XCVR Design Training

Peter Schepers High Speed Specialist

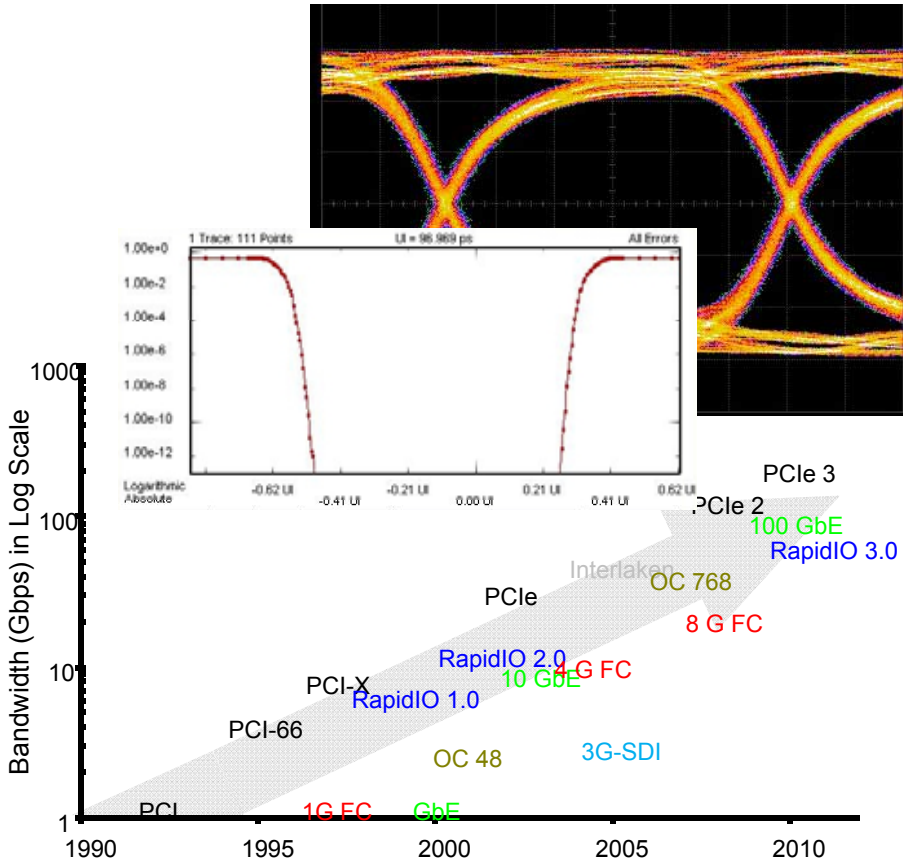
November 2012

Agenda

- Transceiver Architecture
- Clock Recovery and Jitter Tracking
- Transmitter PLL
- Transceiver Design
- Transceiver Reconfiguration
- Demo Design Using Reconfiguration
- Signal Conditioning and Best Practices for Link Training
- Signal Integrity Simulation (with Demo)
- Transceiver Toolkit (with Demo)

Transceiver Design Trends & Challenges

- Data rates
- System integration
  - Optical modules
  - Backplanes
- Statistical reliability
  - Low jitter
  - Bit Error Ratio (**BER**)
- Design productivity
- Lower power consumption



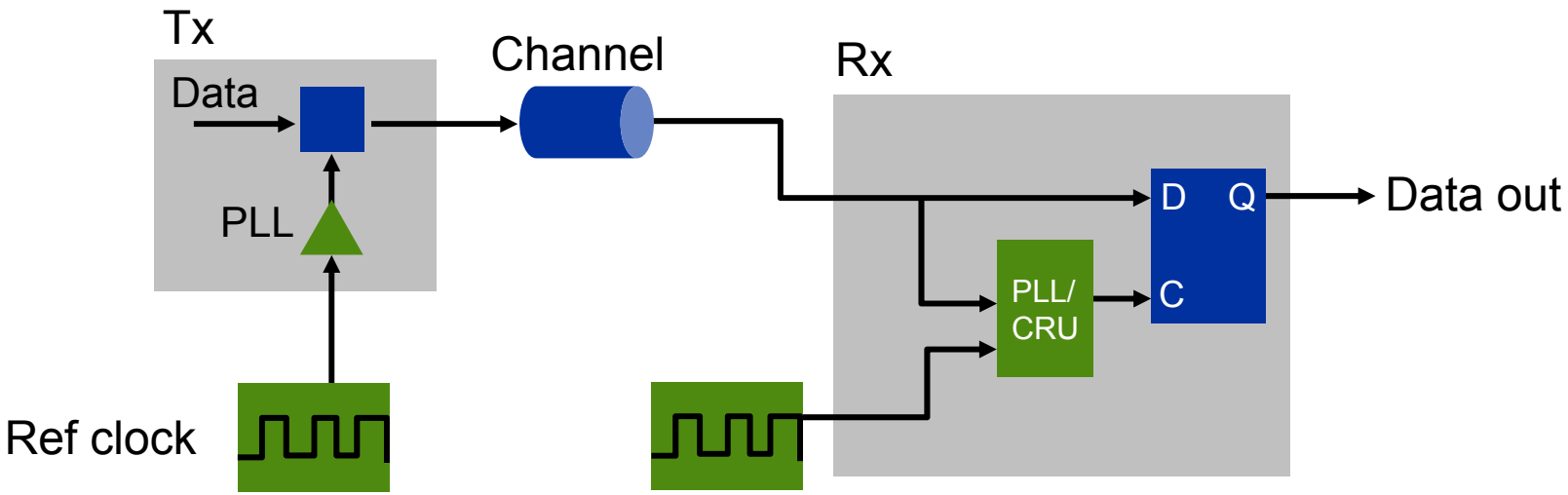
Altera's innovation address the challenges of next generation transceiver design

Transceiver Architecture

What is a Transceiver?

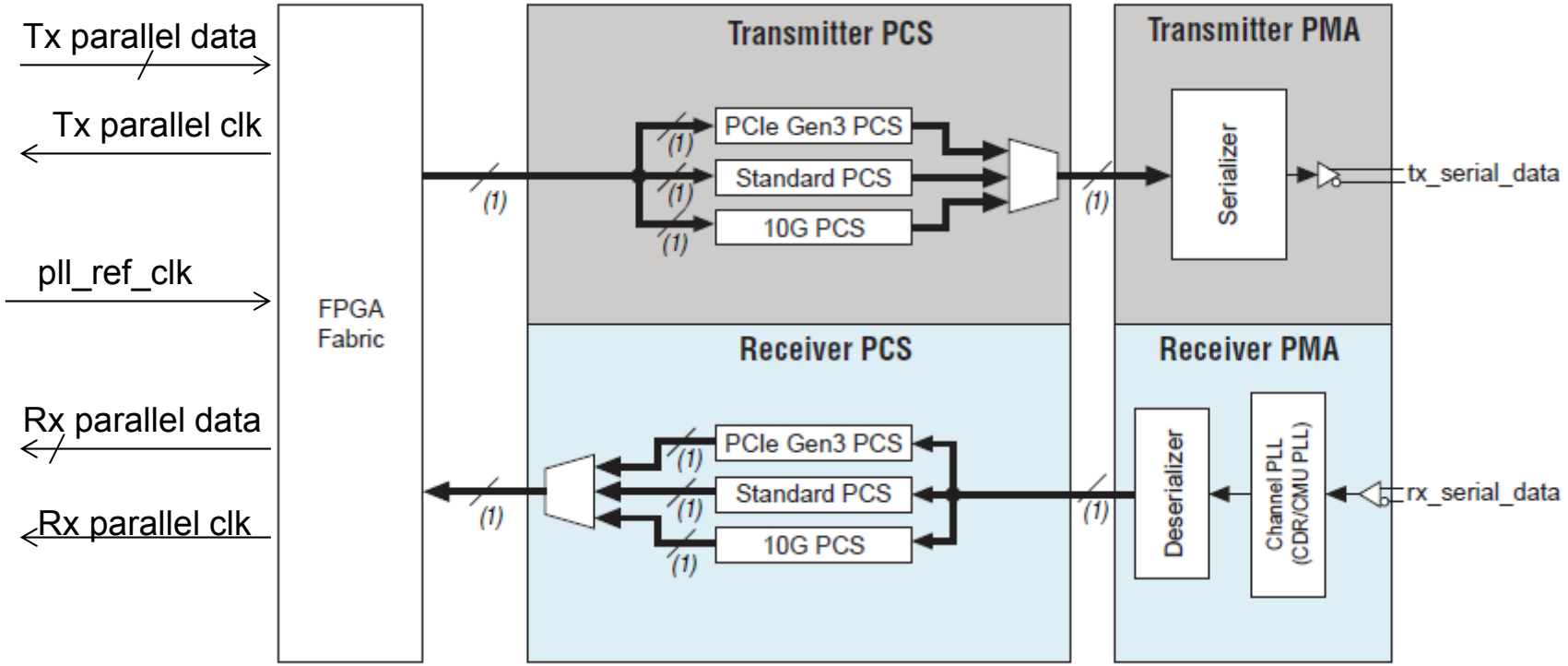
- Combination transmitter/receiver used when sending high-speed digital data/control signals across physical medium
  - Board traces
  - Backplane
  - Optical fiber
  - CAT5 cable
- Used in the PHY (physical) layer of the OSI model
- Made up of the physical coding sub-layer and physical medium attachment

Serial Communication



Ref clock

What are transceivers



- transmit and receive parallel data on a serial link

Definitions

- Media Access Controller (MAC)
  - Assembles packets to be transmitted across link
  - Disassembles packets received from across link
  - Handles error and fault messages from link
- Physical Coding Sub-Layer (PCS)
  - Digital logic that prepares and formats data for transmission across a physical medium type or restores received data to original form
  - Detects link errors
  - Ex. Encoding, decoding, scrambling, descrambling
- Physical Medium Attachment (PMA)
  - Converts digital data to serial analog stream or reverse
  - Connects to physical medium
  - Ex. Parallel to serial conversion

28-nm Device Families

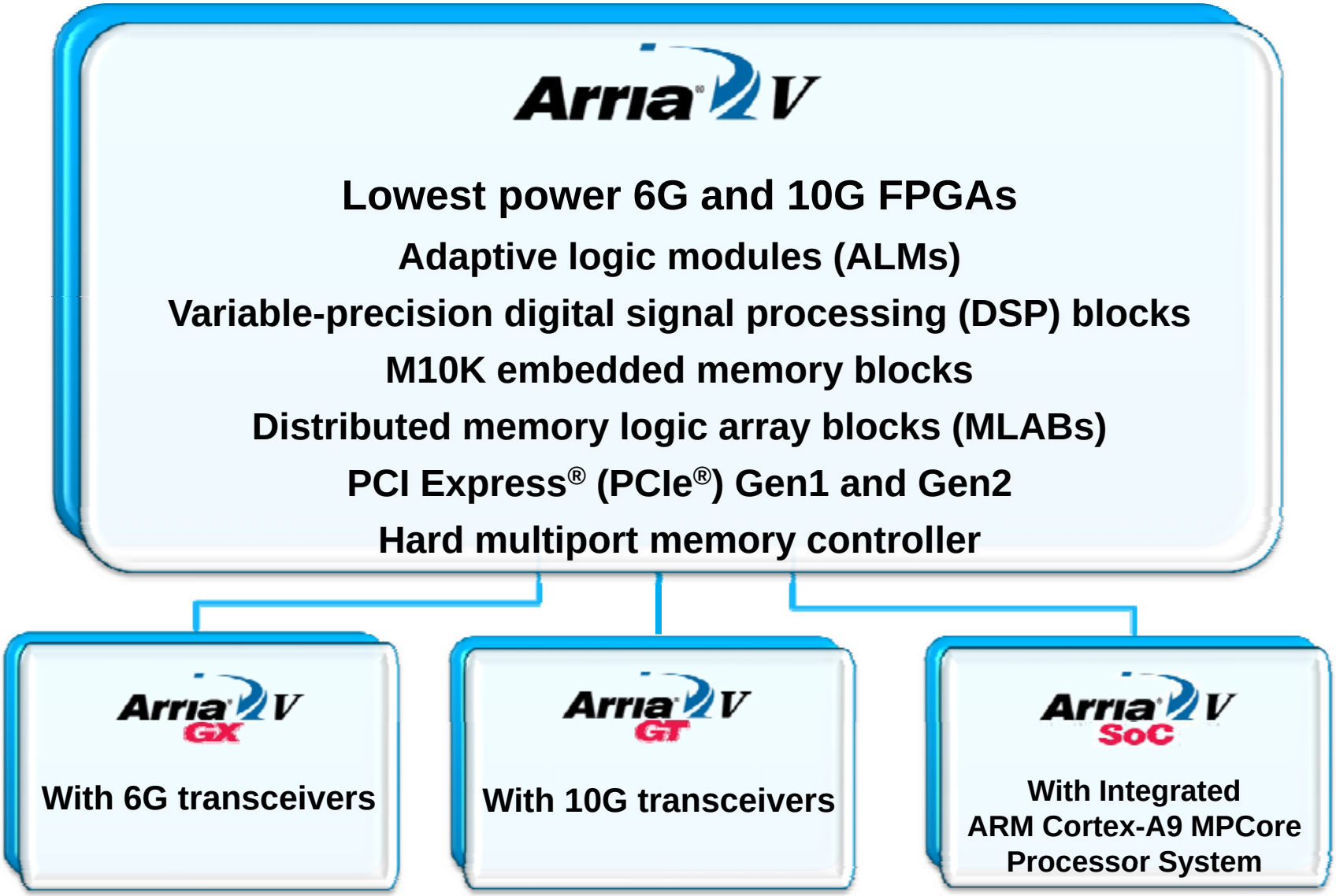
- Cyclone V
- Arria V
- Stratix V

Cyclone V FPGA Family

Opening Up Design Possibilities

	Lowest cost and power	3G transceivers	5G transceivers
	*Optimized for lowest system cost and power for a wide spectrum of general logic and DSP applications*	*Optimized for lowest cost and power for 614 Mbps to 3.125 Gbps transceiver applications*	*FPGA industry's lowest cost and power for 5.0 Gbps transceiver applications*
FPGA	**E Variant**	**GX Variant**	**GT Variant**
Integrated ARM Cortex-A9 MPCore Processor System	**SE Variant**	**SX Variant**	**ST Variant**

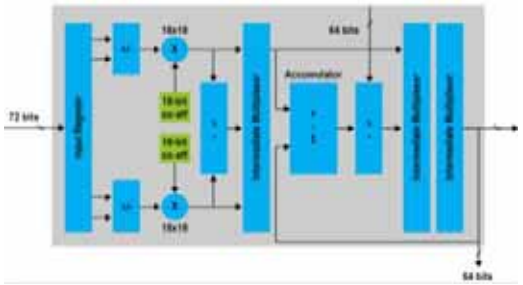
Arria V FPGAs



Stratix V Device Family Variants

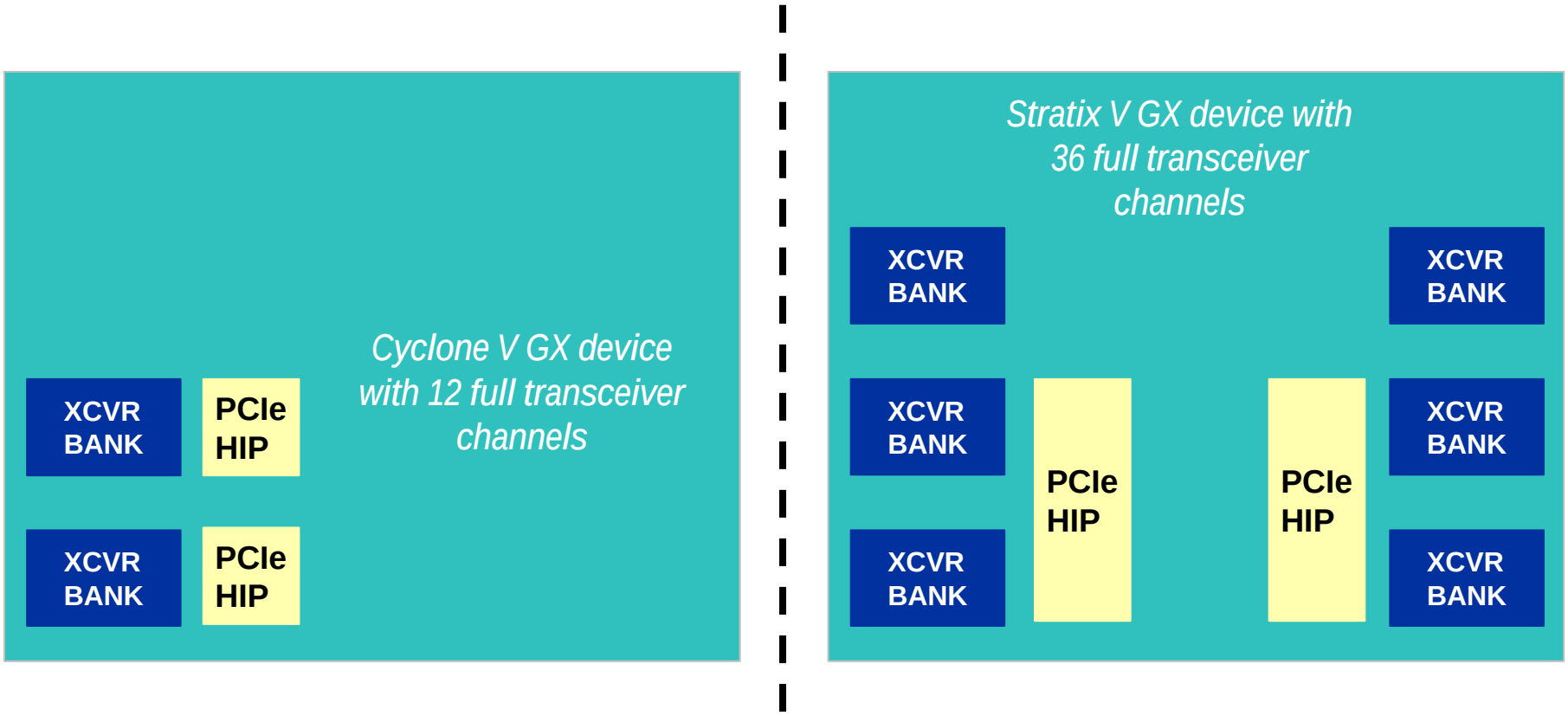
- Stratix V E variant
  - For highest density, high-performance applications
- Stratix V GS variant
  - Optimized for high-performance, high-precision DSP applications with transceivers up to 14.1 Gbps
- Stratix V GX variant
  - Up to 66 transceivers at 14.1 Gbps for high performance, high bandwidth
- Stratix V GT variant
  - 28 Gbps for high-performance, ultra-high bandwidth applications

28-Gbps Transceivers

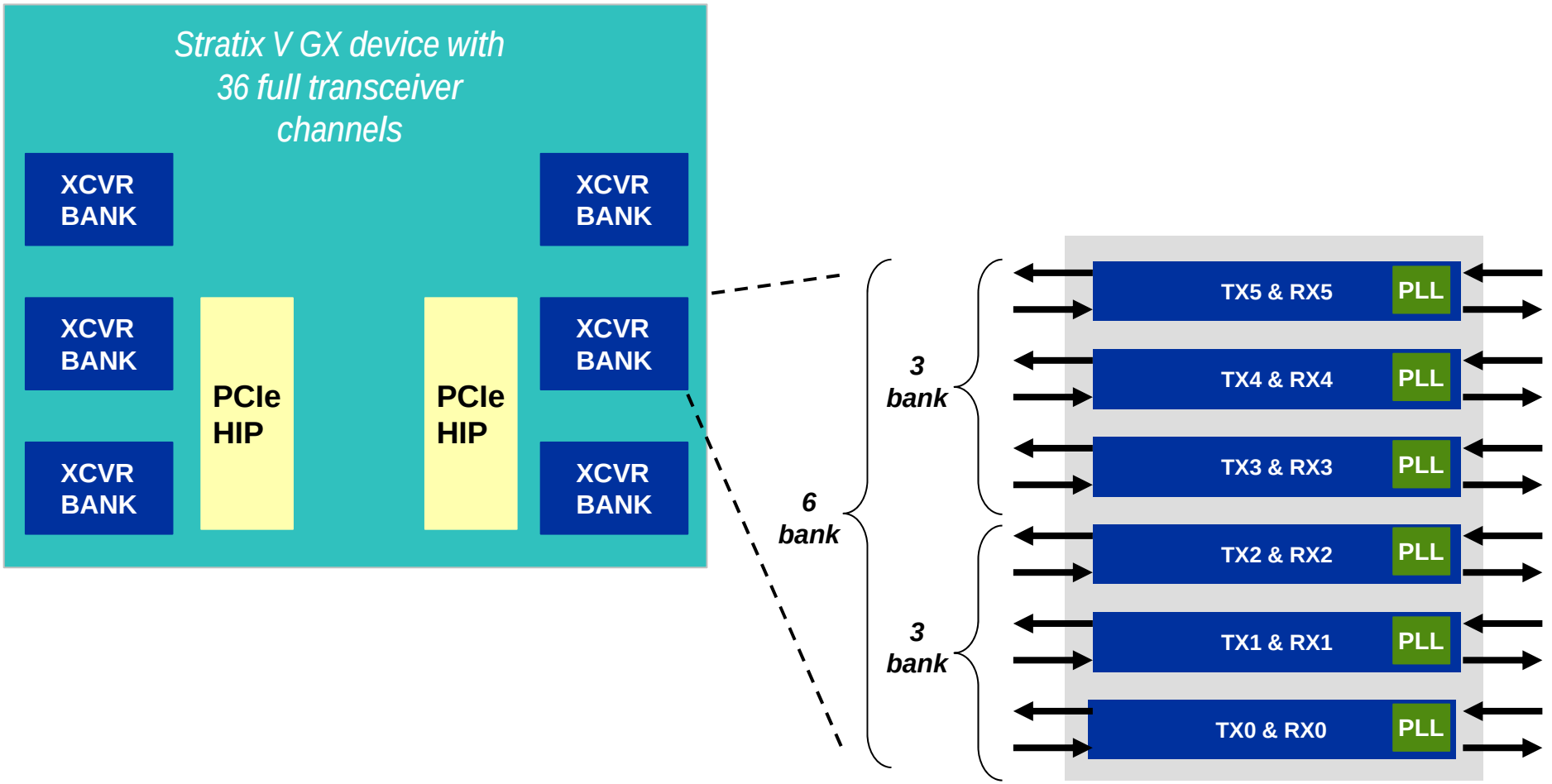


Variable-Precision DSP Block

Transceiver Locations

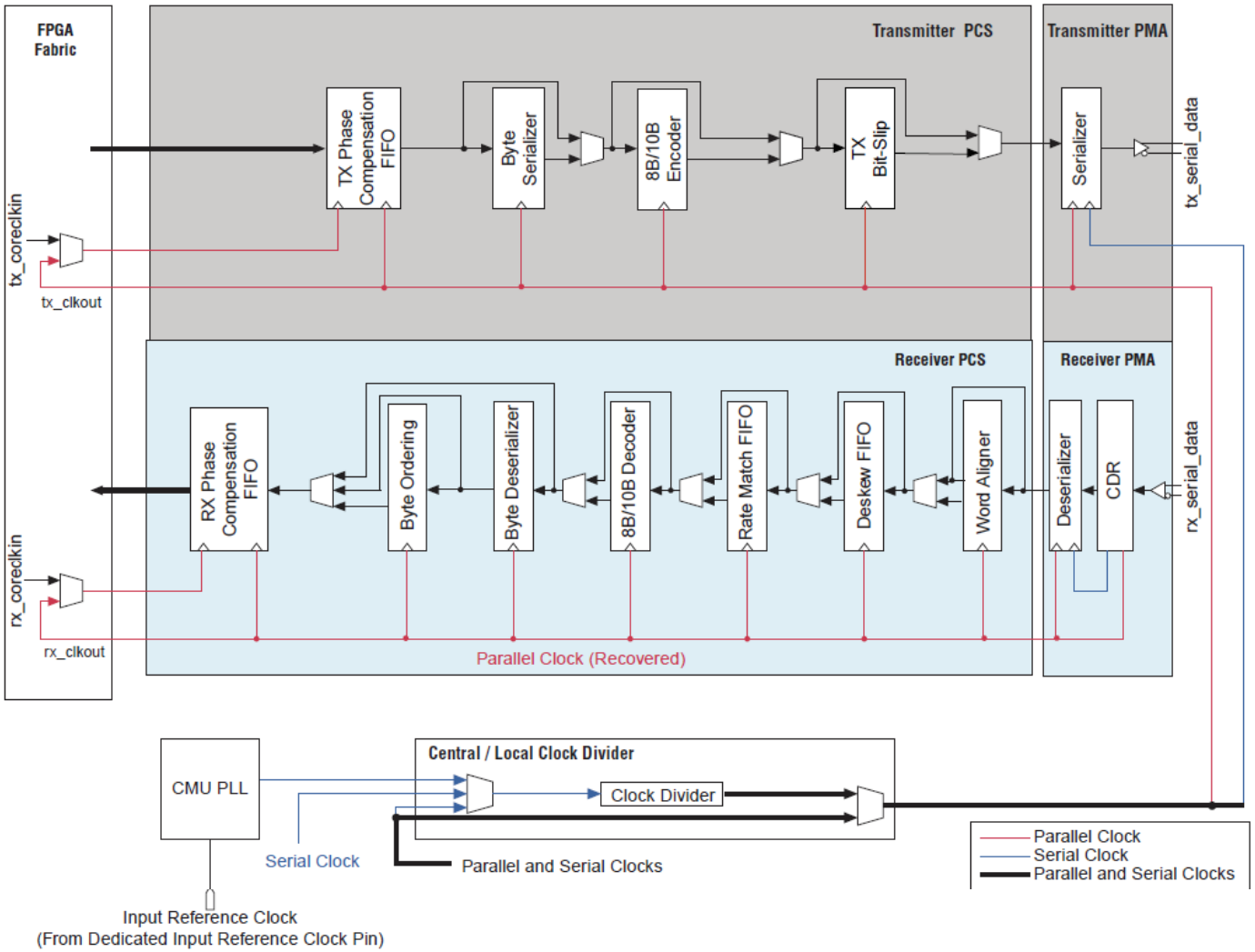


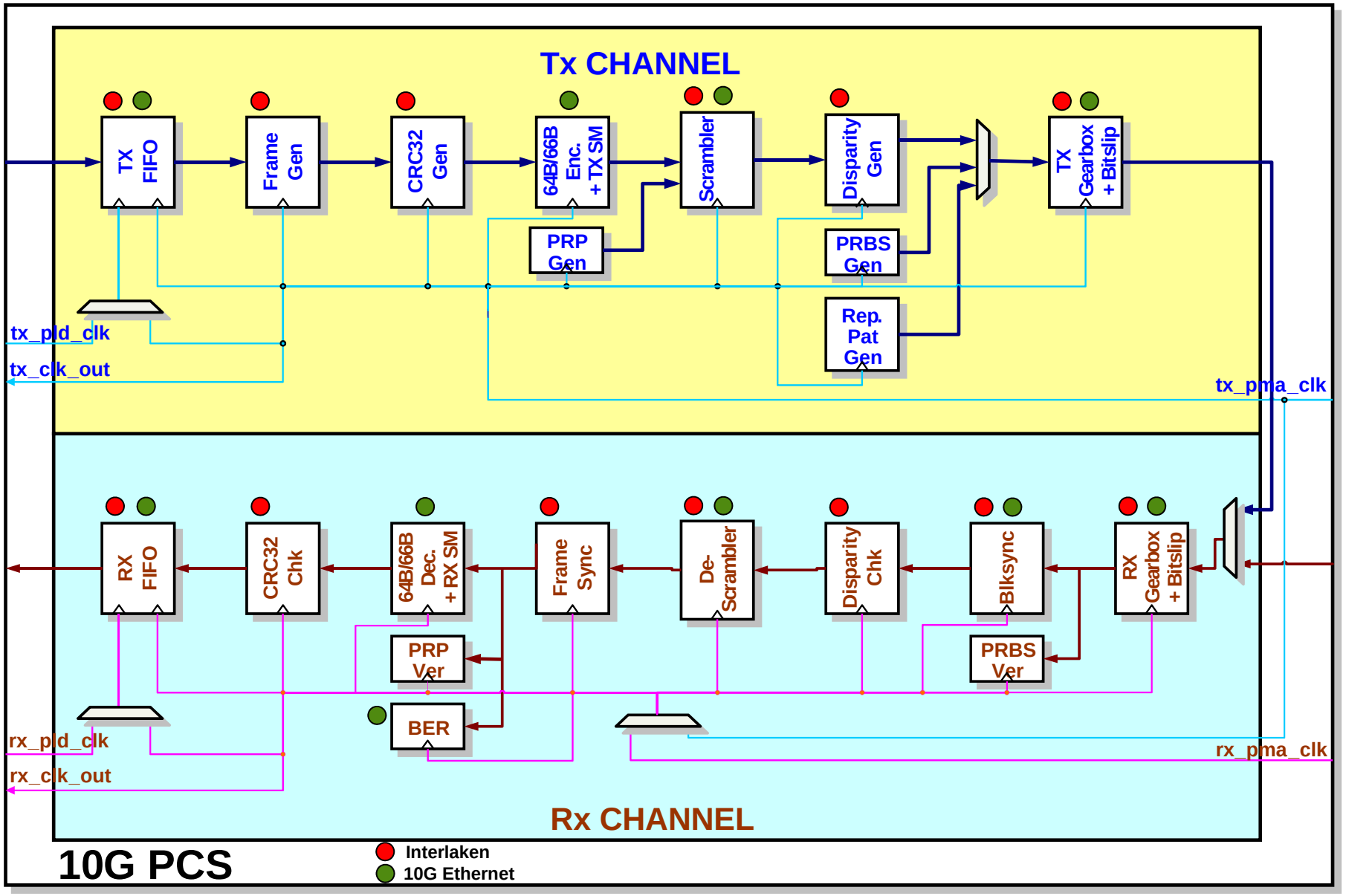
Transceiver Layout



Standard PCS Datapath

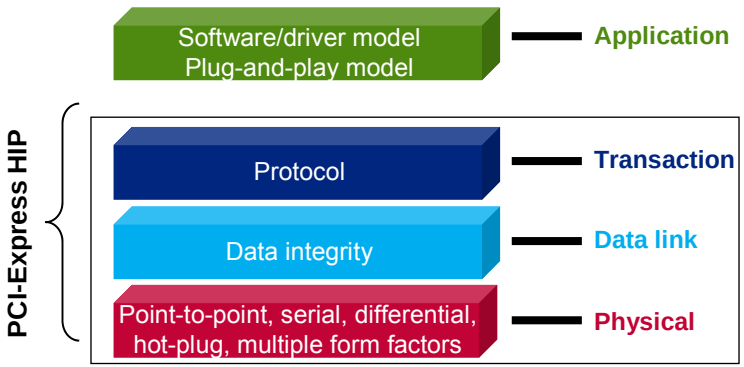
Figure 1–18. Standard PCS Datapath in Stratix V GX Channels

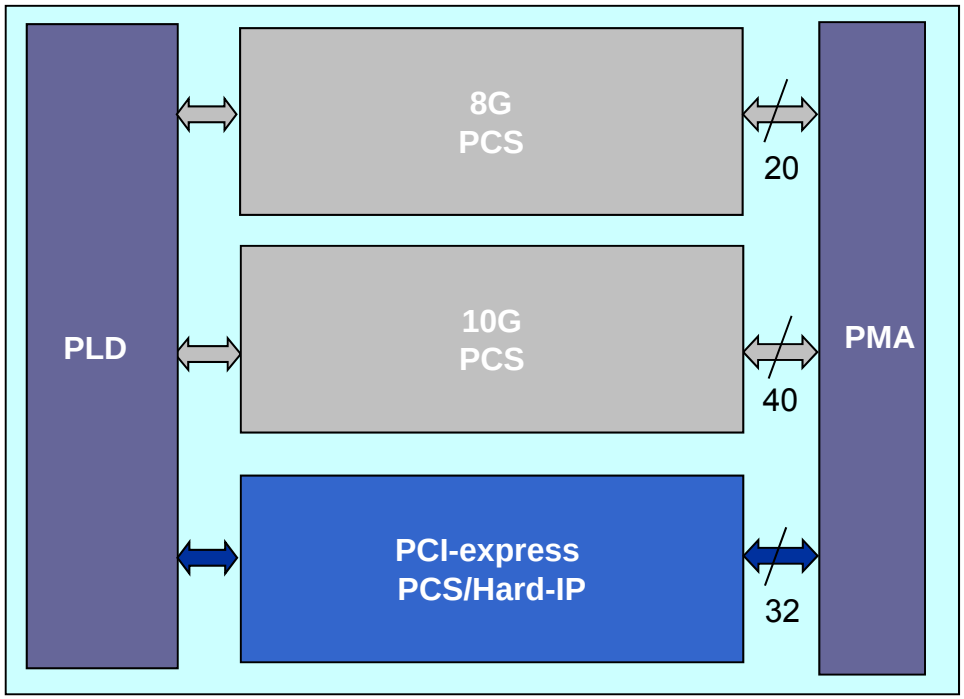




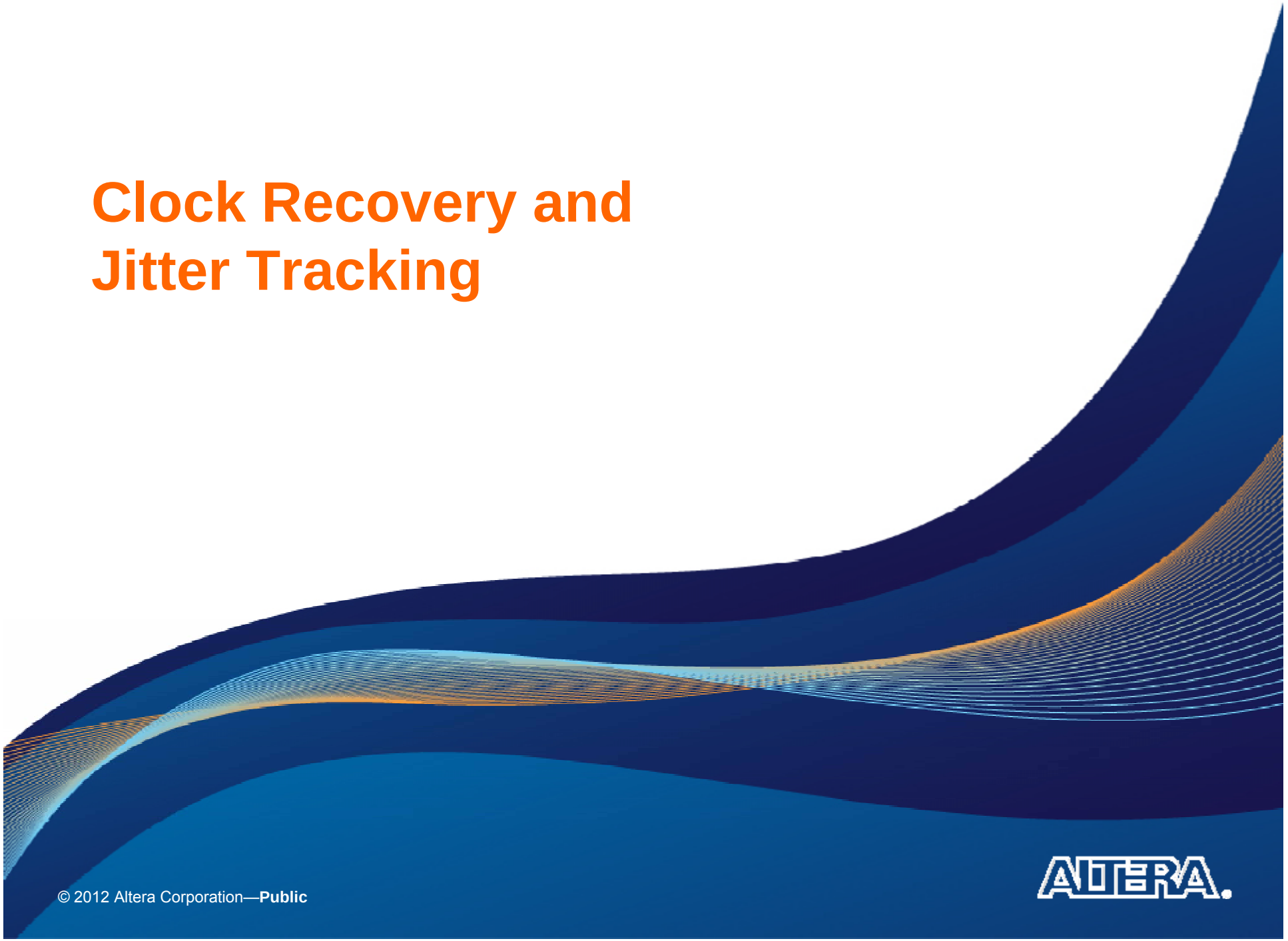
PCI-Express Hard-IP

- Stratix V PCIe Gen 3/2/1 x8 **HIP** with 256 bit architecture
- New Stratix V PCIe Gen3 32-bit **PCS** used with HIP
  - Implements the 128/130 bit encoding/decoding scheme
- Configuration via PCI-Express (CvPCIe) enabled in this block

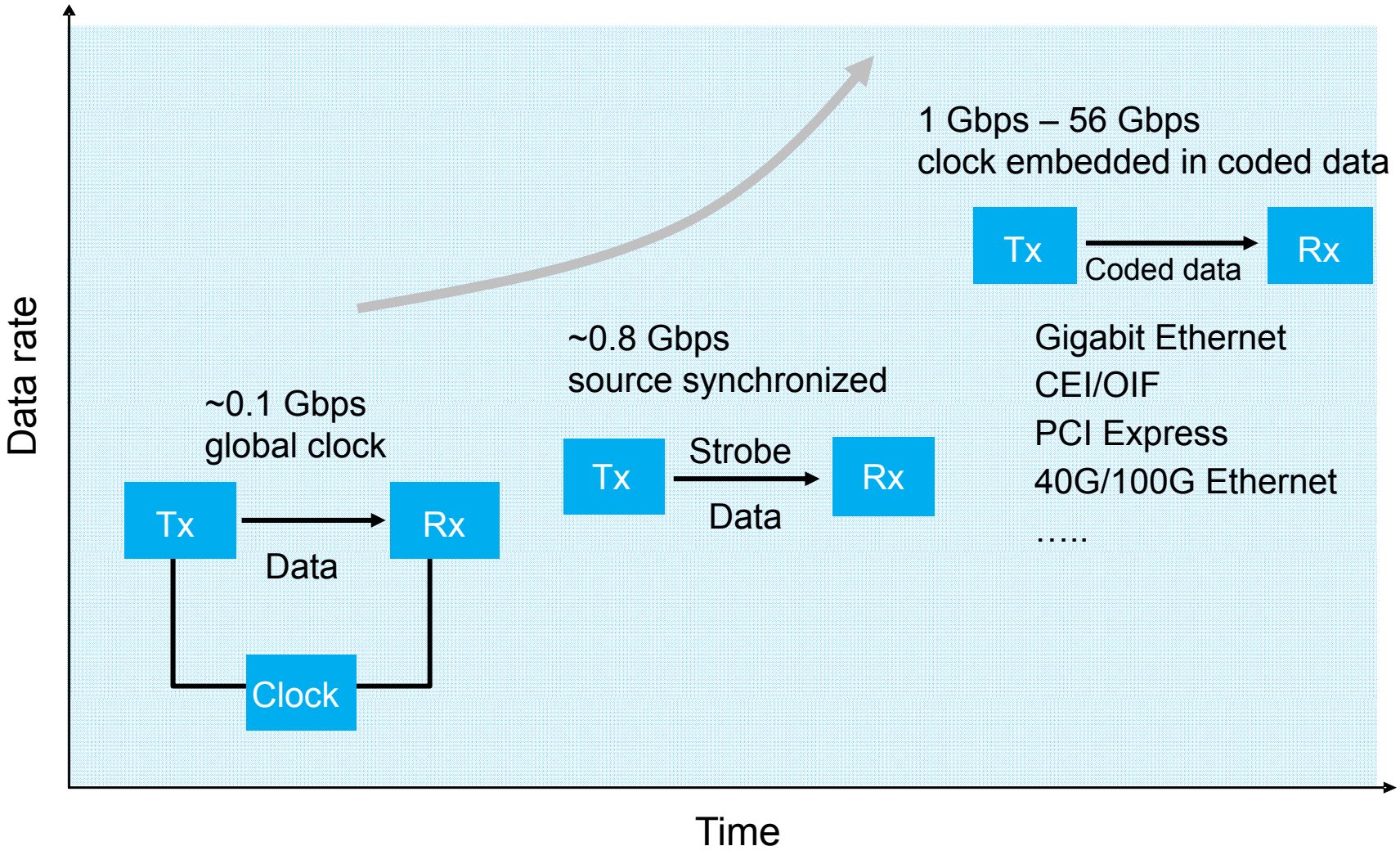




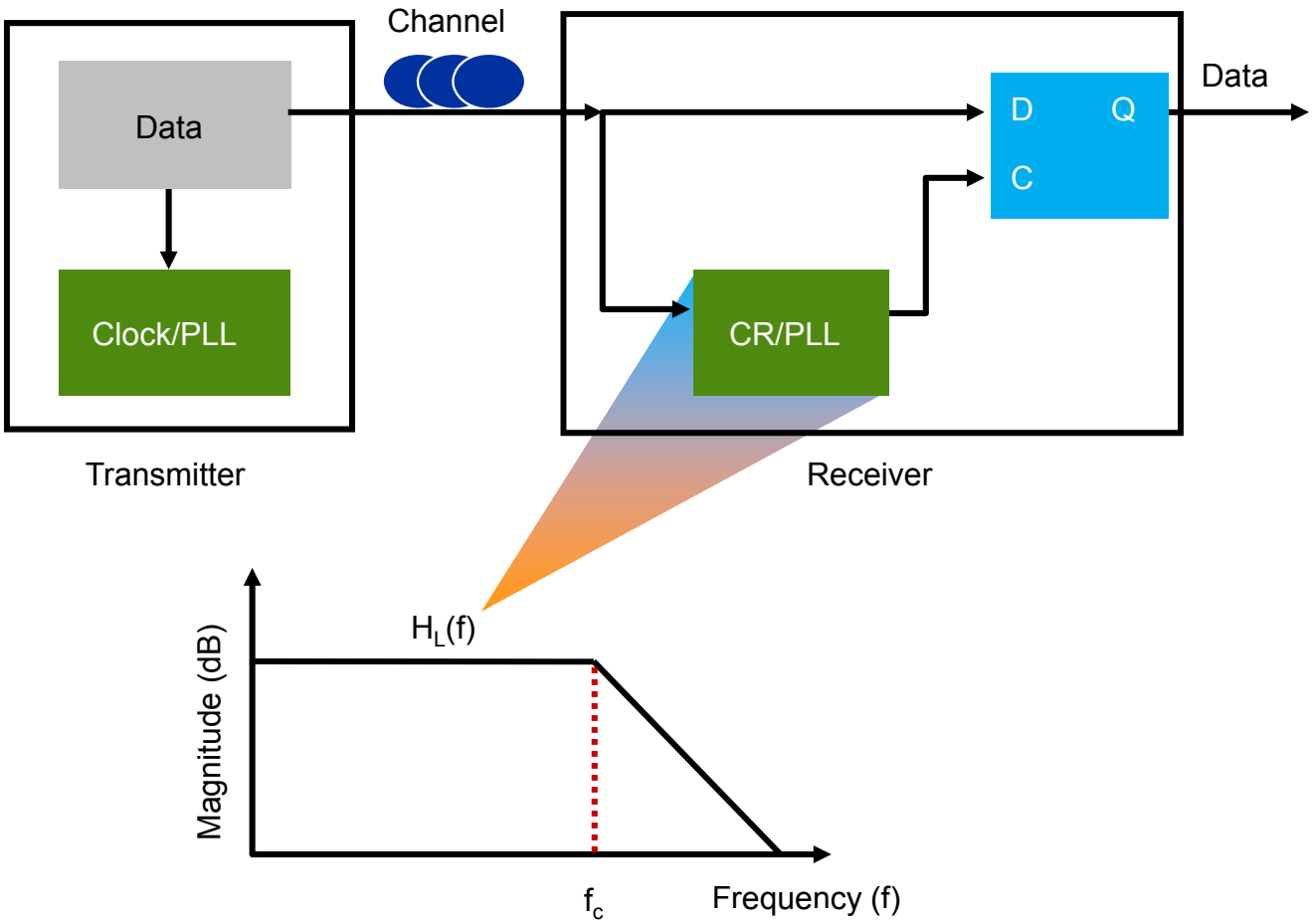
Clock Recovery and Jitter Tracking



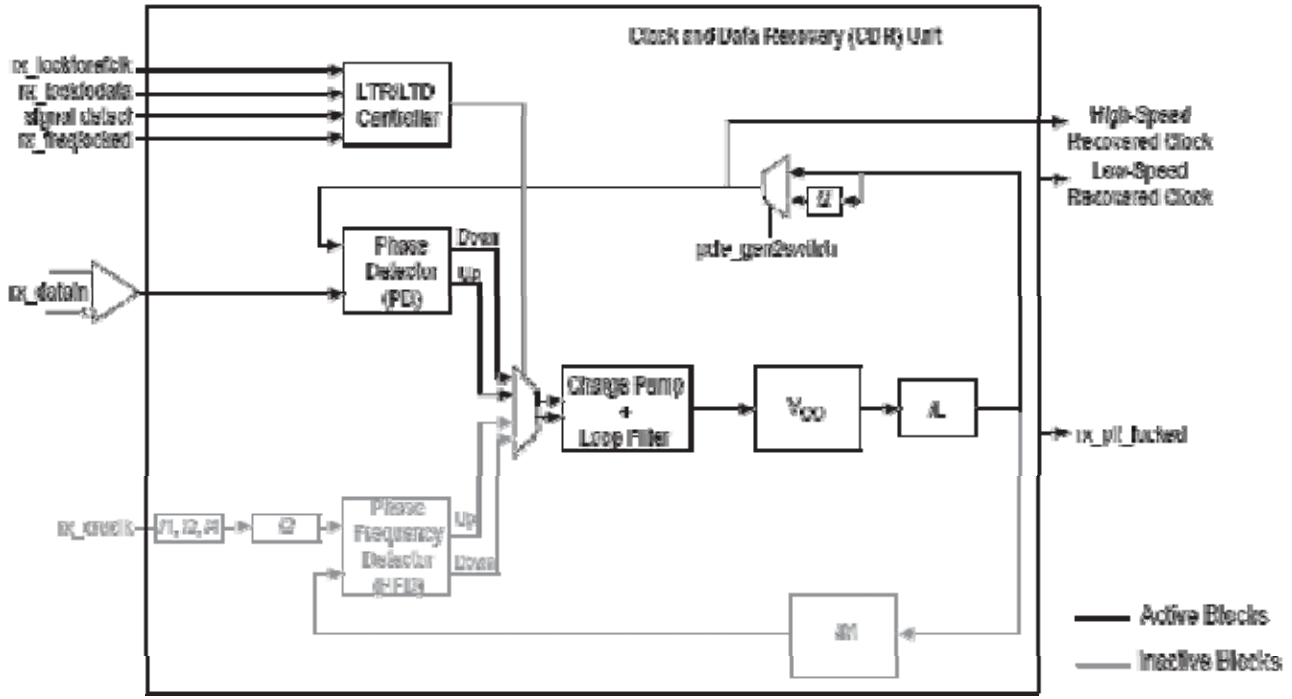
HSIO Link Architecture Advancement Path



Serial Data Communication System Using a Transceiver

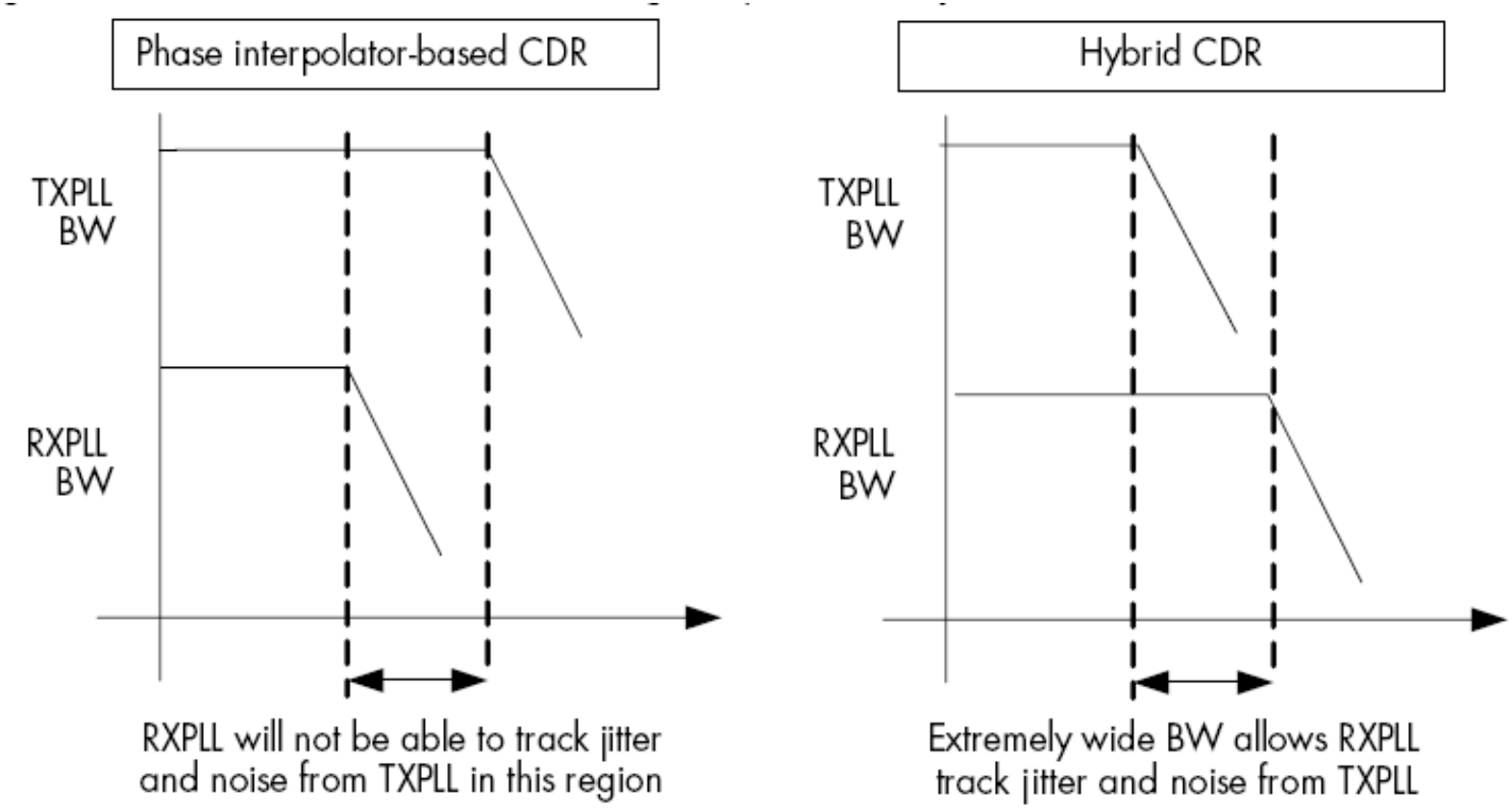


Clock and Data Recovery (CDR)



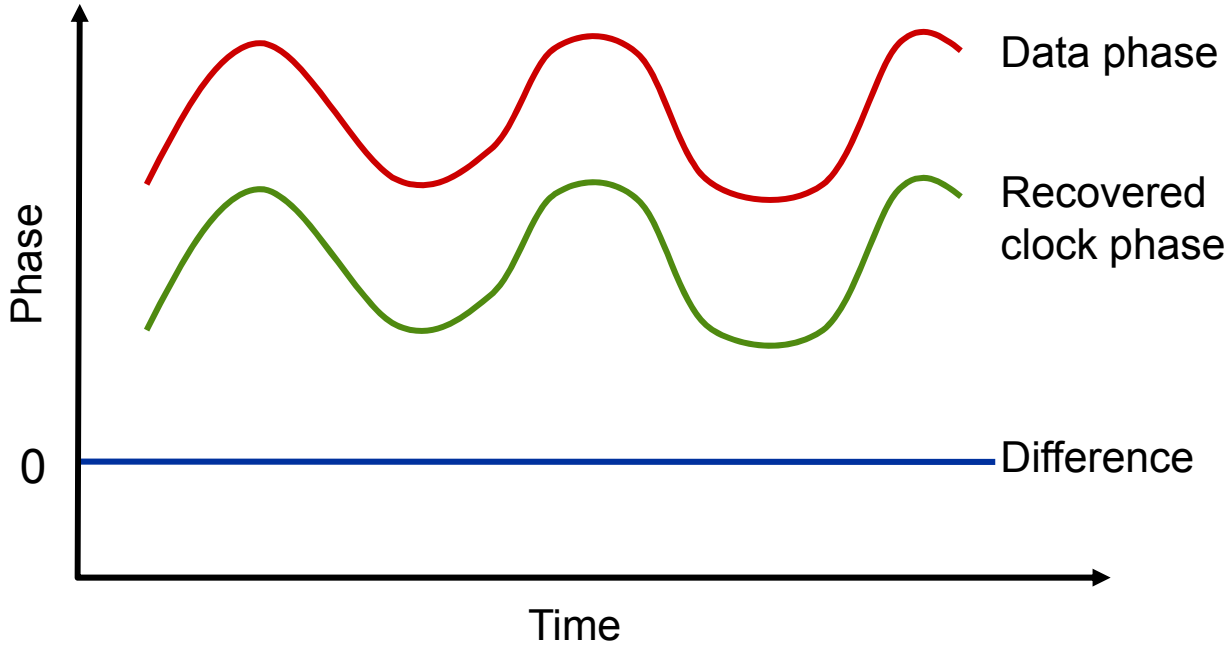
- CDR has lock-to-clock and lock-to-data modes
- CDR is first locked to the reference clock, then switched to lock the data, providing fast locking time
- No unlocked or out-of-lock problems when the received data has excessive jitter
- Reference clock jitter does not affect CDR jitter performance

