MPCS Module

IP Version: 1.8.0

User Guide

FPGA-IPUG-02118-1.8

June 2025

Disclaimers

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

Inclusive Language

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

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
BER	Bit Error Read
CTC	Clock Tolerance Compensation
EPCS	External Physical Coding Sublayer
FPGA	Field Programmable Gate Array
FOM	Figure of Merit
LMMI	Lattice Memory Mapped Interface
MPCS	Multiple-Protocol Physical Coding Sublayer
PIPE	PHY Interface for PCI Express
PMA	Physical Medium Assignment
RTL	Register Transfer Level

1. Introduction

1.1. Overview of the IP

The MPCS module implements many common functionalities of PCS layer defined in different serial I/O protocols. The MPCS module implements up to four channels per Quad of embedded SERDES with associated Physical Coding Sublayer (PCS) logic.

Each channel of PCS contains dedicated transmit and receive logic for high-speed, full duplex serial data transfer at data rates up to 10.3125 Gbps.

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

1.2. Quick Facts

Table 1.1 presents a summary of the MPCS module.

Table 1.1. Quick Facts

IP Requirements	Supported Devices	MachXO5T™-NX (LFMXO5-100T, LFMXO5-55T), CertusPro™-NX (LFCPNX-100, LFCPNX-50, UT24CP100)
	IP Changes	For a list of changes to the IP, refer to the MPCS Module Release Notes (FPGA-RN-02088).
Resource Utilization	Supported User Interface	LMMI (Lattice Memory Mapped Interface)
Design Tool Support	Lattice Implementation	IP Core v1.8.0 – Lattice Radiant software 2025.1
	Synthesis	Lattice Synthesis Engine
		Synopsys® Synplify Pro for Lattice
	Simulation	For a list of supported simulators, see the Lattice Radiant Software User Guide.

1.3. Features

Key features of the MPCS module include:

- Supports up to four Channels per Quad.
- LFMXO5 device supports one quad. Other devices can support up to two quads.
- Supports XAUI, SGMII, and other standards.
- Single Channel PCS Functionalities
  - 270 Mbps to 10.3125 Gbps per channel
  - Word Alignment and link Synchronization FSM
  - Supports popular 8b10b-based packet protocols
  - Supports user-specified generic 8b10b mode
  - Supports Per Channel configuration
- Multiple Channel Alignment
  - Minimized TX lane-lane skew
  - RX multiple lane De-skew
- Multiple Protocol Compliant Clock Tolerance Compensation (CTC) Logic
  - Compensates for frequency differential between reference clock and received data rate.
  - Allows user-defined skip pattern of 1, 2, or 4 bytes in length.
- Integrated Loopback Modes for System Debugging

1.4. Licensing and Ordering Information

The MPCS module is provided at no additional cost with the Lattice Radiant software.

1.5. Naming Conventions

1.5.1. Nomenclature

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

1.5.2. 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

2. Functional Description

2.1. MPCS Module Architecture Overview

The MPCS module supports many common functionalities of PCS layer defined in different serial I/O ports. It is controlled by a number of registers that can be dynamically reconfigured by user logic or configuration module through LMMI ports.

Each channel of the MPCS module contains dedicated transmit and receive logic for high-speed, full-duplex serial data transfer at data rates up to 10.3125 Gbps.

The MPCS module supports the following two modes, as shown in Figure 2.2:

- MPCS mode – uses MPCS interface to configure either 8b10b MPCS or 64b66b MPCS to support an array of popular data protocols including 1KBASEX, 10GE, COAXPRESS, DP, eDP, G8B10B, JESD204B, SGMII, SLVS_EC, QSGMII, and XAUI.
- EPCS mode – uses EPCS interface, allows a direct interface from PMA to fabric. The protocol-based logic can be fully or partially bypassed in a number of configurations to provide you the flexibility in implementing high-speed data interface.

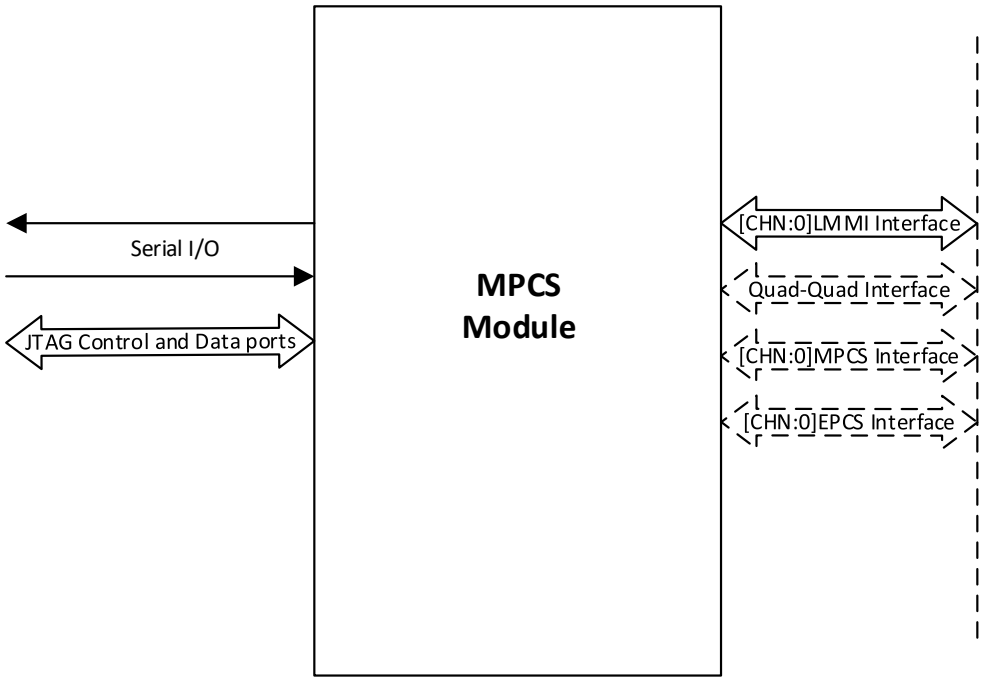

Figure 2.1. MPCS Module Block Diagram

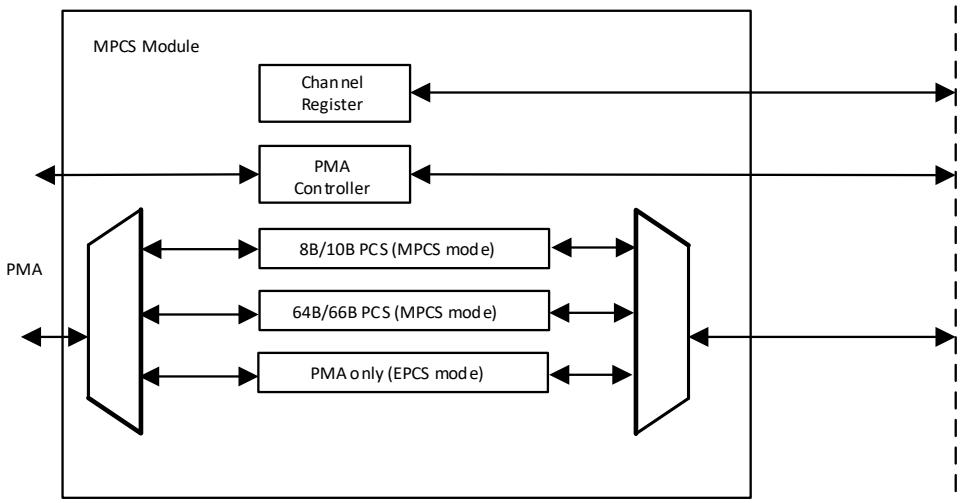

Figure 2.2. MPCS Modes Block Diagram

2.2. Clocking

2.2.1. PLL and Divider Settings

The MPCS module has predefined PLL and Divider Settings for each protocol. You need the following formulas if you want to manually set the PLL and Divider Settings to get your desired output clock frequency. These settings are accessible to you when you are using the EPCS mode and G8B10B protocol.

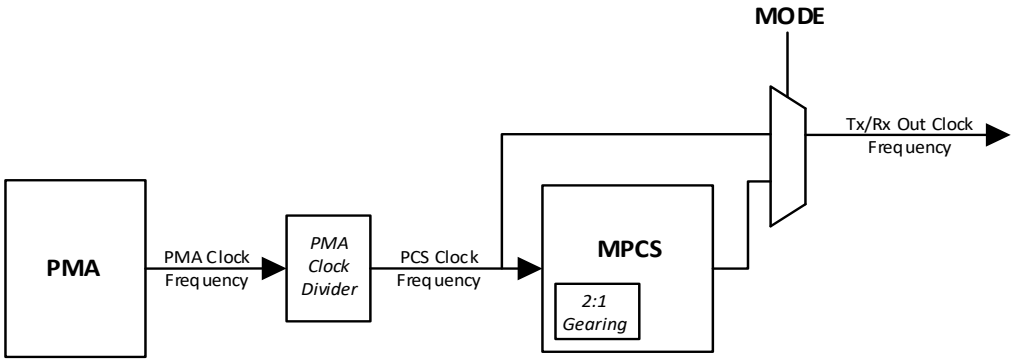

Figure 2.3. PMA and MPCS Clocking Diagram

Data Rate = User Input

Ref Clk Freq = User Input

PLL M Setting = User Input

$$PLL\ F\ Setting = \frac{PMA\ Clock\ Frequency}{Ref\ Clk\ Freq}$$

$$PLL\ N\ Setting = \frac{Bus\ Width}{PMA\ Clock\ Divider}$$

$$PMA\ Clock\ Frequency = \frac{Data\ Rate}{N}$$ (Output clock frequency from PMA)

$$PCS\ Clk\ Freq = \frac{PMA\ Clock\ Frequency}{PMA\ Clock\ Divider}$$ (Input clock frequency to MPCS core)

$$Fout\_clk = \frac{PCS\ Clk\ Freq}{2{:}1\ Gearing}$$ (Output clock frequency from MPCS core)

2.2.2. Clock Generation

The MPCS module generates output clocks mpcs_tx/rx_out_clk_o (when in MPCS mode) or epcs_tx/rx_out_clk_o (when in EPCS mode) with clock frequencies based on Fout_clk in Table 2.1.

Table 2.1. Data Rate and Reference Clock Frequency

Protocol		Data Rate (Gbps)	Fout_clk (MHz)	Bus Width	2:1 Gearing
QSGMII		5	125	20	ENABLED
XAUI		3.125	156.25	10	ENABLED
SGMII		1.25	125	10	DISABLED
SLVS_EC	Grade3	5	125	20	ENABLED
	Grade2	2.5	125	10	ENABLED
	Grade1	1.25	125	10	DISABLED
COAXPRESS	—	6.25	156.25	20	ENABLED
	—	5	125	20	ENABLED
	—	3.125	156.25	10	ENABLED
	—	2.5	125	10	ENABLED
	—	1.25	125	10	DISABLED
DP/eDP	HBR3	8.1	135	20	ENABLED
	HBR2	5.4	135	20	ENABLED
	HBR	2.7	135	10	ENABLED
	RBR	1.62	108	10	DISABLED
10GE	—	10.3125	161.1328125	16	DISABLED
JESD204B	—	8.192	204.8	20	ENABLED

2.2.3. Reference Clock Source Selection

The MPCS module provides dynamic selection of reference clocks. To use this feature, assert the use_refmux_i port to enable the multiplexer that is used for dynamic switching. The available reference clocks are only connected to DIFFCLKIO_CORE and PLL and are not directly connected to fabric.

For applications with more than four lanes, the reference clock source from sd_ext_0/1_refclk _i must be used.

When clock from GPLL output is used as the reference clock for CertusPro-NX SERDES/PCS, the reference clock to the GPLL is assigned to the dedicated GPLL input pad. However, the GPLL output jitter may not meet system specifications at higher data rates. Reference clock source from GPLL output is not recommended in jitter-sensitive applications.

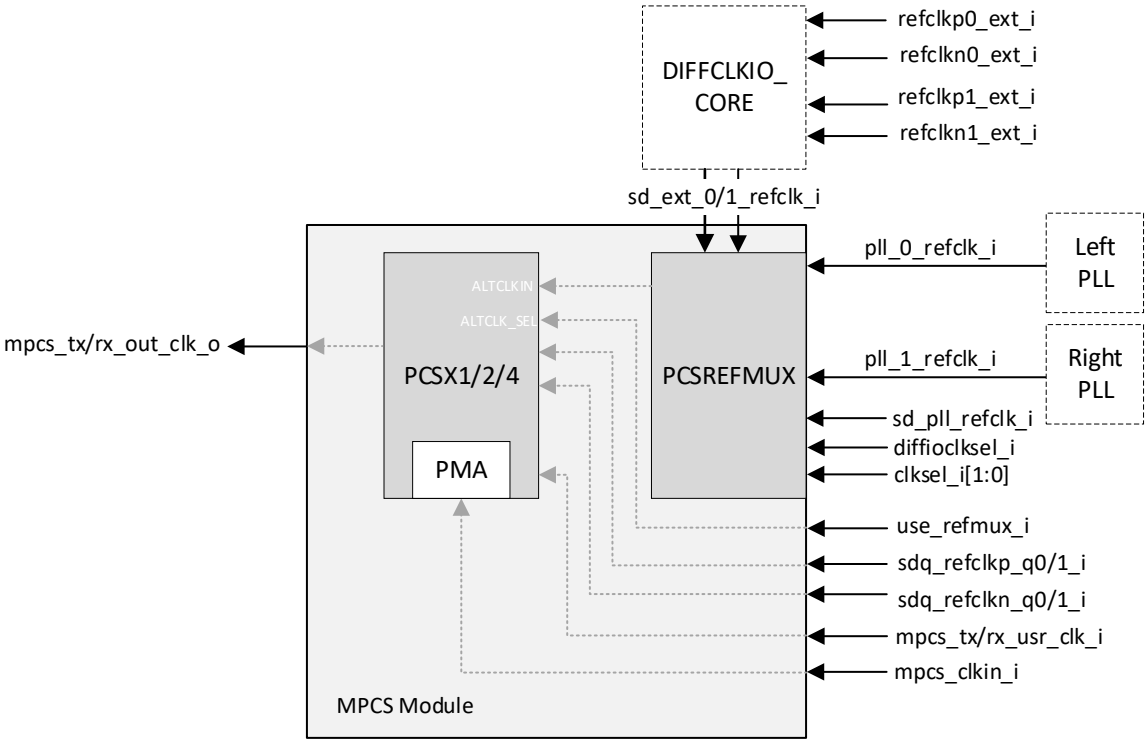

Figure 2.4. Reference Clock Dynamic Selection Block Diagram

The following are the values of *diffioclksel_i* and *clksel_i* ports of PCSREFMUX to properly set the clock selection:

- diffioclksel_i
  1'b0 – sd_ext_0_refclk
  1'b1 – sd_ext_1_refclk
- clksel_i
  2'b00 – pll_0_refclk_i
  2'b01 – pll_1_refclk_i
  2'b10 – output from DIFFCLKIO_CORE
  2'b11 – sd_pll_refclk_i

2.2.4. Two-Quad Clock Connection

For *Number of Lanes* > 4, two quads are instantiated in the design. The output alignment clocks (tx/rx_lalign_clk_out_o) from Quad[0] drive these two quads while the output clocks from Quad[1] are floating and internally gated, as shown in Figure 2.5.

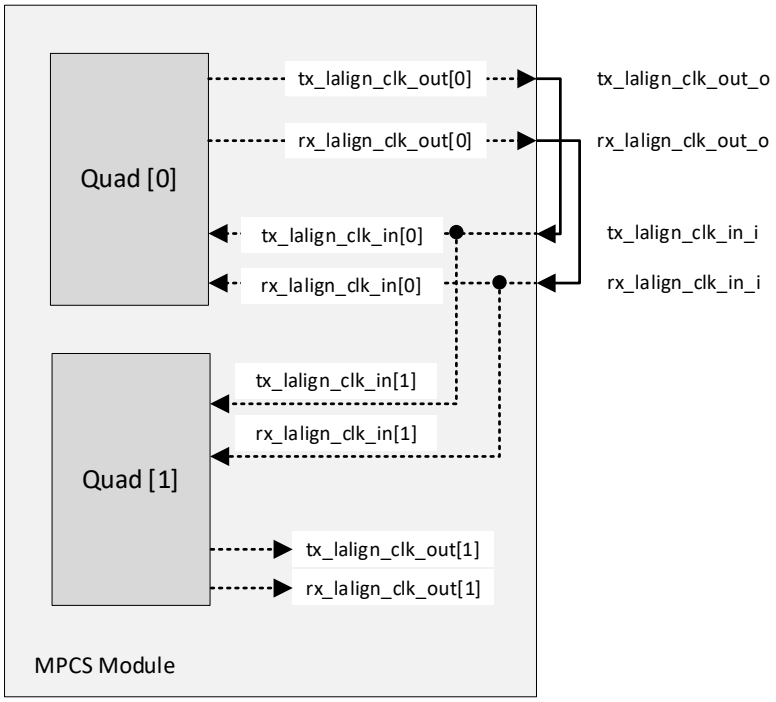

Figure 2.5. Two-Quad Clock Connection Block Diagram

2.2.5. 64b/66b Clocking

Two clocking modes are available for 64b/66b PCS. The modes can be controlled by setting the reg83.pcs_64b66b_nofpll. Figure 2.6 and Figure 2.7 show the detailed clock requirements. This can be controlled by setting the *Use External PLL* or the reg83.pcs_64b66b_nofpll.

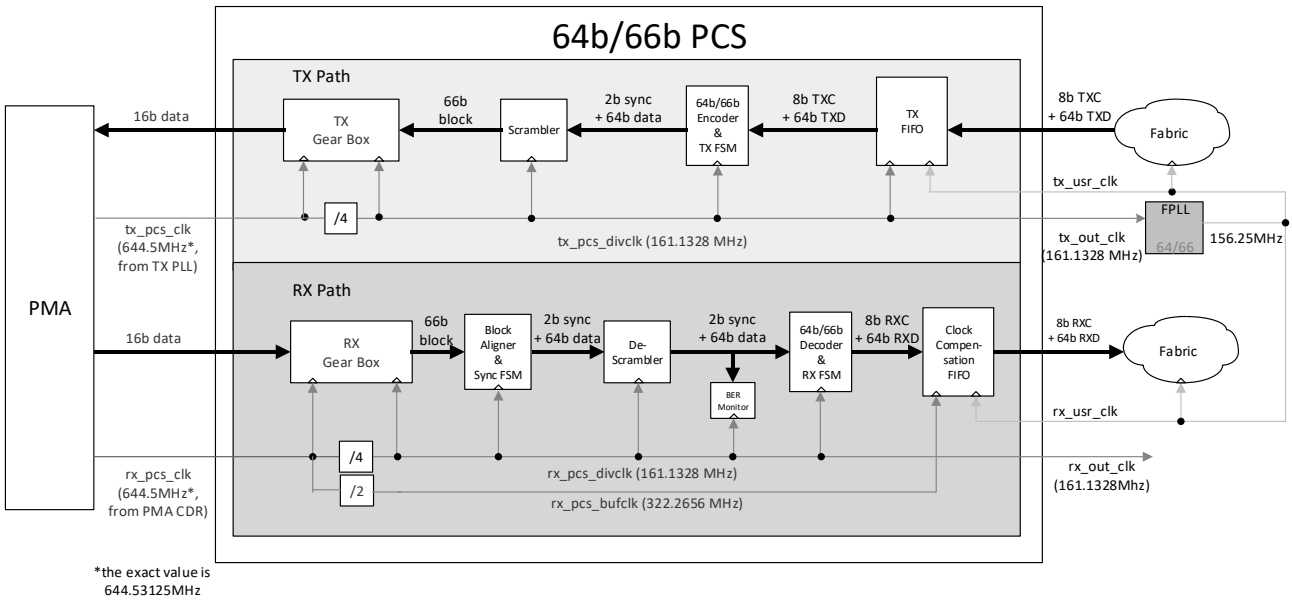

Figure 2.6. 64b/66b PCS Loopback with Fabric PLL Block Diagram

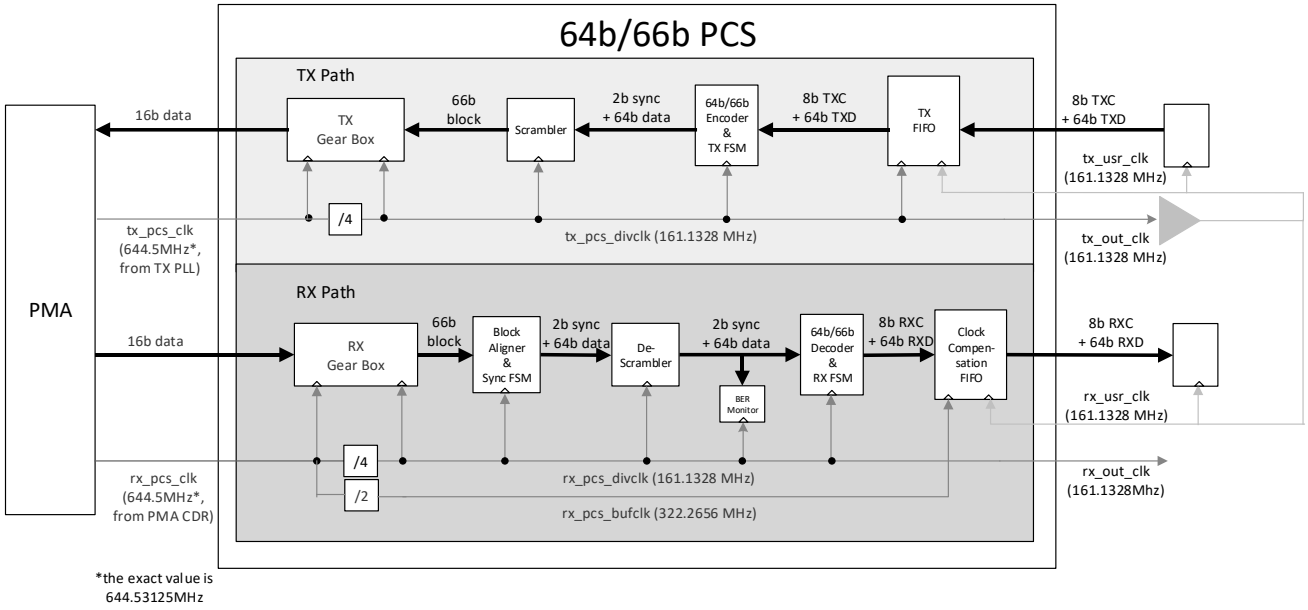

Figure 2.7. 64b/66b PCS Loopback without Fabric PLL Block Diagram

2.3. Reset

2.3.1. Reset Control for Data Transmission

To begin PCS transaction, the PCS driver provides the required clocks with correct frequency and sends valid data to the MPCS module. Set the PMA PLL settings to get the desired output clock frequency. After PLL lock, the mpcs/epcs_ready_o_* signal is asserted indicating that the calibration is complete. This is followed by the assertion of the mpcs/epcs_phyrdy_o_* signal indicating that the MPCS module is ready to transmit data. The MPCS module requires active low transmit and receive resets (mpcs/epcs_tx_pcs_rstn_i_* and mpcs/epcs_rx_pcs_rstn_i_* signals in Figure 2.8). It is recommended for the receive active low reset signal (mpcs/epcs_rx_pcs_rstn_i) to wait for the assertion of the mpcs/epcs_phyrdy_o_* signal.

