Implementing 9.8G CPRI in Arria V GT and ST FPGAs

2013.12.06

AN 686 Subscribe Send Feedback

This application note describes the implementation of 9.8304 Gbps Common Public Radio Interface (CPRI) using the Arria® V GT and Arria V ST FPGA transceivers.

The hard physical coding sublayer (PCS) block in Arria V FPGAs supports data rates up to 6.5536 Gbps. To implement 9.8304 Gbps CPRI, the transceiver is configured in Physical Media Attachment (PMA) direct mode and a soft PCS block is implemented in the FPGA core.

The following sections describe the configuration of Native PHY IP in PMA direct mode, the architecture of the soft PCS in the FPGA core, and the steps for auto rate negotiation from 9.8304 Gbps down to 1.2288 Gbps.

Note: This application note is accompanied by a reference design to demonstrate the soft PCS implementation and auto rate negotiation from 9.8304 Gbps down to 1.2288 Gbps.

Native PHY IP Settings in PMA Direct Mode

The following figures show the Native PHY IP settings to implement a duplex transceiver channel in PMA direct mode. For this example, the reference clock frequency is set to 491.52 MHz. You can change the reference clock frequency using the drop down menu.

Figure 1: Datapath Options, TX PMA and TX PLL Settings

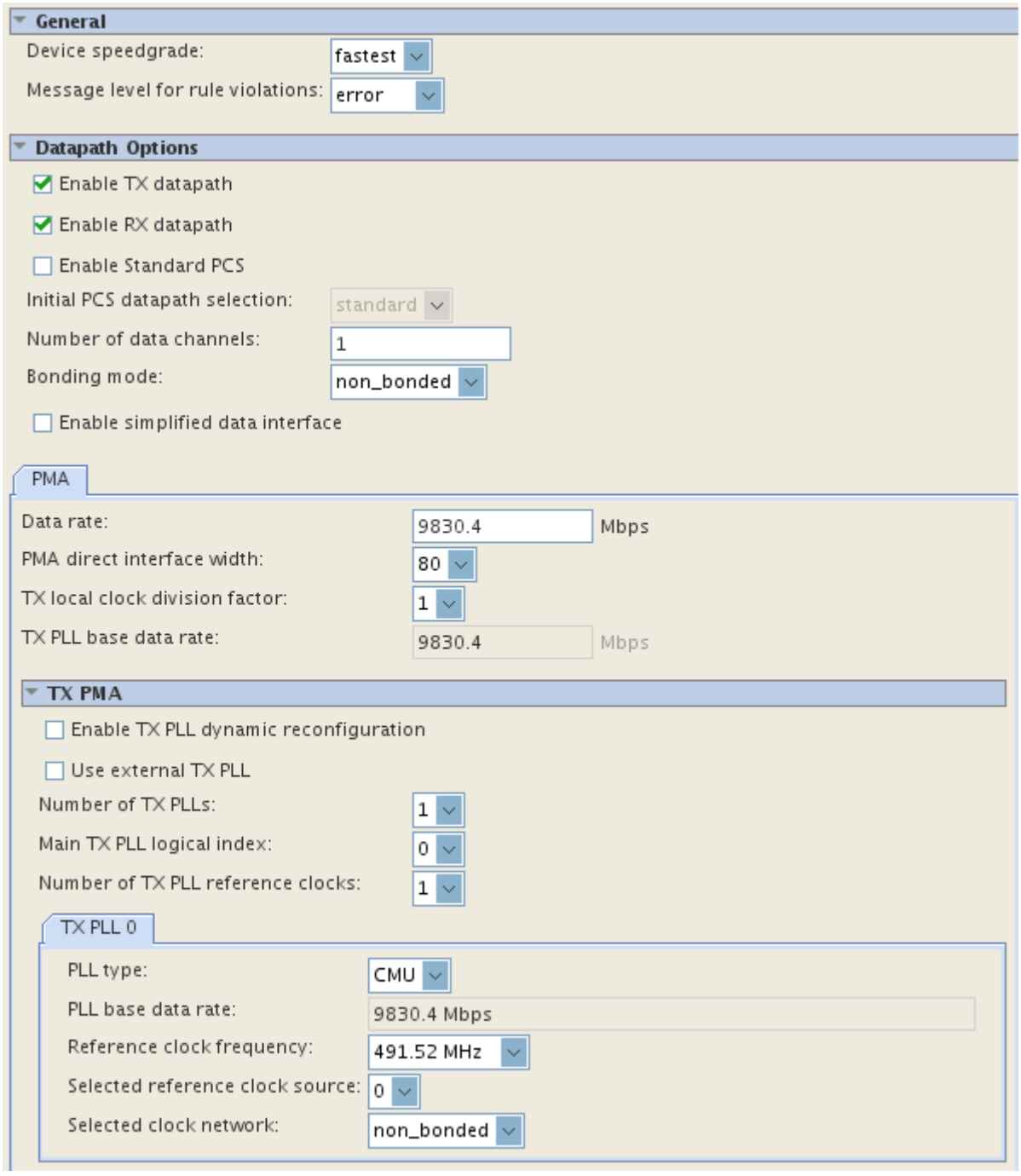
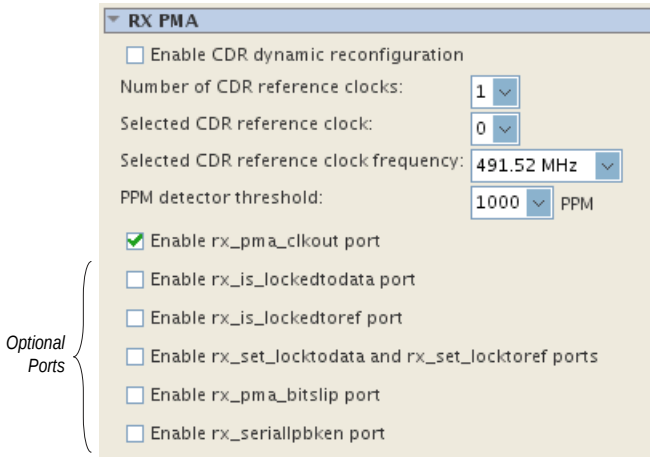


