Lattice Avant-G/X MPPHY Module - Lattice Radiant Software

User Guide

FPGA-IPUG-02233-1.0

December 2023

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 REDUDANCIES, 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

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
BIST	Built-In Self Test
CPRI	Common Public Radio Interface
CTC	Clock Tolerance Compensation
FEC	Forward Error Correction
FPGA	Field Programmable Gate Array
LMMI	Lattice Memory-Mapped Interface
MP	Multi-Protocol
MPP	Multi-Protocol Peripherals
PCS	Physical Coding Sublayer
PHY	Physical layer
PMA	Physical Media Attachment
PRBS	Pseudo Random Bit Stream
RTL	Register Transfer Level

1. Introduction

The Lattice Avant™ MPPHY Module supports the most common high-speed SerDes protocols user for inter-chip connectivity.

It is implemented as a Quad-based (4-lane) SerDes which is the fundamental block of this IP.

This design is implemented in Verilog. It can be targeted to Lattice Avant devices and implemented using the Lattice Radiant™ software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.1. Quick Facts

Table 1.1 shows a summary of the MPPHY Module.

Table 1.1. Quick Facts

IP Requirements	Supported FPGA Family	Lattice Avant-G/X
Resource Utilization	Targeted Devices	LAV-AT-G70, LAV-AT-X70
	Supported User Interface	LMMI (Lattice Memory Mapped Interface)
Design Tool Support	Lattice Implementation	Lattice Radiant software 2023.1 or later
	Synthesis	Synopsys® Synplify Pro® for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant software user guide.

1.2. Features

Key features of the MPPHY Module include:

- Quad-based SerDes.
- Supports Ethernet Protocol in Lane 0.
- MPPCS Features
- 64-bit/66-bit coding
- Multiple Quad Alignment
  - TX lane-lane skew
  - RX lane De-skew
- Integrated Loopback Modes for System Debugging

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal Names that end with:

- _*n* are active low
- _*i* are input signals
- _*o* are output signals
- _io are bi-directional input/output signals

1.3.3. Attribute

The names of attributes in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Description

2.1. Overview

The MPPHY Module supports the most common high-speed SerDes protocols used for inter-chip connectivity. It is controlled by a number of registers that can be dynamically reconfigured by user logic or configuration module through LMMI ports. Reset and Configure Manager (RCM) in the MPPHY module handles the MPPHY initialization and reset sequence.

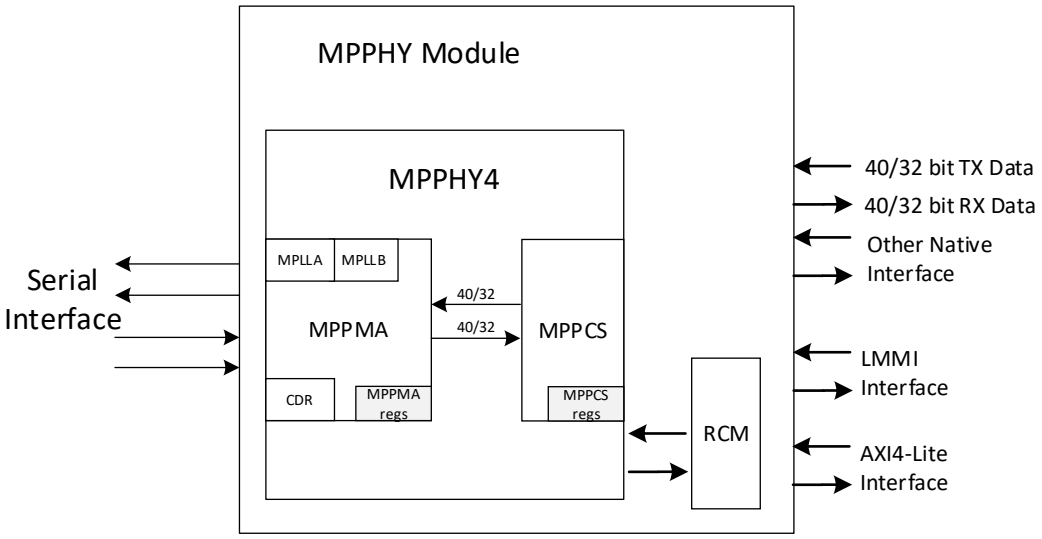

Figure 2.1. MPPHY Module Block Diagram

2.1.1. Transmit Path

Figure 2.2 shows the MPPCS TX Path block diagram for one lane.

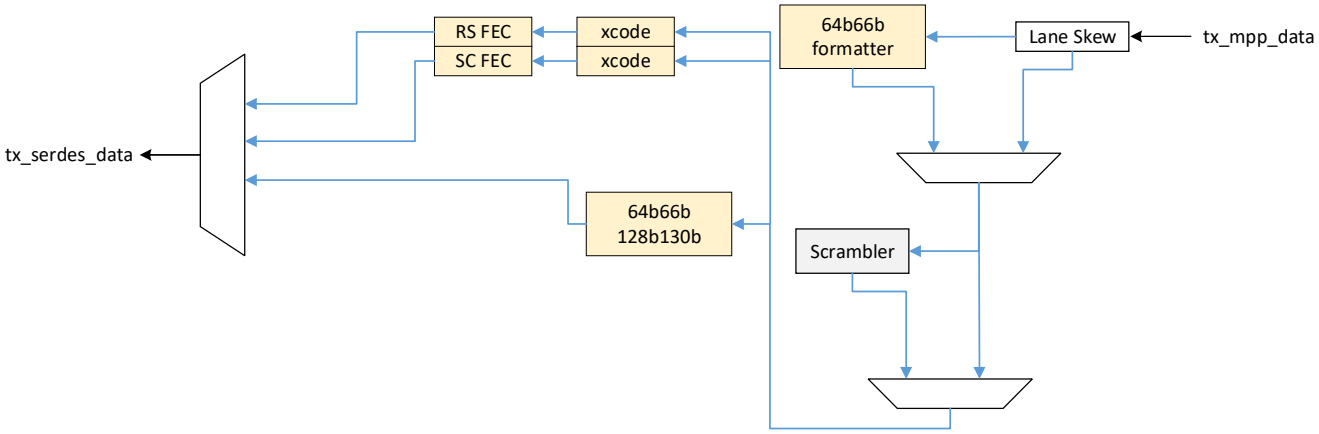

Figure 2.2. MPPCS TX Data Path Block Diagram

Table 2.1 Shows the MPPCS TX Path submodules and how to enable/disable them.

Table 2.1. MPPCS TX Path

Submodule	Description
Lane Skew	Can be optionally bypassed by setting *Clock Tolerance Compensation Enable*. This function is supported by an elastic buffer with controllable latency. It serves as a basic FIFO for clock-domain crossing between lane-specific clocks and consolidated pipe clock. It also provides the capability for intentional skew or de-skew between lanes. In addition, it supports clock tolerance compensation.
Scrambler	Can be optionally bypassed by setting *Scrambler Enable*. This function is supported by a programmable multi-mode LFSR of the order 64.
Block Encoder	Can be set using *Datapath Encoding Mode*. The encoder block is instantiated four times in each lane in order to support a lane width of up to 40 bits.
Forward Error Correction (FEC)	Can be set using *FEC Mode* in the Transmitter Group. It is only supported for the operations at the highest data rate. This function is supported by a programmable multi-mode LFSR of the order 32.

2.1.2. Receive Path

Figure 2.3 shows the MPPCS RX Path block diagram for one lane.

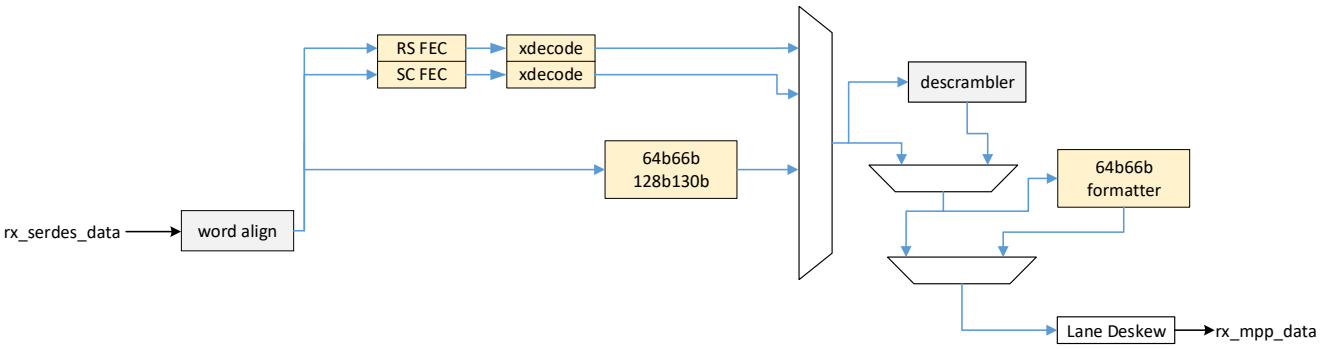

Figure 2.3. MPPCS RX Data Path Block Diagram

Table 2.2 shows the MPPCS RX Path submodules and how to enable/disable them.

