG_WDATA	32	Input	ACLK	AXI4-Lite Write Data
AXI4L_CS_TARG_WSTRB	4	Input	ACLK	AXI4-Lite Write Data Strobe. This port is unused. All write data bytes are considered valid.
Write Response Channel Ports				
AXI4L_CS_TARG_BVALID	1	Output	ACLK	AXI4-Lite Write Response Valid
AXI4L_CS_TARG_BREADY	1	Input	ACLK	AXI4-Lite Write Response Ready
AXI4L_CS_TARG_BRESP	2	Output	ACLK	AXI4-Lite Write Response
Read Address Channel Ports				

.....continued				
Port	**Width**	**Direction**	**Clock Domain**	**Description**
AXI4L_CS_TARG_ARVALID	1	Input	ACLK	AXI4-Lite Read Address Valid
AXI4L_CS_TARG_ARREADY	1	Output	ACLK	AXI4-Lite Read Address Ready
AXI4L_CS_TARG_ARADDR	24	Input	ACLK	AXI4-Lite Read Address
Read Data Channel Ports				
AXI4L_CS_TARG_RVALID	1	Output	ACLK	AXI4-Lite Read Data Valid
AXI4L_CS_TARG_RREADY	1	Input	ACLK	AXI4-Lite Read Data Ready
AXI4L_CS_TARG_RDATA	32	Output	ACLK	AXI4-Lite Read Data
AXI4L_CS_TARG_RRESP	2	Output	ACLK	AXI4-Lite Read Response

Table 3-5. AXI4-Stream I/O Target Interface

Port	Width	Direction	Clock Domain	Description
AXI4S_IO_TARG_TVALID	1	Input	SYS_CLK	AXI4-Stream Valid Indicates that the initiator is driving a valid transfer.
AXI4S_IO_TARG_TREADY	1	Output	SYS_CLK	AXI4-Stream Ready Indicates that the core can accept a transfer in current clock cycle.
AXI4S_IO_TARG_TDATA	64	Input	SYS_CLK	AXI4-Stream Data Host provides the RapidIO header in the first beat. The header format for supported RapidIO packets are defined under the Header Format section. If the RapidIO data payload is available, then host shall provide them in the following beats.
AXI4S_IO_TARG_TLAST	1	Input	SYS_CLK	AXI4-Stream Data Last Host shall assert this input to indicate the last data of each RapidIO packet.
AXI4S_IO_TARG_TUSER	96	Input	SYS_CLK	AXI4-Stream Write Data User Host shall provide the following information in the first beat (data on user signal is considered valid only during the first beat) [95:64] Extended address field of RapidIO packet. If the extended address field is 16 bits, then the most significant bits are padded with zeros. If the extended address field does not exist, then these bits are tied to zero. [63:32] DestinationID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros. [31:0] SourceID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros.

Table 3-6. AXI4-Stream I/O Initiator Interface

Port	Width	Direction	Clock Domain	Description
AXI4S_IO_INIT_TVALID	1	Output	SYS_CLK	AXI4-Stream Valid Indicates that the core drives a valid transfer.
AXI4S_IO_INIT_TREADY	1	Input	SYS_CLK	AXI4-Stream Ready Indicates that the target can accept a transfer in current clock cycle.
AXI4S_IO_INIT_TDATA	64	Output	SYS_CLK	AXI4-Stream Data Core provides the RapidIO header in the first beat. The header format for supported RapidIO packets are defined under the Header Format section. If the RapidIO data payload is available, then core provides them in the following beats.
AXI4S_IO_INIT_TLAST	1	Output	SYS_CLK	AXI4-Stream Data Last Core will assert this output to indicate the last data of each RapidIO packet.
AXI4S_IO_INIT_TUSER	104	Output	SYS_CLK	AXI4-Stream Write Data User Core will provide the following information in the first beat (data on user signal is considered valid only during the first beat) [103:97] Reserved [96] CRC error detected on the received IO packet. This bit is valid when AXI4S_IO_INIT_TLAST is set. [95:64] Extended address field of RapidIO packet. If the extended address field is 16 bits, then the most significant bits are padded with zeros. If the extended address field does not exist, then these bits are tied to zero. [63:32] DestinationID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros. [31:0] – sourceID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros.

Table 3-7. AXI4-Stream Maintenance Target Interface

Port	Width	Direction	Clock Domain	Description
AXI4S_MT_TARG_TVALID	1	Input	SYS_CLK	AXI4-Stream Valid Indicates that the initiator drives a valid transfer.
AXI4S_MT_TARG_TREADY	1	Output	SYS_CLK	AXI4-Stream Ready Indicates that the core can accept a transfer in current clock cycle.

.....continued				
Port	**Width**	**Direction**	**Clock Domain**	**Description**
AXI4S_MT_TARG_TDATA	64	Input	SYS_CLK	AXI4-Stream Data Host shall provide the RapidIO header in the first beat. The header format for supported RapidIO packets are defined under the Header Format section. If the RapidIO data payload is available, then host shall provide them in the following beats.
AXI4S_MT_TARG_TLAST	1	Input	SYS_CLK	AXI4-Stream Data Last Host shall assert this input to indicate the last data of each RapidIO packet.
AXI4S_MT_TARG_TUSER	72	Input	SYS_CLK	AXI4-Stream Write Data User Host shall provide the following information in the first beat (data on user signal is considered valid only during the first beat) [71:64] hop_count field of RapidIO packet. [63:32] destinationID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros. [31:0] – sourceID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros.

Table 3-8. AXI4-Stream Maintenance Initiator Interface

Port	Width	Direction	Clock Domain	Description
AXI4S_MT_INIT_TVALID	1	Output	SYS_CLK	AXI4-Stream Valid Indicates that the core drives a valid transfer.
AXI4S_MT_INIT_TREADY	1	Input	SYS_CLK	AXI4-Stream Ready Indicates that the target can accept a transfer in current clock cycle.
AXI4S_MT_INIT_TDATA	64	Output	SYS_CLK	AXI4-Stream Data Core will provide the RapidIO header in the first beat. The header format for supported RapidIO packets are defined under the Header Format section. If the RapidIO data payload is available, then core will provide them in the following beats.
AXI4S_MT_INIT_TLAST	1	Output	SYS_CLK	AXI4-Stream Data Last Core will assert this output to indicate the last data of each RapidIO packet.

.....continued				
Port	**Width**	**Direction**	**Clock Domain**	**Description**
AXI4S_MT_INIT_TUSER	80	Output	SYS_CLK	AXI4-Stream Write Data User Core will provide the following information in the first beat (data on user signal is considered valid only during the first beat) [79:73] Reserved [72] CRC error detected with the received Maintenance packet. This bit is valid when AXI4S_MT_INIT_TLAST is set. [71:64] hop_count field of RapidIO packet. [63:32] destinationID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros. [31:0] sourceID field of RapidIO packet. If the destinationID field is 8 bits or 16 bits, then the most significant bits are padded with zeros.