Figure 2: RX PMA Settings



In this example, only one TX PLL is used to drive the channels.

The Native PHY IP does not include an embedded reset controller. You can either design custom reset logic or incorporate Altera's *"Transceiver PHY Reset Controller IP Core"* to implement reset functionality.

Related Information

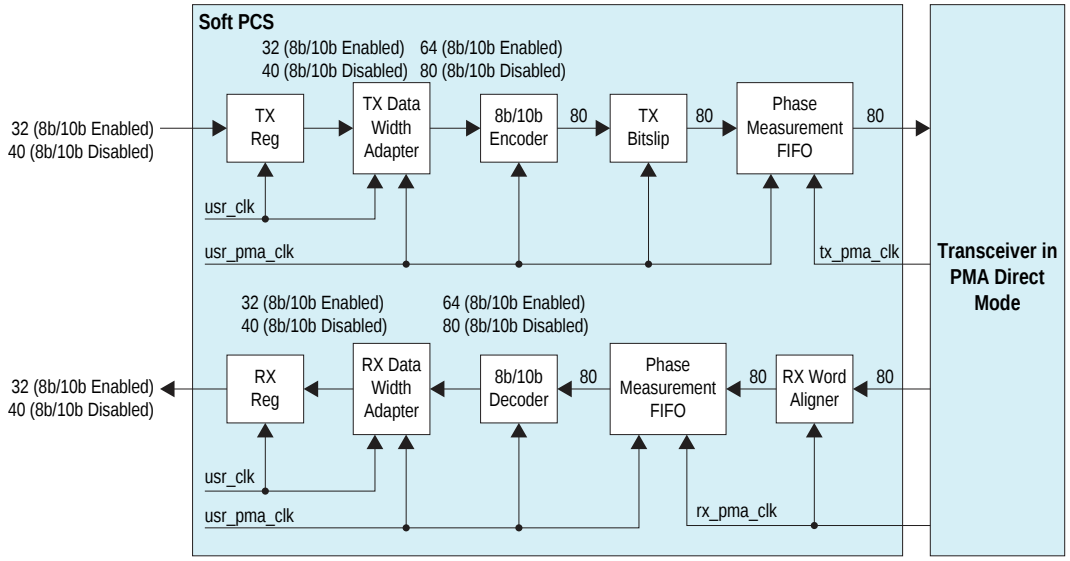
AN 676 - Using the Transceiver Reconfiguration Controller for Dynamic Reconfiguration in Arria V and Cyclone V Devices.
Information about auto rate negotiation applications that require more than one TX PLL.

Altera Transceiver PHY IP Core User Guide
Information about *"Transceiver PHY Reset Controller IP Core"*, and Native PHY IP interfaces and ports.

Soft PCS Architecture

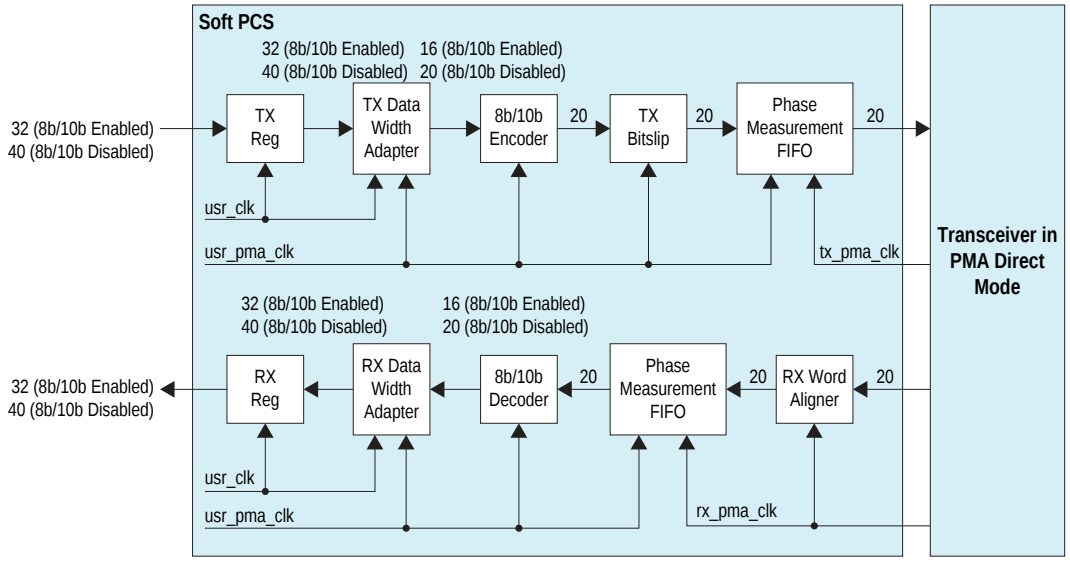
The soft PCS is implemented in the FPGA core and is connected to the transceiver PMA through the 80-bit interface. The input data width to the transmit and receive registers is 32 bits when 8b/10b encoding is enabled and 40 bits when 8b/10b encoding is disabled.