Table 2.2. MPPCS RX Path

Submodule	Description
Word Align	Can be optionally bypassed by setting *Word Aligner Enable*. To determine the word boundary of the data stream, data is compared with a fixed pattern of every possible alignment position equal to the data width. Each data bit is compared with a corresponding pattern bit. The alignment scores a hit when every data bit at a given alignment position matches the pattern bit or the mask bit is set at a particular position. For every alignment position, a counter increments when a hit is scored. An arbiter outputs the alignment position with the highest score. Patterns and other setting can be set using *Word Aligner Pattern, Word Aligner Mask, Word Aligner Operation* and *Word Aligner Threshold.*
Forward Error Correction (FEC)	Can be set using *FEC Mode* in the Receiver Group. It is only supported for the operations at the highest data rate. This function is supported by a programmable multi-mode LFSR of the order 32.
Block Decoder	Can be set using *Datapath Decoding Mode*. The encoder block is instantiated four times in each lane in order to support a lane width of up to 40 bits.
Descrambler	Can be optionally bypassed by setting De*scrambler Enable*. This function is supported by a programmable multi-mode LFSR of the order 64.
Lane De-Skew	This function is supported by an elastic buffer with controllable latency.

2.1.3. PCS 64B66B Transmit-Receive Path

Figure 2.4 shows the detailed PCS data path for 64B66B. Some of the Protocols that use this encoding/decoding scheme are Ethernet 10G/25G.

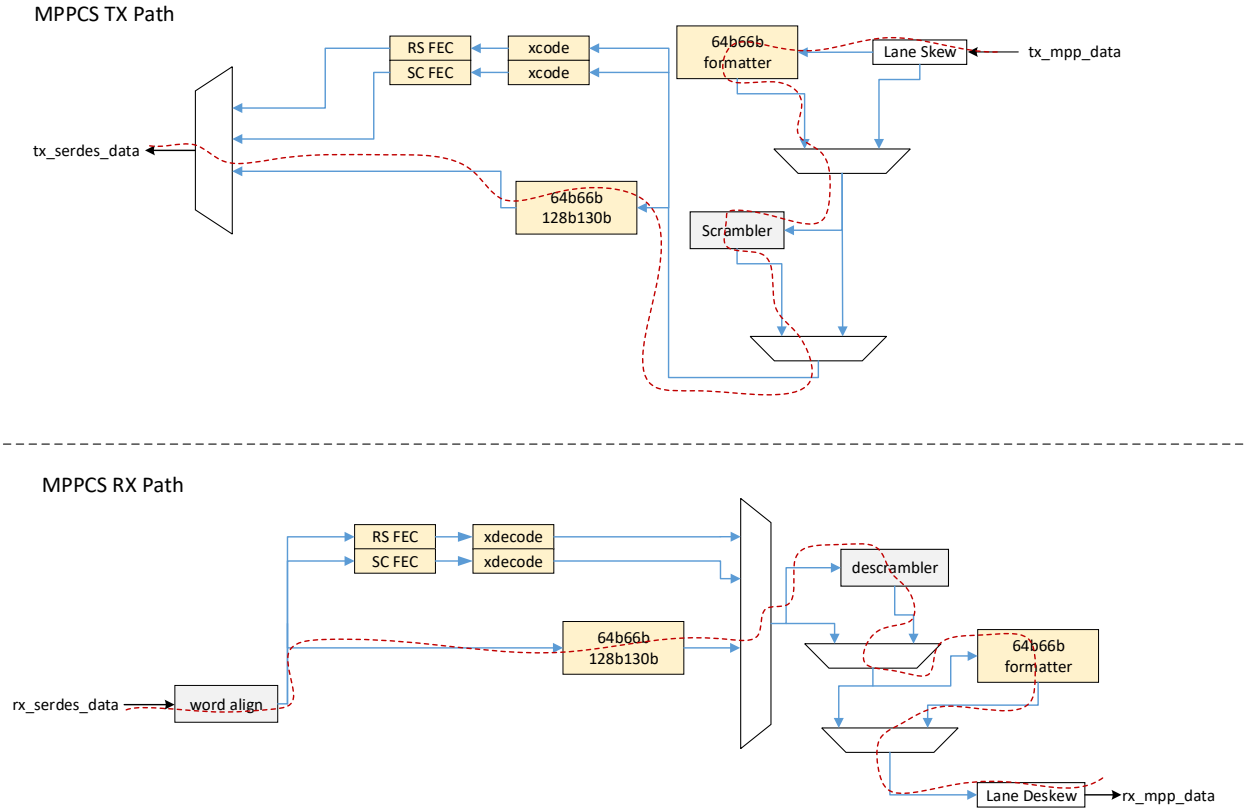

Figure 2.4. PCS 64B66B TX/RX Path

Table 2.3. Data Rate and Reference Clock Frequency

Segment	Standard	Protocol Name	Data Rate (Gbps)	Coding	Reference Clock (MHz)	TX/RX Output Clock (MHz)
Compute	Ethernet	10G BASE-R	10.3125	64b66b	156.25	156.25
		25GBASE-R	25.78125	64b66b	156.25	195.3125

2.1.4. Loopbacks

Figure 2.5 shows the data flow of MPPCS loopback modes. For MPPCS TX and RX Path sub blocks, refer to Figure 2.2 and Figure 2.3 respectively.

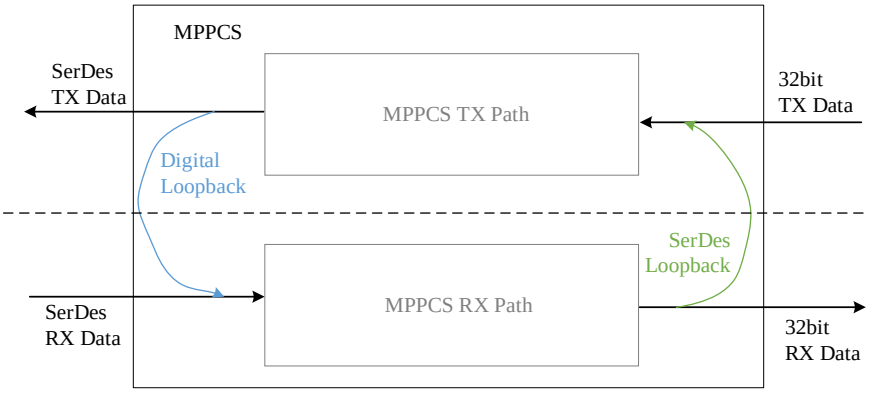

Figure 2.5. MPPCS Loopback Diagram

Table 2.4. MPPCS Loopback Description

Loopback	Description
Digital Loopback	Loops back the TX data from MPPCS back to the input RX data. Can be used to test and debug the digital datapath end-to-end.
SerDes Loopback	Loops back the output fabric RX data of MPPCS back to the input fabric TX data of MPPCS block.

2.1.5. Reset Control Scheme with RCM Enable

Reset and Configure Manage (RCM) soft IP embedded in the MPPHY module supports initialization and warm reset sequence for MPPHY subsystem. RCM supports multiple warm reset entries to MPPHY subsystem through core interface. Table 2.1 below lists the supported warm reset scheme in RCM soft IP for various reset entries.