The importance of a high bandwidth

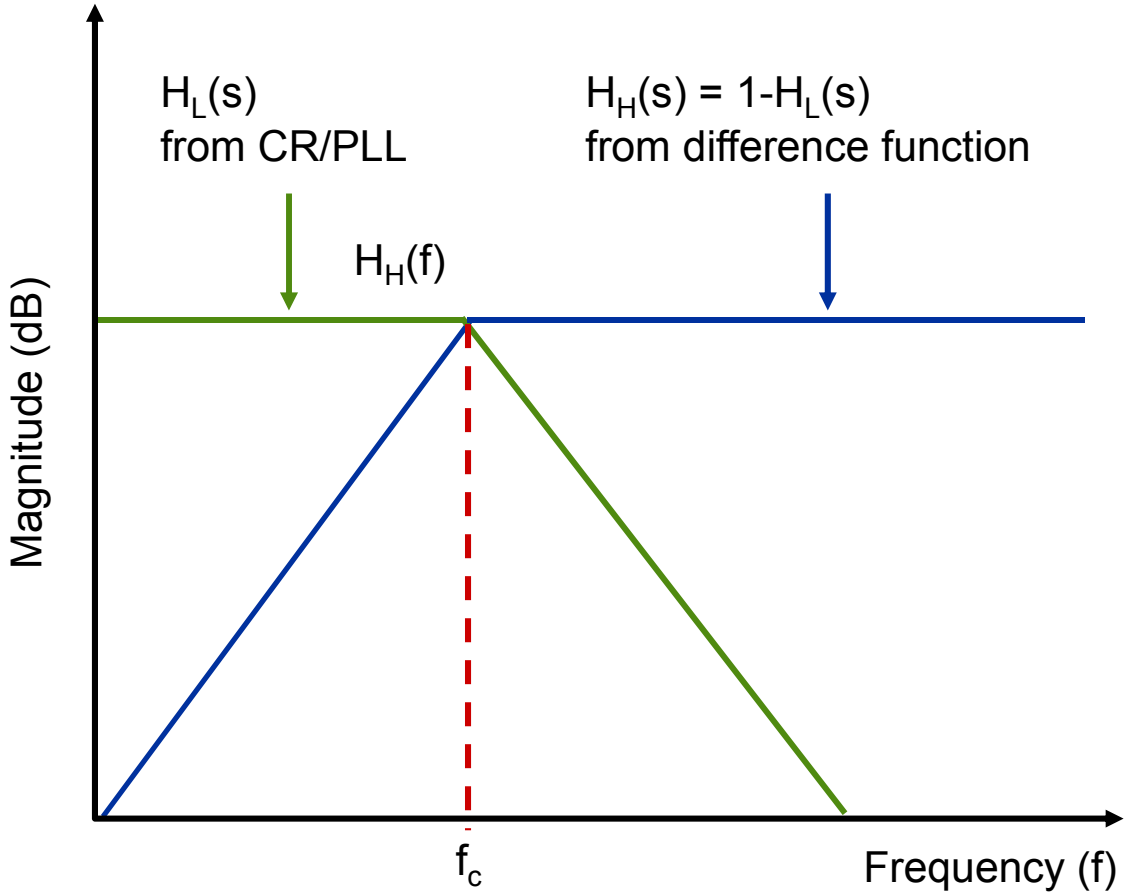


Jitter and Reference Clock

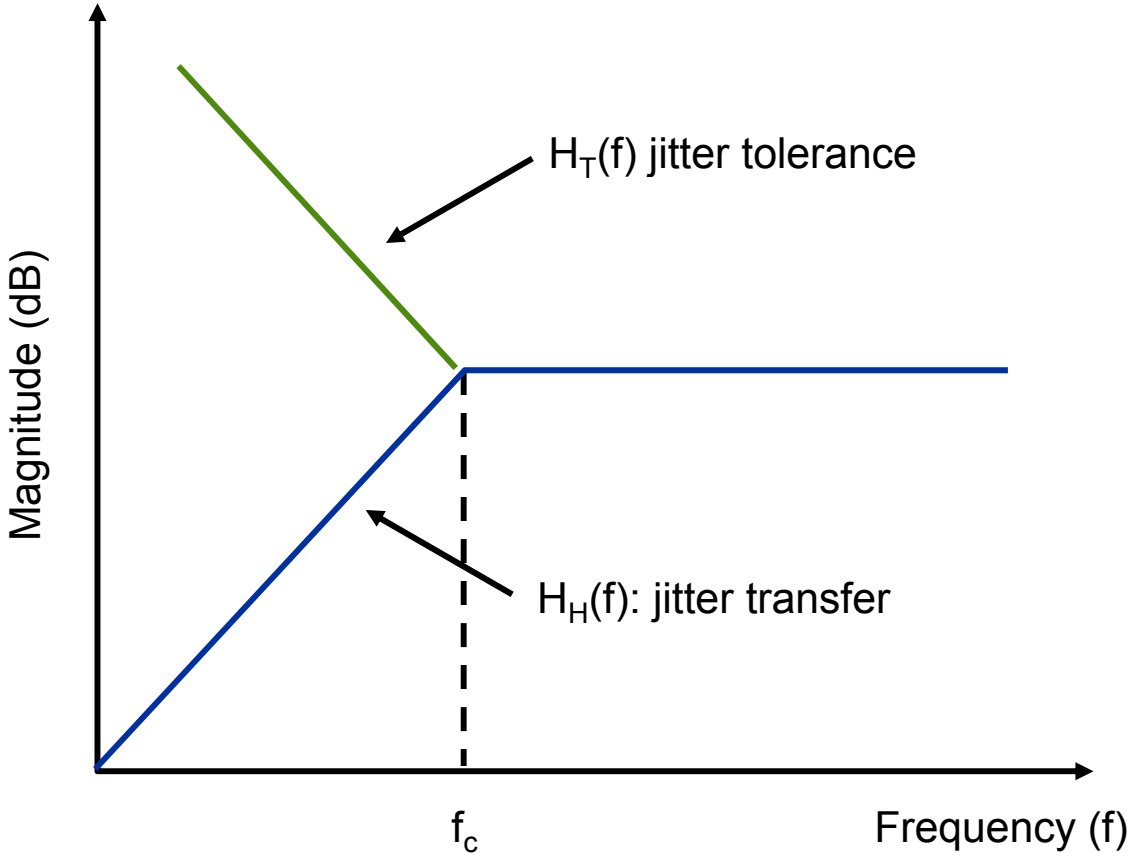
- What matters in a link system is the relative jitter referenced to the recovered clock
- The net jitter is the phase difference between the data and the recovered clock



Jitter Transfer Function

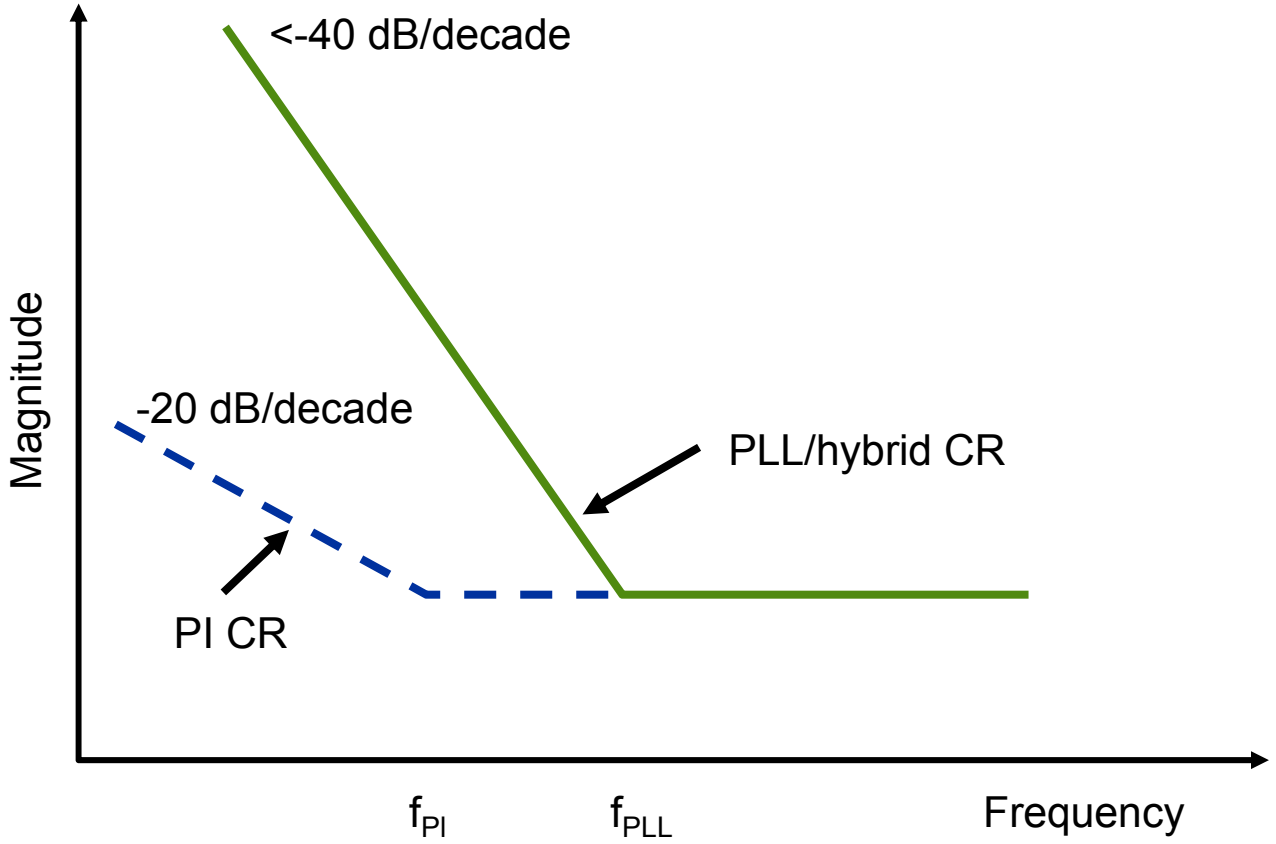


Jitter Tolerance Function



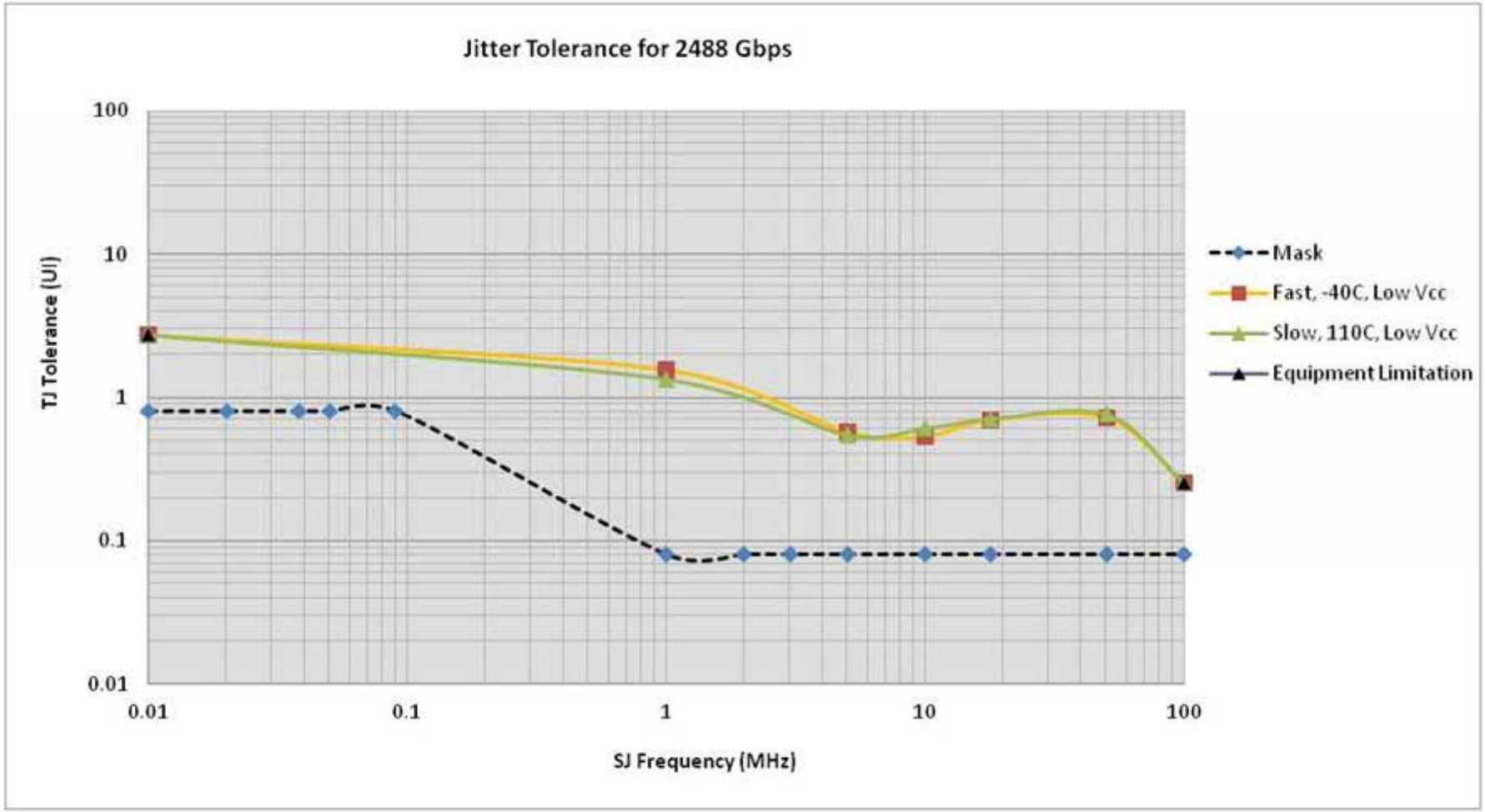
- Higher f_c, better jitter tracking, better jitter tolerance

Jitter Tolerance Comparison



- Receivers have a hybrid phase-locked loop (PLL)-based CDR technology that has the best jitter generation and jitter tolerance performance

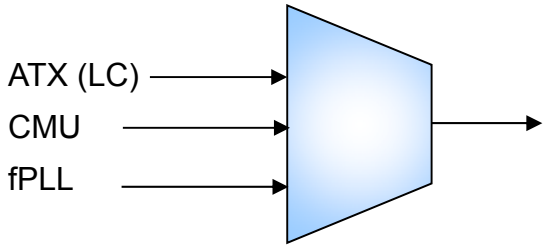
The proof !



Transmitter PLL

Transmit Clocking Options

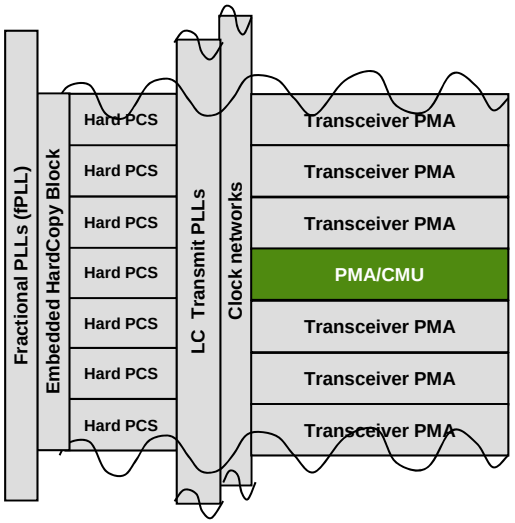
- Clocking options enabling independent data rates and flexible clock distribution network
- ATX PLL (LC)
  - Wider operating range enabled by different VCO modes
  - Sub-picoseconds of jitter for high statistical reliability
- Clock Management Unit (CMU)
  - Ring Oscillator for continuous operating range with best in class jitter
- fPLLs can be used for transceivers in addition to general device clocking



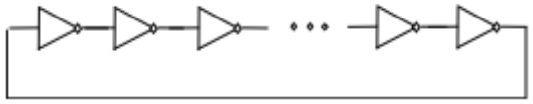
Transmit PLL option	Data Range (Gbps)	Benefit
ATX (LC) PLL	1 – 14.1	Best jitter performance
CMU (Ring Oscillator)	0.6 – 12.5	Best data-rate range
fPLL	0.6 – 3.75	Additional PLL source

Up to 44 independent 12.5Gbps channels on a single Stratix V FPGA

CMU PLL Operating Range



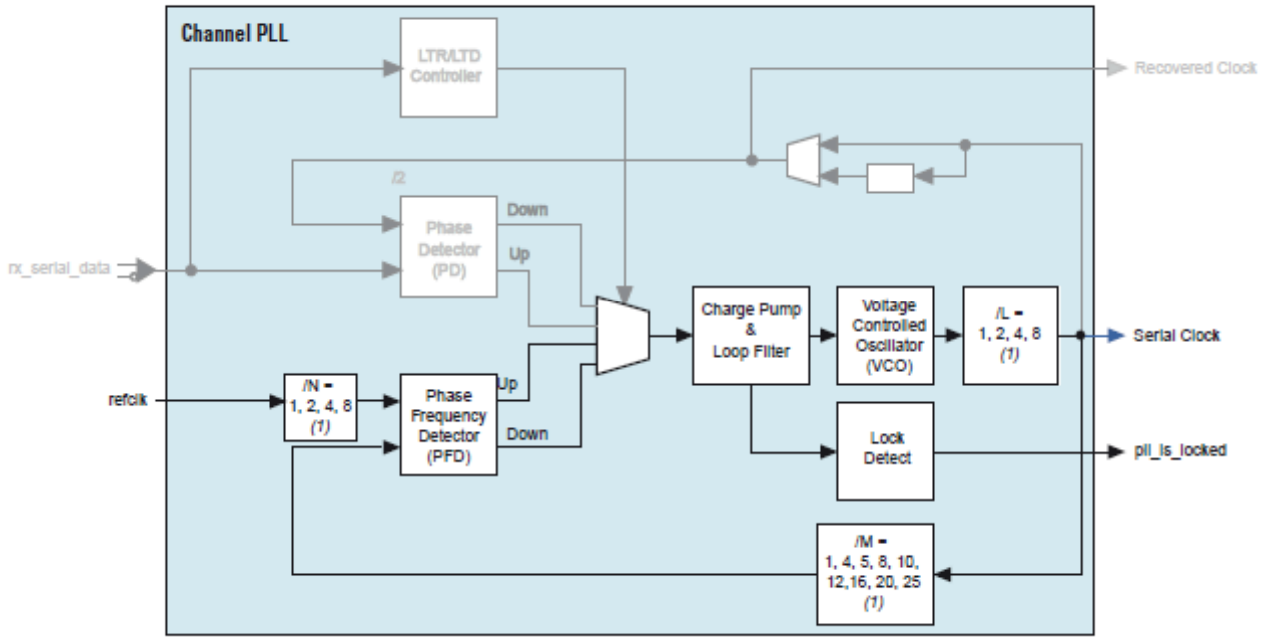
- Each channel can be configured as data channel or as CMU (clock management for the transmit PLL)
- Similar CMU to Stratix IV - Ring-oscillator
- Wide operating range with low transmit jitter
  - 0.6Gbps – 12.5Gbps



CMU PLL

If you use Channel PLL as CMU PLL the RX is unavailable (no CDR)

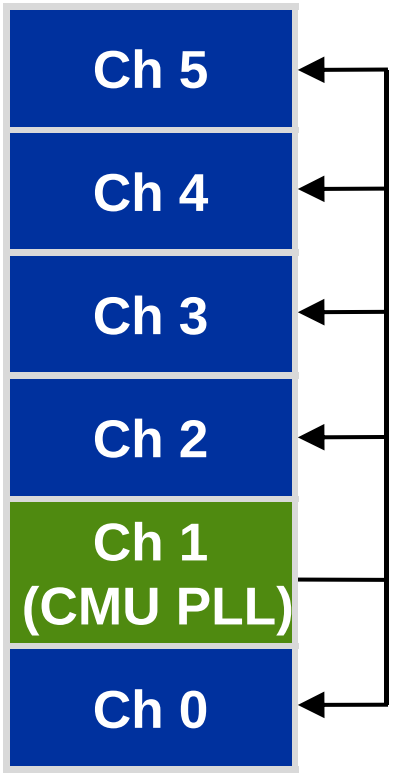
Figure 1–10. Channel PLL Configured as CMU PLL in Stratix V Devices



Note to Figure 1–10:

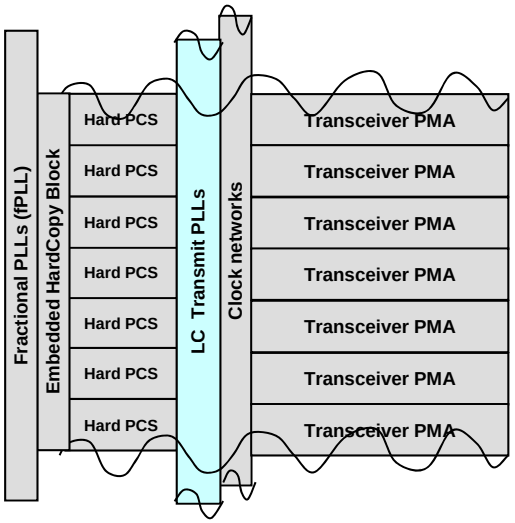
(1) Not all combinations of /N, /M, and/L values are valid. The Quartus II software automatically chooses the optimal values.

CMU PLLs

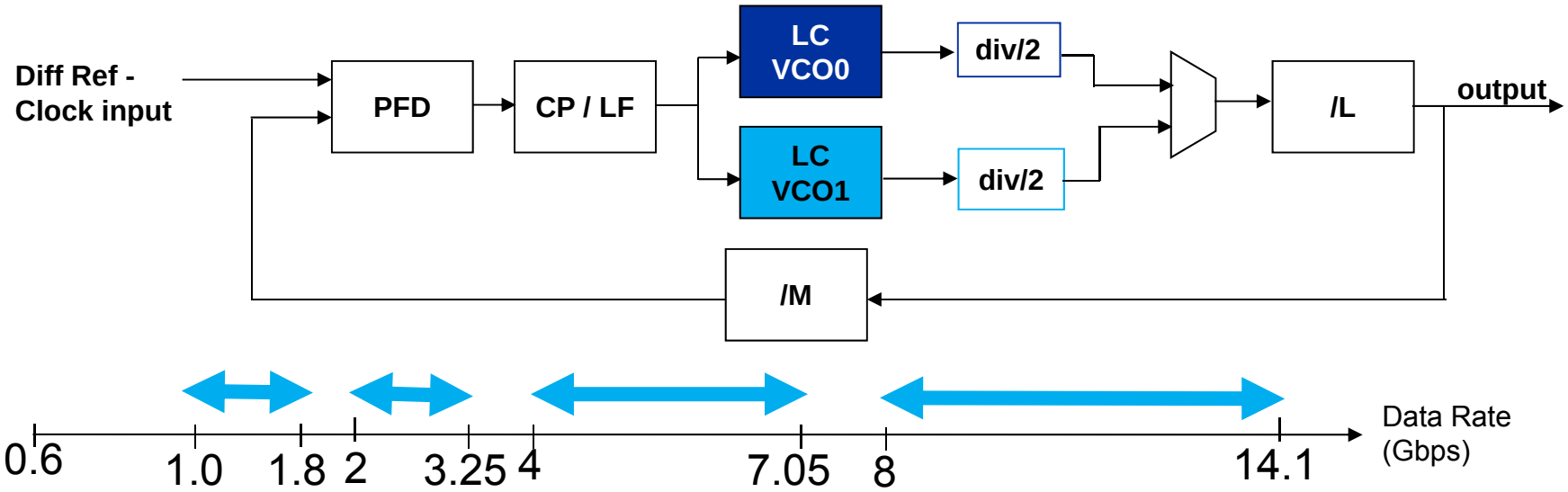


- CMU PLLs in channels 1 and 4 in 6 transceiver bank can generate clocks for other transceiver channels within or outside transceiver bank
- CMU PLL channel may still be used as TX-only channel

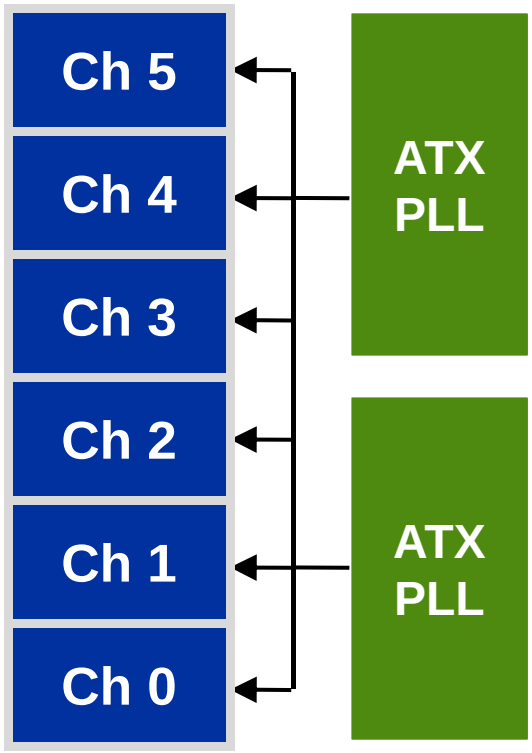
Programmable ATX PLL



- 1 ATX PLL / 3 channels (triplet)
- Sub-picoseconds jitter achieved with LC PLL
- Programmable ATX (LC) PLL Range
  - 8 Gbps - 14.1 Gbps
  - 4 Gbps – 7.05 Gbps
  - 2 Gbps – 3.525 Gbps
  - 1 Gbps – 1.7625 Gbps (using local divider)

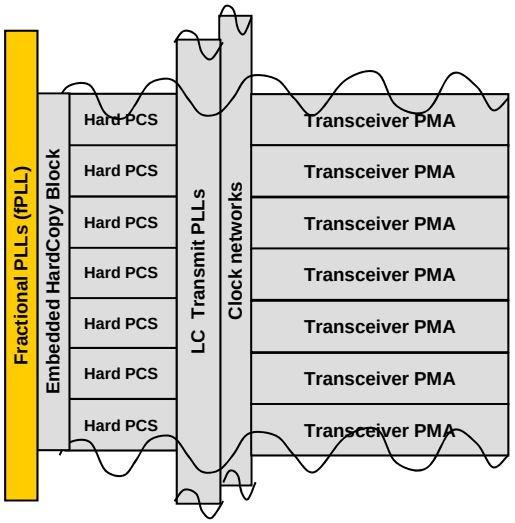


Stratix V ATX PLLs

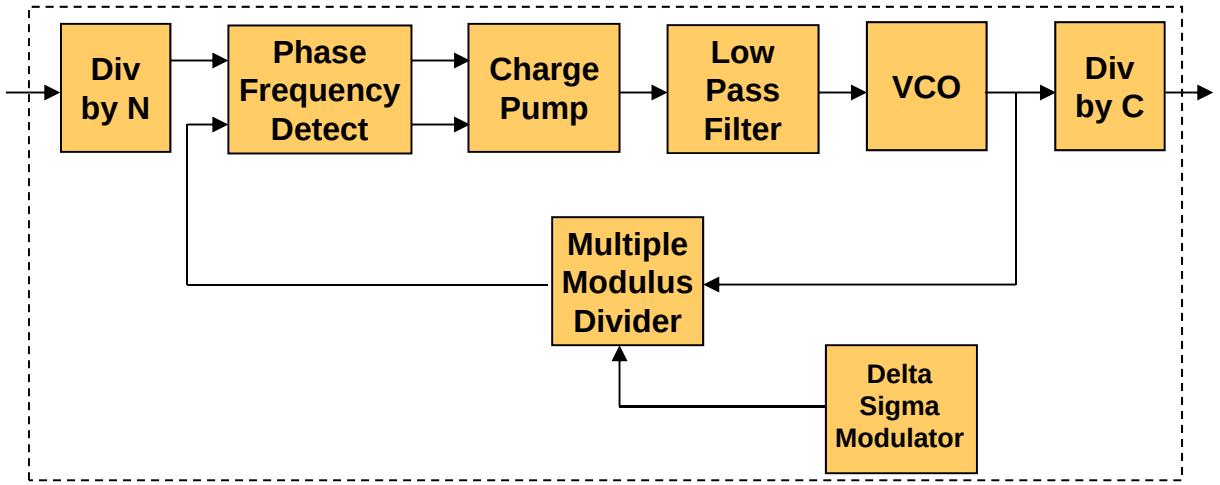


- Better jitter performance at higher data rates than CMU PLLs
- Limited frequency support
  - Must be tuned to support a target data range
- Two per 6-transceiver bank
- Allow full use of device transceiver channels
  - Channel PLL can be used as CDR PLL
- Stratix V GT (28G) channels must use ATX PLLs

Fractional PLL (fPLL)

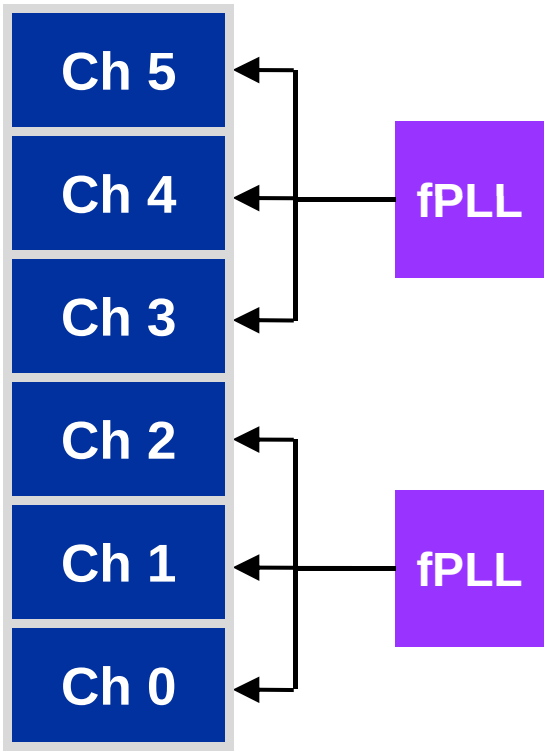


- When used as transmit PLL
  - 0.6Gbps – 3.25Gbps
  - 1 fPLL / 3 channels
  - Integer mode only



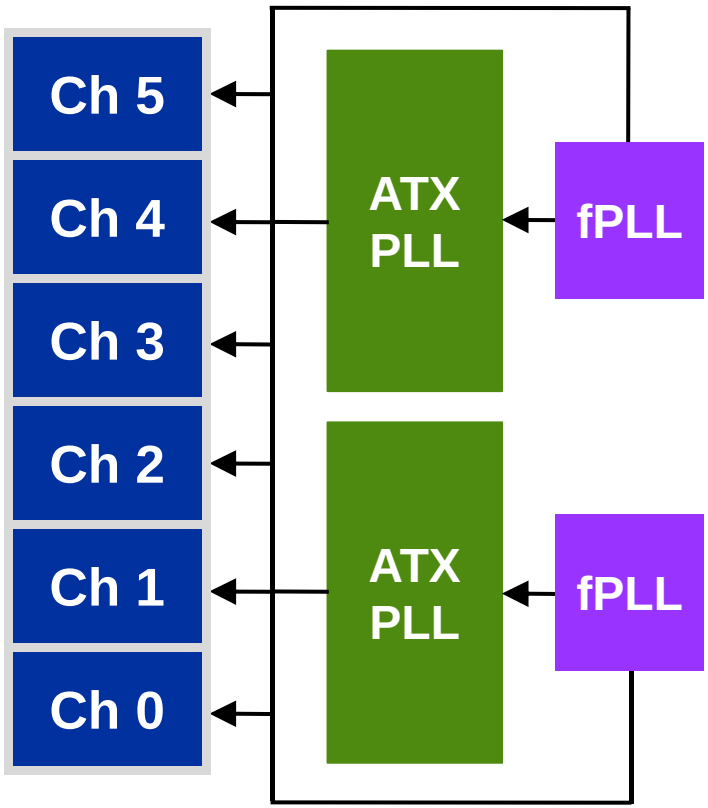
Stratix V fPLL block diagram

Fractional PLLs



- Can serve directly as transmitter PLL
  - Data rate dependant
- Provide increased multiplication/division factors over CMU PLLs
- Allows full use of device transceiver channels
  - Channel PLL can be used as CDR PLL

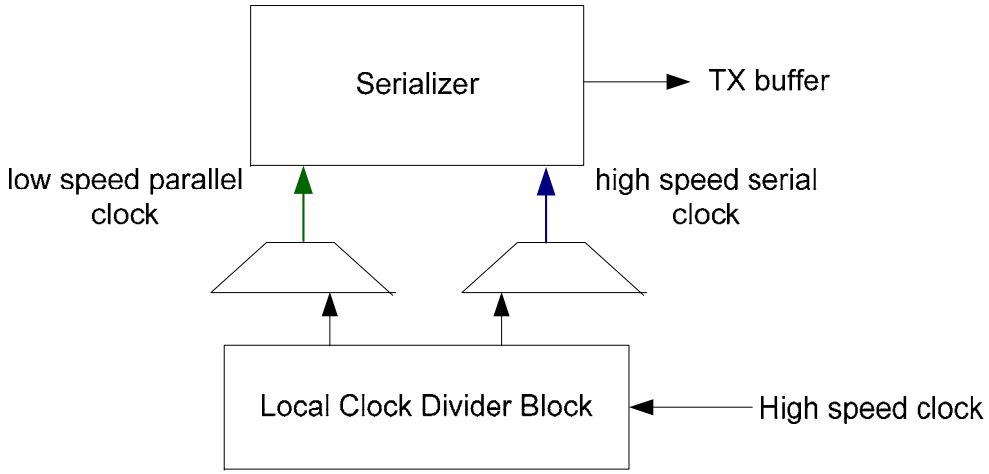
Other use of Fractional PLLs



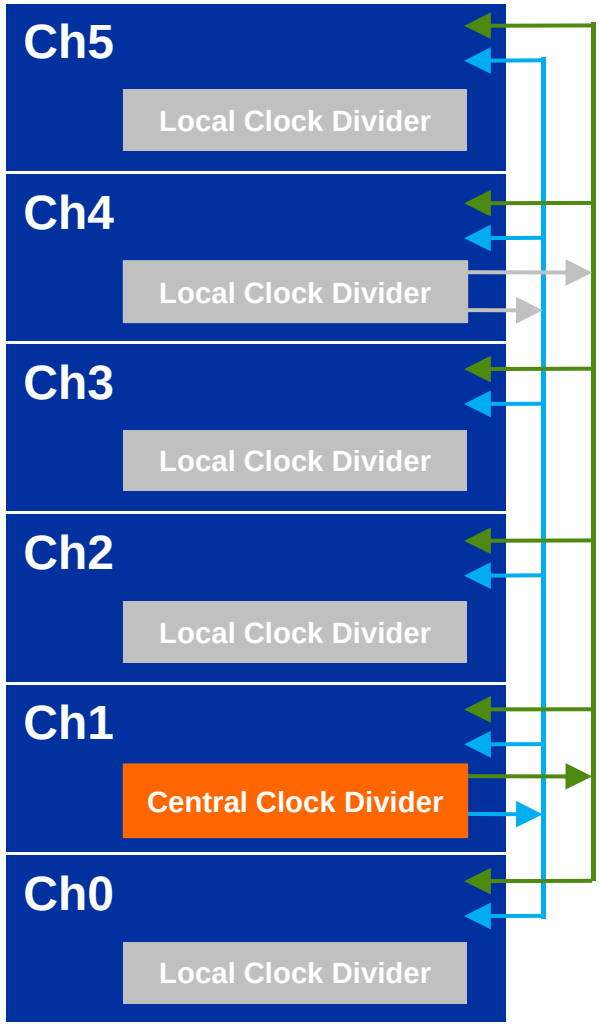
- Supported in Stratix V and Arria V
- Generate reference clock for channel/ATX PLLs
- Provide increased multiplication/division factors
  - Support using a reference clock frequency not directly supported by channel/ATX PLLs
- Drive reference clock lines that span the sides of FPGA
  - Can be segmented per transceiver bank

Transmitter Local Clock Dividers

- One in each channel
- Receive high speed serial clocks from any transmitter PLL
- Generates
  - High-speed serial clock for local PMA
  - Low-speed parallel clock for local PMA/PCS
- Active in *non-bonded* (x1) mode
  - Each channel divides high-speed clock to support its individual target data rate



Central Clock Dividers



- Special functionality of local clock dividers in channels 1 and 4 of transceiver bank
- Each central clock divider can generate high-speed serial AND low-speed parallel clocks for *bonded* channels

High-speed serial clock

Low-speed parallel clock

Transceiver Design

Transceiver PHY IP Cores

- Set of IP Cores that enable/configure the FPGAs PCS/PMA layers for high-speed designs
- Both protocol-specific and generic cores available
- Implementation
  - Embedded transceivers
  - Core logic
  - Both

Protocol-Specific PHY IP Cores

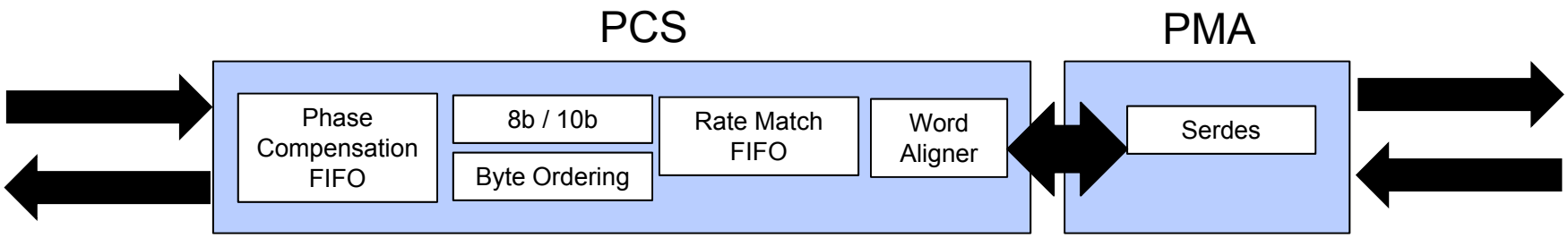
- 10GBASE-R
- 10GBASE-KR
- XAUI
- Interlaken
- PHY IP core for PCI Express
- Hard IP for PCI Express
- …

Non-Protocol-Specific PHY IP Cores

- Custom
- Low Latency
- Deterministic
- Native PHY

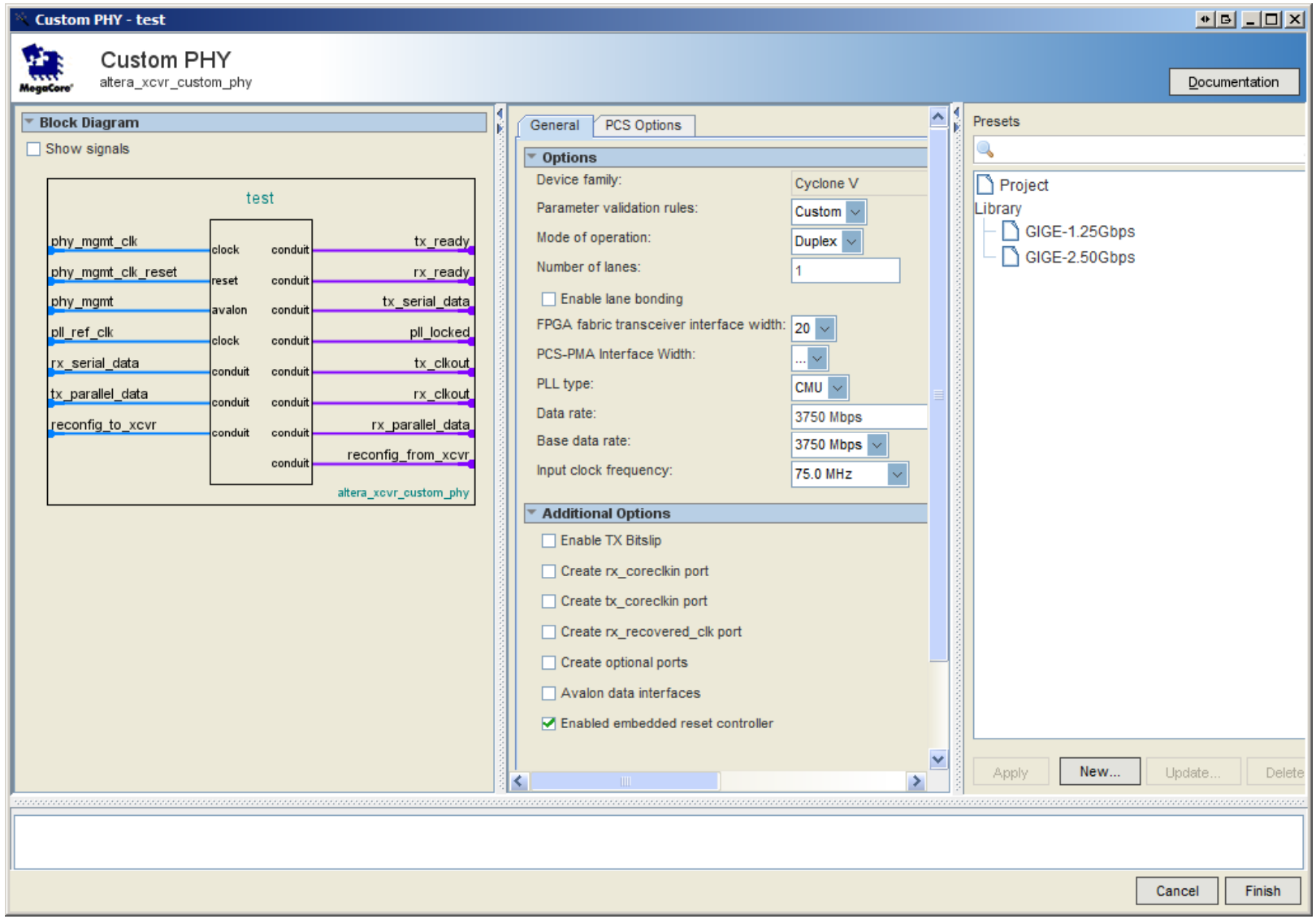
Custom PHY IP

- Overview
  - Provides all PCS blocks
  - Available only for the Standard PCS
  - Alway includes Phase Compensation Fifo
- Generic PHY for supporting custom interfaces
  - Proprietary protocol
  - No protocol-specific PHY IP available

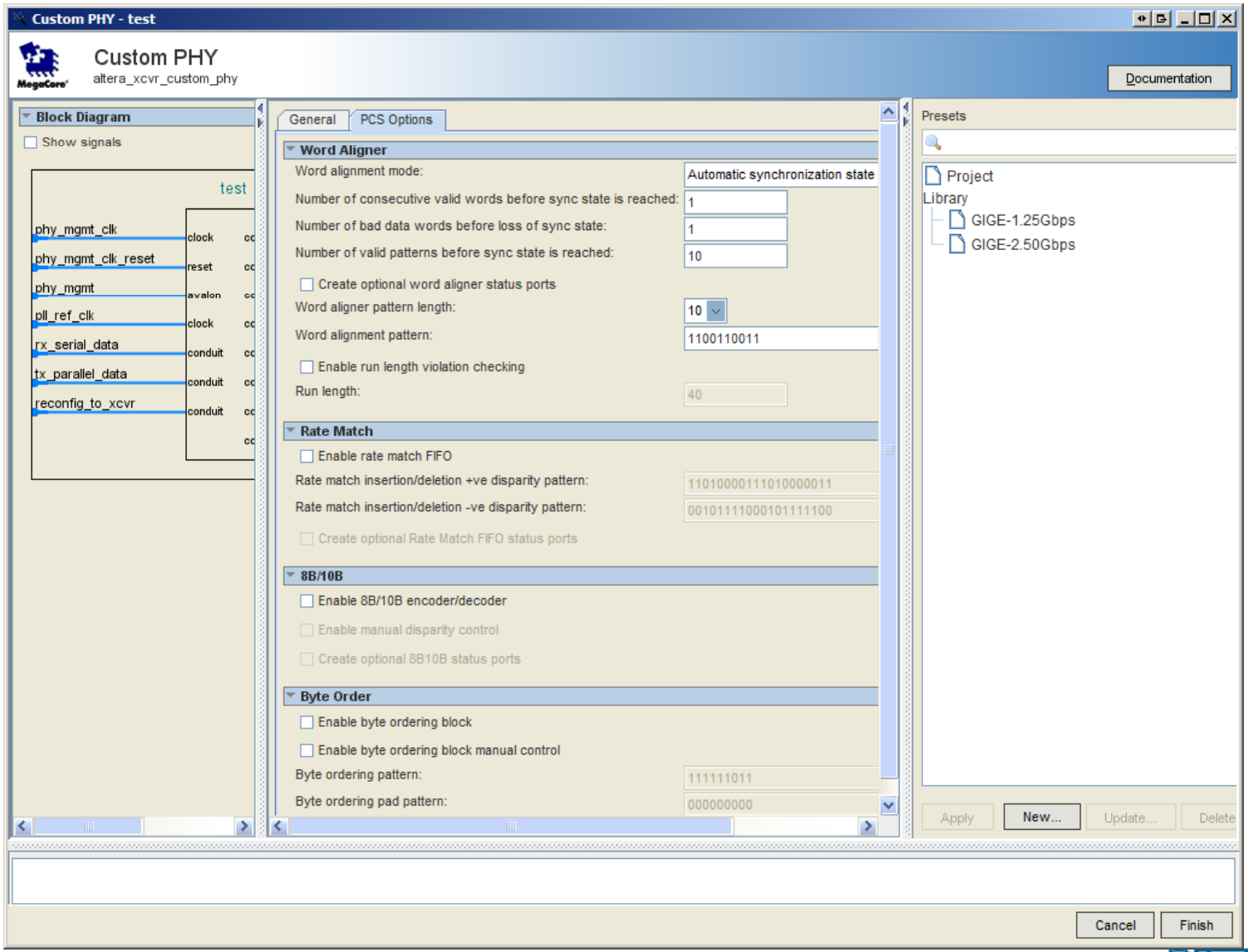


Excellent for custom / proprietary protocols!

Custom PHY IP Parameter Editor (1)



Custom PHY IP Parameter Editor (2)

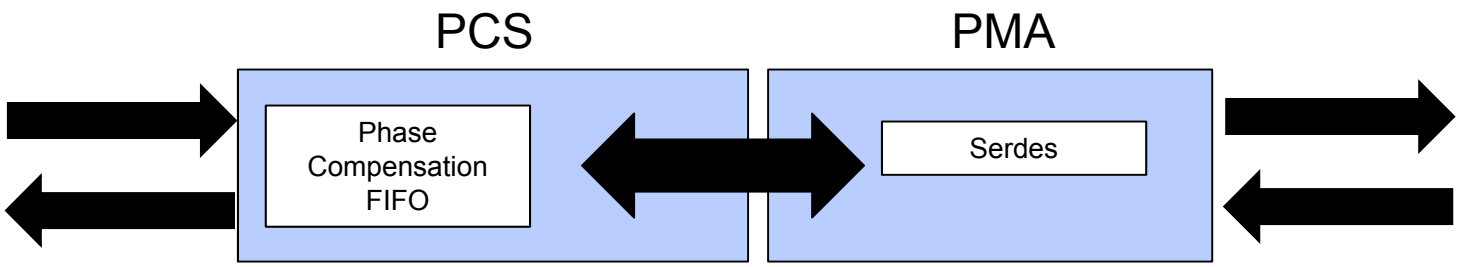


Low Latency PHY IP Core

- Generic PHY for implementing low latency configurations in Stratix V transceivers
  - Protocols/interfaces where specific PCS functionality not available and/or designer wants lowest datapath latency
- PCS functionality must be implemented in the FPGA core
- Contains
  - PMA and minimal (or no) PCS
  - PMA and PCS register map
  - Reset controller (optional)
  - PHY management interface
- Supported devices, PCS configurations and data rates
  - Stratix V Low-Latency PCS : 1 – 14.1 Gbps
    - All protocol-specific blocks bypassed
  - Stratix V GT PMA Direct: 20 – 28 Gbps

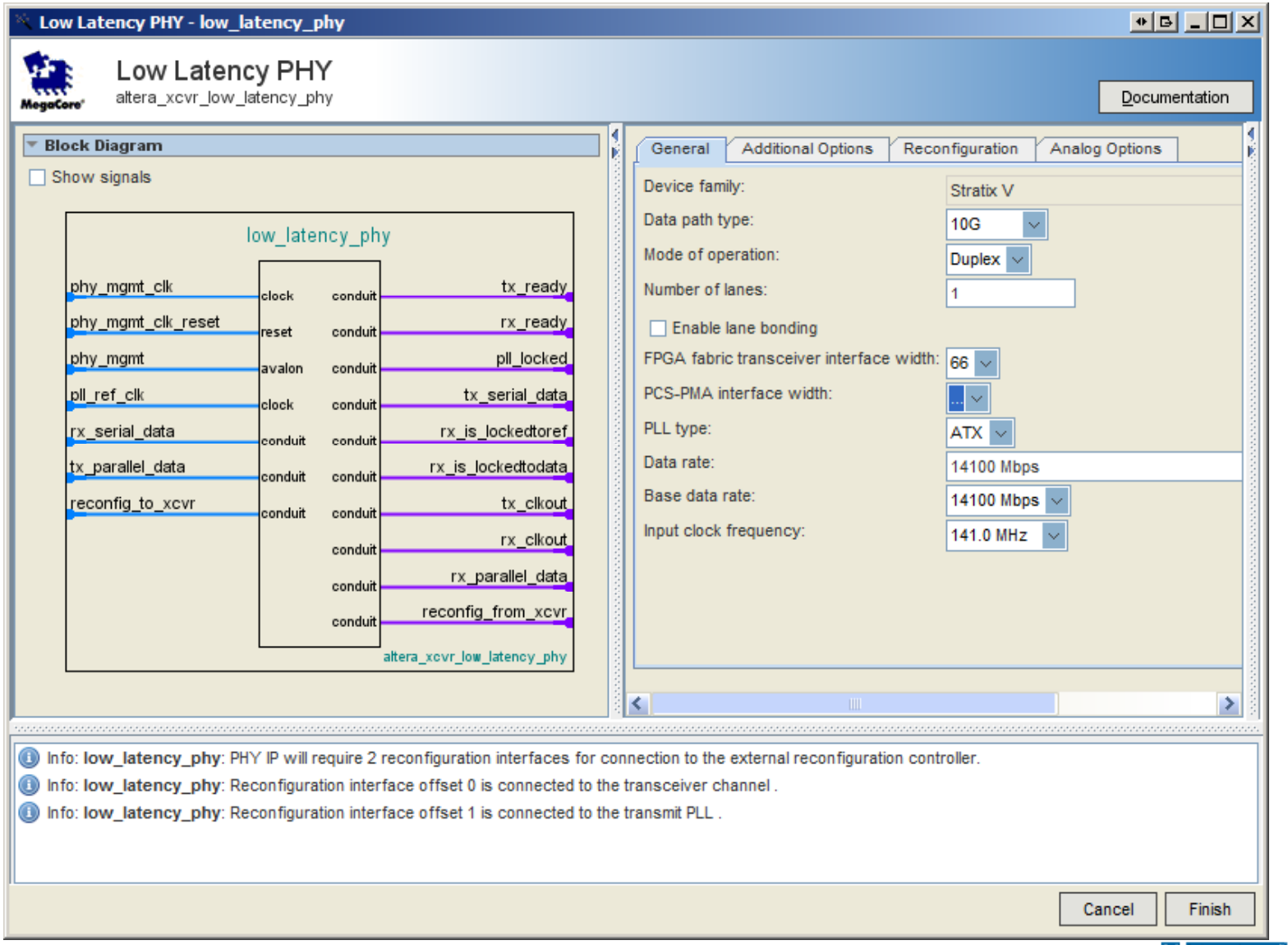
Low Latency PHY IP

- Overview
  - Provides a simple and a low latency path through the PCS
  - Both the Standard and 10G PCS available



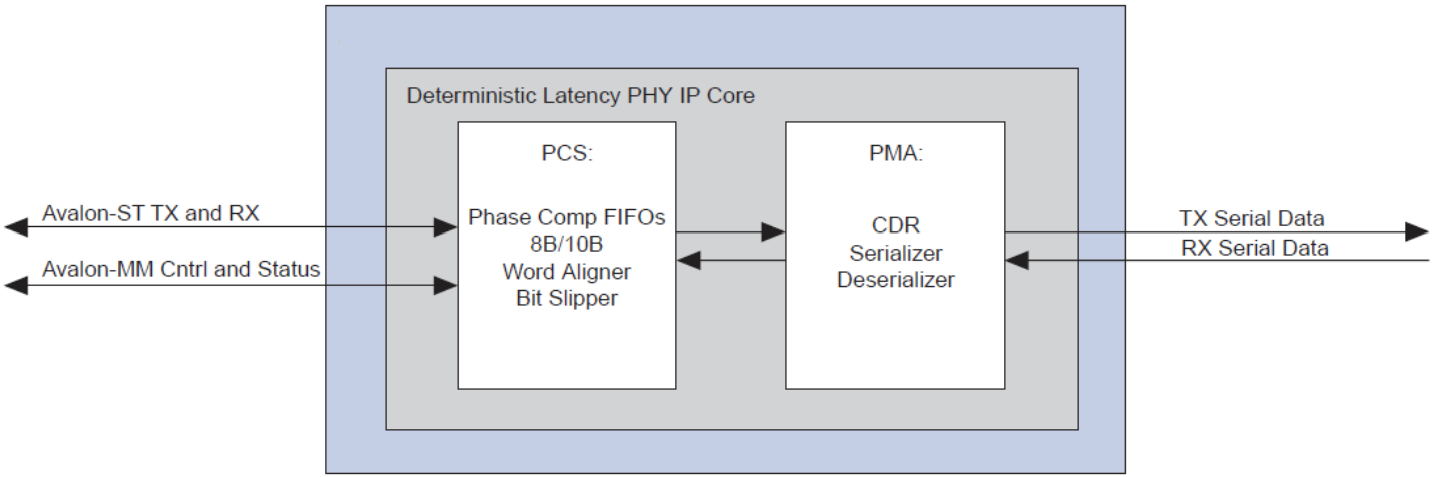
Excellent for custom / proprietary protocols!