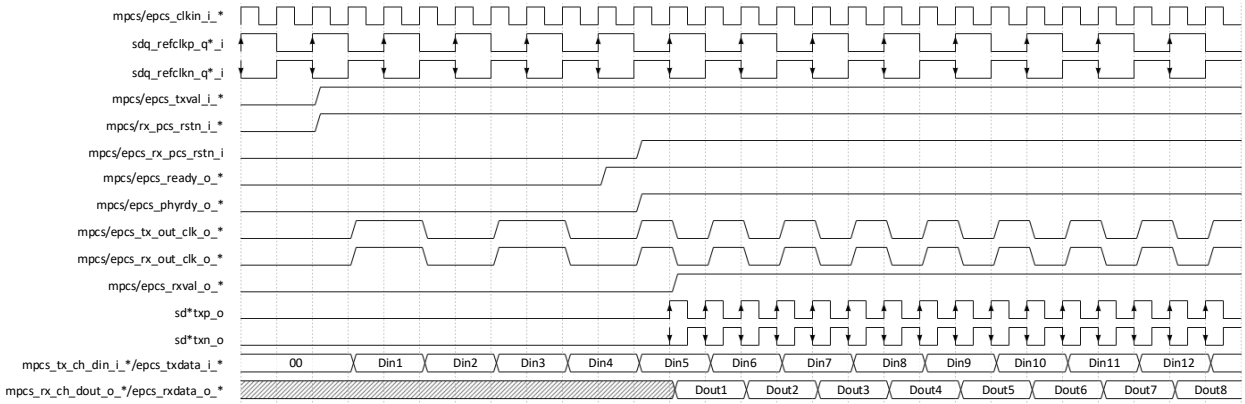

Figure 2.8. Example Reset for PCS Data Transmit and Receive

2.4. MPCS Modes

MPCS Modules supports two MPCS modes as follows:

- MPCS mode – This mode is implemented using either:
  - PMA and 8b/10b PCS to fabric or
  - PMA and 64/66b PCS to fabric
- EPCS mode – This mode is implemented from PMA directly to fabric.

2.4.1. 8b/10b PCS

The 8b/10b of MPCS module implements the most common functionalities required by 8b/10b PCS. The functions are as follows:

- 8b/10b encoding/decoding
- 10b code word boundary detection and alignment
- Clock Frequency Difference Compensation
- Lane-to-lane Deskew

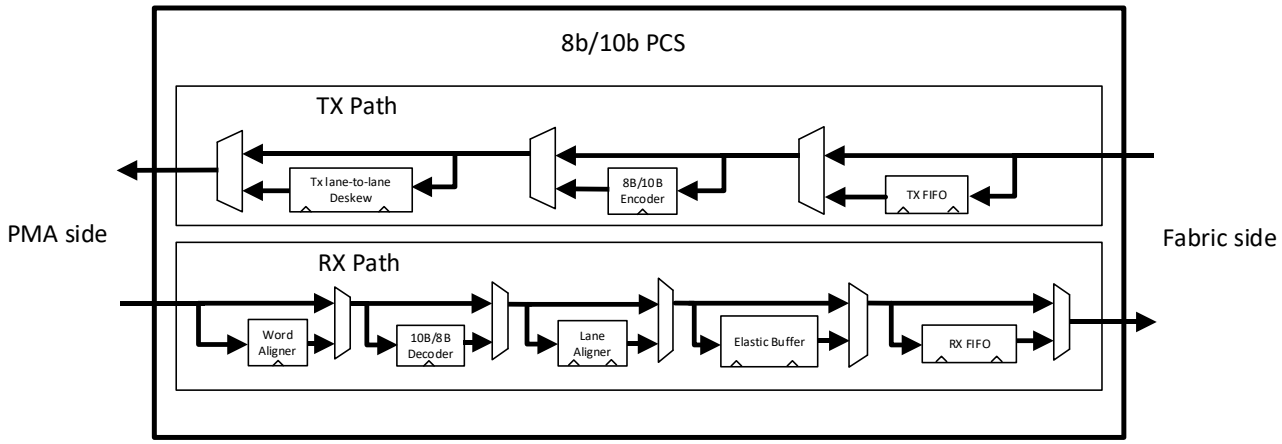

Figure 2.9. 8b/10b PCS Channel Block Diagram

The 8b/10b PCS submodules and how to enable/disable them are explained as follows:

- TX FIFO – can be optionally bypassed by setting TX FIFO and reg10.tx_fifo_dis.
  The TX FIFO submodule serves two purposes:
  - Provides user logic input data with 2:1 gearing to internal MPCS data path.
  - Provides clock phase compensation FIFO to ease MPCS-Fabric interface timing closure.

  The features of TX FIFO are as follows:
  - Converts 4-byte data to 2-byte using 2:1 gearing.
  - Converts 2-byte data to 1-byte using 2:1 gearing.
  - Resolves clock phase difference between its read and write side.
  - Signals overflow and underflow status when occur.
  - Captures data from fabric using synchronous DFF when TX FIFO is bypassed.
- 8b/10b encoder – can be optionally bypassed by setting the 8b10b encoder and reg10.enc_8b10b_dis.
  The encoder performs the 8-bit to 10-bit code conversion while maintaining the running disparity rules.
  The features of 8b/10b encoder are as follows:
  - Supports 1-byte mode; encodes one 8-bit input data in one clock cycle.
  - Supports 2-byte mode; encodes two 8-bit input data in one clock cycle.
  - Forces disparity function.
  - Inverts disparity function.

Force disparity and invert disparity are controlled by the tx_frcdisp and tx_dispval. The combinations of values are listed in Table 2.2.

Table 2.2. Disparity

mpcs_tx_ch_din_i_0 [43:40] tx_frcdisp	mpcs_tx_ch_din_i_0 [47:44] tx_dispval	Disparity
1	0	Force negative disparity
1	1	Force positive disparity
0	0	Calculated by 8b10b encoder
0	1	Invert current disparity

  - Forces this submodule to output the input 10b data (1-byte mode) or 20b data (2-byte mode).
  - Supports interleaving mode performing 8b10b encoding. In this mode, the current input 8b data is used to calculate a new running disparity. The new value is used as running disparity for encoding the input data after the next data. The data streams are separated into two streams, encoded independently, and merged into one stream. Lastly, the encoded data is transmitted to PMA with the same order as the data enters the 8b/10b encoder.
- TX lane-to-lane deskew – can be optionally bypassed by setting TX lane-to-lane deskew and reg10.tx_pmfifo_dis.
- Word aligner – can be optionally bypassed by setting word alignment and reg30.wa_dis.
  The data is serialized before transmission and then de-serialized at the receiver. The data loses the word boundary of the upstream transmitter during deserialization. The word aligner receives parallel data from the de-serializer and restores the word boundary. To enable alignment, transmitters send a recognizable sequence (usually a comma) periodically. The receiver searches for the comma in the incoming data and when the receiver finds the comma, the receiver moves the comma to a byte boundary so that the received parallel words match the transmitted parallel words.
  The features of word aligner are as follows:
  - Supports automatic alignment mode by setting automatic word alignment and reg30.auto_wa_dis.
  - Supports manual alignment mode.
  - Provides link synchronization machine.
  - Optionally enables or disables the link synchronization state machine.
  - Defines the alignment pattern length as 10-bit or 20-bit The maximum allowed alignment pattern length is 20-bit.
  - Provides programmable primary alignment pattern.
  - Provides programmable secondary alignment pattern.
  - Provides programmable alignment pattern mask code.
  - Supports 1-byte mode and 2-byte mode of internal data bus width.
  - Reports the number of bits slipped in the receiver word aligner.
  - Configures the length of synchronization code.
  - Configures and masks all bits of synchronization code.
  - Provides primary and secondary synchronization codes separately.
  - Supports both 10b mode (bypassing 8b10b encoder) and 8b mode (after 8b10b decoding) of input data using the sync_det FSM.
- 10b/8b decoder – can be optionally bypassed by setting 8b10b decoder and reg20.dec_8b10b_dis.
  The features of the 10b/8b decoder are as follows:
  - Supports 1-byte mode; decodes one 10b data in one clock cycle.
  - Supports 2-byte mode; decodes two 10b data in one clock cycle.
  - Detects running disparity error.
  - Detects invalid code.
  - Detects running disparity output.
  - Supports interleaving mode when performing 8b/10b decoding. In this mode, use the current input 8b data to calculate a new value of running disparity. Use the new value as running disparity for decoding the input data after the next data. This function is symmetric to 8b/10b encoder interleaving mode.

- Lane aligner – can be optionally bypassed by setting lane alignment and reg50.lalign_en.
- Elastic buffer – can be optionally bypassed by setting clock frequency compensation and reg60.clk_comp_en.
  The elastic buffer submodule performs clock frequency adjustment between the recovered receive clock domain and the local system clock domain. Inserting or deleting bytes at position where SKIP pattern is detected, without causing loss of packet data, performs clock compensation. A 32-byte elasticity FIFO temporarily buffers coming data from recovered receive clock domain and transfers the data to local system clock domain.
  The features of elastic buffer are as follows:
  - Matches the programmable SKIP pattern as provided in Figure 2.10. The SKIP pattern usually has the following format: a COM byte optionally followed by 1 to 3 SKP bytes. Registers configure all bytes (Byte 0 to 3).

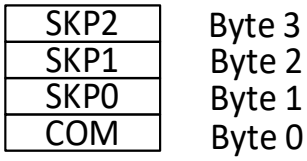

Figure 2.10. SKIP Pattern Format

  - Sets the length of SKIP pattern to 1, 2, or 4 bytes.
  - Provides two SKIP patterns: primary and secondary SKIP.
  - Provides a mask code to allow partially matching the SKIP pattern.

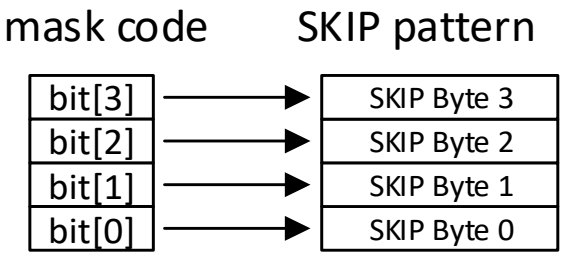

Figure 2.11. Skip Pattern Mask Code

  - Supports both 10b data mode (bypassing 8b10b decoder) and 8b data mode (after 8b10b decoding).
  - Guarantees minimum number of bytes between packets after SKIP deletion.
  - Supports programmable high-water line and low water line.
  - Observes deletion/insertion of SKIP pattern using user logic.
  - Signals the elastic FIFO overrun and underrun status.
- RX FIFO – can be optionally bypassed by setting RX FIFO and reg20.rx_fifo_dis.
  The RX FIFO submodule serves two purposes:
  - Performs 1:2 gearing on MPCS data before forwarding to fabric.
  - Eases MPCS-fabric interface timing closure using clock phase compensation FIFO.

  The features of the RX FIFO are as follows:
  - Converts 2-byte data to 4-byte using 1:2 gearing.
  - Converts 1-byte data to 2-byte using 1:2 gearing.
  - Optionally enables byte shifting feature and reports if byte shifting occurs This submodule can put the word alignment pattern (usually the comma) to byte_0 (LSByte) of the 2-byte or 4-byte data bus.
  - Resolves clock phase difference between read and write clocks. FIFO overflow or underflow may occur if the write and read clocks have frequency difference.
  - Works as synchronous DFF to launch data to Fabric when the RX FIFO is bypassed.
  - Optionally applies the common clock and centralized control logic to the RX FIFO on each lane in multiple-channel mode to minimize the lane-to-lane skew introduced by the uncertain latency of the FIFO.

2.4.2. 64b/66b PCS

The 64b/66b PCS of the MPCS module implements 10GBASE-R PCS defined by the IEEE802.3 protocol. The functional sub-blocks are listed as follows:

- TX path
  - Encodes 72b XGMII data (64b TXD + 8b TXC) to 66b packet (2b header + 64b body) using packet body encoding.
  - Checks the validity of the 64-bit data from the MAC layer and ensures proper block sequencing using transmit FSM.
  - Supports packet body scrambler.
  - Supports PCS-PMA gearbox (66b to 16b conversion).
  - Supports test pattern generation (PRBS and square wave).
  - Supports loopback mode.
- RX path
  - Supports PMA-PCS gearbox (16b to 66b conversion).
  - Aligns (delimits) 66b packet and locks state machine.
  - Supports packet body descrambler.
  - Decodes 66b packet to 72b XGMII data (64b RXD + 8b RXC) using packet body decoding.
  - Supports receive FSM.
  - Supports test pattern compare and error counting.
  - Monitors BER in normal mode (disabled in test mode).
  - Supports loopback mode.
  - Compensates up to a ±100 ppm clock difference between the remote transmitter and the local receiver (inserts idles, deletes idles, or deletes sequence ordered sets) using clock compensation.

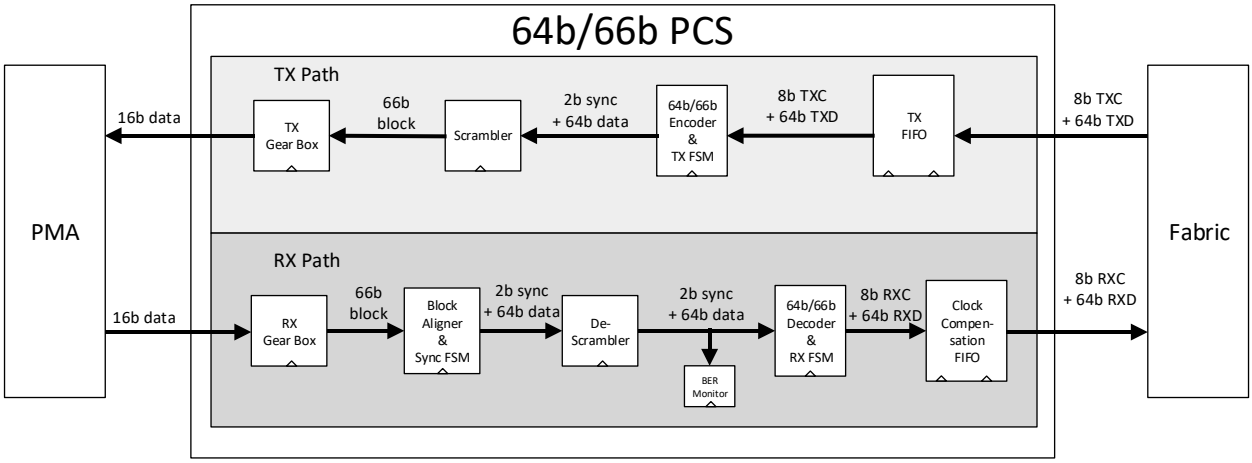

Figure 2.12. 64b/66b PCS Channel Block Diagram

The MPCS module only supports 64b/66b PCS when the protocol is 10GE.The 64b/66b PCS submodules and how to enable or disable them are explained as follows:

- TX FIFO – can be optionally bypassed by setting the TX FIFO.
  This asynchronous FIFO is used to adapt TX path clock frequency and phase difference between 64b/66b PCS and user logic. The user logic can monitor the FIFO Almost Full and FIFO Almost Empty signals. When FIFO Almost Full signal is detected high, to avoid FIFO overflow, user logic pauses FIFO by writing immediately, and resumes writing until the status signal returns low. The high level of FIFO Almost Empty indicates that user logic writes the FIFO immediately when FIFO underflow.
  The features of TX FIFO are as follows:
  - Reports the following FIFO status in mpcs_tx_fifo_st_o:
    - Almost Full
    - Almost Empty
    - FIFO Overflow
    - FIFO Underflow
  - Sets Almost Full and Almost Empty water line using TX FIFO Almost Full and TX FIFO Almost Empty, respectively. The water line is also configurable using mpcs registers reg81.tx_fifo_af and reg82.tx_fifo_ae.
- 64b/66b encoder – can be optionally bypassed by setting 64b66b encoder and reg80.enc_64b66b_dis.
  This submodule encodes 64-bit XGMII data and 8-bit XGMII control into 10GBASE-R 66-bit control or data blocks in accordance with Clause 49 of the IEEE802.3-2008 specification.
  The features of 64b/66b encoder are as follows:
  - Directly sends the 66-bit input data to the output of this submodule in force data mode.
  - Encodes 64-bit XGMII data and 8-bit XGMII control into 10GBASE-R 66-bit packet if not bypassed.
  - Implements the TX FSM in accordance with the IEEE802.3-2008 specification.
  - Registers and accesses the coding of FSM using register access interface for debug purpose.
- Scrambler – can be optionally bypassed by setting 64b66b scrambler and reg80.src_64b66b_dis.
  This submodule scrambles the 64-bit block payload data using x58 + x39 +1 polynomial specified by the IEEE802.3-2008 specification.
  The features of scrambler are as follows:
  - Scrambles the entire 64-bit payload of a 66-bit block, except for the two sync header bits.
  - Directly sends the 66-bit input data to the output of this submodule if bypassed.
- TX gear box – cannot be bypassed.
  This submodule adapts between the 66-bit width of 64b/66b block and the 20-bit width of PMA data bus. This submodule also generates bit sequence of square wave and PRBS pattern and sends them to PMA for transmitter or receiver tests.
  The features of the TX gear box include:
  - Converts the 66-bit 10GBASE-R block into 16-bit PMA data.
  - Generates square wave pattern.
  - Generates PRBS9 and PRBS31 pseudo-random bit sequence.
- RX gear box – cannot be bypassed.
  This submodule adapts between the 66-bit width of 64b/66b block and the 16-bit width of PMA. This submodule also performs PRBS32 pattern checking.
  The features of RX gear box are as follows:
  - Converts 16-bit PMA data into 66-bit data block (without block aligning).
  - Checks the PRBS31 pattern and counts bit errors.
- Block aligner – can be optionally bypassed by setting block aligner and reg83.balign_64b66b_dis.
  This submodule determines the block boundary of a 66-bit word received from the Rx gearbox. The incoming 66-bit data stream is slipped one bit at a time until the required number of valid synchronization header (bit 0 and bit 1) is detected in the received data stream.
  The features of block aligner are as follows:
  - Implements block lock FSM defined in Clause 49 of IEEE802.3-2008.

- Shifts the incoming data stream and determines the block boundary.
- Reports the bit number of shifting through the status register.
- Drives mpcs_rx_blk_lock_o high to signal fabric when block lock is achieved.

- Descrambler – can be optionally bypassed by setting *64b66b Descrambler* and reg83.descr_64b66b_dis.
  The features of Descrambler are as follows:
  - Descrambles the entire 64-bit payload of a 66-bit block, except for the two sync header bits.
  - Checks the pseudo-random test-pattern and counts the mismatch error.
- BER monitor – cannot be bypassed.
  The 10GBASE-R BER monitor is implemented in accordance with the 10GBASE-R protocol. After block lock synchronization is achieved, the BER checker starts to count the number of invalid synchronization headers within a 125-µs period.
  The features of BER monitor are as follows:
  - Provides BER checker status signal mpcs_rx_hi_ber_o indicating a high bit error rate condition, if more than 16 invalid synchronization headers are observed in a 125-µs period.
  - Counts the number of times the BER FSM has entered the "BER_BAD_SH" state and "RX_E" state.
- 64b66b decoder – can be optionally bypassed by setting the 64b66b decoder and reg83.dec_64b66b_dis.
  This submodule reverses the 64b/66b encoding process. The decoder block also contains a state machine (RX SM) designed in accordance with the IEEE802.3-2008 specification.
  The features of the 64b66b decoder are as follows:
  - Converts 10GBASE-R 66-bit packet to 64-bit XGMII data and 8-bit XGMII control.
  - Implements the RX FSM in accordance with the IEEE802.3-2008 specification.
  - Performs functions such as sending local faults to the Media Access Control (MAC)/Reconciliation Sublayer (RS) under reset and substituting error codes when the 10GBASE-R and 10GBASE-KR PCS rules are violated.
- Clock compensation FIFO – can be optionally bypassed by setting the RX FIFO and reg83.ctc_64b66b_dis.
  This submodule compensates clock frequency ± 100 PPM difference (up to one clock edge difference every 5000 clock periods) in the RX path by monitoring the RX FIFO status and inserting or deleting characters specified by the IEEE802.3-2008 specification if the pre-defined criteria is met.
  The features of the clock compensation FIFO are as follows:
  - Deletes character if the number of data residing in the RX FIFO exceeds the high-water line.
  - Inserts character if the number of data residing in the RX FIFO is less than the low water line.
  - Deletes or inserts idle control character (/I/) according to the following rules:
    - /I/s are inserted or deleted in groups of four (4).
    - /I/s may be added following the idle or ordered sets.
    - /I/s must not be added while data is being received.
    - When deleting /I/s, the first four characters after a /T/ must not be deleted.
  - Deletes the sequence ordered_sets (/O/) according to the following rules:
    - Only Sequence ordered set can be deleted.
    - Deleting only when two consecutive sequence ordered sets have been received and shall delete only one of the two.
    - Signal ordered_sets are not deleted for clock compensation.
  - Configures the high-water line and low water line using the register reg84.ctc_64b66b_high and reg85.ctc_64b66b_low.
  - Reports the insert and delete actions through rx_fifo_del and rx_fifo_add signals. See the Data Bus Sharing and Mapping section for more information.

2.4.3. PMA Only

In this mode, the PMA EPCS data bus is accessed by user logic with very low latency. There are only clock phase compensation logic and 2:1 bus width gearing logic between the PMA and user logic.

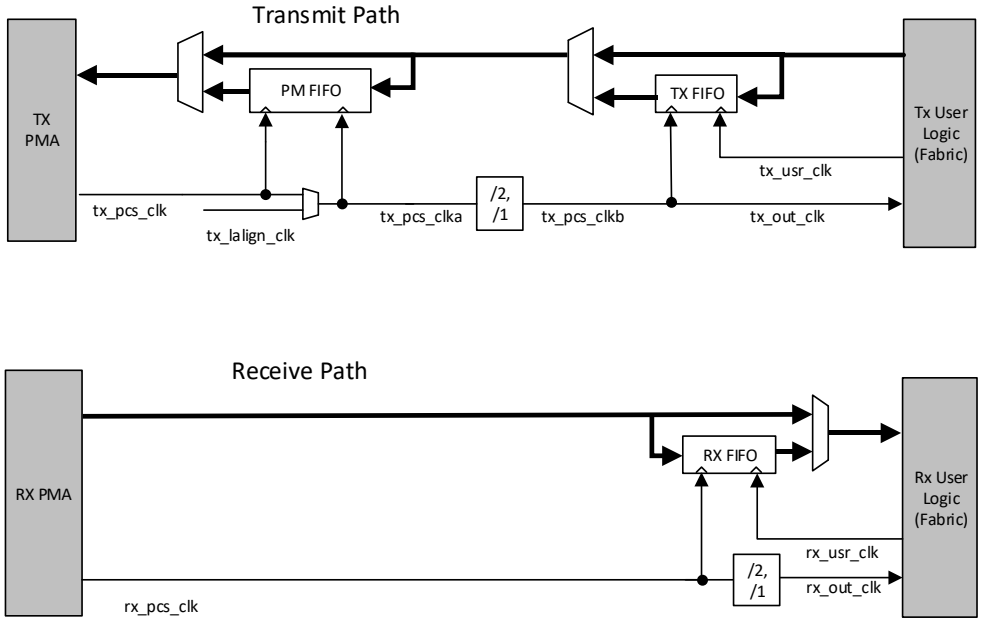

Figure 2.13. PMA Only Mode Block Diagram

The PMA only mode submodules and how to enable or disable them are explained as follow:

- TX FIFO – can be accessed to transmit data to MPCS using epcs_tx_usr_clk_i.
  The TX FIFO submodule serves for clock phase compensation and data bus gearing between user logic and MPCS.
  The features of TX FIFO are as follows:
  - Works as a synchronous register to capture data from user logic and then sends them to the PMA when disabled.
  - Provides the data bus of user logic with two-to-one gearing. With this function enabled, the (2×N)-bit (N=10 or 20) data bus of user logic can be converted to N-bit (N=10 or 20) before sending to the PMA. This function can be optionally enabled.
  - Supports the transmit path data bus width of 10-bit or 20-bit.
- PM FIFO – can be disabled if no channel bonding is required. This is when *Number of Lanes* < 5.
  To minimize lane-to-lane skew in a multiple lane application, the transmit path introduces a common clock, tx_lalign_clk, to drive all lanes that are bonded. In this mode, the PM FIFO must be enabled to eliminate the clock phase difference between PMA clock (tx_pcs_clk) and the common clock (tx_lalign_clk). When disabled, the FIFO works as a group of registers.
- RX FIFO – can be accessed to receive data from MPCS using epcs_rx_usr_clk_i.
  The RX FIFO submodule serves for clock phase compensation and data bus gearing between user logic and MPCS.
  The features of RX FIFO are as follows:
  - Works as a synchronous register when disabled.
  - Converts N-bit (N=10 or 20) data from PMA to (2×N)-bit (N=10 or 20) using the optional RX data bus gearing before sending the data to user logic.
  - Supports the receive path data bus width of 10-bit or 20-bit.