Figure 3: Soft PCS Architecture for Data Rates from 4.9152 Gbps to 9.8304 Gbps



Note: The PMA interface width is 80 bits for data rates of 4.9152 Gbps and above.

Figure 4: Soft PCS Architecture for Data Rates from 1.2288 Gbps to 3.072 Gbps



Note: The PMA interface width is 20 bits for data rates of 3.072 Gbps and below.

Transmit and Receive Registers

The transmit and receive registers synchronize the data from the core or PMA with the soft PCS clocks. The reset signal is synchronized with the `usr_clk` before it is fed to all the blocks. The functionality of the

8b/10b encoder and decoder, the TX Bitslip, and Word Aligner is identical to those in the hard PCS. The word aligner outputs the number of bits slipped on its `rx_boundary_sel` output port.

Related Information

- **Transceiver Architecture in Arria V Devices**
  For information about the 8b/10b encoder/decoder, TX Bitslip and Word Aligner.

Transmit and Receive Width Adapters

The transmit width adapter block is used to convert the input data width in the soft PCS from 32 bits to 64 bits (with 8b/10b encoding) or 80 bits (without 8b/10b encoding) for data rates of 4.9152 Gbps and above. The same block is reused for lower data rates in auto rate negotiation. For data rates of 3.072 Gbps or below, the transmit width adapter converts the data width from 32 bits to 16 bits (with 8b/10b encoding) or 20 bits (without 8b/10b encoding). Because the PMA direct interface mode has a minimum frequency limitation, data width conversion is required.

Similarly, the receive width adapter block reverses the data width from 64 bits (with 8b/10b encoding) or 80 bits (without 8b/10b encoding) to 32 bits for data rates of 4.9152 Gbps and above. For data rates of 3.072 Gbps or below, the receive width adapter converts the data width from 16 bits (with 8b/10b enconding) or 20 bits (without 8b/10b encoding) to 32 bits.

Phase Measuring FIFO

The phase measuring FIFO transfers the data from the write clock domain to the read clock domain. For example, in the TX path it transfers the data from `usr_pma_clk` clock domain to `tx_pma_clk` clock domain and similarly, for the RX path it transfers the data from `rx_pma_clk` clock domain to `usr_pma_clk` clock domain. The FIFO calculates the phase difference between the read and write clock domains as well as the number of data bits stored in the FIFO for latency calculation.

You can measure the FIFO latency to the desired precision using the dedicated `fifo_calc_clk` clock signal. The frequency of `fifo_calc_clk` is related to the `usr_pma_clk` period. N clock periods of the `fifo_calc_clk` are equal to M clock periods of `usr_pma_clk` where N and M are integers. For example, N may be multiple of M, or based on the required accuracy the ratio of M/N may be greater than 1 (such as 64/63 or 128/127). The accuracy for measuring the FIFO latency using the `fifo_calc_clk` signal increases as the M/N ratio approaches 1. For the TX and RX phase measuring FIFOs, set the value of N using the `tx_fifo_sample_size` and `rx_fifo_sample_size` input ports respectively.

The accuracy for measuring the FIFO latency is N/(least common multiple of `usr_pma_clk` periods). The FIFO latency can be read from the `fifo_latency` port.

Note: If your application does not require high precision, drive the `fifo_calc_clk` input port with the `usr_pma_clk` signal. In this case, the M/N ratio is 1 because the frequencies are the same. You can also connect the `fifo_calc_clk` signal to logic 0.

The accumulated phase difference measured across the sample size of N clock periods can be accessed through `phase_measure_acc` port. The `ph_acc_valid` port indicates that the `phase_measure_acc` port is updated with the new data in `fifo_calc_clk` domain.

Soft PCS Clocking

Two system clocks are provided from the core to the soft PCS, unlike the hard PCS where the clock from the transceiver is sent to the core. The upper layer clock `usr_clk` is used to clock the 32 bit data. The `usr_pma_clk` is used to clock the 80 bit or 20 bit data. These two clocks are generated from the user system clock.