Table 2.5. RCM IP Reset Scheme Support

Reset Scheme							
Reset Port	PMA Rx Channel	PMA Tx Channel	PCS Rx Channel	PCS Tx Channel	PCIE Core	PCIe core user Interface	DMA
pin_perst_n_i(1)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
rx_phy_rst_n_i[3:0]	Yes	No	Yes	No	Yes	Yes	Yes
tx_phy_rst_n_i[3:0]	No	Yes	No	Yes	Yes	Yes	Yes

Note:

1. Only applicable for PCIe protocols.

2.2. Signal Description

Table 2.6. MPPHY Module Signal Description

Port Name	I/O	Width	Description
Serial IO[1]			
txp_l[n]_o	Out	NL	Serial transmitter lane, differential signal with positive polarity
txn_l[n]_o	Out	NL	Serial transmitter lane, differential signal with negative polarity
rxp_l[n]_i	In	NL	Serial receiver lane, differential signal with positive polarity
rxn_l[n]n_i	In	NL	Serial receiver lane, differential signal with negative polarity
Clock and Reset[2]			
txinclk_i	In	NL	Input clock in the Transmit path. Loopback clock from PMA for source-synchronous clocking. It is recommended to use a clock that is faster than tx_out_clk for asynchronous operation.
txrst_l[n]_i	In	NL	Active high TX reset on each lane.
rxinclk_i	In	NL	Input clock in the Transmit path. It is recommended to use a clock that is faster than rx_out_clk for asynchronous operation.
rxrst_l[n]_i	In	NL	Active high RX reset on each lane.
txout_clk_l[n]_o	Out	NL	Output transmit clock from PMA with applicable down-gear.
txout_gclk_q[m]_o	Out	1	Output transmit clock forwarded to global clock distribution.
rxout_clk_l[n]_o	Out	NL	Output receive clock on lane each lane from PMA with applicable down-gear. It is recommended to use this clock for source-synchronous clocking.
rxout_gclk_q[m]_o	Out	1	Output receive clock forwarded to global clock distribution.
pcs_txrst_l[n]_i	In	NL	Active high TX PCS reset on each lane.
pcs_rxrst_l[n]_i	In	NL	Active high RX PCS reset on each lane.
pma_txrst_l[n]_i	In	NL	Active high TX PMA and PCS reset on each lane. Map to RCM tx_phy_rst_n_i[n] reset port.
pma_rxrst_l[n]_i	In	NL	Active high RX PMA and PCS reset on each lane. Map to RCM rx_phy_rst_n_i[n] reset port.
reset_done_l[n]_o	Out	NL	Reset completion status on each lane. Reset is in progress when resset_done is low.
MPP Signals[2]			
txdata_l[n]_i	In	192	Input transmit data on each lane. Refer to Table 2.7 for the data mapping of each mode.
txvalid_l[n]_i	In	NL	Input flow-control feed-forward signal.
txready_l[n]_o	Out	NL	Output flow-control feed-back signal.
rxdata_l[n]_o	Out	192	Output received data on each lane. Refer to Table 2.7 for the data mapping of each mode.
rxvalid_l[n]_o	Out	NL	Output flow-control feed-forward signal.
rxready_l[n]_i	In	NL	Input flow-control feed-back signal.
LMMI Interface			
lmmi_clk_q[m]_i	In	1	LMMI Clock Input
lmmi_resetn_q[m]_i	In	1	Active Low LMMI Reset
lmmi_request_q[m]_i	In	1	Start Transaction
lmmi_wr_rdn_q[m]_i	In	1	Write = HIGH, Read = LOW
lmmi_offset_q[m]_i	In	16	Address/Offset
lmmi_wdata_q[m]_i	In	16	Write Data
lmmi_rdata_valid_q[m]_o	Out	1	Valid Data Indicator
lmmi_ready_q[m]_o	Out	16	Slave ready signal
lmmi_rdata_q[m]_o	Out	16	Read data
AXI4-Lite Interface			
axi_q[m]_clk_i	In	1	AXI4-Lite clock

Port Name	I/O	Width	Description
axi_q[m]_rst_n_i	In	1	Active Low AXI4-Lite Reset
axi_q[m]_awvalid_i	In	1	Write Address Valid
axi_q[m]_awready_o	Out	1	Write Address Ready
axi_q[m]_awaddr_i	In	17	Write Address
axi_q[m]_awprot_i	In	3	Unused
axi_q[m]_wvalid_i	In	1	Write Data Valid
axi_q[m]_wready_o	Out	1	Write Data Ready
axi_q[m]_wdata_i	In	32	Write Data
axi_q[m]_wstrb_i	In	4	Write Data Strobe
axi_q[m]_bvalid_o	Out	1	Write Response Valid
axi_q[m]_bready_i	In	1	Write Response Ready
axi_q[m]_bresp_o	Out	2	Write Response. Always 2’b00 (Okay)
axi_q[m]_arvalid_i	In	1	Read Address Valid
axi_q[m]_arready_o	Out	1	Read Address Ready
axi_q[m]_araddr_i	In	17	Read Address
axi_q[m]_arprot_i	In	3	Unused
axi_q[m]_rvalid_o	Out	1	Read Data Valid
axi_q[m]_rready_i	In	1	Read Data Ready
axi_q[m]_rdata_o	Out	32	Read Data
axi_q[m]_rresp_o	Out	2	Read Response. Always 2’b00 (Okay)
Quad-Quad Signals			
refclkp_q[m]_i	In	1	Quad reference clock, positive polarity.
refclkn_q[m]_i	In	1	Quad reference clock, negative polarity.

Notes:

1. [n] indicates lane/channel number.
2. These ports are generated with prefix mpp_*
3. [m] indicates Quad number.
4. NL means Number of Lanes.

2.2.1. Data Bus Mapping

MPPHY Module data bus width is up to 40-bit wide which is usually occupied by up to 32-bit of data and 8-bit of control signals. Additional control signals are mapped into sideband signals.

Table 2.7. Data Bus Mapping

[191:184] Sideband	[183:176] Sideband	[175:168] Sideband	[167:160] Sideband	[159:152] Control	[151:120] Data	[119:112] Control	[111:80] Data	[79:72] Control	[71:40] Data	[39:32] Control	[31:0] Data

2.2.1.1. 64b66b Protocols

For 64b66b protocols do not require 4 control bits per octet as the data width is always 32 bits. Some of the control bits apply to the entire data packet of 64 or 128 bits.

Table 2.8. TX and RX Mapping for 64b66b

Sideband	Control[1]	Control[0] (Direct Mode)	Control[0] (Codec Mode)	8b data
[167:160] = [7] RSVD [6] pipe_rx_elecidle [5] rx_serdes_valid & word_align_valid [4] word_align_detect [3] pipe_phy_status [2:0] RxPipeStatus [2] |ctc_skip, |ctc_wr_skip_lat, underflow_stretch, overflow_cdc, rx_serdes_valid, pipe_rx_detect, word_align_valid, word_align_detect [1] dec_8b10b_error_status //Decode Error [0] dec_8b10b_rd_status //Disparity Error	[36] = block start [39:37] = reserved	[33:32] = sync header for 802.3 etc. [35:34] = extra sync header	[32] = K indicator of 1st octet	[7:0] = 1st octet
			[33] = K indicator of 2nd octet	[15:8] =2nd octet
			[34] = K indicator of 3rd octet	[23:16] =3rd octet
			[35] = K indicator of 4th octet	[31:24] =4th octet
[175:168] *same mapping as [167:160]	[76] = block start [79:77] = reserved	[73:72] = sync header for 802.3 etc. [75:74] = extra sync header	[72] = K indicator of 1st octet	[47:40] = 1st octet
			[73] = K indicator of 2nd octet	[55: 48] =2nd octet
			[74] = K indicator of 3rd octet	[63:56] =3rd octet
			[75] = K indicator of 4th octet	[71:64] =4th octet
[183:176] *same mapping as [167:160]	[116] = block start [119:117] = reserved	[113:112] = sync header for 802.3 etc. [115:114] = extra sync header	[112] = K indicator of 1st octet	[87:80] = 1st octet
			[113] = K indicator of 2nd octet	[95: 88] =2nd octet
			[114] = K indicator of 3rd octet	[103:96] =3rd octet
			[115] = K indicator of 4th octet	[111:104] =4th octet
[191:184] *same mapping as [167:160]	[156] = block start [159:157] = reserved	[153:152] = sync header for 802.3 etc. [155:154] = extra sync header	[152] = K indicator of 1st octet	[127:120] = 1st octet
			[153] = K indicator of 2nd octet	[135: 128] =2nd octet
			[154] = K indicator of 3rd octet	[143:136] =3rd octet
			[155] = K indicator of 4th octet	[151:144] =4th octet