Low Latency PHY IP Parameter Editor

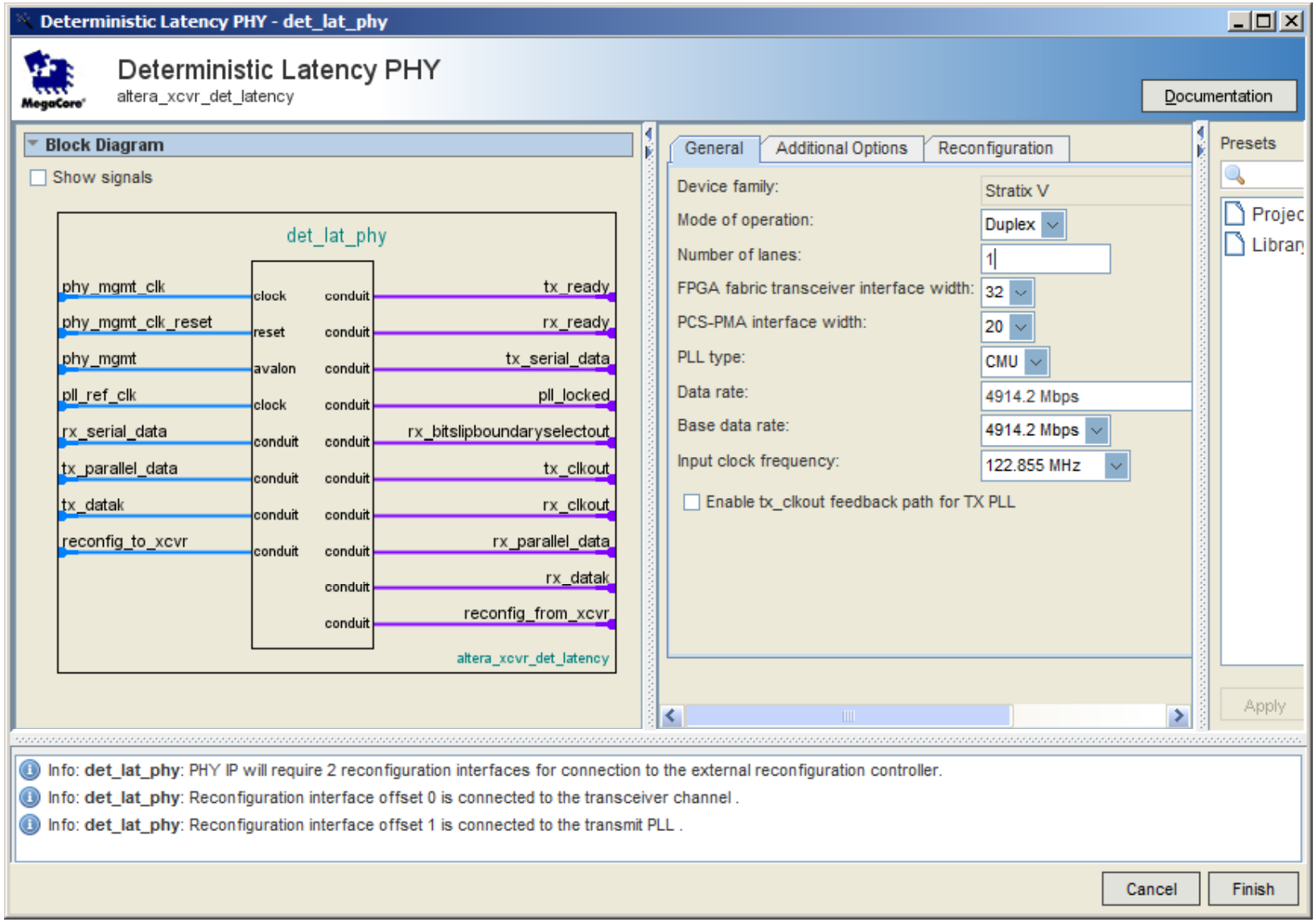


Deterministic PHY

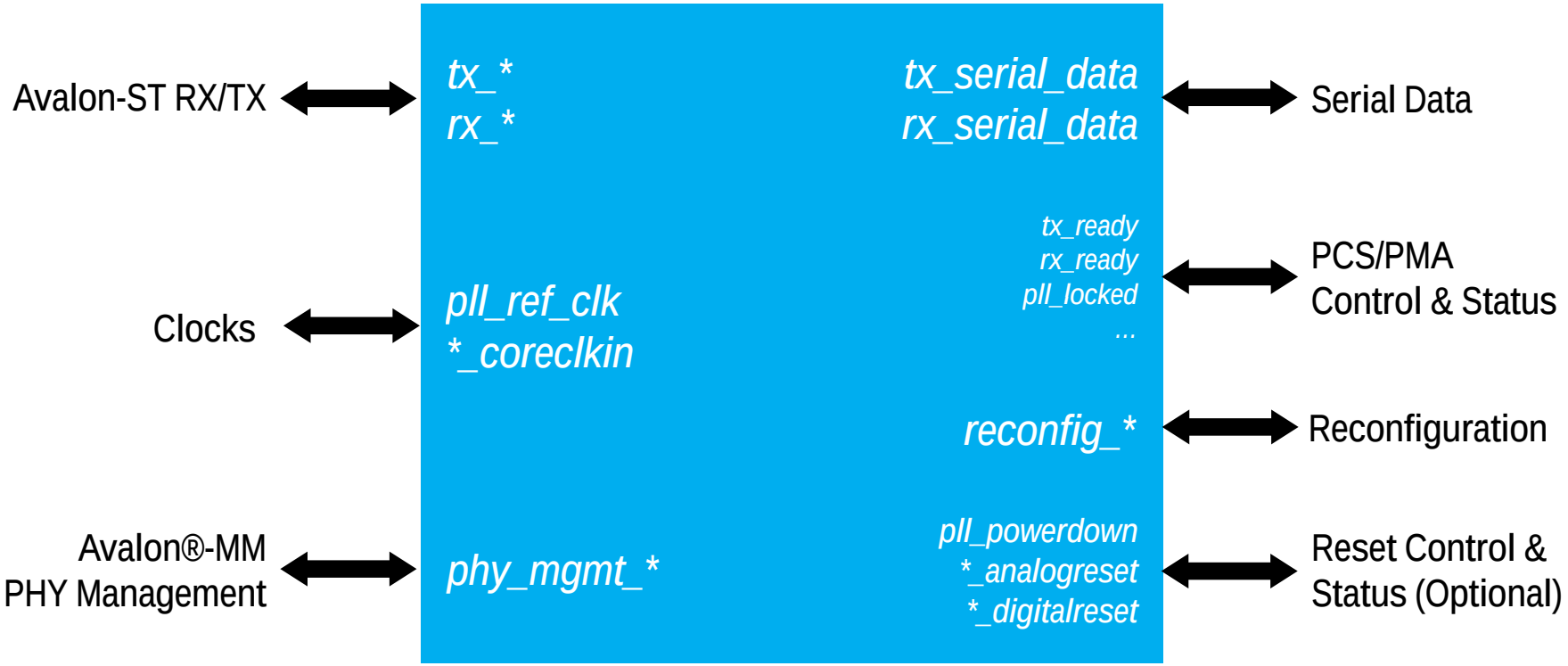
- Overview
  - known timing for the transmit (TX) and receive (RX) datapaths
  - Phase comp fifo in register mode
- Supports
  - Auto-Rate Configuration
    - User initiated via Reconfiguration Controller
  - Support all CPRI/OBSAI data rates in 8G Hard PCS
    - CPRI – 614.4Mbps to 9.8304Gbps
    - OBSAI – 768Mbps to 6.144Gbps



Deterministic PHY IP Parameter Editor



Custom/Low Latency/Deterministic PHY IP Block Diagram



Custom/Low Latency/Deterministic PHY IP Interfaces

- Clock interfaces
  - To reference clock sources
  - To MAC
  - To any additional PCS blocks implemented in FPGA (as needed)
- Avalon-ST TX/RX interfaces – to MAC or user logic
- Serial data interface – to external channel
- Avalon-MM PHY management interface – to MAC or PCS/PMA control logic
- Transceiver reconfiguration interface – to reconfiguration controller
- Reset control and status – reset control logic
- PCS/PMA control and status (optional) – to MAC or PHY control logic

Clock Interfaces

- *pll_ref_clk*: Input reference clock(s)
- *tx_clkout*: Output clock from PHY to use for synchronizing TX output data, control and status signals
- *rx_clkout*: Output clock from PHY; synchronized to RX data, control and status signals
- *tx_coreclkin*: Optional write-side transmitter phase compensation FIFO input clock
- *rx_coreclkin*: Optional read-side receiver phase compensation FIFO input clock

Avalon-ST Input/Ouput Interfaces

- Connects MAC to Custom PHY IP core
- Implements simple Avalon-ST interface with no backpressure or latency
  - Interface is always ready to send/receive data
  - Logic must be ready to send/receive as soon as reset is complete
- Transmit signals
  - *tx_parallel_data*: Outgoing input data to PHY
- Receive signals
  - *rx_parallel_data*: Incoming output data from PHY

** For more information on the Avalon specification, please see the Avalon Interface Specification .*

Serial Interface

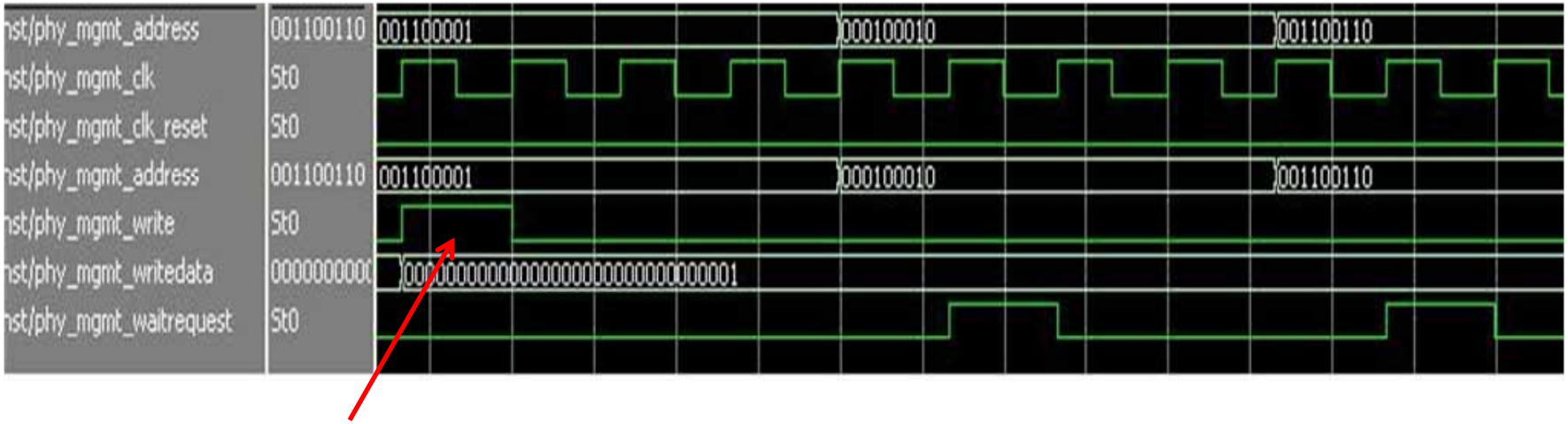
- Connects transmitter data output(s) and receiver data input(s) to external serial interface
  - Backplane
  - Physical medium dependent (PMD) interface
  - Another FPGA
- Signals
  - *tx_serial_data*: Transmitter data output
  - *rx_serial_data*: Receiver data input

PHY Management Interface

- Provides a memory-mapped register space used to access
  - PCS/PMA control and status registers
  - Reset control registers
  - Transceiver reconfiguration registers
- Avalon-MM standard interface
  - 32-bit data, 9-bit address
  - Use read/write transactions to access register space
- Example control/status registers
  - PLL locked status
  - Reset RX/TX channel
  - Bit/byte reversal
  - Manual Wordalignment
  - FIFO overflow/underflow

Note: See Altera Transceiver PHY IP Core User Guilde for register mapping.

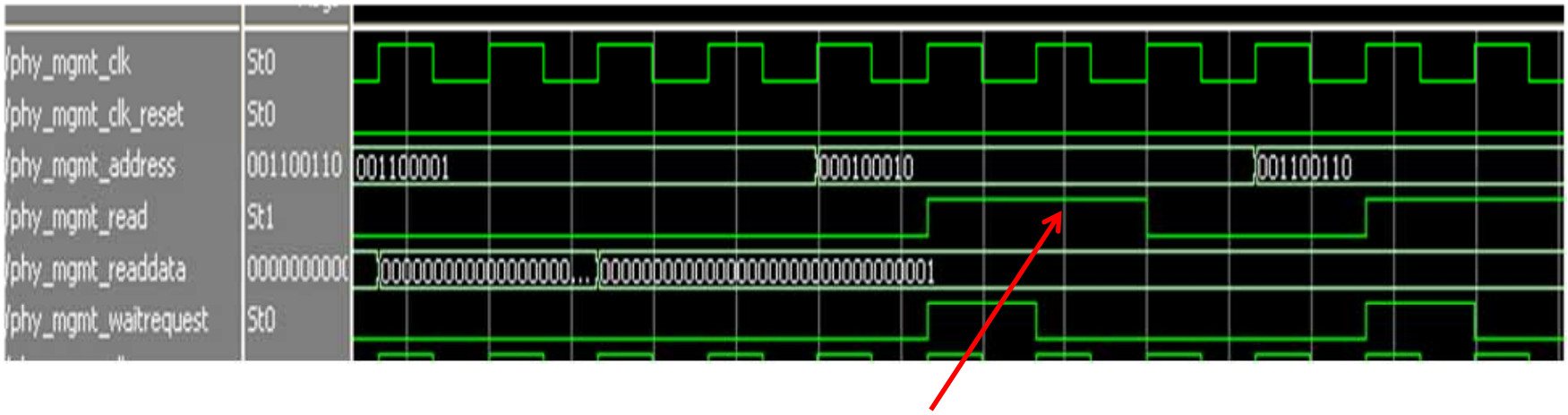
PHY Management Avalon Write Transaction



phy_mgmt_write should be asserted for one clock cycle, because ***phy_mgmt_waitrequest*** is de-asserted

- At rising edge of *phy_mgmt_clk*, provide *phy_mgmt_ address[8:0]*, *phy_mgmt_writedata[31:0]* and assert *phy_mgmt_write*
- Hold these values until PHY IP de-asserts *phy_mgmt_waitrequest*
- PHY IP captures *phy_mgmt_writedata[31:0]*, de-asserts *phy_mgmt_waitrequest* and ends the transfer

PHY Management Avalon Read Transaction



phy_mgmt_read should be asserted for 2 clock cycles, because ***phy_mgmt_waitrequest*** is asserted

- At rising edge of *phy_mgmt_clk*, provide *phy_mgmt_address[8:0]* and assert *phy_mgmt_read*
- Hold these values until PHY IP de-asserts *phy_mgmt_waitrequest*
- PHY IP presents valid *phy_mgmt_readdata[31:0]* and de-asserts *phy_mgmt_waitrequest*

Reset Control & Status Interfaces

- Embedded reset controller enabled
  - *phy_mgmt_clk_reset* (Avalon-MM inteface): initiates reset of PHY
  - *tx_ready*: PHY has exited reset and is ready to transmit data
  - *rx_ready*: PHY has exited reset and is ready to receive data

- Embedded reset controller disabled
  - PHY provides signals to connect Transceiver PHY Reset Controller IP core or user-designed reset controller
    - See device handbook for reset timing diagrams
  - Examples
    - *pll_powerdown* : Resets TX PLL
    - *tx_cal_busy*: Indicates transmit channel is being calibrated
    - *rx_analogreset* : Resets the RX PLL (CDR)
    - *tx_digitalreset* : Resets the TX PCS blocks

Optional PCS/PMA Control and Status Interface

- Additional signals to determine and control state of PCS/PMA
- Provides instantaneous interaction over using PHY management interface
- Examples
  - *tx_datak*: Input to indicate data/control code (8B/10B encoding enabled)
  - *rx_syncstatus* : Indicates single-lane word alignment
  - *rx_rmfifodatainserted* : Indicates Rate Match has inserted skip character
  - *rx_errdetect* : Indicates an 8B/10B code violation or disperity error has occurred
  - *tx_rlv*: Indicates a run length violation has occurred in the receiver
  - *rx_datak*: Data/control code indicator (8B/10B decoding enabled)
  - *rx_runningdisp*: Indicates disparity of incoming data (8B/10B decoding enabled)
  - *rx_enabyteordflag*: Triggers byte ordering

Native PHY

- Give control back to the user
- First IP offering is always a family specific “Native” PHY IP.
  - Lower level than PHY IP and altgx.
  - Nothing but the ports and parameters. No embedded reset controller, no AVMM CSR, no fPLLs, etc.
- Protocol specific PHY IP when it makes sense.
  - PIPE, Interlaken, 10GBase-KR, XAUI are good examples.
- Schedule
  - 12.0 Direct Mode (only PMA)
    - Reset controller as a separate optional IP
  - 12.1 Full PCS Support:
    - Megawizard performs validation and provides feedback for all interfaces and parameters.
    - Megawizard gives warnings rather than errors when rules are violated.

Stratix V Native PHY

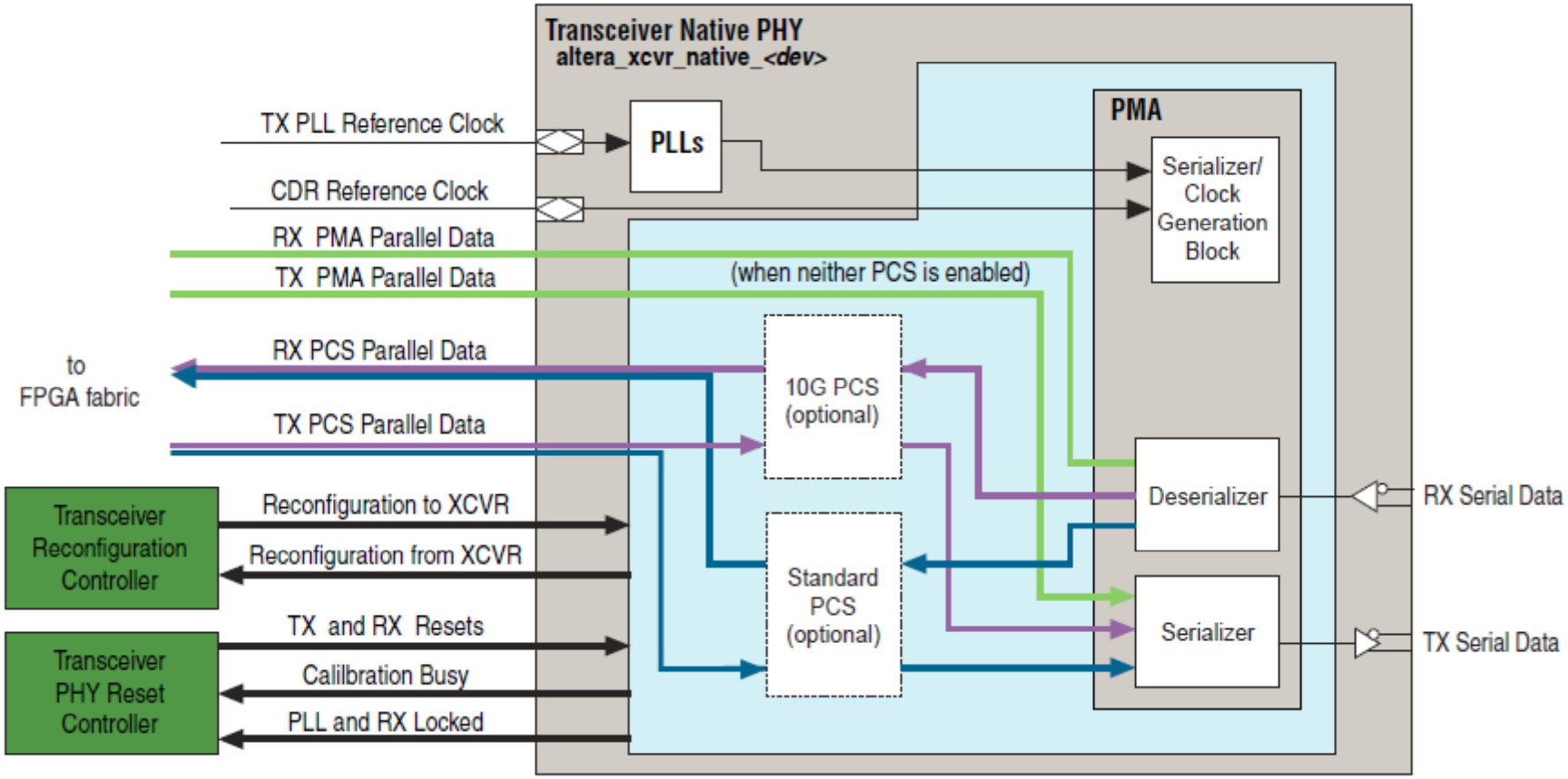
- GUI
  - Parameters are grouped by functional block (8B/10B, word aligner, etc.)
  - Order matches the order from the PCS diagram
- HDL
  - Parameters and ports names are intended to be self explanatory:
  - Parameter naming convention:
    - Standard(8G) PCS – std_rx_<param>, std_tx_<param>
    - 10G PCS – teng_rx_<param>, teng_tx_<param>
  - Port naming convention
    - Standard(8G) PCS – tx_std_..., rx_std_...
    - 10G PCS – tx_10g_..., rx_10g_...
    - PMA – tx_pma_..., rx_pma_...

Native PHY IP Cores

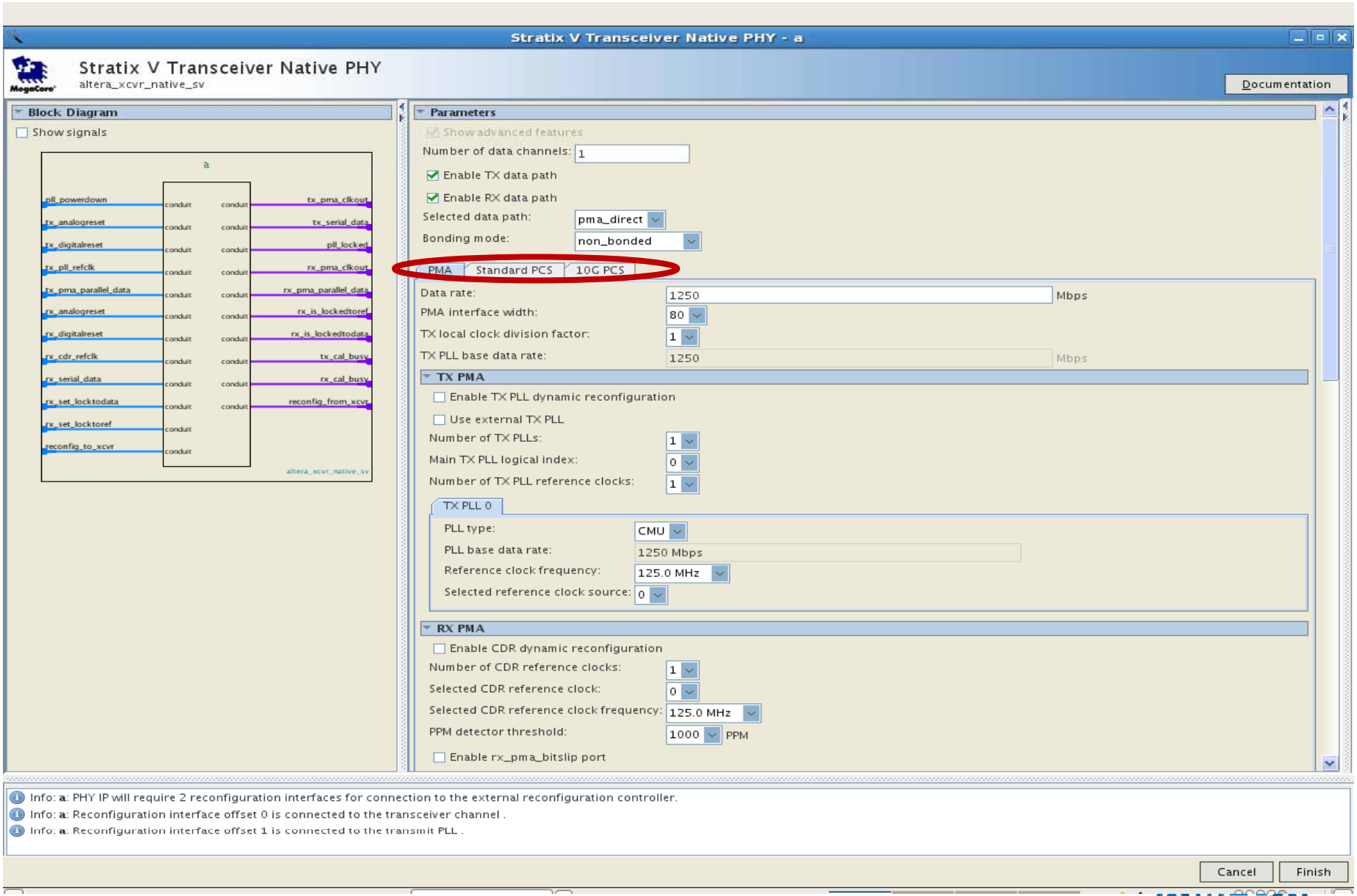
- All PHY control/status signals exposed as ports
  - Similar to enabling “all optional ports” on other PHY IP cores
  - No memory-mapped register interface to control and monitor PHY
  - Must build own “register-space” using ports to access in memory-mapped system (if required)
- Contains PHY (PCS/PMA) only
  - Reconfiguration controller must be connected manually
  - Reset controller must be connected manually
    - Designer can use Transceiver PHY Reset Controller IP core or create own reset controller

**** Other modes enabled in a future version of the Quartus II software***

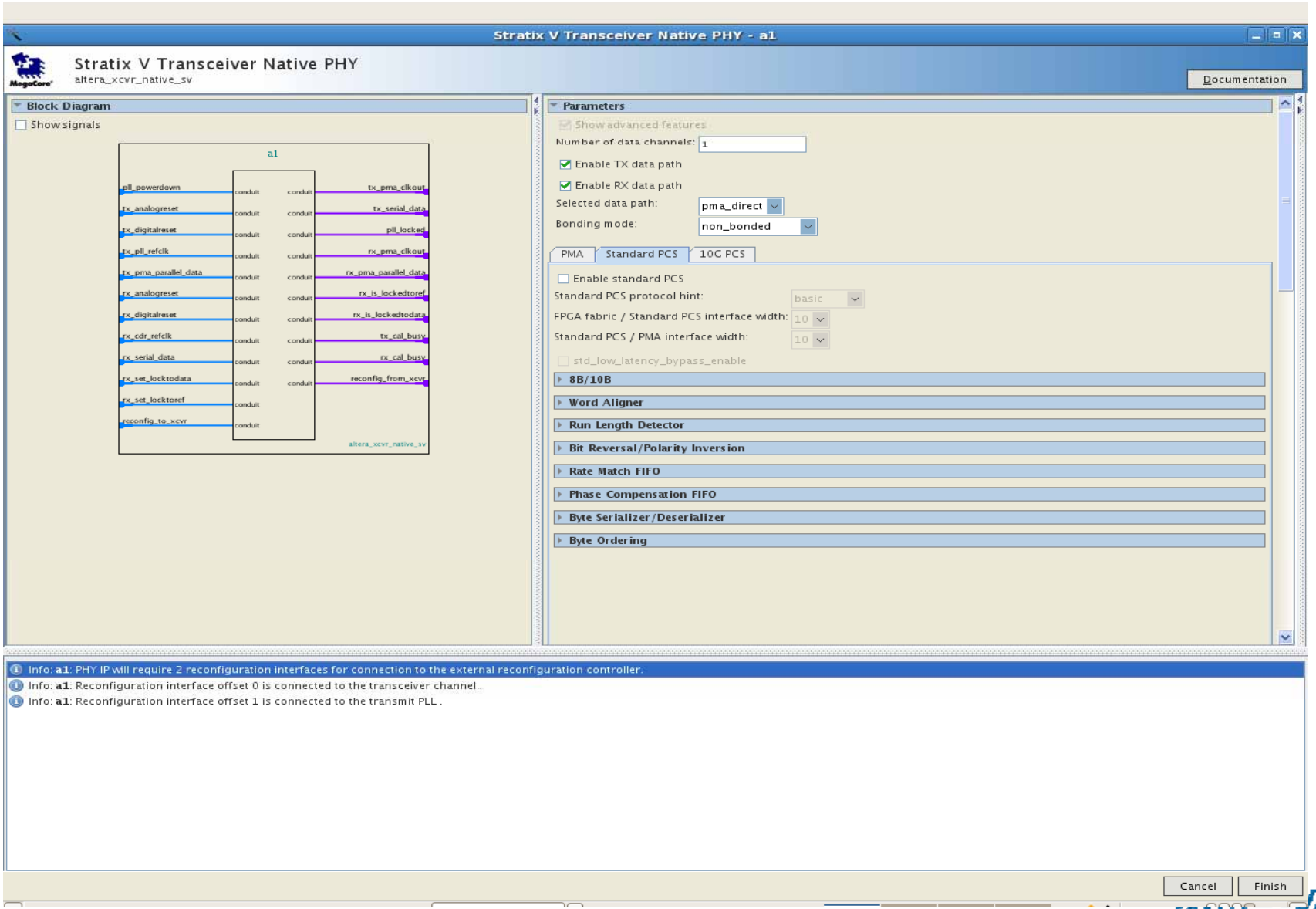
Native PHY



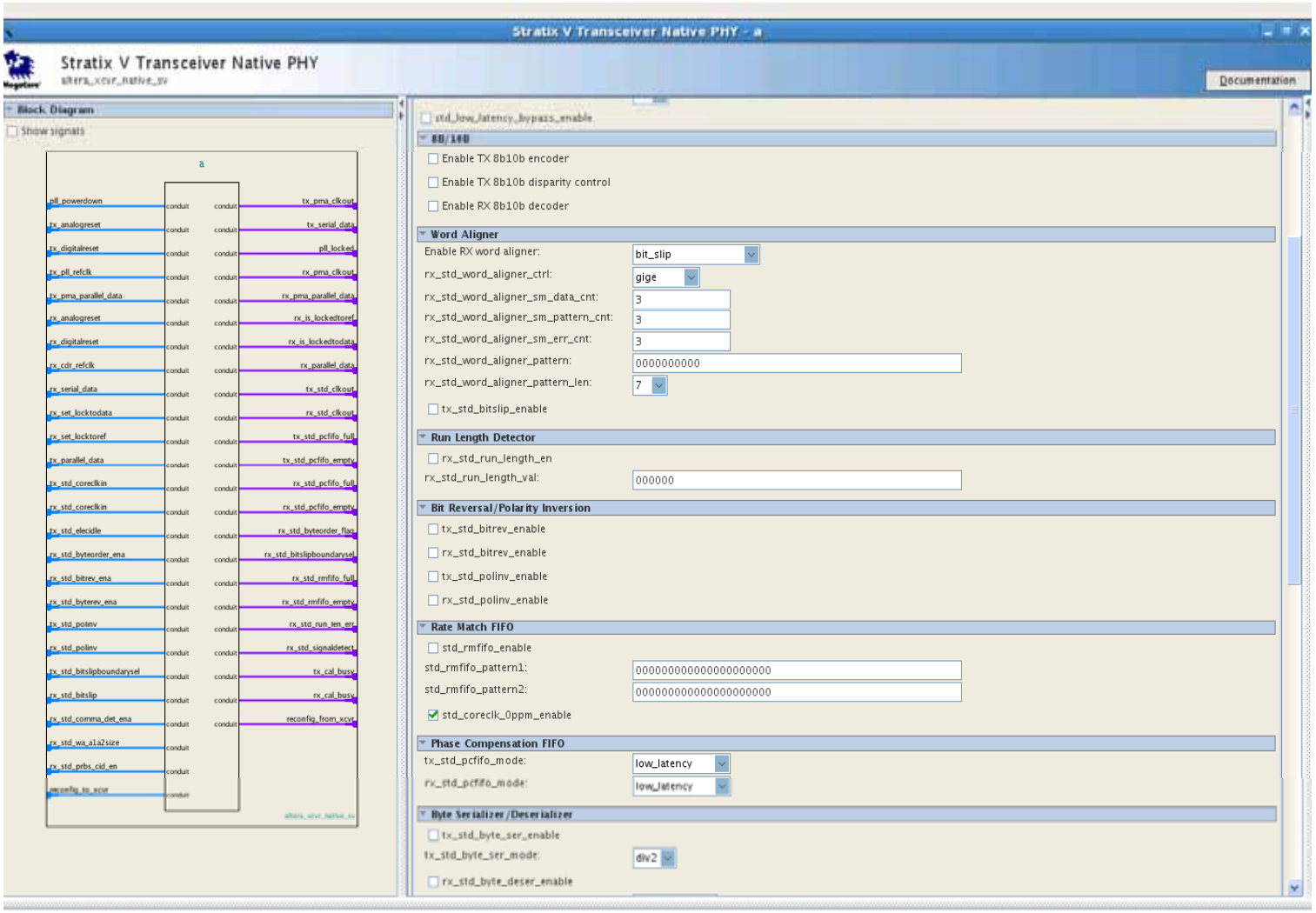
PMA GUI



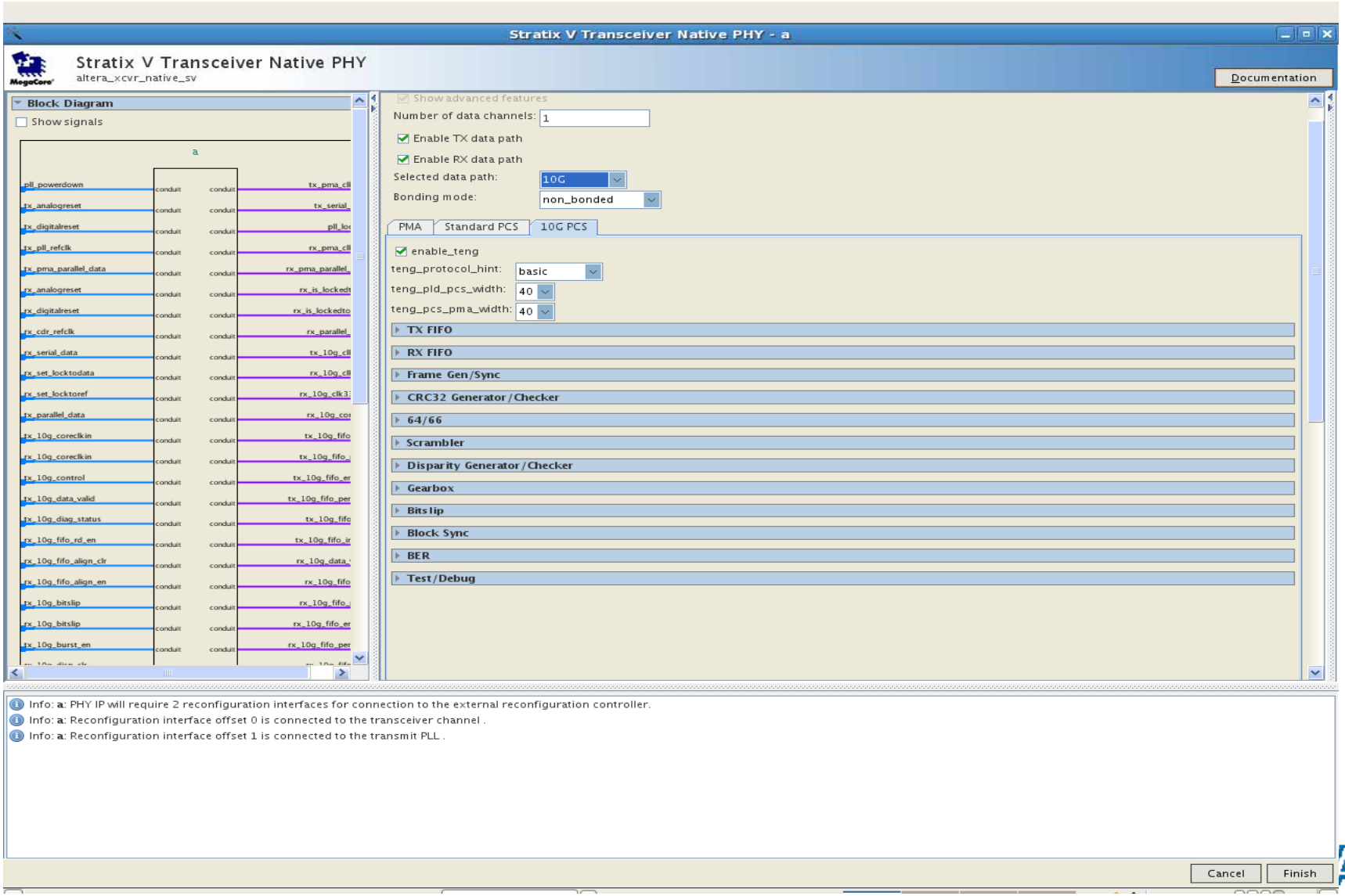
Standard PCS



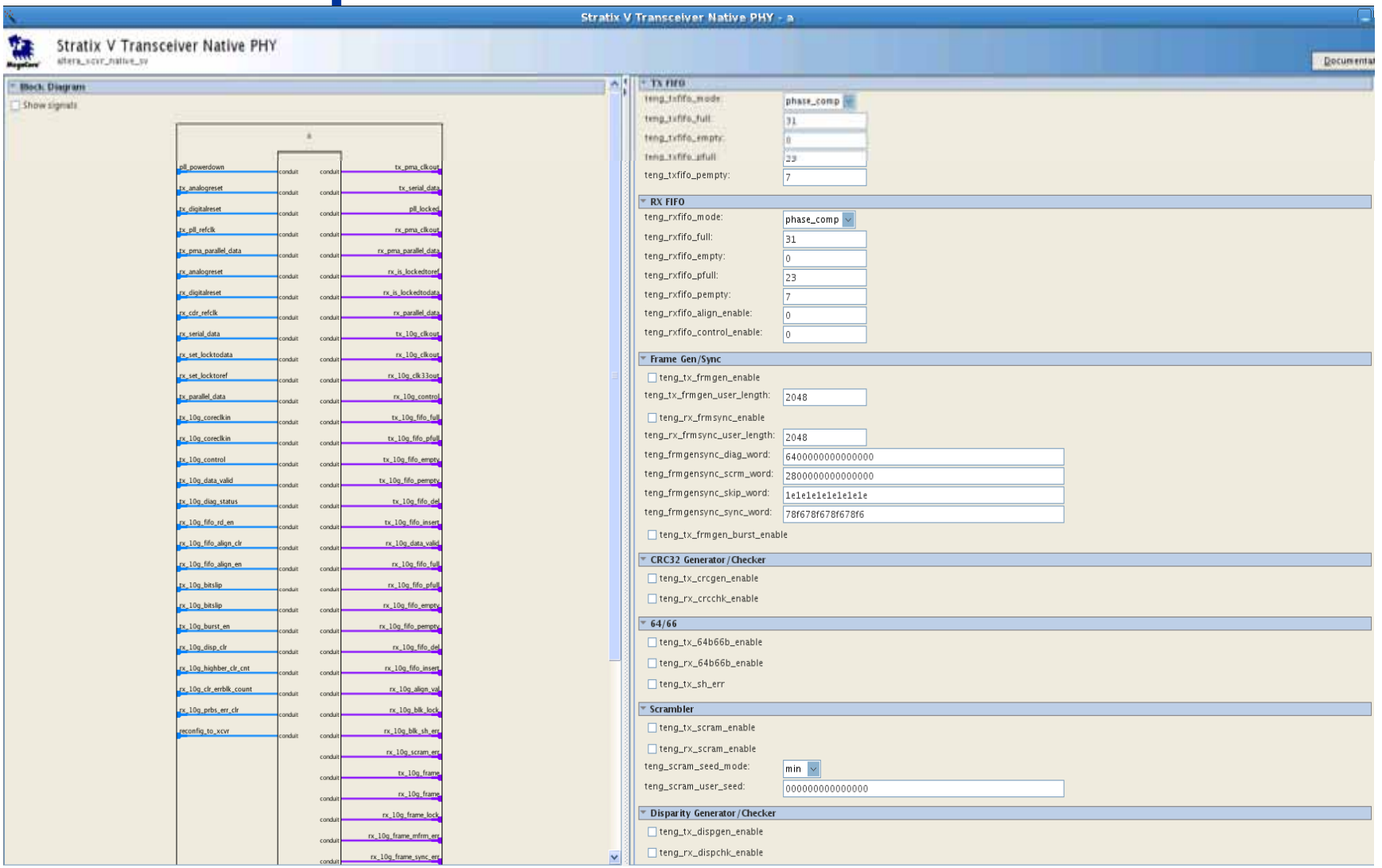
Standard PCS parameters



10G PCS – Parameter groups



10G PCS parameters



PHY IP Output Files for Compilation

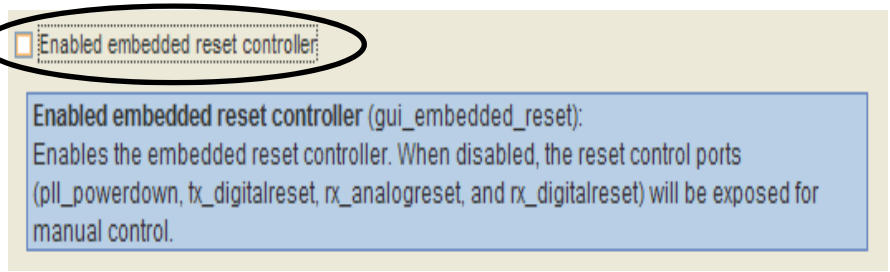
- <*phy_instance_name*>**.qip**
  - Script file that points to all files needed for synthesis
  - Add file to Quartus II project
- <*phy_instance_name*>**.v/.vhd**
  - Wrapper file that instantiates and configures the PHY IP core megafunction
- <*phy_instance_name*> folder
  - Combination of Verilog and SystemVerilog files representing PHY IP core components
- <*phy_instance_name*>**.ppf**
  - Stores top-level I/O and node information for importing into Pin Planner
  - Useful when pin layout must be assigned before top-level file is completed

Additional PHY IP Cores

- Transceiver Reconfiguration Controller
  - Discussed later
- Transceiver PHY Reset Controller

Transceiver PHY Reset Controller IP Core

- Fully customizable reset solution
  - Provides most flexible pre-built reset solution
  - Enable as many or as few control/status signals as you need
- Works with all non-protocol-specific PHY IP cores
  - Must disable embedded controller, if enabled by default

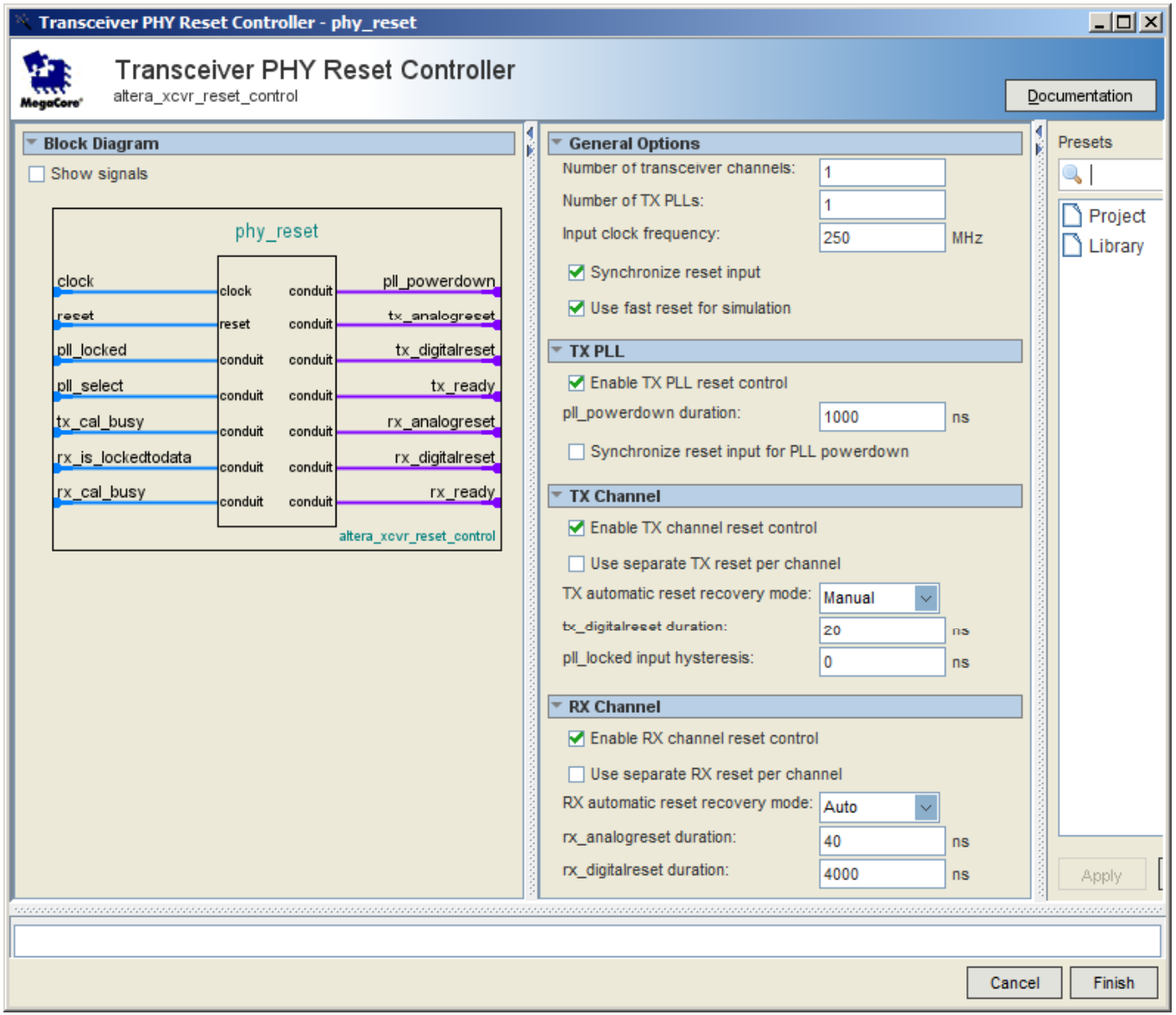


- Generates clear text Verilog file
  - User can modify as desired

Reset Controller Options

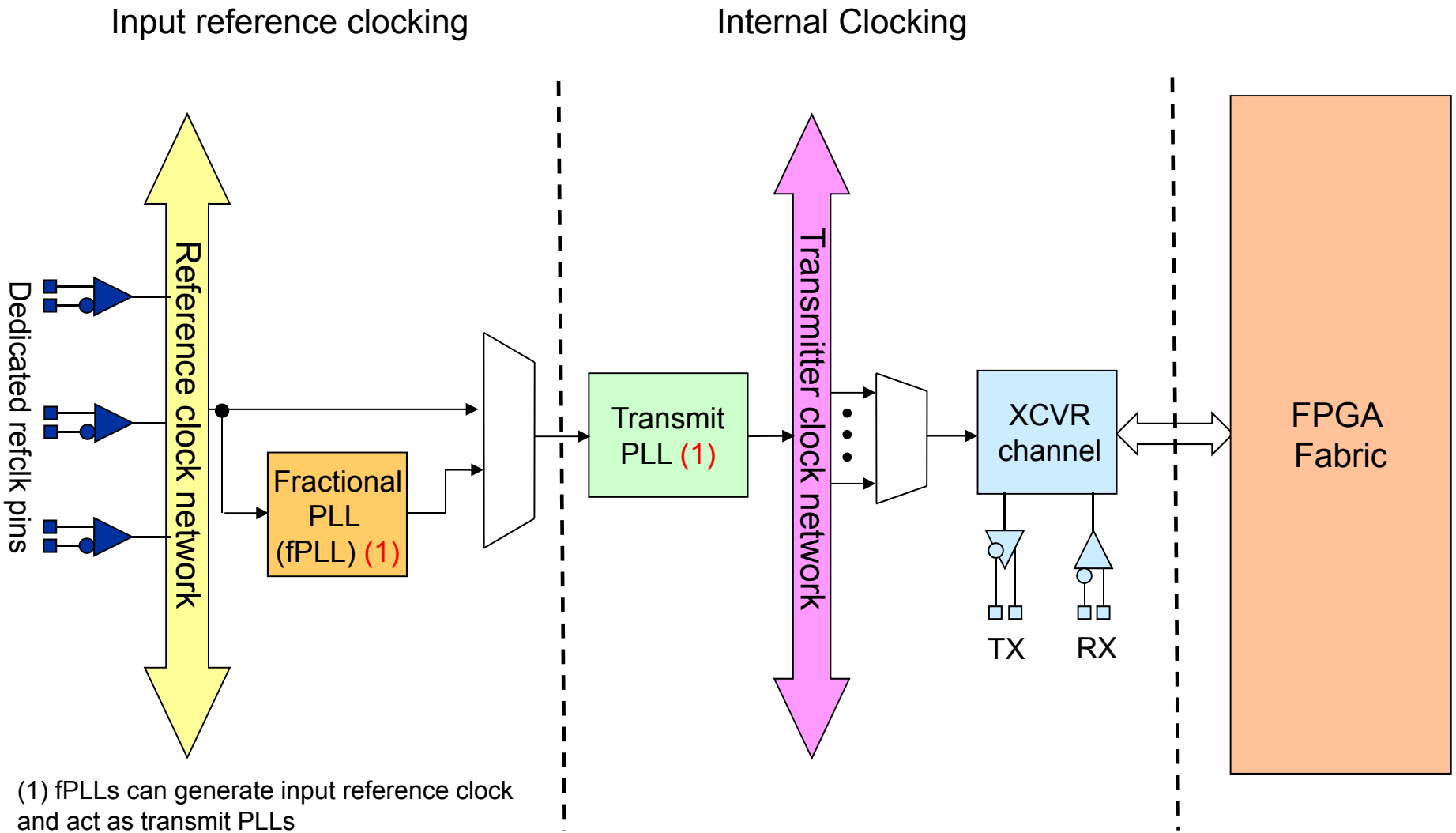
- Shared or individual reset controls per channel in transceiver instance
- Shared or separate reset controls per each RX channel
- Shared or separate reset controls per each TX channel
- Option for manual or automatic RX/TX reset recovery when PLLs lose lock
- Configurable timing delay