Table 1: Clock Frequencies for Data Rates from 9.8304 Gbps to 1.2288 Gbps

Data Rate (Gbps)	Base Data Rate (Gbps)	Local Clock Divider Factor	usr_clk (MHz)	usr_pma_clk (MHz)	Core to Soft PCS Data Width	PMA Width
9.8304	9.8304	1	245.76	122.88	32	80
6.144	6.144	1	153.6	76.8	32	80
4.9152	9.8304	2	122.88	61.44	32	80
3.072	6.144	2	76.8	153.6	32	20
2.4576	9.8304	4	61.44	122.88	32	20
1.2288	9.8304	8	30.72	61.44	32	20

Both the `usr_clk` and `usr_pma_clk` must have 0 parts per million (ppm) phase difference to avoid timing violations in the TX and RX data width adapter blocks. To ensure 0 ppm phase difference between both the clock domains, generate the `usr_clk` and `usr_pma_clk` internally using the fPLL from the reference clock source.

You may choose to generate both clocks from an external PLL; however, ensure that the skew between these two clocks is minimized on the PCB. You can select the pin assignments such that dedicated clock pins are used for both clocks. Use the QSF assignments below if you want to use the dedicated clock pins.

set_instance_assignment -name GLOBAL_SIGNAL "GLOBAL CLOCK" -to usr_clk

set_instance_assignment -name GLOBAL_SIGNAL "GLOBAL CLOCK" -to usr_pma_clk

Soft PCS Latency

Because the designs for the soft PCS and the hard PCS are different, their latencies are different. The extra flip-flops that are added to the soft PCS datapath to meet the tight timing requirements increase the overall soft PCS latency.

Table 2: Soft PCS Latency

The table shows the latencies measured in number of `usr_clk` signal clock cycles for various blocks in the soft PCS datapath.

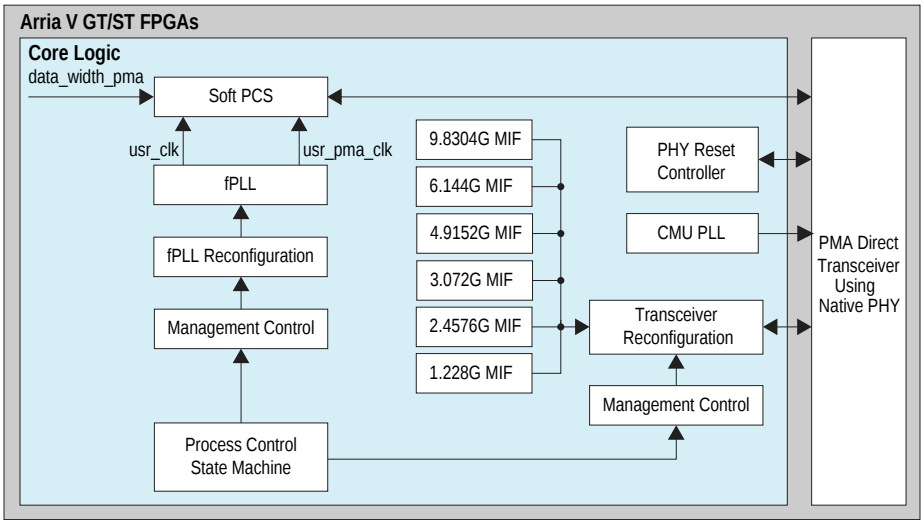
Data width conversion	TX reg	TX data width adapter	8b/10b encoder	TX bit slip	TX Phase measuring FIFO	RX reg	word aligner	RX Phase measuring FIFO	8b10b decoder	RX data width adapter
32 -> 80	1	4 or 5(1)	2	2	(1+`tx_fifo_latency`)*2	1	10	(1+`rx_fifo_latency`)*2	2	4
32 -> 20	1	2	0.5	1	(1+`tx_fifo_latency`)/2	1	2.5	(1+`rx_fifo_latency`)/2	0.5	2.5 or 2 (2)

Auto Rate Negotiation

In Arria V GT and Arria V ST FPGAs, a soft PCS implementation supports auto rate negotiation from 9.8304 Gbps to 1.2288 Gbps. Altera recommends to use the hard PCS for CPRI interfaces where the maximum interface data rate is 6.144 Gbps or lower. In this example, the channels are initialized at the highest supported data rate and are switched to successive lower data rates.

Figure 5: Design Block Diagram for Auto Rate Negotiation from 9.8304 Gbps to Lower Data Rates

The following figure shows an example design of auto rate negotiation for 1.2G /2.5G /3.1G /4.9G/ 6.1G/ 9.8G.



(1) Latency of 5 usr-clk cycles for BC_{hex} (control character K28.5) at lower word and 4 usr_clk cycles for BC_{hex} at upper word.