2.3. Attributes Summary

The configurable attributes of the MPPHY Module are shown in Table 2.9 and are described in Table 2.10. The attributes can be configured through the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Table 2.9. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
Protocol	"Ethernet"	"Ethernet"	
Protocol Mode	"10GBASE-R", "25GBASE-R"	"10GBASE-R"	If Device == "LAV-AT-X70", Selectable Values are: • "10GBASE-R" • "25GBASE-R"
Link Width	1x1	1x1	Uneditable
Link Direction	"Tx and Rx", "Tx only", "Rx only"	"Tx and Rx"	Uneditable
Refclk Frequency	N/A	156.25	Uneditable
MPLL_A Max Data Rate (Gbps)	N/A	10.3125	Available only if *Protocol Mode == "10GBASE-KR"*
MPLL_A SSC Enable	"Unchecked", "Checked"	"Unchecked"	Available only if *Protocol Mode == "10GBASE-KR"*
MPLL_B Max Data Rate (Gbps)	N/A	25.78125	Available only if *Protocol Mode == "25GBASE-R"*
MPLL_B SSC Enable	"Unchecked", "Checked"	"Unchecked"	Available only if *Protocol Mode == "25GBASE-R"*
RX Line Rate (Gbps)	N/A	10.3125	Uneditable
RX Data Width	N/A	32B	Uneditable
RX FPGA Fabric Gearbox Ratio	1-4	1	Uneditable
RX FPGA Bus Frequency (MHz)	N/A	156.25	Uneditable
TX Line Rate (Gbps)	N/A	10.3125	Uneditable
TX FPGA Bus Width	N/A	32B	Uneditable
TX FPGA Fabric Gearbox Ratio	1-4	1	Uneditable
TX FPGA Bus Frequency	N/A	156.25	Uneditable
RCM Enable	"Unchecked", "Checked"	"Unchecked"	
User Interface Type	"LMMI", "AXI4LITE"	"LMMI",	Editable only if "*RCM Enable*" == "Checked"
PMA Setup			
RX Adaptive Equalization Enable	"Unchecked", "Checked"	"Checked"	Editable if *DFE Enable* == "Unchecked"
DFE Enable	"Unchecked", "Checked"	"Checked"	Uneditable
RX Attenuation Block	(0 - 7)	0	Editable if *RX Adaptive Equalization Enable* == "Unchecked"
RX CTLE Boost	(0 - 31)	0	Editable if *RX Adaptive Equalization Enable* == "Unchecked"
RX VGA1 Gain	(0 - 7)	5	Editable if *RX Adaptive*

Attribute	Selectable Values	Default	Dependency on Other Attributes
			Equalization Enable == "Unchecked"
RX VGA2 Gain	(0 - 7)	5	Editable if *RX Adaptive Equalization Enable* == "Unchecked"
Protocol	N/A	"Ethernet"	Reflects *Protocol*
PCS Setup			
Protocol	N/A	"Ethernet"	Reflects *Protocol*
PCS Loopback	"No Loopback", "SerDes Loopback", "Digital Loopback"	"No Loopback"	
Word Aligner Enable	"Unchecked", "Checked"	"Checked"	Uneditable
Word Aligner Pattern (Hex)	<128-bit hex value>	00000000000000000000000000000000	Uneditable
Word Aligner Mask (Hex)	<128-bit hex value>	00000000000000000000000000000000	Uneditable
Word Aligner Operation Mode		"Align Overwrite"	Uneditable
Word Aligner Threshold	<8-bit hex value>	1B	Uneditable
FEC Mode	"None", "RS-FEC", "SC-FEC"	"None"	Uneditable
Datapath decoding Mode	"8b/10b", "64b/66b"	"64b/66b"	Uneditable
Descrambler Enable	"Unchecked", "Checked"	"Checked"	Uneditable
Clock Tolerance Compensation Enable	"Unchecked", "Checked"	"Checked"	Uneditable
CTC SKIP Pattern	<32-bit hex value>	07070707	Uneditable
Scrambler Enable	"Unchecked", "Checked"	"Checked"	Uneditable
Datapath Encoding Mode	"8b/10b", "64b/66b"	"64b/66b"	Uneditable
FEC Mode	"None", "RS-FEC", "SC-FEC"	"None"	Uneditable

Table 2.10. Attributes Descriptions

Attribute	Description
General	
Protocol	Specifies the selected protocol of the MPPHY module.
Protocol Mode	Specifies the mode of the selected protocol.
Link Width	Specifies the instance x number of lanes. It is automatically defined based on the *Protocol Mode*.
Link Direction	Specifies the direction of PHY: "Tx and Rx", "Tx only", "Rx only"
RefClk Frequency (MHz)	Specifies the PHY Reference Clock Frequency.
Maximum Data Rate	Specifies the data rate for the selected *Protocol Mode.*
SSC Enable	Specifies whether SSC is enabled or not.
RX Line Rate (Gbps)	Specifies the line rate in the receive path.
RX Data Width	Specifies the data width in the receive path.
RX FPGA Fabric Gearbox Ratio	Specifies the gearbox ratio in the receive path.
RX FPGA Bus Frequency	Specifies the bus frequency in the receive path.
TX Line Rate (Gbps)	Specifies the line rate in the transmit path.

Attribute	Description
TX Data Width	Specifies the data width in the transmit path.
TX FPGA Fabric Gearbox Ratio	Specifies the gearbox ratio in the transmit path.
TX FPGA Bus Frequency	Specifies the bus frequency in the transmit path.
RCM Enable	Enabled/Disabled RCM
User Interface Type	Select the user interface protocol either LMMI or AXI4 Lite
PMA Setup	
RX Adaptive Equalization Enable	Specifies whether the Adaptive Equalization is enabled or not.
DFE Enable	Specifies whether DFE is enabled or not.
RX Attenuation Block	Subsequent EQ control of *RX Adaptive Equalization Enable*
RX CTLE Boost	Subsequent EQ control of *RX Adaptive Equalization Enable*
RX VGA1 Gain	Subsequent EQ control of *RX Adaptive Equalization Enable*
RX VGA2 Gain	Subsequent EQ control of *RX Adaptive Equalization Enable*
Protocol	Reflects the *Protocol* attribute
PCS Setup	
Protocol	Reflects the *Protocol* attribute
PCS Loopback	Specifies the available loopbacks inside the PCS module.
Word Aligner Enable	Specifies whether the Word Aligner module inside the MPPHY is enabled or not.
Word Aligner Pattern (Hex)	Specifies the pattern for Word Aligner module.
Word Aligner Mask (Hex)	Specifies the mask for Word Aligner module.
Word Aligner Operation Mode	Specifies the mode of Word Aligner module.
Word Aligner Threshold (Hex)	Specifies the threshold of Word Aligner module.
FEC Mode	When enabled, is supported by a programmable multi-mode LFSR of the order 32. Currently, value is fixed for this attribute.
Datapath Decoding Mode	Specifies the type of decoding mode. Currently, value is fixed for this attribute.
Descrambler Enable	Specifies whether the descrambler is enabled or not.
Clock Tolerance Compensation Enable	Serves as a basic FIFO for clock-domain crossing between lane-specific clocks and consolidated pipe clock. Currently, value is fixed for this attribute.
CTC SKIP Pattern (Hex)	Specifies the Skip Pattern for Clock Tolerance Compensation.
Scrambler Enable	Specifies whether the descrambler is enabled or not.
Datapath Encoding Mode	Specifies the encoding mode. Currently, value is fixed for this attribute.
FEC Mode	When enabled, is supported by a programmable multi-mode LFSR of the order 32. Currently, value is fixed for this attribute.

2.4. Operation Details

2.4.1. PHY Initialization Sequence using test simulation model (RCM-disabled)

To initialize the PHY, all registers needed to monitor are shown in Figure 2.6. Currently, the registers are being checked using simulation model available in the customer testbench.

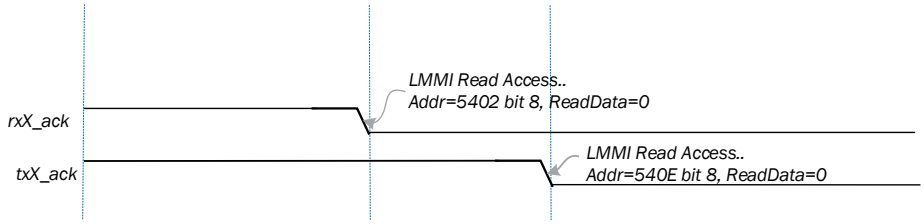

Figure 2.6. PHY Initialization Sequence Timing Diagram

This initialization sequence applies for all operations, unless otherwise stated. After PHY initialization, user can proceed to send and receive data to and from the MPPHY module by following the procedures in the PHY Reset Sequence with RCM Enabled section.

2.4.2. PHY Reset Sequence with RCM Enabled

When RCM (Reset and Configure Manager) is enabled, RCM block embedded in the MPPHY module orchestras the initialization and reset sequence of the entire PHY after powering up.

2.4.2.1. PHY Initialization/Cold Reset Release

After powered up, Avant device control block control and release the global reset. INITBUS start to configure MPPHY release. After that, RCM will take over the sequence in cold reset release phase until user mode.