Reset Controller Parameter Editor



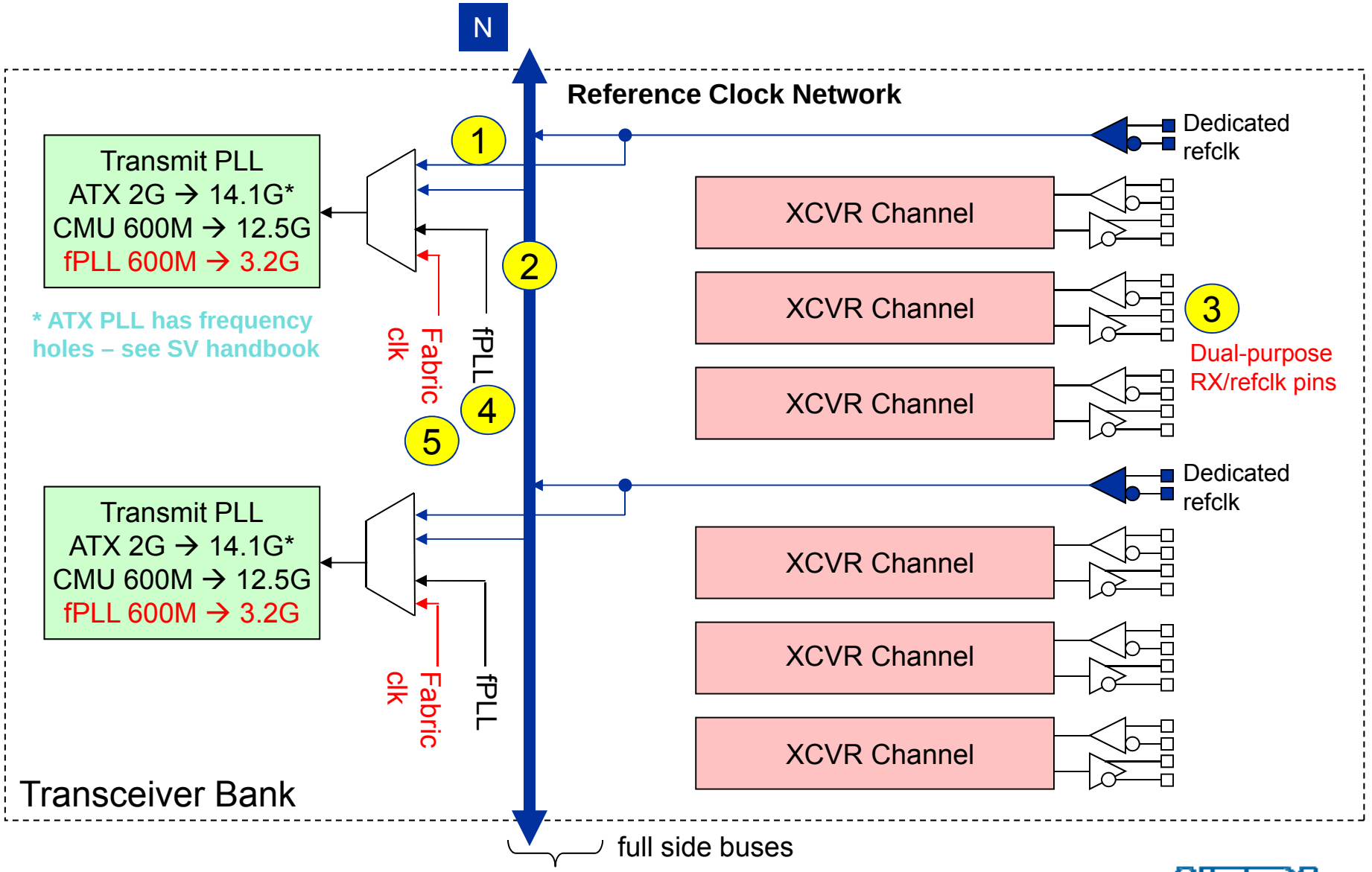
Transceiver Clocking

XCVR Clocking Overview

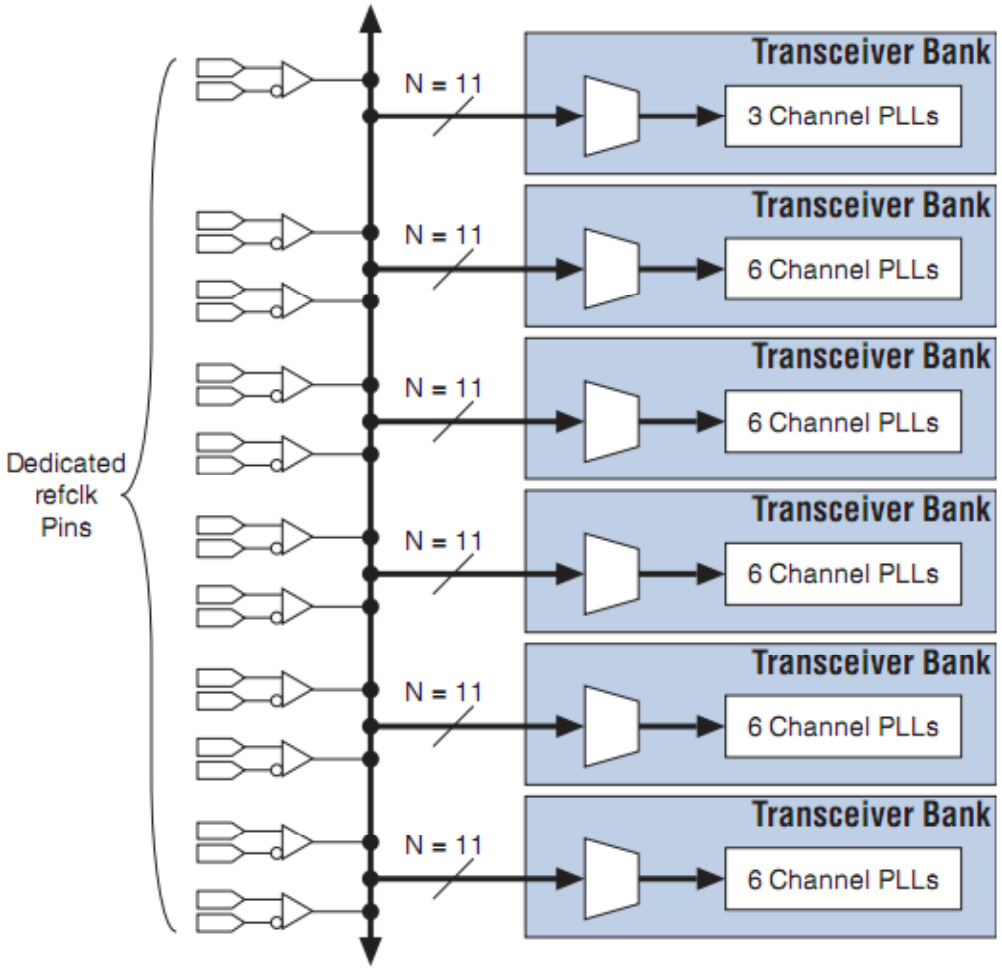


(1) fPLLs can generate input reference clock and act as transmit PLLs

Input Reference Clocking



Reference Clock Network

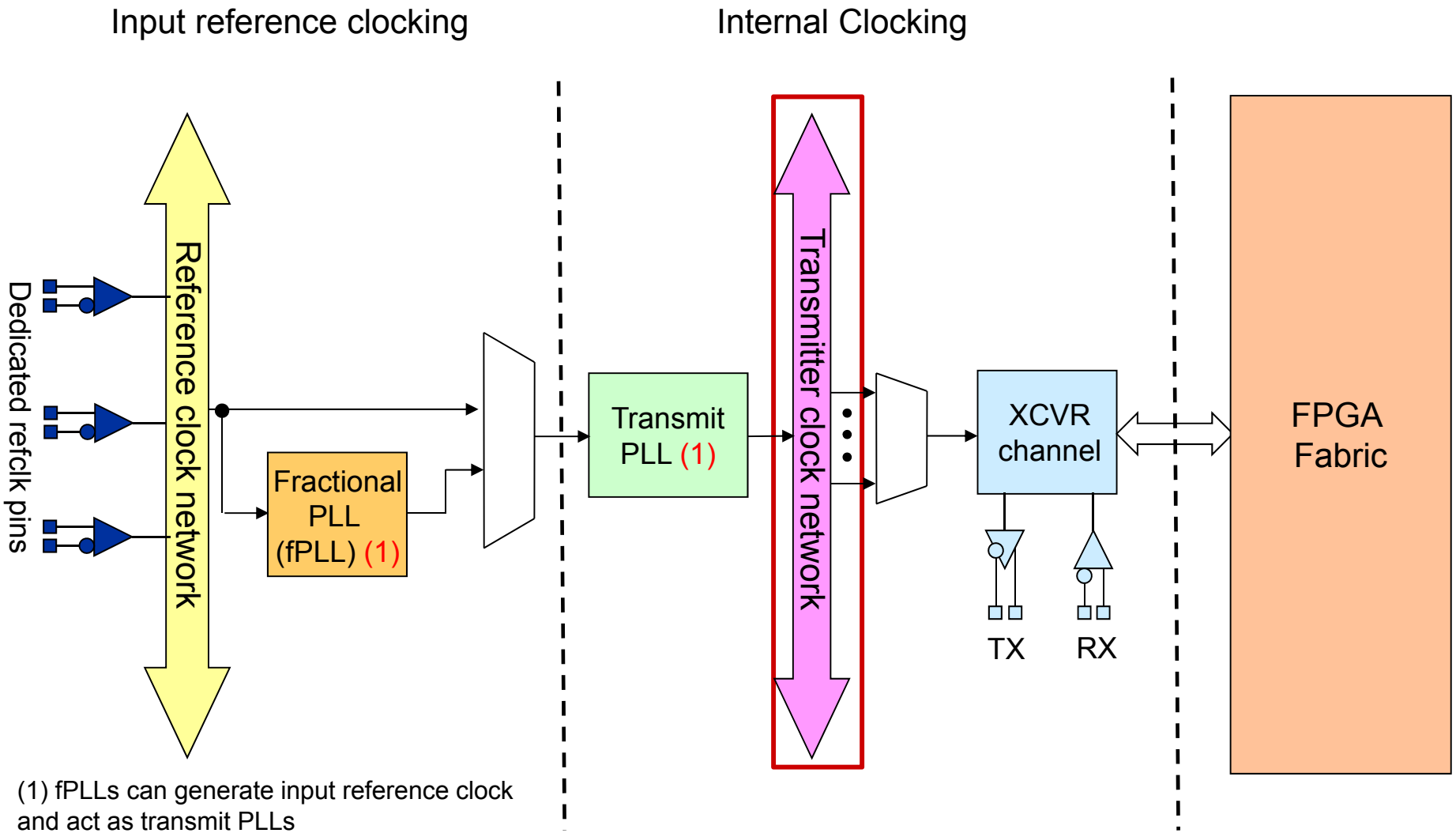


- Dedicated clock routing resources spanning the sides of the FPGA
  - Driven by reference clock pins
  - One clock route per reference clock pin
  - Exception: Arria V GT reference clock networks are segmented
- Allow reference clocks to be used by any PLL on that side of the FPGA

Input Reference Clocking

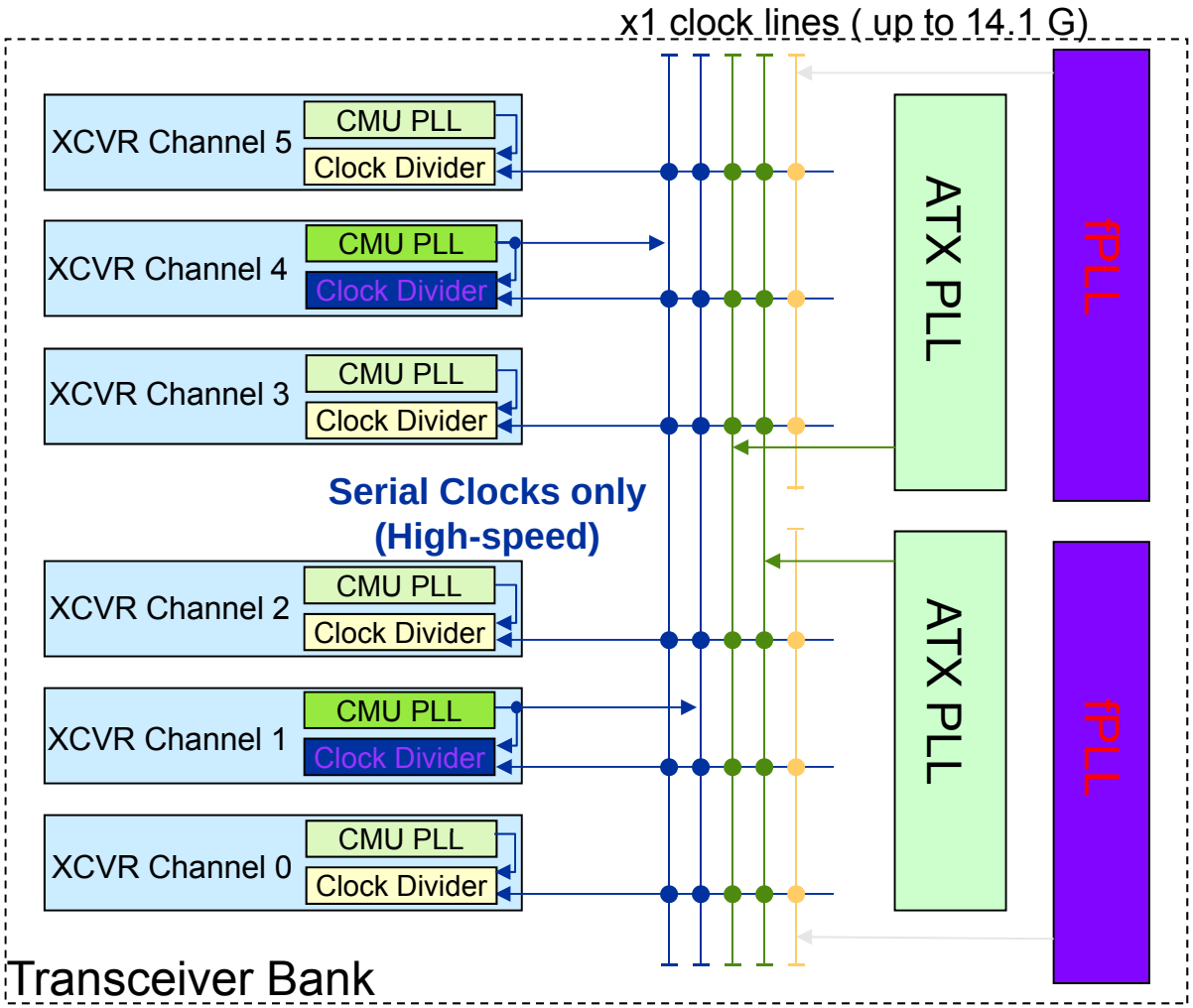
- Clock source in order of performance
  1. Dedicated refclk pin (direct path)
     - Ideally closest to the TX PLL
  2. Dedicated refclk pin using reference clock network
     - Within transceiver bank
     - Outside of transceiver bank
  3. Dual purpose RX/refclk pins
  4. fPLL
     - fPLL → TX PLL cascading can lead to high jitter
  5. Clock from FPGA Fabric

XCVR Clocking Overview



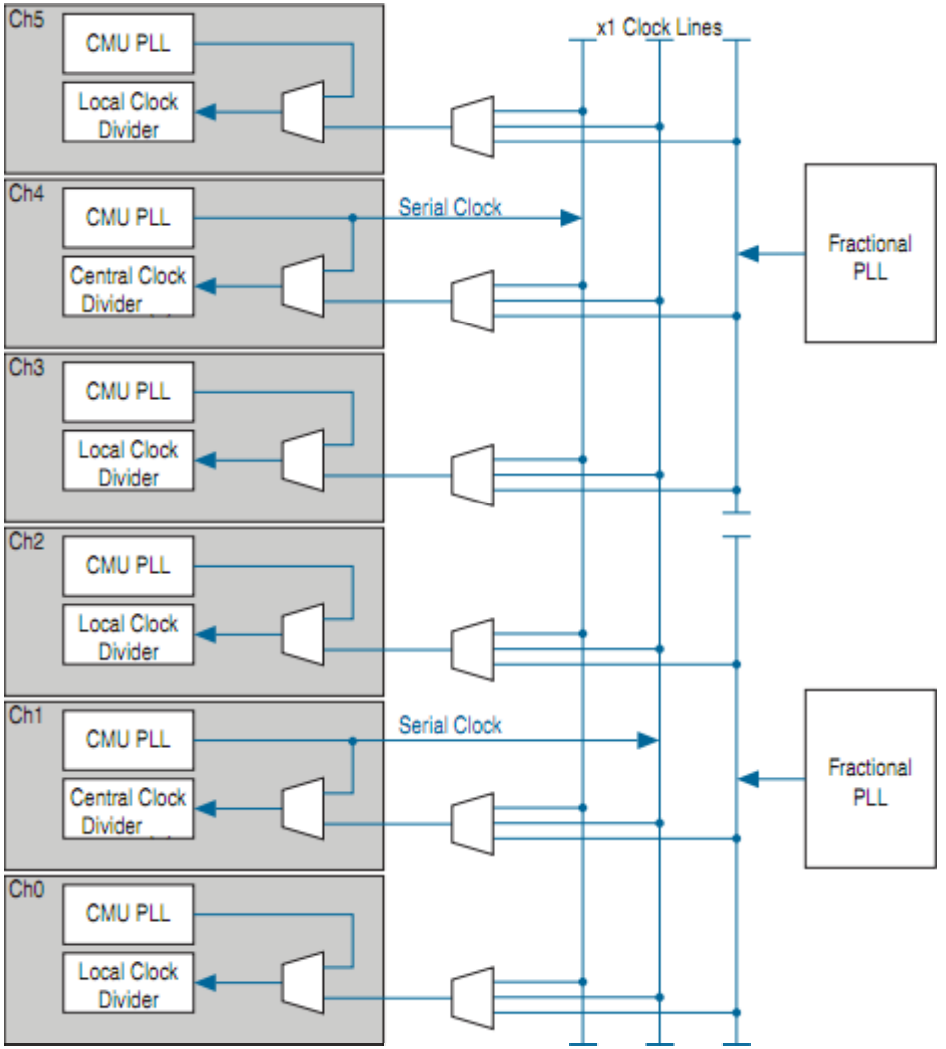
(1) fPLLs can generate input reference clock and act as transmit PLLs

Internal Clocking: Non-bonded Configurations

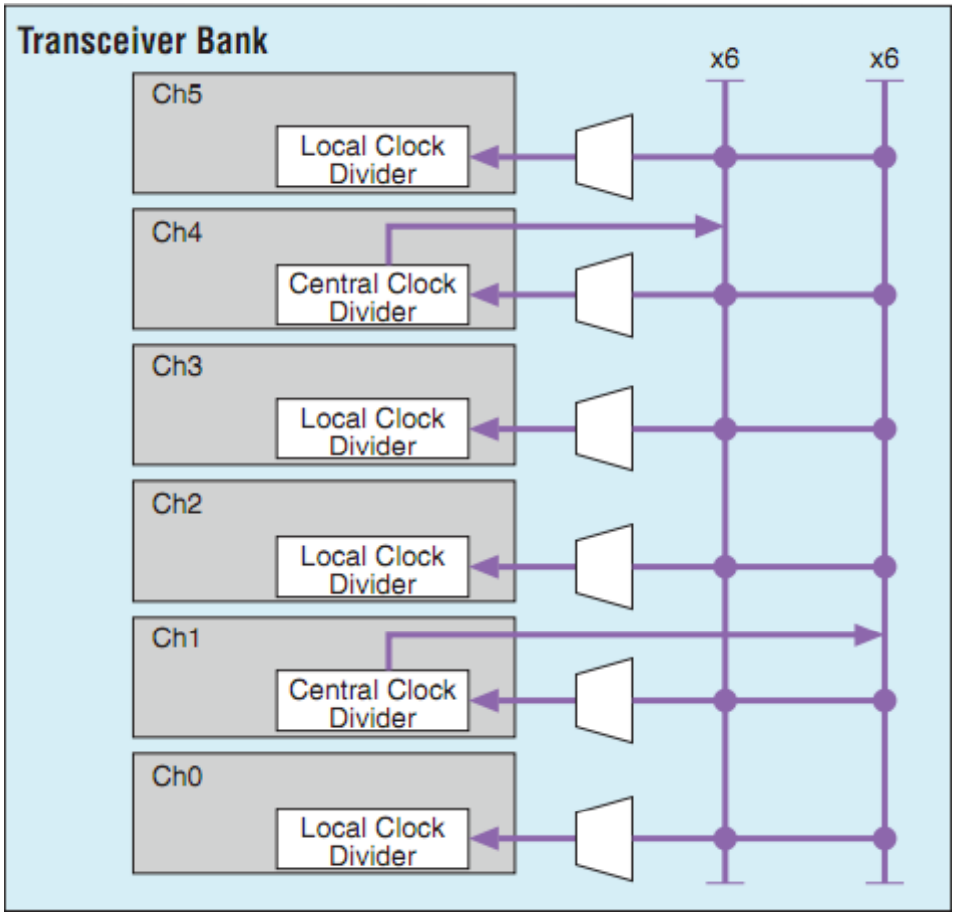


x1 Clock Network

- Span up to one 6-transceiver bank
- Used to carry high-speed clock to transceiver channels
  - Drives each channel's local clock divider
  - Non-bonded operation only
- Clock source
  - Ch1/Ch4 CMU PLL
  - Fractional PLL
  - ATX PLL



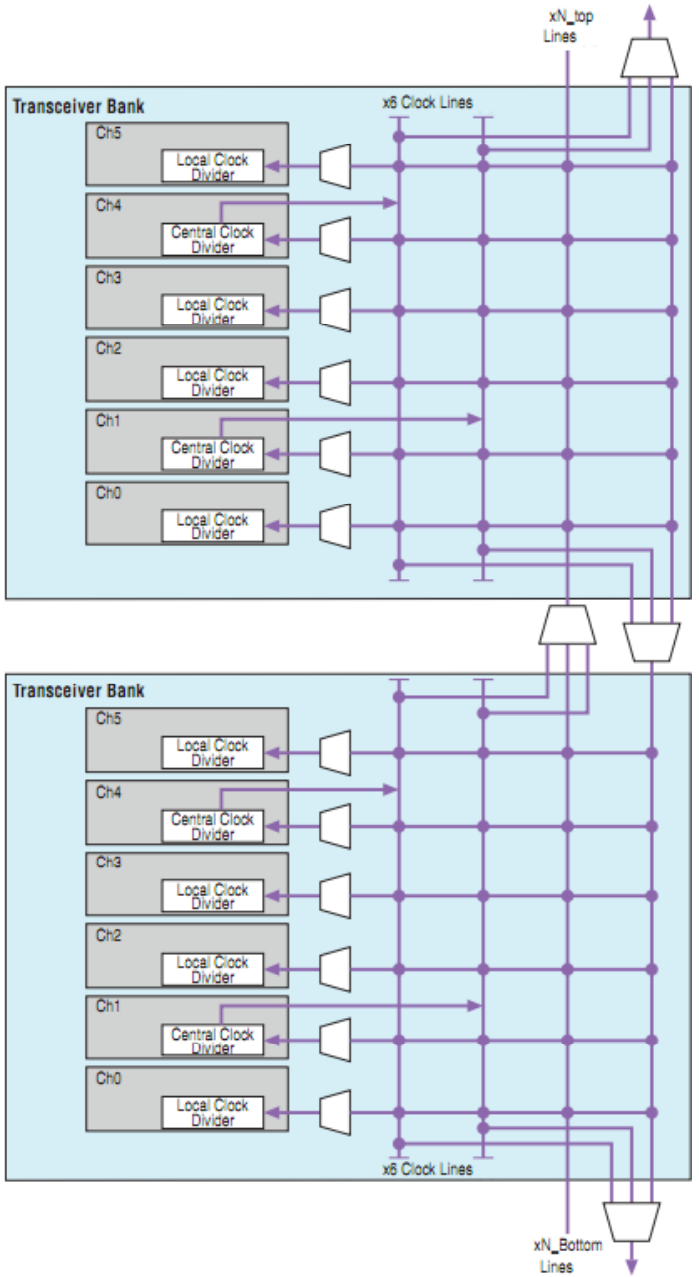
x6 Clock Network



- Span up to one 6-transceiver bank
- Used to carry both high-speed serial and low-speed parallel clocks to channels within a 6-transceiver bank
  - Bonded operation only
- Clock source
  - Central clock divider

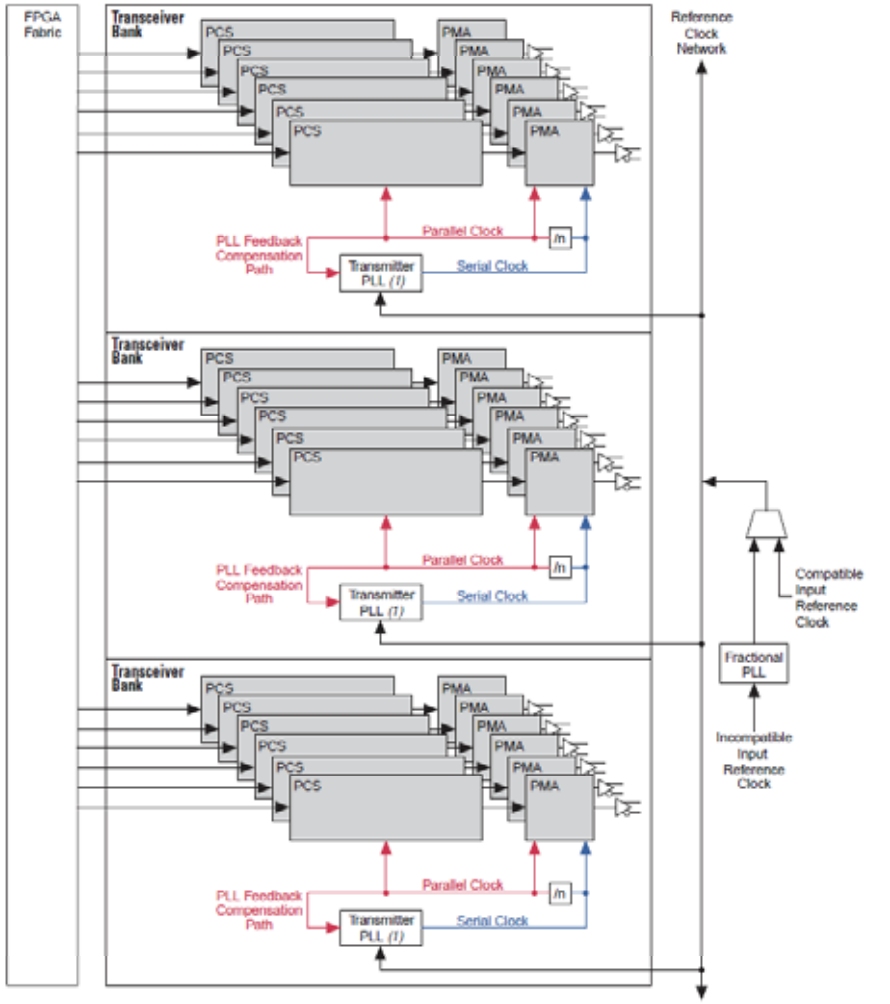
xN Clock Network

- Span the entire side of the device
- Used to carry both high-speed serial and low-speed parallel clocks from x6 line to other transceiver banks
  - Bonded operations only
- Clock source
  - x6 clock network (from central clock divider)



Internal Clocking: PLL Feedback Compensation Path Bonding

Figure 2–15. Three Transceiver Bank Channels Bonded Using the PLL Feedback Compensation Path

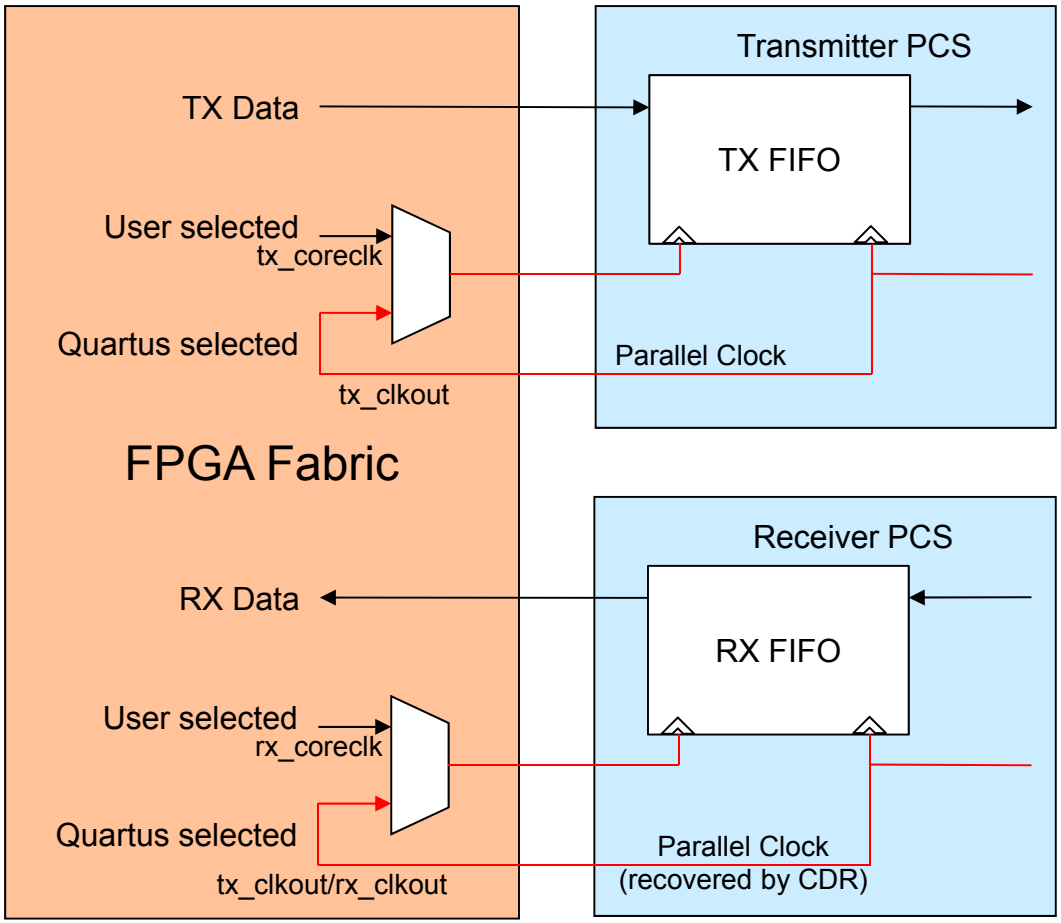


Note to Figure 2–15:

(1) The transmitter PLL can be an ATX PLL or a CMU PLL. You can have up to six channels per bank with an ATX PLL and five channels per bank with a CMU PLL.

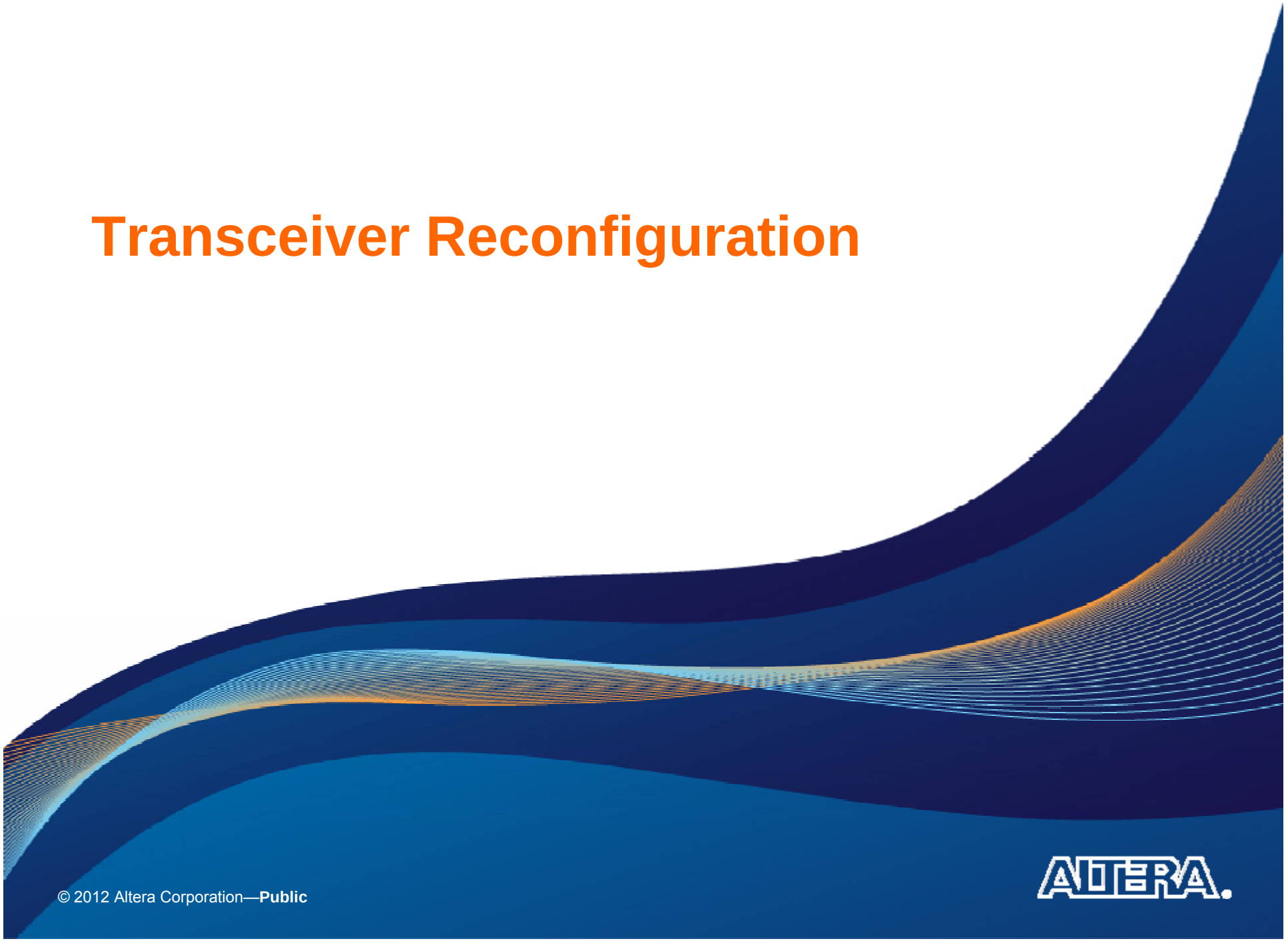
- For bonding channels beyond a transceiver bank with the controlled skew
- Removes divider uncertainty by aligning parallel clk with refclk
- Refclk restriction
  - refclk frequency has to be the same as tx_clkout
  - fPLL can be used to match refclk and parallel clock

FPGA Fabric-Transceiver Interface Clocking



- Quartus selected option is best for:
  - Bonded configurations
- User selected option is best for:
  - Saving FPGA fabric clock resources for multiple identical channels

Transceiver Reconfiguration



Introduction

What is transceiver reconfiguration?

- Reconfiguration of single or multiple transceiver channel settings during device operation
- Reconfiguration of
  - Physical media attachment (PMA) settings
  - Physical coding sublayer (PCS) settings
  - Transceiver Clocking (PLL settings)
- Run time modification does not interrupt operation of adjacent transceiver channel(s)

Transceiver Reconfiguration Uses

- Adjust transmitter/receiver buffer settings while bringing up link to fine-tune signal integrity
  - Increases flexibility in board/system design
- Increase/decrease data rate due to downstream/upstream device
- Support newer, changing serial protocols
- Add design flexibility by supporting multiple protocols with same hardware

Reconfguration and Device Support

Reconfiguration Feature		Device Family		
		Stratix V	Arria V	Cyclone V
Calibration		✓	✓	✓
Analog	PMA Reconfiguration	✓	✓	✓
	EyeQ, AEQ, DFE	✓		
Loopback	Pre-CDR reverse serial	✓	✓	✓
	Post-CDR reverse serial	✓	✓	✓
PLL Reconfiguration	Ref Clk Switching	✓	✓*	✓*
	TX PLL Reconfiguration	✓	✓	✓*
Channel Reconfiguration	RX CDR Reconfiguration	✓	✓	✓*
	PCS Reconfiguration	✓	✓	✓*
	TX PLL Switching	✓	✓	✓*
	TX Channel Divider	✓	✓	✓*
	FPGA/Transceiver Data Width	✓	✓	✓*

**** Support enabled in a future version of the Quartus II software.***

Transceiver Reconfiguration Controller

- Provides simple way to change transceiver settings dynamically
  - All types of reconfiguration require using controller
  - Users must design custom hardware or software to interact with controller based supported reconfiguration types
- Uses general FPGA resources (soft IP)
- Connects to transceiver megafunctions/IP cores using dedicated interface
  - Some PHY IP blocks implemented with embedded controller core
- Required in all PHY IP core designs, even if no intention of changing transceiver settings

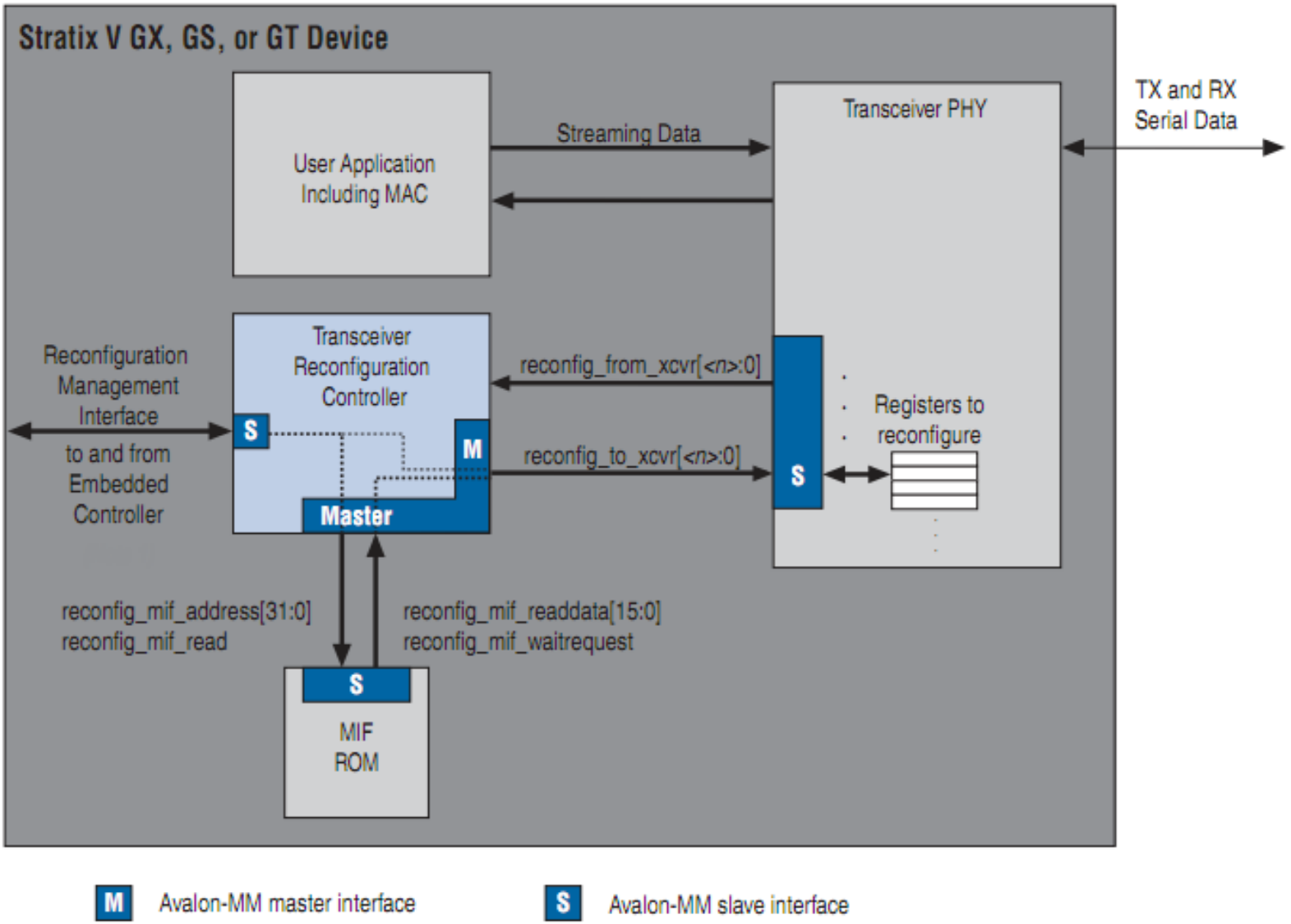
Reconfiguration Modes

- Register-based
  - Reconfiguration initiated by read and write operations to controller registers to reconfigure individual transceiver settings
  - Controller translates operations to specific transceiver registers
- Streamer-based (MIF Mode)
  - Transceiver configuration data stored in ROM/RAM using a memory initialization file (MIF)
  - Reconfiguration initiated by read and write operations to controller registers
  - Upon initialization, controller steams MIF configuration data into transceiver registers to update transceiver settings all in one step
- Streamer-based (Direct Write Mode)
  - Reconfiguration initiated by read and write operations to controller registers
  - the user individually writes MIF words into individual transceiver registers

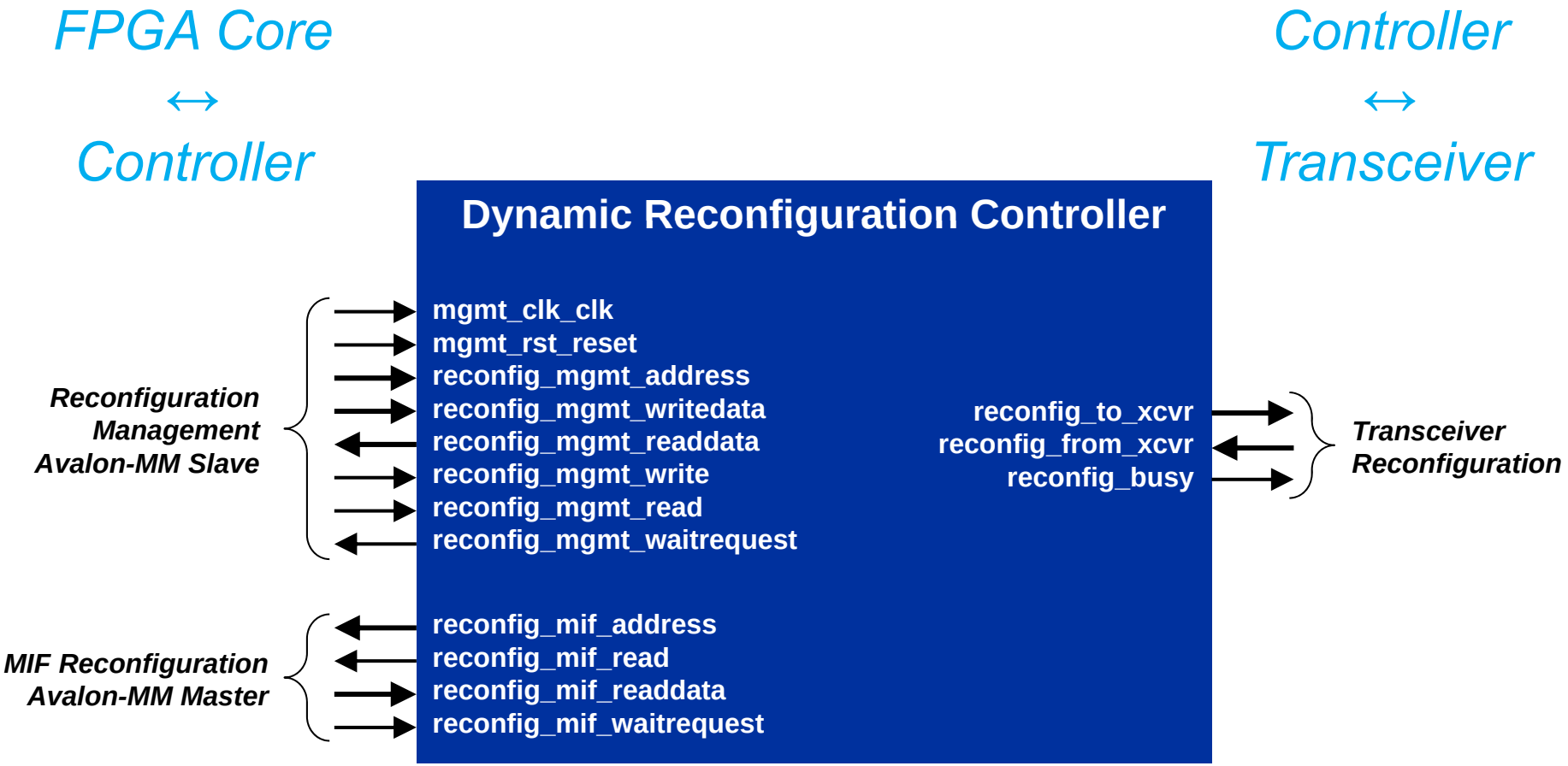
Reconfiguration Features vs. Modes

Feature	Register-Based	Streamer-Based
PMA	✓	✓
Loopback	✓	
EyeQ	✓	
AEQ	✓	
DFE	✓	
ATX Tuning (Calibration)	✓	
Reference Clock Switch	✓	✓
PLL Reconfiguration	✓	✓
Channel Reconfiguration		✓

Reconfiguration Diagram



Reconfiguration Controller Interfaces



Reconfiguration Management Interface

- Interfaces between reconfiguration control logic and controller
  - User logic employs read and write transfers using Avalon-MM master to setup, start and monitor reconfiguration
  - Example Avalon-MM masters: embedded processor, state machine, JTAG Avalon Master component
- Interfaces
  - *mgmt_clk_clk*
    - Provides a clock for the reconfiguration interface
    - Supported frequency range: 100 – 125 MHz
  - *mgmt_rst_reset*
    - Resets the controller
  - *reconfig_mgmt_**
    - Avalon-MM slave interface made up of 7-bit address, 32-bit data and read/write enable signals

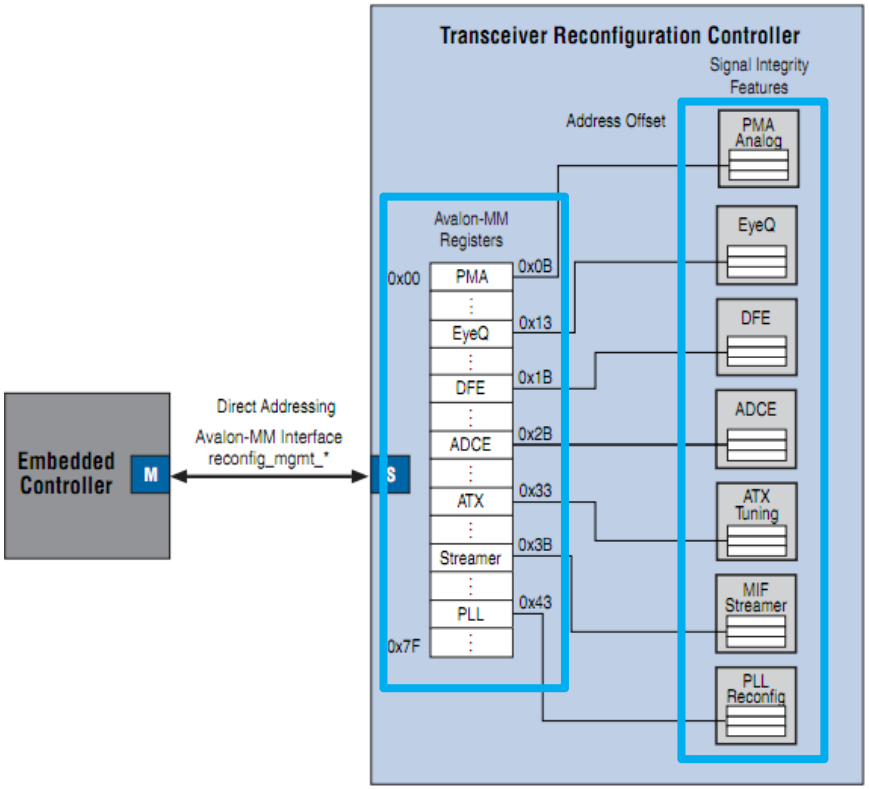
MIF Reconfiguration Interface

- Avalon-MM master interface between controller and MIF storage location
  - Controller accesses MIF data based on requests received through its slave interface
- Interface
  - *reconfig_mif_**
    - Master interface made up of 32-bit address, 16-bit data, read enable and waitrequest signals

Transceiver Reconfiguration Interface

- Dedicated interface between controller and transceiver IP core
  - Controller reads values from and writes values to transceiver registers based on requests received through its slave interface
- Each transceiver instance has 1 reconfiguration interface per each duplex transceiver channel and per each TX PLL
- Signals
  - *reconfig_to_xcvr[(n*70)-1..0]*
    - Output signal from controller to transceiver IP core instance(s)
    - n = number of reconfiguration interfaces
  - *reconfig_from_xcvr[(n*46)-1..0]*
    - Input signal to controller from transceiver IP core instance(s)
    - n = number of reconfiguration interfaces
  - *reconfig_busy*
    - Output signal that indicates when a reconfiguration operation is in progress
    - Similar to the busy bits (bit 8) in the control and status registers

Reconfiguration Controller Address Map

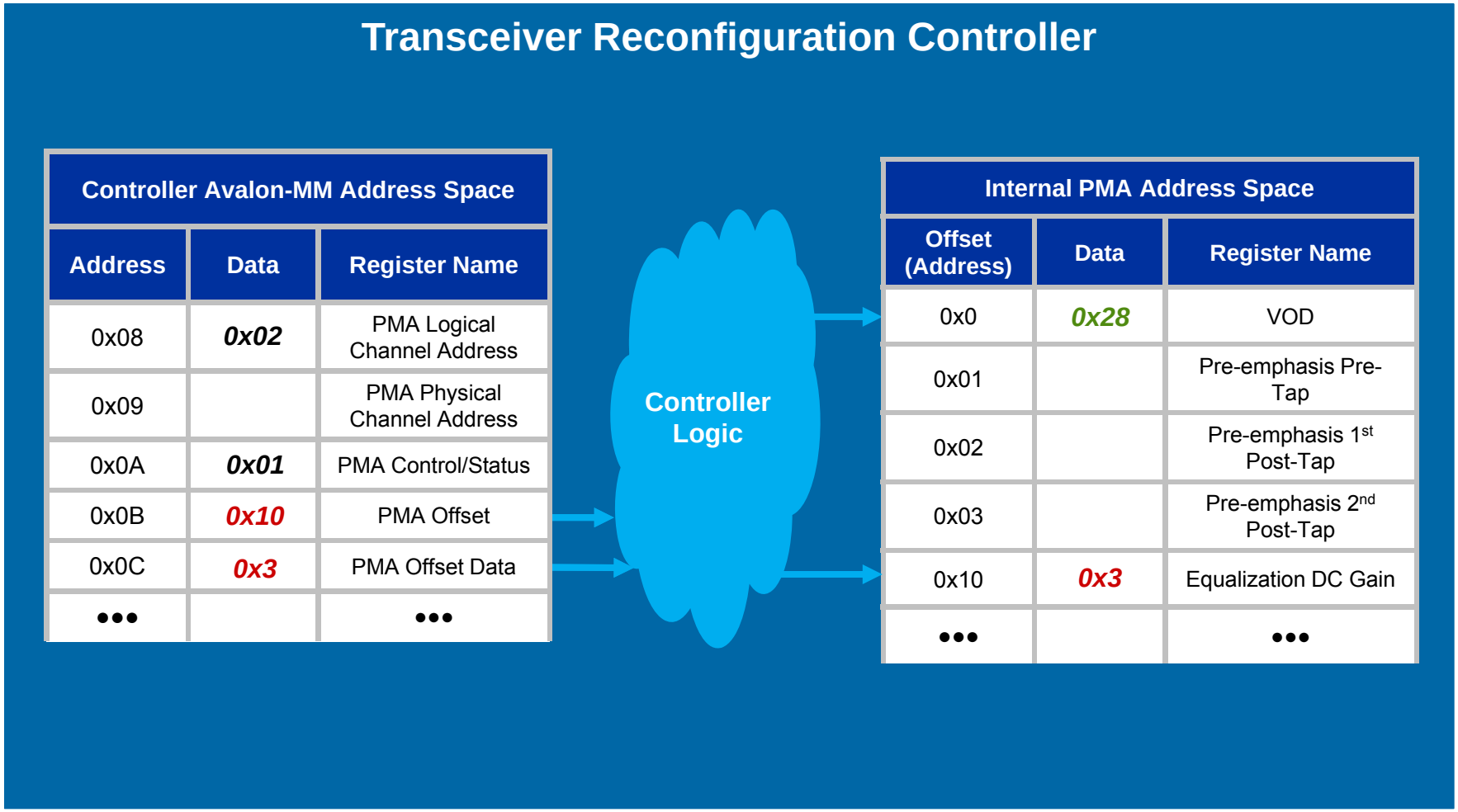


Reconfiguration Feature	7-bit Address Range	
	Start	End
PMA	0x08	0x0C
EyeQ	0x10	0x14
DFE	0x18	0x1C
AEQ	0x28	0x2C
ATX PLL Calibration	0x30	0x34
Streamer-Based/ Direct Write-Based	0x38	0x3C
PLL Reconfiguration	0x40	0x44

- Features assigned defined address ranges in reconfiguration controller's Avalon-MM address space
- User logic sets up/activates a feature in controller by accessing registers in feature's address range
- Within controller, features have own internal address spaces for accessing specific feature settings
- User logic programs feature setting by writing a value into an offset register (e.g. PMA=0x0B, EyeQ=0x13) . The offset value corresponds to the internal address of that feature setting.
  - Reconfiguration controller uses offset value to access the correct internal address for that feature

** Any undefined register addresses are reserved.*

Controller Offset Example



Logical Channel Numbers

- Reference numbers used to represent actual physical transceiver channels and TX PLLs
- Used during reconfiguration to target specific channels or PLLs
  - Controller translates read/write operations to logical channel numbers to their corresponding physical channels
- Assigned to physical transceivers automatically based on
  - Number of channels and TX PLLs in transceiver IP cores
  - Order in which the reconfiguration interfaces are physically connected to controller
  - View the assignment results in the Transceiver Reconfiguration Report (Compilation Report → Fitter → GXB Reports)

Basic Register-Based Write Operation

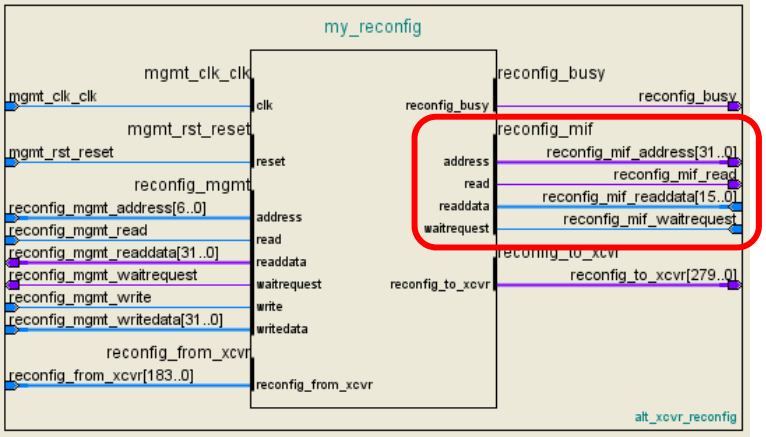
1. Read feature *control and status* register to determine *busy* bit is 0
2. Write target logical channel number to feature's *logical_channel_address* register
3. Write the internal address (offset) value to feature's *offset* register
4. Write the target value for the feature setting into the feature *data* register
5. Write feature's *control and status* register *write* bit with 1
6. Transceiver is programmed when *busy* bit is 0 again

Basic Register-Based Read Operation

1. Read feature *control and status* register to determine *busy* bit is 0
2. Write target logical channel number to feature's *logical_channel_address* register
3. Write the internal address (offset) value to feature's *offset* register
4. Write feature's *control and status* register *read* bit with 1
5. Read operation is complete when *busy* bit is 0 again
6. Read the feature's *data* register for the value of the feature setting returned by the read operation