2.4.3.1. Data Bus Description

Table 2.3. PMA Only Mode Data Bus

PMA Data Bus Width	Mode	MPCS Data Bus Width	Gearing	User Logic Data Bus Mapping[1]	Transfer Ordering[2, 3]
20-bit	Mode 1-a	20-bit	1:1	usr_dbus[19:0]	T0: usr_dbus[19:0]
	Mode 1-b		2:1	usr_dbus[39:0]	T0: usr_dbus[19:0] T1: usr_dbus[39:20]
10-bit	Mode 2-a	10-bit	1:1	usr_dbus[9:0]	T0: usr_dbus[9:0]
	Mode 2-b		2:1	usr_dbus[19:0]	T0: usr_dbus[9:0] T1: usr_dbus[19:0]
	Mode 2-c	20-bit	1:1	usr_dbus[19:0]	T0: usr_dbus[9:0] T1: usr_dbus[19:0]
	Mode 2-d		2:1	usr_dbus[39:0]	T0: usr_dbus[9:0] T1: usr_dbus[19:10] T2: usr_dbus[29:20] T3: usr_dbus[39:30]
5-bit	Mode 3-a	10-bit	1:1	usr_dbus[9:0]	T0: usr_dbus[4:0] T1: usr_dbus[9:5]
	Mode 3-b		2:1	usr_dbus[19:10]	T0: usr_dbus[4:0] T1: usr_dbus[9:5] T2: usr_dbus[14:10] T3: usr_dbus[19:15]
16-bit	Mode 4-a	20-bit	1:1	usr_dbus [15: 0]	T0: usr_dbus[15:0]
	Mode 4-b		2:1	usr_dbus[35: 20] usr_dbus[15: 0]	T0: usr_dbus[15:0] T1: usr_dbus[35:20]
8-bit	Mode 5-a	10-bit	1:1	usr_dbus[7:0]	T0: usr_dbus[7:0]
	Mode 5-b		2:1	usr_dbus[17:10] usr_dbus[7:0]	T0: usr_dbus[7:0] T1: usr_dbus[17:10]

Notes:

1. usr_dbus = epcs_txdata_i in TX path; or epcs_rxdata_o in RX path.
2. bit[0] is transmitted/received first.
3. The following are the transfer orders of the Transmit/Receive data bus:
   a. T0: the first PMA clock cycle to capture/launch data in a data block transfer.
   b. T1: the second PMA clock cycle to capture/launch data.
   c. T2: the third PMA clock cycle to capture/launch data.
   d. T3: the fourth PMA clock cycle to capture/launch data.

2.5. MPCS Component Merging and Lane Mapping

2.5.1. MPCS Component Merging

The MPCS module supports up to two quads, each quad has four available lanes. During post-synthesis, PCS lanes are collected and evaluated if the lanes can be merge into one quad. The requirements and guidelines for the PCS merging are as follows:

- Lanes in one quad share reference clocks (sdq_refclkp/n_i) and jtag mode port (acjtag_mode_i).
- Total *Number of Lanes* must not be greater than eight lanes.
- Total number of PCSREFMUX must not be greater than two for LFCPNX-100.
- Different *Protocol*s must have different *Group Name*s.
- *Lane ID* must be in valid range and no conflict.
- *Lane ID* must not be duplicated.

- *Lane ID*s can be merged if multiple *Lane ID*s are placed in the same quad.
  - When in MPCS mode, the *Lane ID* must follow alignment mapping rules, see Table 2.4.

2.5.2. MPCS Lane Mapping

The 10GE protocol is only supported by Lanes 2, 3, 6, and 7, while other protocols support all lanes. The lane mapping for both Quad 0 and Quad 1 is shown in Table 2.4.

Table 2.4. Lane Mapping

	Quad0				Quad1			
Lane ID	0	1	2	3	4	5	6	7
Supported Protocol	1KBASEX, COAXPRESS, DP eDP, JESD204B, QSGMII, SGMII, SLVS_EC, XAUI							
Supported Protocol	—	—	10GE	10GE	—	—	10GE	10GE

2.6. Multiple Data Rate Support

The MPCS module offers up to three PLL and divider settings, these settings are applied to PMA to obtain the proper output clock frequency. For protocols with more than three data rates, such as COAXPRESS, DP, and eDP, the protocols require dynamic switching. The MPCS module has *Rate0, Rate1, and Rate2* settings, and you must assign in the user interface which data rates the module may need for the dynamic switching, as shown in Figure 2.14.

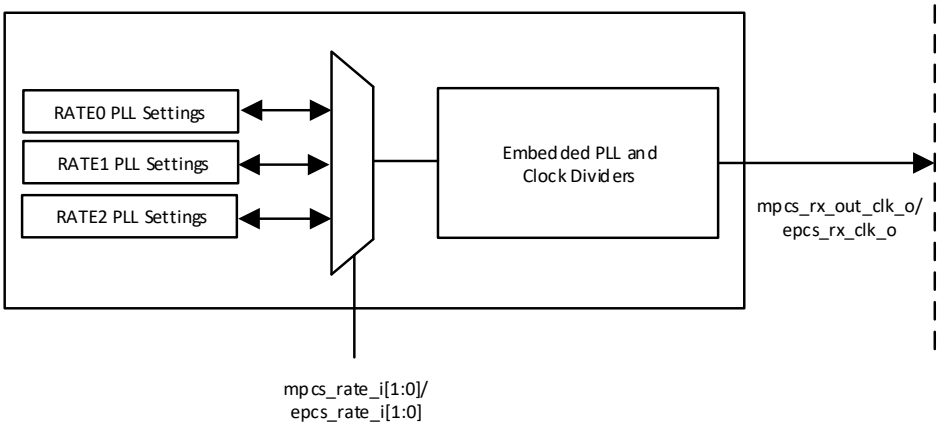

Figure 2.14. Multi-rate Selection Block Diagram

3. Module Parameter Description

The configurable attributes of the MPCS module are shown in Table 3.1. The attributes can be configured through the IP Catalog Module/IP wizard of the Lattice Radiant software.

Wherever applicable, default values are in bold.

3.1. Attributes Table

Table 3.1. Attributes Table

Attribute	Selectable Values	Description	Dependency on Other Attributes
General			
Protocol	**1KBASEX** 10GE COAXPRESS DP eDP G8B10B JESD204B QSGMII SGMII SLVS_EC XAUI	Specifies the selected protocol of the MPCS module.	Active if *Bypass PCS* == Unchecked
Bypass PCS	Checked, **Unchecked**	If enabled, the MPCS module is in EPCS mode.	—
Override TX PCS Mode	Checked, **Unchecked**	Specifies the value of reg00. tx_src_ovrd.	—
Override RX PCS Mode	Checked, **Unchecked**	Specifies the value of reg00. rx_src_ovrd.	—
Number of Lanes	**1**, 2, 4, 6, 8	Specifies the number of lanes of the *Protocol*.	See Table 3.2.
Lane ID	**AUTO**, 0, 1, 2, 3, 4, 5, 6, 7	Specifies the location of the first lane of the PCS instance.	See Table 3.2.
Group Name	—	Automatically takes the component name followed by suffix _PCSGRP.	Uneditable.
Mode	Rx_only, Tx_only, **Rx_and_Tx**	Specifies the selected path mode of the *Protocol*.	See Table 3.2.
Use External PLL	**Checked**, Unchecked	Specifies the option to use a fabric PLL to generate the preferred clock frequency of mpcs_tx/rx_usr_clk_i. See the 64b/66b Clocking section for more details.	Available if *Protocol* == 10GE
Data Rate (Gbps)	(0.625–8.1), **1.25**	Specifies the data rate of the selected *Protocol* in Gbps.	See Table 2.1; Editable in EPCS mode or *Protocol* == G8B10B
Ref Clk Freq (MHz)	See Table 2.1, **125**	Specifies the reference clock frequency of the selected *Data Rate*.	Editable in EPCS mode or *Protocol* == G8B10B
VCO Frequency (GHz)	5–10	Displays the allowable VCO Frequency by the hardware.	Uneditable.

Attribute	Selectable Values	Description	Dependency on Other Attributes
Bus Width	See Table 2.1, **10**	Specifies the user bus width available for the selected *Protocol.*	Editable in EPCS mode or *Protocol* == G8B10B
PMA Clock Divider	1, 2	Specifies the divider that is used to the clock frequency from PMA block.	Available in EPCS mode or *Protocol* == G8B10B
PMA Clock Frequency (MHz)	—, **125**	—	—
PCS Clk Frequency (MHz)	—, **125**	—	—
2:1 Gearing	See Table 2.1, **DISABLED**	Specifies the gearing of the selected *Protocol.*	Uneditable.
Output Clk Frequency (MHz)	—, **125**	Displays the expected output clock of MPCS based on the PLL Settings provided.	—
PLL M Setting	1, **2**, 4, 8	—	Available in EPCS mode or *Protocol* == G8B10B
PLL F Setting	1–6, **5**	—	Available in EPCS mode or *Protocol* == G8B10B
PLL N Setting	—, **10**	—	Available in EPCS mode or *Protocol* == G8B10B
SERDES Setup			
250 ns Timer Base Count	0–255, **31**	Specifies the base count of a 250 ns event based on PMA Clock Frequency. This counter is used by the CDR PLL and PMA Controller for operations such as receiver detects and electrical Idle. This is calculated using ((250 ns / (PMA Clk Period) – 1). In case of non-integer value, the base count must be rounded up.	—
TX Amplitude Ratio	0–**128**	Specifies the Tx Amplitude ratio used by the Tx driver. A value of 128 corresponds to 100% (full voltage) whereas a value of 0 corresponds to 0%. Values higher than 128 are forbidden.	—
Output Termination	**100 Ω**, 150 Ω	Specifies the impedance ratio of the PMA macro with a nominal value of 100 Ω corresponding to multiplication factor of 1, which is encoded 8'd128. A 150 Ω impedance corresponds to 2/3 ratio encoded 8'd85.	—
TX EQ Setting: Pre-Cursor Ratio	**0**–128	Specifies the Tx pre-cursor ratio for gen1 speed used for selecting the de-emphasis of the switching bit versus non-switching bit. A value of 128 corresponds to 100% (full voltage) whereas a value of 0 corresponds to 0%.	—

Attribute	Selectable Values	Description	Dependency on Other Attributes
TX EQ Setting: Post-Cursor Ratio	0–128, **21**	Specifies the Tx post-cursor ratio for gen1 speed used for selecting the de-emphasis of the switching bit versus non-switching bit. A value 128 corresponds to 0%. A -3.5dB corresponds to 8'd21 encoding.	—
Input Termination	**100 Ω**, 150 Ω	Specifies the impedance ratio of the PMA macro with a nominal value of 100 Ω corresponding to multiplication factor of 1, which is encoded 8'd128. A 150 Ω impedance correspond to 2/3 ratio encoded 8'd85.	—
Setting1 Enable	Checked, **Unchecked**	Enables/disables adaptive equalization in data rate 0.	—
Setting1: Adaptive Algorithm	**RL2plus**, SS_LMS	Specifies which equalization algorithm to apply in data rate0.	Editable if *Setting1 Enable* == Checked
Setting2 Enable	**Checked**, Unchecked	Enables/disables adaptive equalization in data rate 1.	—
Setting2: Adaptive Algorithm	**RL2plus**, SS_LMS	Specifies which equalization algorithm to apply in data rate1.	Editable if *Setting2 Enable* == Checked
Setting3 Enable	Checked, **Unchecked**	Enables/disables adaptive equalization in data rate 2.	—
Setting 3: Adaptive Algorithm	**RL2plus**, SS_LMS	Specifies which equalization algorithm to apply in data rate2.	Editable if *Setting3 Enable* == Checked
Setting ½: Preliminary Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting1 Enable* or *Setting 2 Enable* == Checked
Setting ½: Training phase Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting1 Enable* or *Setting 2 Enable* == Checked
Setting ½: Post-phase Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting1 Enable* or *Setting 2 Enable* == Checked
Setting 3: Preliminary Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting 3 Enable* == Checked
Setting 3: Training phase Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting 3 Enable* == Checked
Setting 3: Post-phase Adaptive EQ	ENABLED, **DISABLED**	—	Editable if *Setting 3 Enable* == Checked
PCS Setup			
Invert TX Data Polarity	**NORMAL**, INVERT	Specifies the polarity of data on the transmitted data.	—
TX FIFO	Checked, Unchecked	—	Available if *Protocol* == 10GE
TX FIFO Almost Full	0–15	—	Available if *Protocol* == 10GE Editable if *TX FIFO* == Checked
TX FIFO Almost Empty	0–15	—	Available if *Protocol* == 10GE Editable if *TX FIFO* == Checked
64b66b Encoder	ENABLED, DISABLED	Specifies the reset value of reg80.end_64b66b_dis.	Available if *Protocol* == 10GE
Scrambler	ENABLED, DISABLED	Specifies the reset value of reg80.src_64b66b_dis.	Available if *Protocol* == 10GE
Block Aligner	ENABLED, DISABLED	Specifies the reset value of reg83. balign_64b66b_dis.	Available if *Protocol* == 10GE

Attribute	Selectable Values	Description	Dependency on Other Attributes
Descrambler	ENABLED, DISABLED	Specifies the reset value of reg83. descr_64b66b_dis.	Available if *Protocol* == 10GE
64b66b Decoder	ENABLED, DISABLED	Specifies the reset value of reg83. dec_64b66b_dis.	Available if *Protocol* == 10GE
RX FIFO	ENABLED, DISABLED	Specifies whether the RX phase compensation FIFO is enabled or disabled. For more details, refer to register field rx_fifo_dis in Table 5.7.	Available if *Protocol* == 10GE
TX FIFO	**ENABLED**, DISABLED	Specifies whether the TX phase compensation FIFO is enabled or disabled.	Available if *Protocol* != 10GE
8b10b Encoder	**ENABLED**, DISABLED	Specifies whether the 8b10b encoding is enabled or disabled.	Available if *Protocol* != 10GE
TX Lane-to-Lane Deskew	**ENABLED**, DISABLED	—	Available if *Protocol* != 10GE
Invert RX Data Polarity	**NORMAL**, INVERT	Specifies the polarity of data on the received data.	—
Word Alignment	**ENABLED**, DISABLED	Specifies whether the Word Alignment is enabled or not. For more details, refer to register field wa_dis in Table 5.10.	Available if *Protocol* != 10GE Active if *Number of Lanes* > 1
Word Alignment Bit Width	**10BIT_WIDTH**, 20BIT_WIDTH	—	Available if *Protocol* != 10GE
Put the COMMA byte to LSByte	ENABLED, **DISABLED**	Specifies the reset value of reg20. rfifo_com_align.	Available if *Protocol* != 10GE
Automatic Word Alignment	**ENABLED**, DISABLED	Specifies the use of automatic word alignment or use of manual alignment. For more details, refer to register field auto_wa_dis in Table 5.10.	Available if *Protocol* != 10GE Active if *Word Alignment* == ENABLED
Primary Word Alignment Pattern Symbol 1 10B (HEX)	<hex value>, **000**	Primary Pattern Value in hex form. For more details, refer to register field pri_wa_ptn in Table 5.11 to Table 5.13.	Available if *Protocol* != 10GE Active if Word Alignment == ENABLED and Word Alignment Bit Width == 20BIT_WIDTH
Primary Word Alignment Pattern Symbol 0 10B (HEX)	<hex value>, **17C**	Primary Pattern Value in hex form. For more details, refer to register field pri_wa_ptn in Table 5.11 to Table 5.13.	Available if *Protocol* != 10GE Active if Word Alignment == ENABLED
Word Alignment Pattern Mask Code Symbol 1 10B (HEX)	<hex value>, **000**	—	Available if *Protocol* != 10GE Active if Word Alignment == ENABLED and Word Alignment Bit Width == 20BIT_WIDTH
Word Alignment Pattern Mask Code Symbol 0 10B (HEX)	<hex value>, **000**	—	Available if *Protocol* != 10GE Active if Word Alignment == ENABLED
Secondary Word Alignment	**ENABLED**, DISABLED	Specifies whether the Secondary Word Alignment Pattern Matching is enabled or not. For more details, refer to register field sec_waptn_dis in Table 5.10.	Available if *Protocol* != 10GE Active if *Word Alignment* == ENABLED

Attribute	Selectable Values	Description	Dependency on Other Attributes
Secondary Word Alignment Pattern Symbol 1 10B (HEX)	<hex value>, **000**	Secondary Pattern Value in hex form. For more details, refer to register field pri_wa_ptn in Table 5.14 to Table 5.16.	Available if Protocol != 10GE Active if Word Alignment == ENABLED and Word Alignment Bit Width == 20BIT_WIDTH
Secondary Word Alignment Pattern Symbol 0 10B (HEX)	<hex value>, **283**	Secondary Pattern Value in hex form. For more details, refer to register field pri_wa_ptn in Table 5.14 to Table 5.16.	Available if Protocol != 10GE Active if Word Alignment == ENABLED
Use LSByte of the Word Alignment	**ENABLED**, DISABLED	Specifies whether to put the LSByte of the word alignment pattern to [9:0] of the data bus or not. For more details, refer to register field align_2byte_dis in Table 5.10.	Available if *Protocol* != 10GE Active if *Word Alignment* == ENABLED
Use ‘sync_det’ FSM	**ENABLED**, DISABLED	Specifies whether to use sync_det FSM to control the automatic word alignment. For more details, refer to register field syncdet_fsm_dis in Table 5.10.	Available if *Protocol* != 10GE Active if *Word Alignment* == ENABLED
Number of Valid Sync Code Groups	(**3**–255)	The number of valid synchronization code groups or ordered sets in decimal form that sync_det FSM must receive to achieve synchronization state.	Available if *Protocol* != 10GE Active if (*Word Alignment* == ENABLED) and (*Use ‘sync_det’ FSM* == ENABLED)
Number of Bad Code Groups	(3–63), **4**	The number of bad code groups in decimal form received by sync_det FSM to conclude the loss of synchronization.	Available if *Protocol* != 10GE Active if (*Word Alignment* == ENABLED) and (*Use ‘sync_det’ FSM* == ENABLED)
Number of Good Code Groups	(3–255), **4**	The number of continuous good code in decimal form groups received by sync_det FSM to reduce the error count by one.	Available if *Protocol* != 10GE Active if (*Word Alignment* == ENABLED) and (*Use ‘sync_det’ FSM* == ENABLED)
8b10b Decoder	**ENABLED**, DISABLED	Specifies whether the 8b10b decoding is enabled or disabled. For more details, refer to register field dec_8b10b_dis in Table 5.7.	Available if *Protocol* != 10GE
Clock Frequency Compensation	ENABLED, **DISABLED**	Specifies the reset value of reg83. ctc_64b66b_dis.	Available if *Protocol* != 10GE
CTC FIFO	ENABLED, **DISABLED**	Specifies whether the Clock Compensation FIFO is enabled or not.	Available if *Protocol* != 10GE
Use CC Clock port	Checked, **Unchecked**	Enables and uses mpcs/epcs_cc_clk_i.	Available if *Protocol* != 10GE Active if *Enable CTC FIFO* == ENABLED
Skip Pattern Mask Code	(**0**–15)	—	Available if *Protocol* != 10GE
Skip Pattern Length	**1_BYTE**, 2_BYTE, 4_BYTE	—	Available if *Protocol* != 10GE
Primary Skip Pattern Byte 0 (HEX)	<hex value>, **17C**	—	Available if *Protocol* != 10GE
Primary Skip Pattern Byte 1 (HEX)	<hex value>, **00**	—	Available if *Protocol* != 10GE
Primary Skip Pattern Byte 2 (HEX)	<hex value>, **00**	—	Available if *Protocol* != 10GE

Attribute	Selectable Values	Description	Dependency on Other Attributes
Primary Skip Pattern Byte 3 (HEX)	<hex value>, **00**	—	Available if *Protocol* != 10GE
Sync_det Pattern Length	1, **2**, 4	Specifies the length of sync_det patten.	Available if *Protocol* != 10GE
Sync Pattern Alignment	ENABLED, **DISABLED**	Specifies whether the Sync Pattern is enabled or disabled.	Available if *Protocol* != 10GE
Sync Pattern Code	8B_CODE, **10B_CODE**	Specifies the pattern code for sync_det.	Available if *Protocol* != 10GE
Secondary Sync Pattern	ENABLED, **DISABLED**	Specifies whether the Secondary Sync Pattern is enabled or disabled.	Available if *Protocol* != 10GE
Primary Sync_det Pattern Byte 0 (HEX)	<hex value>, **17C**	Specifies the primary pattern Byte 0 in hex form for sync_det.	Available if *Protocol* != 10GE
Primary Sync_det Pattern Byte 1 (HEX)	<hex value>, **00**	Specifies the primary pattern Byte 1 in hex form for sync_det.	Available if *Protocol* != 10GE
Primary Sync_det Pattern Byte 2 (HEX)	<hex value>, **00**	Specifies the primary pattern Byte 2 in hex form for sync_det.	Available if *Protocol* != 10GE
Primary Sync_det Pattern Byte 3 (HEX)	<hex value>, **00**	Specifies the primary pattern Byte 3 in hex form for sync_det.	Available if *Protocol* != 10GE
Secondary Sync_det Pattern Byte 0 (HEX)	<hex value>, **17C**	Specifies the secondary pattern Byte 0 in hex form for sync_det.	Available if *Protocol* != 10GE
Secondary Sync_det Pattern Byte 1 (HEX)	<hex value>, **00**	Specifies the secondary pattern Byte 0 in hex form for sync_det.	Available if *Protocol* != 10GE
Secondary Sync_det Pattern Byte 2 (HEX)	<hex value>, **00**	Specifies the secondary pattern Byte 0 in hex form for sync_det.	Available if *Protocol* != 10GE
Secondary Sync_det Pattern Byte 3 (HEX)	<hex value>, **00**	Specifies the secondary pattern Byte 0 in hex form for sync_det.	Available if *Protocol* != 10GE
Lane Alignment	ENABLED, **DISABLED**	Specifies whether the Lane Alignment is enabled or not.	Available if *Protocol* != 10GE
Input Data Code Mode	**8B_CODE**, 10B_CODE	Specifies the code mode of the input data. For more details, refer to register field lalign_10b in Table 5.40.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Lane Alignment Pattern	**1_BYTE**, 2_BYTE, 4_BYE	Specifies the lane alignment pattern length in byte. For more details, refer to register field lalign_ptn_len in Table 5.40.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Maximum Lane-to-Lane Skew	**NO_SKEW**, 1_SKEW, 2_SKEW, 3_SKEW, 4_SKEW, 5_SKEW, 6_SKEW, 7_SKEW, 8_SKEW, 9_SKEW, 10_SKEW,	Specifies the maximum lane-to-lane skew in byte. For more details, refer to register field max_lskew in Table 5.41.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Primary Lane Alignment Pattern Byte 0 (HEX)	—, **17C**	Specifies the primary pattern Byte 0 in hex form for Lane Alignment.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Primary Lane Alignment Pattern Byte 1 (HEX)	—, **00**	Specifies the primary pattern Byte 1 in hex form for Lane Alignment.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED

Attribute	Selectable Values	Description	Dependency on Other Attributes
Primary Lane Alignment Pattern Byte 2 (HEX)	—, **00**	Specifies the primary pattern Byte 2 in hex form for Lane Alignment.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Primary Lane Alignment Pattern Byte 3 (HEX)	—, **00**	Specifies the primary pattern Byte 3 in hex form for Lane Alignment.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Secondary Lane Alignment	ENABLED, **DISABLED**	Specifies whether Secondary Lane Alignment is enabled or not.	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
Lane Alignment Mask Code	(0, 15), **1**	—	Available if *Protocol* != 10GE Active if *Lane Alignment* == ENABLED
RX FIFO	**ENABLED**, DISABLED	Specifies whether the RX phase compensation FIFO is enabled or disabled. For more details, refer to register field rx_fifo_dis in Table 5.7.	Available if *Protocol* != 10GE
Loopback Mode	8b/10b PCS Near-End Parallel Loopback Mode, 8b/10b PCS Far-End Parallel Loopback Mode, 64b/66b PCS Loopback Path A, 64b/66b PCS Loopback Path C, PMA Near-End Serial Loopback Mode, PMA Far-End Parallel Loopback Mode, **Disabled**	Specifies the preferred loopback. Parallel Loopbacks are only available in MPCS mode.	—
GSR	**ENABLED**, DISABLED	Specifies whether GSR is enabled or not.	—

3.2. Protocol Presetting

Table 3.2. Protocol Presetting

Protocol	Number of Lanes	Lane ID	Mode
10GE	1	AUTO, 2, 3, 6, 7	Rx_and_Tx
1KBASEX	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_and_Tx
	2	AUTO, 0, 2, 4, 6	Rx_and_Tx
	4	AUTO, 0, 4	Rx_and_Tx
	6	AUTO, 0, 2	Rx_and_Tx
	8	AUTO, 0	Rx_and_Tx
COAXPRESS	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_only, Tx_only
	2	AUTO, 0, 2, 4, 6	Rx_only, Tx_only
	4	AUTO, 0, 4	Rx_only, Tx_only
	6	AUTO, 0, 2	Rx_only, Tx_only
	8	AUTO, 0	Rx_only, Tx_only
XAUI	4	AUTO, 0, 4	Rx_and_Tx

Protocol	Number of Lanes	Lane ID	Mode
SGMII	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_and_Tx
	2	AUTO, 0, 2, 4, 6	Rx_and_Tx
	4	AUTO, 0, 4	Rx_and_Tx
	6	AUTO, 0, 2	Rx_and_Tx
	8	AUTO, 0	Rx_and_Tx
SLVS_EC	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_only, Tx_only
	2	AUTO, 0, 2, 4, 6	Rx_only, Tx_only
	4	AUTO, 0, 4	Rx_only, Tx_only
	6	AUTO, 0, 2	Rx_only, Tx_only
	8	AUTO, 0	Rx_only, Tx_only
DP, eDP, JESD204B	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_only, Tx_only, Rx_and_Tx
	2	AUTO, 0, 2, 4, 6	Rx_only, Tx_only, Rx_and_Tx
	4	AUTO, 0, 4	Rx_only, Tx_only, Rx_and_Tx
G8B10B, QSGMII	1	AUTO, 0, 1, 2, 3, 4, 5, 6, 7	Rx_only, Tx_only, Rx_and_Tx
	2	AUTO, 0, 2, 4, 6	Rx_only, Tx_only, Rx_and_Tx
	4	AUTO, 0, 4	Rx_only, Tx_only, Rx_and_Tx
	6	AUTO, 0, 2	Rx_only, Tx_only, Rx_and_Tx
	8	AUTO, 0	Rx_only, Tx_only, Rx_and_Tx

3.3. IP Parameter Settings for Example Use Cases

For G8B10B configurations, the following are the recommended settings for the *Lane Alignment* attribute.