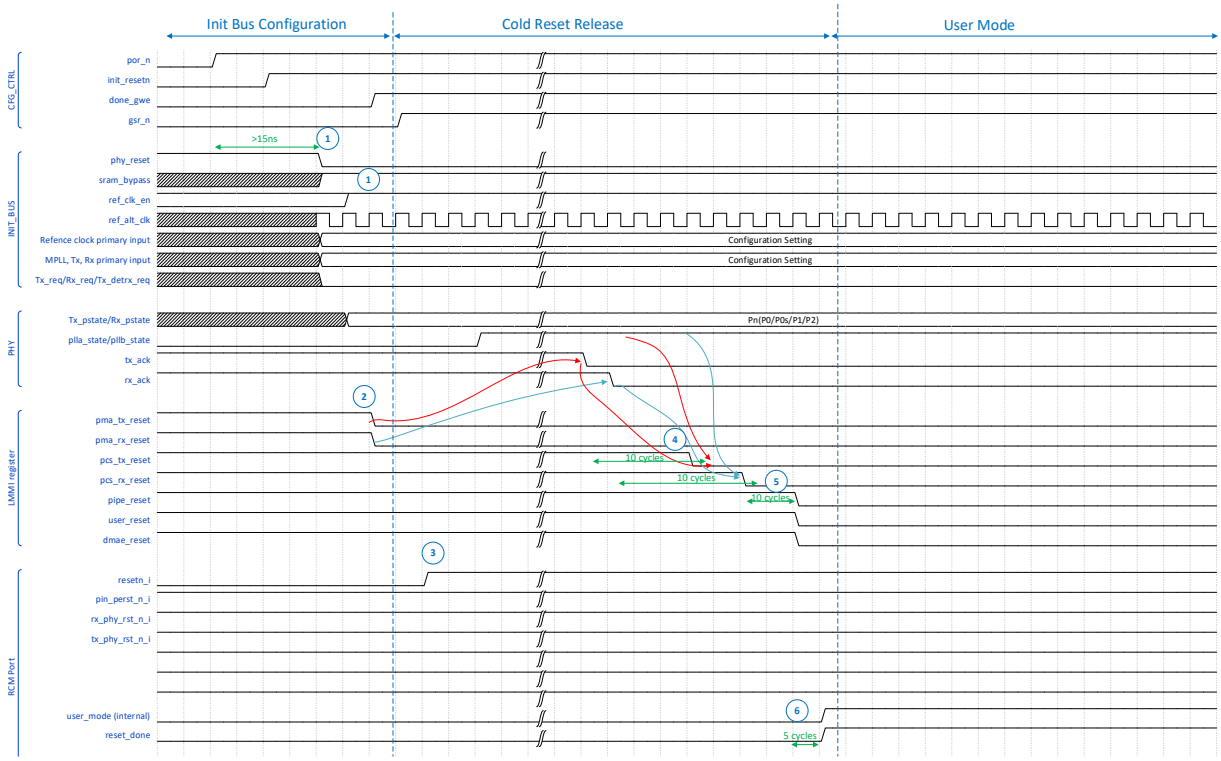

Figure 2.7 HSSI PHY Initialization Sequence

1. After powering up, INITBUS configure the MPPHY with the necessary settings. INITBUS de-assert the PHY_RESET after waiting more than 15ns after power is stable. INITBUS assert ref_clk_en when reference clock is available.
2. INITBUS release PMA channel by de-asserting the pma_tx_reset and pma_rx_reset, then RCM wait for the tx_ack and rx_ack to de-assert.
3. User logic to de-assert RCM reset/LMMI reset when the core fabric is available.
4. INITBUS release PMA channel by de-asserting the pma_tx_reset and pma_rx_reset, then RCM wait for the tx_ack and rx_ack to de-assert.
5. After tx_ack and rx_ack deassert and plla_state/pllb_state is asserted (indicate PLL lock), RCM de-assert the pcs_tx_reset and pcs_rx_reset after 10 cycles to release the PCS channel.
6. After 10 cycles later, RCM IP release PCIe core by de-asserting the pipe_reset, user_reset and dma_reset
7. User_mode and reset_done are asserted after 5 cycles of pipe_reset, user_reset and dma_reset released. MPPHY enter user mode.

2.4.2.2. Warm Reset (PMA reset)

PMA Rx channel and Tx channel reset is trigger by rx_phy_rst_n_i[n] and tx_phy_rst_n_i[n] respectively. Each channel has dedicated PMA reset which allow users to reset each channel independently. Figure 2.5 below shows the reset sequence when both rx_phy_rst_n_i[n] and tx_phy_rst_n_i[n] are triggered together.

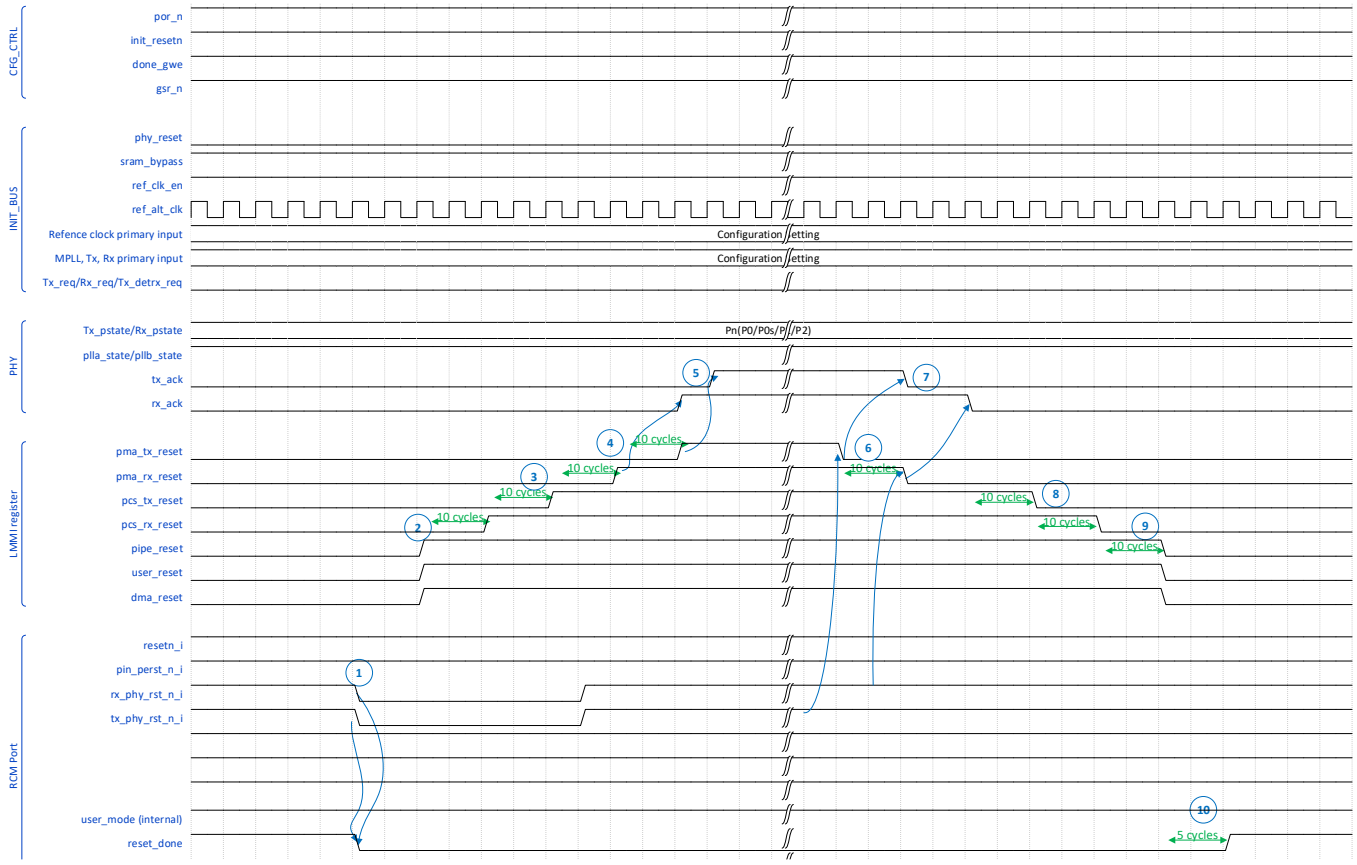

Figure 2.8. PMA Tx and Rx Reset

1. Once rx_phy_rst_n_i and tx_phy_rst_n_i are asserted, reset_done is de-asserted to indicate reset is in progress.
2. RCM assert pipe_reset, dma_reset and user_reset to reset the PCIe core.
3. After 10 cycles, RCM assert pcs_tx_reset and pcs_rx_reset to reset PCS channel.
4. After 10 cycles, RCM assert pma_tx_reset and pma_rx_reset to reset PMA channel.
5. PHY tx_ack and rx_ack are asserted after pma_tx_reset and pma_rx_reset are asserted.
6. After 10 cycles later, RCM de-assert pma_tx_reset and pma_rx_reset to release PMA RX and TX channel if tx_phy_rst_n_i and tx_phy_rst_n_i are de-asserted otherwise will remain resetting PMA channel.
7. PHY tx_ack and rx_ack are deasserted after pma_tx_reset and pma_rx_reset are released.
8. RCM de-assert pcs_tx_reset and pcs_rx_reset 10 cycles after tx_ack and rx_ack are de-asserted to release the PCS channel.
9. After 10 cycles, RCM de-assert pipe_reset, dma_reset and user_reset to release the PCIe core.
10. Reset_done are asserted after 5 cycles of pipe_reset, user_reset and dma_reset released. MPPHY enter user mode.