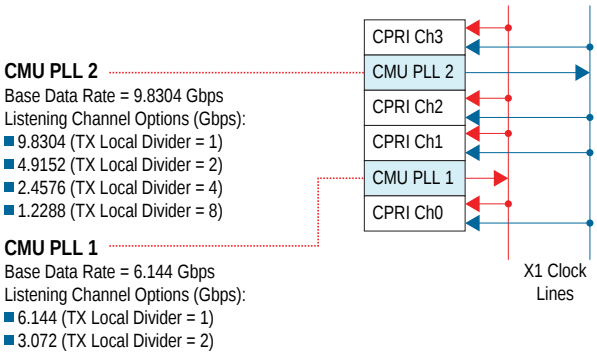
(2) Latency of 2.5 usr-clk cycles for BC_{hex} (control character K28.5) at lower word and 2 usr_clk cycles for BC_{hex} at upper word.

CMU PLL Optimization

To minimize the CMU PLL usage, choose a common appropriate base data rate. To achieve the desired effective data rate, use the local clock divider block. Two CMU PLLs are required to support multiple CPRI channels which require auto rate negotiation independently within the range from 9.8304 Gbps to 1.2288 Gbps.

Figure 6: CMU PLL Optimization and Channel Utilization

The following figure shows the four CPRI channels that are required to change the data rate by performing CMU PLL switching and channel reconfiguration.



Related Information

- **Transceiver Architecture in Arria V Devices**
  For information about the location of transceiver channels which support data rates of 10 Gbps.

Configurations

Use the following configuration settings for soft PCS and fPLL:

Soft PCS Configuration

- Set data rate to 9.8304 Gbps.
- Set the transmit PLL reference clock to 122.88 MHz.
- Set data_width_pma port to 7'd80.

fPLL Configuration

- Set fPLL reference clock to 122.88 MHz
- Set 2 different output clocks.
- Set Output clock 0 as usr_clk at 245.76 MHz and set output clock 1 as usr_pma_clk at 122.88 MHz.

Perform channel reconfiguration to switch between TX PLL 0 and TX PLL1, to change the local clock divider factor, and to change the CDR PLL settings.

Steps for Auto Rate Negotiation

1. Create the original design for the data rate of 9.8304 Gbps. Refer to the following figures for information about data path options, TX and RX PMA settings for the original design.