Table 3-9. SRIO Transmit Interface

Port	Width	Direction	Clock Domain	Description
EPCS_TX_DATA_LN0 [EPCS_DATA_WIDTH -1: 0]	EPCS_DATA_WIDTH	Output	LANE0_TX_CLK	Lane 0 Transmit Data to SERDES.
EPCS_0_TX_VAL_LN0	1	Output	LANE0_TX_CLK	Lane 0 Transmit valid: This signal is used to transmit valid data. If deasserted, the PMA macro is put in electrical idle 1. This signal must be generated one clock cycle earlier than corresponding EPCS_TXDATA signals. These signals are active-high.

Table 3-10. SRIO Receive Interface

Port	Width	Direction	Clock Domain	Description
EPCS_RX_DATA_LN0 [EPCS_DATA_WIDTH-1: 0]	EPCS_DATA_WIDTH	Input	LANE0_RX_CLK	Lane 0 Receive Data from SERDES.
EPCS_0_RX_VAL_LN0	1	Input	LANE0_RX_CLK	Lane 0 Receiver detects incoming data (not in electrical idle) and CDR PLL is locked to the input bit stream.
EPCS_RX_IDLE_LN0	1	Input	LANE0_TX_CLK	Lane 0 PHY Receive Idle: This signal is used to signal an electrical idle condition detected by the PMA control logic.

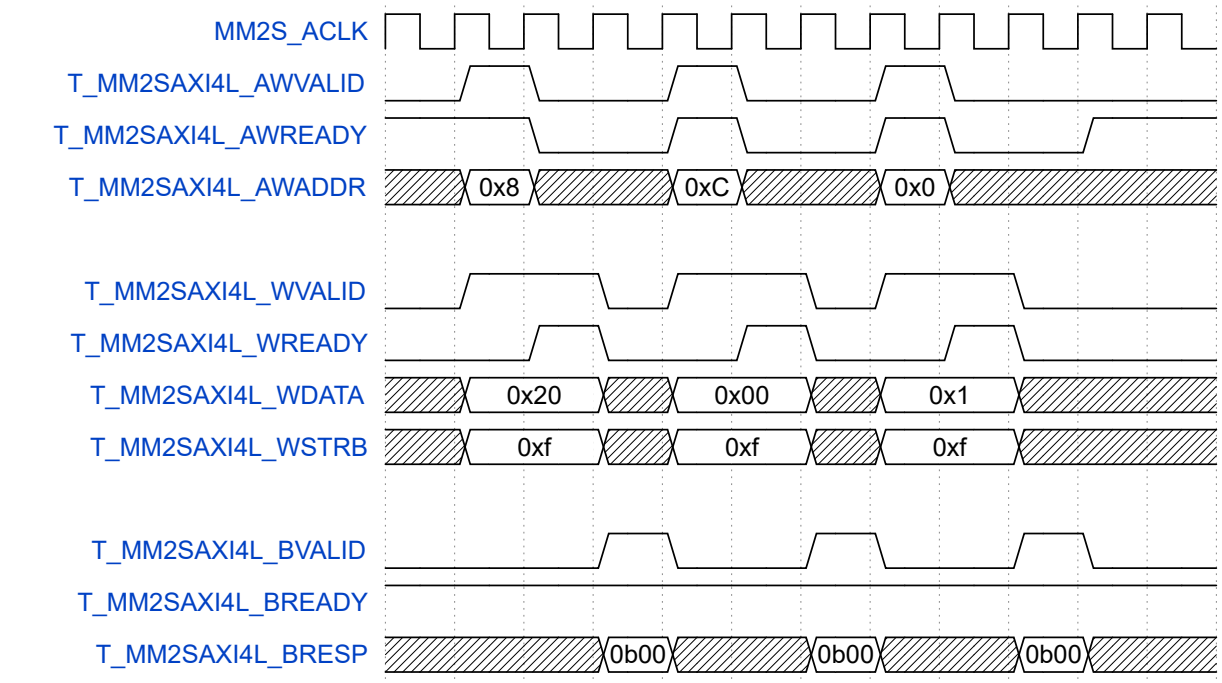
4. Timing Diagrams

This section discusses various timing diagrams.

4.1 AXI4 Lite Target Write Transaction

The following figure shows the AXI4 lite target write transaction timing diagram.

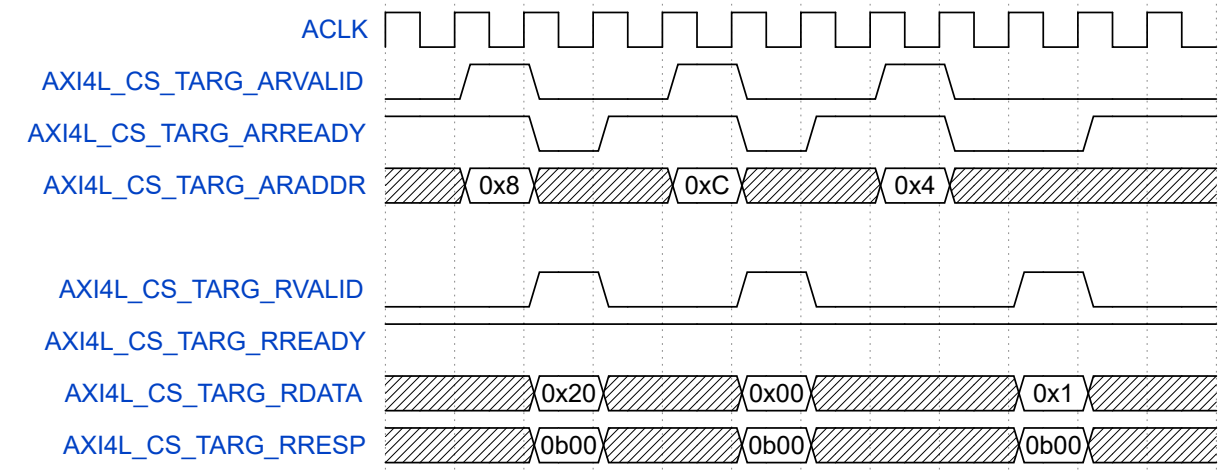
Figure 4-1. AXI4 Lite Target Write Transaction



4.2 AXI4 Lite Target Read Transaction

The following figure shows the AXI4 lite target read transaction timing diagram.

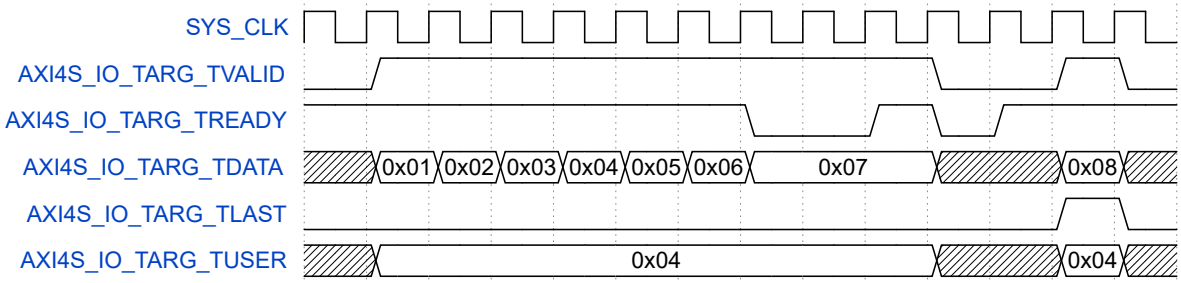
Figure 4-2. AXI4 Lite Target Read Transaction



4.3 AXI4 Stream Interface

The following figure shows the AXI4 stream target transaction timing diagram.