- Single-link usage (consisting of multiple-lane): LALIGN_EN = *ENABLED*
- Multi-link usage (consisting of single-lane): LALIGN_EN = *DISABLED*

4. Signal Description

The ports available in the MPCS module are defined in the following subsections.

Table 4.1. MPCS Module Signal Description

Port	Type	Width	Description
MPCS Interface[1, 5]			
Clock and Reset			
mpcs_rx_usr_clk_i	Input	NL	User Interface RX Clock Input
mpcs_tx_usr_clk_i	Input	NL	User Interface TX Clock Input
mpcs_tx_pcs_rstn_i	Input	NL	Active low signal used to reset the TX path of MPCS module. This signal must only be released when PMA has completed calibration.
mpcs_rx_pcs_rstn_i	Input	NL	Active low signal used to reset the RX path of MPCS module. This signal must only be released when PMA has completed calibration.
mpcs_cc_ clk_i	Input	NL	Input clock for Clock Frequency Compensation; CTC Clock Input.
mpcs_rx_out_clk_o	Output	NL	PCS RX Output Clock
mpcs_tx_out_clk_o	Output	NL	PCS TX Output Clock
mpcs_perstn_i	Input	NL	Fundamental reset, triggers PCS auto calibration. User logic can release this signal to inform SERDES/PCS module that user logic is ready and no more registers need to be configured from user logic which requires PMA and PMA controller under the reset state. For more information, refer to *Section 7.7.3 Reset Sequence* of the CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245).
TX/RX FIFO Signals			
mpcs_tx_ch_din_i	Input	80×NL	For the signal mapping of this port, refer to Table 4.2 and Table 4.3.
mpcs_tx_fifo_st_o	Output	4×NL	bit[0]: FIFO is almost empty bit[1]: FIFO is almost full bit[2]: FIFO underflow bit[3]: FIFO overflow
mpcs_rx_ch_dout_o	Output	80×NL	For the signal mapping of this port, refer to Table 4.5 and Table 4.6.
mpcs_rx_fifo_st_o	Output	4×NL	When using 8b10b PCS: bit[0]: FIFO is almost empty bit[1]: FIFO is almost full bit[2]: FIFO underflow bit[3]: FIFO overflow When using 64b66b PCS or EPCS Mode: bit[0]: Reserved bit[1]: Reserved bit[2]: FIFO underflow bit[3]: FIFO overflow
Elastic Buffer Signals			
mpcs_ebuf_empty_o	Output	NL	Elastic Buffer Empty output port. 1'b1 – the frequency compensation buffer (Elastic Buffer) is empty. 1'b0 – the frequency compensation buffer (Elastic Buffer) is not empty.
mpcs_ebuf_full_o	Output	NL	Elastic Buffer Full output port. 1'b1 – the frequency compensation buffer (Elastic Buffer) is full. 1'b0 – the frequency compensation buffer (Elastic Buffer) is not full.
mpcs_anxmit_i	Input	NL	In GIGE application, the high level of this signal indicates the current state is GIGE Auto-negotiation. In this state, replacing /C1/, /C2/ with /I1/, /I2/ ordered sets periodically so that the following stage (CTC) gets an opportunity to perform a clock frequency compensation.

Port	Type	Width	Description
Word Aligner Signals			
mpcs_walign_en_i	Input	NL	Word align enable input port. This function is useful if the automatic synchronization is not enabled. The rising edge of this signal triggers the word alignment operation.
mpcs_get_lsync_o	Output	NL	Link Synchronization output port. 1'b1 – link synchronization is acquired. 1'b0 – loss of link synchronization.
Lane-to-Lane Deskew Signals			
mpcs_rx_get_lalign_o	Output	NL	Receive Lane align output port. 1'b1 – alignment acquired. 1'b0 – loss of alignment.
mpcs_rx_deskew_en_i	Input	NL	Receive Deskew enable port. The rising edge of this signal triggers the lane-to-lane deskew operation.
BER Monitor (64b/66b PCS)			
mpcs_rx_hi_ber_o	Output	NL	The high level of this signal indicates HIGH BER is indicated.
Block Aligner (64b/66b PCS)			
mpcs_rx_blk_lock_o	Output	NL	The high level of this signal indicates the block lock is achieved.
PMA Control and Status Signals			
mpcs_rstn_i	Input	NL	This signal resets signals in mpcs_clkin_i clock domain. Also, this reset drives the PMA epcs_rstn_i port.
mpcs_clkin_i	Input	NL	This low-speed clock also drives the PMA epcs_clkin_i clock port.
mpcs_pwrdn_i	Input	2×NL	This signal is used to put the PMA in power down and it has only three states: • 2'b00 – operational • 2'b10 – low-power state • 2'b11 – deep low-power state
mpcs_txhiz_i	Input	NL	This signal is used to load Electrical Idle III in the Tx driver of the PMA macro.
mpcs_rxidle_o	Output	NL	This port is used to signal the Electrical Idle condition detected by the PMA control logic.
mpcs_rxerr_i	Input	NL	This signal is used to report to the PMA control logic that error data has been detected by the PCS-PCIE logic.
mpcs_fomreq_i	Input	NL	This signal is used to request a Figure of Merit (FOM) evaluation in PCS-PCIE.
mpcs_fomack_o	Output	NL	This signal is used to report to the PMA control logic that error data has been detected.
mpcs_fomrslt_o	Output	8×NL	This signal is the evaluated FOM result. This signal is synchronous to mpcs_tx_out_clk_o.
mpcs_rate_i	Input	2×NL	MPCS data rates, only multi-rate protocol can access this signal. For more information, refer to Table 2.1.
mpcs_speed_o	Output	2×NL	MPCS current speeds.
mpcs_txval_i	Input	NL	PHY transmit valid: This signal is used to transmit valid data. If deasserted, the PMA macro is put in Electrical Idle 1. It can be used for protocol requiring Electrical Idle (SATA) and must also be deasserted as long as mpcs_ready_o is not asserted. This signal is also required to be generated one clock cycle earlier than the corresponding mpcs_tx_data_i signals.
mpcs_phyrdy_o	Output	NL	When asserted, this signal tells you that the PHY is ready to transmit while using mpcs_txval_i.
mpcs_ready_o	Output	NL	PHY ready: This signal is asserted when the PHY has completed the calibration sequence for each specific lane. This signal is driven by mpcs_clkin_i.

Port	Type	Width	Description
mpcs_rxoob_i	Input	NL	This signal configures the activity detector to detect OOB (out-of-band) accurately.
mpcs_rxval_o	Output	NL	This signal is used to signal receiving valid data.
mpcs_txdeemp_i	Input	NL	When asserted high, programmed de-emphasis is applied to the transmitter driver.
mpcs_pwrst_o	Output	2×NL	This signal is used to report the PHY current power state. This signal is driven by mpcs_clkin_i.
mpcs_skipbit_i	Input	NL	Pipe control to skip 1 bit on the receiver. When asserted, this causes the receiver to freeze 1-bit clock. This function can be used to control word alignment. This signal is considered as an asynchronous clock.
EPCS Interface[2, 5]			
Clock and Reset Signals			
epcs_rx_usr_clk_i	Input	NL	User interface RX clock input
epcs_tx_usr_clk_i	Input	NL	User interface TX clock input
epcs_tx_pcs_rstn_i	Input	NL	Active low signal used to reset the TX path of the MPCS module
epcs_rx_pcs_rstn_i	Input	NL	Active low signal used to reset the RX path of the MPCS module
epcs_rx_clk_o	Output	NL	PCS RX output clock
epcs_tx_clk_o	Output	NL	PCS TX output clock
epcs_rstn_i	Input	NL	Fundamental reset, triggers PCS auto calibration.
TX/RX FIFO Signals			
epcs_txdata_i	Input	80×NL	For the signal mapping of this port, refer to Table 4.2 and Table 4.3.
epcs_rxdata_o	Output	80×NL	For the signal mapping of this port, refer to Table 4.5 and Table 4.6.
PMA Control and Status Signals			
epcs_clkin_i	Input	NL	This low speed clock also drives the PMA epcs_clkin_i clock port.
epcs_pwrdn_i	Input	2×NL	This signal is used to put the PMA in power down and it only has three states. • 2'b00 – operational • 2'b10 – low-power state • 2'b11 – deep low-power state
epcs_txhiz_i	Input	NL	This signal is used to load Electrical Idle III in the Tx driver of the PMA macro.
epcs_rxidle_o	Output	NL	This port is used to signal the Electrical Idle condition detected by the PMA control logic.
epcs_rxerr_i	Input	NL	This signal is used to report to the PMA control logic that error data has been detected by the PCS-PCIE logic.
epcs_fomreq_i	Input	NL	This signal is used to request a Figure of Merit (FOM) evaluation in PCS-PCIE.
epcs_fomack_o	Output	NL	This signal is used to report the PMA control logic that error data has been detected.
epcs_fomrslt_o	Output	8×NL	This signal is the evaluated FOM result. This signal is synchronous to epcs_tx_out_clk_o.
epcs_rate_i	Input	2×NL	EPCS data rates, only multi-rate protocol can access this signal. For more information, refer to Table 2.1.
epcs_speed_o	Output	2×NL	EPCS current speeds.
epcs_txval_i	Input	NL	PHY transmit valid: This signal is used to transmit valid data. If deasserted, the PMA macro is put in Electrical Idle 1. It can be used for protocol requiring Electrical Idle (SATA) and must also be deasserted as long as epcs_ready_o is not asserted. This signal is also required to be generated one clock cycle earlier than corresponding epcs_tx_data_i signals.

Port	Type	Width	Description
epcs_phyrdy_o	Output	NL	When asserted, this signal indicates that the PHY is ready to transmit while using epcs_txval_i.
epcs_ready_o	Output	NL	PHY ready: This signal is used to release the reset for the external PCS and controller, start transmitting data to the PMA. This signal is driven by epcs_clkin_i.
epcs_rxoob_i	Input	NL	This signal configures the activity detector to detect OOB (out-of-band) accurately.
epcs_txdeemp_i	Input	NL	When asserted high, programmed de-emphasis is applied to the transmitter driver.
epcs_pwrst_o	Output	2×NL	This signal is used to report the PHY current power state. This signal is driven by epcs_clkin_i.
epcs_rxval_o	Output	NL	This signal is used to signal receive valid data.
epcs_skipbit_i	Input	NL	Pipe control to skip 1 bit on receiver. When asserted, this causes the receiver to freeze 1-bit clock. This function can be used to control word alignment. This signal is considered as an asynchronous clock.
LMMI Interface[5]			
lmmi_clk_i	Input	NL	LMMI clock input
lmmi_resetn_i	Input	NL	Active low LMMI reset
lmmi_request_i	Input	NL	Start transaction
lmmi_wr_rdn_i	Input	NL	Write = HIGH, Read = LOW
lmmi_offset_i	Input	9×NL	Address/Offset
lmmi_wdata_i	Input	8×NL	Write data
lmmi_rdata_valid_o	Output	NL	Valid data indicator
lmmi_ready_o	Output	NL	LMMI ready indicator
lmmi_rdata_o	Output	8×NL	Read data
Serial I/O			
sdq_refclkp_i	Input	1	Reference clock of SERDES PLL in one Quad
sdq_refclkn_i	Input	1	Reference clock of SERDES PLL in one Quad
sd[n]rxp_i [3]	Input	NL	Analog RX diffIO
sd[n]rxn_i [3]	Input	NL	Analog RX diffIO
sd[n]txp_o [3]	Output	NL	Analog TX diffIO
sd[n]txn_o [3]	Output	NL	Analog TX diffIO
sd[n]_rext_i [3]	Input	NL	External resistance
sd[n]_refret_i [3]	Input	NL	—
Quad-Quad Interface			
tx_lalign_clk_out_o	Output	1	The outputted clock for sharing among Quads are hooked up during Quad integration.
rx_lalign_clk_out_o	Output	1	The outputted clock for sharing among Quads are hooked up during Quad integration.
tx_lalign_clk_in_i	Input	1	The input clock shared by all channels within a Quad to implement lane alignment.
rx_lalign_clk_in_i	Input	1	The input clock shared by all channels within a Quad to implement lane alignment.
lalign_out_up_o	Output	8	The connection signals between Quads for lane alignment across Quad boundary.
lalign_in_up_i	Input	8	The connection signals between Quads for lane alignment across Quad boundary.
lalign_out_down_o	Output	8	The connection signals between Quads for lane alignment across Quad boundary.

Port	Type	Width	Description
lalign_in_down_i	Input	8	The connection signals between Quads for lane alignment across Quad boundary.
JTAG Interface			
acjtag_mode_i	Input	1	When asserted, this signal activates the ACJTAG controller of the PMA control logic, which now controls the PMA hard macro. The PMA hard macro is thus disconnected from the PMA control logic: This signal is used for two purposes: • Selects the multiplexer between ACJTAG controller and functional logic at the PMA interface directly. • Puts out of reset the ACJTAG controller. This signal is used as the reset input for the embedded.
acjtag_enable_i	Input	1	This signal configures the PMA in ACTAG test mode. By default, it resets the Tx and Rx driver and receiver followed by loading the driver and receiver with default settings. The PMA receiver is by default in the DC test mode and the transmitter is driving 0.
acjtag_acmode_i	Input	1	When acjtag_enable_i == 1'b1, this signal selects between AC mode (1'b1) or DC mode (1'b0).
acjtag_drive1_i	Input	1	When acjtag_enable_i == 1'b1, this signal selects what differential value does the transmitter drive.
acjtag_highz_i	Input	1	When acjtag_mode_i == 1'b1, assertion of this signal puts the PMA driver in high impedance.
acjtagpout_o	Output	1	ACJTAG output data.
acjtagnout_o	Output	1	ACJTAG output data.
Reference Clock Ports			
use_refmux_i	Input	1	When asserted, the alternative reference clock is used. See the Reference Clock Source Selection section for more details.
sd_ext_0_refclk_i	Input	1	PMA PLL refclk from external I/O pad0
sd_ext_1_refclk_i	Input	1	PMA PLL refclk from external I/O pad1
pll_0_refclk_i	Input	1	Reference clock from Left PLL
pll_1_refclk_i	Input	1	Reference clock from right PLL
sd_pll_refclk_i	Input	1	Reference clock from fabric

Notes:

1. These ports are available only when in MPCS mode.
2. These ports are available only when in EPCS mode.
3. [n] indicates lane/channel number.
4. Displayed signals have suffix channel identifier. For example, for mpcs_tx_ch_din_i with two channels, the displayed signals are mpcs_tx_ch_din_i_0[79:0] and mpcs_tx_ch_din_i_1[79:0].
5. NL means Number of Lanes.

4.1. Data Bus Sharing and Mapping

The data bus of MPCS module is shared by 8B10B PCS, 64B66B PCS, and EPCS Mode. Table 4.2 to Table 4.7 specify the mapping in each mode and the port definition.

Table 4.2. TX Data Bus Sharing [39:0]

MODE	mpcs_tx_ch_din/ epcs_txdata [39:32]	mpcs_tx_ch_din/ epcs_txdata [31:24]	mpcs_tx_ch_din/ epcs_txdata [23:16]	mpcs_tx_ch_din/ epcs_txdata [15:8]	mpcs_tx_ch_din/ epcs_txdata [7:0]
MPCS 8b10b	tx_data_8b[39:0]				
MPCS 64b66b	tx_data_64b[39:0]				
EPCS	tx_data[39:0]				

Table 4.3. TX Data Bus Sharing [79:40]

MODE	mpcs_tx_ch_din/ epcs_txdata [79:72]	mpcs_tx_ch_din/ epcs_txdata [71:64]	mpcs_tx_ch_din/ epcs_txdata [63:56]	mpcs_tx_ch_din/ epcs_txdata [55:48]	mpcs_tx_ch_din/ epcs_txdata [47:40]
MPCS 8b10b	—	—	—	{4'b0, tx_frcdata[3:0]}	{tx_dispval[3:0], tx_frcdisp[3:0]}
MPCS 64b66b	{tx_fifo_wr, 6'b0, tx_frcpkt}	tx_control[7:0]	tx_data_64b[63:40]		
EPCS	—	—	—	—	—

4.1.1. MPCS Mode 8b10b Tx Data

tx_data_8b – 4-byte input data.

Table 4.4. Bit Mapping for MPCS Mode 8b10b Tx Data

Bit Mapping	Description
Byte_3 – bit[39:30]	Bit[39]: cordisp bit or data bit[9] in 10b mode 1'b1 – replace the data D16.2 with D5.6 in a certain condition or replace /I2/ symbol with /I1/ in GigE protocol. 1'b0 – do not replace. Bit[38]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[37:30]: data bit[7:0]
Byte_2 – bit[29:20]	Bit[29]: cordisp bit or data bit[9] in 10b mode 1'b1 – replace the data D16.2 with D5.6 in a certain condition or replace /I2/ symbol with /I1/ in GigE protocol. 1'b0 – do not replace. Bit[28]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[27:20]: data bit[7:0]
Byte_1 – bit[19:10]	Bit[19]: cordisp bit or data bit[9] in 10b mode 1'b1 – replace the data D16.2 with D5.6 in a certain condition or replace /I2/ symbol with /I1/ in GigE protocol. 1'b0 – do not replace. Bit[18]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[17:10]: data bit[7:0]

Bit Mapping	Description
Byte_0 – bit[9:0]	Bit[9]: cordisp bit or data bit[9] in 10b mode 1'b1 – replace the data D16.2 with D5.6 in a certain condition or replace /I2/ symbol with /I1/ in GigE protocol. 1'b0 – do not replace. Bit[8]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[7:0]: data bit[7:0]

- tx_frcdisp
  1'b1 – Use *force running disparity* mode; in this mode the running disparity is indicated by *tx_dispval* signal.
  1'b0 – Do not use *force running disparity* mode.

- tx_dispval
  In *force running disparity* mode, this signal indicates the forced running disparity:
  1'b1 – Force positive running disparity.
  1'b0 – Force negative running disparity.
  Otherwise (not in *force running disparity* mode),
  1'b1 – Revert running disparity.
  1'b0 – Use running disparity calculated by 8b10b encoder.

- tx_frcdata
  1'b1 – Force 8b10b encoder to output *tx_data*.
  1'b0 – Do not force the output data of 8b10b encoder.

4.1.2. MPCS Mode 64b66b Tx Data

- tx_control – 8-bit control indication
  bit[0] is the control signal for tx_data_64b[7:0];
  bit[1] is the control signal for tx_data_64b[15:8];
  …
  bit[7] is the control signal for tx_data_64b[63:56].

- tx_frcpkt
  When this signal is asserted high, the 64-bit data (tx_data_64b) and 2-bit header (tx_control[1:0]) on the same clock cycle is used to drive TX Gear Box directly. The Encoder and Scrambler are bypassed in this case.
  Once deasserted, the 64-bit data and 8-bit control is encoded and scrambled before they go to the TX Gear Box.
  1'b1 – On the same cycle, use the 64-bit data and 2-bit header coming from user logic to feed the Gear Box.
  1'b0 – Process the 64-bit data and 8-bit control as normal (go through the Encoder, Scrambler and Gear Box).

- tx_fifo_wr
  1'b1 – Write 64-b data and 8-b control to TX FIFO.
  User logic should monitor the FIFO status and properly control the writing operation to avoid the FIFO overflow or underflow.

Table 4.5. RX Data Bus Sharing [39:0]

MODE	mpcs_rx_ch_dout/ epcs_rxdata [39:32]	mpcs_rx_ch_dout/ epcs_rxdata [31:24]	mpcs_rx_ch_dout/ epcs_rxdata [23:16]	mpcs_rx_ch_dout/ epcs_rxdata [15:8]	mpcs_rx_ch_dout/ epcs_rxdata [7:0]
MPCS 8b10b	rx_data_8b[39:0]				
MPCS 64b66b	rx_data_64b[39:0]				
EPCS	rx_data[39:0]				

Table 4.6. RX Data Bus Sharing [79:0]

MODE	mpcs_rx_ch_dout/ epcs_rxdata [79:72]	mpcs_rx_ch_dout/ epcs_rxdata [71:64]	mpcs_rx_ch_dout/ epcs_rxdata [63:56]	mpcs_rx_ch_dout/ epcs_rxdata [55:48]	mpcs_rx_ch_dout/ epcs_rxdata [47:40]
MPCS 8b10b	—	—	—	{ rx_skp_del[3:0]}, rx_skp_add[3:0]}	{rx_errcode[3:0], rx_errdisp[3:0]}
MPCS 64b66b	{ rx_data_valid, 3'b0, rx_fifo_del[1:0], rx_fifo_add[1:0]}	rx_control[7:0]	rx_data_64b[63:40]		
EPCS	—	—	—	—	—

4.1.3. MPCS Mode 8b10b Rx Data

rx_data_8b – 4-byte output data.