Figure 7: Datapath Options and PMA Settings

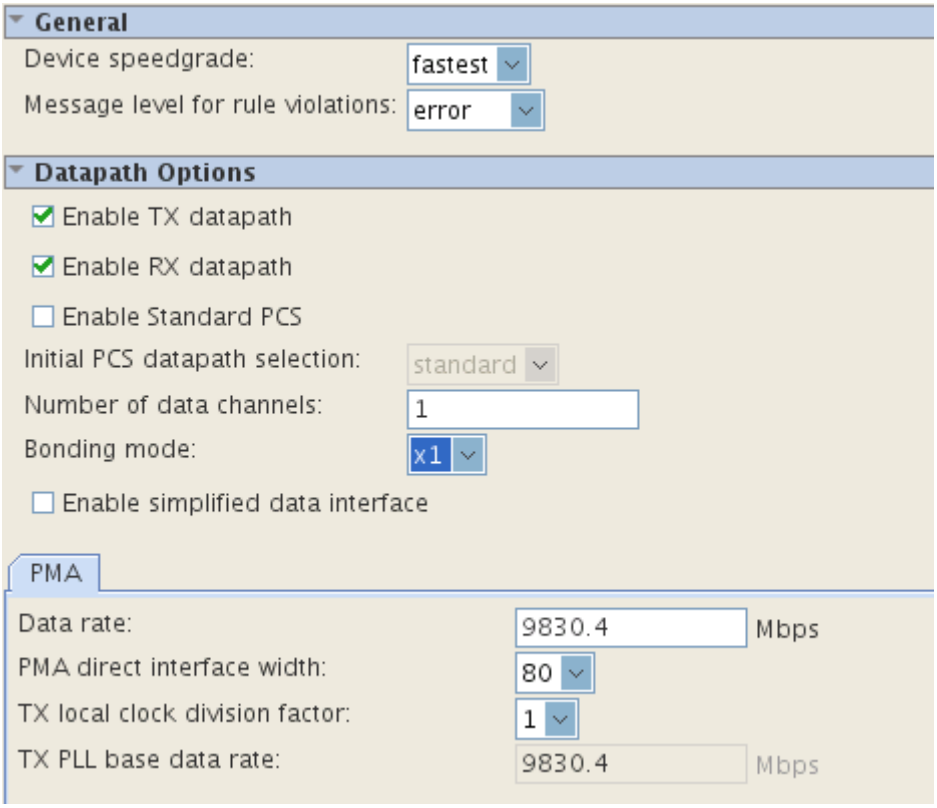


Figure 8: TX PLL 0 Settings

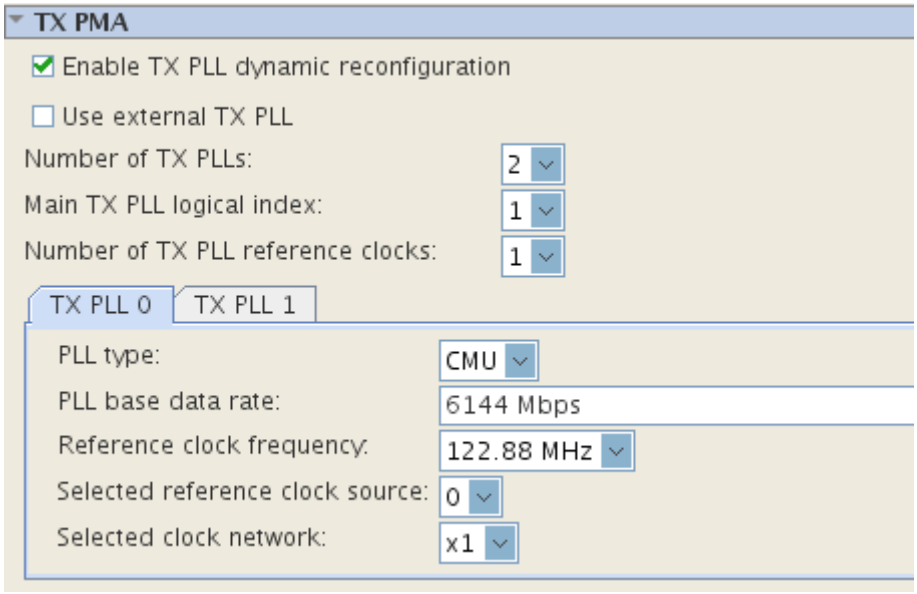


Figure 9: TX PLL 1 Settings

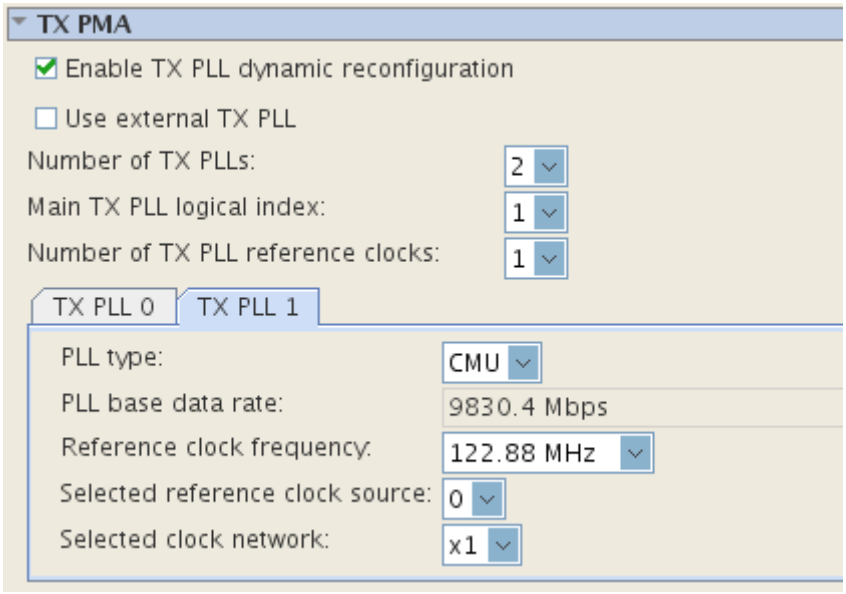
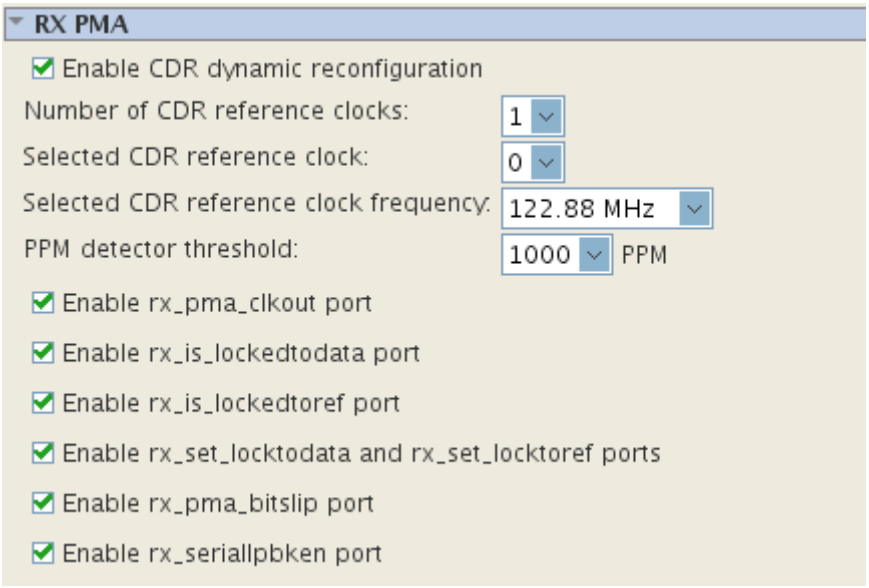


Figure 10: RX PMA Settings



Note: If **Enable rx_pma_bitslip port** option is turned on, the deserializer slips one clock edge each time this signal is asserted. This feature is used to reduce the uncertainty in the serialization process for the data path with deterministic latency.