Reconfiguration Modes: Streamer Based

- Two available modes:
  - Mode 0: MIF Streaming
  - Mode 1: Direct Writes



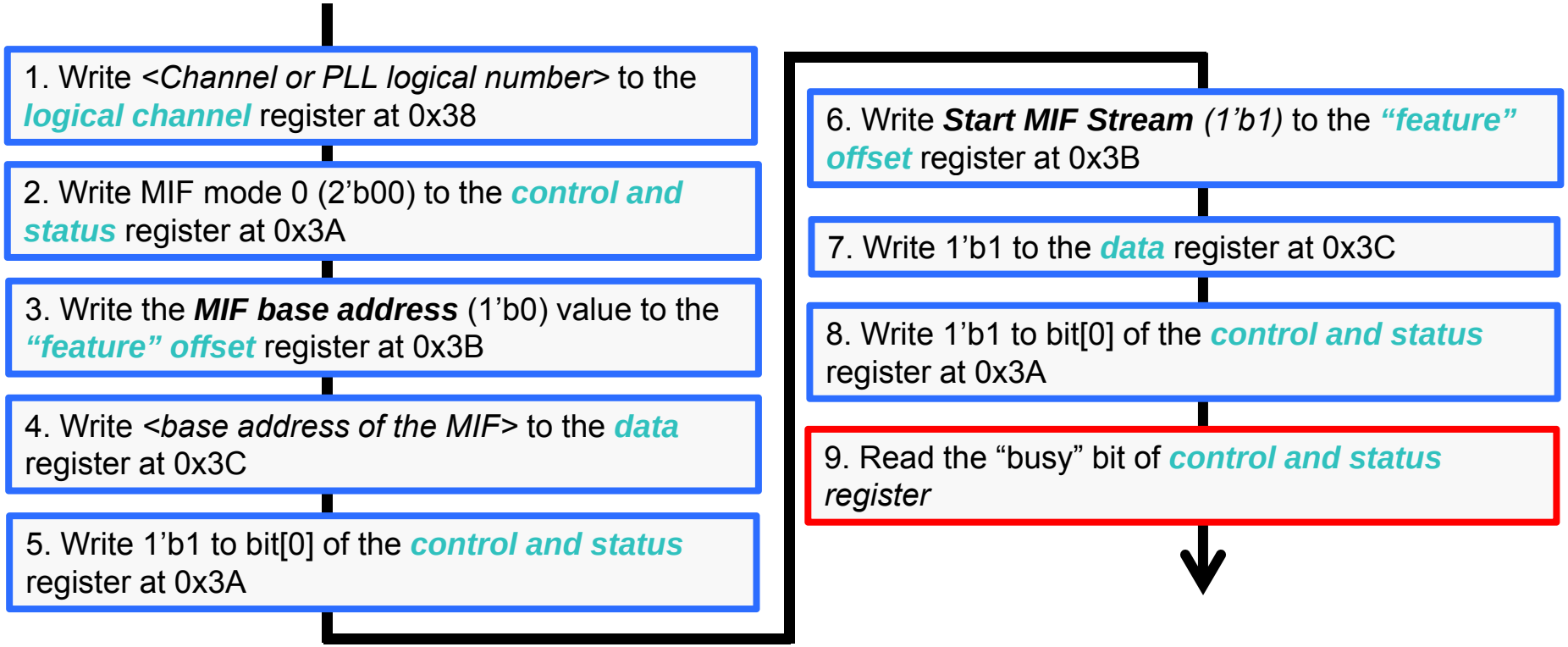
- Reconfiguration data is contained in a MIF
- Supports:
  - PLL counters
  - Reference clocks
  - Local clock dividers

… basically you get just about everything not available through register based

Streamer Based - Modes

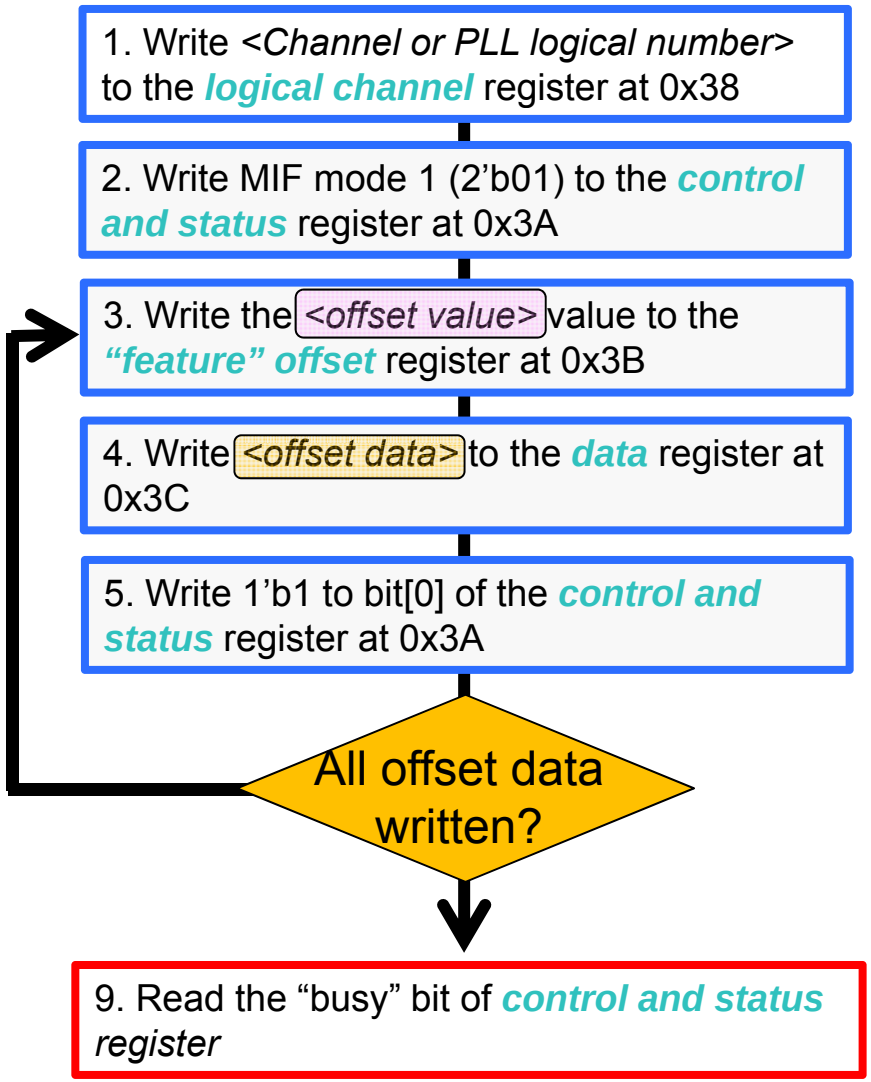
- Both modes uses the Streamer module in the Reconfiguration controller
  - Unlike dedicated feature blocks (PMA, Reference Clock, etc) the Streamer module uses the same address to carry out reconfiguration
    - Data values are different though
- Mode 0: MIF Streaming
  - Streams the entire content of a MIF
  - No nonsense reconfiguration
  - “One command” and reconfiguration is done
- Mode 1: Direct Writes
  - No MIF streaming
  - Selectively write reconfiguration data
  - May require multiple writes / reads

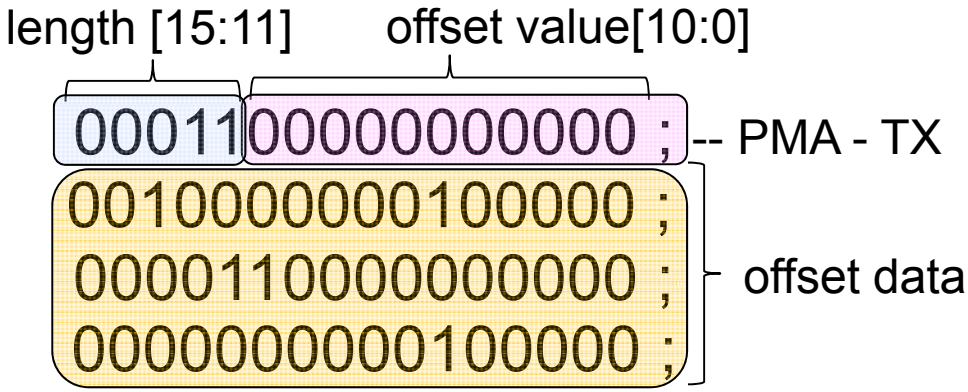
Modes 0 MIF Streamer - Flow



- Steps similar to register based reconfiguration
  - Steps { 3 , 4 , 5 } and { 6 , 7 , 8 } use the same sequence just writing different values

Modes 1 Direct Writes - Flow



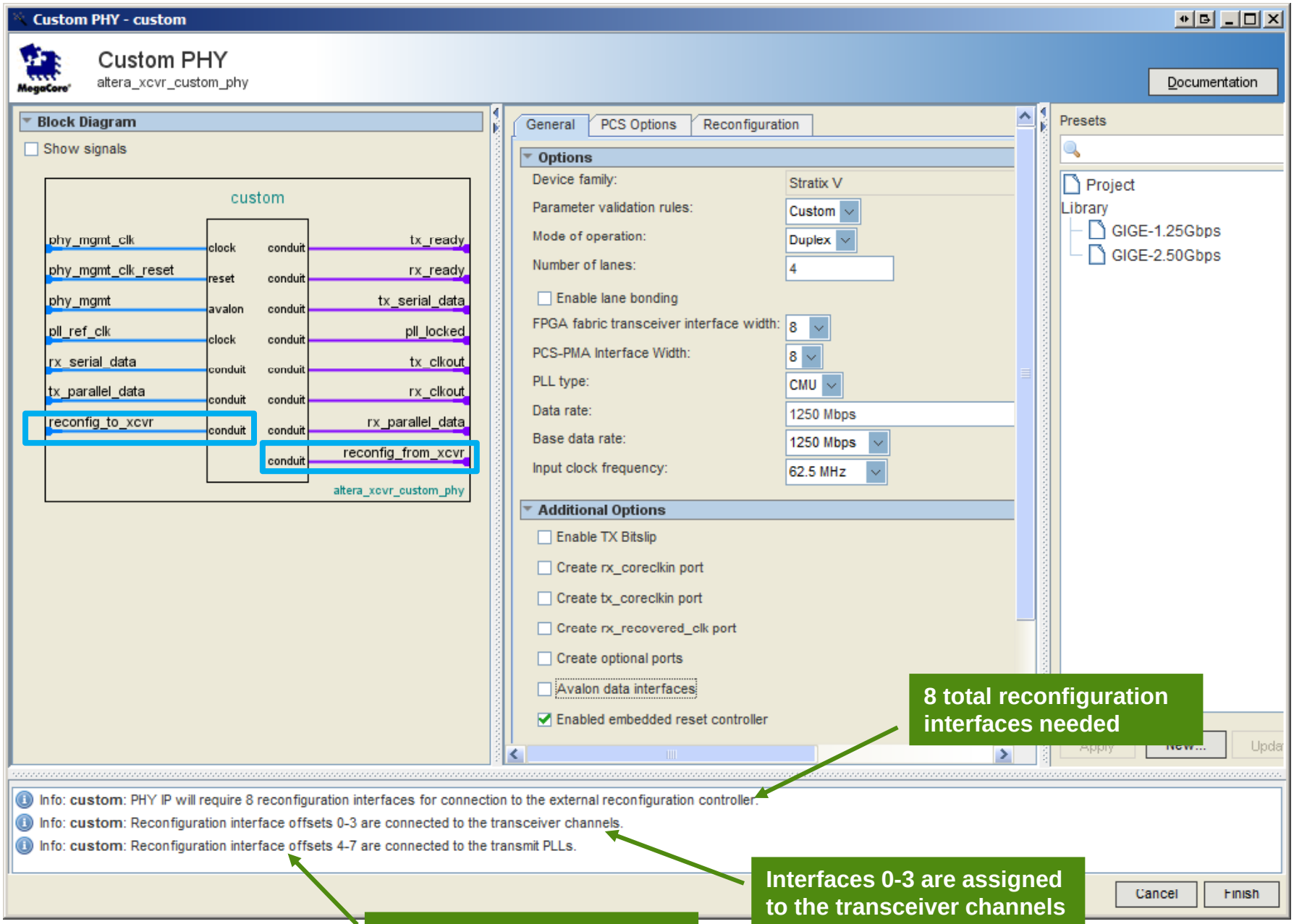


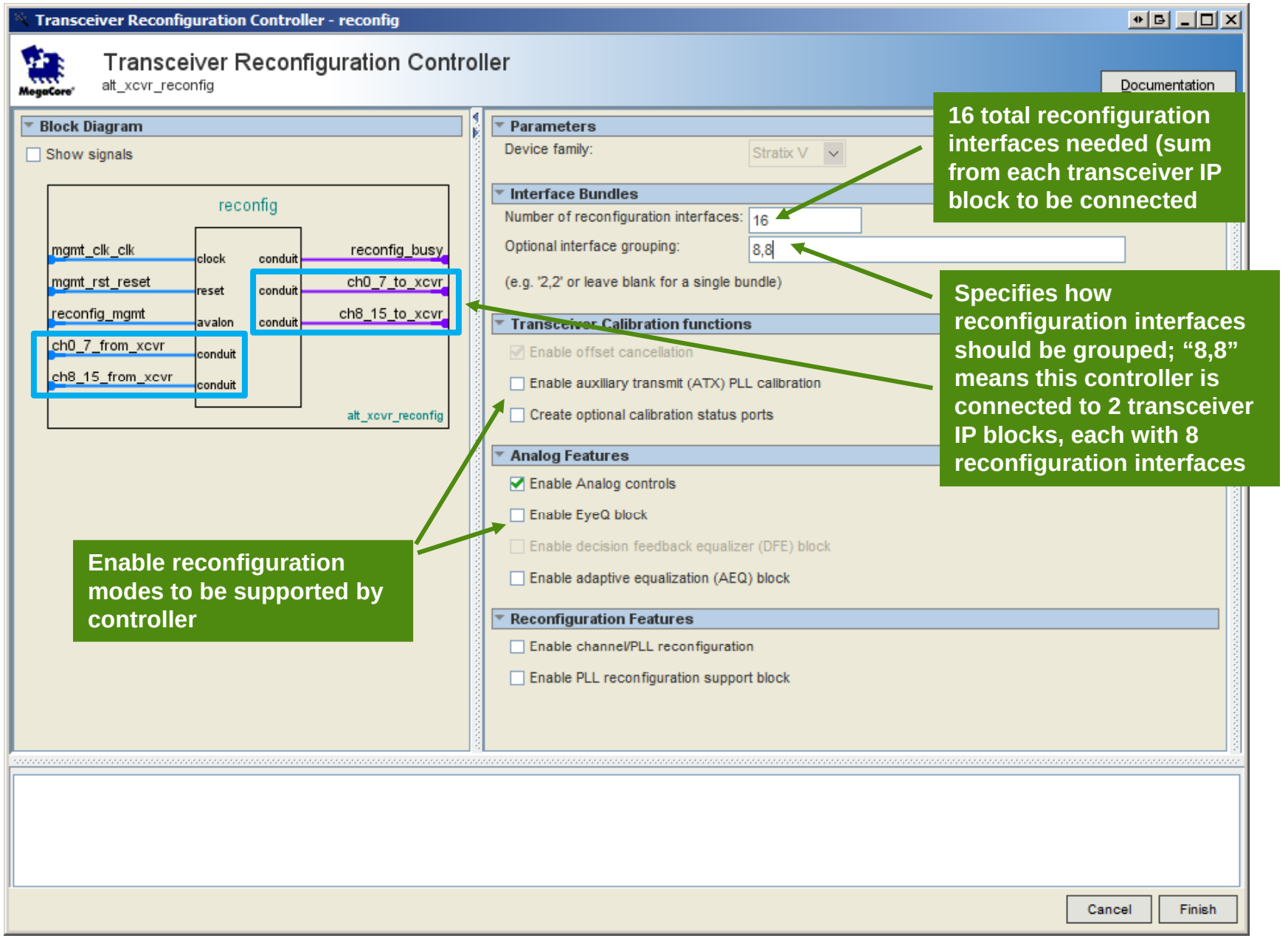
- The length field determines the number of loops
- Must increment the offset value by 0x1 after each loop

Enabling Transceiver Reconfiguration

- User must configure
  1. IP megafunction containing embedded transceivers
  2. Transceiver Reconfiguration Controller megafunction

- Purpose
  - Enable reconfiguration options
  - Ensure controller has the correct number of reconfiguration interfaces





Reconfiguration Interface Merging

- All transceiver channels and TX PLLs generate an interface on the controller and transceiver megafunction
- Quartus II software automatically merges interfaces as transceiver functionality is merged
  - e.g. Shared TX PLLs
- Starting with separate interfaces gives Fitter more flexibility in placement as it can merge any interfaces as needed

Controller Design Example Scenarios

1. Connecting to 1 PHY IP core with bonded channels
2. Connecting to 1 PHY IP core with non-bonded channels
3. Connect to 2 PHY IP cores

One PHY IP Core with Bonded Channels

- One 4-channel Custom PHY IP core instance
- One transceiver reconfiguration controller

Example 1 Setup

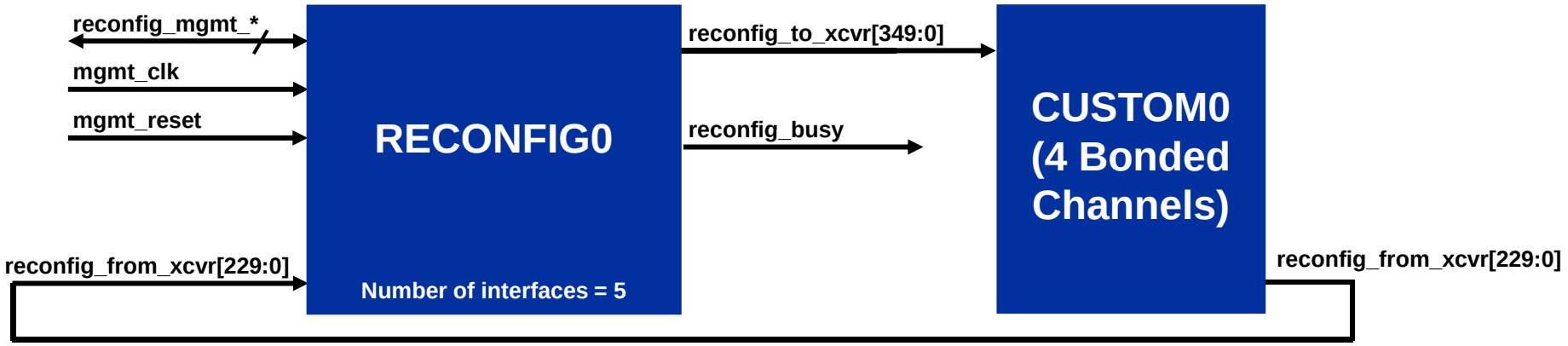
PHY IP Instance (CUSTOM0)

1. Configure Custom PHY IP for 4 bonded, full-duplex channels
2. Enable reconfiguration features (if needed)
3. Note the number of reconfiguration interfaces in Messages window
   - 5 interfaces total
   - 0 – 3 : Transceiver channels
   - 4 : TX PLLs

Reconfiguration Controller Instance (RECONFIG0)

1. Enable reconfiguration features
2. Set number of reconfiguration interfaces to 5

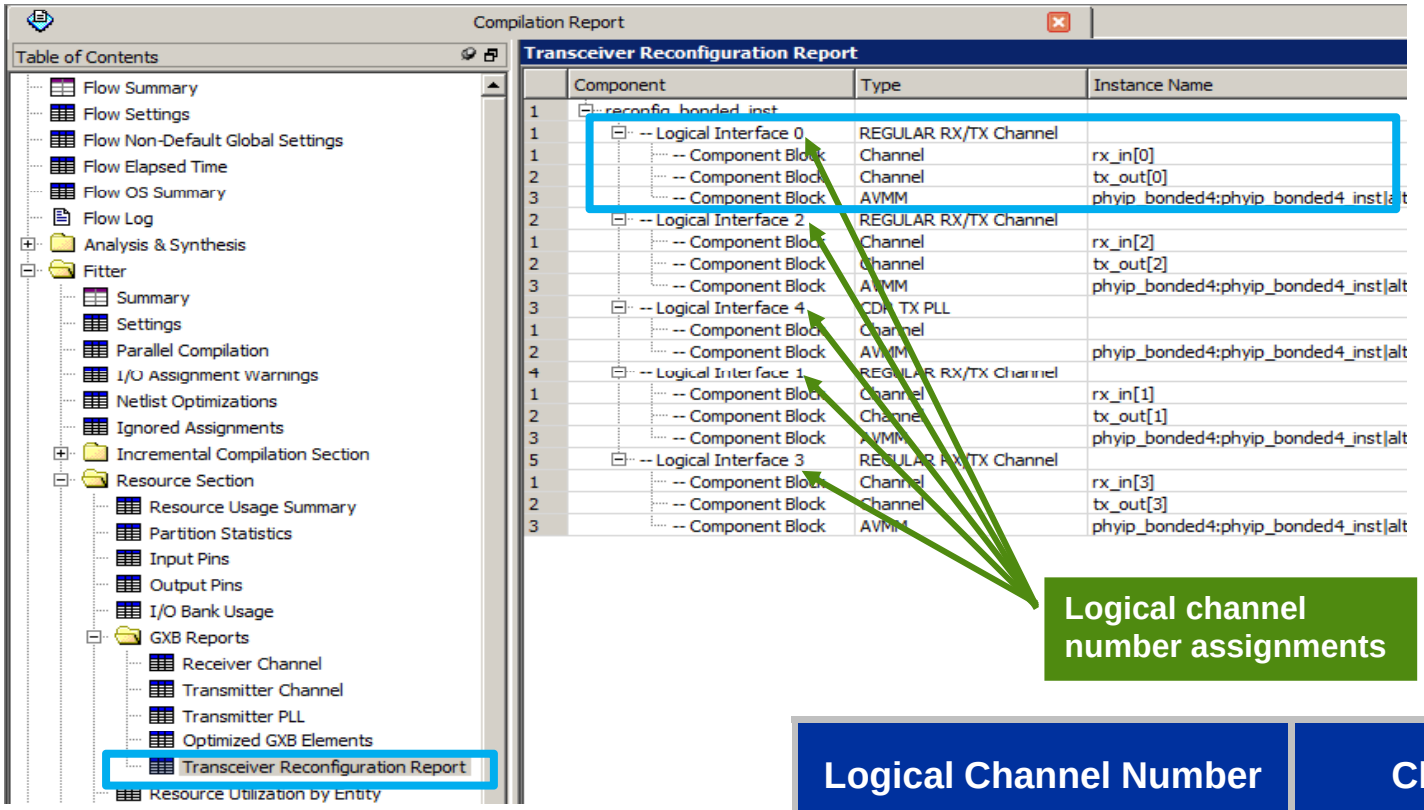
Example 1 Block Diagram w/ Connections



CUSTOM0.reconfig_from_xcvr[229:0] ⇒ RECONFIG0.reconfig_from_xcvr[229:0]

RECONFIG0.reconfig_to_xcvr[349:0] ⇒ CUSTOM0.reconfig_to_xcvr[349:0]

Transceiver Reconfiguration Report



Logical Channel Number	Channel Name
0	Channel 0
1	Channel 1
2	Channel 2
3	Channel 3
4	TX (CMU) PLL

One PHY IP Core with Non-Bonded Channels

- One 4-channel Custom PHY IP core instance
- One transceiver reconfiguration controller

Example 2 Setup

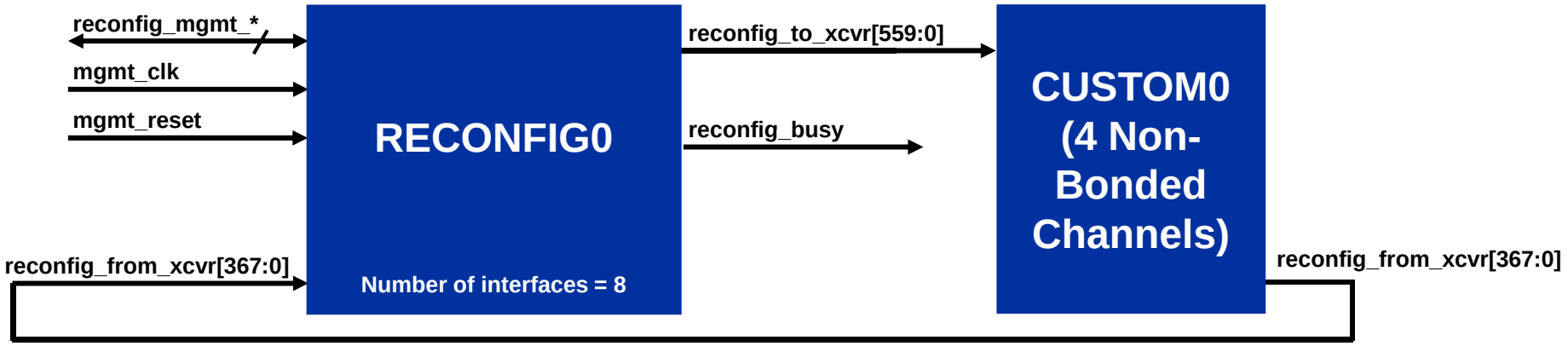
PHY IP Instance (CUSTOM0)

1. Configure Custom PHY IP for 4 non-bonded, full-duplex channels
2. Enable reconfiguration features (if needed)
3. Note the number of reconfiguration interfaces in Messages window
   - 8 interfaces total
   - 0 – 3 : Transceiver channels
   - 4 – 7 : TX PLLs

Reconfiguration Controller Instance (RECONFIG0)

1. Configure transceiver reconfiguration controller
2. Enable reconfiguration features
3. Set number of reconfiguration interfaces to 8

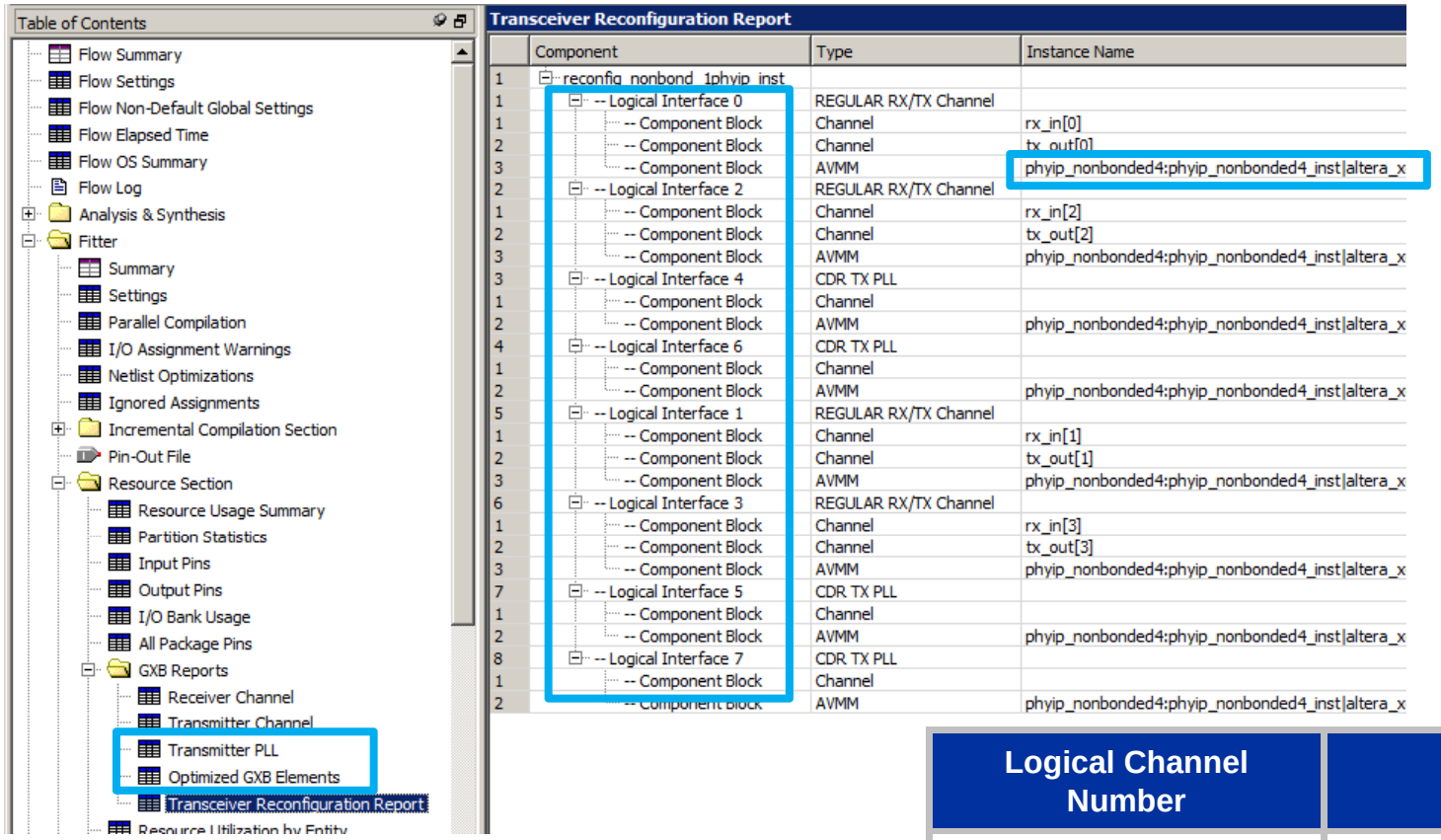
Example 2 Block Diagram w/ Connections



CUSTOM0.reconfig_from_xcvr[367:0] ⇒ RECONFIG0.reconfig_from_xcvr[367:0]

RECONFIG0.reconfig_to_xcvr[559:0] ⇒ CUSTOM0.reconfig_to_xcvr[559:0]

Transceiver Reconfiguration Report



Logical Channel Number	Channel Name
0	Channel 0
1	Channel 1
2	Channel 2
3	Channel 3
4, 5, 6, 7	TX (CMU) PLL (Channels 0 – 3)

Two PHY IP Cores with Non-Bonded Cores

- Two 4-channel Custom PHY IP core instances
- One transceiver reconfiguration controller

Example 3 Setup

4-Channel Instance (CUSTOM0)

1. Configure Custom PHY IP for 4 non-bonded, full-duplex channels
2. Enable reconfiguration features (if needed)
3. Note the number of reconfiguration interfaces in Messages window
   - 8 interfaces total
   - 0 – 3 : Transceiver channels
   - 4 – 7 : TX PLLs

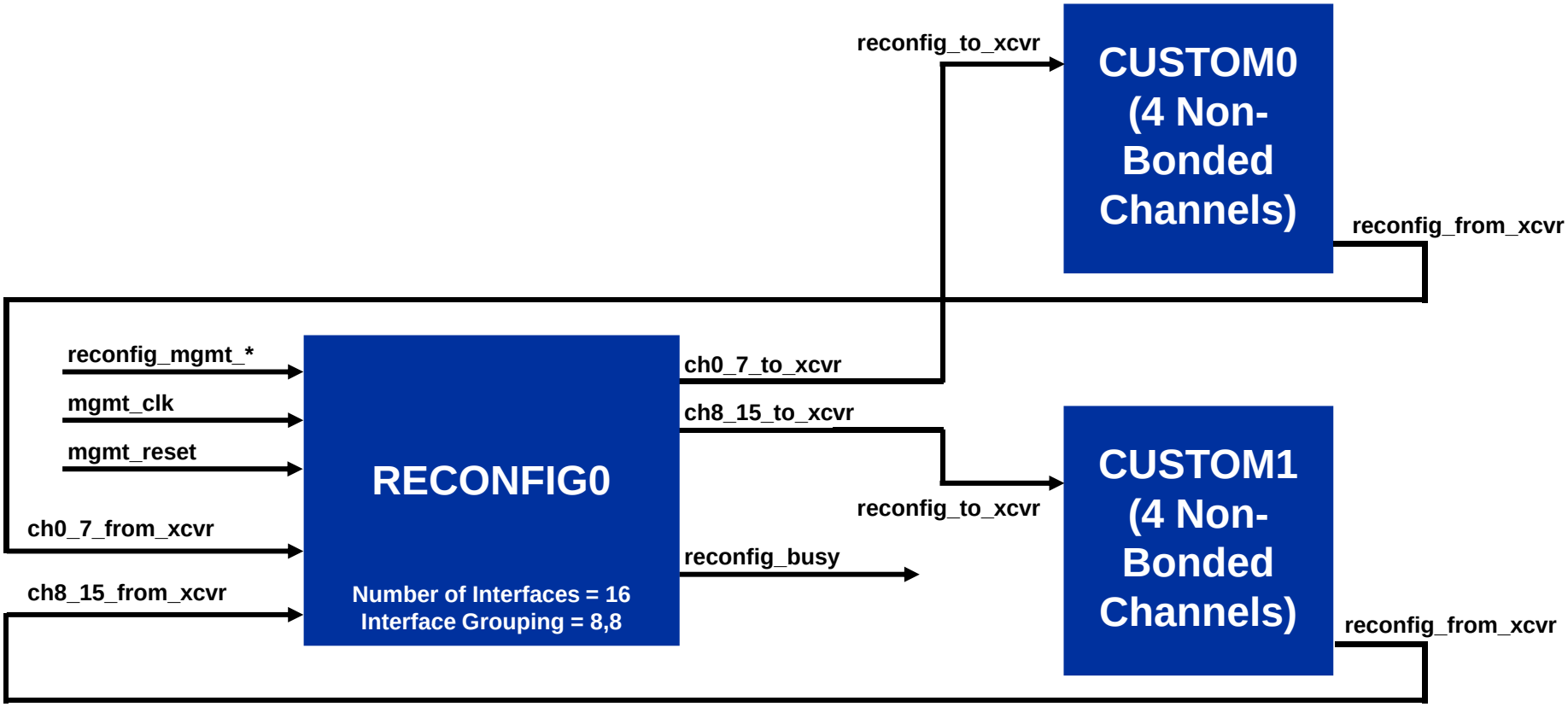
4-Channel Instance (CUSTOM1)

1. Configure Custom PHY IP for 4 non-bonded, full-duplex channels
2. Enable reconfiguration features (if needed)
3. Note the number of reconfiguration interfaces in Messages window
   - 8 interfaces total
   - 0 – 3 : Transceiver channels
   - 4 – 7 : TX PLLs

Reconfiguration Controller Instance (RECONFIG0)

1. Configure transceiver reconfiguration controller
2. Enable reconfiguration features
3. Set number of reconfiguration interfaces to 16
4. Set the grouping to 8,8

Example 3 Block Diagram w/Connections



CUSTOM0.reconfig_from_xcvr ⇒ RECONFIG0.ch0_7_from_xcvr

CUSTOM1.reconfig_from_xcvr ⇒ RECONFIG0.ch8_15_from_xcvr

RECONFIG0.ch0_7_to_xcvr ⇒ CUSTOM0.reconfig_to_xcvr

RECONFIG0.ch8_15_to_xcvr ⇒ CUSTOM1.reconfig_to_xcvr

Example 3 Logical Channel Number Assignments*

Logical Channel Number	Channel Name
0	CUSTOM0 Channel 0
1	CUSTOM0 Channel 1
2	CUSTOM0 Channel 2
3	CUSTOM0 Channel 3
8	CUSTOM1 Channel0
9	CUSTOM1 Channel1
10	CUSTOM1 Channel2
11	CUSTOM1 Channel3
4, 5, 6, 7	CUSTOM0 CMU PLL
12,13,14,15	CUSTOM1 CMU PLL

** One possible solution*

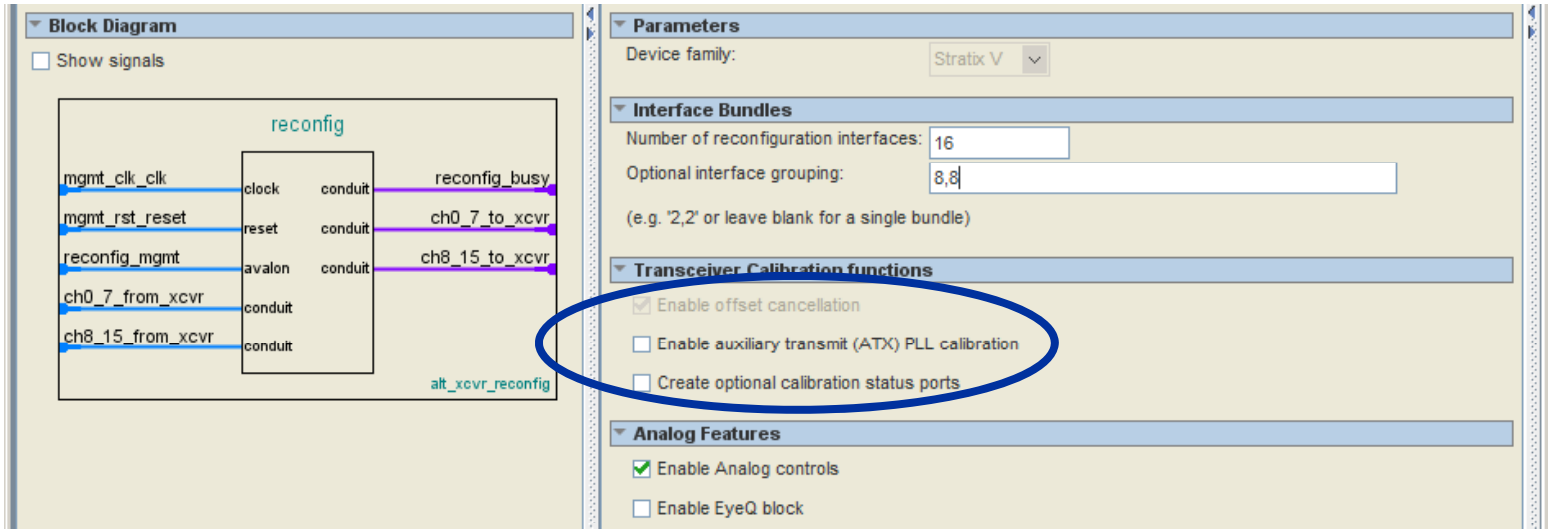
Calibration

- Transceiver reconfiguration controller automatically initiates calibration at power-up
- Types of calibration
  - Offset cancellation
    - Compensates for p-n voltage offsets due to process variations
    - Required for all channels
  - ATX PLL calibration
    - Tunes Stratix V ATX PLL parameters
    - Required for all Stratix V designs using ATX PLL
    - May be rerun after power-up (e.g. PLL does not lock after power-up)

Megafunction Settings for Calibration

- Offset cancellation automatically enabled in controller megafunction
- Other calibration types/options must be enabled

Transceiver Reconfiguration Controller



Performing Calibration

1. Upon power-up, transceiver reconfiguration controller initiates calibration
2. Embedded reset controller triggers reset when complete
3. Monitor *reconfig_busy* output flag or *busy* register to determine when controller is done
4. Monitor *tx_ready* and *rx_ready* signals to learn when channels are ready to send/receive data

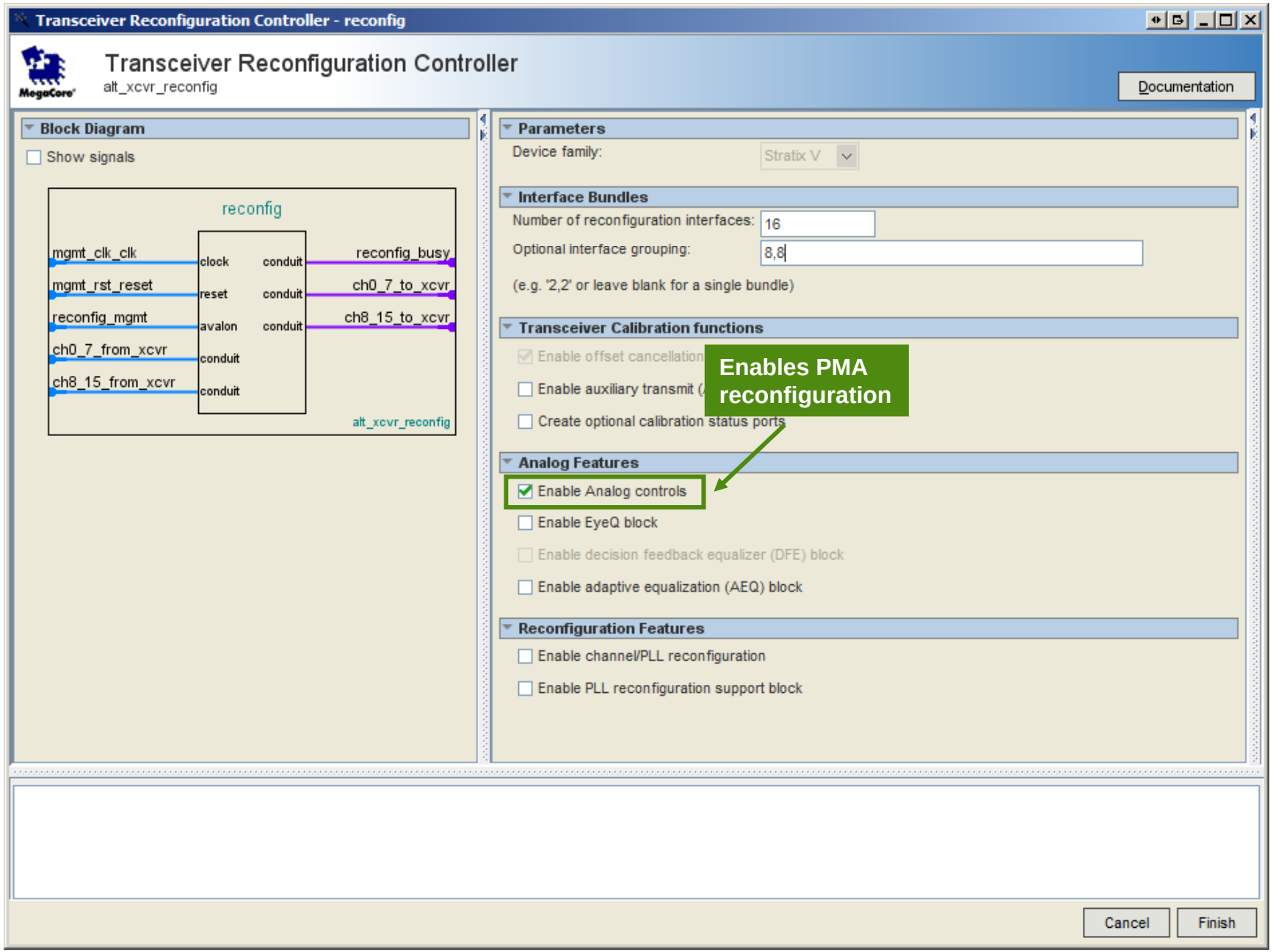
PMA Reconfiguration

- Selects from thousands of transmit and receive PMA analog settings dynamically

- Use to
  - Improve signal integrity during in-system tests and debugging
    - Fine tune transmit/receive buffers according to specific board/system conditions
  - Manually adjust settings to achieve target BER in FPGA or upstream device

Configurable PMA Settings

- Output differential voltage (V_{OD})
- Pre-emphasis support
- Equalization
- Equalizer DC gain



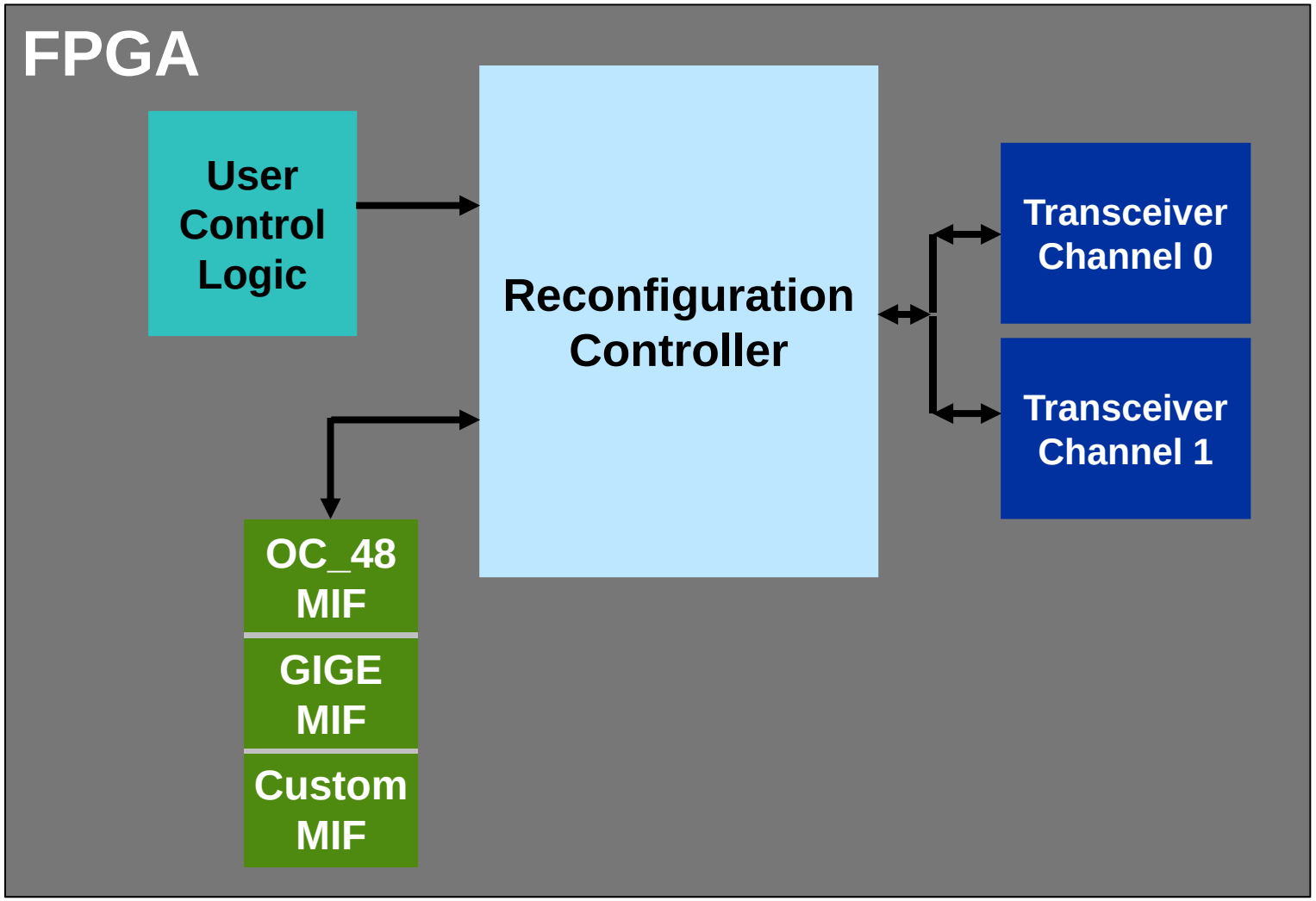
Memory Initialization File

- Used during reconfiguration to store settings for a single transceiver channel and/or PLL
  - Stores a single transceiver state
  - To support multiple transceiver configurations, you must generate and store multiple MIFs
- Generated automatically by Quartus II Assembler based on transceiver PHY IP core settings
- For design implementation, configure transceiver IP core for each set of reconfiguration options and recompile
  - Each recompilation creates a reconfiguration MIF
  - For faster compile times, consider creating a simplified reference or template design
- Reconfigure channels by writing new MIF into channel

Using MIFs

- Each MIF files has the settings for single or full duplex channel
  - Each MIF file is made up of 16-bit records (words)
    - Number of words determined by target device and target channel(s)
  - No limitation on the number of MIFs implemented in a single design
  - Single MIF can be used for reconfiguring multiple channels
- MIFs can be stored in embedded RAM or in off-chip memory
- Reconfiguration controller automatically accesses MIF records through MIF reconfiguration Avalon-MM master interface
- Reconfiguration controller writes MIF records into actual transceiver channel(s)

MIF Example



Note: MIFs can be stored in single memory or separate memory spaces

Locating Generated MIFs

- ***reconfig_mif*** subdirectory created in project directory
- Separate MIF created for each transceiver IP instance and each TX PLLs referenced in those instances
- Default filenames based on the transceiver IP instance names
  - User can rename MIF

Input Reference Clocks

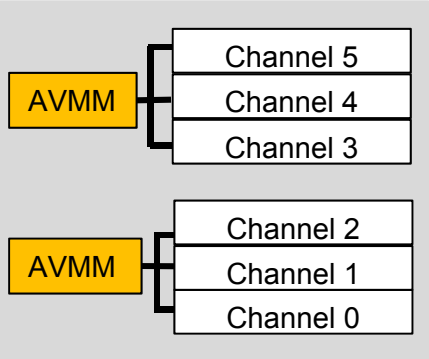
- PHY IP clock sources for TX PLLs and RX CDRs
- Transceiver megafunction supports up to 5 different input reference clocks when channel and PLL reconfiguration enabled
- Changing input reference clocks changes input clock frequency to PLLs and CRUs
  - Allows support for wider variety of data rates vs. changing PLL settings alone

More Details on Channel & PLL Reconfiguration

- Online documentation
  - Stratix V Device Handbook, Volume 3, Chapter 6: Dynamic Reconfiguration in Stratix V Devices
  - Arria V Device Handbook, Volume 3, Chapter 7: Dynamic Reconfiguration in Arria V Devices
  - Altera Transceiver PHY IP Core User Guide

Connection Considerations

- Stratix V supports one Avalon interface per triplet
  - Two AVMM per six-pack
- AVMM interface needs to be taken into account when planning the Reconfiguration Controller connections



Connection Considerations

- Good

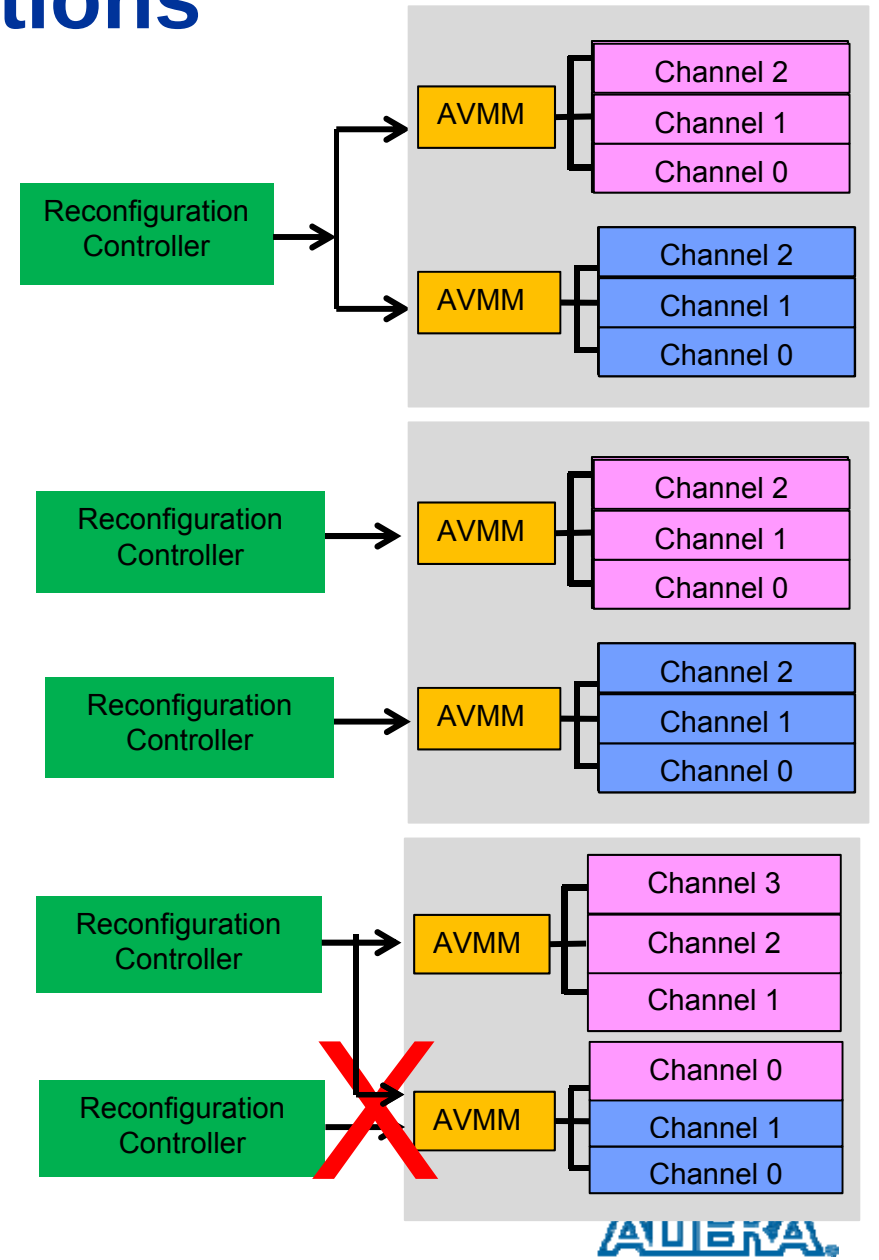
  – Do use one Reconfiguration Controller to control both AVMM interfaces within a 6-pack

  OR

  – Do use one Reconfiguration Controller per AVMM interface

- Bad

  – Do not use Two Reconfiguration Controller to connect to the same AVMM interface



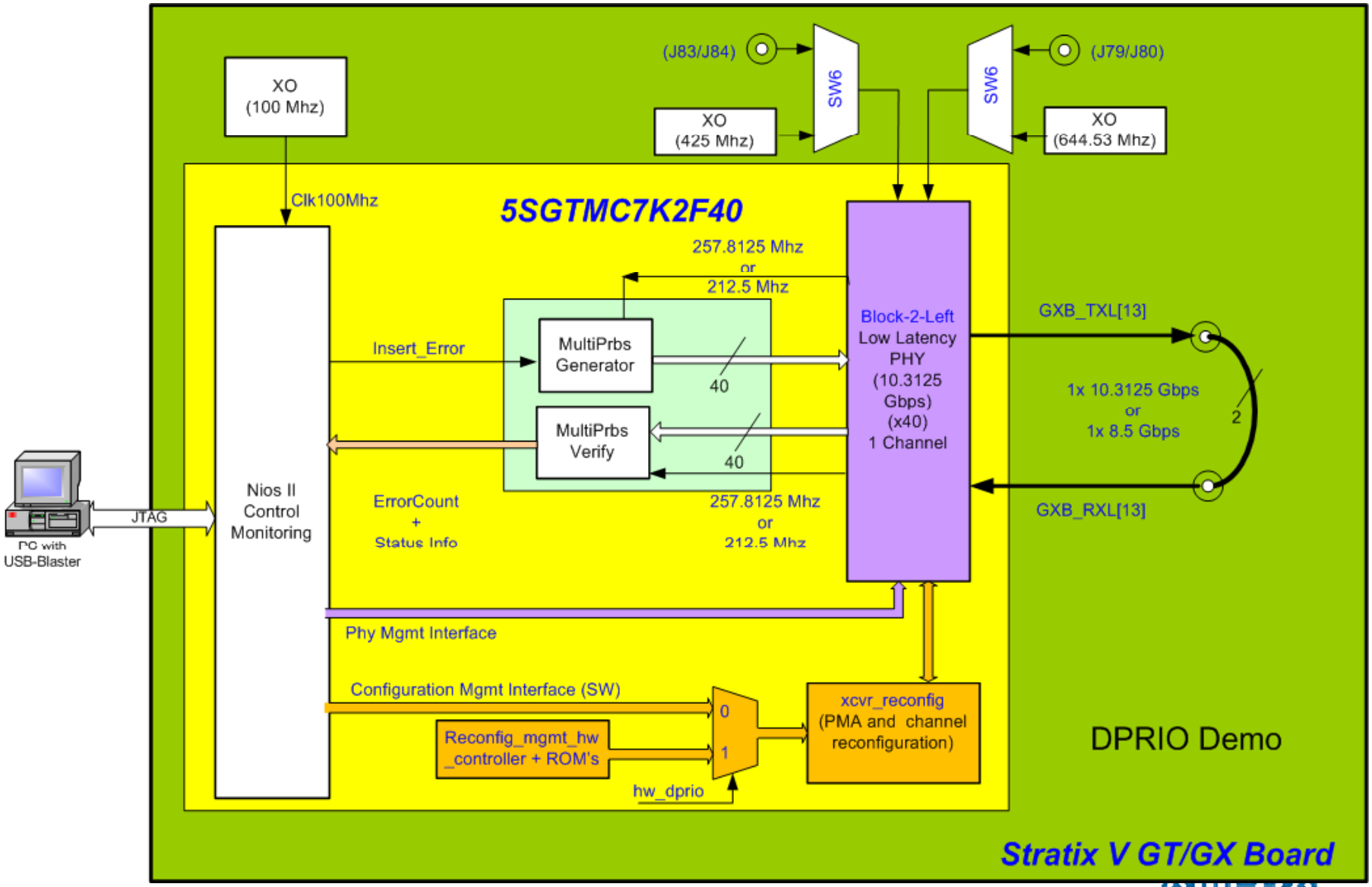
DPRIO Demo Design using MIF Streamer Mode 0 V12.0.0
For the Stratix V GX SI Board
Dynamically reconfigure between 2 datarates