Table 4.7. Bit Mapping for MPCS Mode 8b10b Rx Data

Bit Mapping	Description
Byte_3 – bit[39:30]	Bit[39]: rundisp bit represents running disparity or data bit[9] in 10b mode. 1'b1 – positive 1'b0 – negative Bit[38]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[37:30]: data bit[7:0]
Byte_2 – bit[29:20]	Bit[29]: rundisp bit represents running disparity or data bit[9] in 10b mode. 1'b1 – positive 1'b0 – negative Bit[28]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[27:20]: data bit[7:0]
Byte_1 – bit[19:10]	Bit[19]: rundisp bit represents running disparity or data bit[9] in 10b mode. 1'b1 – positive 1'b0 – negative Bit[18]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[17:10]: data bit[7:0]
Byte_0 – bit[9:0]	Bit[9]: rundisp bit represents running disparity or data bit[9] in 10b mode. 1'b1 – positive 1'b0 – negative Bit[8]: control character or data bit[8] in 10b mode 1'b1 – 10b control code 1'b0 – 10b data code Bit[7:0]: data bit[7:0]

- rx_errdisp
  1'b1 – Disparity error
  1'b0 – No error

- rx_errcode
  1'b1 – Invalid code group
  1'b0 – Valid code group

- rx_skp_add
  1'b1 – Added byte indication given by Elastic Buffer
  1'b0 – Normal byte

- rx_skp_del
  1'b1 – Deleted byte indication given by Elastic Buffer
  1'b0 – Normal byte

4.1.4. MPCS Mode 64b66b Rx Data

- rx_control – 8-bit control indication.
  bit[0] is the control signal for rx_data_64b[7:0];
  bit[1] is the control signal for rx_data_64b[15:8];
  …
  bit[7] is the control signal for rx_data_64b[63:56].

- rx_fifo_add – The indication of insertion for clock frequency difference compensation.
  bit[1] corresponds to 4-byte Idles which are carried by bit[63:32] of data bus (rx_data_64b).
  bit[0] corresponds to 4-byte Idles which are carried by bit[31:0] of data bus (rx_data_64b).

- rx_fifo_del – The indication of deletion for clock frequency difference compensation.
  bit[1] corresponds to 4-byte Idles or sequence Ordered-set which are carried by bit[63:32] of data bus (rx_data_64b).
  bit[0] corresponds to 4-byte Idles or sequence Ordered-set which are carried by bit[31:0] of data bus (rx_data_64b).

- rx_data_valid
  1'b1 – The output data is valid.
  1'b0 – There is no valid data on the output of RX FIFO.

5. Register Description

5.1. Overview

Table 5.1 lists the address map and specifies the registers available to you. The PMA and MPCS blocks need 8-bit offset only. To separate the PMA internal register from MPCS internal register, the lmmi_offset_i[8] is used.

- lmmi_offset_i[8] == 1'b1 – MPCS Register space
- lmmi_offset_i[8] == 1'b0 – PMA Register space

Table 5.1. Summary of MPCS Registers

Offset LMMI	Register Name	Access Type	Description
8'h00	reg00	RW	MPCS Data Path Selection
8'h10	reg10	RW	MPCS TX Path Control
8'h11	reg11	RW	8b/10b Encoder Control
8'h20	reg20	RW	MPCS RX Path Control
8'h21	reg21	RO	MPCS RX Path Status
8'h22	reg22	RW	8b/10b Decoder Control
8'h30	reg30	RW	Word Alignment Control
8'h31	reg31	RW	Primary Word Alignment Pattern Byte 0
8'h32	reg32	RW	Primary Word Alignment Pattern Byte 1
8'h33	reg33	RW	Primary Word Alignment Pattern MSB
8'h34	reg34	RW	Secondary Word Alignment Pattern Byte 0
8'h35	reg35	RW	Secondary Word Alignment Patter Byte 1
8'h36	reg36	RW	Secondary Word Alignment Pattern MSB
8'h37	reg37	RW	Word Alignment Pattern Mask Code Byte 0
8'h38	reg38	RW	Word Alignment Pattern Mask Code Byte 1
8'h39	reg39	RW	Word Alignment Pattern Mask Code MSB
8'h3a	reg3a	RW	Sync_Det FSM Configuration 0
8'h3b	reg3b	RW	Sync_Det FSM Configuration 1
8'h3c	reg3c	RW	Sync_Det FSM Configuration 2
8'h3d	reg3d	RW	Sync_Det FSM Configuration 3
8'h3e	reg3e	RO	Number of Bit Slipped During Word Alignment
8'h3f	reg3f	RW	Primary Sync_Det Pattern Byte 0
8'h40	ref40	RW	Primary Sync_Det Pattern Byte 1
8'h41	reg41	RW	Primary Sync_Det Pattern Byte 2
8'h42	reg42	RW	Primary Sync_Det Pattern Byte 3
8'h43	reg43	RW	Primary Sync_Det Pattern MSB
8'h44	reg44	RW	Secondary Sync_Det Pattern Byte 0
8'h45	reg45	RW	Secondary Sync_Det Pattern Byte 1
8'h46	reg46	RW	Secondary Sync_Det Pattern Byte 2
8'h47	reg47	RW	Secondary Sync_Det Pattern Byte 3
8'h48	reg48	RW	Secondary Sync_Det Pattern MSB
8'h49	reg49	RW	Sync_Det Pattern Mask Code Byte 0
8'h4A	reg4a	RW	Sync_Det Pattern Mask Code Byte 1
8'h4B	reg4b	RW	Sync_Det Pattern Mask Code Byte 2
8'h4C	reg4c	RW	Sync_Det Pattern Mask Code Byte 3
8'h4D	reg4d	RW	Sync_Det Pattern Mask Code MSB
8'h50	reg50	RW	Lane Alignment Control
8'h51	reg51	RW	Maximum Lane-to-lane Skew

Offset LMMI	Register Name	Access Type	Description
8'h52	reg52	RW	Primary Lane Alignment Pattern Byte 0
8'h53	reg53	RW	Primary Lane Alignment Pattern Byte 1
8'h54	reg54	RW	Primary Lane Alignment Pattern Byte 2
8'h55	reg55	RW	Primary Lane Alignment Pattern Byte 3
8'h56	reg56	RW	Primary Lane Alignment Pattern MSB
8'h57	reg57	RW	Secondary Lane Alignment Pattern Byte 0
8'h58	reg58	RW	Secondary Lane Alignment Pattern Byte 1
8'h59	reg59	RW	Secondary Lane Alignment Pattern Byte 2
8'h5a	reg5a	RW	Secondary Lane Alignment Pattern Byte 3
8'h5b	reg5b	RW	Secondary Lane Alignment Pattern MSB
8'h5c	reg5c	RW	Lane Alignment Pattern Mask Code
8'h60	reg60	RW	Clock Frequency Compensation Control
8'h61	reg61	RW	SKIP Pattern Insertion/Deletion Control
8'h62	reg62	RW	Elastic FIFO High Water Line
8'h63	reg63	RW	Elastic FIFO Low Water Line
8'h64	reg64	RW	Primary SKIP Pattern Byte 0
8'h65	reg65	RW	Primary SKIP Pattern Byte 1
8'h66	reg66	RW	Primary SKIP Pattern Byte 2
8'h67	reg67	RW	Primary SKIP Pattern Byte 3
8'h68	reg68	RW	Primary SKIP Pattern Byte MSB
8'h69	reg69	RW	Secondary SKIP Pattern Byte 0
8'h6a	reg6a	RW	Secondary SKIP Pattern Byte 1
8'h6b	reg6b	RW	Secondary SKIP Pattern Byte 2
8'h6c	reg6c	RW	Secondary SKIP Pattern Byte 3
8'h6d	reg6d	RW	Secondary SKIP Pattern Byte MSB
8'h6e	reg6e	RW	SKIP Pattern Mask Code
8'h80	reg80	RW	64b/66b PCS TX Path Control
8'h81	reg81	RW	64b/66b PCS TX FIFO Almost Full Setting
8'h82	reg82	RW	64b/66b PCS TX FIFO Almost Empty Setting
8'h83	reg83	RW	64b/66b PCS RX Path Control
8'h84	reg84	RW	64b/66b PCS CTC High Water Line
8'h85	reg85	RW	64b/66b PCS CTC Low Water Line
8'h86	reg86	RO	64b/66b PCS Block Align Shift
8'h90	reg90	RO	10GBASE-R BER Counter
8'h91	reg91	RO	10GBASE-R Block Error Counter
8'h92	reg92	RW	10GBASE-R Test Pattern Seed A Byte0
8'h93	reg93	RW	10GBASE-R Test Pattern Seed A Byte1
8'h94	reg94	RW	10GBASE-R Test Pattern Seed A Byte2
8'h95	reg95	RW	10GBASE-R Test Pattern Seed A Byte3
8'h96	reg96	RW	10GBASE-R Test Pattern Seed A Byte4
8'h97	reg97	RW	10GBASE-R Test Pattern Seed A Byte5
8'h98	reg98	RW	10GBASE-R Test Pattern Seed A Byte6
8'h99	reg99	RW	10GBASE-R Test Pattern Seed A Byte7
8'h9A	reg9a	RW	10GBASE-R Test Pattern Seed B Byte0
8'h9B	reg9b	RW	10GBASE-R Test Pattern Seed B Byte1
8'h9C	reg9c	RW	10GBASE-R Test Pattern Seed B Byte2
8'h9D	reg9d	RW	10GBASE-R Test Pattern Seed B Byte3
8'h9E	reg9e	RW	10GBASE-R Test Pattern Seed B Byte4

Offset LMMI	Register Name	Access Type	Description
8'h9F	reg9f	RW	10GBASE-R Test Pattern Seed B Byte5
8'hA0	rega0	RW	10GBASE-R Test Pattern Seed B Byte6
8'hA1	rega1	RW	10GBASE-R Test Pattern Seed B Byte7
8'hA2	rega2	RW	10GBASE-R Test Pattern Control 0
8'hA3	rega3	RW	10GBASE-R Test Pattern Control 1
8'hA4	rega4	RO	10GBASE-R Test Pattern Error Counter Byte0
8'hA5	rega5	RO	10GBASE-R Test Pattern Error Counter Byte1
8'he0	rege0	RW	Loop-back Mode Control
8'he1	rege1	RW	MPCS BIST Control 0
8'he2	rege2	RW	MPCS BIST Control 1
8'he3	rege3	RW	User-defined BIST Constant 1 Byte_0
8'he4	rege4	RW	User-defined BIST Constant 1 Byte_1
8'he5	rege5	RW	User-defined BIST Constant 1 MSByte
8'he6	rege6	RW	User-defined BIST Constant 2 Byte_0
8'he7	rege7	RW	User-defined BIST Constant 2 Byte_1
8'he8	rege8	RW	User-defined BIST Constant 2 MSByte
8'he9	rege9	RO	BIST Status 0
8'hea	regea	RO	BIST Status 1

The PMA register space is implemented up to 256 registers of 8 bits. The register space can be divided into protocol-specific RW registers and generic purpose registers.

Table 5.2. Summary of PMA Registers

Offset LMMI	Register Name	Access Type	Description
8'h00 – 8'hFF	PMA registers	RW	Refer to Appendix A.1 of CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245).

The behavior of the registers on the write and read access is defined by their access type, which is defined in Table 5.3.

Table 5.3. Access Type Definition

Access Type	Behavior on Read Access	Behavior on Write Access
RO	Returns register value.	Ignores write access.
WO	Returns 0.	Updates register value.
RW	Returns register value.	Updates register value.
RW1C	Returns register value.	Writing 1'b1 on a register bit clears the bit to 1'b0. Writing 1'b0 on a register bit is ignored.
CR	Clears on read.	Clears register value on read.

5.2. MPCS Data Path Selection

Table 5.4. TX Path Control Register

Field	Name	Description	Access	Width	Reset
[7]	reserved	—	RW	1	1'b0
[6:5]	rx_src_sel	RX Path Source Selector. Specifies the selected source/interface for the RX Path. 2'b00 – PMA+8b/10b PCS. 2'b01 – PMA+64b/66b PCS. 2'b1x – PMA only.	RW	2	2'b00

Field	Name	Description	Access	Width	Reset
[4]	rx_src_ovrd	RX Path override. Specifies whether to override or not the selected source/interface for the RX Path. 1'b0 – Use the MODESEL to set the MPCS mode for the RX path. 1'b1 – Use rx_src_sel to override the MODESEL to set the MPCS mode for the RX path.	RW	1	1'b0
[3]	reserved	—	RW	1	1'b0
[2:1]	tx_src_sel	TX Path Source Selector. Specifies the selected source/interface for the TX Path. 2'b00 – PMA+8b/10b PCS. 2'b01 – PMA+64b/66b PCS. 2'b1x – PMA only.	RW	2	2'b00
[0]	tx_src_ovrd	TX Path override. Specifies whether to override or not the selected source/interface for the TX Path. 1'b0 – Use the MODESEL to set the MPCS mode for the TX path. 1'b1 – Use tx_src_sel to override the MODESEL to set the MPCS mode for the TX path.	RW	1	1'b0

5.3. MPCS TX Path Control

Table 5.5. TX Path Control Register

Field	Name	Description	Access	Width	Reset
[7]	tx_pmfifo_dis	Phase Matching FIFO. Specifies whether the Phase Matching FIFO is enabled or disabled. (TX lane-to-lane deskew) 1'b0 – Phase Matching FIFO enabled. 1'b1 – Phase Matching FIFO disabled.	RW	1	*TX Lane-to-Lane Deskew*
[6]	tx_bond_mask	Bond Mask. The bonded channel mode is defined by the BOND MODE parameter. 1'b0 – Do not exclude this channel. 1'b1 – Exclude this channel from the bonded channel group.	RW	1	1'b0
[5]	tx_dbus_20b	Bus Width. Specifies the internal data bus width for the TX Path. 1'b0 – Internal data bus is 10-bit width. 1'b1 – Internal data bus is 20-bit width.	RW	1	*Bus Width*
[4]	enc_8b10b_dis	8b10b Encoding Enable. Specifies whether the 8b10b encoding is enabled or disabled. 1'b0 – Encoding enabled. 1'b1 – Encoding disabled.	RW	1	*8b10b Encoder*
[3]	tx_fifo_dis	TX FIFO Enable. Specifies whether the TX phase compensation FIFO is enabled or disabled. 1'b0 – TX phase compensation FIFO enabled. 1'b1 – TX phase compensation FIFO disabled.	RW	1	*TX FIFO*
[2]	tx_gear_en	Gearing Enable. Specifies whether the 2:1 Gearing is enabled or disabled. 1'b0 – Gearing is disabled. 1'b1 – Gearing is enabled.	RW	1	*2:1 Gearing*

Field	Name	Description	Access	Width	Reset
[1]	tx_mpcs_rst	TX MPCS Reset. Soft resets the TX Path not including the register space. 1'b0 – Do not reset the TX MPCS path. 1'b1 – Reset the TX MPCS path.	RW	1	1'b0
[0]	tx_mpcs_dis	TX Path Enable. Specifies whether the MPCS channel TX Path is enabled or disabled. 1'b0 – MPCS channel TX Path is enabled. 1'b1 – MPCS channel TX Path is disabled.	RW	1	1'b0

5.4. 8b/10b Encoder Control

Table 5.6. 8b/10b Encoder Control Register

Field	Name	Description	Access	Width	Reset
[7:1]	reserved	—	RW	7	—
[0]	enc_8b10b_interleave	8b/10b Encoder Interleave. Specifies whether the 8b/10b Encoder Interleave is enabled or disabled. 1'b0 – Interleaving mode in the 8b/10b encoder is disabled. 1'b1 – Interleaving mode in the 8b/10b encoder is enabled.	RW	1	1'b0

5.5. MPCS RX Path Control

Table 5.7. RX Path Control Register

Field	Name	Description	Access	Width	Reset
[7]	rfifo_com_align	COMMA Byte alignment. Specifies whether the feature of putting the COMMA byte to the LSByte (Byte_0) of the data bus is enabled or disabled. 1'b0 – Disable this feature. 1'b1 – Enable this feature.	RE	1	*Put the COMMA byte to LSByte*
[6]	rx_bond_mask	Bond Mask. The bonded channel mode is defined by the BOND MODE parameter. 1'b0 – Do not exclude this channel. 1'b1 – Exclude this channel from the bonded channel group.	RW	1	1'b0
[5]	rx_dbus_20b	Bus Width. Specifies the internal data bus width for the RX Path. 1'b0 – Internal data bus is 10-bit width. 1'b1 – Internal data bus is 20-bit width.	RW	1	*Bus Width*
[4]	dec_8b10b_dis	8b10b Decoding Enable. Specifies whether the 8b10b decoding is enabled or disabled. 1'b0 – Decoding enabled. 1'b1 – Decoding disabled.	RW	1	*8b10b Decoder*
[3]	rx_fifo_dis	RX FIFO Enable. Specifies whether the RX phase compensation FIFO is enabled or disabled. 1'b0 – RX phase compensation FIFO enabled. 1'b1 – RX phase compensation FIFO disabled.	RW	1	*RX FIFO*

Field	Name	Description	Access	Width	Reset
[2]	rx_gear_en	Gearing Enable. Specifies whether the 1:2 Gearing is enabled or disabled. 1'b0 – Gearing disabled. 1'b1 – Gearing enabled.	RW	1	*2:1 Gearing*
[1]	rx_mpcs_rst	RX MPCS Reset. Soft resets the RX Path not including the register space. 1'b0 – Do not reset the RX MPCS path. 1'b1 – Reset the RX MPCS path.	RW	1	1'b0
[0]	rx_mpcs_dis	RX Path Enable. Specifies whether the MPCS channel RX Path is enabled or disabled. 1'b0 – MPCS channel RX Path is enabled. 1'b1 – MPCS channel RX Path is disabled.	RW	1	1'b0

5.6. MPCS RX Path Status

Table 5.8. MPCS RX Path Status Register

Field	Name	Description	Access	Width	Reset
[7]	reserved	—	RW	7	—
[6]	rfifo_byte_shift	Byte shift. Specifies the data shift caused by the COMMA byte alignment operation. 1'b0 – There is no byte shift. 1'b1 – There is one byte data shift.	RO	1	1'b0

5.7. 8b/10b Decoder Control

Table 5.9. 8b/10b Decoder Control Register

Field	Name	Description	Access	Width	Reset
[7:1]	reserved	—	RW	7	—
[0]	dec_8b10b_interleave	8b/10b Decoder Interleave. Specifies whether the 8B/10B Decoder Interleave is enabled or disabled. 1'b0 – Interleaving mode in 8b/10b decoder is disabled. 1'b1 – Interleaving mode in 8b/10b decoder is enabled.	RW	1	1'b0

5.8. Word Alignment Control

Table 5.10. Word Alignment Control Register

Field	Name	Description	Access	Width	Reset
[7:6]	reserved	—	RW	2	—
[5]	wa_dis	Word Alignment. Specifies whether the Word Alignment Module is enabled or disabled. 1'b0 – Word Alignment Module is enabled. 1'b1 – Word Alignment Module is disabled and the input data is bypassed.	RW	1	*Word Alignment*

Field	Name	Description	Access	Width	Reset
[4]	align_2byte_dis	2-Byte Alignment Pattern. Specifies where the pattern appears in the data bus. 1'b0 – In a 2-byte internal data bus with mode, always put the LSByte of the word alignment pattern to [9:0] of the data bus. 1'b1 – The pattern could appear at either [9:0] or [19:10] of the data bus.	RW	1	*Use LSByte of the Word Alignment*
[3]	sec_waptn_dis	Pattern Matching Enable. Specifies whether the secondary word alignment pattern matching is enabled or disabled. 1'b0 – Secondary Word Alignment Pattern Matching is enabled. 1'b1 – Secondary Word Alignment Pattern Matching is disabled.	RW	1	*Secondary Word Alignment*
[2]	wa_ptn_20b	Word Align Pattern Bit Width. Specifies the bit width that is used for the word alignment pattern. 1'b0 – Word Alignment Pattern is 10-bit width; only [9:0] of the 20-bit pattern and mask code are used. 1'b1 – Word Alignment Pattern is 20-bit width.	RW	1	*Word Alignment Bit Width*
[1]	syncdet_fsm_dis	Sync_Det FSM. Specifies whether the "sync_det" FSM is used or disabled. 1'b0 – Use "sync_det" FSM. 1'b1 – Disable "sync_det" FSM.	RW	1	*Use sync_det FSM*
[0]	auto_wa_dis	Word Align Mode. Specifies the use of automatic word alignment or the use of manual alignment. 1'b0 – Use the Automatic Word Alignment mode. 1'b1 – Use the Manual Alignment mode.	RW	1	*Automatic Word Alignment*

5.9. Primary Word Alignment Pattern Byte 0

Table 5.11. Primary Word Alignment Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_wa_ptn [7:0]	Specifies the 20-bit primary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	8	*Primary Word Alignment Pattern Value[7:0]*

5.10. Primary Word Alignment Pattern Byte 1

Table 5.12. Primary Word Alignment Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_wa_ptn [17:10]	Specifies the 20-bit primary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	8	*Primary Word Alignment Pattern Value[17:10]*

5.11. Primary Word Alignment Pattern MSB

Table 5.13. Primary Word Alignment Pattern MSB Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:2]	pri_wa_ptn[19:18]	Specifies the 20-bit primary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	2	*Primary Word Alignment Pattern Value[19:18]*
[1:0]	pri_wa_ptn[9: 8]	Specifies the 20-bit primary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	2	*Primary Word Alignment Pattern Value[9:8]*

5.12. Secondary Word Alignment Pattern Byte 0

Table 5.14. Secondary Word Alignment Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_wa_ptn [7:0]	Specifies the 20-bit secondary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	8	*Secondary Word Alignment Pattern Value[7:0]*

5.13. Secondary Word Alignment Pattern Byte 1

Table 5.15. Secondary Word Alignment Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_wa_ptn [17:10]	Specifies the 20-bit secondary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	8	*Secondary Word Alignment Pattern Value[17:0]*

5.14. Secondary Word Alignment Pattern MSB

Table 5.16. Secondary Word Alignment Pattern MSB Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:2]	sec_wa_ptn[19:18]	Specifies the 20-bit secondary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	2	*Secondary Word Alignment Pattern Value[19:18]*
[1:0]	sec_wa_ptn[9: 8]	Specifies the 20-bit secondary word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied.	RW	2	*Secondary Word Alignment Pattern Value[9:8]*

5.15. Word Alignment Pattern Mask Code Byte 0

Table 5.17. Word Alignment Pattern Mask Code Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	wa_mask_code [7:0]	Word Align Mask Mode. Specifies the 20-bit word alignment pattern. In the 10-bit width mode, only bits 9 to 0 are applied. 1'b0 – The corresponding bit of the word alignment pattern is ignored during alignment pattern matching. 1'b1 – The corresponding bit of the word alignment pattern is not ignored during alignment pattern matching.	RW	8	8'h0

5.16. Word Alignment Pattern Mask Code Byte 1

Table 5.18. Word Alignment Pattern Mask Code Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	wa_mask_code [17:10]	Word Align Mask Mode. Specifies the 20-bit word alignment pattern. In the 10-bit width mode, only bits 9 to 0 are applied. 1'b0 – The corresponding bit of the word alignment pattern is ignored during alignment pattern matching. 1'b1 – The corresponding bit of the word alignment pattern is not ignored during alignment pattern matching.	RW	8	8'h0

5.17. Word Alignment Pattern Mask Code MSB

Table 5.19. Word Alignment Pattern Mask Code MSB Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:2]	wa_mask_code[19:18]	Word Align Mask Mode. Specifies the 20-bit word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied. 1'b0 – The corresponding bit of the word alignment pattern is ignored during alignment pattern matching. 1'b1 – The corresponding bit of the word alignment pattern is not ignored during alignment pattern matching.	RW	2	2'h0
[1:0]	wa_mask_code [9: 8]	Word Align Mask Mode. Specifies the 20-bit word alignment pattern. In 10-bit width mode, only bits 9 to 0 are applied. 1'b0 – The corresponding bit of the word alignment pattern is ignored during alignment pattern matching. 1'b1 – The corresponding bit of the word alignment pattern is not ignored during alignment pattern matching.	RW	2	2'h0

5.18. sync_det FSM Configuration 0

Table 5.20. sync_det FSM Configuration 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	num_cal_sync	Specifies the number of valid synchronization code groups or ordered sets that *sync_det* FSM must receive to achieve synchronization state.	RW	8	*Number of Valid Sync Code Groups*