Figure 4-3. AXI4 Stream Target Transaction



5. Implementation of IP Core in Libero Design Suite

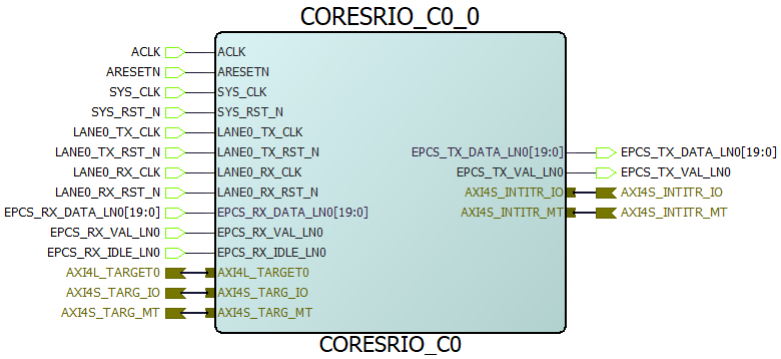
This section describes implementation of the CoreSRIO in Libero Design Suite.

5.1 SmartDesign

CoreSRIO is available for download to the SmartDesign IP Catalog through the Libero SoC web repository. For more information on using SmartDesign to instantiate, configure, connect, and generate cores, see the Libero SOC online help.

The following figure shows an example of an instantiated view of CoreSRIO on the SmartDesign canvas.

Figure 5-1. Instantiation of CoreSRIO on Smart Design Canvas



5.1.1 Configuring the CoreSRIO

This following figures show how the core instance can be configured using its configuration GUI.