2.4.3. Data Transmission

To begin PHY transaction, user should provide the required reference clocks with correct frequency to the MPPHY module. MPPMA PLL Settings should be properly set to get the desired TX/RX output clock frequency. After PLL lock, rxvalid_o signal is asserted indicating that the calibration is completed and the MPPHY module is ready to transmit and receive data. MPPHY module requires active high transmit and receive resets. For more details, refer to Figure 2.9.

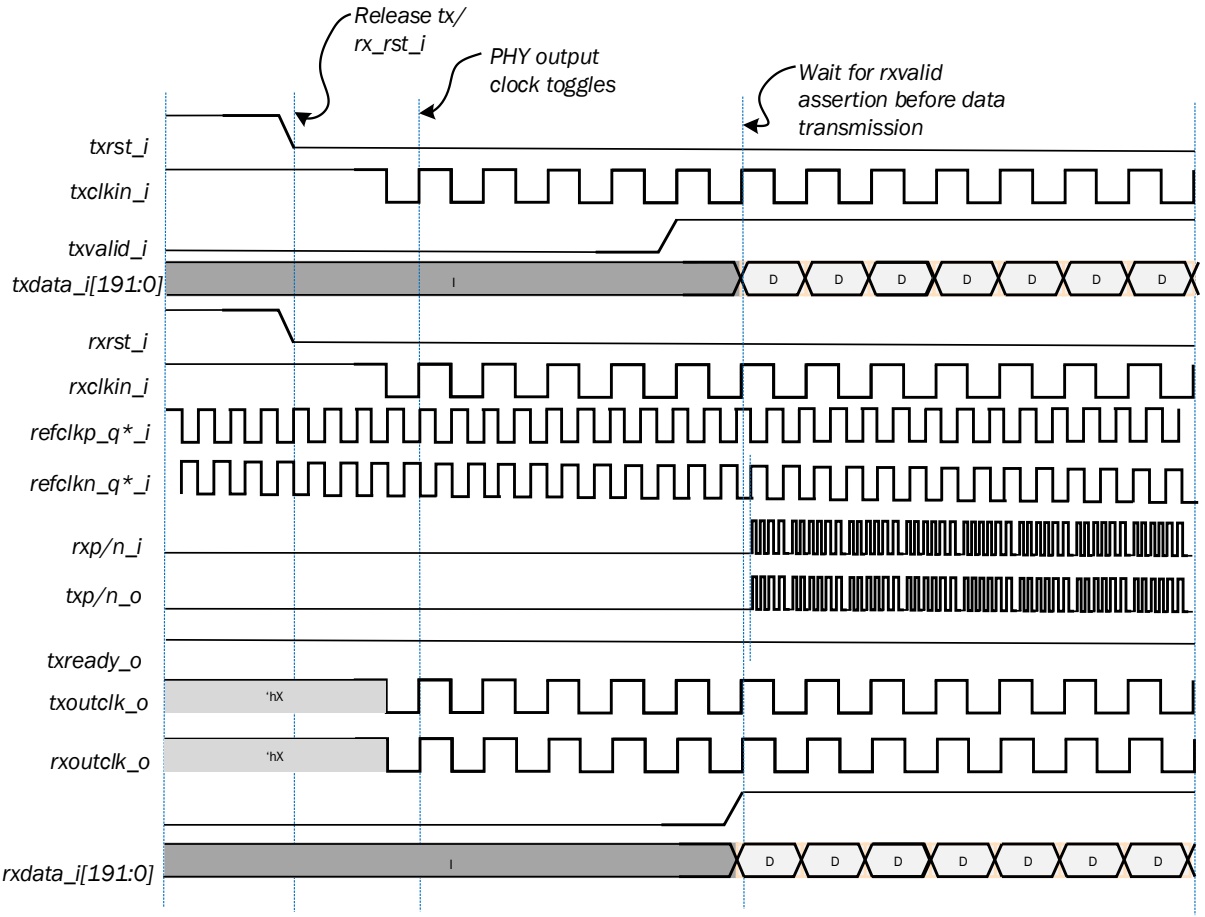

Figure 2.9. PHY Data Transmission Timing Diagram

For rxvalid_o, sideband signals are used to detect the assertion. Table 2.8 shows the mapping: rx_serdes_valid & word_align_valid.

See sample code below:

```
      assign rxvalid_o = mpp_rxdata_l0_o[165] & mpp_rxdata_l0_o[173]; //10G
  assign rxvalid_o = mpp_rxdata_l0_o[165] & mpp_rxdata_l0_o[173] & mpp_rxdata_l0_o[181] &
mpp_rxdata_l0_o[189]; //25G
```

Table 2.11. Expected rxvalid_o assertion for Disabled RCM block in RTL simulation

Protocol Mode/ Loopback Mode	Time (ns) LAV-AT-G70-1LFG1156I	Time (ns) LAV-AT-X70-1LFG1156I
10GBASE-R/ No Loopback	100347 ns	100347 ns
10GBASE-R/ Digital Loopback	82076 ns	82076 ns
10GBASE-R/ SerDes Loopback	94028 ns	94028 ns
25GBASE-R/ No Loopback	–	83981 ns
25GBASE-R/ Digital Loopback	–	102199 ns
25GBASE-R / SerDes Loopback	–	89020 ns

Table 2.12. Expected rxvalid_o assertion for Enabled RCM block using LMMI Interface in RTL simulation

Protocol Mode/ Loopback Mode	Time (ns) LAV-AT-G70-1LFG1156I	Time (ns) LAV-AT-X70-1LFG1156I
10GBASE-R/ No Loopback	123437 ns	123437 ns
10GBASE-R/ Digital Loopback	132755 ns	132755 ns
10GBASE-R/ SerDes Loopback	123706 ns	123706 ns
25GBASE-R/ No Loopback	–	123309 ns
25GBASE-R/ Digital Loopback	–	130925 ns
25GBASE-R / SerDes Loopback	–	121606 ns

Table 2.13. Expected rxvalid_o assertion for Enabled RCM block using AXI Interface in RTL simulation

Protocol Mode/ Loopback Mode	Time (ns) LAV-AT-G70-1LFG1156I	Time (ns) LAV-AT-X70-1LFG1156I
10GBASE-R/ No Loopback	111455 ns	111455 ns
10GBASE-R/ Digital Loopback	113218 ns	113218 ns
10GBASE-R/ SerDes Loopback	114588 ns	114588 ns
25GBASE-R/ No Loopback	–	110335 ns
25GBASE-R/ Digital Loopback	–	112117 ns
25GBASE-R / SerDes Loopback	–	109635 ns

Figure 2.10 shows the overall timing sequence from PHY initialization up to data transmission.