5.19. sync_det FSM Configuration 1

Table 5.21. sync_det FSM Configuration 1 Register

Field	Name	Description	Access	Width	Reset
[7:6]	reserved	—	RW	2	—
[5:0]	num_bad_code	Specifies the number of bad code groups received by *sync_det* FSM to conclude the loss of synchronization.	RW	6	*Number of Bad Code Groups*

5.20. sync_det FSM Configuration 2

Table 5.22. sync_det FSM Configuration 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	num_good_code	Specifies the continuous good code groups received by *sync_det* FSM to reduce the error count by one.	RW	8	*Number of Good Code Groups*

5.21. sync_det FSM Configuration 3

Table 5.23. sync_det FSM Configuration 3 Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	3	—
4	sec_sync_ptn_dis	Secondary Sync Detect Pattern. Specifies whether the secondary detect pattern is enabled or disabled. 1'b0 – Use the secondary sync_det pattern for matching. 1'b1 – Disable the secondary sync_det pattern and do not use it for matching.	RW	1	*Secondary Sync Pattern*
3	sync_ptn_10b	Sync Detect Pattern. Specifies the pattern code. 1'b0 – The sync_det pattern is 8b code. 1'b1 – The sync_det pattern is 10b code.	RW	1	*Sync Pattern Code*
2	sync_ptn_align	Pattern Alignment check. Specifies whether the pattern alignment check is enabled or disabled. 1'b0 – Disable the sync detect pattern alignment check. 1'b1 – The first byte of sync_det pattern must appear on N-byte boundary, where N is the sync_det pattern length defined by the sync_ptn_len register. Once a sync_det pattern is detected, the sync_det FSM applies this creation to the validity check of subsequent incoming data.	RW	1	*Sync Pattern Alignment*

Field	Name	Description	Access	Width	Reset
[1:0]	sync_ptn_len	Pattern Length. Specifies the length of the pattern. 2'b00 – The length is 1. 2'b01 – The length is 2. 2'b1x – The length is 4.	RW	2	*sync_det Pattern Length*

5.22. Number of Bit Slipped During Word Alignment

Table 5.24. Number of Bit Slipped Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	3	—
[4:0]	num_bit_slipped	Code Group Length. Specifies the length of the code group. 5'b00000 – There is no bit slipped. 5'b00001 – 1 bit is slipped. 5'b00010 – 2 bits are slipped. … 5'b10011 – 19 bits are slipped. *19 is the maximum bit number.*	RO	5	5'b0

5.23. Primary sync_det Pattern MSB

Table 5.25. Primary sync_det Pattern Byte MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	pri_sdptn_byte3 [9:8]	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	2	*Primary sync_det Pattern Byte 3[9:8]*
[5:4]	pri_sdptn_byte2 [9:8]	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	2	*Primary sync_det Pattern Byte 2[9:8]*
[3:2]	pri_sdptn_byte1 [9:8]	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	2	*Primary sync_det Pattern Byte 1[9:8]*
[1:0]	pri_sdptn_byte0 [9:8]	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	2	*Primary sync_det Pattern Byte 0[9:8]*

5.24. Primary sync_det Pattern Byte 3

Table 5.26. Primary sync_det Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_sdptn_byte3	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	8	*Primary sync_det Pattern Byte 3[7:0]*

5.25. Primary sync_det Pattern Byte 2

Table 5.27. Primary sync_det Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_sdptn_byte2	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	8	*Primary sync_det Pattern Byte 2[7:0]*

5.26. Primary sync_det Pattern Byte 1

Table 5.28. Primary sync_det Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_sdptn_byte1	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	8	*Primary sync_det Pattern Byte 1[7:0]*

5.27. Primary sync_det Pattern Byte 0

Table 5.29. Primary sync_det Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_sdptn_byte0	The Primary sync_det Pattern MSB Register reflects the bits 9 to 8 of the primary sync_det pattern byte 3 to 0.	RW	8	*Primary sync_det Pattern Byte 0[7:0]*

5.28. Secondary sync_det Pattern MSB

Table 5.30. Secondary sync_det Pattern Byte MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	sec_sdptn_byte3 [9:8]	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	2	*[7:6]Secondary sync_det Pattern Byte 3[9:8]*
[5:4]	sec_sdptn_byte2 [9:8]	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	2	*[5:4]Secondary sync_det Pattern Byte 2[9:8]*
[3:2]	sec_sdptn_byte1 [9:8]	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	2	*[3:2]Secondary sync_det Pattern Byte 1[9:8]*
[1:0]	sec_sdptn_byte0 [9:8]	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	2	*[1:0]Secondary sync_det Pattern Byte 0[9:8]*

5.29. Secondary sync_det Pattern Byte 3

Table 5.31. Secondary sync_det Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_sdptn_byte3	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	8	*Secondary sync_det Pattern Byte 3[7:0]*

5.30. Secondary sync_det Pattern Byte 2

Table 5.32. Secondary sync_det Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_sdptn_byte2	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	8	*Secondary sync_det Pattern Byte 2[7:0]*

5.31. Secondary sync_det Pattern Byte 1

Table 5.33. Secondary sync_det Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_sdptn_byte1	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	8	*Secondary sync_det Pattern Byte 1[7:0]*

5.32. Secondary sync_det Pattern Byte 0

Table 5.34. Secondary sync_det Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_sdptn_byte0	The Secondary sync_det Pattern MSB Register reflects the bits 9 to 8 of the secondary sync_det pattern byte 3 to 0.	RW	8	*Secondary sync_det Pattern Byte 0[7:0]*

5.33. sync_det Pattern Mask Code MSB

Table 5.35. Sync_det Pattern Mask Code Register

Field	Name	Description	Access	Width	Reset
[7:6]	sdptn_mask_byte3[9:8]	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	2	2'h0
[5:4]	sdptn_mask_byte2[9:8]	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	2	2'h0
[3:2]	sdptn_mask_byte1[9:8]	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	2	2'h0
[1:0]	sdptn_mask_byte0[9:8]	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	2	2'h0

5.34. sync_det Pattern Mask Code Byte 3

Table 5.36. sync_det Pattern Mask Code Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sdptn_mask_byte3	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	8	8'h0

5.35. sync_det Pattern Mask Code Byte 2

Table 5.37. sync_det Pattern Mask Code Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sdptn_mask_byte2	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	8	8'h0

5.36. sync_det Pattern Mask Code Byte 1

Table 5.38. sync_det Pattern Mask Code Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sdptn_mask_byte1	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	8	8'h0

5.37. sync_det Pattern Mask Code Byte 0

Table 5.39. sync_det Pattern Mask Code Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sdptn_mask_byte0	The sync_det Pattern Mask Code MSB Register reflects the bits 9 to 8 of the sync_det pattern mask code byte 3 to 0.	RW	8	8'h0

5.38. Lane Alignment Control

Table 5.40. Lane Alignment Control Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	3	—
[4]	sec_laptn_en	Secondary Lane Alignment. Specifies whether the secondary lane alignment pattern matching is enabled or not. 1'b0 – Pattern Matching is disabled. 1'b1 – Pattern Matching is enabled.	RW	1	*[4]Secondary Lane Alignment*
[3:2]	lalign_ptn_len	Lane Alignment Pattern Length. Specifies the lane alignment pattern length in byte. 2'b00 – 1-byte 2'b01 – 2-byte 2'b1x – 4-byte	RW	2	*[3:2]Lane Alignment Pattern Length*

Field	Name	Description	Access	Width	Reset
[1]	lalign_10b	Lane Alignment Coding Mode. Specifies the Lane Alignment coding scheme of the input data. 1'b0 – Input data is in 8b code mode. 1'b1 – input data is in 10b code mode.	RW	1	*[1] Input Data Code Mode*
[0]	lalign_en	Lane Alignment Enable. Specifies whether the lane alignment is enabled or not. 1'b0 – Lane Alignment is disabled. 1'b1 – Lane Alignment is enabled.	RW	1	*[0]Lane Alignment*

5.39. Maximum Lane-to-lane Skew

Table 5.41. Maximum Lane-to-lane Skew Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	6	—
[3:0]	max_lskew	Maximum Lane-to-lane skew. Specifies the maximum lane-to-lane skew in byte. 4'd1 – 1-byte skew. 4'd2 – 2-byte skew. … 4'd10 – 10-byte skew.	RW	4	*Maximum Lane-to-Lane Skew*

5.40. Primary Lane Alignment Pattern MSB

Table 5.42. Primary Lane Alignment Pattern Byte MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	pri_laptn_byte3[9:8]	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	2	*Primary Lane Alignment Pattern Byte 3[9:8]*
[5:4]	pri_laptn_byte2[9:8]	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	2	*Primary Lane Alignment Pattern Byte 2[9:8]*
[3:2]	pri_laptn_byte1[9:8]	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	2	*Primary Lane Alignment Pattern Byte 1[9:8]*
[1:0]	pri_laptn_byte0[9:8]	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	2	*Primary Lane Alignment Pattern Byte 0[9:8]*

5.41. Primary Lane Alignment Pattern Byte 3

Table 5.43. Primary Lane Alignment Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_laptn_byte3	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	8	*Primary Lane Alignment Pattern Byte 3[7:0]*

5.42. Primary Lane Alignment Pattern Byte 2

Table 5.44. Primary Lane Alignment Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_laptn_byte2	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	8	*Primary Lane Alignment Pattern Byte 2[7:0]*

5.43. Primary Lane Alignment Pattern Byte 1

Table 5.45. Primary Lane Alignment Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_laptn_byte1	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	8	*Primary Lane Alignment Pattern Byte 1[7:0]*

5.44. Primary Lane Alignment Pattern Byte 0

Table 5.46. Primary Lane Alignment Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_laptn_byte0	The Primary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the primary lane alignment pattern byte 3 to 0.	RW	8	*Primary Lane Alignment Pattern Byte 0[7:0]*

5.45. Lane Alignment Mask Code

Table 5.47. Lane Alignment Mask Code Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:0]	lalign_mask_code	Lane Alignment Mask Code. Specifies the bit 3 to 0 of the Lane Alignment Pattern Mask Code1'b0 – Lane Alignment is disabled. 1'b1 – the corresponding byte of lane alignment pattern is ignored during alignment pattern matching. 1'b0 – the corresponding byte of lane alignment pattern is not ignored during alignment pattern matching.	RW	4	*Lane Alignment Pattern Mask Code*

5.46. Secondary Lane Alignment Pattern MSB

Table 5.48. Secondary Lane Alignment Pattern Byte MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	sec_laptn_byte3[9:8]	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	2	2'h0
[5:4]	sec_laptn_byte2[9:8]	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	2	2'h0
[3:2]	sec_laptn_byte1[9:8]	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	2	2'h0
[1:0]	sec_laptn_byte0[9:8]	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	2	2'h1

5.47. Secondary Lane Alignment Pattern Byte3

Table 5.49. Secondary Lane Alignment Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte3	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	8	8'h0

5.48. Secondary Lane Alignment Pattern Byte2

Table 5.50. Secondary Lane Alignment Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte2	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	8	8'h0

5.49. Secondary Lane Alignment Pattern Byte1

Table 5.51. Secondary Lane Alignment Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte1	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	8	8'h0

5.50. Secondary Lane Alignment Pattern Byte0

Table 5.52. Secondary Lane Alignment Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte0	The Secondary Lane Alignment Pattern MSB Register reflects the bits 9 to 8 of the secondary lane alignment pattern byte 3 to 0.	RW	8	8'h7C

5.51. Clock Frequency Compensation Control

Table 5.53. Clock Frequency Compensation Control Register

Field	Name	Description	Access	Width	Reset
[7:6]	reserved	—	RW	3	—
[5]	sec_skip_en	Enable Secondary Skip. Specifies whether the secondary skip pattern is enabled or not. 1'b0 – Pattern Matching is disabled. 1'b1 – Pattern Matching is enabled.	RW	1	1'b0
[4:3]	skip_ptn_len	Skip Pattern Length. Specifies the skip pattern length in bytes. 2'b00 – 1-byte 2'b01 – 2-byte 2'b1x – 4-bytes	RW	2	*Skip Pattern Length*
[2]	clk_comp_10b	Clock Compensation Coding Mode. Specifies the clock compensation coding scheme for the input data. 1'b0 – Input data is in 8b code mode. 1'b1 – Input data is in 10b code mode.	RW	1	1'b0
[1]	ctc_fifo_en	Clock Compensation FIFO. Specifies whether the Clock Compensation FIFO is enabled or not. 1'b0 – Clock Compensation FIFO is disabled. 1'b1 – Clock Compensation FIFO is enabled.	RW	1	*CTC FIFO*
[0]	clk_comp_en	Enable Clock Frequency Compensation. Specifies whether the clock frequency compensation is enabled or not. 1'b0 – Clock Frequency Compensation is disabled. 1'b1 – Clock Frequency Compensation is enabled.	RW	1	*Clock Frequency Compensation*

Note: The combination of *ctc_fifo_en* and *clk_comp_en* are listed below: {ctc_fifo_en, clk_comp_en}

- 2'b0x – bypass the input data of this module
- 2'b10 – the module works as an asynchronous FIFO without clock frequency compensation
- 2'b11 – the module performs the clock frequency compensation function

5.52. SKIP Pattern Insertion/Deletion Control

Table 5.54. SKIP Pattern Insertion/Deletion Control Register

Field	Name	Description	Access	Width	Reset
[7]	reserved	—	RW	1	—
[6:2]	ctc_repeat_wait	Clock Correction Frequency Control. Defines the minimum number of clock cycles between two clock correction events, including SKIP deletion and insertion. 1'b0 – On constraint on SKIP deletion and insertion.	RW	5	5'b0
[1:0]	min_ipg_cnt	Minimum SKIP pattern count. Used to guarantee the minimum number of bytes between packets (that is minimum Inter Packet Gap) after SKIP deletion.	RW	2	2'b0

Field	Name	Description	Access	Width	Reset
		2'b00 – No constraint on SKIP deletion. 2'b01 – At least one SKIP pattern is kept in IPG. 2'b10 – At least two SKIP patterns are kept in IPG. 2'b11 – At least three SKIP patterns are kept in IPG.			

Note: The length in byte of SKIP pattern is defined in *skp_ptn_len* domain register *Clock Frequency Compensation Control*.

5.53. Elastic FIFO High Water Line

Table 5.55. Elastic FIFO High Water Line Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	3	—
[4:0]	high_water_line	Specifies the Clock Compensation FIFO high water line. Mean is 5'b10000.	RW	5	5'b10110

5.54. Elastic FIFO Low Water Line

Specifies the Clock Compensation FIFO low water line. Mean is 5'b10000.

Table 5.56. Elastic FIFO Low Water Line Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	3	—
[4:0]	low_water_line	Specifies the Clock Compensation FIFO low water line. Mean is 5'b10000.	RW	5	5'b01010

5.55. Primary SKIP Pattern MSB

Table 5.57. Primary SKIP Pattern MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	pri_skip_byte3[9:8]	—	RW	2	*Primary Skip Pattern Byte 3[9:8]*
[5:4]	pri_skip_byte2[9:8]	—	RW	2	*Primary Skip Pattern Byte 2[9:8]*
[3:2]	pri_skip_byte1[9:8]	—	RW	2	*Primary Skip Pattern Byte 1[9:8]*
[1:0]	pri_skip_byte0[9:8]	—	RW	2	*Primary Skip Pattern Byte 0[9:8]*

5.56. Primary SKIP Pattern Byte 3

Table 5.58. Primary SKIP Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_skip_byte3[7:0]	—	RW	8	*Primary Skip Pattern Byte 3[7:0]*

5.57. Primary SKIP Pattern Byte 2

Table 5.59. Primary SKIP Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_skip_byte2[7:0]	—	RW	8	*Primary Skip Pattern Byte 2[7:0]*

5.58. Primary SKIP Pattern Byte 1

Table 5.60. Primary SKIP Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_skip_byte1[7:0]	—	RW	8	*Primary Skip Pattern Byte 1[7:0]*

5.59. Primary SKIP Pattern Byte 0

Table 5.61. Primary SKIP Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	pri_skip_byte0[7:0]	—	RW	8	*Primary Skip Pattern Byte 0[7:0]*

5.60. SKIP Pattern Mask Code

Table 5.62. SKIP Pattern Mask Code Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:0]	skp_mask_code	SKIP Pattern Mask Code. Specifies whether the SKIP Pattern is ignored or not. 1'b0 – The corresponding byte of the SKIP pattern is not ignored during SKIP pattern matching. 1'b1 – The corresponding byte of the SKIP pattern is ignored during SKIP pattern matching.	RW	4	*Skip Pattern Mask Code*

5.61. Secondary SKIP Pattern MSB

Table 5.63. Secondary SKIP Pattern MSB Register

Field	Name	Description	Access	Width	Reset
[7:6]	sec_skip_byte3[9:8]	—	RW	2	2'h0
[5:4]	sec_skip_byte2[9:8]	—	RW	2	2'h0
[3:2]	sec_skip_byte1[9:8]	—	RW	2	2'h0
[1:0]	sec_skip_byte0[9:8]	—	RW	2	2'h1

5.62. Secondary SKIP Pattern Byte3

Table 5.64. Secondary SKIP Pattern Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte3[7:0]	—	RW	8	8'h0

5.63. Secondary SKIP Pattern Byte2

Table 5.65. Secondary SKIP Pattern Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte2[7:0]	—	RW	8	8'h0

5.64. Secondary SKIP Pattern Byte1

Table 5.66. Secondary SKIP Pattern Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte1[7:0]	—	RW	8	8'h0

5.65. Secondary SKIP Pattern Byte0

Table 5.67. Secondary SKIP Pattern Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	sec_laptn_byte0[7:0]	—	RW	8	8'h7C

5.66. 64b/66b PCS TX Path Control

Table 5.68. 64b/66b PCS TX Path Control Register

Field	Name	Description	Access	Width	Reset
[7:2]	reserved	—	RW	6	—
[1]	end_64b66b_dis	Enable 64b/66b Encoder. Specifies whether the 64b/66b Encoder is enabled or disabled. 1'b0 – 64b/66b Encoder is enabled. 1'b1 – 64b/66b Encoder is disabled.	RW	1	*64b66b Encoder*
[0]	src_64b66b_dis	Enable 64b/66b Scrambler. Specifies whether the 64b/66b Scrambler is enabled or disabled. 1'b0 – 64b/66b Scrambler is enabled. 1'b1 – 64b/66b Scrambler is disabled.	RW	1	*Scrambler*

5.67. 64b/66b PCS TX FIFO Almost Full Setting

Table 5.69. 64b/66b PCS TX FIFO Almost Full Setting Control Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:0]	tx_fifo_af	64b/66b TX FIFO Almost Full. When the number of blocks residing in the FIFO is more than this setting, the *almost full* status is reported.	RW	4	*TX FIFO Almost Full*

5.68. 64b/66b PCS TX FIFO Almost Empty Setting

Table 5.70. 64b/66b PCS TX FIFO Almost Empty Setting Control Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3:0]	tx_fifo_ae	64b/66b TX FIFO Almost Empty. When the number of blocks residing in the FIFO is less than this setting, the *almost empty* status is reported.	RW	4	*TX FIFO Almost Empty*

5.69. 64b/66b PCS RX Path Control

Table 5.71. 64b/66b PCS TX Path Control Register

Field	Name	Description	Access	Width	Reset
[7:6]	reserved	—	RW	2	—
[5]	del_os_cont	Delete Ordered Set. Specifies whether to delete one sequence of an ordered set each time or delete continuously. 1'b0 – Allow deleting one sequence ordered set each time when doing clock frequency compensation. 1'b1 – Allow continuously deleting sequence ordered sets when doing clock frequency compensation.	RW	1	1'b0
[4]	pcs_64b66b_nofpll	Use Fabric PLL. Specifies whether to use the clock from Fabric PLL or the clock derived from MPCS. 1'b0 – FPLL is used to generate user clock. 1'b1 – No FPLL, user clock is derived from tx_out_clk of MPCS.	RW	1	*Use External PLL*
[3]	balign_64b66b_dis	Enable 64b/66b Block Aligner. Specifies whether the 64b/66b block aligner is enabled or disabled. 1'b0 – 64b/66b block aligner is enabled. 1'b1 – 64b/66b block aligner is disabled.	RW	1	*Block Aligner*
[2]	ctc_64b66b_dis	Enable 64b/66b CTC. Specifies whether the 64b/66b Clock Frequency Compensation is enabled or disabled. 1'b0 – 64b/66b Clock Frequency Compensation is enabled. 1'b1 – 64b/66b Clock Frequency Compensation is disabled.	RW	1	*RX FIFO*

Field	Name	Description	Access	Width	Reset
[1]	dec_64b66b_dis	Enable 64b/66b Decoder. Specifies whether the 64b/66b Decoder is enabled or disabled. 1'b0 – 64b/66b Decoder is enabled. 1'b1 – 64b/66b Decoder is disabled.	RW	1	*64b66b Decoder*
[0]	descr_64b66b_dis	Enable 64b/66b Descrambler. Specifies whether the 64b/66b Descrambler is enabled or disabled. 1'b0 – 64b/66b Descrambler is enabled. 1'b1 – 64b/66b Descrambler is disabled.	RW	1	*Descrambler*

5.70. 64b/66b PCS CTC High Water Line

Table 5.72. 64b/66b PCS CTC High Water Line Control Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	4	—
[4:0]	ctc_64b66b_high	The 64b/66b PCS CTC High water line reflects the high water line of clock frequency compensation.	RW	4	4'h0C

5.71. 64b/66b PCS CTC Low Water Line

The 64b/66b PCS CTC Low water line reflects the low water line of clock frequency compensation.

Table 5.73. 64b/66b PCS CTC Low Water Line Control Register

Field	Name	Description	Access	Width	Reset
[7:5]	reserved	—	RW	4	—
[4:0]	ctc_64b66b_low	The 64b/66b PCS CTC Low water line reflects the low water line of clock frequency compensation.	RW	4	4'h04

5.72. 64b/66b PCS Block Align Shift

Table 5.74. 64b/66b PCS Block Align Shift Register

Field	Name	Description	Access	Width	Reset
[7]	reserved	—	RW	1	—
[6:0]	balign_64b66b_shift	The 64b/66b PCS Block Align Shift register reflects the bit shifting of block alignment.	RO	7	7'h0

5.73. 10GBASE-R BER Counter

Table 5.75. 10GBASE-R BER Counter Register

Field	Name	Description	Access	Width	Reset
[7:6]	reserved	—	RW	1	—
[5:0]	ber_count	The 10GBASE-R BER Counter register reflects the BER counter is a 6-bit count as defined by the ber_count variable in 49.2.14.2 for	RO/CR	6	6'h0

Field	Name	Description	Access	Width	Reset
		10GBASE-R. These bits are reset to all zeros when the register is read or upon execution of the PCS reset. These bits are held at all ones in the case of overflow.			