Figure 5-2. CoreSRIO Configuration GUI

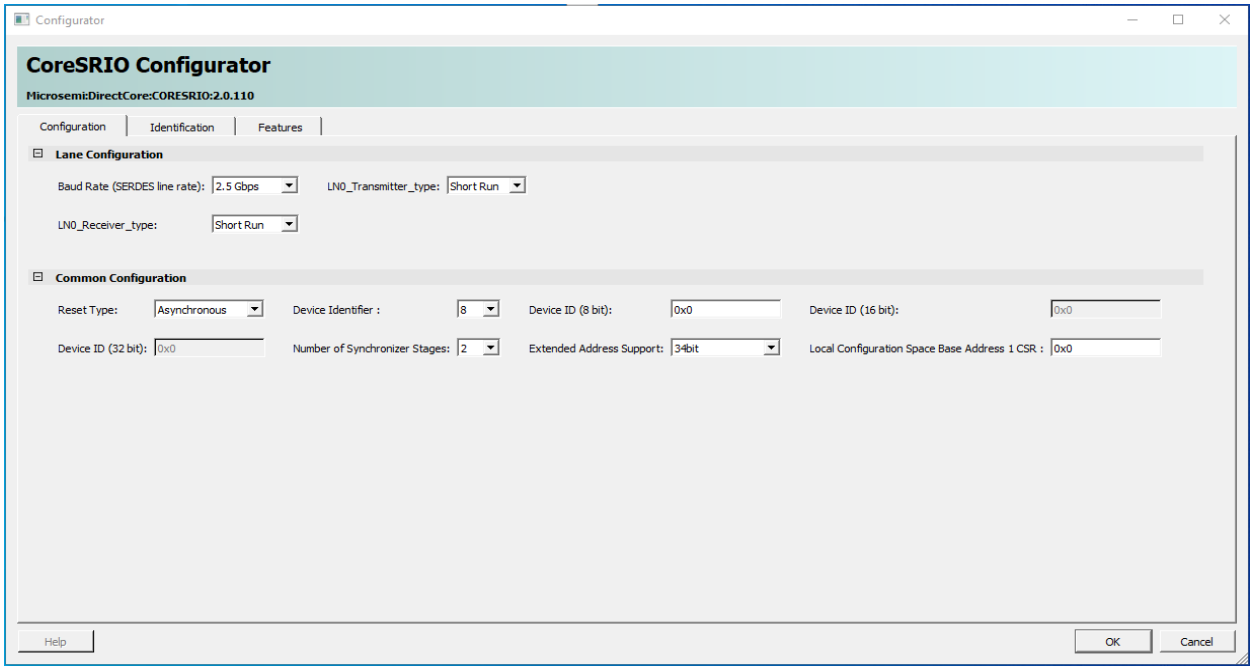


Figure 5-3. CoreSRIO Identification Configuration GUI

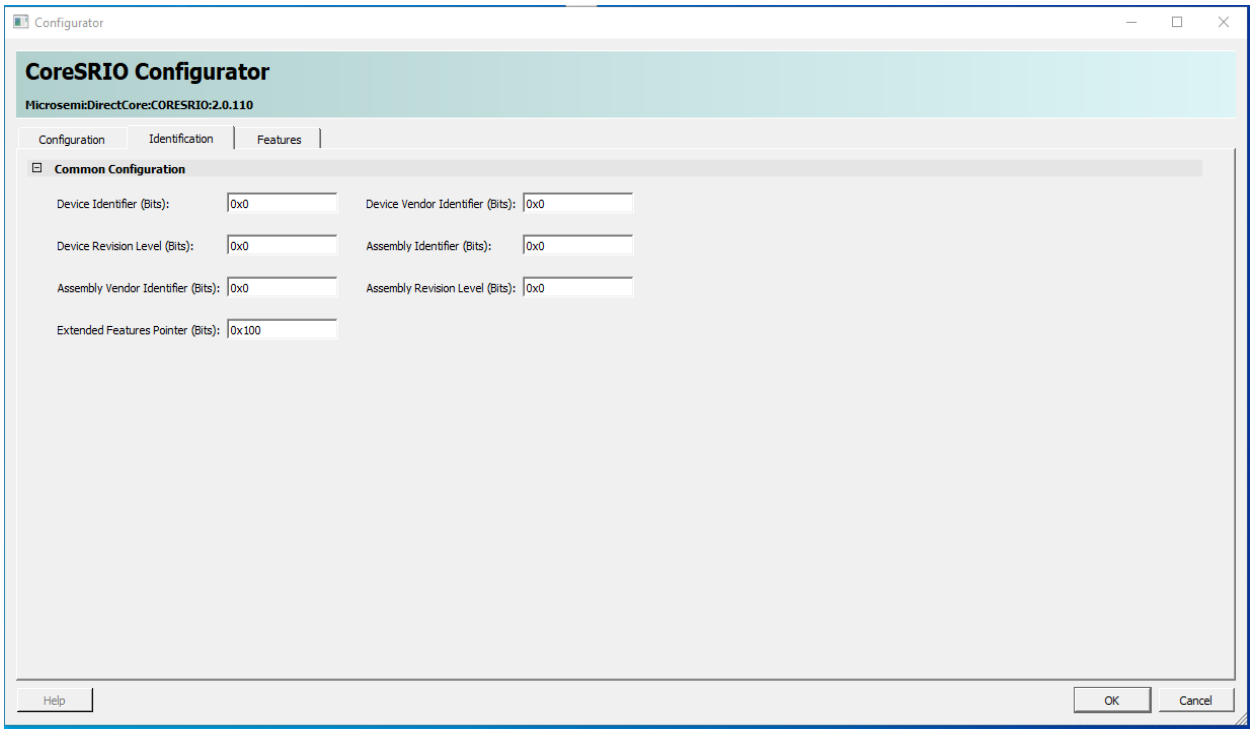
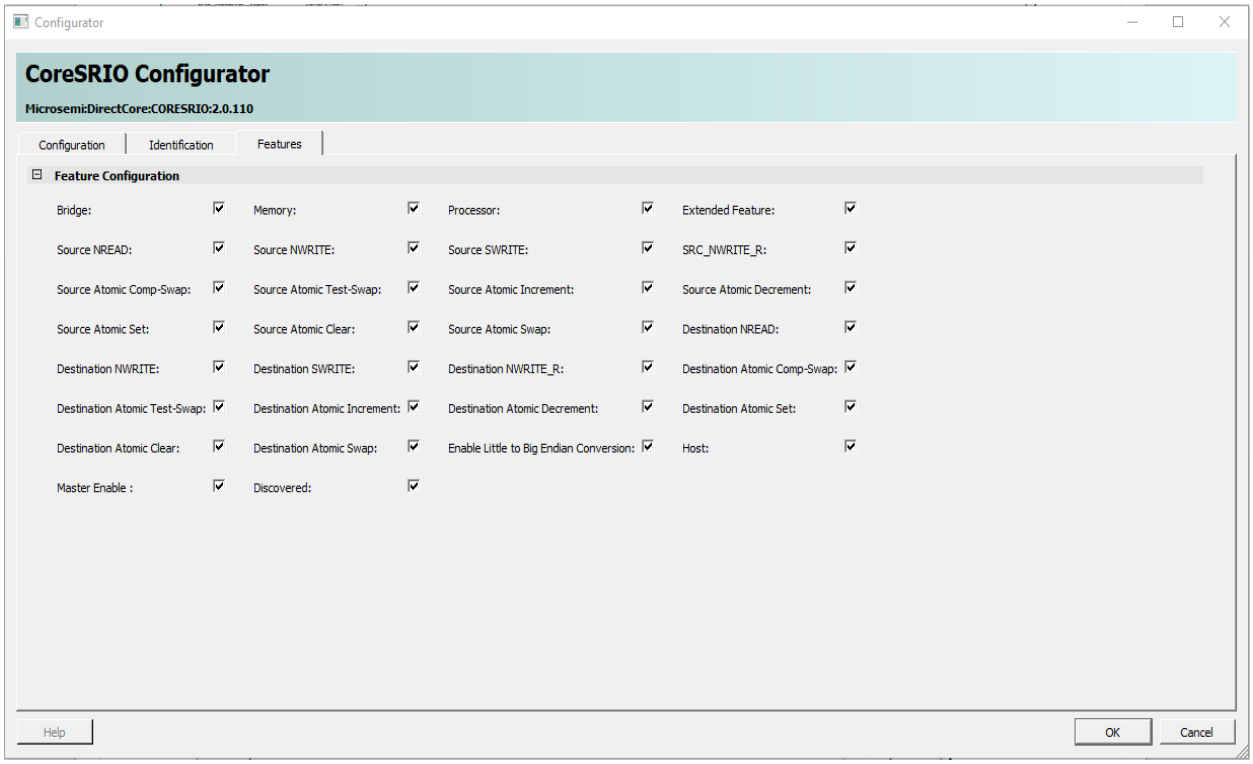


Figure 5-4. CoreSRIO Feature Configuration GUI



5.1.2 Synthesis in Libero SoC

To run synthesis, perform the following steps:

1. With the configuration selected in the configuration GUI, set the design root appropriately.

2. Under **Implement Design**, on the **Design Flow** tab, right click **Synthesize** and click **Run**.

5.1.3 Place and Route in Libero SoC

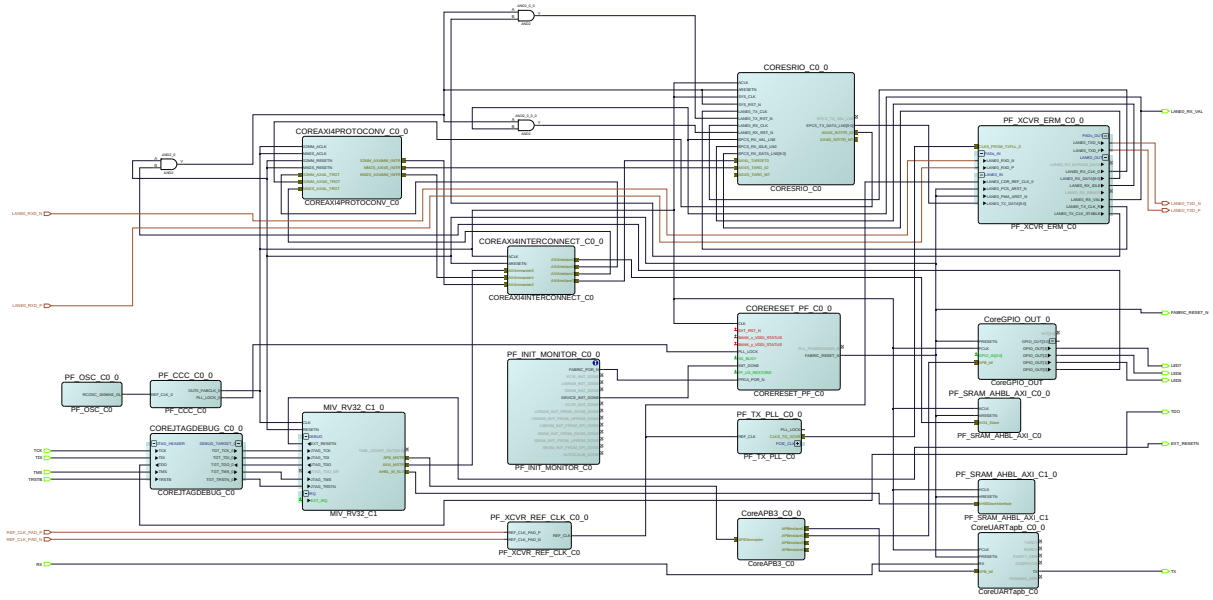
To run the place and route, perform the following steps:

1. With the configuration selected in the configuration GUI, set the design root appropriately.
2. Under **Implement Design**, on the **Design Flow** tab, right click **Place and Route** and click **Run**.

5.1.4 System Integration

This section provides hints to ease the integration of CoreSRIO.

Figure 5-5. CoreSRIO System Integration



Reset for SRIO is generated from MiV by driving GPIO output.