Peter Schepers

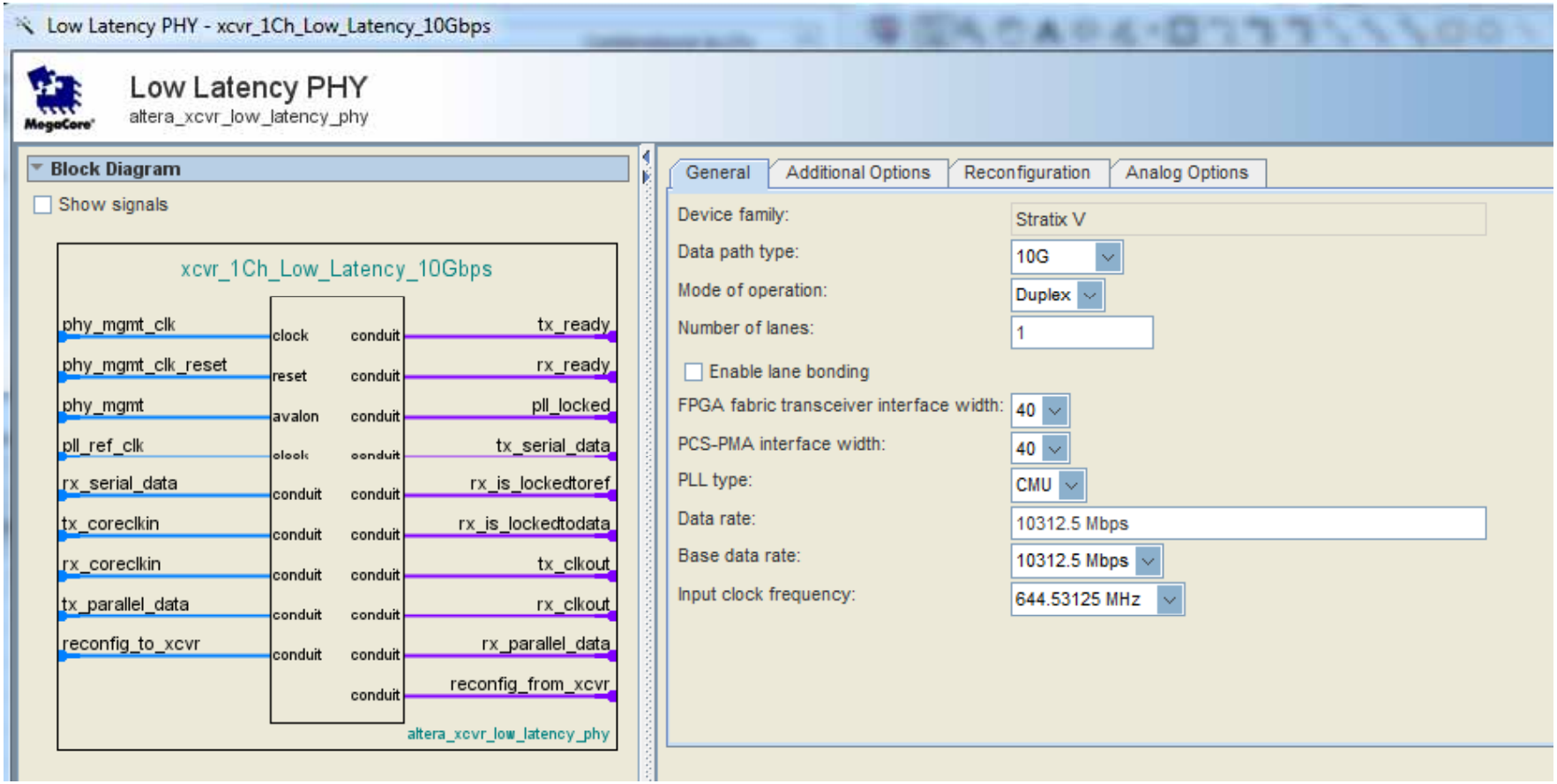
Staff Technology Specialist FAE High Speed Interfaces

June 28th , 2012

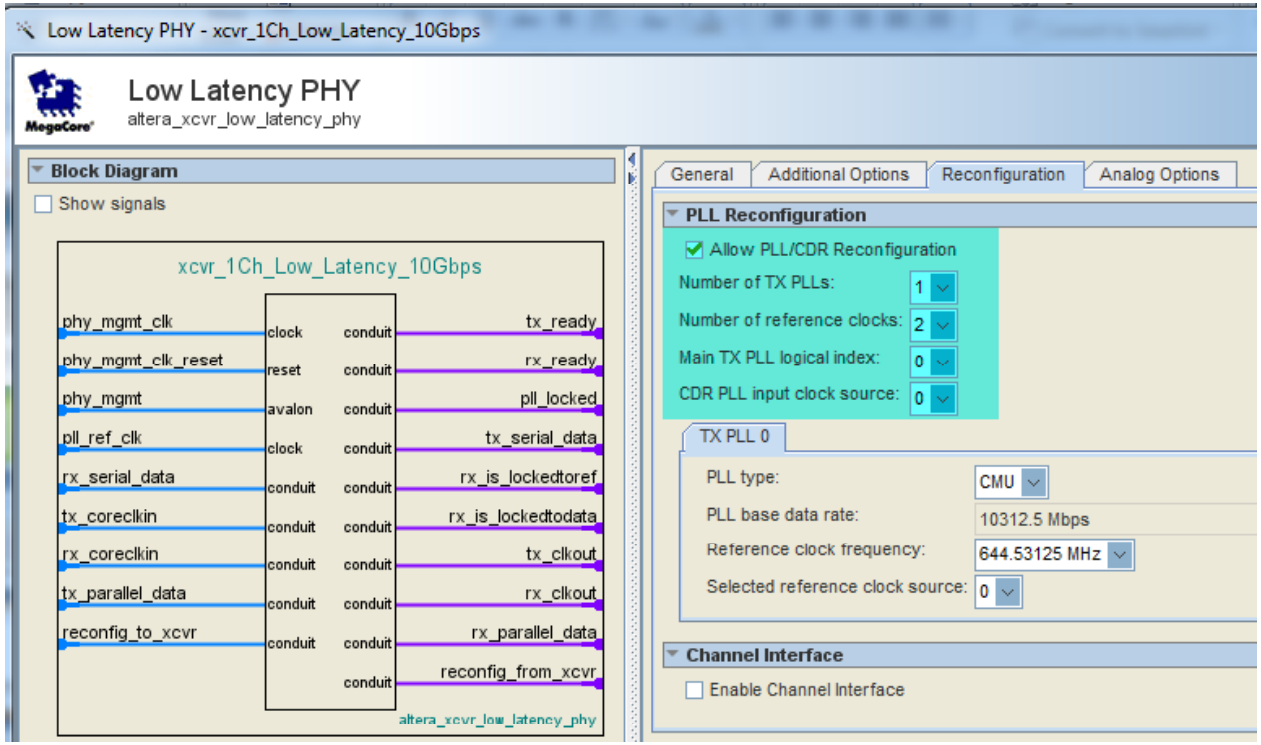
Blockdiagram



Transceiver Configuration 1st datarate (General Tab)

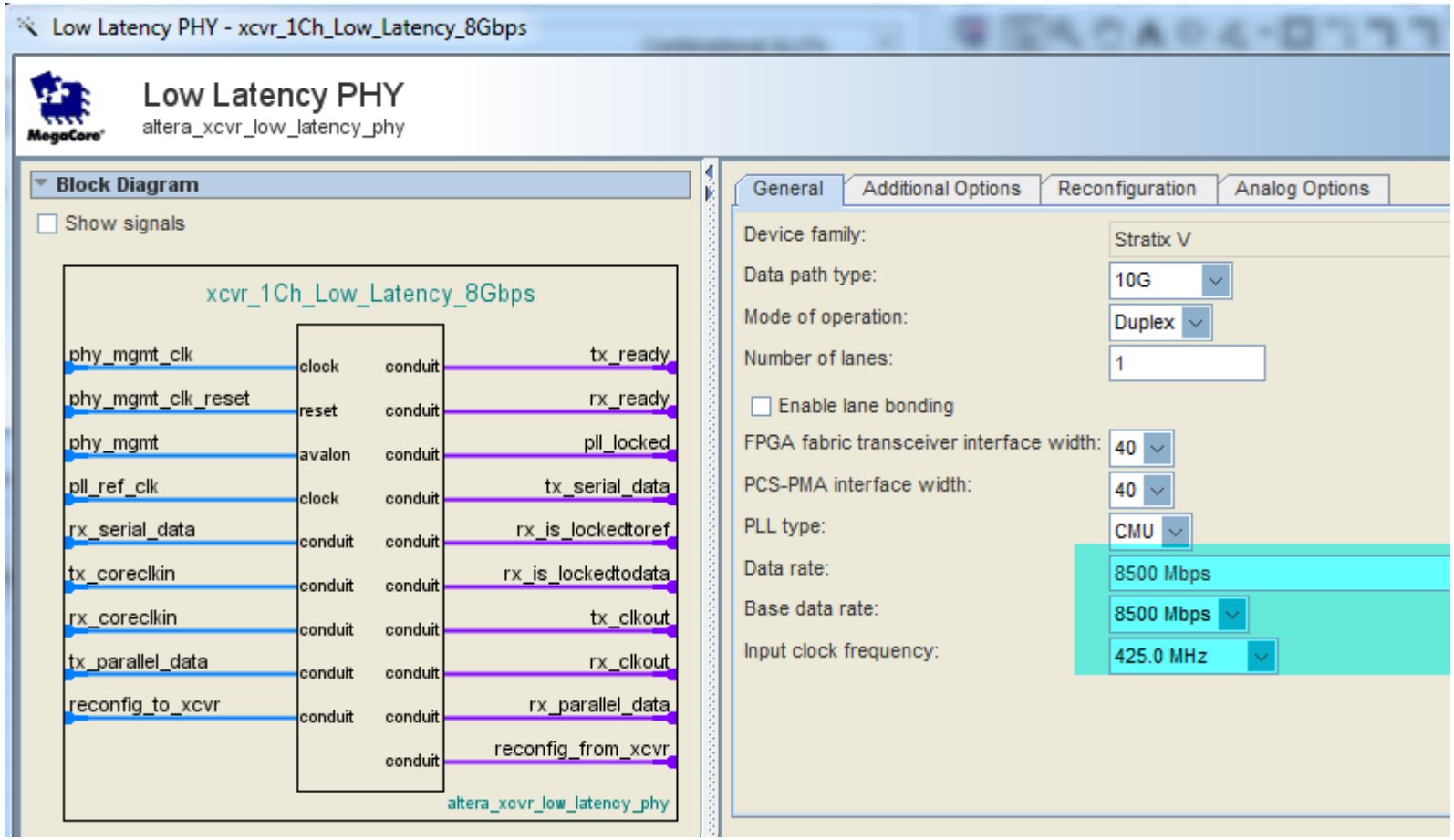


Transceiver Configuration 1st Datarate (Reconfiguration Tab)



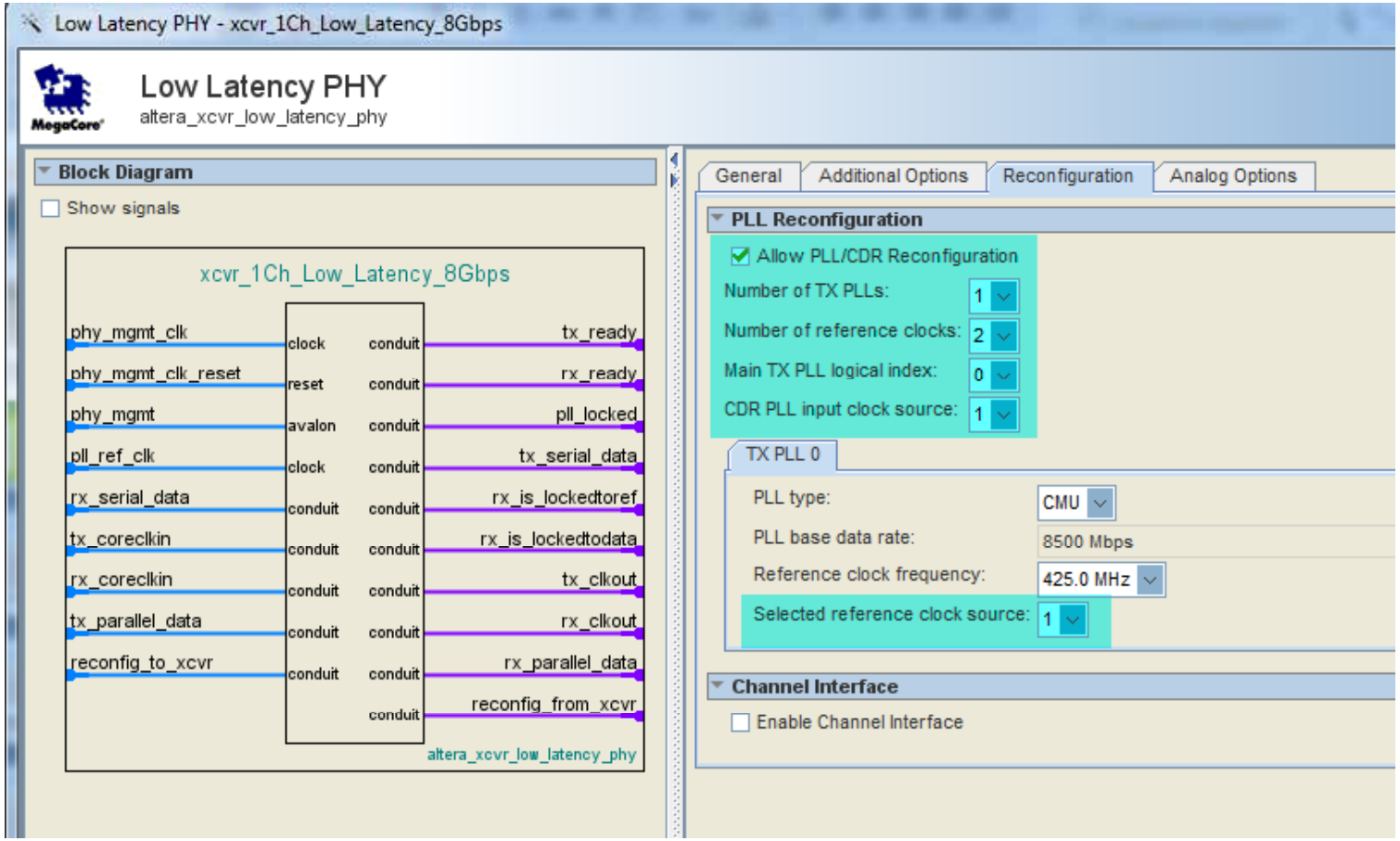
- "Allow PLL Reconfiguration" must be checked
- As 2 input referenceclocks will be used the number of input clocks should be set to '2'.
- For the first datarate the referenceclock with index 0 will be used for both the Tx as the Rx

Transceiver Configuration 2nd datarate (General Tab)



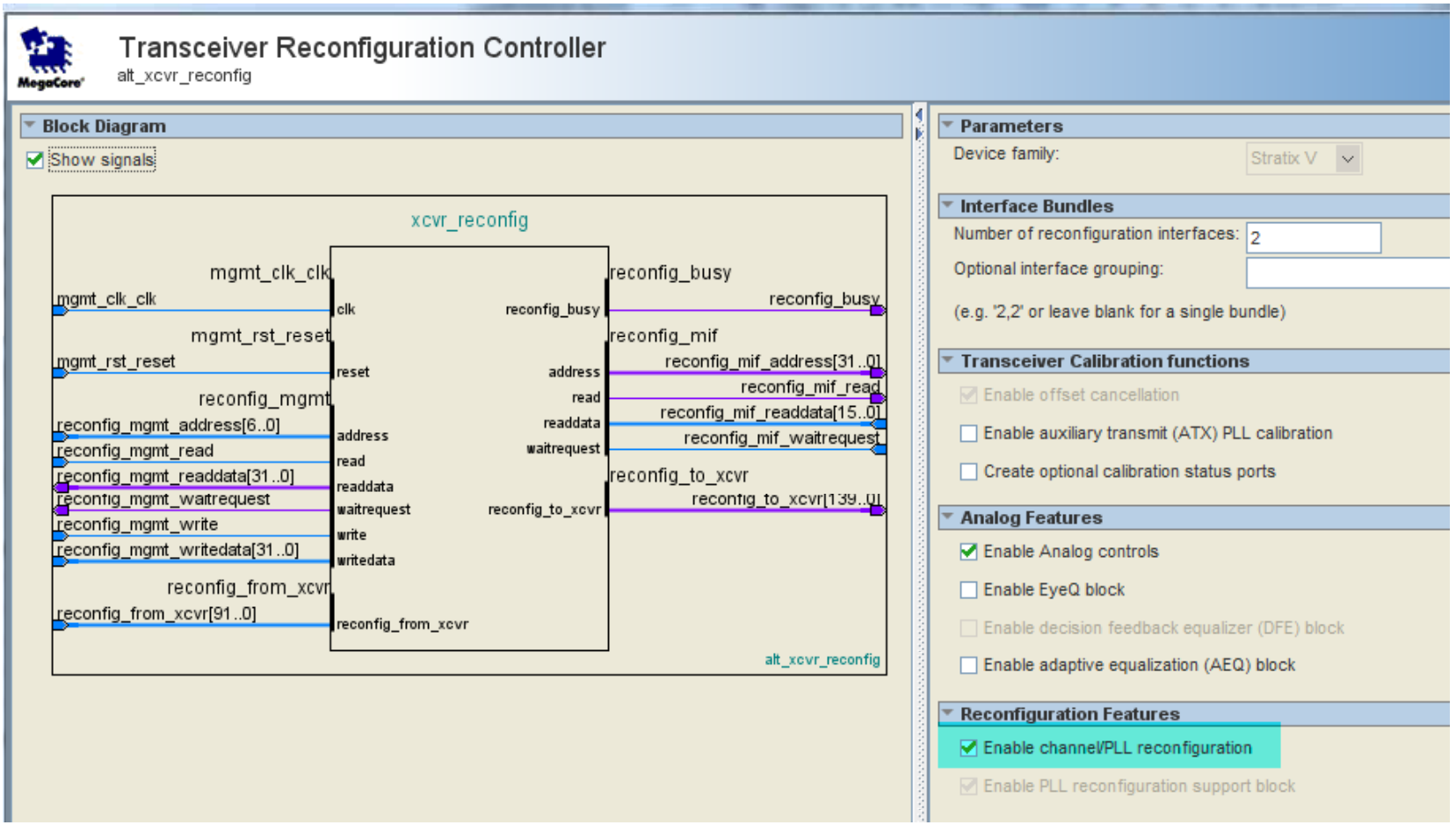
- The 2nd datarate is 8.5 Gbps from a 425 Mhz reference clock

Transceiver Configuration 2nd Datarate (Reconfiguration Tab)



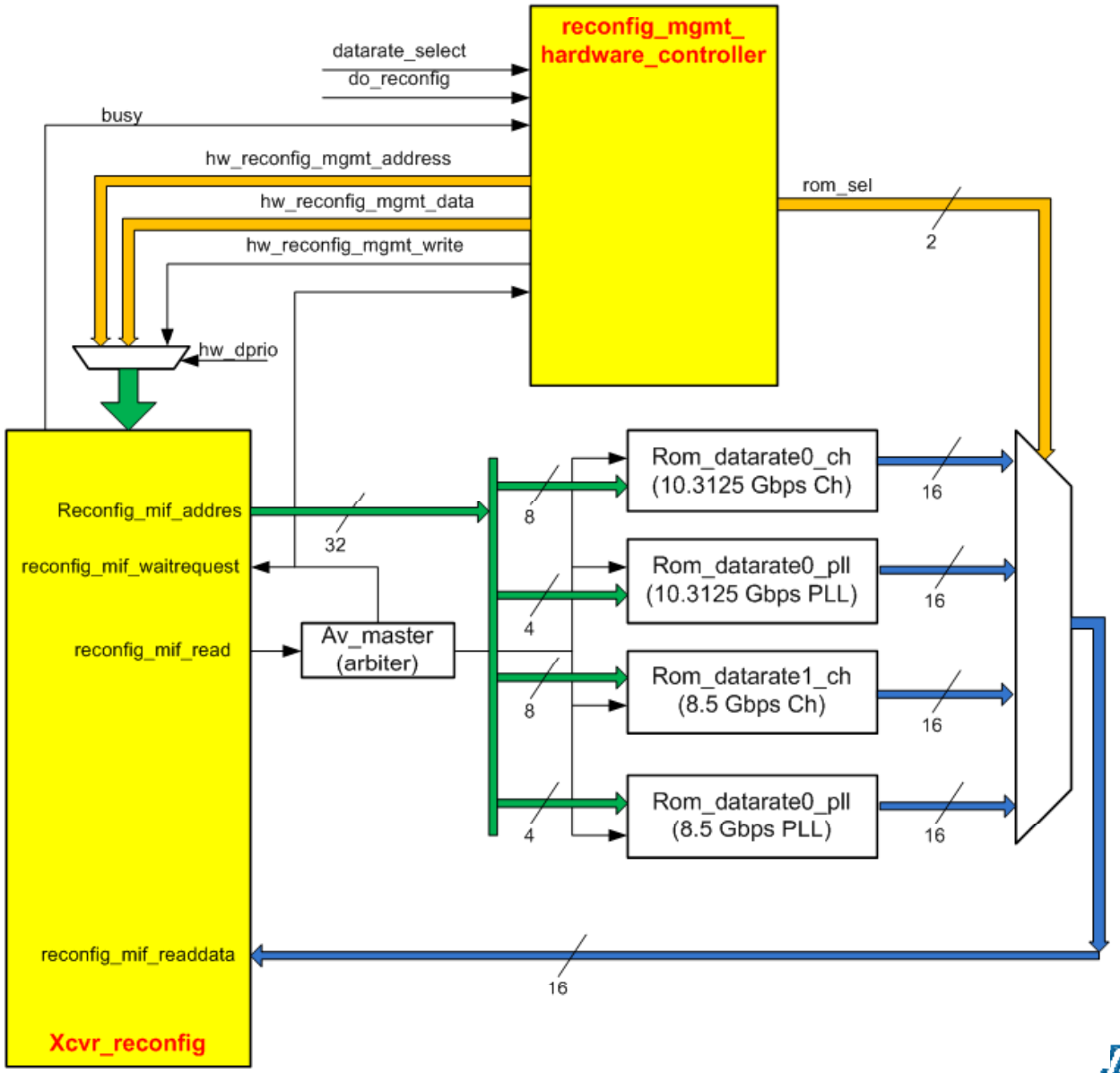
- Note that the TX PLL will be reconfigured in this example so there is only one TX PLL Logical Index
- For the first datarate the referenceclock with index 1 will be used for both the Tx as the Rx (Input clock source)

Reconfiguration Controller



- "Enable channel/PLL reconfiguration support block" must be checked (This will enable the reconfig_mif interface in the reconfiguration controller).

Reconfiguration Hardware Controller (Blockdiagram)

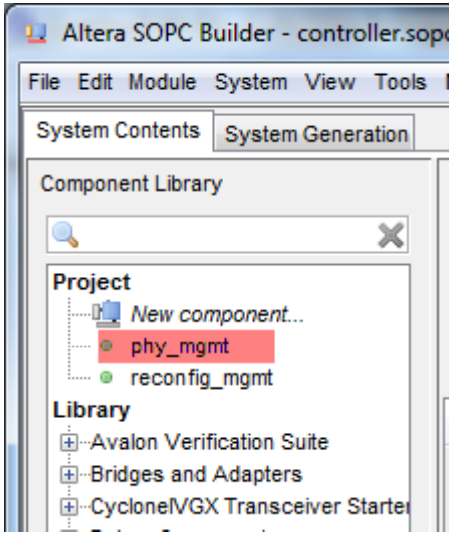


Reconfiguration Hardware Controller (Description)

- The Reconfiguration_Hardware_Controller Instance is a module which controls the reconfiguration controller (xcvr_reconfig) module through a HW statemachine (see further).
- Since there are 2 datarates being used there are in total 4 MIF files required : 2 for each datarate : one for the Channel (Tx/Rx) and one for the PLL.
- These MIF files are automatically generated during the compilation.
- In order to generate the MIF files for the 2nd datarate one needs to instantiate the 2nd datarate PHY instead of the 1st one and recompile the design (alternatively one can also use a dummy design which only instantiates the PHY and the reconfiguration controller using the same pinouts as the final design).
- The MIF files are stored in the ROM as indicated in the blockdiagram and a small arbiter (av_master) controls the accesses to the ROM's and generates the reconfig_mif_waitrequest signal accordingly.
- There is a MUX after the ROM's which will select the proper mif_readdata to feed it back to the reconfiguration controller.

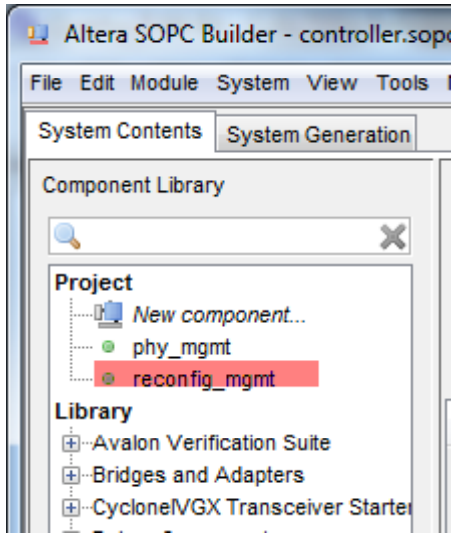
Phy Management Interface.

- The Stratix V PHY instance has a Phy Management Interface which is used to control the resets and the serial loopback.
- This PHY Management Interface is an Avalon MM Interface.
- This design is using a QSYS system which transparently maps an Avalon MM interface to the PHY Management Interface of the PHY.
- The component created for this in QSYS is called “phy_mgmt” and the phy_mgmt_hw.tcl file included in the project describes it’s implementation.
- You can verify it’s content by editing the phy_mgmt component.

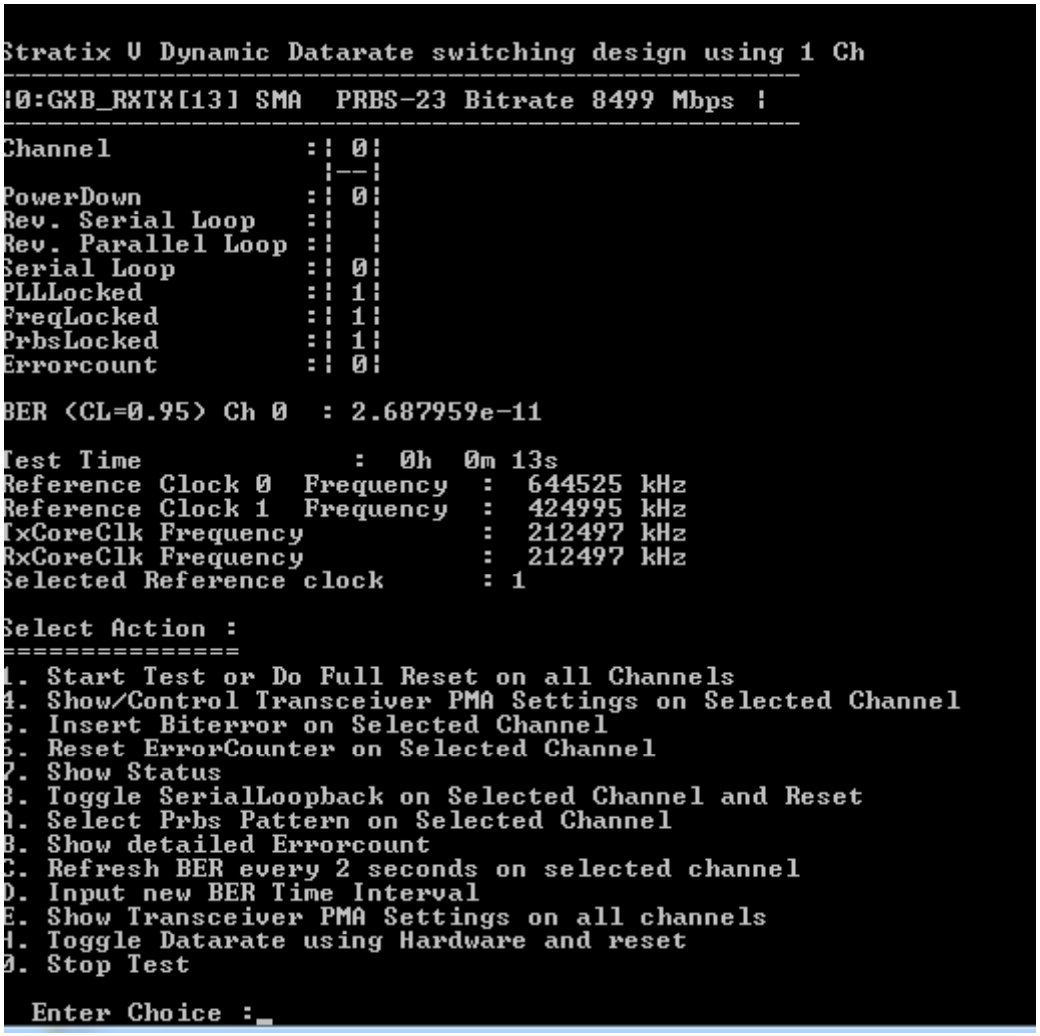


Reconfig Management Interface.

- The transceiver reconfiguration controller also has an Avalon MM Interface which is used to control the PMA settings of the transceivers the reconfiguration controller is connected to.
- Similar to the Phy Management Interface this design is also using the QSYS system which transparently maps another Avalon MM interface to the Reconfiguration Management Interface of the reconfiguration controller.
- The component created for this in QSYS is called “reconfig_mgmt” and the reconfig_mgmt_hw.tcl file included in the project describes it’s implementation.
- You can verify it’s content by editing the reconfig_mgmt component.

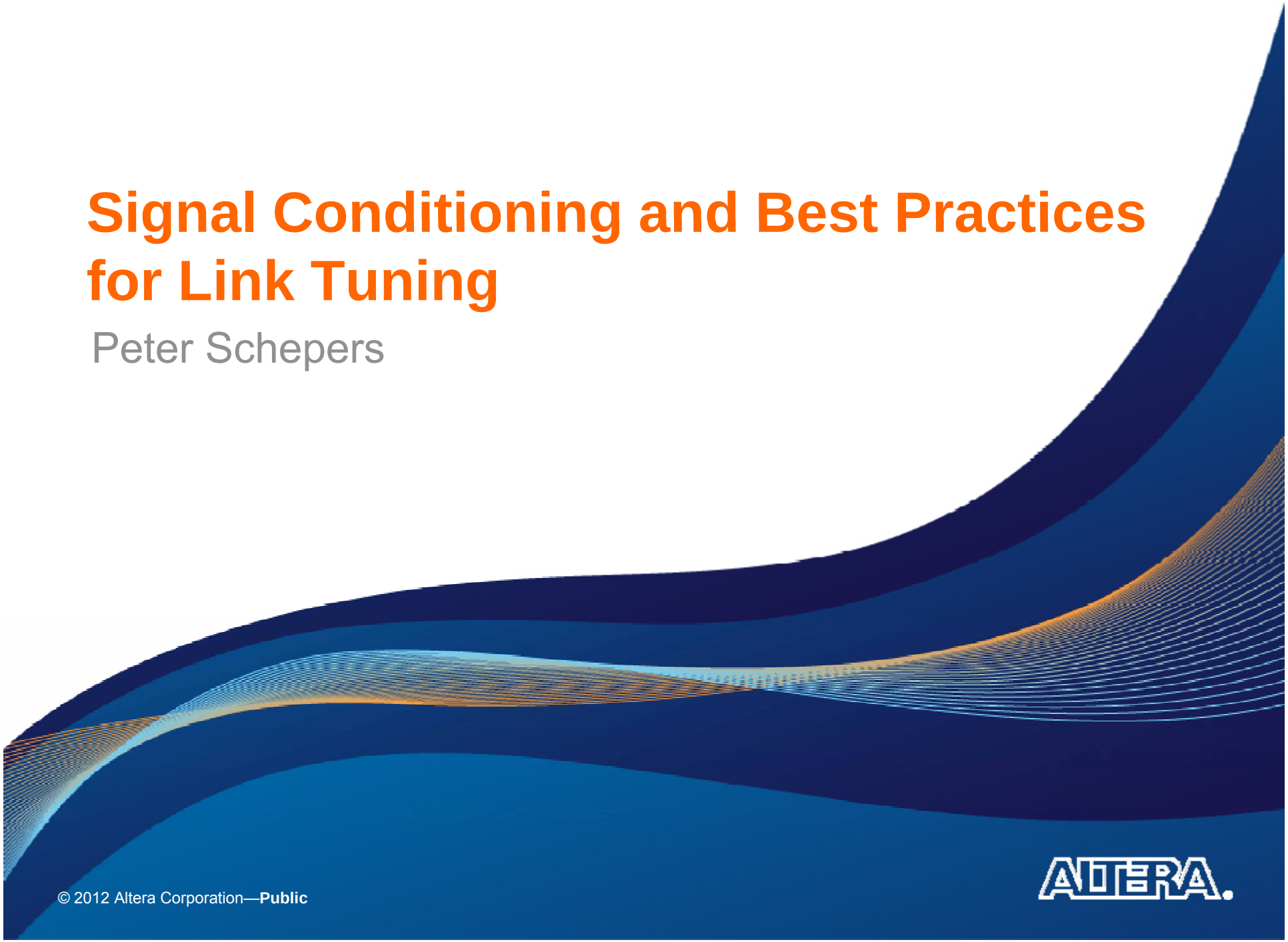


Nios 2 Output in Nios II SDK Shell



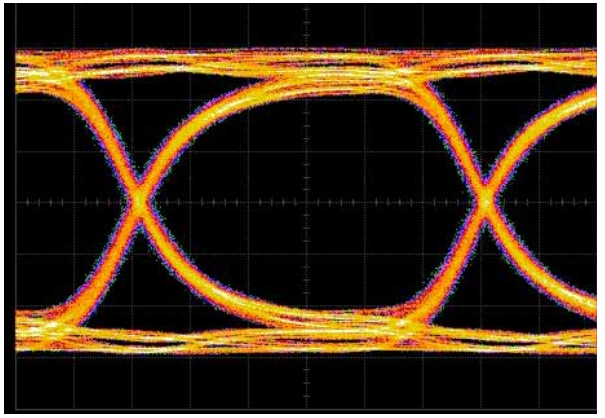
Signal Conditioning and Best Practices for Link Tuning

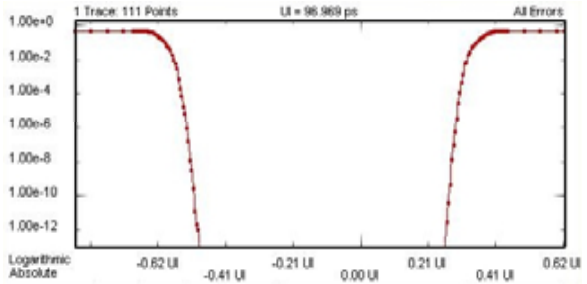
Peter Schepers



Signal Integrity Challenge at 10 Gbps and beyond

- Shrinking Margins
  - 1 UI = 80 ps @ 12.5 Gbps !
  - 1 UI = 35 ps @ 28 Gbps !!
- BER targets more stringent
  - Interlaken, CEI-11G, SFI-5.2, SFI-S : **1E-15**
- Tighter skew requirements
  - SFI-S requires skew of 5.5 UI at Tx
- Need to drive backplanes



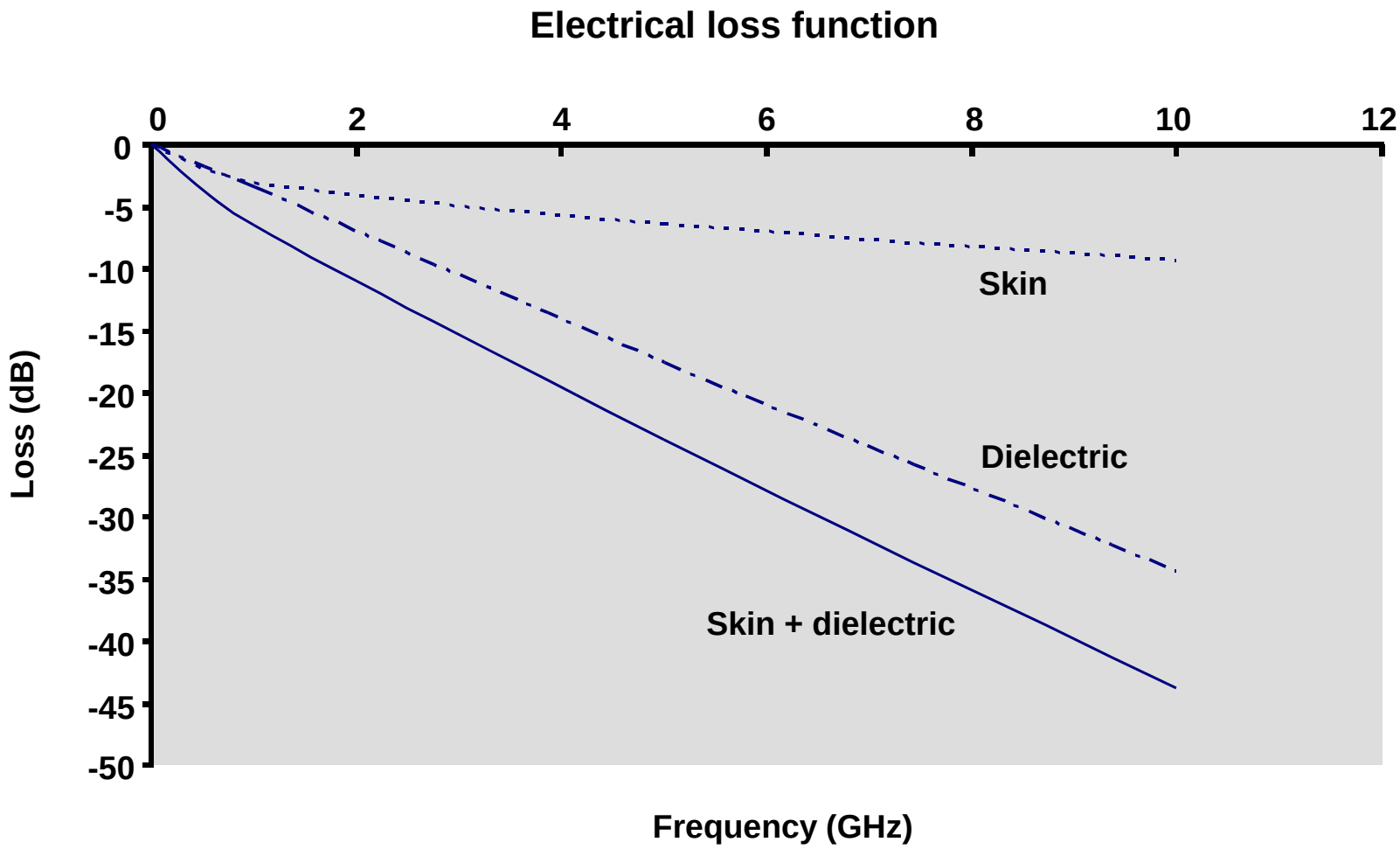


Specification	Data Rate	Reach	BER
10GBASE-KR	10.3125Gbps	1m (39”) + 2 connectors	< 1E-12
CEI-11G-LR	11.1Gbps	1m (39”) + 2 connectors	<1E-15

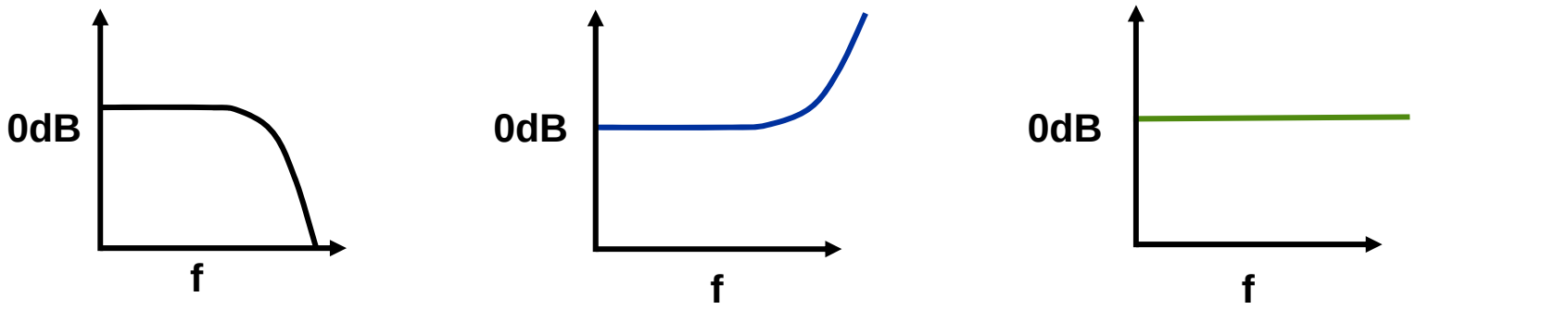
Backplane Challenges - Transceivers

- The burden is on the transceivers driving these backplanes to compensate all of the above described effects
- Altera Stratix V Transceivers provide different equalization tools to compensate backplane non-idealities
  - TX Finite Impulse Response(FIR) or Pre emphasis/de emphasis
  - Continuous time linear equalizer (CTLE) on the Rx side
  - Decision feedback Equalizer(DFE) on the Rx side

Physics for Electrical Channel Loss



Mechanisms for Equalization



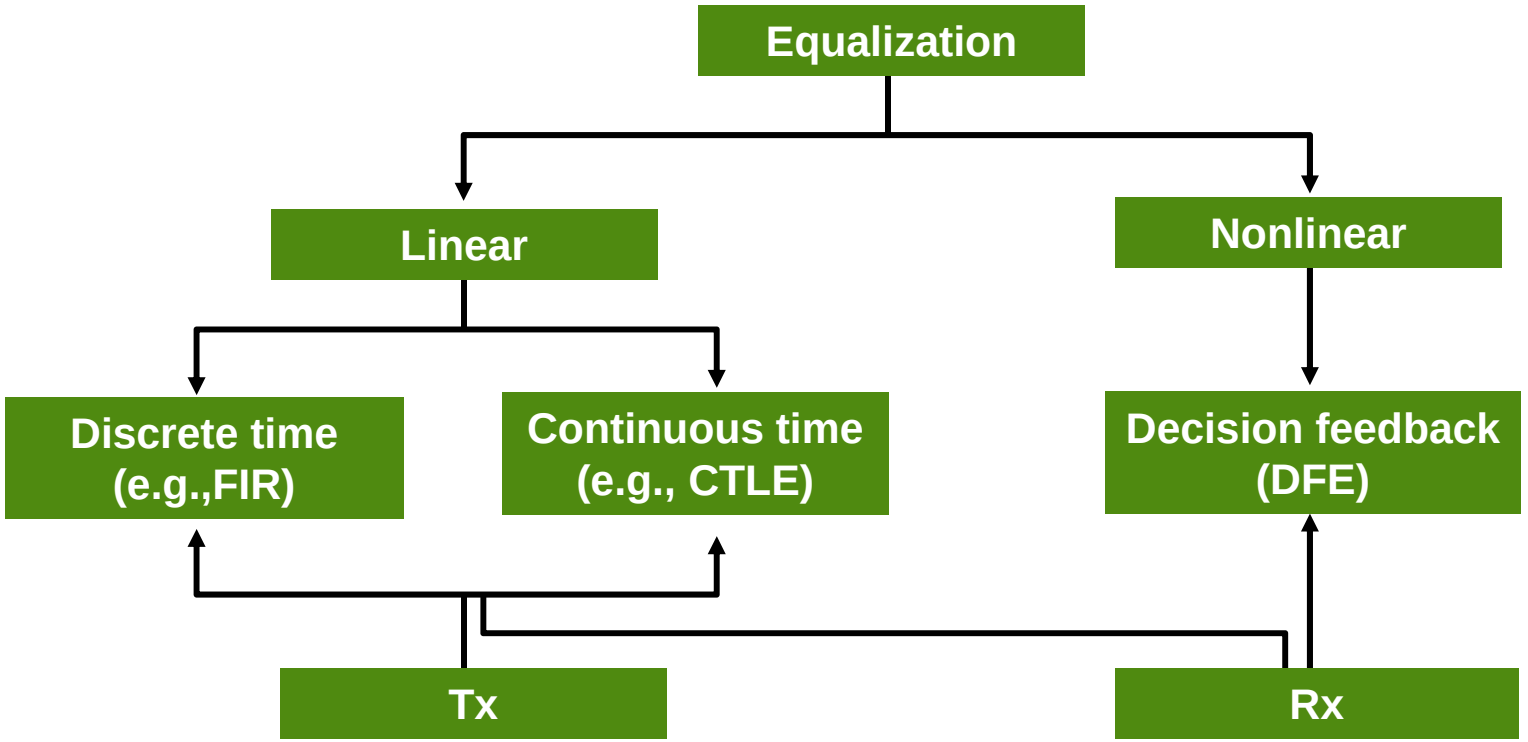
Channel **+** ***Equalizer*** **=** ***Flat overall response***

Tx equalizer:
pre-emphasis,
de-emphasis

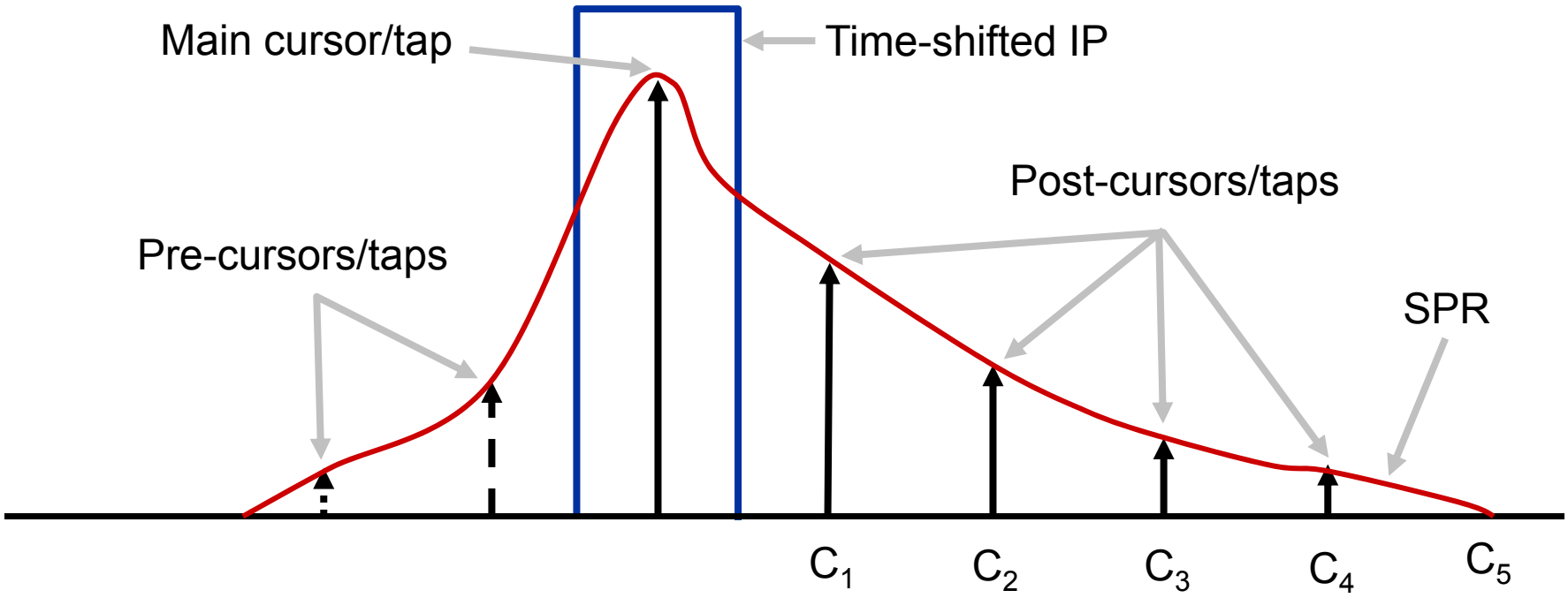
Rx equalizer:
linear (CTLE or FFE)
or adaptive (DFE)

Make the lossy channel a non-lossy channel so the overall "effective channel" is an "all-pass" function or has a flat response

Equalization Architecture Overview

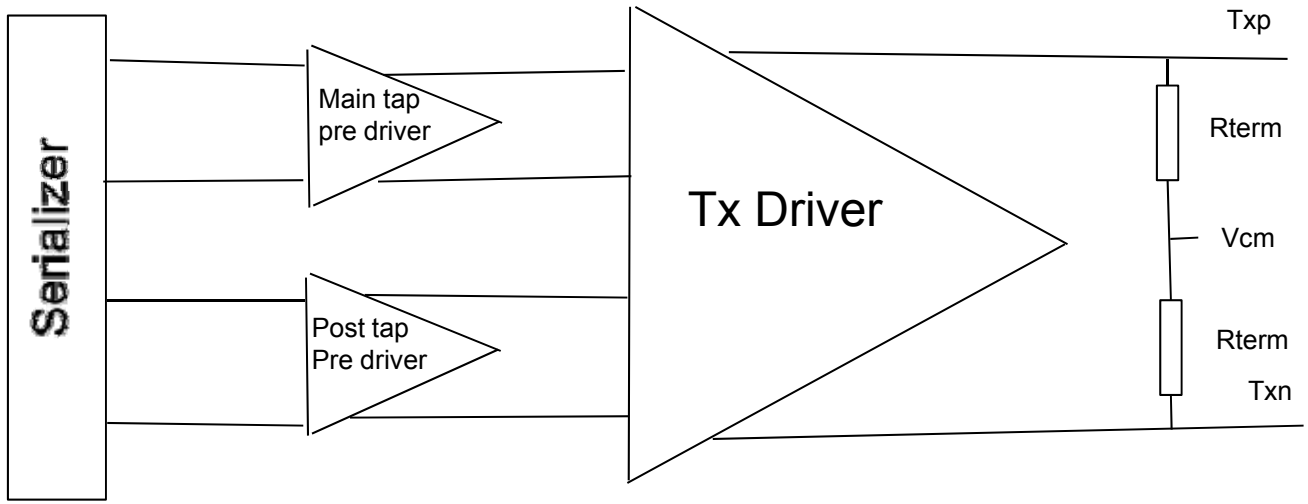


Ideal Pulse (IP), Single Pulse Response (SPR), and DFE Tap Coefficients



Transmitter (Tx) Buffer

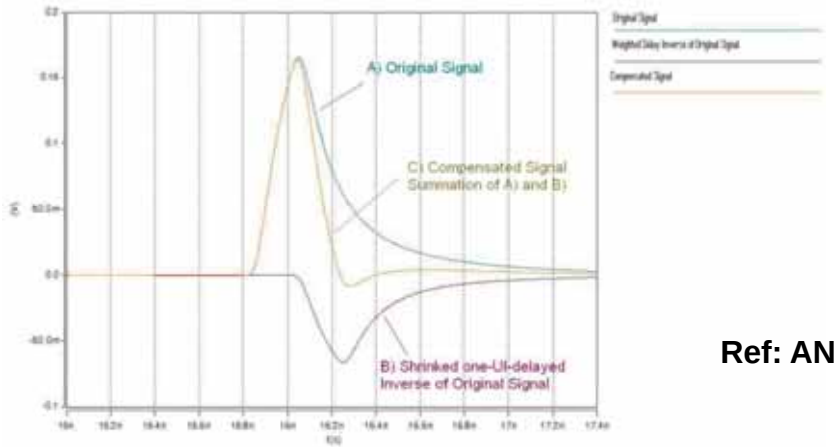
- Altera Stratix V Transmit buffer drives pair of differential signals through a pair of 50ohm impedance
- Tx offers programmable drive strength, pre emphasis and common mode voltage for enhanced signaling



Tx Buffer Structure

Tx Pre/de emphasis

- When signal travelling through a lossy (insertion loss) back plane, the transition expands to adjacent intervals, creating Inter symbol interference(ISI)
- Tx signal can be pre distorted(emphasis) so that after it goes through backplane, resulting signal is cleaner for Rx
- In general, two ways of pre conditioning the signal
  - Amplify high frequency contents → Pre-emphasis
  - Reduce low frequency contents → De-emphasis

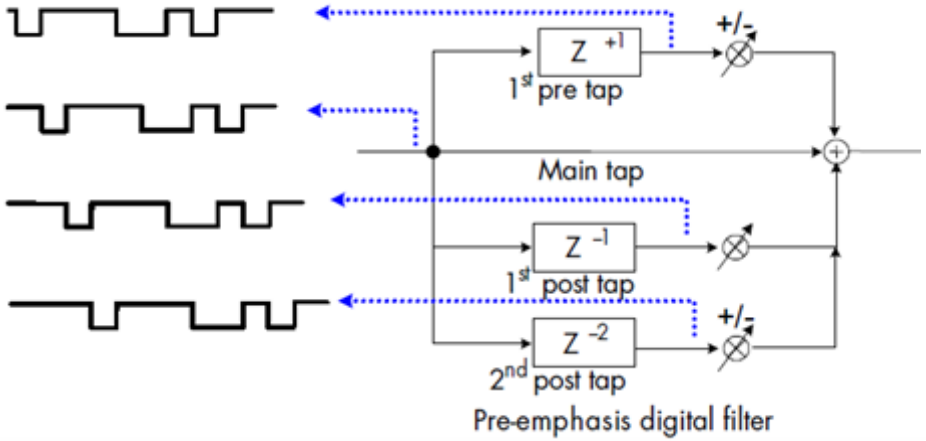


Ref: AN 602

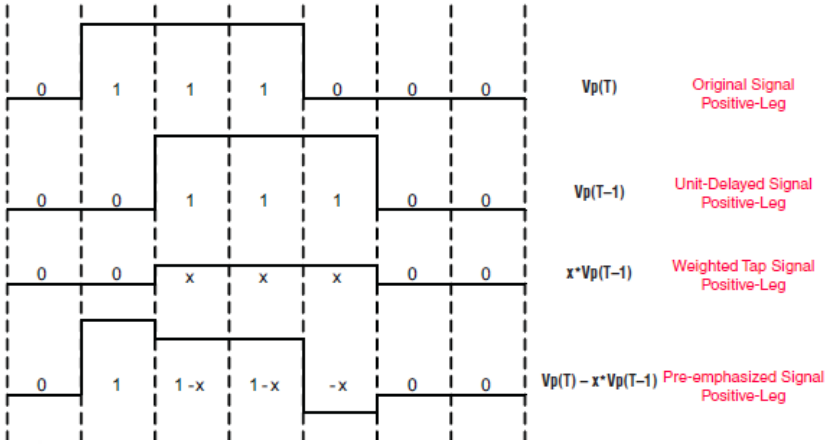
Pulse response – pre conditioning

Tx Pre/de emphasis contd..