2. Create 9.8304 Gbps, 6.144 Gbps, 4.9152 Gbps, 3.072 Gbps, 2.4576 Gbps and 1.2288 Gbps Memory Initialization File (MIF) design. The MIF design is the original design with different settings specified for Native PHY IP. In the original design, the initial data rate is 9.8304 Gbps.
3. Refer to the table below for the Native PHY IP settings used to generate the appropriate MIF. The PHY IP settings not listed in the table below are same as the original design. Set PMA data rate to the desired data rate for the static design.

Table 3: Native PHY IP PMA Settings

Data Rate for MIF (Gbps)	PMA			TX PMA			TX PLL1		TX PLL0	
	Data Rate (Mbps)	TX Local Clock Divider Factor	TX PLL Base Data Rate (Mbps)	Use External TX PLL	Number of TX PLLs	Main TX PLL Logical Index	PLL Base Data Rate (Mbps)	Selected Clock Network	PLL Base Data Rate (Mbps)	Selected Clock Network
9.8304	9830.4	1	9830.4	Disabled	2	1	9830.4	x1	6144	x1
6.144	6144	1	6144	Disabled	2	0	9830.4	x1	6144	x1
4.9152	4915.2	2	9830.4	Disabled	2	1	9830.4	x1	6144	x1
3.072	3072	2	6144	Disabled	2	0	9830.4	x1	6144	x1
2.4576	2457.6	4	9830.4	Disabled	2	1	9830.4	x1	6144	x1
1.2288	1228.8	8	9830.4	Disabled	2	1	9830.4	x1	6144	x1

4. Connect the design as shown in **Figure 5**
5. For the original design running at the data rate of 9.8304 Gbps:
   - Set usr_clk to 245.76 MHz and usr_pma_clk to 122.88 MHz.
   - Set data_width_pma to 7'd80.
   - Set TX PLL 1 base data rate to 9.8304 Gbps and TX PLL 0 base data rate to 6.144 Gbps.
6. Follow the appropriate option from the list below to reconfigure to the required data rate.

Note:
- Refer to *AN676 Using the Transceiver Reconfiguration Controller for Dynamic Reconfiguration in Arria V and Cyclone V Devices* for information about the reset controller, CMU PLL, and the reconfiguration controller settings.
- Refer to *AN 661 Implementing Fractional PLL Reconfiguration with ALTERA_PLL and ALTERA_PLL_RECONFIG Megafunctions* for information about implementing fPLL reconfiguration.

- To reconfigure the data rate to 6.144 Gbps:
  1. Reconfigure fPLL
     - Change usr_clk to 153.6 MHz.
     - Change usr_pma_clk to 76.8 MHz.
  2. Use the 6.144 Gbps MIF to perform channel reconfiguration and TX PLL switching.
  3. Set data_width_pma to 7'd80.
- To reconfigure the data rate to 4.9512 Gbps:
  1. Reconfigure fPLL:
     - Change usr_clk to 122.88 MHz.
     - Change usr__pma_clk to 61.44 MHz.
  2. Use the 4.9152 Gbps MIF to perform channel reconfiguration and TX PLL switching.
  3. Set data_width_pma to 7'd80.
- To reconfigure the data rate to 3.072 Gbps:
  1. Reconfigure fPLL:
     - Change usr_clk to 76.8 MHz.
     - Change usr_pma_clk to 153.6 MHz.
  2. Use the 3.072 Gbps MIF to perform channel reconfiguration and TX PLL switching.
  3. Set data_width_pma to 7'd20.
- To reconfigure the data rate to 2.4576 Gbps:
  1. Reconfigure fPLL:
     - Change usr_clk to 61.44 MHz.
     - Change usr_pma_clk to 122.88 MHz.
  2. Use the 2.4576 Gbps MIF to perform channel reconfiguration and TX PLL switching.
  3. Set data_width_pma to 7'd20.
- To reconfigure the data rate to 1.2288 Gbps:
  1. Reconfigure fPLL:
     - Change usr_clk to 30.72 MHz.
     - Change usr_pma_clk to 61.44 MHz.
  2. Use the 1.2288 Gbps MIF to perform channel reconfiguration and TX PLL switching.
  3. Set data_width_pma to 7'd20.

Related Information

AN676-Using the Transceiver Reconfiguration Controller for Dynamic Reconfiguration in Arria V and Cyclone V Devices.

AN661-Implementing Fractional PLL Reconfiguration with ALTERA_PLL and ALTERA_PLL_RECONFIG_Megafunctions.

Soft PCS Parameters and Ports

Most of the parameters and ports for the soft PCS are identical to hard PCS, except for the parameters and ports for the phase measuring FIFO which do not exist in the hard PCS.