- SRIO_C0_0 has LANE0_RX_CLK and LANE0_TX_CLK clocks.
- LANE0_TX_CLK is connected to 125 MHz LANE0_TX_CLK_R of PF_XCVR_0.
- LANE0_RX_CLK is connected to 125 MHz LANE0_RX_CLK_G of PF_XCVR_0.
- The AXI4ProtoConv is used for memory mapped to stream interface and vice-versa.
- AXI4Lite interface is used to configure SRIO Registers.
- The ACLK and SYS_CLK of SRIO is connected to 50 MHz OUT0_FAB_CLK_0 of PF_CCC_C0_0.

6. Register Map and Descriptions

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DEV_ID_CAR	7:0	DEV_VEN_ID[7:0]							
		15:8	DEV_VEN_ID[15:8]							
		23:16	DEV_ID[7:0]							
		31:24	DEV_ID[15:8]							
0x04	DEV_INFO_CAR	7:0	DEV_REV[7:0]							
		15:8	DEV_REV[15:8]							
		23:16	DEV_REV[23:16]							
		31:24	DEV_REV[31:24]							
0x08	ASSY_ID_CAR	7:0	ASSY_VEN_ID[7:0]							
		15:8	ASSY_VEN_ID[15:8]							
		23:16	ASSY_ID[7:0]							
		31:24	ASSY_ID[15:8]							
0x0C	ASSY_INFO_CAR	7:0	EXTD_FEATURES_PTR[7:0]							
		15:8	EXTD_FEATURES_PTR[15:8]							
		23:16	ASSY_REV[7:0]							
		31:24	ASSY_REV[15:8]							
0x10	PE_FEATURES_CAR	7:0				DEV16_SPRT	EXTD_FEATURES	EXTD_ADDR_SPRT[2:0]		
		15:8				DEV32_SPRT				
		23:16								
		31:24		MEMORY	PROCESSOR					
0x14 ... 0x17	Reserved									
0x18	SRC_OPR_CAR	7:0	SRC_AINC	SRC_ADEC	SRC_ASET	SRC_ACLR	SRC_ASWP			
		15:8	SRC_NREAD	SRC_NWRITE	SRC_SWRITE	SRC_NWRITE_R			SRC_ACSWP	SRC_ATSWP
		23:16								
		31:24								
0x1C	DEST_OPR_CAR	7:0	DEST_AINC	DEST_ADEC	DEST_ASET	DEST_ACLR	DEST_ASWP			
		15:8	DEST_NREAD	DEST_NWRITE	DEST_SWRITE	DEST_NWRITE_R			DEST_ACSWP	DEST_ATSWP
		23:16								
		31:24								
0x20 ... 0x4B	Reserved									
0x4C	PE_LL_CTRL_CSR	7:0						EXTD_ADDR_CTRL[2:0]		
		15:8								
		23:16								
		31:24								
0x50 ... 0x5B	Reserved									
0x5C	LCSBA1_CSR	7:0	LCSBA[7:0]							
		15:8	LCSBA[15:8]							
		23:16	LCSBA[23:16]							
		31:24		LCSBA[30:24]						
0x60	BDID_CSR	7:0	DEV16_BASE_DEV_ID[7:0]							
		15:8	DEV16_BASE_DEV_ID[15:8]							
		23:16	DEV8_BASE_DEV_ID[7:0]							
		31:24								
0x64	DEV32_BDID_CSR	7:0	DEV32_BDID_CSR[7:0]							
		15:8	DEV32_BDID_CSR[15:8]							
		23:16	DEV32_BDID_CSR[23:16]							
		31:24	DEV32_BDID_CSR[31:24]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x68	HOST_BDID_LOCK_CSR	7:0	HOST_BASE_DEV32_ID[7:0]							
		15:8	HOST_BASE_DEV32_ID[15:8]							
		23:16	HOST_BASE_DEV_ID[7:0]							
		31:24	HOST_BASE_DEV_ID[15:8]							
0x6C	COMP_TAG_CSR	7:0	COMP_TAG_CSR[7:0]							
		15:8	COMP_TAG_CSR[15:8]							
		23:16	COMP_TAG_CSR[23:16]							
		31:24	COMP_TAG_CSR[31:24]							
0x70 ... 0xFF	Reserved									
0x0100	LP_SRB_HDR	7:0	EF_ID[7:0]							
		15:8	EF_ID[15:8]							
		23:16	EF_PTR[7:0]							
		31:24	EF_PTR[15:8]							
0x0104 ... 0x013B	Reserved									
0x013C	LP_PORT_GEN_CTRL	7:0								
		15:8								
		23:16								
		31:24	HOST	MASTER	DISCOVERED					
0x0140 ... 0x0153	Reserved									
0x0154	LP_PORT_0_CTRL2	7:0								
		15:8								
		23:16	2.5_GBAUD_SUPPORT	2.5_GBAUD_ENABLE	3.125_GBAUD_SUPPORT	3.125_GBAUD_ENABLE	5.0_GBAUD_SUPPORT	5.0_GBAUD_ENABLE		
		31:24	SELECTED_BAUDRATE[3:0]						1.25_GBAUD_SUPPORT	1.25_GBAUD_ENABLE
0x0158	LP_PORT_0_ERR_STAT	7:0							PORT_OK	PORT_UNINITIALIZED
		15:8						INPUT_RETRY_STOPPED		
		23:16				OUTPUT_RETRY_ENCOUNTERED	OUTPUT_RETRIED	OUTPUT_RETRY_STOPPED		
		31:24			IDLE_SEQUENCE[1:0]		FLOW_CONTROL_MODE			
0x015C	LP_PORT_0_CTRL	7:0								PORT_TYPE
		15:8								
		23:16								
		31:24			INITIALIZED_PORT_WIDTH[2:0]			PORT_WIDTH_OVERRIDE[2:0]		
0x0160 ... 0x0FFF	Reserved									
0x1000	LP_SRB_LANE_HDR	7:0	EF_ID[7:0]							
		15:8	EF_ID[15:8]							
		23:16	EF_PTR[7:0]							
		31:24	EF_PTR[15:8]							
0x1004 ... 0x100F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1010	LP_LN0_STATUS_0	7:0	LANE SYNC STATE CHANGE							
		15:8			RECEIVER LANE SYNC	RECEIVER LANE READY	8B/10B DECODING ERRORS [3:0]			
		23:16	LANE_NUMBER[3:0]				TRANSMITTER_TYPE	TRANSMITTER_MODE	RECEIVER_TYPE[1:0]	
		31:24	PORT_NUMBER[7:0]							

6.1 SRC_OPR_CAR

Name: SRC_OPR_CAR
Offset: 0x018
Reset: 0x0
Property: Read-only

Processing Element Features CAR

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SRC_NREAD	SRC_NWRITE	SRC_SWRITE	SRC_NWRITE_R			SRC_ACSWP	SRC_ATSWP
Access	R	R	R	R			R	R
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	SRC_AINC	SRC_ADEC	SRC_ASET	SRC_ACLR	SRC_ASWP			
Access	R	R	R	R	R			
Reset	0	0	0	0	0			

Bit 15 – SRC_NREAD PE supports read operation.

Bit 14 – SRC_NWRITE PE supports write operation.

Bit 13 – SRC_SWRITE PE supports streaming-write operation.