- Altera Stratix V devices provide one pre-tap to address pre cursor(before transition of bit) ISI and two post taps to compensate post cursor(after transition of bit) ISI
  - This is accomplished by taking delayed (z^{-1} → Tx(n-1)) version of transmit data and adding its weighted value to the actual data



Tx Pre/de emphasis

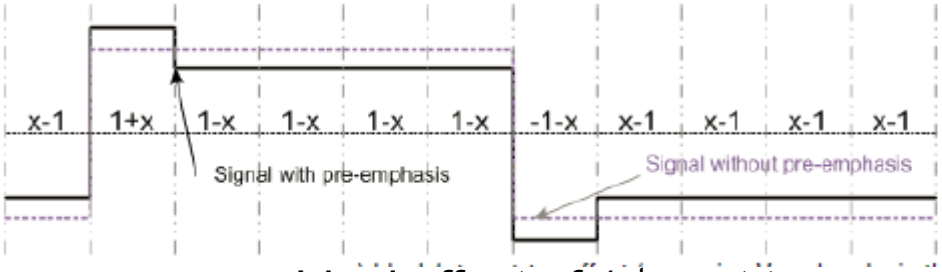


Effect of pre emphasis

Ref: AN 602

Tx 1st post tap effect

- 1st post tap is most effective tap
  - It emphasizes bit period immediately after transition and de emphasizes remaining bits
  - The emphasis part is not as significant as de emphasis part
  - De emphasis reduces vod level and minimizing signal power
  - Only positive polarity for first post tap

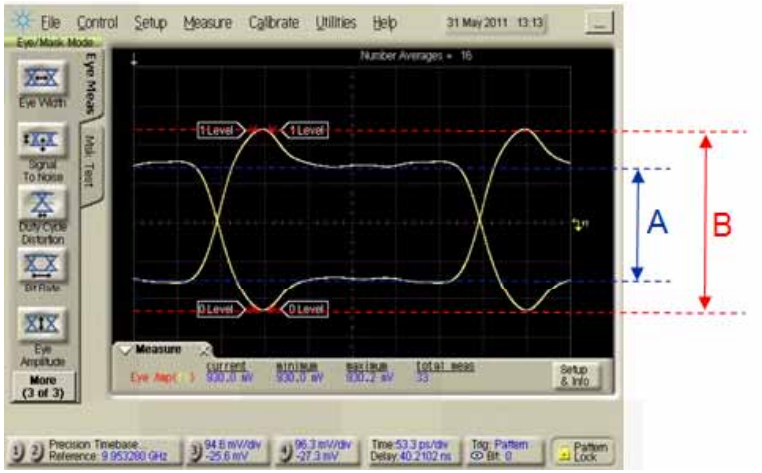


Vp(T)-x*Vp(T-1)-Vn(T)+x*Vn(T-1)

1st Post-Tap

Ideal effect of 1st post tap

  - Measure of post tap effect
    - 20log(B/A) in dB



Tx 1st post tap effect

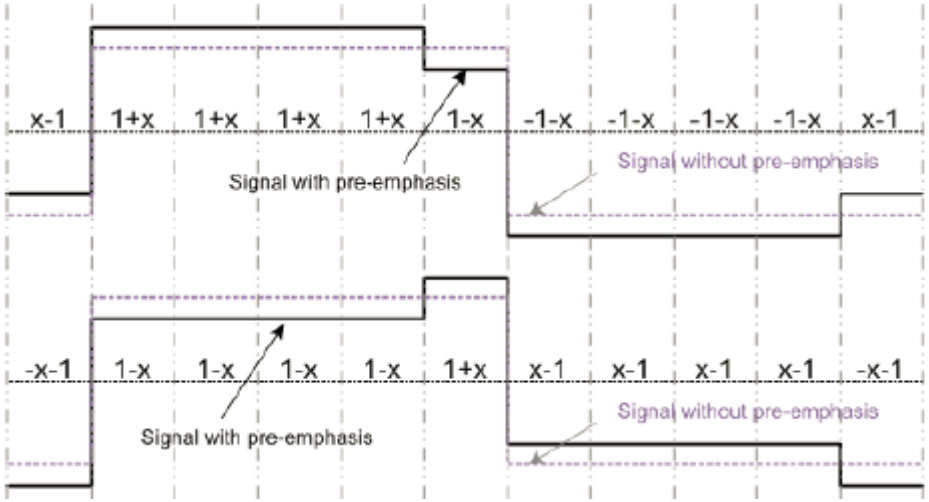
- Stratix V 1st post tap effect in dB
  - Note that pre emphasis is not very significant until the setting of 15

1stPostTap (mA)	A (mV)	B (mV)	dB
0	1.0124	0.7996	-2.05
0.2	0.9834	0.7807	-2.00
0.4	0.9682	0.7844	-1.83
0.6	0.9556	0.7866	-1.69
0.8	0.9438	0.7912	-1.53
1	0.9308	0.7974	-1.34
1.2	0.9179	0.8018	-1.17
1.4	0.905	0.8063	-1.00
1.6	0.8975	0.8135	-0.85
1.8	0.8654	0.8006	-0.68
2	0.8523	0.8049	-0.50
2.2	0.8379	0.8115	-0.28
2.4	0.8264	0.8183	-0.09
2.6	0.8113	0.8244	0.14
2.8	0.8001	0.8321	0.34
3	0.7865	0.8386	0.56

1stPostTap (mA)	A (mV)	B (mV)	dB
3.2	0.7781	0.8462	0.73
3.4	0.7456	0.8338	0.97
3.6	0.7317	0.8411	1.21
3.8	0.7192	0.8484	1.44
4	0.7068	0.8562	1.67
4.2	0.6854	0.8520	1.89
4.4	0.6845	0.8748	2.13
4.6	0.6697	0.8809	2.38
4.8	0.6560	0.8816	2.57
5	0.6291	0.885	2.96
5.2	0.6087	0.8813	3.21
5.4	0.5968	0.8917	3.49
5.6	0.5851	0.8974	3.72
5.8	0.5802	0.92	4.00
6	0.5693	0.93	4.26
6.2	0.5574	0.9366	4.51

Tx pre tap effect

- The pre tap de emphasizes bit before transition and emphasizes remaining bits
  - Negative polarity of pre tap does the opposite effect
  - Stratix V Tx buffer has +/-15 pre tap values
    - Pre tap has range of -1db to 2db



Vp(T)+x*Vp(T+1)-Vn(T)-x*Vn(T+1)

Pre-Tap

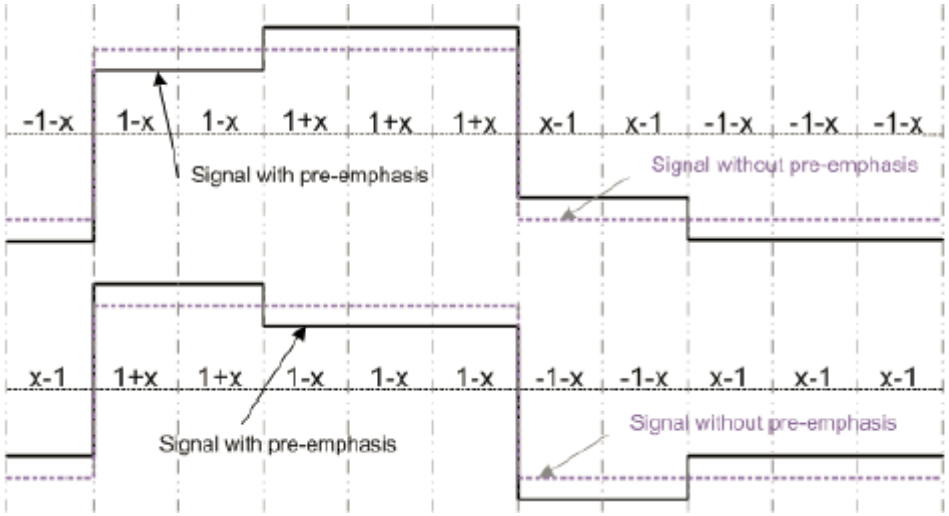
Vp(T)-x*Vp(T+1)-Vn(T)+x*Vn(T+1)

Inverted Pre-Tap

Ideal effect of pre tap

Tx 2nd post tap effect

- 2nd post tap de emphasizes first two bits after transition and emphasizes remaining bits
  - Negative polarity does the opposite effect
  - Stratix V has +/-15 2nd post tap values
    - 2nd post tap has less effect relatively, it has range of 0.8db to 1.5db



Vp(T)+x*Vp(T-2)-Vn(T)-x*Vn(T-2)

2nd Post-Tap

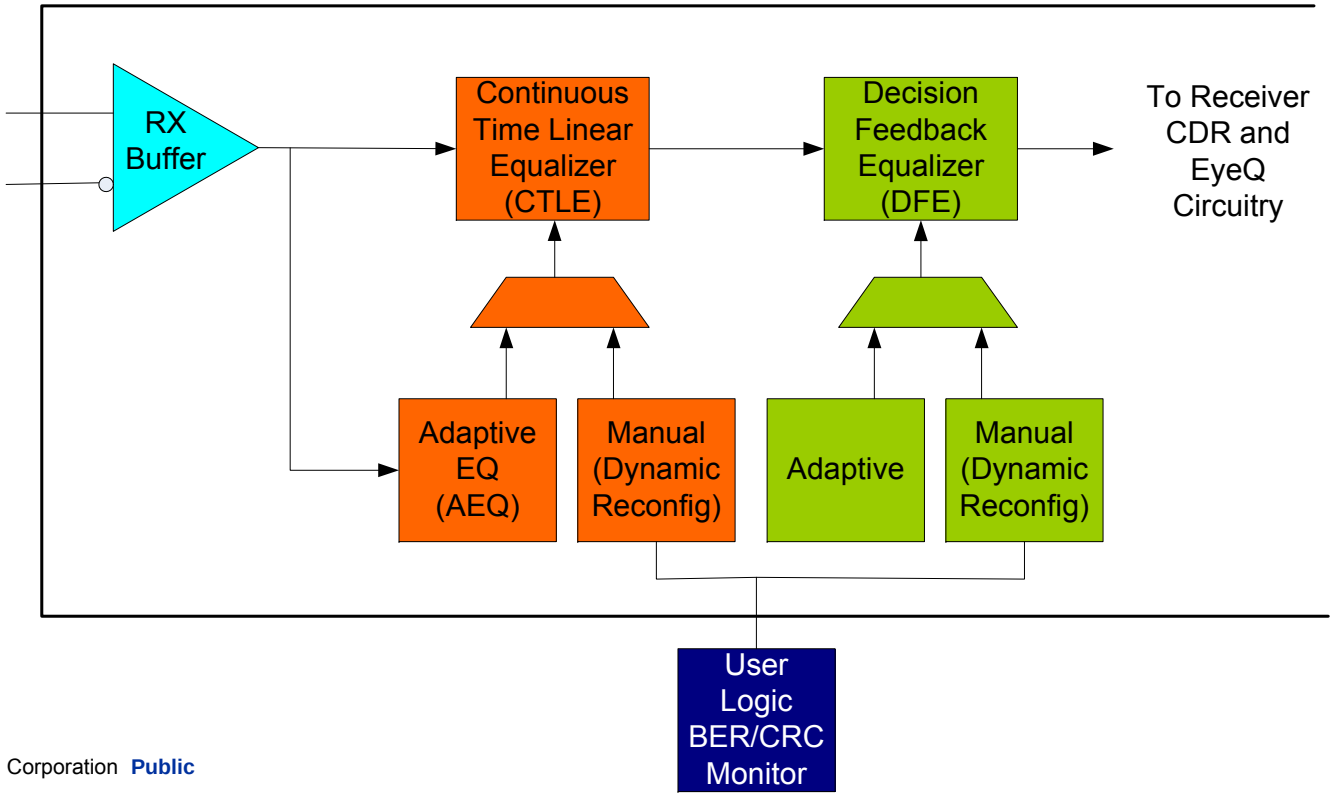
Vp(T)-x*Vp(T-2)-Vn(T)+x*Vn(T-2)

Inverted 2nd Post-Tap

Ideal effect of 2nd post tap

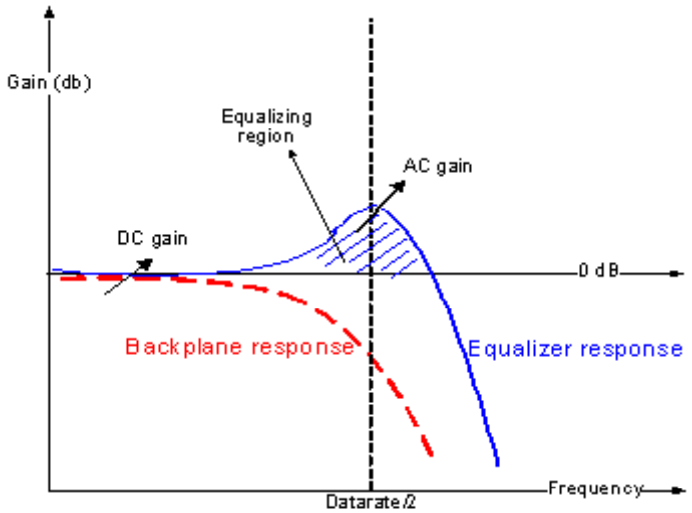
Receiver Equalization

- Stratix V receivers feature the following two types of equalizers
  - Continuous Time Linear Equalizer (CTLE)
  - Decision Feedback Equalizer (DFE)

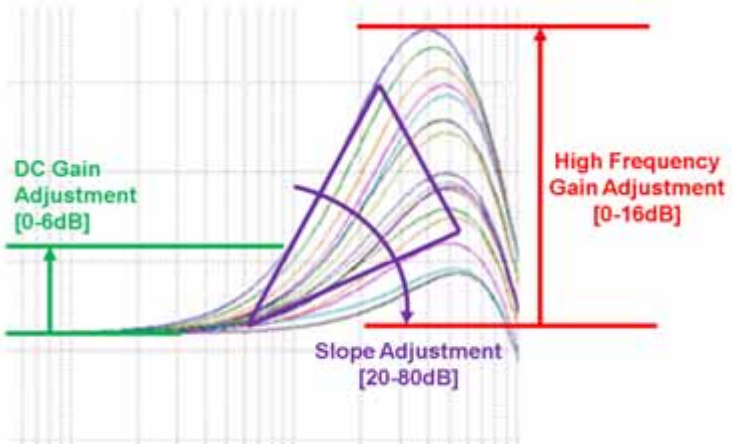


Rx Equalizer/CTLE

- The equalizer boosts high frequency components and compensates backplane loss
- The goal of equalizer is able to fit/compensate different backplane losses
  - Based on analysis of different backplanes at 10.3Gbps, Stratix V 4 stage Rx Equalizer is designed



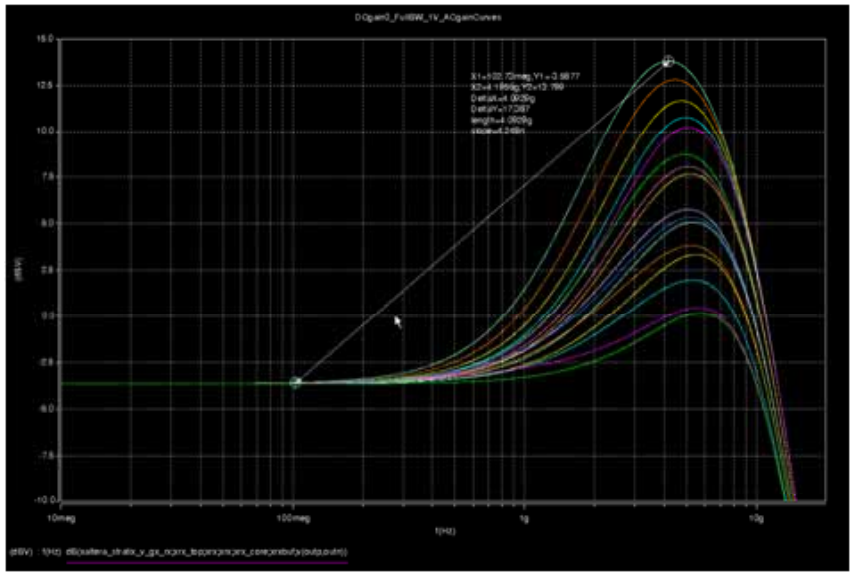
Simple Equalization response



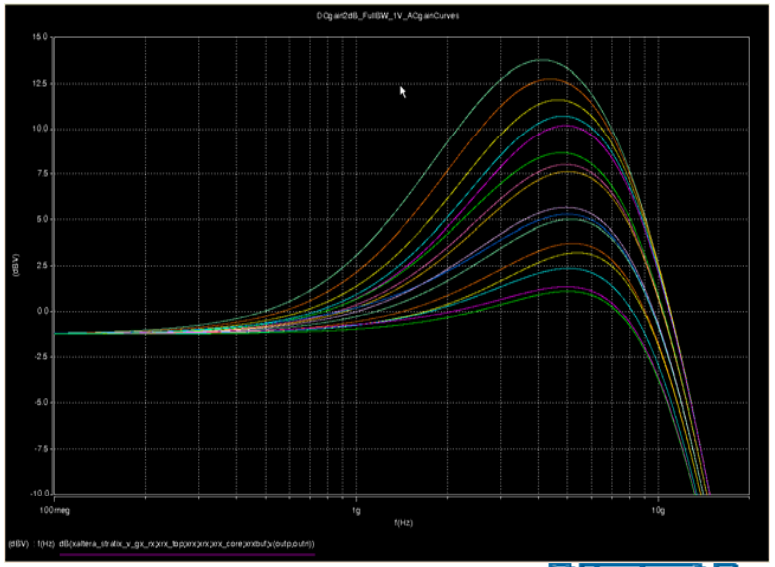
4 stage CTLE curves

Rx Equalizer/CTLE

- Rx Equalizer has programmable controls on
  - DC Gain boost up to 12dB and serves as variable Gain amplifier
  - Slope adjustment up to 80db/dec with 4 stages
  - High frequency gain adjustment up to 16db
    - With out enabling of any stages, there is still AC gain of ~5dB for Equalizer
  - Bandwidth adjustment for 6.25Gbps and 12.5Gbps
- Stratix V default DC gain is changed to 3dB



DC GAIN 0dB



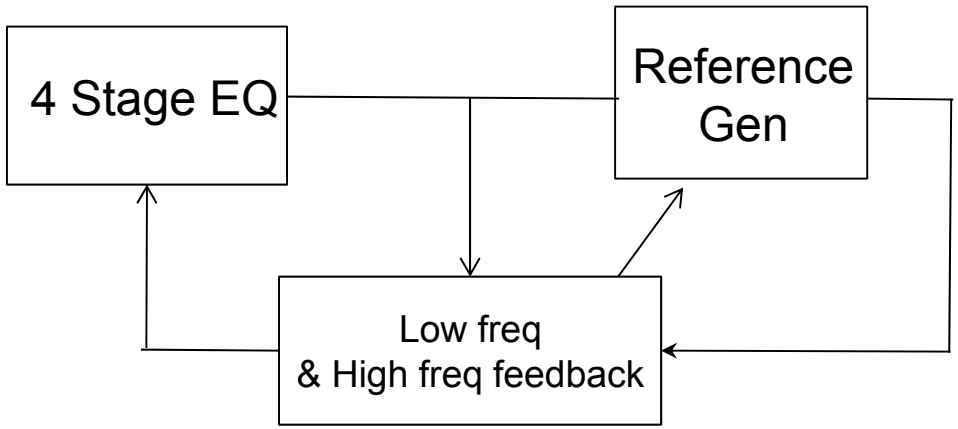
DC GAIN 3DB

Adaptive Equalization (AEQ)

- In order to tune CTLE to fit to many of different customer backplanes and to avoid manual selection of ~240 settings on CTLE, Adaptive engine is required
- Modes of AEQ
  - Continuous adaptation
    - Continuously adjusting 4stage equalizer settings based on input data
  - One time Adaptation
    - Adapts 4 stage EQ settings and freezes those values. Low power consumption
  - Manual
    - User picks one of 16 available AC settings and DC settings.

Adaptive Equalization (AEQ)

- Continuous Adaptation
  - The low frequency content and high frequency content of reference edge generator and equalizer are matched in a feed back mechanism
    - Low frequency and high frequency content are extracted using low pass and high pass filters
  - Adaptation done signal goes high after matching of signals frequency content

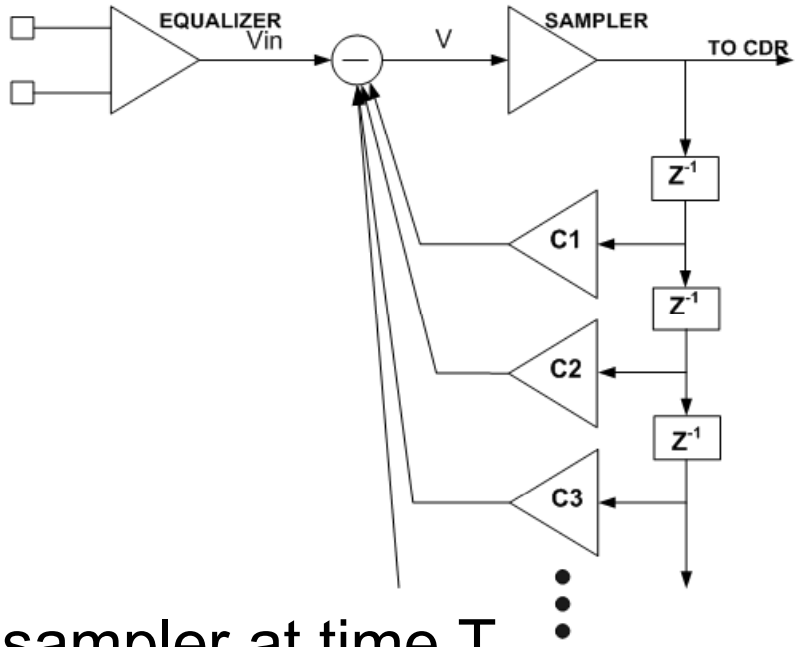


Block diagram of AEQ Adaptation mechanism

Decision feedback Equalization (DFE)

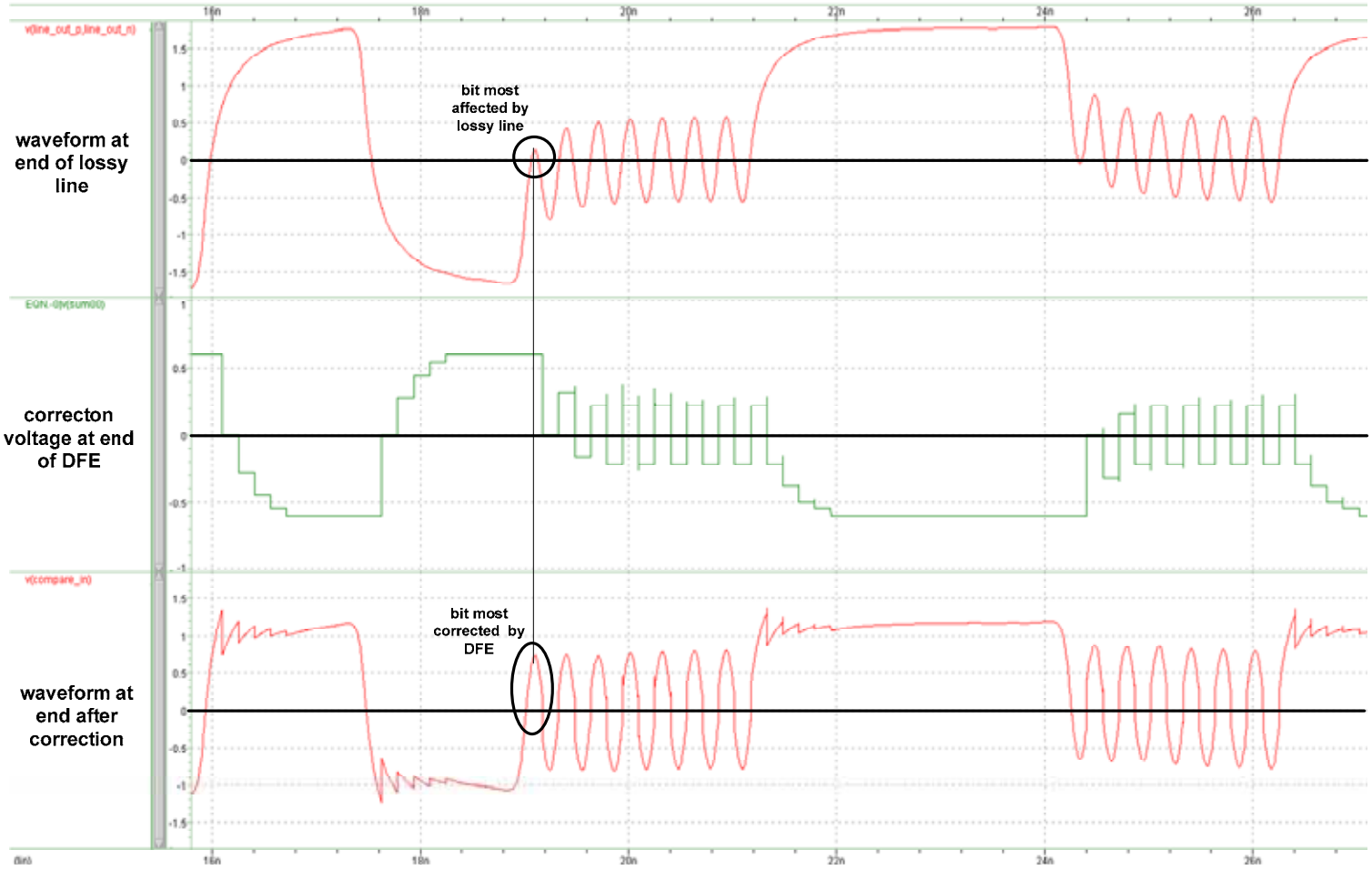
- DFE is non linear system of equalization
- DFE works by actively shifting the incoming signal based on history of received data
- DFE removes signal energy that leak from one bit to the following bit.
  - DFE effectively cancels out post cursor ISI.
- The advantage of DFE is to boost the power of high frequency component of received data with out increasing noise power

Basic DFE structure

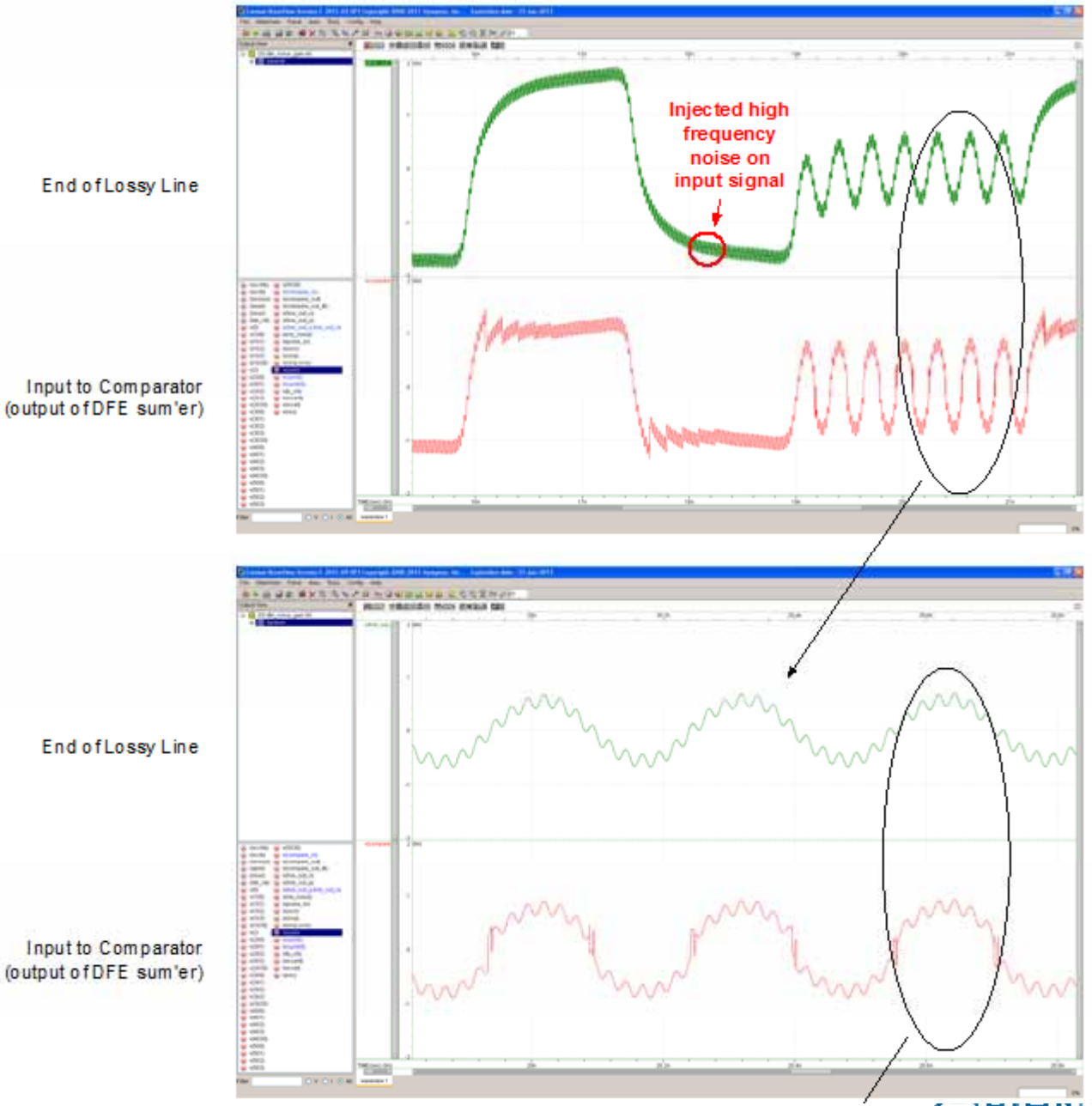


- The signal at sampler at time T_0
  - $V(T_0) = Vin(T_0) - C1*D(T_{-1}) - C2*D(T_{-2}) - C3*D(_{T-3})\ldots.$
    - $Vin(T_0)$ is output of equalizer at time T_0
    - $D(T_n)$ is quantified signal at time T_n
    - The component $C1*D(T_{-1})$ compensates for post cursor signal at time T_{-1} leaking into signal at T_0
- In theory you would need infinite taps to compensate all post cursor effects
  - But post cursor effect decreases exponentially. So, finite number is sufficient

DFE Simulation

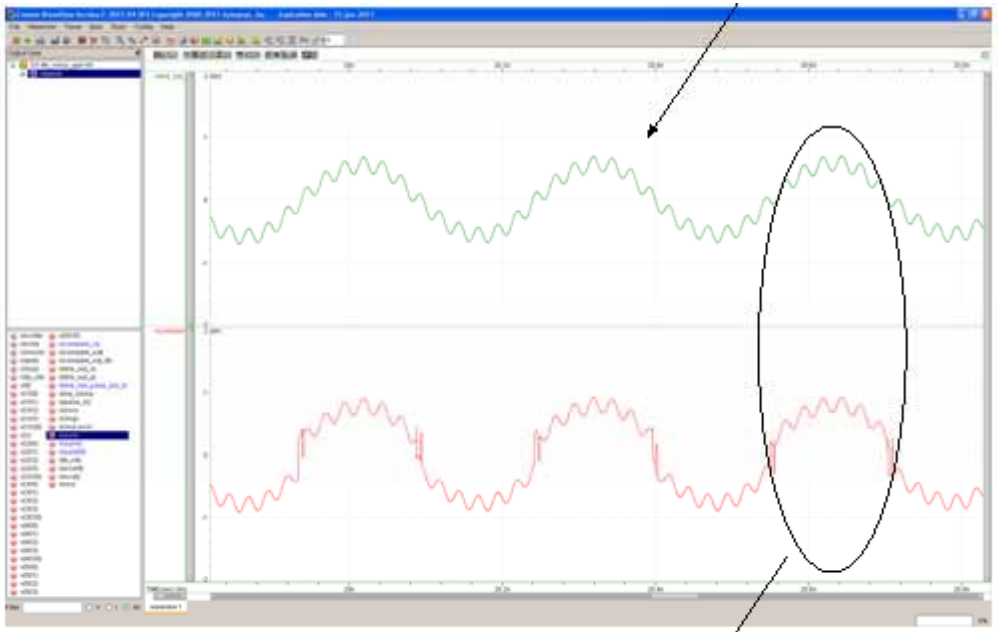


DFE Simu



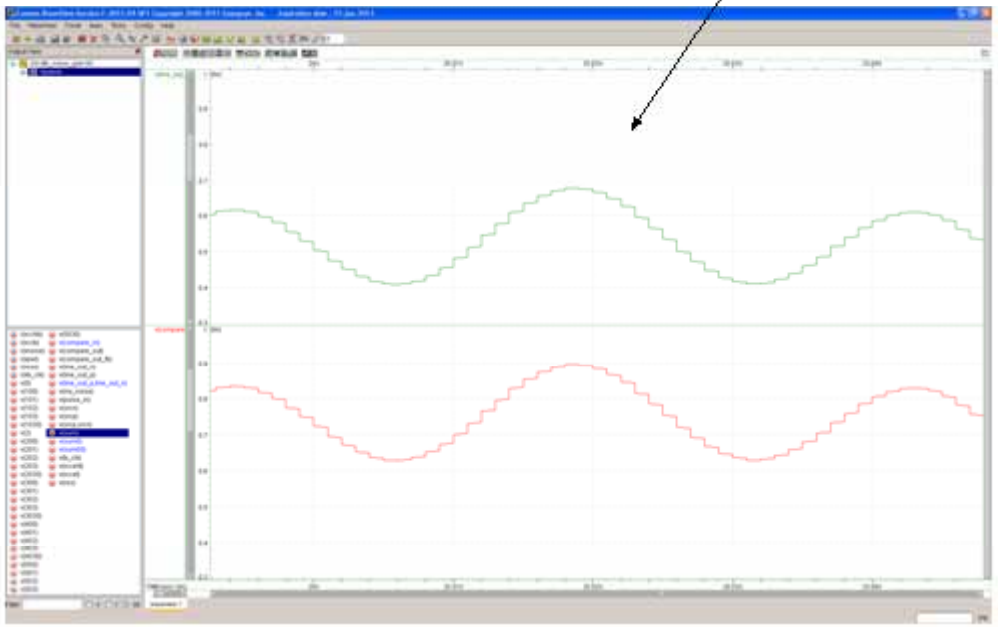
End of Lossy Line

Input to Comparator
(output of DFE sum'er)



End of Lossy Line

Input to Comparator
(output of DFE sum'er)



Peak-to-peak noise is exactly the same amplitude before and after DFE circuit

therefore

there is no gain being applied at high frequencies

Link tuning – Tx pre emphasis role

- As backplane loss reduces high frequency content, a positive 1st post tap and negative pre tap boosts high frequency content
  - Also improves rise/fall times of Tx buffer output data
- Using Tx Pre emphasis with DFE
  - DFE compensates for post cursor ISI effects, so negative pre tap is required to offset pre cursor effects of ISI
  - High values of 1st post tap and negative pre tap reduces Vod level of bits after transition. DFE need comfortable level of eye envelope. So, positive second post tap should be used where losses are high(in addition to 1st post tap and –ve pre tap)
  - DFE require a launch voltage of 800mv to 1000mv(corresponds to ~40 to 50 on Tx Vod setting)
- The only disadvantage of Tx pre emphasis is in systems with high coupling, cross talk amount would be increased
  - Because high frequency content on signals is increased with pre emphasis
  - In other words, if cross talk is not major concern, Tx pre emphasis is very effective

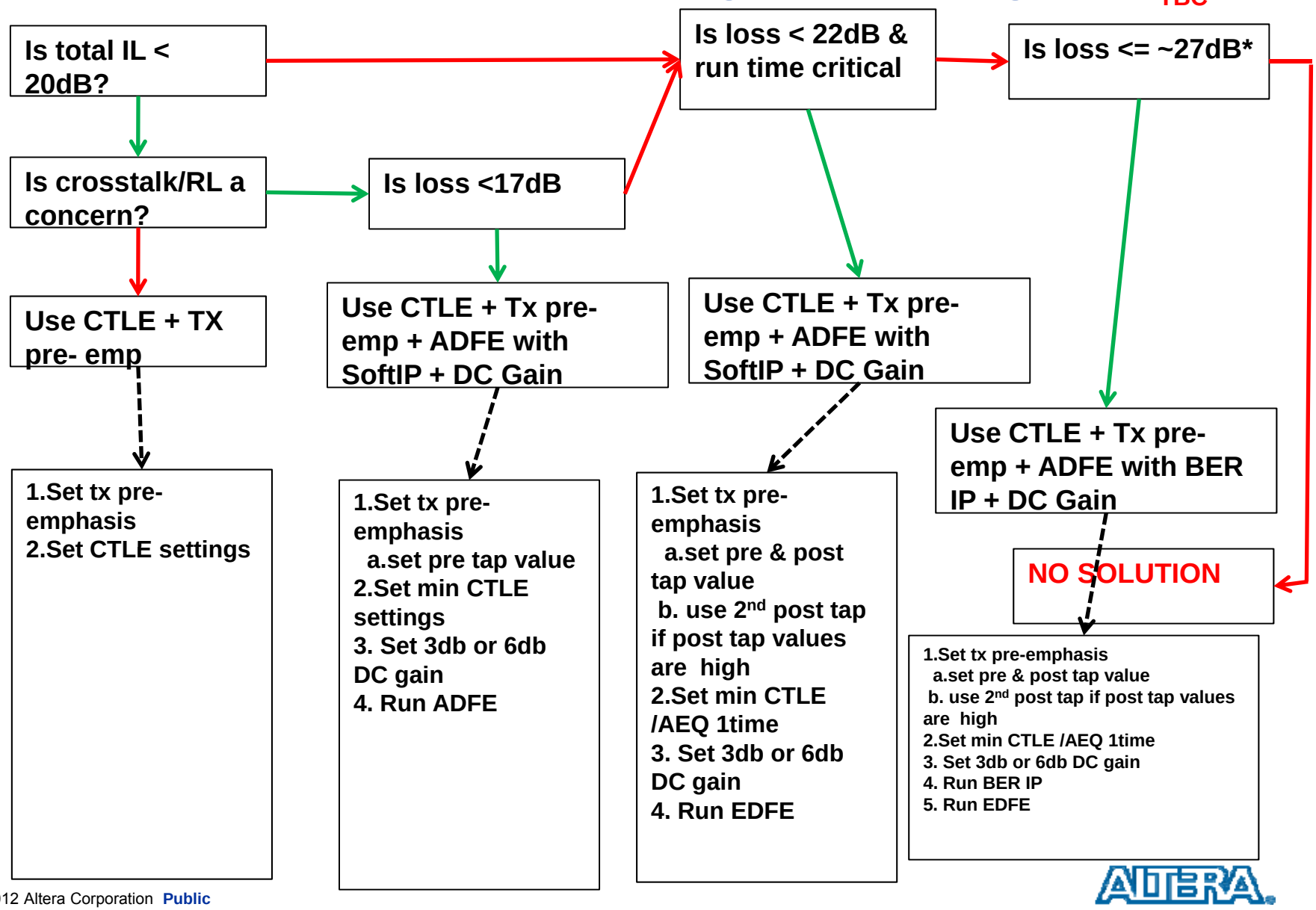
Link Tuning & Rx Equalization

- Rx Equalization is well suited for real time adaptation and equalized signal is readily available for Rx
  - Use Rx equalizer when backplane loss is less than 17db
- In case tighter power requirements, use either one time adaptation or Manual CTLE
  - Disadvantage of these modes is the performance impact with VT changes
  - Use adaptive mode or provide enough margin to overcome this disadvantage
- CTLE can boost high frequency noise in addition to signal content – poor signal to noise ratio
  - In multi lane systems with high cross talk, use lower CTLE settings in order not to amplify noise/crosstalk
- If backplane loss is high and cross talk is high, DFE is better choice of equalization

Link Tuning & DFE

- DFE is well suited in systems with high cross talk/high reflections
  - Use higher DC gain to use less equalizer values and more DFE (better SNR)
- Use DFE + CTLE if total loss of system is above 20dB
  - Recommended mode of operation
    - continuous AEQ + one time/Manual DFE
    - One time AEQ/Manual CTLE + continuous DFE(ADFE)
  - Three adaptive loops(CDR, DFE, CTLE) cannot be used at the same time to avoid loop interaction BW need to be separated by decade
- Continuous DFE(ADFE) can increase Transceiver power by 2X
  - One time adaptation helps in achieving additional power savings and reduced effort compared to manual DFE

Flow chart for Link tuning – in progress



Signal Integrity Simulation

10GBASE-KR Backplane Electrical

- TX
  - Eye mask
- Channel
  - Channel description
  - Insertion loss
  - Return loss
- RX
  - Jitter Tolerance
  - Return loss
- System
  - < BER = 1E-12

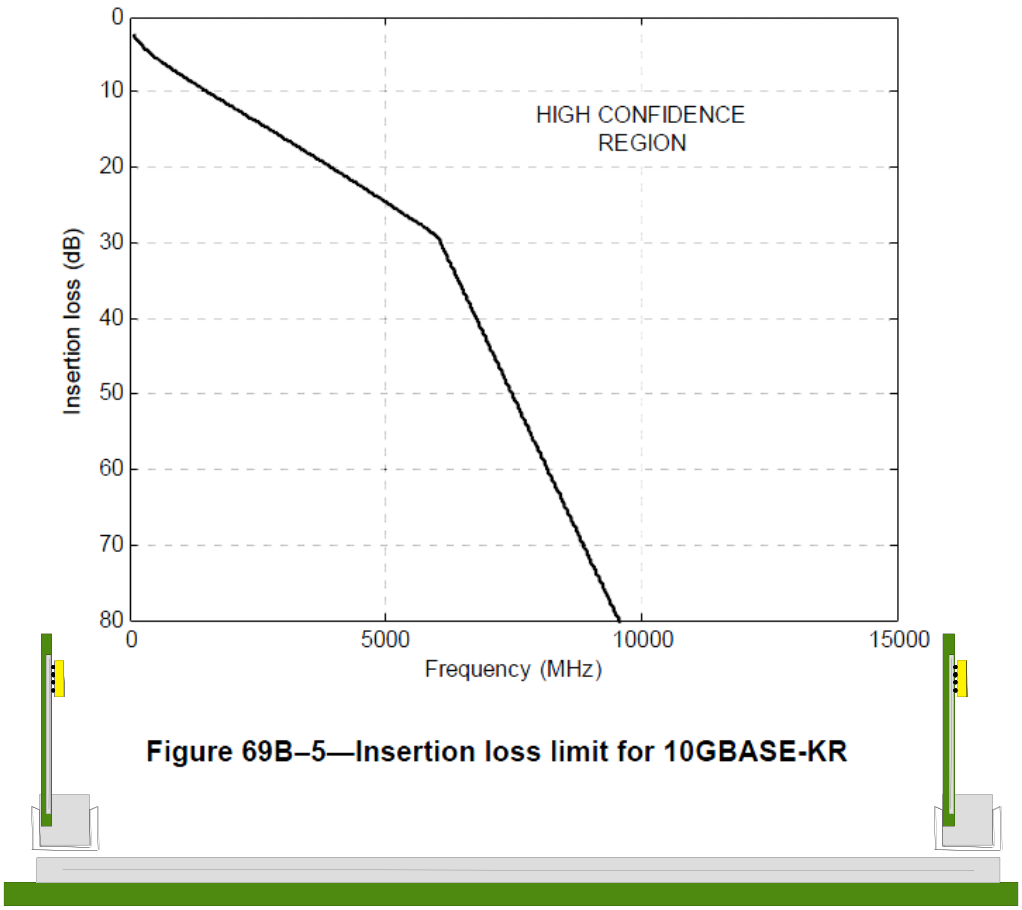
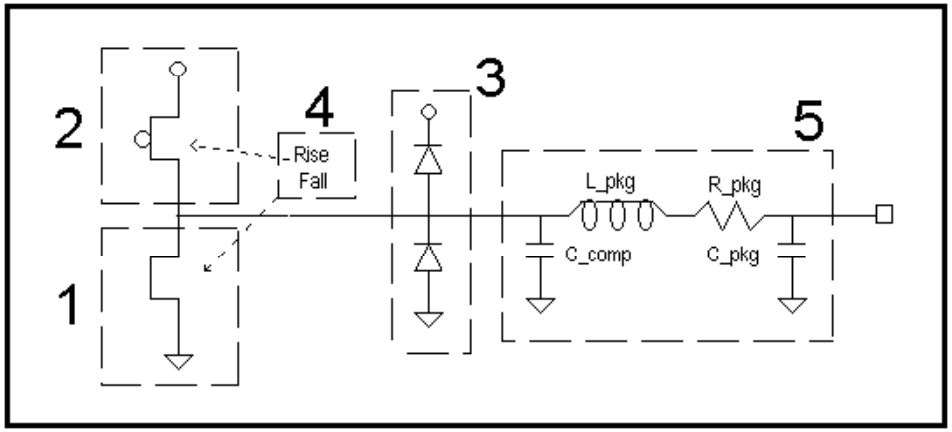
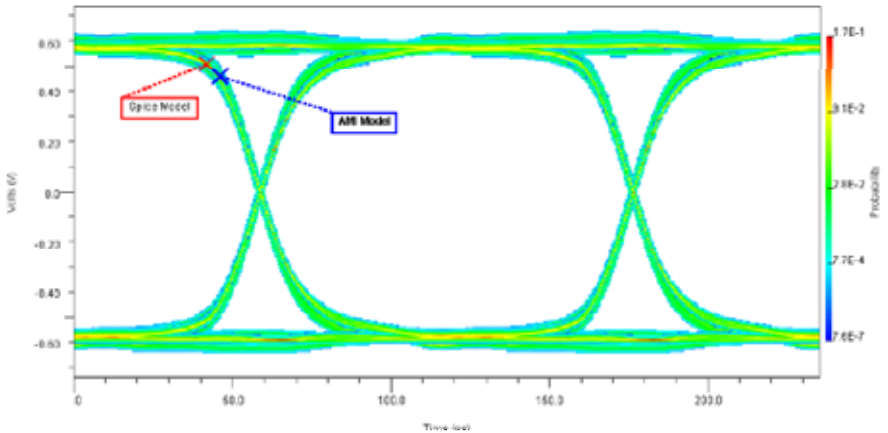


Figure 69B–5—Insertion loss limit for 10GBASE-KR

Transceiver Simulation Models

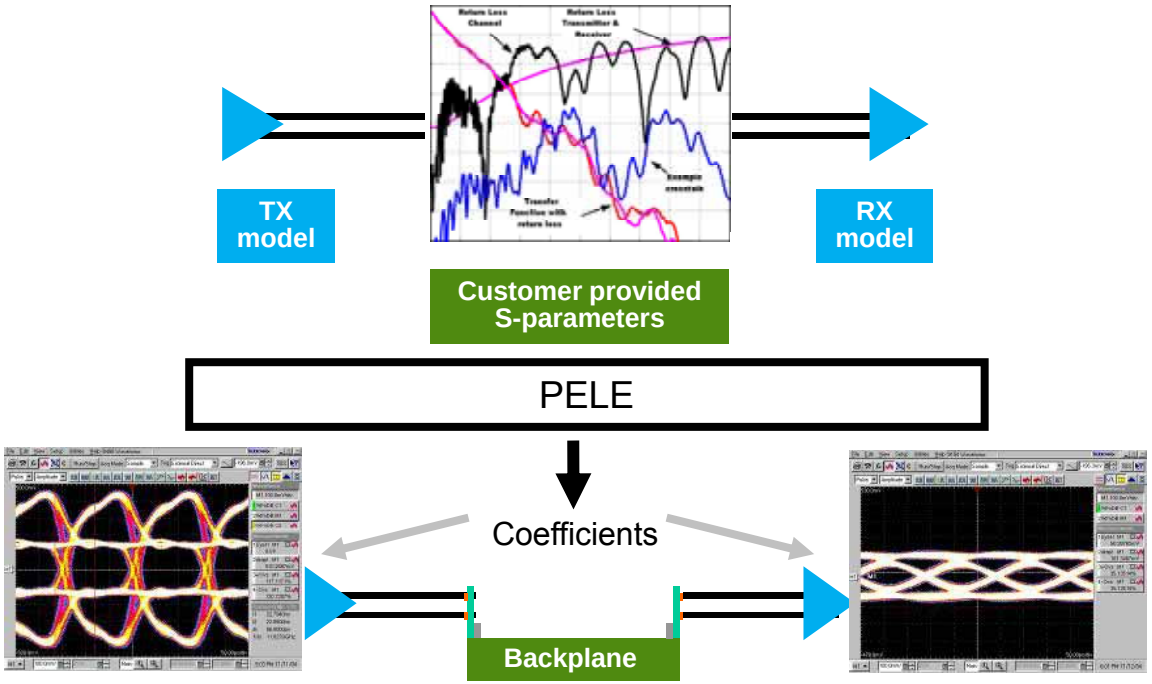
- Altera's suite of transceiver design tools
  - Evaluate performance in custom application
  - Run "What if" simulations for early analysis
  - Create design constraints in layout and design
  - Run in-system verification for board bring-up and live debug





- HSPICE full circuit models
- IBIS-AMI behavioral models
  - Fast simulation
  - Analog and algorithmic description of all major transceiver components
  - Analysis of millions of bits

PELE – Pre-emphasis/Equal Link Estimator



- **Optimize** the equalization coefficients for the transceiver
- Early estimate of link performance
- Inputs: Channel / settings

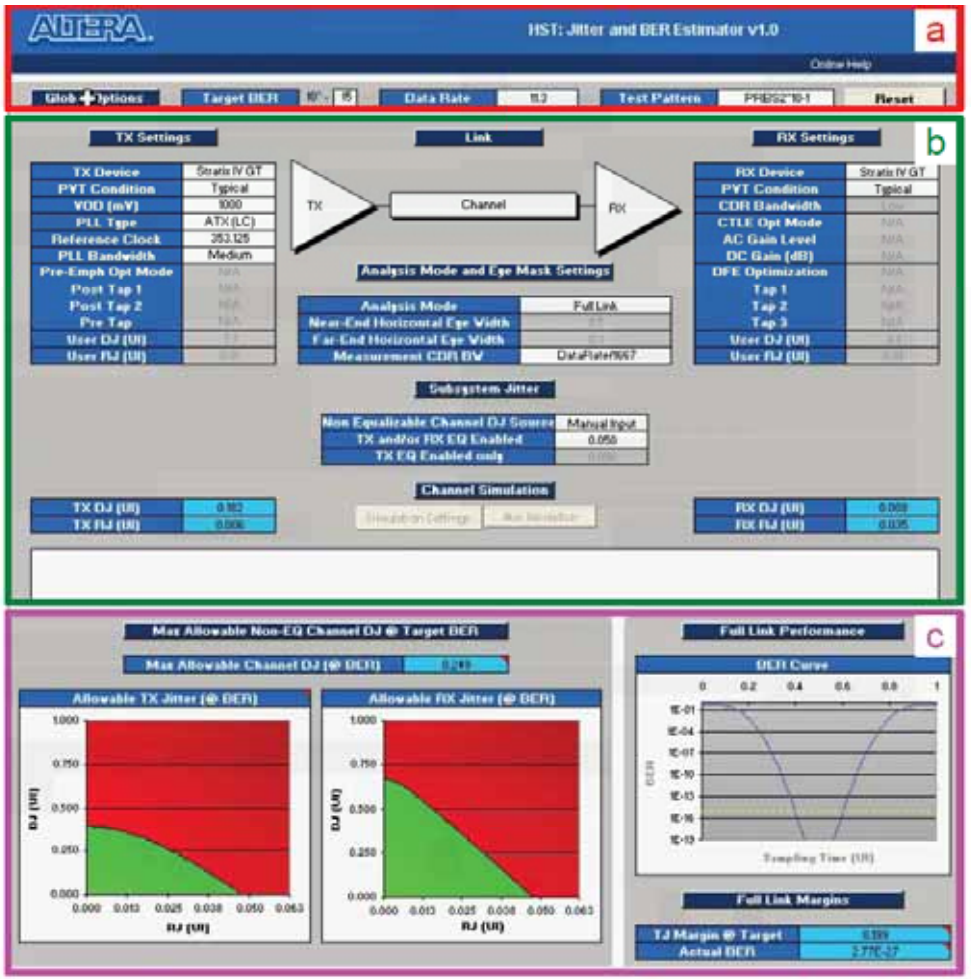
Simulation Model Comparison

	HSPICE	IBIS-AMI	PELE
Accuracy	High	High/medium	Medium
Time consumption	Hours to days	Minutes to hours	Minutes
Corner model availability	Full	Full	TT/NormV/85C
Flexible data inputs	Yes	Yes	PRBS-7/10
Link to other devices	Yes	Yes	No
EDA tool requirement	Synopsys HSPICE	Yes, independent	NA
Simulation platform requirement	64-bit Linux, 8 GB memory	EDA-tool dependent	32-bit system, 1 GB memory