Table 4: Soft PCS Parameters

Name	Type	Default Value	Description
operation_mode	String	Duplex	Defines the mode in which the transceiver channel is instantiated.
lanes	Integer	1	Sets the number of transceiver channels in the design. In the attached reference design, it is set to one.
ser_base_factor	Integer	8	Enables or disables the 8B/10B encoder and decoder within the soft PCS. To bypass the 8B/10B logic within the soft PCS, set this parameter's value to 10.
ser_words	Integer	4	Determines the data width from the upper layer. Data_width = (ser_base_factor * ser_words). Set this value to 4.
pcs_pma_width	Integer	80	Determines the PCS to PMA interface data width. For 9.8 Gbps data rate, data width must be 80 bits.
ser_word_pma_size	Integer	4	The value is ceil[log2(number of 8-bit data on PCS to PMA interface)]. In this case, the PCS to PMA interface data width is 80 and the number of 8-bit data is 10. Ceil[log2(10)] is 4.
data_width_pma_size	Integer	7	The value is ceil[log2(PCS to PMA interface data width)]. The PCS to PMA interface data width is 80 bits. Ceil[log2(80)] is 7.
base_data_rate	String	0 Mbps	Determines the clock generation block (CGB) factor. Can be ignored for 9.8 Gbps data rate.
tx_pma_clk_div	Integer	1	Determines the clock generation block (CGB) division factor. Can be ignored for 9.8 Gbps data rate.
pll_feedback_path	String	No compensation	Disables the PLL feedback path. Because the phase measuring FIFO is in the soft PCS, disable the PLL feedback path.
word_aligner_mode	String	Deterministic Latency	Determines the word aligner's mode of operation.

Name	Type	Default Value	Description
pll_refclk_cnt	Integer	1	Specifies the number of reference clocks for the PLL. Match the number of reference clocks for the PLL with the Native PHY IP instance.
plls	Integer	2	Specifies the number of TX PLLs in the design. For auto rate negotiation from 9.8304 Gbps to 1.2288 Gbps set the
cdr_refclk_cnt	Integer	1	Specifies the number of reference clocks for the CDR. Match the number of reference clocks for the CDR with the Native PHY IP instance.
tx_fifo_depth	Integer	4	Determines the depth of the transmit phase measuring FIFO. The value is log2(FIFO buffer depth). In this case, the FIFO buffer depth is 16 and the value of log2(16) is 4.
rx_fifo_depth	Integer	4	Determines the depth of the receive phase measuring FIFO. The value is log2(FIFO buffer depth). In this case, the FIFO buffer depth is 16 and the value of log2(16) is 4.
ref_design	Integer	1	Set this value to 1 if you choose to use the IP as a reference design.

Table 5: Soft PCS Ports

Signal Name	Direction	Description
usr_clk	Input	Clocks the 32-bit data from the upper layer to the soft PCS.
usr_pma_clk	Input	Clocks the soft PCS data before transferring to the `tx_pma_clk` or `rx_pma_clk` domain through the phase measuring FIFO.
fifo_calc_clk	Input	Clocks the phase measuring FIFO to calculate the phase difference between the write clock and read clock of the phase measuring FIFO.
cdr_ref_clk	Input	Input reference clock for the CDR.
tx_fifo_sample_size	Input	Value to determine `fifo_calc_clk` frequency. User-defined value N where M/N=`fifo_calc_clk` period/`usr_pma_clk` period. Synchronize internally to `fifo_calc_clk`. Set the value of N using the `tx_fifo_sample_size` input port.

Signal Name	Direction	Description
rx_fifo_sample_size	Input	Value to determine `fifo_calc_clk` frequency. Sample size for calculating the phase difference in the RX phase measuring FIFO. User-defined value N where M/N=`fifo_calc_clk` period/`usr_pma_clk` period. Synchronize internally to `fifo_calc_clk`. Set the value of N using the `rx_fifo_sample_size` input port.
tx_phase_measure_acc	Output	Measures the accumulated delay through the transmit buffer in `fifo_calc_clk` clock domain.
rx_phase_measure_acc	Output	Measures the accumulated delay through the receive buffer in `fifo_calc_clk` clock domain.
tx_fifo_latency	Output	Latency of TX phase measuring FIFO buffer in `usr_pma_clk` domain.
rx_fifo_latency	Output	Latency of RX phase measuring FIFO buffer in `usr_pma_clk` domain.
tx_ph_acc_valid	Output	Indicates that the `phase_measure_acc` port for RX phase measuring FIFO contains updated data in `fifo_calc_clk` domain.
rx_ph_acc_valid	Output	Indicates that the `phase_measure_acc` port for the RX phase measuring FIFO contains updated data in `fifo_calc_clk` clock domain.
tx_wr_full	Output	Indicates that the TX phase measuring FIFO in `usr_pma_clk` clock domain is full.
rx_wr_full	Output	Indicates that the RX phase measuring FIFO in `rx_pma_clk` clock domain is full.
tx_rd_empty	Output	Indicates that the TX phase measuring FIFO in `tx_pma_clk` clock domain is empty.
rx_rd_empty	Output	Indicates that the RX phase measuring FIFO in `usr_pma_clk` clock domain is empty.
data_width_pma	Input	Specifies the effective PMA width used during that period.
error	Output	Indicates that the value for `data_width_pma` is invalid.

Document Revision History

Table 6: Document Revision History

Date	Version	Changes
December 2013	2013.12.06	Initial release.