5.74. 10GBASE-R Block Error Counter

Table 5.76. 10GBASE-R Block Error Counter Register

Field	Name	Description	Access	Width	Reset
[7:0]	errored_block_count	The 10GBASE-R Block Error Counter register reflects the errored blocks counter is an eight-bit count defined by the errored_block_count counter specified in 49.2.14.2 for 10GBASE-R. These bits are reset to all zeros when the errored blocks count is read or upon execution of the PCS reset. These bits are held at all ones in the case of overflow.	RO/CR	8	8'h0

5.75. 10GBASE-R Test Pattern Seed A Byte7

Table 5.77. 10GBASE-R Test Pattern Seed A Byte 7 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[57:56]	This register defines byte 7 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.76. 10GBASE-R Test Pattern Seed A Byte6

Table 5.78. 10GBASE-R Test Pattern Seed A Byte 6 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[55:48]	This register defines byte 6 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.77. 10GBASE-R Test Pattern Seed A Byte5

Table 5.79. 10GBASE-R Test Pattern Seed A Byte 5 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[47:40]	This register defines byte 5 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.78. 10GBASE-R Test Pattern Seed A Byte4

Table 5.80. 10GBASE-R Test Pattern Seed A Byte 4 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[39:32]	This register defines byte 4 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.79. 10GBASE-R Test Pattern Seed A Byte3

Table 5.81. 10GBASE-R Test Pattern Seed A Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[31:24]	This register defines byte 3 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.80. 10GBASE-R Test Pattern Seed A Byte2

Table 5.82. 10GBASE-R Test Pattern Seed A Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[23:16]	This register defines byte 2 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.81. 10GBASE-R Test Pattern Seed A Byte1

Table 5.83. 10GBASE-R Test Pattern Seed A Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[15:8]	This register defines byte 1 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.82. 10GBASE-R Test Pattern Seed A Byte0

Table 5.84. 10GBASE-R Test Pattern Seed A Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_a[7:0]	This register defines the byte 0 of 10GBASE-R PCS test pattern seed A.	RW	8	8'h0

5.83. 10GBASE-R Test Pattern Seed B Byte7

Table 5.85. 10GBASE-R Test Pattern Seed B Byte 7 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[57:56]	This register defines byte 7 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.84. 10GBASE-R Test Pattern Seed B Byte6

Table 5.86. 10GBASE-R Test Pattern Seed B Byte 6 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[55:48]	This register defines byte 6 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.85. 10GBASE-R Test Pattern Seed B Byte5

Table 5.87. 10GBASE-R Test Pattern Seed B Byte 5 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[47:40]	This register defines byte 5 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.86. 10GBASE-R Test Pattern Seed B Byte4

Table 5.88. 10GBASE-R Test Pattern Seed B Byte 4 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[39:32]	This register defines byte 4 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.87. 10GBASE-R Test Pattern Seed B Byte3

Table 5.89. 10GBASE-R Test Pattern Seed B Byte 3 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[31:24]	This register defines byte 3 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.88. 10GBASE-R Test Pattern Seed B Byte2

Table 5.90. 10GBASE-R Test Pattern Seed B Byte 2 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[23:16]	This register defines byte 2 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.89. 10GBASE-R Test Pattern Seed B Byte1

Table 5.91. 10GBASE-R Test Pattern Seed B Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[15:8]	This register defines byte 1 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.90. 10GBASE-R Test Pattern Seed B Byte0

Table 5.92. 10GBASE-R Test Pattern Seed B Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	prtp_seed_b[7:0]	This register defines byte 0 of 10GBASE-R PCS test pattern seed B.	RW	8	8'h0

5.91. 10GBASE-R Test Pattern Control 0

Table 5.93. 10GBASE-R Test Pattern Control 0 Register

Field	Name	Description	Access	Width	Reset
[7:5]	tx_sw_pattern	TX Square Wave Pattern. Defines the square wave repeating pattern of n ones followed by n zeros where n is configurable between 4 and 11. 3'b0 – n = 4 3'b1 – n = 5 3'b2 – n = 6 3'b3 – n = 8 3'b4 – n = 10 3'b5 – n = 11 3'b6 – reserved 3'b7 – reserved	RW	3	3'b0
[4]	tx_prbs9_en	TX PRBS9 Test-Pattern mode. Specifies whether the PRBS9 Test-Pattern mode is enabled or disabled. 1'b0 – PRBS9 test-pattern mode on the TX path is disabled. 1'b1 – PRBS9 test-pattern mode on the TX path is enabled.	RW	1	1'b0
[3]	tx_prbs31_en	TX PRBS31 Test-Pattern mode. Specifies whether the PRBS31 Test-Pattern mode is enabled or disabled. 1'b0 – PRBS31 test-pattern mode on the TX path is disabled. 1'b1 – PRBS31 test-pattern mode on the TX path is enabled.	RW	1	1'b0
[2]	tx_prtp_en	TX test pattern. Specifies whether the TX test pattern is enabled or disabled. 1'b0 – Transmit test pattern is disabled. 1'b1 – Transmit test pattern is enabled.	RW	1	1'b0
[1]	tx_prtp_test_sel	TX test pattern Selector. Specifies the selected test pattern on the TX path. 1'b0 – Pseudo random test pattern. 1'b1 – Square wave test pattern.	RW	1	1'b0
[0]	tx_prtp_data_sel	TX Data Pattern Selector. Specifies the selected data pattern on the TX path. 1'b0 – LF data pattern. 1'b1 – Zeros data pattern.	RW	1	1'b0

5.92. 10GBASE-R Test Pattern Control 1

Table 5.94. 10GBASE-R Test Pattern Control 1 Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	—
[3]	rx_prbs31_en	RX PRBS31 Test-Pattern Mode. Specifies whether the PRBS31 Test-Pattern mode is enabled or disabled. 1'b0 – PRBS31 test-pattern mode on the RX path is disabled. 1'b1 – PRBS31 test-pattern mode on the RX	RW	1	1'b0

Field	Name	Description	Access	Width	Reset
		path is enabled.			
[2]	rx_prtp_en	RX Test Pattern. Specifies whether the RX test pattern is enabled or disabled. 1'b0 – Receive test pattern is disabled. 1'b1 – Receive test pattern is enabled.	RW	1	1'b0
[1]	rx_prtp_test_sel	RX Test Pattern Selector. Specifies the selected test pattern on the RX path. 1'b0 – Pseudo random test pattern. 1'b1 – Square wave test pattern.	RW	1	1'b0
[0]	rx_prtp_data_sel	RX Data Pattern Selector. Specifies the selected data pattern on the RX path. 1'b0 – LF data pattern. 1'b1 – Zeros data pattern.	RW	1	1'b0

5.93. 10GBASE-R Test Pattern Error Counter Byte0

Table 5.95. 10GBASE-R Test Pattern Error Counter Byte 0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	tp_error_cnt[7:0]	The 10GBASE-R Test Pattern Error Counter Byte 0 register reflects the counter contains the number of errors received during a pattern test. These bits are reset to all zeros when the test-pattern error counter is read or upon execution of the PCS reset. These bits are held at all ones in the case of overflow.	RO/CR	8	8'h0

5.94. 10GBASE-R Test Pattern Error Counter Byte1

Table 5.96. 10GBASE-R Test Pattern Error Counter Byte 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	tp_error_cnt[15:8]	The 10GBASE-R Test Pattern Error Counter Byte 1 register reflects the counter contains the number of errors received during a pattern test. These bits are reset to all zeros when the test-pattern error counter is read or upon execution of the PCS reset. These bits are held at all ones in the case of overflow.	RO/CR	8	8'h0

5.95. Loop Back Mode Control

Table 5.97. Loop Back Mode Control Register

Field	Name	Description	Access	Width	Reset
[7:3]	reserved	—	RW	5	—
[2]	fbrc_lpbck_en	64b/66b Loopback B. Specifies whether the PCS Loopback B for 64b/66b is enabled or disabled. 1'b0 – 64b/66b PCS Loopback B is disabled. 1'b1 – 64b/66b PCS Loopback B is enabled.	RW	1	1'b0

Field	Name	Description	Access	Width	Reset
[1]	far_lpbk_en	Far end Parallel Loopback. Specifies whether the Far-End Parallel Loopback is enabled or disabled. 1'b0 – Far end parallel loopback is disabled. 1'b1 – Far end parallel loopback is enabled.	RW	1	1'b0
[0]	near_lpbk_en	Near End Parallel Loopback. Specifies whether the Near End Parallel Loopback is enabled or disabled. 1'b0 – Near end parallel loopback is disabled. 1'b1 – Near end parallel loopback is enabled.	RW	1	1'b0

5.96. MPCS BIST Control 1

Table 5.98. MPCS BIST Control 1 Register

Field	Name	Description	Access	Width	Reset
[7:6]	bist_ch_sel	BIST Channel Selector. Specifies the selected BIST Channel. 2'b00 – MPCS channel_0 2'b01 – MPCS channel_1 2'b10 – MPCS channel_2 2'b11 – MPCS channel_3	RW	2	2'b0
[5:4]	bist_res_sel	BIST Resolution Selector. Specifies the selected BIST resolution. 2'b00 – No error. 2'b01 – Less than 2 errors. 2'b10 – Less than 16 errors. 2'b11 – Less than 128 errors.	RW	2	2'b0
[3:2]	bist_time_sel	BIST Time Selector. Specifies the selected BIST Time. 2'b00 – 5e+8 cycles. 2'b01 – 5e+9 cycles. 2'b10 – 5e+6 cycles. 2'b11 – 100k cycles.	RW	2	2'b0
[1:0]	bist_sync_head_reg	BIST Sync Header Selector. Specifies the selected BIST sync header counter selection. 2'b00 – 5 2'b01 – 8 2'b10 – 14 2'b11 – 24	RW	2	2'b0

5.97. MPCS BIST Control 0

Table 5.99. MPCS BIST Control 0 Register

Field	Name	Description	Access	Width	Reset
[7:5]	bist_ptn_sel	BIST Pattern Selection. Specifies the selected BIST Pattern. 3'b000 – PRBS7 3'b001 – PRBS11 3'b010 – PRBS23 3'b011 – PRBS31 3'b100 – PRBS15	RW	3	3'b0

Field	Name	Description	Access	Width	Reset
		3'b101 – Constant Value 1 3'b110 – Constant Value 2 3'b111 – Alternate Constant Values 1 and 2.			
[4]	bist_rx_data_sel	Data receive selector. Specifies the selector where the data is received. 1'b0 – Data from PMA. 1'b1 – Data from after 8b10b decoder.	RW	1	1'b0
[3]	bist_bypass_tx_gate	Bypass Transmit gate. 1'b0 – Start to send BIST data after finding the head. 1'b1 – Force to send BIST data whether the head is found or not.	RW	1	1'b0
[2]	bist_bus8bit_sel	BIST bus width selector. 1'b0 – 10-bit BIST data. 1'b1 – 8-bit BIST data.	RW	1	1'b0
[1]	bist_mode	BIST mode. Specifies the BIST mode to be used. 1'b0 – Timed BIST mode. 1'b1 – Continuous BIST mode.	RW	1	1'b0
[0]	bist_en	Enable MPCS BIST. Specifies whether the MPCS BIST is enabled or not. 1'b0 – MPCS BIST is disabled. 1'b1 – MPCS BIST is enabled.	RW	1	1'b0

5.98. User Defined BIST Constant 1 MSByte

Table 5.100. User Defined BIST Constant 1 MSByte Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	4'h0
[3:2]	udbc[19:18]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	2	2'h0
[1:0]	udbc[9: 8]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	2	2'h0

5.99. User-Defined BIST Constant 1 Byte_1

Table 5.101. User-Defined BIST Constant 1 Byte_1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	udbc[7:0]	The User-defined BIST constant 1 Byte_1 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	8	8'h0

5.100.User-Defined BIST Constant 1 Byte_0

Table 5.102. User-Defined BIST Constant 1 Byte_0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	udbc[17:10]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	8	8'h0

5.101.User-Defined BIST Constant 2 MSB

Table 5.103. User-Defined BIST Constant 2 MSB Register

Field	Name	Description	Access	Width	Reset
[7:4]	reserved	—	RW	4	4'h0
[3:2]	udbc[19:18]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	2	2'h0
[1:0]	udbc[9: 8]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	2	2'h0

5.102.User-Defined BIST Constant 2 Byte_1

Table 5.104. User-Defined BIST Constant 2 Byte_1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	udbc2[7:0]	The User-defined BIST constant 1 Byte_1 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	8	8'h0

5.103.User-Defined BIST Constant 2 Byte_0

Table 5.105. User-Defined BIST Constant 2 Byte_0 Register

Field	Name	Description	Access	Width	Reset
[7:0]	udbc2[17:10]	The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.	RW	8	8'h0

5.104.BIST Status 1

Table 5.106. BIST Status 1 Register

Field	Name	Description	Access	Width	Reset
[7:0]	bist_err_cnt[7:0]	The BIST Status 1 register reflects the BIST error count.	RO	8	8'h0

5.105. BIST Status 0

The User-defined BIST constant 1 Byte_0 register reflects the lower 8 bits of the 10b user-defined BIST constant value pattern 1.

Table 5.107. BIST Status 0 Register

Field	Name	Description	Access	Width	Reset
[7:3]	bist_err_cnt[11:8]	BIST Error Count. Specifies the BIST error count.	RO	5	5'h0
[2]	bist_ok	BIST mode. Specifies the BIST mode to be used. 1'b0 – BIST is timed out or not started. 1'b1 – No error for bist_res_sel = 0; Less than 2 errors for bist_res_sel = 1; Less than 16 errors for bist_rel_sel = 2 and less than 128 errors for bist_res_rel = 3.	RO	1	1'h0
[1]	bist_done	BIST Done. Specifies the status of the BIST timer. 1'b0 – BIST is ongoing or not started. 1'b1 – BIST timer expires.	RO	1	1'h0
[0]	bist_time_out	BIST Time Out. Specifies the BIST timeout status. 1'b0 – BIST is ongoing or not started. 1'b1 – BIST is timed out but it is still in "sync header detection" stage.	RO	1	1'h0

6. Example Design

The MPCS module example design is available only for the simulation purposes. The example design is generated when the IP module is generated. Refer to Generation and Synthesis section to generate the IP module.

6.1. Running the Functional Simulation

You can run the functional simulation after the IP is generated.

To run functional simulation, follow these steps:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard**, as shown in Figure 6.1.

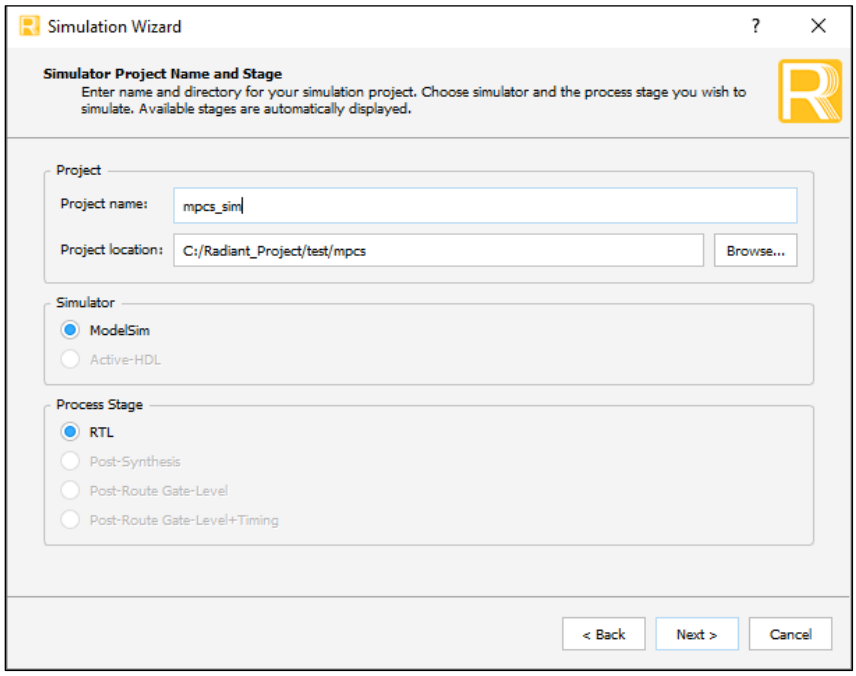

Figure 6.1. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window, as shown in Figure 6.2.

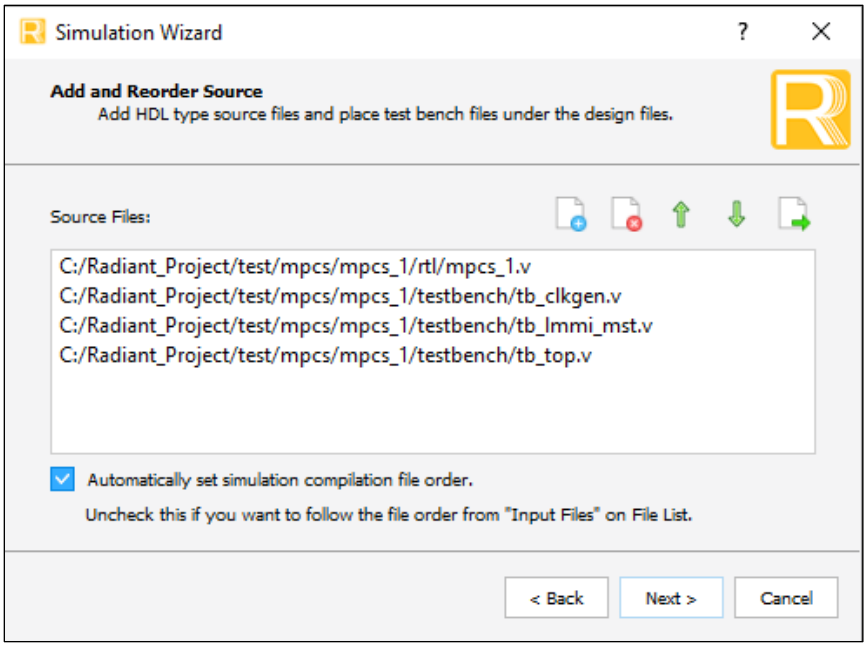

Figure 6.2. Adding and Reordering Source

3. Click **Next**.
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 simulation results of this example design are provided in Figure 6.3.

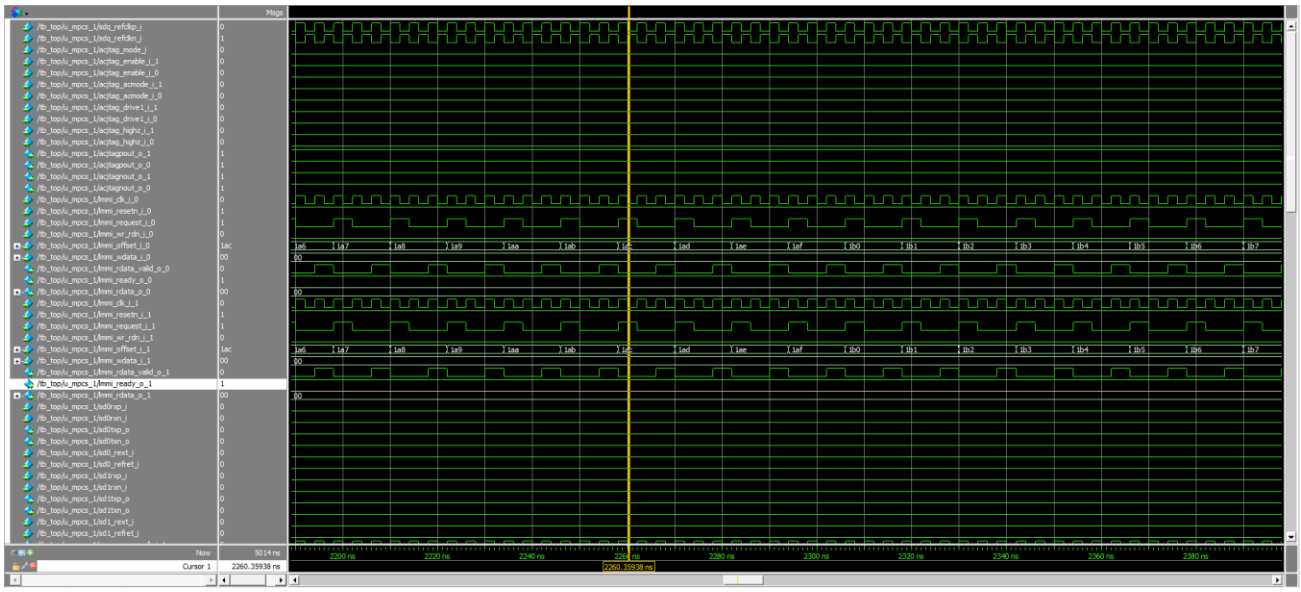

Figure 6.3. Simulation Waveform

7. Designing with the MPCS Module

This section provides information on how to generate the MPCS 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.

7.1. Generation and Synthesis

You can use the Lattice Radiant software to customize and generate modules and IPs and integrate them into the device architecture. The following steps describe on generating the MPCS module in Lattice Radiant software.

To generate the MPCS module, follow these steps:

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

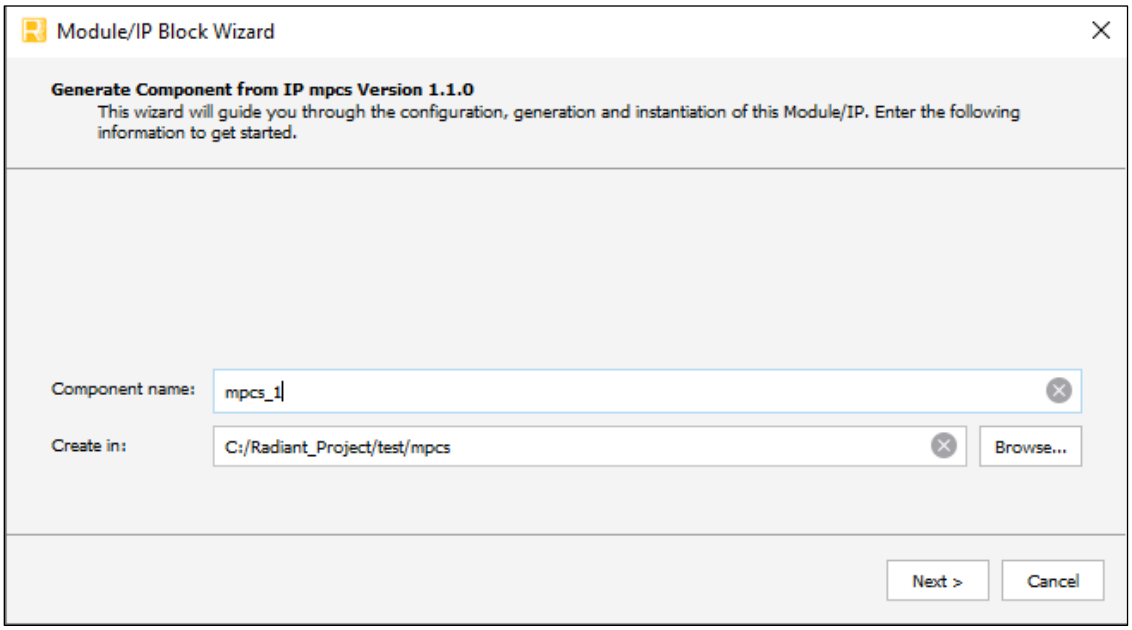

Figure 7.1. Module/IP Block Wizard

4. In the module dialog box of the **Module/IP Block Wizard** window, customize the selected MPCS module using the drop-down menus and check boxes. As a sample configuration, see Figure 7.2. For configuration options, see the Module Parameter Description section.