Bit 12 – SRC_NWRITE_R PE supports write-with-response operation.

Bit 9 – SRC_ACSWP PE supports atomic compare-and-swap operation.

Bit 8 – SRC_ATSWP PE supports test-and-swap operation.

Bit 7 – SRC_AINC PE supports atomic increment operation.

Bit 6 – SRC_ADEC PE supports atomic decrement operation.

Bit 5 – SRC_ASET PE supports atomic set operation.

Bit 4 – SRC_ACLR PE supports atomic clear operation.

Bit 3 – SRC_ASWP PE supports atomic swap operation.

6.2 PE_LL_CTRL_CSR

Name: PE_LL_CTRL_CSR
Offset: 0x04C
Reset: 0x1
Property: Read/Write

Processing Element Logical Layer Control CSR

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EXTD_ADDR_CTRL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	1

Bits 2:0 – EXTD_ADDR_CTRL[2:0]

Controls the number of address bits generated by the PE as a source and processed by the PE as the target of an operation.

Value	Description
0b001	PE supports 34-bit addresses
Others	Reserved

6.3 PE_FEATURES_CAR

Name: PE_FEATURES_CAR
Offset: 0x010
Reset: 0x0
Property: Read-only

Processing Element Features CAR

Bit	31	30	29	28	27	26	25	24
		MEMORY	PROCESSOR					
Access		R	R					
Reset		0	0					

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DEV32_SPRT				
Access				R				
Reset				0				

Bit	7	6	5	4	3	2	1	0
				DEV16_SPRT	EXTD_FEATURES	EXTD_ADDR_SPRT[2:0]		
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 30 – MEMORY PE has physically addressable local address space and can be accessed as an end point through non-maintenance (that is, non-coherent read and write) operations. This local address space may be limited to local configuration registers, or could be on-chip SRAM, and so on.

Bit 29 – PROCESSOR PE physically contains a local processor or similar device that executes code. A device that bridges to an interface that connects to a processor does not count (see bit 31 above).

Bit 12 – DEV32_SPRT

Value	Description
0	PE does not support common transport Dev32
1	PE supports common transport Dev32

Bit 4 – DEV16_SPRT

Value	Description
0	PE does not support common transport Dev16
1	PE supports Dev16

Bit 3 – EXTD_FEATURES PE can bridge to another interface

Bits 2:0 – EXTD_ADDR_SPRT[2:0] Indicates the number of address bits supported by the PE both as a source and target of an operation.

Value	Description
0b001	PE supports 34 bit addresses
Others	Reserved

6.4 LP_SRB_LANE_HDR

Name: LP_SRB_LANE_HDR
Offset: 0x1000
Reset: 0x2000000d
Property: Read-only

Serial Lane Register Block Header. The register contains the extended features pointer (EF_PTR) to the next extended features block and the EF_ID that identifies this as the generic end point LP-Serial register lane block header.

Bit	31	30	29	28	27	26	25	24
	EF_PTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EF_PTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EF_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EF_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 31:16 – EF_PTR[15:0] Pointer to the next block in the extended features data structure.

Bits 15:0 – EF_ID[15:0] Hard-wired extended features ID of LP-Serial Lane Extended Features Block

6.5 LP_SRB_HDR

Name: LP_SRB_HDR
Offset: 0x100
Reset: 0x10000001
Property: Read-only

LP Serial Register Block Header. The register contains the extended features pointer (EF_PTR) to the next extended features block and the EF_ID that identifies this as the generic end point LP-Serial register block header.

Bit	31	30	29	28	27	26	25	24
	EF_PTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EF_PTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EF_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EF_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 31:16 – EF_PTR[15:0] Hard wired pointer to the next block in the data structure, if one exists.

Bits 15:0 – EF_ID[15:0] Hard wired Extended Features Block ID.

6.6 LP_PORT_GEN_CTRL

Name: LP_PORT_GEN_CTRL
Offset: 0x13C
Reset: 0x0
Property: Read/Write

Port General Control CSR. The bits accessible through the Port General Control CSR are bits that apply to all ports in a device. There is a single copy of each such bit per device. These bits are also accessible through the Port General Control CSR of any other Physical Layers implemented on a device.

Bit	31	30	29	28	27	26	25	24
	HOST	MASTER	DISCOVERED					
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – HOST A Host device is a device that is responsible for system exploration, initialization, and maintenance. Agent or slave devices are initialized by Host devices.

Value	Description
0	Agent or slave device
1	Host device

Bit 30 – MASTER The Master Enable bit controls whether or not a device is allowed to issue requests into the system. If the Master Enable is not set, the device may only respond to requests.

Value	Description
0	Processing element cannot issue requests
1	Processing element can issue request

Bit 29 – DISCOVERED This device has been located by the processing element responsible for system configuration.

Value	Description
0	The device has not been previously discovered
1	The device has been discovered by another processing element

6.7 LP_PORT_0_ERR_STAT

Name: LP_PORT_0_ERR_STAT
Offset: 0x158
Reset: 0x0
Property: Read-only,Read-write 1 to clear

Port0 Error and Status CSR

Bit	31	30	29	28	27	26	25	24
			IDLE_SEQUENCE[1:0]		FLOW_CONTROL_MODE			
Access			R	R	R			
Reset			0	0	0			

Bit	23	22	21	20	19	18	17	16
				OUTPUT_RETRY_ENCOUNTERED	OUTPUT_RETRIED	OUTPUT_RETRY_STOPPED		
Access				R/W	R	R		
Reset				0	0	0		

Bit	15	14	13	12	11	10	9	8
						INPUT_RETRY_STOPPED		
Access						R		
Reset						0		

Bit	7	6	5	4	3	2	1	0
							PORT_OK	PORT_UNINITIALIZED
Access							R	R
Reset							0	0

Bits 29:28 – IDLE_SEQUENCE[1:0] Indicates which IDLE sequence is active.

Value	Description
0b00	Idle sequence 1 is active
Others	Reserved

Bit 27 – FLOW_CONTROL_MODE Indicates which flow control mode is active (read-only).

Value	Description
0	Receiver-controlled flow control is active
Others	Reserved

Bit 20 – OUTPUT_RETRY_ENCOUNTERED Output port has encountered a retry condition. This bit is set when bit 18 is set. Once set, remains set until written with a logic 1 to clear.

Bit 19 – OUTPUT_RETRIED Output port has received a packet-retry control symbol and cannot make forward progress. This bit is set when bit 18 is set and is cleared when a packet-accepted or a packet-not-accepted control symbol is received.

Bit 18 – OUTPUT_RETRY_STOPPED Output port has received a packet-retry control symbol and is in the “output retry-stopped” state

Bit 10 – INPUT_RETRY_STOPPED Input port is in the “input retry-stopped” state.

Bit 1 – PORT_OK The input and output ports are initialized, and the port is exchanging error-free control symbols with the attached device.

Bit 0 – PORT_UNINITIALIZED Input and output ports are not initialized. This bit and bit 1 are mutually exclusive.

6.8 LP_PORT_0_CTRL2

Name: LP_PORT_0_CTRL2
Offset: 0x154
Reset: 0x0
Property: Read-only,Read/Write

Port0 Control 2 CSR.These registers are accessed when a local processor or an external device wishes to examine the port baud rate information.