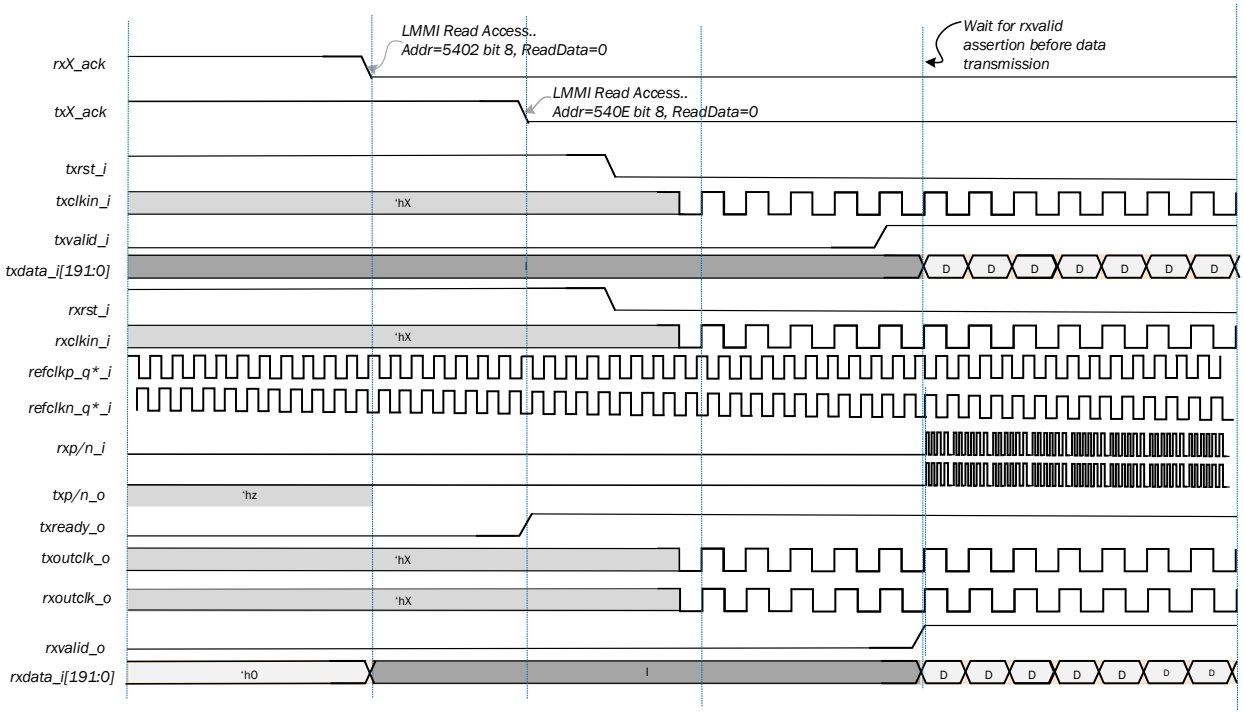

Figure 2.10. MPPPHY Timing Diagram

3. IP Generation and Evaluation

This section provides information on how to generate the MPPHY Module using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant Software, refer to the Lattice Radiant software user guide.

3.1. Licensing the IP

No license is required for this module.

3.2. Generation and Synthesis

The Lattice Radiant software allows users to customize and generate modules and IPs and integrate them into the device's architecture. The procedure for generating the MPPHY Module in Lattice Radiant software is described below.

To generate the MPPHY Module:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the IP Catalog tab, double-click on **MPPHY** under **Module, Architecture_Modules** category.
3. The **Module/IP Block Wizard** opens as shown in Figure 3.1.
4. Enter values in the **Component name** and the **Create in** fields and click **Next**.

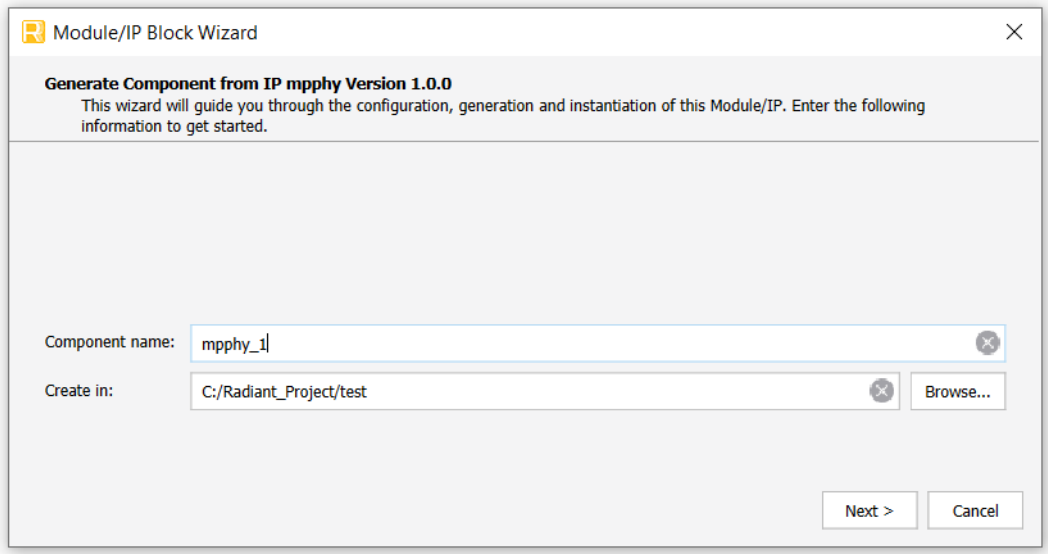

Figure 3.1. Module/IP Block Wizard

5. In the module's dialog box of the **Module/IP Block Wizard** window, customize the selected MPPHY module using the drop-down menus and check boxes. Refer to Figure 3.2 for a sample configuration. For configuration options, refer to the Attributes Summary section.

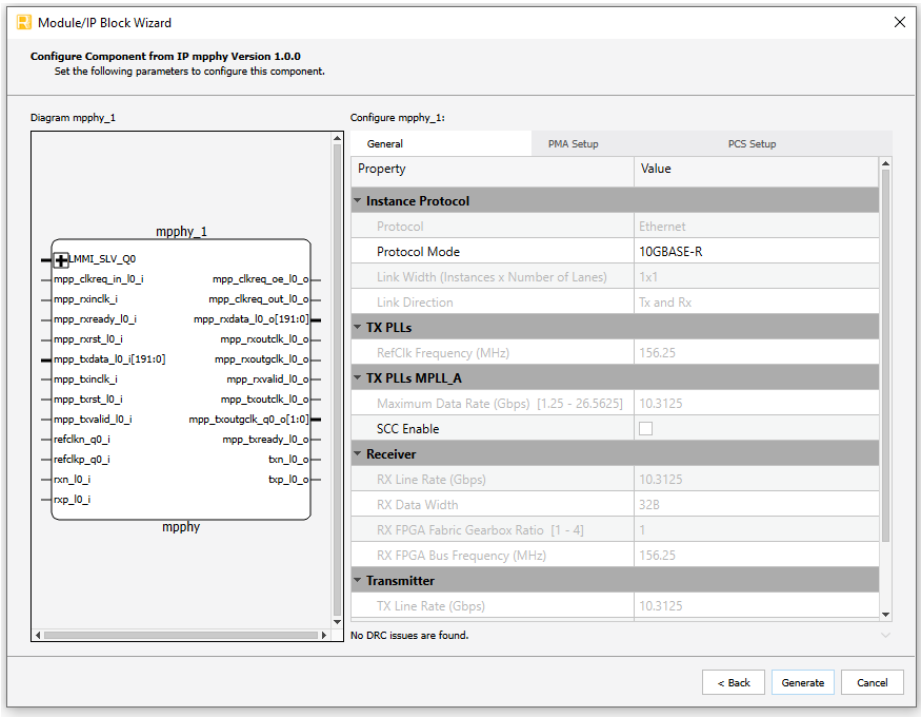

Figure 3.2. Configure User Interface of MPPHY Module

6. Click **Generate**. This shows the design block messages and results as shown in Figure 3.3.

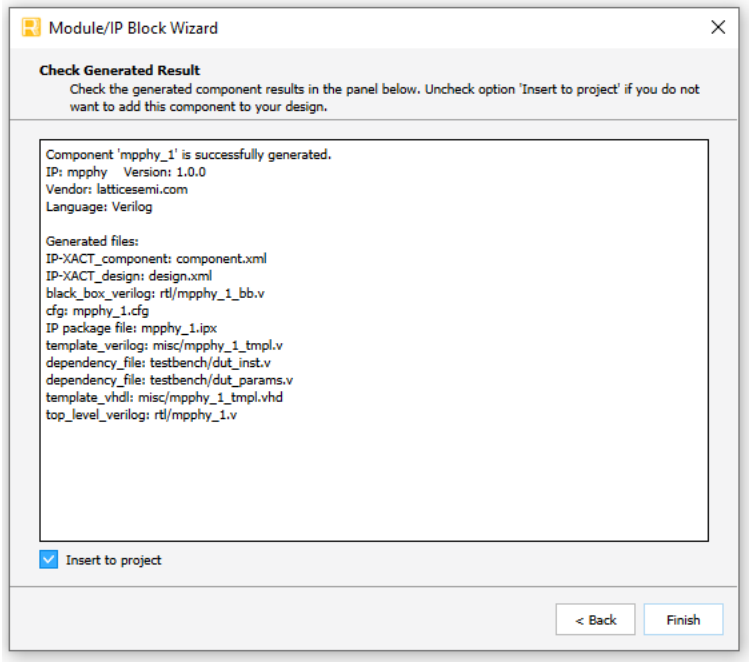

Figure 3.3. Check Generating Result

7. Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 3.1.

The generated MPPHY 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 example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the module is also provided. Users may also use this top-level reference as the starting template for the top-level for their complete design. The generated files are listed in Table 3.1.