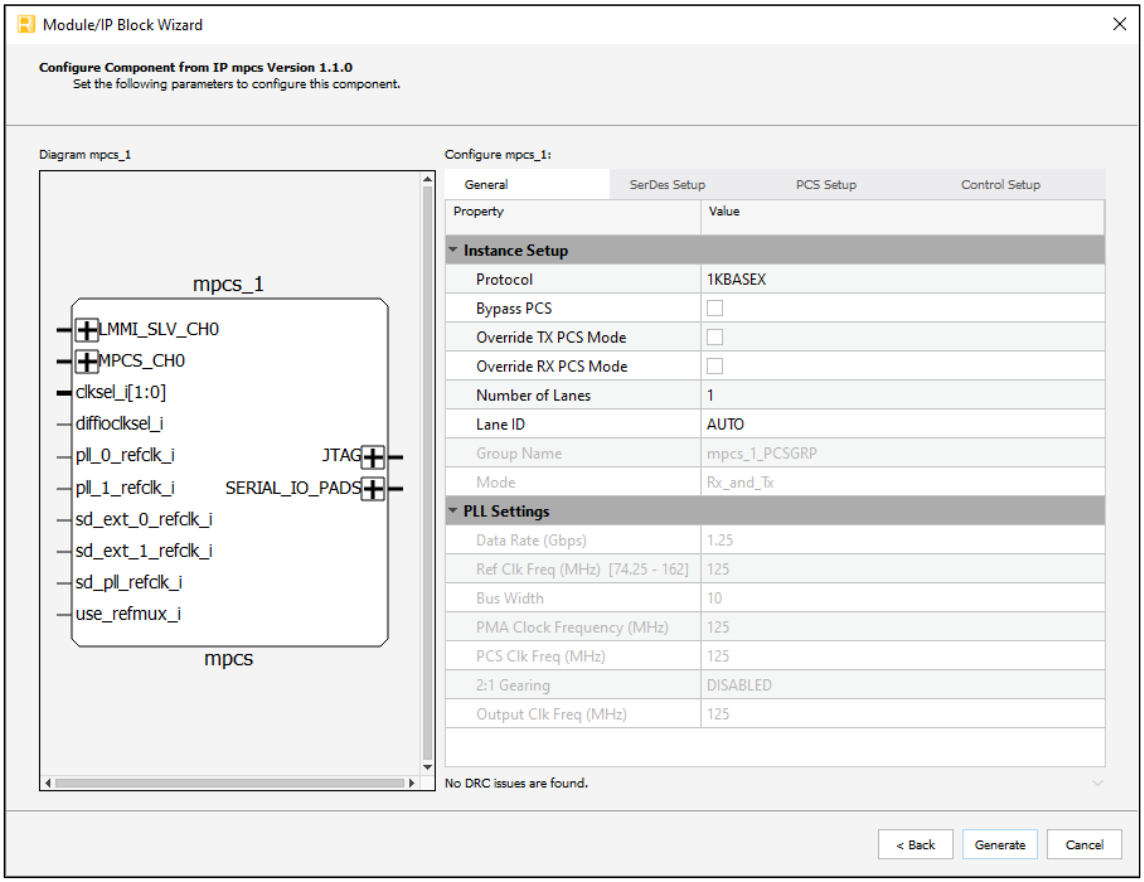

Figure 7.2. Configure User Interface of MPCS Module

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

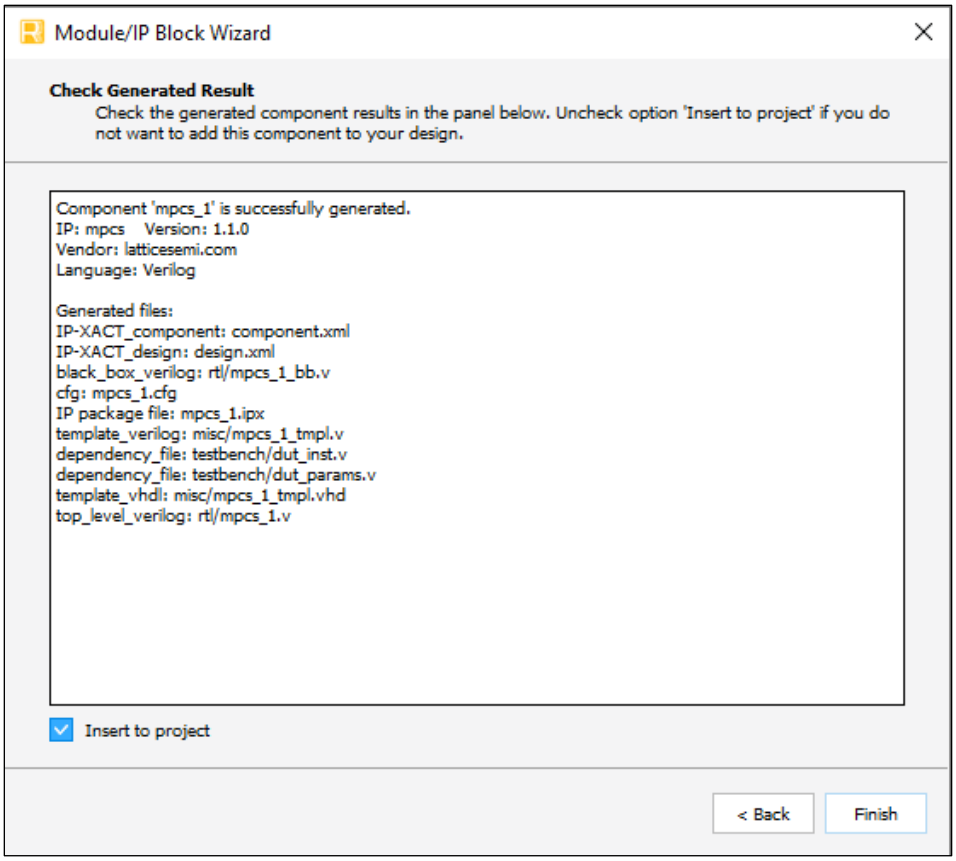

Figure 7.3. Check Generated Result

6. Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 7.1.

The generated MPCS module package includes the closed-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. You 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 7.1.

Table 7.1. 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 closed-box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the module.

7.2. Constraining the IP

You need to provide proper timing and physical design constraints to ensure that your 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 constraint file has been verified with the IP instantiated directly in the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each 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 the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constraint the design and the IP device constraint discussed in the Adding Constraints for MPCS Module section.

7.3. Adding Constraints for MPCS Module

The MPCS module has a known limitation where the Lattice Radiant software cannot provide device constraints for the MPCS generated output clock. You can dynamically set the reference clock sources and parameters to calculate the generated output clock. To address this limitation, refer to Figure 7.4 which shows the clock data path.

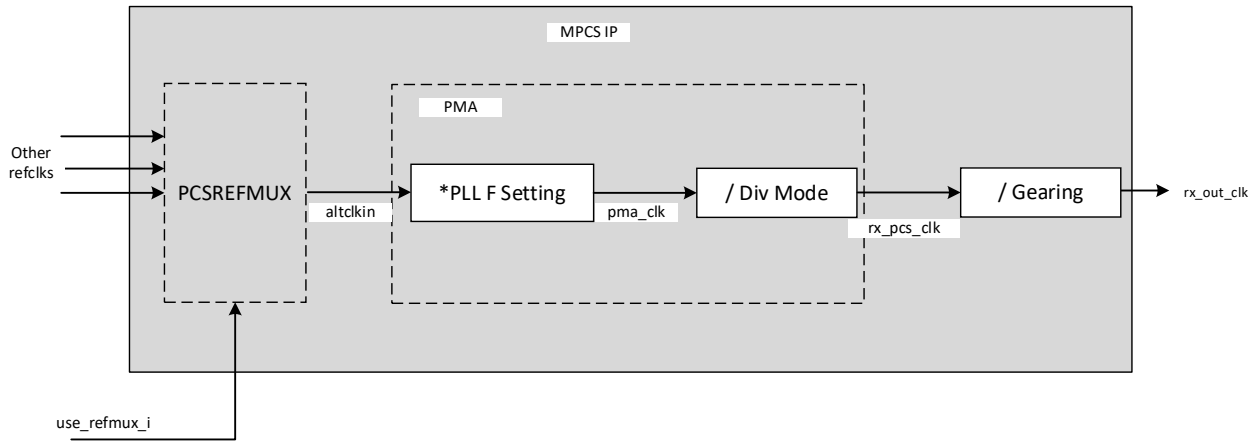

Figure 7.4. Clock Data Path

Sample for SLVS_EC Rx Grade 1:

Fref = 125 MHz

Rx_out_clk = ((125MHz × 1)/1)/1

Rx_out_clk = 125 MHz

Sample for SLVS_EC Rx Grade 2:

Fref = 125 MHz

Rx_out_clk = ((125MHz × 2)/1)/2

Rx_out_clk = 125 MHz

Sample for SLVS_EC RxGrade 3:

Fref = 125 MHz

Rx_out_clk = ((125MHz × 4)/2)/2

Rx_out_clk = 125 MHz

Sample clock constraint for SLVS-EC Rx:

```
create_generated_clock -name {rx_out_clk_c} -source [get_ports {clk_ref_p}] [get_pins {slvs_ec_pcs_top.genblk1.slvs_ec_pcs.lscc_mpcs_top_inst.Q0_PCSX4.u_PCSX4_Q0_pcs_merge_pcs_merge_pcs_merge.PCSX4_inst/CH0_RXOUTCLK}] -divide_by 1
```

8. Debugging

The MPCS module provides loopback support for you to perform debugging activity.

8.1. Loopback Modes

8.1.1. 8b/10b PCS Loopback Modes

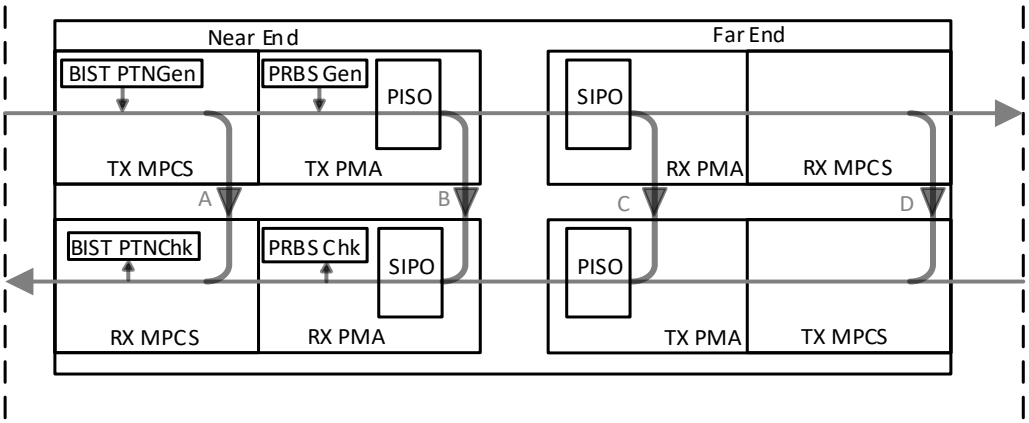

Figure 8.1. 8b/10b PCS Block Diagram

- Loopback A – This loopback can be enabled when *Loopback = 8b/10b PCS Near-End Parallel Loopback Mode*.
- Loopback B – This loopback can be enabled when *Loopback = PMA Near-End Serial Loopback Mode*. The serial data is fed back to the CDR block and the CDR block extracts clock and data generated by PCS TX.
- Loopback C – This loopback can be enabled when *Loopback = PMA Far-End Parallel Loopback Mode*. When enabled, the parallel data received from PMA macro is re-transmitted to PMA macro TX interface. This mode requires that no PPM exists between RX data and TX data, hence alignment of CDR clock to TX clock using the skip bit functionality is performed automatically by FSM.
- Loopback D – This loopback can be enabled when *Loopback = 8b/10b PCS Far-End Parallel Loopback Mode*.

8.1.2. 64b/66b PCS Loopback Modes

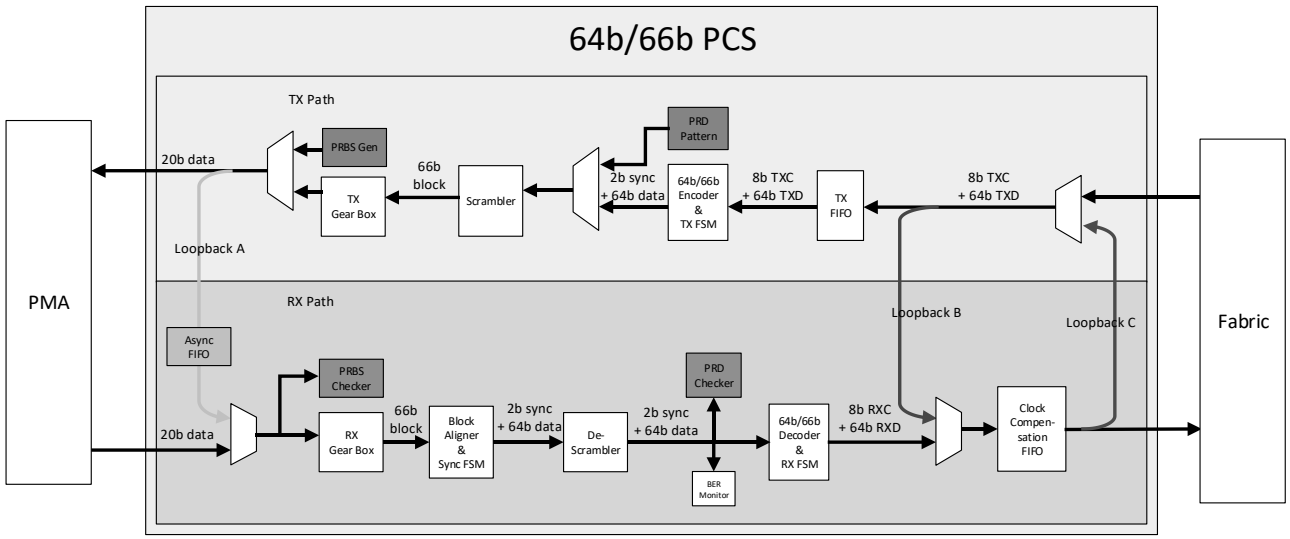

Figure 8.2. 64b/66b PCS Block Diagram

- Loopback A – This loopback can be enabled when *Loopback = 64b/66b Loopback Path A* or by using bit 0 of the Loopback Mode Control register.
  In this mode, the 20-bit input data of RX path comes from TX path. The TX path clock drives both TX and RX path. The asynchronous FIFO between the TX path and RX path is for clock phase compensation.
- Loopback B – This loopback can be enabled using bit 2 of the Loopback Mode Control register.
  In this mode, the PCS accepts data on the transmit path from the XGMII and returns the data on the receive path to the XGMII through the RX FIFO.
- Loopback C – This loopback can be enabled when *Loopback = 64b/66b Loopback Path C* or by using bit 1 of the Loopback Mode Control register.
  In this mode, the received data by RX PCS is returned to TX PCS.

References

- MPCS Module Release Notes (FPGA-RN-02088)
- CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- CertusPro-NX web page
- MachXO5-NX web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Revision 1.8, IP v1.8.0, June 2025

Section	Change Summary
All	Changed SerDes to SERDES.
Introduction	Updated Table 1.1. Quick Facts as follows: • Renamed *Supported FPGA Families* to *Supported Devices*. • Removed the *Targeted Devices* row. • Added *IP Changes*. • Added IP version.
Functional Description	Updated the signal names in the Reset Control for Data Transmission section and Figure 2.8. Example Reset for PCS Data Transmit and Receive.
Signal Description	Updated the description for mpcs_perstn_i in Table 4.1. MPCS Module Signal Description.
References	Updated references.

Revision 1.7, June 2024

Section	Change Summary
All	Renamed document from *MPCS Module - Lattice Radiant Software* to *MPCS Module*.
Inclusive Language	Added inclusive language boilerplate.
Introduction	• Reworked section content and renamed to subsection 1.1 Overview of the IP. • Reworked subsection 3.1 *Licensing the IP* and renamed to subsection 1.4 Licensing and Ordering Information. • Renamed subsection 1.3 *Conventions* and moved to subsection 1.5 Naming Conventions. • Removed subsection 1.3.3 *Attribute*.
Functional Description	• Reworked subsection 2.1 *Overview* and renamed to subsection 2.1 MPCS Module Architecture Overview. • Added subsection 2.2 Clocking. • Moved subsection 2.5.2 *PLL and Divider Settings* to subsection 2.2.1 PLL and Divider Settings. • Reworked subsection 2.5.3 *Clock Generation* and moved to subsection 2.2.2 Clock Generation. • Moved subsection 2.5.4 *Reference Clock Source Selection* to subsection 2.2.3 Reference Clock Source Selection. • Moved subsection 2.5.5 *Two-Quad Clock Connection* to subsection 2.2.4 Two-Quad Clock Connection. • Moved subsection 2.5.8 *64b/66b Clocking* to subsection 2.2.5 64b/66b Clocking. • Added subsection 2.3 Reset. • Reworked subsection 2.5.1 *Data Transmission* and renamed to subsection 2.3.1 Reset Control for Data Transmission. • Added subsection 2.4 MPCS Modes. • Reworked subsection 2.1.1 *8b/10b PCS* and moved to subsection 2.4.1 8b/10b PCS. • Reworked subsection 2.1.2 *64b/66b PCS* and moved to subsection 2.4.2 64b/66b PCS. • Reworked subsection 2.1.3 *PMA Only* and moved to subsection 2.4.3 PMA Only. • Added subsection 2.5 MPCS Component Merging and Lane Mapping. • Reworked subsection 2.1.4 *MPCS Component Merging* and moved to subsection 2.5.1 MPCS Component Merging. • Reworked subsection 2.1.4.1 *Lane Mapping* and renamed to subsection 2.5.2 MPCS Lane Mapping. • Reworked subsection 2.5.6 *Multiple Data Rate Support* and moved to subsection 2.6 Multiple Data Rate Support.

Section	Change Summary
Module Parameter Description	• Reworked subsection 2.3 *Attributes Summary* and renamed to section 3 Module Parameter Description. • Added subsection 3.3 IP Parameter Settings for Example Use Cases.
Signal Description	Reworked subsection 2.2 *Signal Description* and moved to section 4 Signal Description.
Register Description	Reworked subsection 2.4 *Register Description* and moved to section 5 Register Description.
Example Design	• Added this section. • Reworked subsection 3.3 *Running the Functional Simulation* and moved to subsection 6.1 Running the Functional Simulation.
Designing with the MPCS Module	• Renamed section 3 *IP Generation and Evaluation* and moved to section 7 Designing with the MPCS Module. • Reworked subsection 3.2 *Generation and Synthesis* and moved to subsection 7.1 Generation and Synthesis. • Moved subsection 3.4 *Constraining the IP* to subsection 7.2 Constraining the IP. • Reworked Appendix A *Limitation* and renamed to subsection 7.3 Adding Constraints for MPCS Module.
Debugging	• Added this section. • Reworked subsection 2.5.7 *Loopback Modes* and moved to subsection 8.1 Loopback Modes.
References	Updated references.

Revision 1.6, November 2023

Section	Change Summary
Disclaimers	Updated this section.

Functional Description	• Updated tx_data_8b – 4-byte input data information in MPCS Mode 8b10b Tx Data section.
	• Updated the title of section 2.2.1.2 from *MPCS Mode 64b66b* to MPCS Mode 64b66b Tx Data.
	• Updated rx_data_8b – 4-byte output data information in MPCS Mode 8b10b Rx Data section.
	• Updated Table 2.9. Attributes Table:
	  • Deleted attributes of *Rate0 (Gbps)*, *Rate1 (Gbps)*, *Rate2 (Gbps)*, *Select Default Rate*, *Ref Clk Freq (MHz)*, *PMA Clock Frequency (MHz)*, and *PCS Clk Frequency (MHz)*.
	  • Added attributes of *Ref Clk Freq (MHz)*, *VCO Frequency (GHz)*, *PMA Clock Divider*, *PMA Clock Frequency (MHz)*, *PCS Clk Frequency (MHz)*, *Setting1 Enable*, *Setting1: Adaptive Algorithm*, *Setting ½: Training phase Adaptive EQ*, *Setting ½: Post-phase Adaptive EQ*, *Setting 3: Preliminary Adaptive EQ*, and *Setting 3: Training phase Adaptive EQ*.
	  • Updated attributes *Setting2 Enable*, *Setting 3: Adaptive Algorithm*, *Setting ½: Preliminary Adaptive EQ*, and *Setting 3: Post-phase Adaptive EQ*.
	• Updated Table 2.10. Attributes Descriptions:
	  • Added attributes *Ref Clk Freq (MHz)*, *VCO Frequency (GHz)*, *PMA Clock Divider*, *Setting1 Enable: Adaptive Equalization*, *Setting1: Adaptive Algorithm*, *Setting2 Enable: Adaptive Equalization*, and *Setting2: Adaptive Algorithm*.
	  • Deleted attributes *Adaptive Equalization Enable (Data Rate 1)*, *Adaptive Algorithm (Data Rate 1)*, *Adaptive Equalization Enable (Data Rate 0)*, and *Adaptive Algorithm (Data Rate 0)*.
	  • Updated attribute information of *Setting3 Enable: Adaptive Equalization*, *Setting3: Adaptive Algorithm*, and *Output Clk Freq (MHz)Ref Clk Freq (MHz)*.
	• Replaced the text *Control* with *Constant* in the description of registers rege3, rege4, rege5, rege6, rege7, and rege8 in Table 2.12. Summary of MPCS Registers.

IP Generation and Evaluation	Added Constraining the IP section.
References	Added Lattice Insights webpage link.

Revision 1.5, July 2023

Section	Change Summary
Introduction	• Added UT24CP100 to Targeted Devices in Table 1.1. Quick Facts. • Replaced bullet information Supports up to two Quads with LFMXO5 device supports one quad. Other devices can support up to two quads in Features section.
Appendix A. Limitations	• Added the sample computation in Appendix A. Limitations section. • Added Figure A.1. Clock Data Path.
References	Added References section.

Revision 1.4, May 2023

Section	Change Summary
Introduction	Removed PCI Express from the supported standards in the Features section.

Functional Description	• Removed the PIPE Bypass mode from the supported modes of the MPCS module in the Overview section of Functional Description.
	• Removed the PIPE Interface from Figure 2.1. MPCS Module Block Diagram.
	• Removed the PCIE-PCS path and the External PCIE block from Figure 2.2. MPCS Modes Block Diagram.
	• Removed the ports of the PIPE Interface from Table 2.4. MPCS Module Signal Description.
	• Removed the following note of Table 2.4. MPCS Module Signal Description: These ports are available only when Protocol == “PCIE”.
	• Removed PCIE from the Selectable Values of the Protocol in Table 2.9. Attributes Table.
	• Removed PCIE from the Protocol in Table 2.11. Protocol Presetting.
	• In the Overview section of Register Description:
	  • changed *The PCIE and MPCS cores need 8-bit offset only* to *The PMA and MPCS blocks need 8-bit offset only*;
	  • replaced *To separate the PCIE internal register from MPCS internal register, the lmmi_offset_i[8] is used* with *To separate the PMA internal register from MPCS internal register, the lmmi_offset_i[8] is used*;
	  • changed *lmmi_offset_i[8] == 1’b0 – PMA+PCIE Register space* to *lmmi_offset_i[8] == 1’b0 – PMA Register space*;
	  • changed *The PCIE register space is implemented up to 256 registers of 8 bits* to *The PMA register space is implemented up to 256 registers of 8 bits*.
	• Table 2.13. Summary of PMA Registers:
	  • updated its caption to Summary of PMA Registers from Summary of PCIE Registers;
	  • removed the *Offset LMMI*, *Register Name*, *Access Type*, and *Description* values of PCIE registers;
	  • added the *Offset LMMI*, *Register Name*, *Access*, and *Type Description* values of PMA Registers.
	• Updated the descriptions for bist_bypass_tx_gate and bist_bus8bit_sel in MPCS BIST Control 0.
	• Removed pipe_rate_i[1:0] and pipe_pclkout_o from Figure 2.12. Multi-rate Selection Block Diagram.

Revision 1.3, April 2022

Section	Change Summary
Functional Description	• Updated rules in section 2.5 MPCS Component Merging and Lane Mapping • MPCS Component Merging. • Updated Table 4.1 to add mpcs_rxval_o and epcs_rxval_o.

Revision 1.2, November 2021

Section	Change Summary
All	Minor adjustment in formatting.
Introduction	Updated Radiant software version to 3.0 in Table 1.1.
Functional Description	• Updated Table 4.1 to remove lmmi_addr_i. • Updated Table 2.9 to update Data Rate selectable value, added and removed rows in PCS setup group, change EDP protocol to eDP, add new rows in SerDes Setup group, and change selectable value of Loopback Mode from PMA Far-End Serial Loopback Mode to PMA Far-End Parallel Loopback Mode. • Updated Table 2.10 to add new rows for SerDes Setup group. • Updated Table 2.11 to add G8B10B and QSGMII protocol. • Updated Figure 2.10. • Added Two-Quad Clock Connection section.

Revision 1.1, June 2021

Section	Change Summary
All	Minor adjustment in formatting.
Introduction	Updated Table 1.1 to change supported FPGA family to CertusPro-NX.
Functional Description	• Updated Overview, Lane Alignment Mask Code, Register Description, and Operation Details section content, including tables. • Updated Figure 2.2 and Figure 2.3. • Added 8b/10b PCS, 64b/66b PCS, PMA Only, MPCS Component Merging, MPCS Mode 8b10b, MPCS Mode 64b66b, Data Transmission, PLL and Divider Settings, Clock Generation, Reference Clock, Multi, Multi, and Loopback Modes sections. • Updated Table 2.4, Table 2.9, Table 2.10, Table 2.11, and Table 2.12.
IP Generation and Evaluation	Added Running the Functional Simulation section and updated Figure 3.2.
Appendix A. Limitation	Added this section.

Revision 1.0, December 2020

Section	Change Summary
All	Initial release.

www.latticesemi.com