Bit	31	30	29	28	27	26	25	24
	SELECTED_BAUDRATE[3:0]						1.25_GBAUD_SUPPORT	1.25_GBAUD_ENABLE
Access	R	R	R	R			R	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
	2.5_GBAUD_SUPPORT	2.5_GBAUD_ENABLE	3.125_GBAUD_SUPPORT	3.125_GBAUD_ENABLE	5.0_GBAUD_SUPPORT	5.0_GBAUD_ENABLE		
Access	R	R/W	R	R/W	R	R/W		
Reset	0	0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:28 – SELECTED_BAUDRATE[3:0] Indicates the baud rate at which the initialized port is operating. (read-only).

Value	Description
0b0000	No rate selected
0b0001	1.25 Gbaud
0b0010	2.5 Gbaud
0b0011	3.125 Gbaud
0b0100	5.0 Gbaud
Others	Reserved

Bit 25 – 1.25_GBAUD_SUPPORT Indicates whether port operation at 1.25 Gbaud is supported.

Value	Description
0b0	1.25 Gbaud operation not supported
0b1	1.25 Gbaud operation supported

Bit 24 – 1.25_GBAUD_ENABLE Controls whether port operation at 1.25 Gbaud is enabled.

Value	Description
0b0	1.25 Gbaud operation disabled
0b1	1.25 Gbaud operation enabled **Note:** The port shall not allow this bit to be set unless it supports 1.25 Gbaud.

Bit 23 – 2.5_GBAUD_SUPPORT Indicates whether port operation at 2.5 Gbaud is supported.

Value	Description
0b0	2.5 Gbaud operation not supported
0b1	2.5 Gbaud operation supported

Bit 22 – 2.5_GBAUD_ENABLE Controls whether port operation at 2.5 Gbaud is enabled.

Value	Description
0b0	2.5 Gbaud operation disabled
0b1	2.5 Gbaud operation enabled **Note:** The port shall not allow this bit to be set unless it supports 2.5 Gbaud

Bit 21 – 3.125_GBAUD_SUPPORT Indicates whether port operation at 3.125 Gbaud is supported.

Value	Description
0b0	3.125 Gbaud operation not supported
0b1	3.125 Gbaud operation supported

Bit 20 – 3.125_GBAUD_ENABLE Controls whether port operation at 3.125 Gbaud is enabled.

Value	Description
0b0	3.125 Gbaud operation disabled
0b1	3.125 Gbaud operation enabled **Note:** The port shall not allow this bit to be set unless it supports 3.125 Gbaud

Bit 19 – 5.0_GBAUD_SUPPORT Indicates whether port operation at 5.0 Gbaud is supported.

Value	Description
0b0	5.0 Gbaud operation not supported
0b1	5.0 Gbaud operation supported

Bit 18 – 5.0_GBAUD_ENABLE Controls whether port operation at 5.0 Gbaud is enabled.

Value	Description
0b0	5.0 Gbaud operation disabled
0b1	5.0 Gbaud operation enabled **Note:** The port shall not allow this bit to be set unless it supports 5.0 Gbaud

6.9 LP_PORT_0_CTRL

Name: LP_PORT_0_CTRL
Offset: 0x15C
Reset: 0x0
Property: Read-only, Read/Write

Port0 Control CSR. The port n control registers contain control register bits for individual ports on a processing element.

Bit	31	30	29	28	27	26	25	24
			INITIALIZED_PORT_WIDTH[2:0]			PORT_WIDTH_OVERRIDE[2:0]		
Access			R	R	R	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PORT_TYPE
Access								R
Reset								0

Bits 29:27 – INITIALIZED_PORT_WIDTH[2:0] Indicates the width of the link after port initialization.

Value	Description
0b000	1 lane (1x mode)
Others	Reserved

Bits 26:24 – PORT_WIDTH_OVERRIDE[2:0] Override for Initialized Port Width. A change in the value of the Port Width Override field causes the port to re-initialize using the new field value.

Value	Description
0b000	No override
0b001	Reserved
0b010	Force 1x lane 0
0b011	Force 1x lane R
Others	Reserved

Bit 0 – PORT_TYPE This indicates the port type.

Value	Description
0b0	Reserved
0b1	Serial port

6.10 HOST_BDID_LOCK_CSR

Name: HOST_BDID_LOCK_CSR
Offset: 0x068
Reset: 0xffff
Property: Read/Write Once

Host Base Device ID Lock CSR. This register is a write-once/reset-able register which provides a lock function. This register is writable only once after the reset. All subsequent writes after the first write is ignored except when the write data matches the current value of this register. If the write data matches the current value of this register, then register is reset to default value (32'h0000FFFF). After writing this register, a processing element must read this register to verify that it owns the lock before attempting to initialize this processing element.

Bit	31	30	29	28	27	26	25	24
	HOST_BASE_DEV_ID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HOST_BASE_DEV_ID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOST_BASE_DEV32_ID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	HOST_BASE_DEV32_ID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:16 – HOST_BASE_DEV_ID[15:0] This is the base device ID for the PE that is initializing this PE.

Bits 15:0 – HOST_BASE_DEV32_ID[15:0] If the Processing Element Features CAR Dev32 Support bit is 0, then this field is reserved and shall have a constant value of 0. If the Processing Element Features CAR Dev32 Support bit is 1, this field contains the most significant 16 bits of the Dev32 base device ID for the PE that is initializing this PE.

6.11 LP_LN0_STATUS_0

Name: LP_LN0_STATUS_0
Offset: 0x1010
Reset: 0x0
Property: Read-only, Read on clear

Lane0 Status 0 CSR This register contains status information about the local lane transceiver. There is one register per lane, based on the lane width of the generated core. All bits in the register are read-only. Lane0 Status CSRs

Bit	31	30	29	28	27	26	25	24
	PORT_NUMBER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LANE_NUMBER[3:0]				TRANSMITTER_TYPE	TRANSMITTER_MODE	RECEIVER_TYPE[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RECEIVER LANE SYNC	RECEIVER LANE READY	8B/10B DECODING ERRORS [3:0]			
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LANE SYNC STATE CHANGE							
Access	R							
Reset	0							

Bits 31:24 – PORT_NUMBER[7:0] Port number to which lane is assigned. Hard-wired to 0x00.

Bits 23:20 – LANE_NUMBER[3:0] The number of the lane within the port to which the lane is assigned. n is the Lane number. Hard wired to 0b0000 for Lane 0, and 0b0001 for Lane 1

Bit 19 – TRANSMITTER_TYPE Transmitter type.

Value	Description
0b0	Short run
Others	Reserved

Bit 18 – TRANSMITTER_MODE Transmitter operating mode.

Value	Description
0b0	Short run
Others	Reserved

Bits 17:16 – RECEIVER_TYPE[1:0] Receiver type.

Value	Description
0b00	Short run
Others	Reserved

Bit 13 – RECEIVER LANE SYNC Indicates the state of the lane_sync signal

Bit 12 – RECEIVER LANE READY Indicates the state of the lane_ready signal.