Table 3.1. Generated File List

Attribute	Description
<Component name>.ipx	This file contains the information on the files associated to the generated IP.
<Component name>.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an example RTL top file that instantiates the module.
rtl/<Component name>_bb.v	This file provides the synthesis black box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the module.

3.3. Running the Functional Simulation

Running functional simulation can be performed after the IP is generated.

To run functional simulation:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.4.

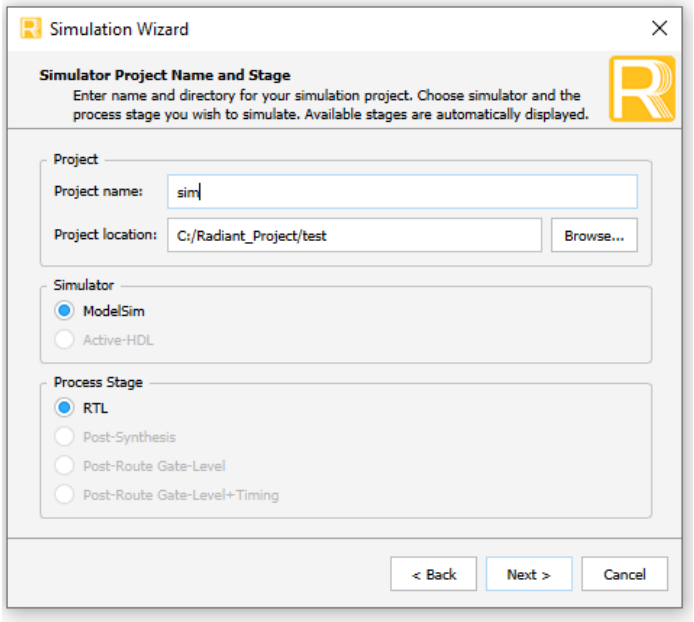

Figure 3.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5.

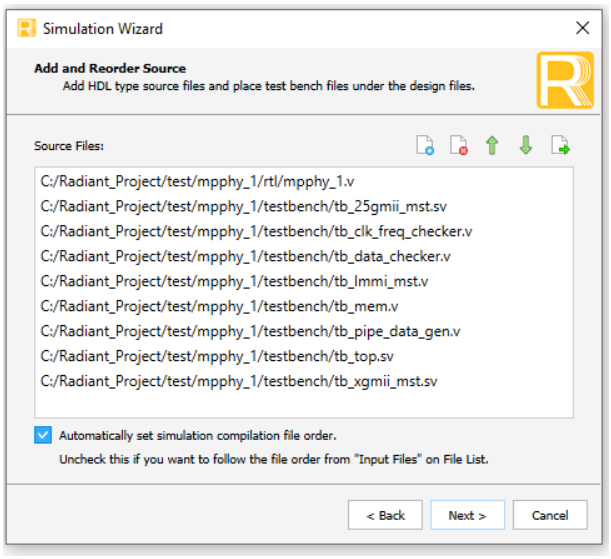

Figure 3.5. Adding and Reordering Source

3. Click **Next**. It shows the **Summary** window.
4. Click **Finish** to run the simulation.

Note: It is necessary to follow the procedure above until it is fully automated in the Lattice Radiant Software Suite.

The results of the simulation in our example are provided in Figure 3.6.

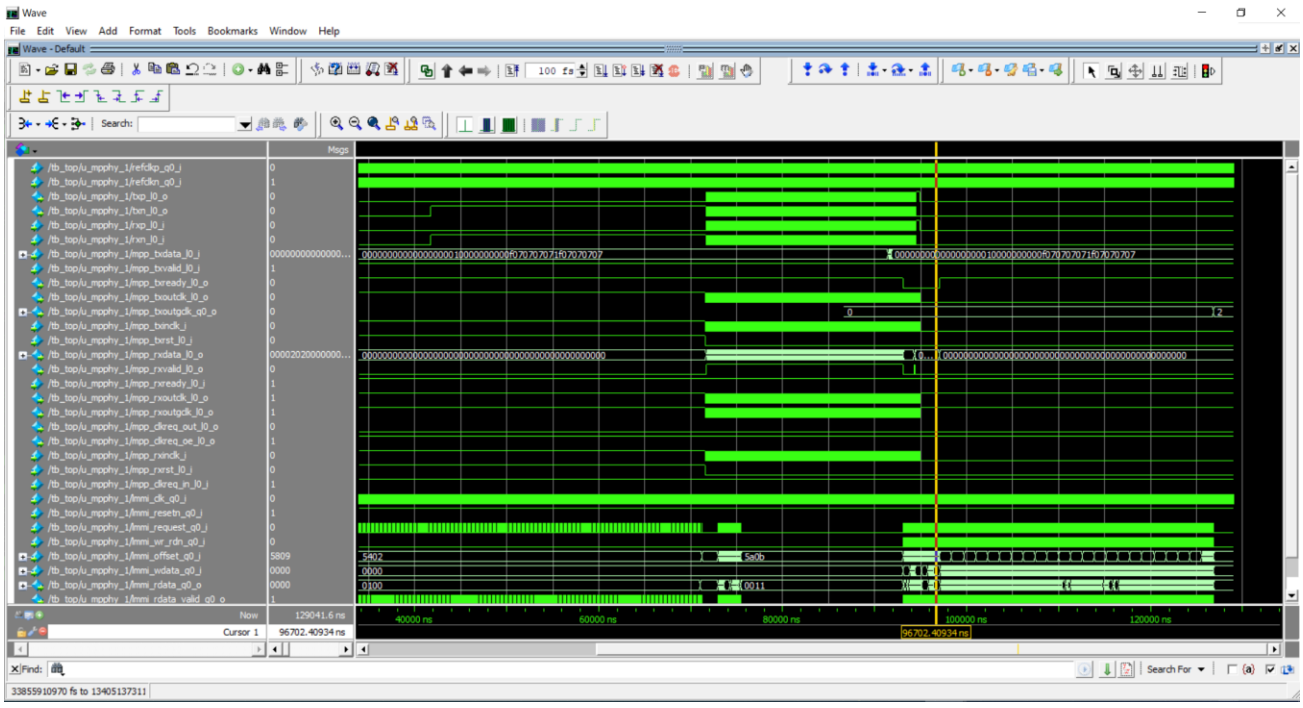

Figure 3.6. Simulation Waveform

3.4. Constraining the IP

It is the responsibility of the user to provide proper timing and physical design constraints to ensure that the design meets the desired performance goals on the FPGA. The content of the following IP constraint file can be added to the user design constraints:

<Instance_Path>/<Instance_Name>/constraints/<Instance_Name>.ldc

The above constraint file has been verified with the IP instantiated directly in the top-level module. The constraint in this file can be modified given a complete understanding of the effect of the constraint.

To use this constraint file, copy the content of the <Instance_Name>.ldc to the top-level design constraint for post-synthesis.

Refer to Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constraint the design.

References

- For complete information on the Lattice Radiant Project-Based Environment, Design Flow, Implementation Flow, and Tasks, as well as on the Simulation Flow, refer to the Lattice Radiant software user guide.
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Avant-G web page
- Avant-X web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plan.

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 1.0, December 2023

Section	Change Summary
All	• Replaced LAV-AT-500G with LAV-AT-G70 • Replaced LAV-AT-500X with LAV-AT-X70
Disclaimers	Updated with the latest disclaimers.
Inclusive Language	Newly added section.
Functional Description	• Updated Figure 2.1. MPPHY Module Block Diagram to Figure 2.4. PCS 64B66B TX/RX Path. • Added the Reset Control Scheme with RCM Enable section. • Updated Table 2.5. RCM IP Reset Scheme Support. • Removed table notes no.2 and no.3 from Table 2.5. RCM IP Reset Scheme Support. • Updated Table 2.6. MPPHY Module Signal Description. • Updated the title of Figure 2.7 HSSI PHY Initialization Sequence. • Updated Table 2.9. Attributes Table and Table 2.10. Attributes Descriptions to add RCM Enable and User Interface Type. • Updated Table 2.11. Expected rxvalid_o assertion for Disabled RCM block in RTL simulation to Table 2.13. Expected rxvalid_o assertion for Enabled RCM block using AXI Interface in RTL simulation. • Removed the Warm Reset (PCS Reset) (internal use only) section. • Updated Figure 2.7 HSSI PHY Initialization Sequence and Figure 2.8. PMA Tx and Rx Reset. • Added the PHY Reset Sequence with RCM Enabled section.
IP Generation and Evaluation	Added the Constraining the IP section
References	Newly added section.

Revision 0.80, June 2023

Section	Change Summary
All	Preliminary Release

www.latticesemi.com