Bits 11:8 – 8B/10B DECODING ERRORS [3:0] Indicates the number of 8B/10B decoding errors that have been detected for this lane since the register was last read. This field is reset to 0b0000 when the register is read (errors during the read might be dropped). If >15 errors are received before the register is cleared, the register holds the value of 0b1111.

Bit 7 – LANE SYNC STATE CHANGE When set, indicates that there has been a change in the value of the lane_sync signal. Resets to 0b0 when the register is read.

6.12 LCSBA1_CSR

Name: LCSBA1_CSR
Offset: 0x05C
Reset: 0x0
Property: Read/Write

Local Configuration Space Base Address 1 CSR

Bit	31	30	29	28	27	26	25	24
		LCSBA[30:24]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LCSBA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LCSBA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LCSBA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:0 – LCSBA[30:0] Bits [33:3] of a 34-bit local physical address

6.13 DEV_INFO_CAR

Name: DEV_INFO_CAR
Offset: 0x004
Reset: 0x0
Property: Read-only

Device Info CAR

Bit	31	30	29	28	27	26	25	24
	DEV_REV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DEV_REV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DEV_REV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEV_REV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DEV_REV[31:0] Device Revision Level identifies the revision level of the device

6.14 DEV32_BDID_CSR

Name: DEV32_BDID_CSR
Offset: 0x064
Reset: 0x0
Property: Read/Write

Dev32 Base Device ID CSR

Bit	31	30	29	28	27	26	25	24
	DEV32_BDID_CSR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DEV32_BDID_CSR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DEV32_BDID_CSR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEV32_BDID_CSR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DEV32_BDID_CSR[31:0] This is the Dev32 device ID of the device (must be valid for end point device and if bit 12 of the Processing Element Features CAR is set)

6.15 DEV_ID_CAR

Name: DEV_ID_CAR
Offset: 0x000
Reset: 0x0
Property: Read-only

Device Identity CAR

Bit	31	30	29	28	27	26	25	24
	DEV_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DEV_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DEV_VEN_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEV_VEN_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DEV_ID[15:0] Device Identifier identifies the type of device from the vendor

Bits 15:0 – DEV_VEN_ID[15:0] Device Vendor Identifier identifies the type of device from the vendor

6.16 DEST_OPR_CAR

Name: DEST_OPR_CAR
Offset: 0x01C
Reset: 0x0
Property: Read-only

Processing Element Features CAR

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DEST_NREAD	DEST_NWRITE	DEST_SWRITE	DEST_NWRITE_R			DEST_ACSWP	DEST_ATSWP
Access	R	R	R	R			R	R
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	DEST_AINC	DEST_ADEC	DEST_ASET	DEST_ACLR	DEST_ASWP			
Access	R	R	R	R	R			
Reset	0	0	0	0	0			

Bit 15 – DEST_NREAD PE supports read operation.

Bit 14 – DEST_NWRITE PE supports write operation.

Bit 13 – DEST_SWRITE PE supports streaming-write operation.

Bit 12 – DEST_NWRITE_R PE supports write-with-response operation.

Bit 9 – DEST_ACSWP PE supports atomic compare-and-swap operation.

Bit 8 – DEST_ATSWP PE supports test-and-swap operation.

Bit 7 – DEST_AINC PE supports atomic increment operation.

Bit 6 – DEST_ADEC PE supports atomic decrement operation.

Bit 5 – DEST_ASET PE supports atomic set operation.

Bit 4 – DEST_ACLR PE supports atomic clear operation.

Bit 3 – DEST_ASWP PE supports atomic swap operation.

6.17 COMP_TAG_CSR

Name: COMP_TAG_CSR
Offset: 0x06C
Reset: 0x0
Property: Read/Write

Component Tag CSR

Bit	31	30	29	28	27	26	25	24
	COMP_TAG_CSR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMP_TAG_CSR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMP_TAG_CSR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP_TAG_CSR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP_TAG_CSR[31:0] This is a component tag for PE.

6.18 BDID_CSR

Name: BDID_CSR
Offset: 0x060
Reset: 0x0
Property: Read/Write

Base Device ID CSR

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DEV8_BASE_DEV_ID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DEV16_BASE_DEV_ID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DEV16_BASE_DEV_ID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – DEV8_BASE_DEV_ID[7:0] This is the Dev8 device ID of the device (endpoint devices only)

Bits 15:0 – DEV16_BASE_DEV_ID[15:0] This is the Dev16 device ID of the device (must be valid for end point device and if bit 4 of the Processing Element Features CAR is set)

6.19 ASSY_INFO_CAR

Name: ASSY_INFO_CAR
Offset: 0x00C
Reset: 0x0
Property: Read-only

Assembly Information CAR

Bit	31	30	29	28	27	26	25	24
	ASSY_REV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ASSY_REV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EXTD_FEATURES_PTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTD_FEATURES_PTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ASSY_REV[15:0] Assembly Revision Level identifies the revision level of the assembly

Bits 15:0 – EXTD_FEATURES_PTR[15:0] Extended Features Pointer Pointer to first entry in Extended Feature List

6.20 ASSY_ID_CAR

Name: ASSY_ID_CAR
Offset: 0x008
Reset: 0x0
Property: Read-only

Assembly Identity CAR

Bit	31	30	29	28	27	26	25	24
	ASSY_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ASSY_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ASSY_VEN_ID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ASSY_VEN_ID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ASSY_ID[15:0] Assembly Identifier identifies the type of assembly from the vendor

Bits 15:0 – ASSY_VEN_ID[15:0] Assembly Vendor Identifier identifies the vendor that manufactured the assembly or subsytem containing the device

7. Testing of CoreSRIO

This core has been validated using the PolarFire evaluation board in the external loopback mode only.

8. Additional References

For updates and additional information about the software, devices, and hardware, visit the **Intellectual Property** pages on the Microchip FPGAs and PLDs website.

8.1 Known Issues and Workarounds

There are no known limitations and workarounds for CoreSRIO v2.0 because this is the initial release.

8.2 Discontinued Features and Devices

There are no discontinued features and devices for CoreSRIO v2.0 because this is the initial release.

8.3 Ordering Codes

Order CoreSRIO through your local Microchip sales representative or through microchipDIRECT. Use the following number convention when ordering: CoreSRIO-XX. XX is listed in the following table.

Table 8-1. Ordering Codes

XX	Description
OM	Obfuscated RTL node locked license for Windows platform
RM	RTL source node locked license for Windows platform
OMFL	Obfuscated floating license for Windows or Linux platform
RMFL	RTL source floating license for Windows or Linux platform

9. Glossary

Following are the list of terms and definitions used in the document.

Table 9-1. Terms and Definitions

Term	Definition
AXI4	AMBA Advanced eXtensible Interface
CAR	Capability Register
CDC	Clock Domain Crossing
CRC	Cyclic Redundancy Check
CSR	Control and Status Register
PCS	Physical Coding Sublayer
PHY	Physical Layer
PMA	Physical Media Access
RTL	Register Transfer Level
SRIO	Serial Rapid IO

10. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the current publication.

Revision	Date	Description
A	02/2023	Initial release

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-

ISBN: 978-1-6683-1913-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820