HST Jitter and BER Estimator

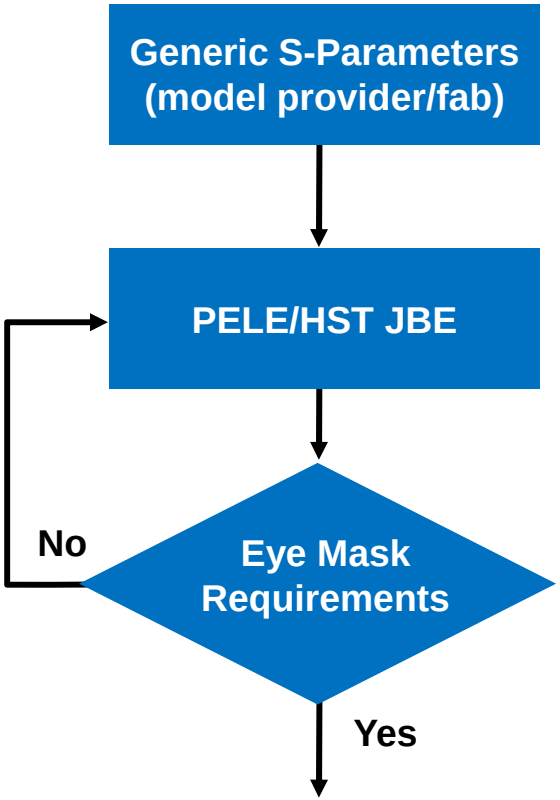
- Custom characterization
  - Quickly and accurately **estimate** system link reliability (BER)
  - Utilize customer-specific channel (S-Parameter)
  - Run statistical analysis using characterization data
- Margin analysis
  - TX
  - RX
  - Channel
- Reduction of system cost
  - Cost-effective alternatives for the same system performance

Currently Available for Stratix IV and V FPGAs



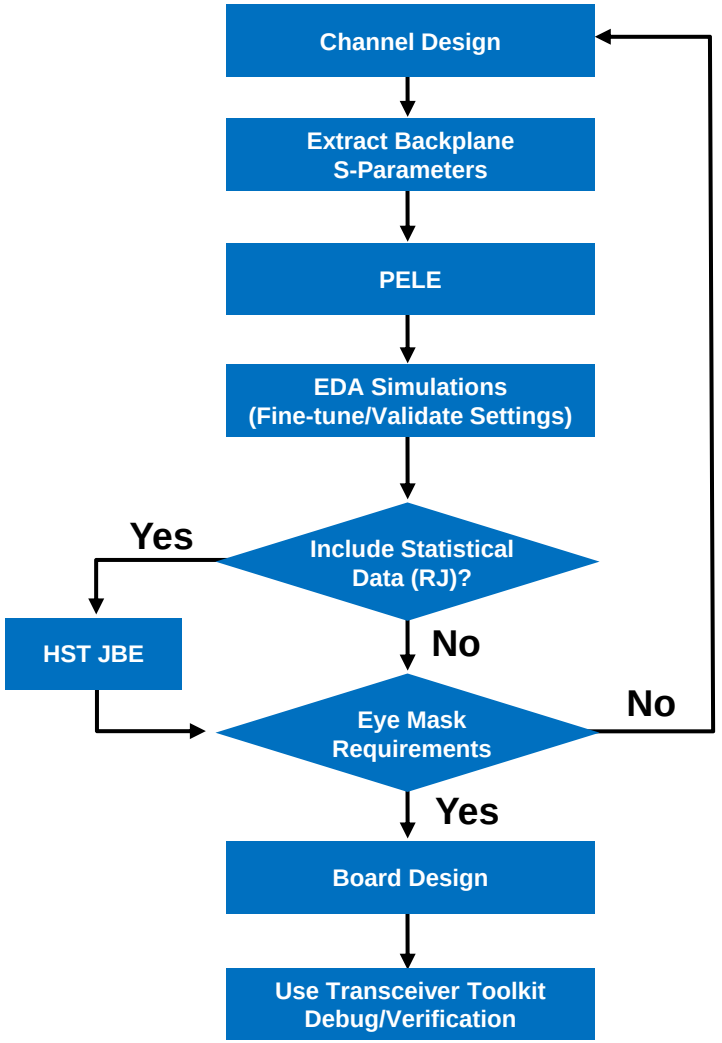
Link Simulation Flow – Early Stage

- Use generic S-parameter file
  - From backplane model provider, EDA simulation tool extraction or VNA measurement
- Use PELE/JBE to see if the selected device compensates channel losses using pre-emphasis or equalization, or both
- Check to see if the eye opening meets the protocol requirements or device requirements



Link Simulation Flow – Design Phase

- Channel design
  - Further analysis
- Pre-emphasis and/or equalization settings selection
- Fine tune/validate settings
  - HSPICE
  - IBIS-AMI behavioral models
- Use JBE to include the statistical data
- Use the transceiver toolkit to verify and debug

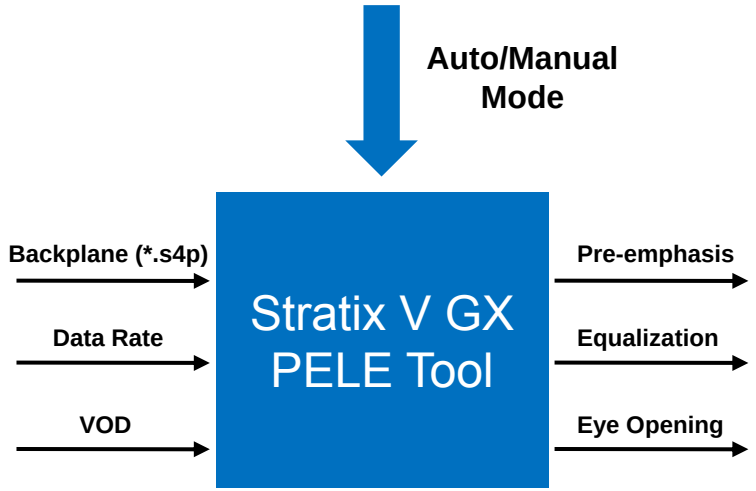


PELE Configuration

- Standalone mathematical tool
  - Requires MATLAB run-time library
- Inputs
  - Data rate
  - V_{OD}
  - Backplane
  - TX pre-emphasis setting
  - RX equalization setting
    - AC gain (CTLE)
    - DC gain
    - DFE
- Outputs
  - Deterministic eye opening at TX, RX, and post equalization
  - Optimal pre-emphasis and equalization setting

Optimization Method	TX Pre-emphasis	RX CTLE
1	Manual	Auto
2	Auto	Auto
3	Auto	Manual
4	Manual	Manual

Optimization Method	DFE
1	Disable
2	Auto
3	Manual

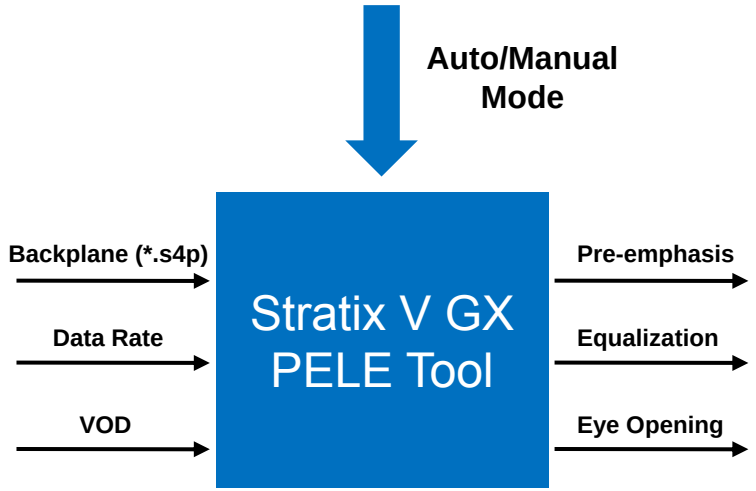


PELE Configuration

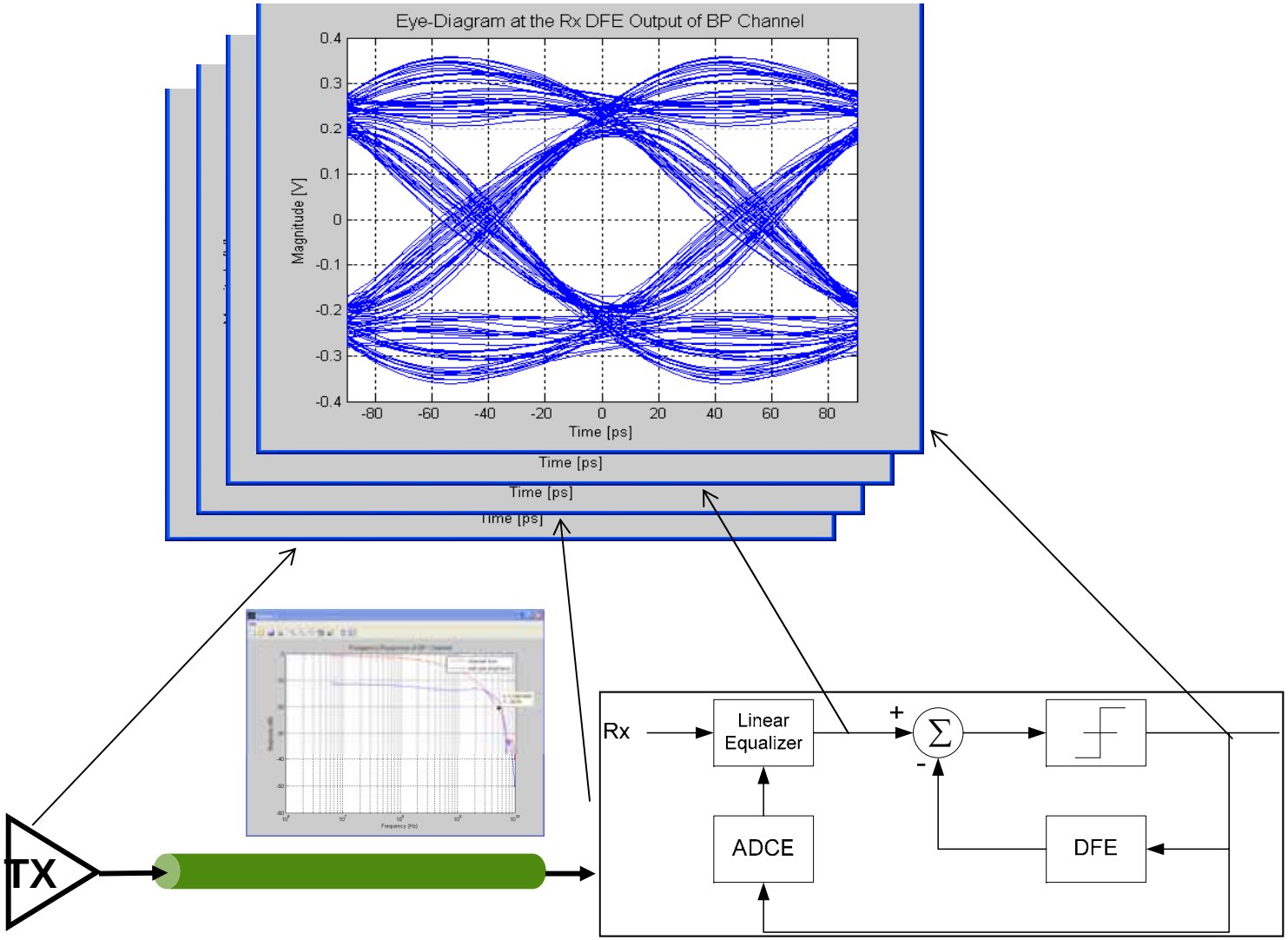
- Standalone mathematical tool
  - Requires MATLAB run-time library
- Inputs
  - Data rate: 10.3125 Gbps
  - V_{OD} : 1000 mV
  - Backplane: "30inches_2connectors_backplane.s4p"
  - TX pre-emphasis setting: Auto
  - RX equalization setting
    - AC gain (CTLE) : Auto
    - DC gain: 4 (0-8 dB)
    - DFE: Auto
- Outputs
  - Deterministic eye opening at TX, RX, and post equalization
  - Optimal pre-emphasis and equalization setting

Optimization Method	TX Pre-emphasis	RX CTLE
1	Manual	Auto
2	Auto	Auto
3	Auto	Manual
4	Manual	Manual

Optimization Method	DFE
1	Disable
2	Auto
3	Manual

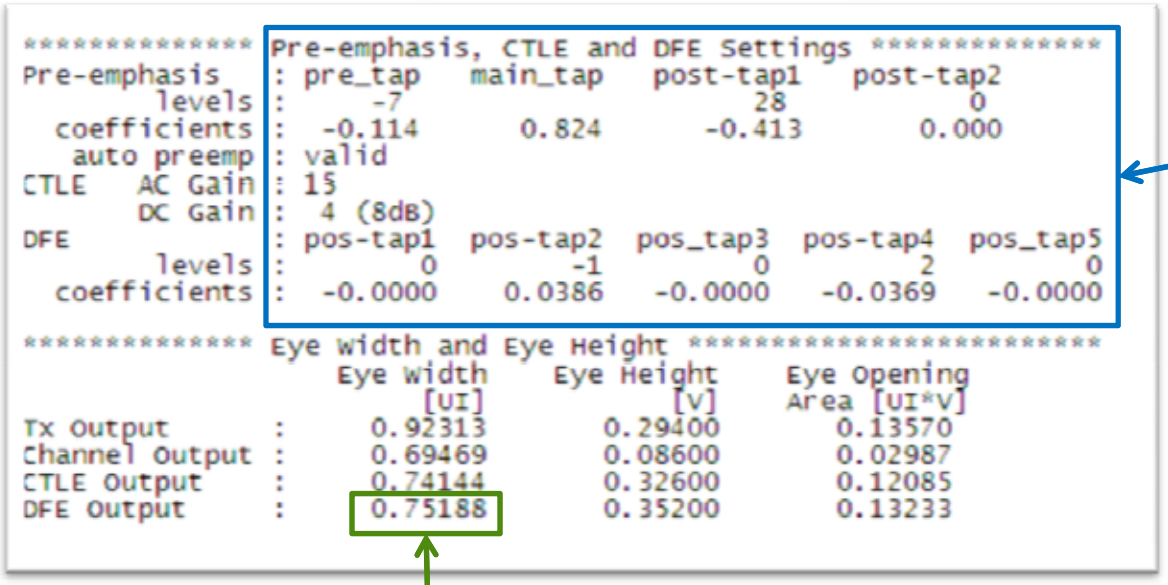


PELE Simulation (30” link @10.3125G)



PELE Simulation Output

- Refer to Stratix V user guide on PELE instructions
- PELE output results:



```
**************** Pre-emphasis, CTLE and DFE Settings ***************
Pre-emphasis    : pre_tap    main_tap    post-tap1    post-tap2
         levels :     -7                      28            0
   coefficients :  -0.114       0.824      -0.413        0.000
    auto preemp : valid
CTLE    AC Gain : 15
        DC Gain :  4 (8dB)
DFE             : pos-tap1  pos-tap2  pos_tap3  pos-tap4  pos_tap5
         levels :        0        -1         0         2         0
   coefficients :  -0.0000    0.0386   -0.0000   -0.0369   -0.0000

**************** Eye Width and Eye Height ***************************
                      Eye Width     Eye Height     Eye Opening
                          [UI]            [V]     Area [UI*V]
Tx Output         :     0.92313         0.29400         0.13570
Channel Output    :     0.69469         0.08600         0.02987
CTLE Output       :     0.74144         0.32600         0.12085
DFE Output        :     0.75188         0.35200         0.13233
```

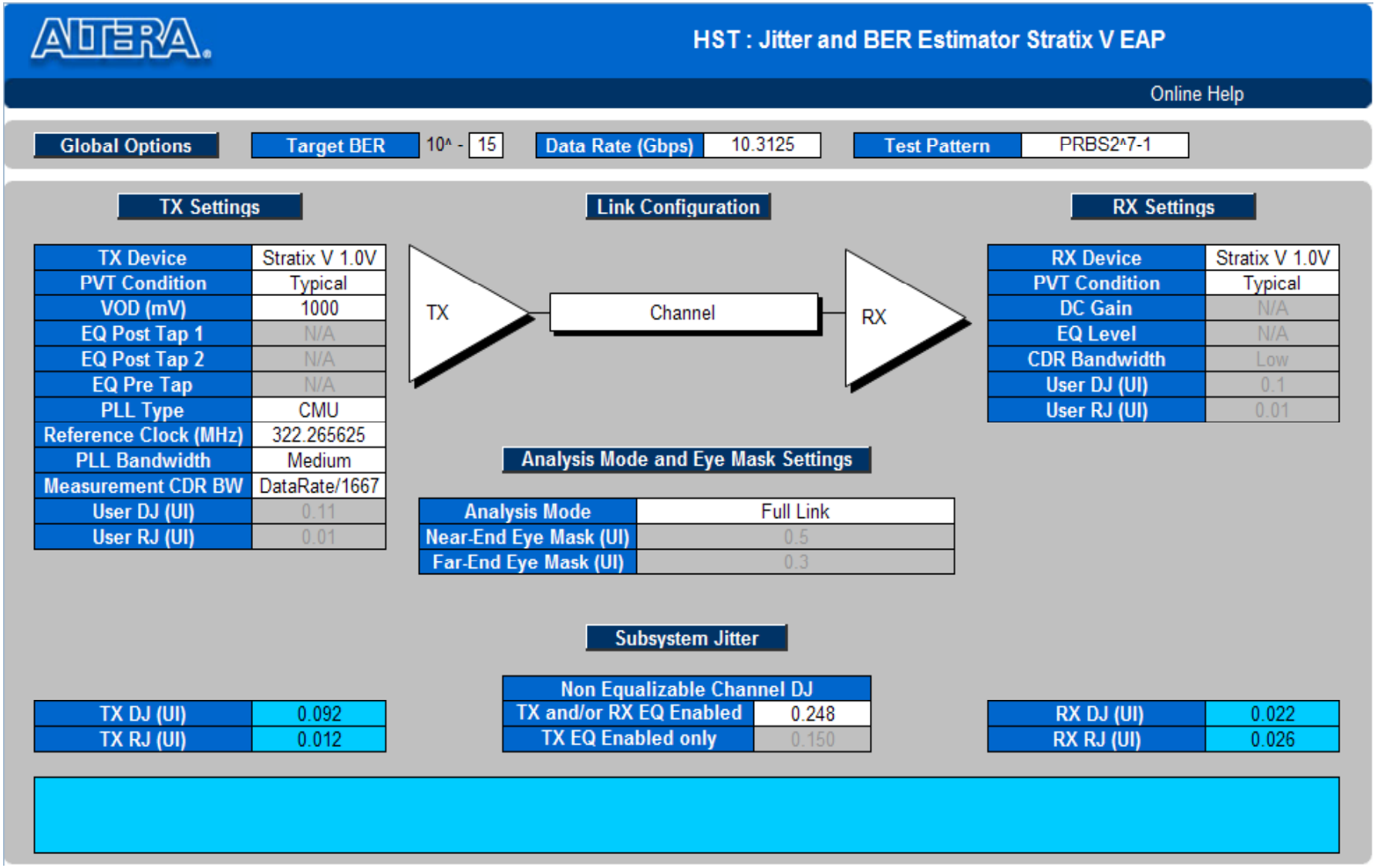
Starting point for optional simulation analysis

0.75 UI = deterministic eye opening
1- 0.75 UI = **0.25 UI** = non compensated jitter

System Performance

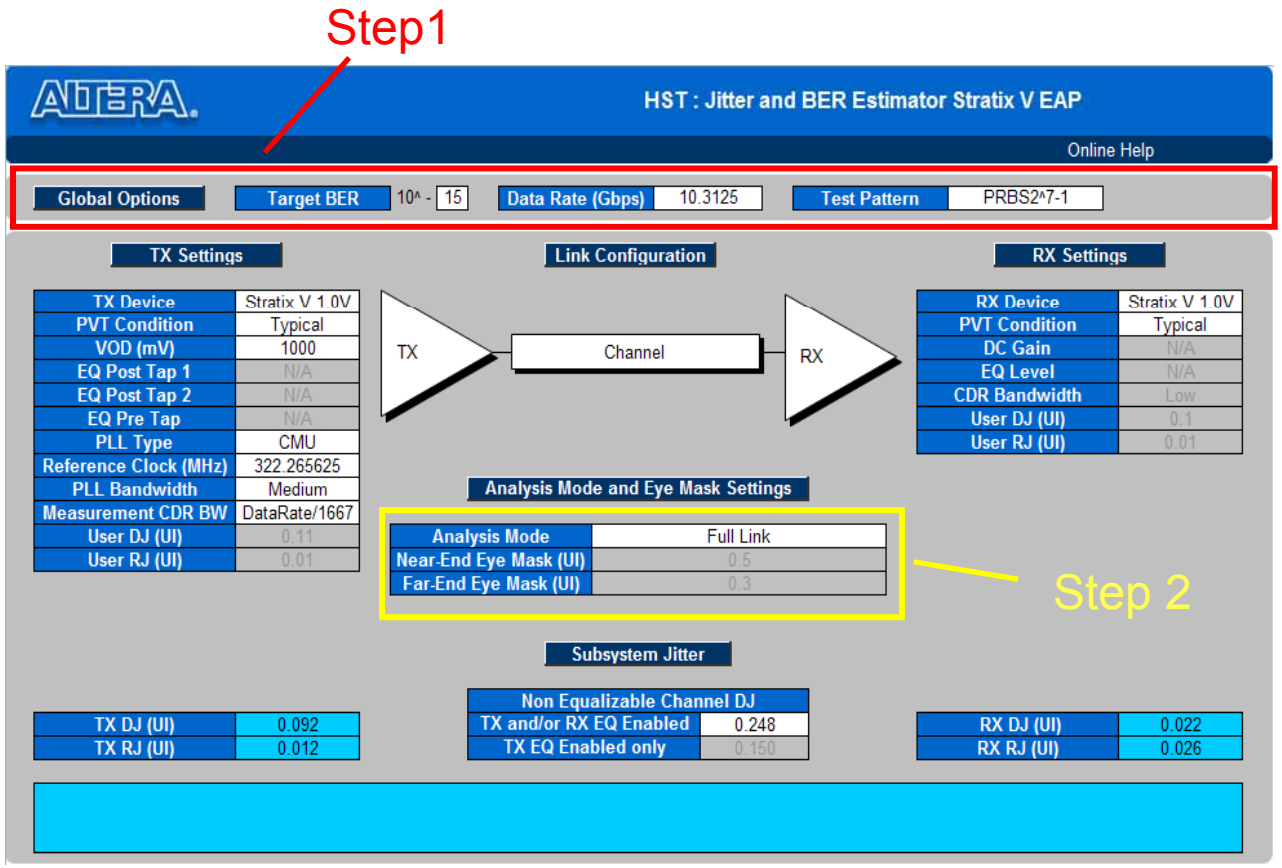
- PELE analysis is a deterministic simulator
- Jitter and BER Estimator (JBE) incorporates random jitter components of transmitter and receiver through characterized data
  - Early version (EAP) of Stratix V JBE is based on Stratix IV data
  - Final version will incorporate actual silicon measurement
- JBE will determine Bit Error Ratio performance of link

Jitter and BER Estimator Tool



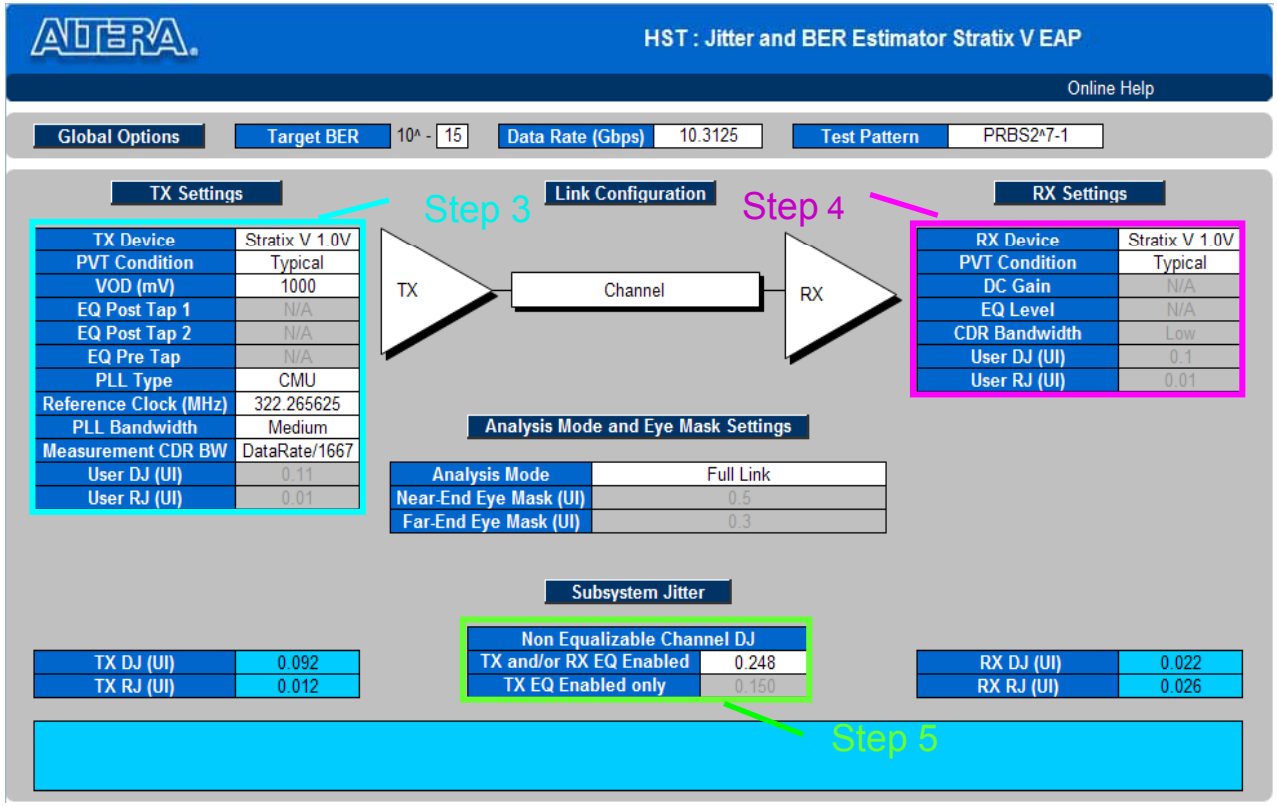
JBE Configuration Steps

1. Setup global parameters
   - Target BER
   - Data Rate (Gbps)
   - Test Pattern
2. Link configuration
   - Analysis mode selection/eye mask setup
   - Options: Full Link, TX, RX



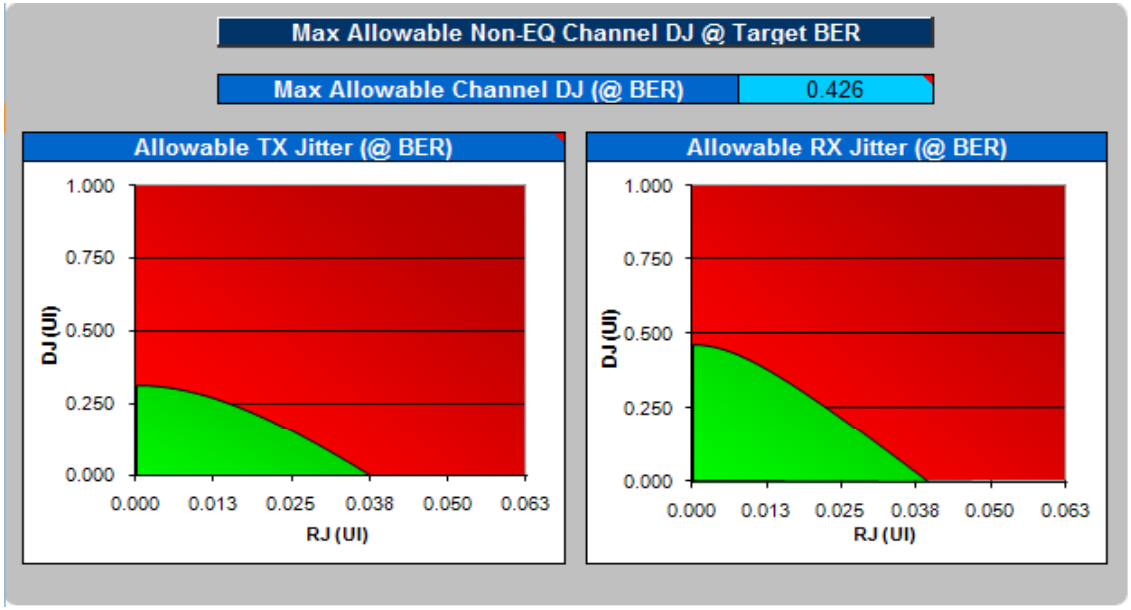
JBE Configuration Steps

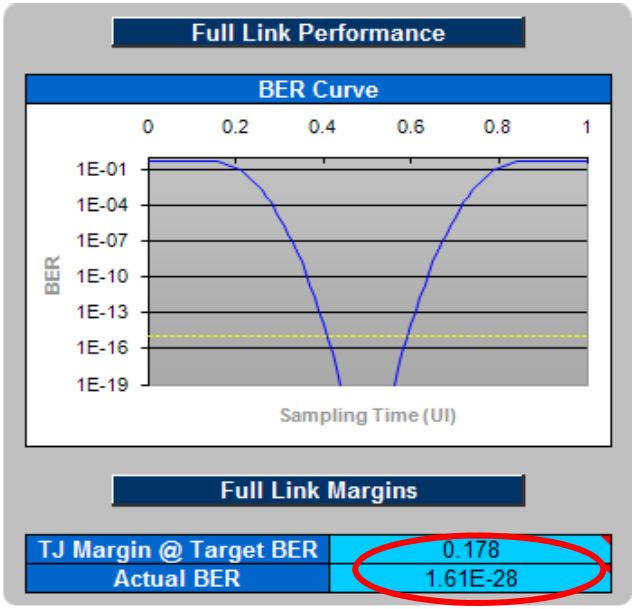
3. Configure TX settings
4. Configure RX settings
5. Input the non-equalized channel DJ from PELE simulation output
   - "1 – Eye opening" post equalization
   - May add margin to this number to account for cross-talk



Link Analysis: Full Link Mode

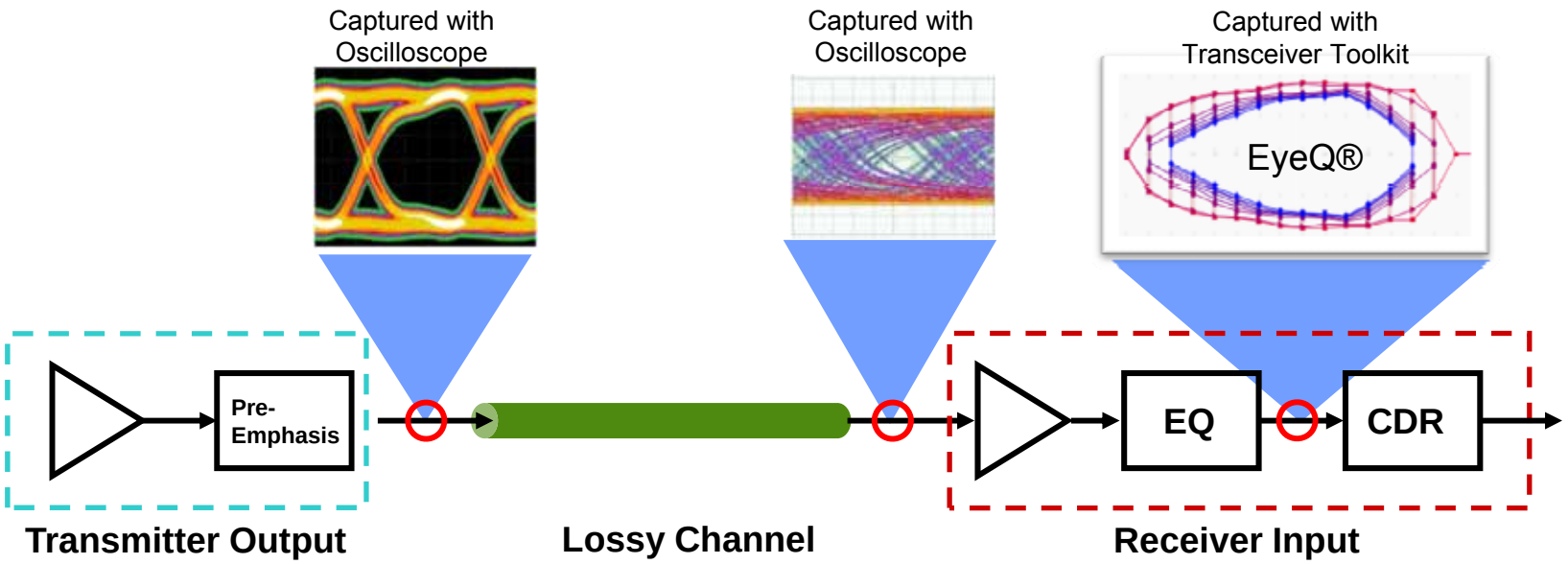
- Full link simulation shows that the link meets the BER target of 10^{-15}
- Margin analysis





EyeQ Enables On-Chip Analog Debug

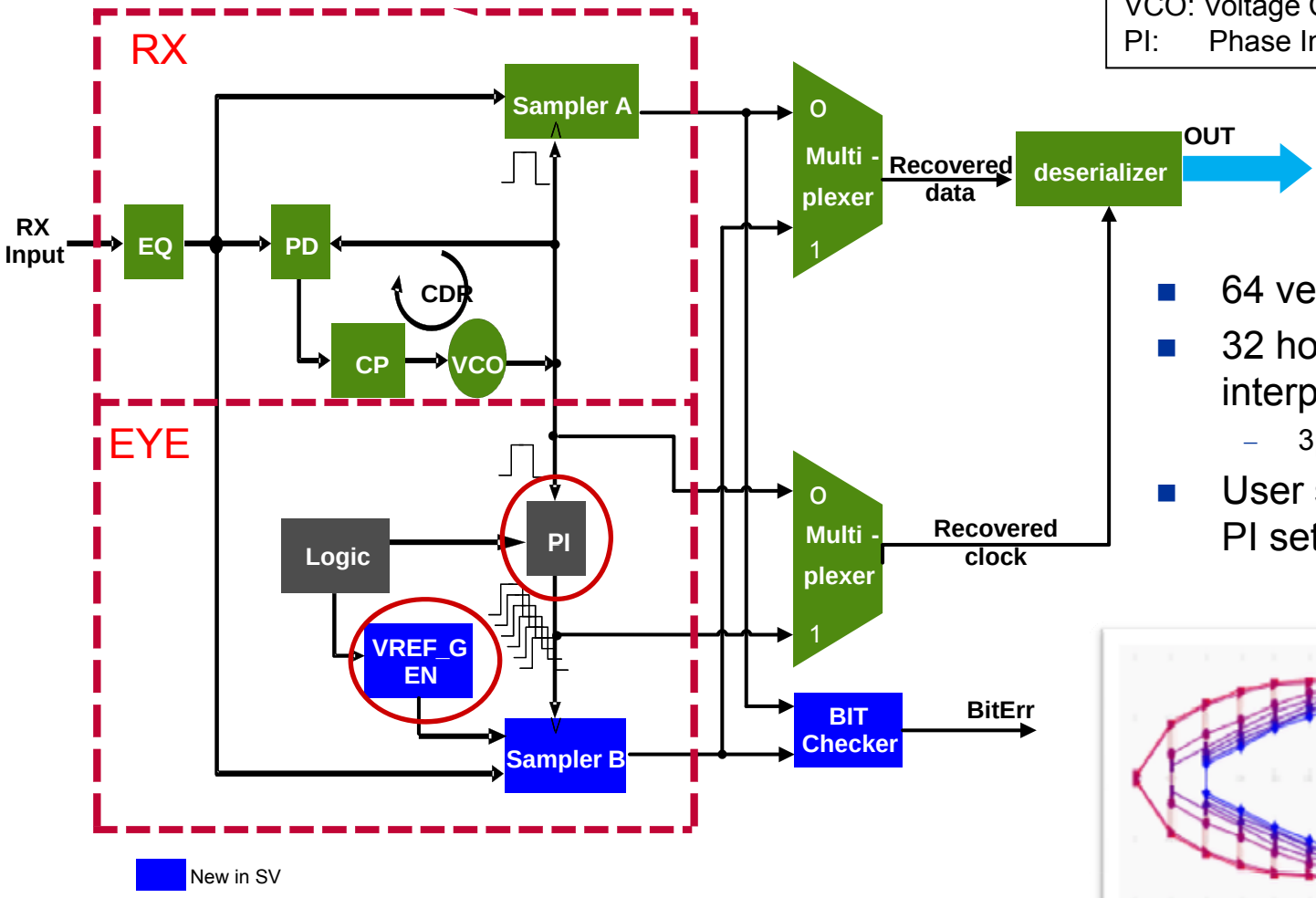
- EyeQ is a circuit that enables users to “probe” the CDR input
- Users can view the eye opening at the CDR input
- Users can see the correlation of EQ settings to eye opening
- EyeQ enables users to use the debug feature with live traffic



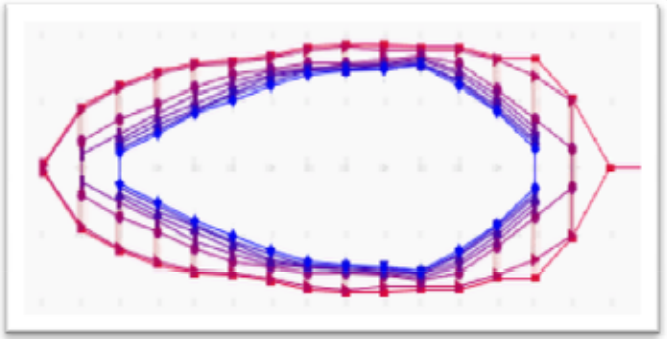
Minimize board bring up / debug time with Dynamic reconfiguration and EyeQ

EyeQ Circuit

EQ: Equalizer
PD: Phase Detector
CP: Charge Pump
VCO: Voltage Controlled Oscillator
PI: Phase Interpolator



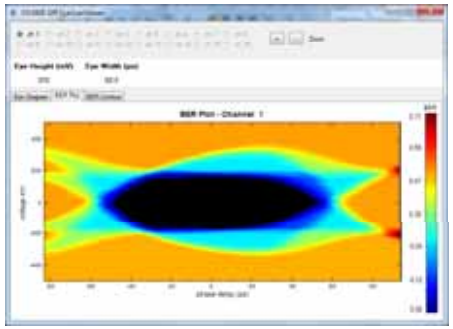
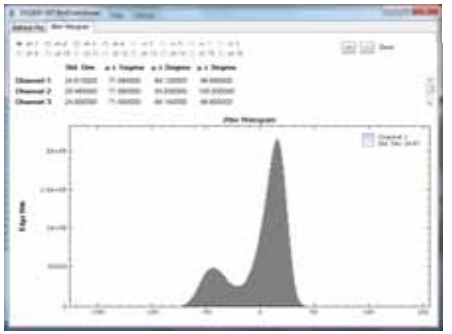
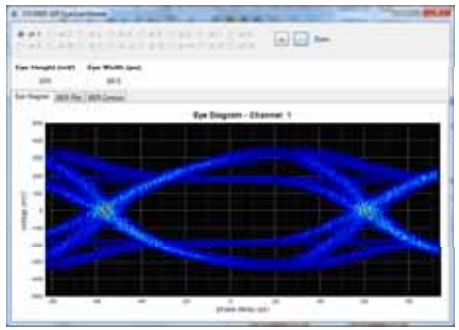
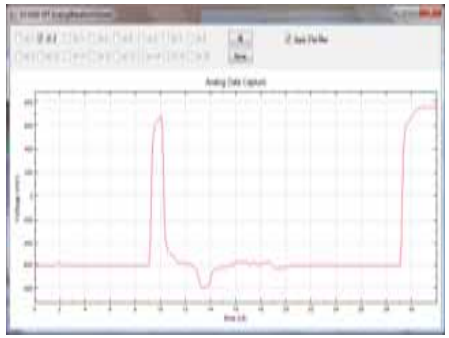
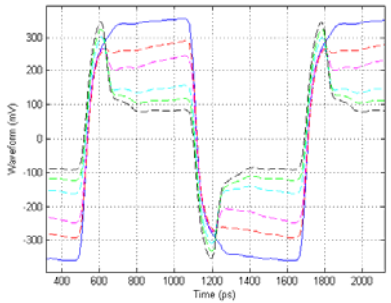
- 64 vertical threshold levels
- 32 horizontal phase-interpolator steps
  - 3ps / steps (@ 10Gbps)
- User selectable threshold / PI setting

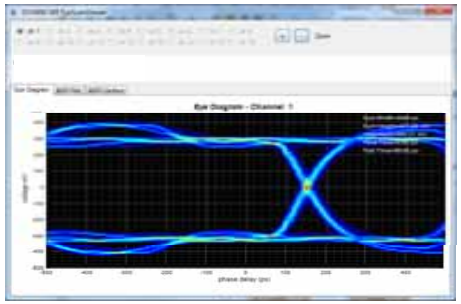


More Sophisticated Tools

- DFT
  - Eye Diagram (Persistence mode)
  - Eye Diagram (BER plot)
  - BER Contours
  - True Oscilloscope Capability
  - Jitter Measurement
  - Edge rates & Pre-emphasis measurements
  - and more …

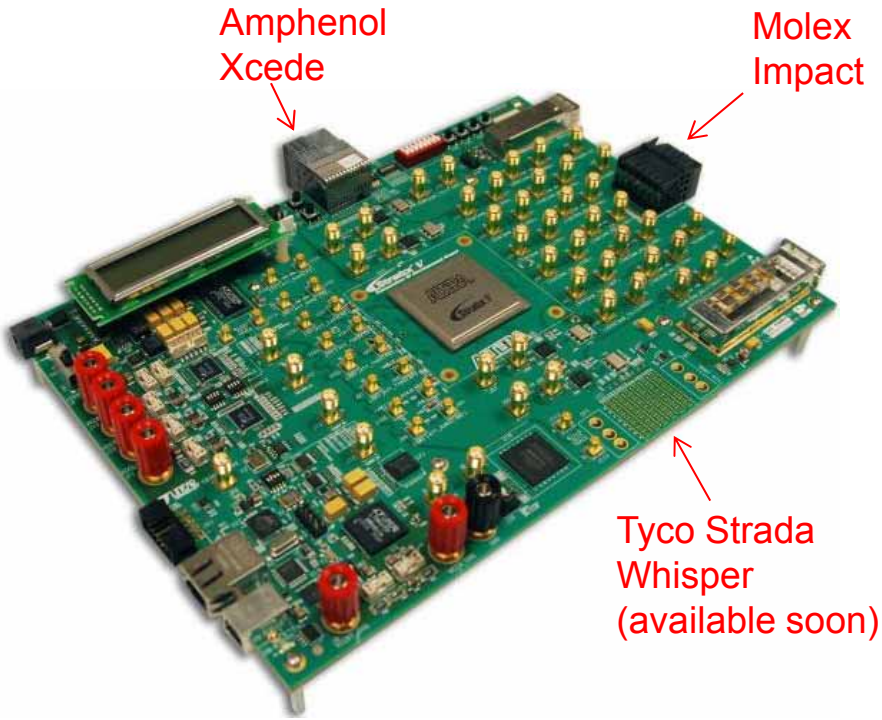
DFT



SI Board Backplane Support

- Designed with mating BP connector from each vendor
  - Amphenol, Molex, Tyco
- Can directly plug into vendor's standard backplanes
- Characterize performace at 10G & 12.1G
- Evaluate performance at higher data rates (ie, 14.1G)



Stratix V SI Board

High-Speed Transceiver Toolkit

3.0

High-Level Challenges – Debug Phases

Phase 1:

Check if data is passing through transceiver channel during PCB bring-up

- Test BER by generating and verifying industry-standard PRBS data patterns
- Dynamically reconfigure the pre-emphasis and equalization settings
- Analyze the link by using Stratix V EyeQ feature to find optimal settings

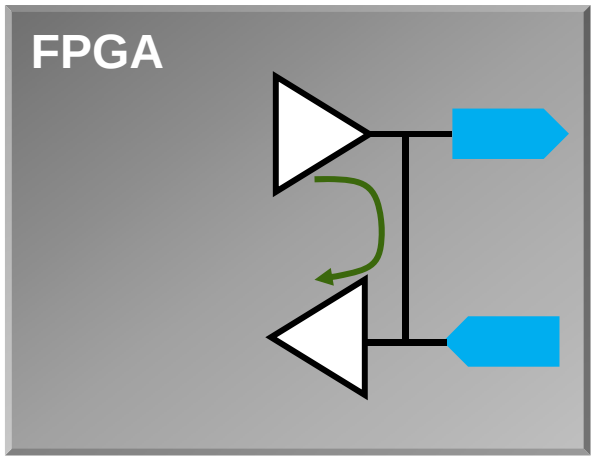
Phase 2:

Perform in-system or mission-mode link analysis with real-time data in an operating hardware system

- Integrate completed user design for full system testing
- Verify BER by using Stratix V bit comparator

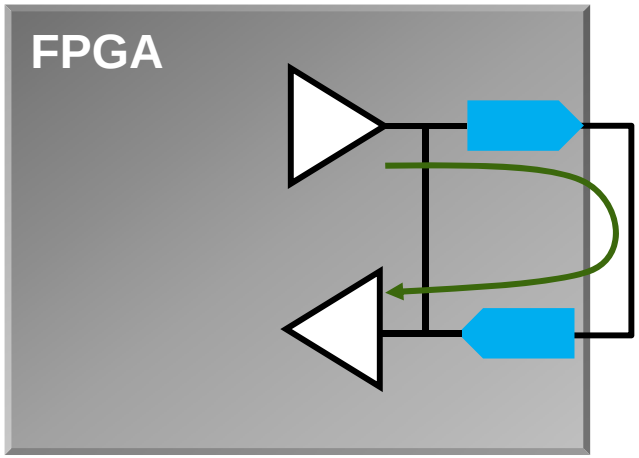
Debug Phase 1 – Q & A (1/3)

- Can I see data in the transceiver channels?
  - Perform internal serial loopback
  - TX and RX loopback directly within the silicon



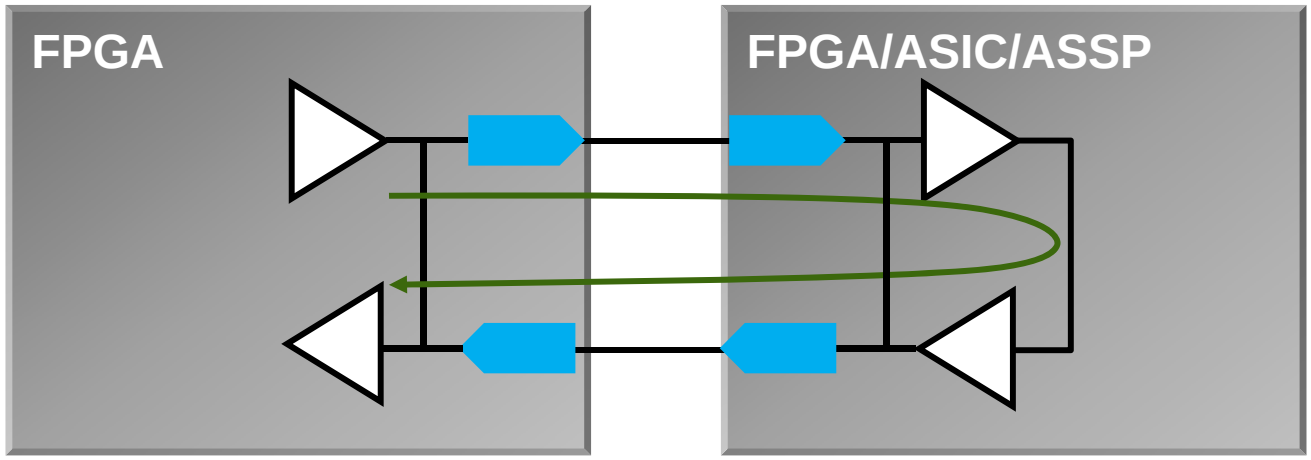
Debug Phase 1 – Q & A (2/3)

- Can I see data drive out of the FPGA TX pin and the same data drive back into the RX pin, and vice versa?
  - Perform external loopback
  - TX and RX drive in and out of the FPGA pins



Debug Phase 1 – Q & A (3/3)

- Can I see data drive out of the FPGA TX pin and the same data received in another device or external test equipment, and vice versa?
  - Perform reverse serial loopback
  - TX from device 1 drives RX in device 2 and vice versa
  - External test equipment drives RX in device 1 and vice versa



Debug Phase 2 – Q & A

- Can I see data running in the completed hardware system?
  - Run final test by using the fully integrated user design

Transceiver Toolkit Overview

What is the Transceiver Toolkit?

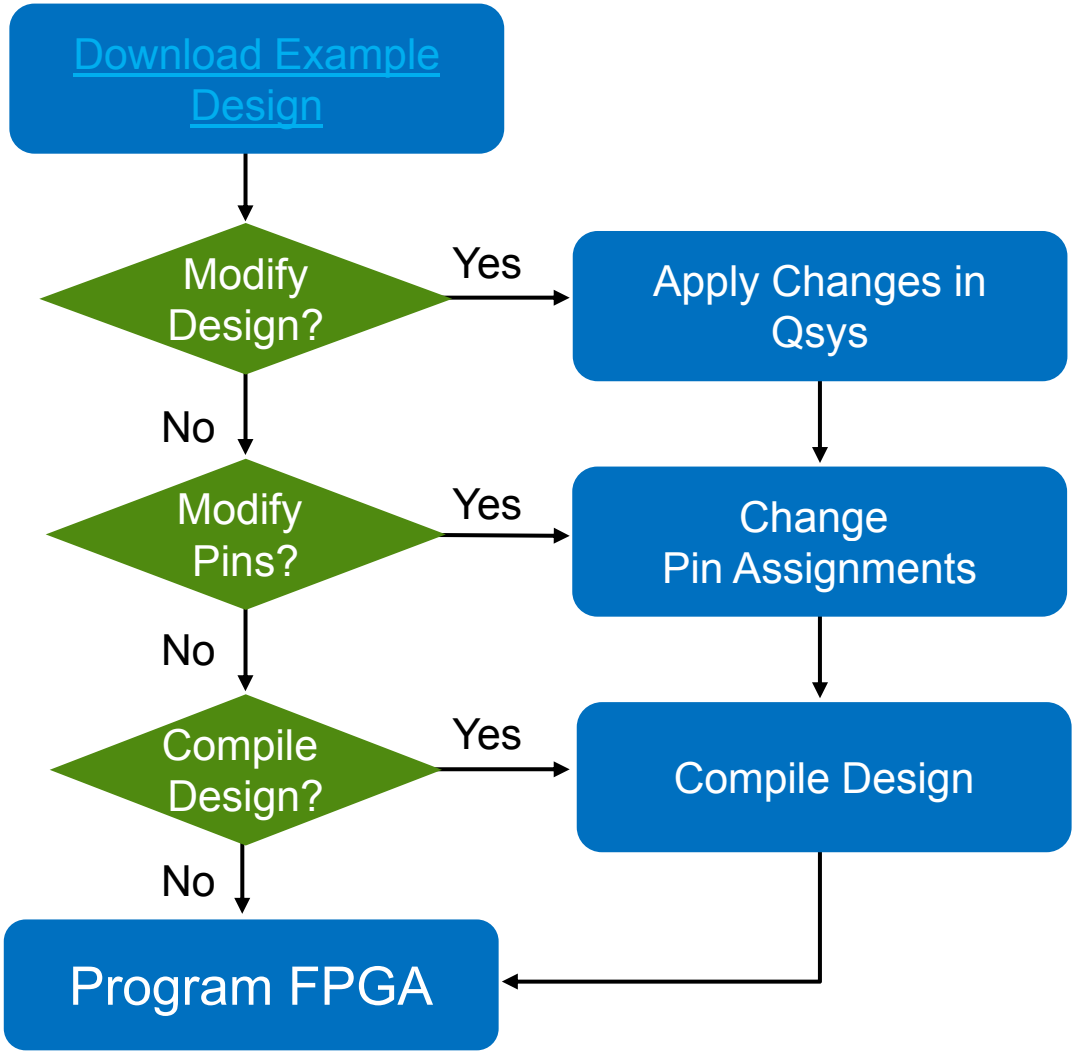
- The Transceiver Toolkit helps users generate designs to control and test transceivers during PCB bring up or in-system signal integrity analysis
- Users can easily control PMA settings, generate and check PRBS patterns to ensure optimal link operation
- Both command line and graphical user interfaces are available

Transceiver Toolkit Features Overview

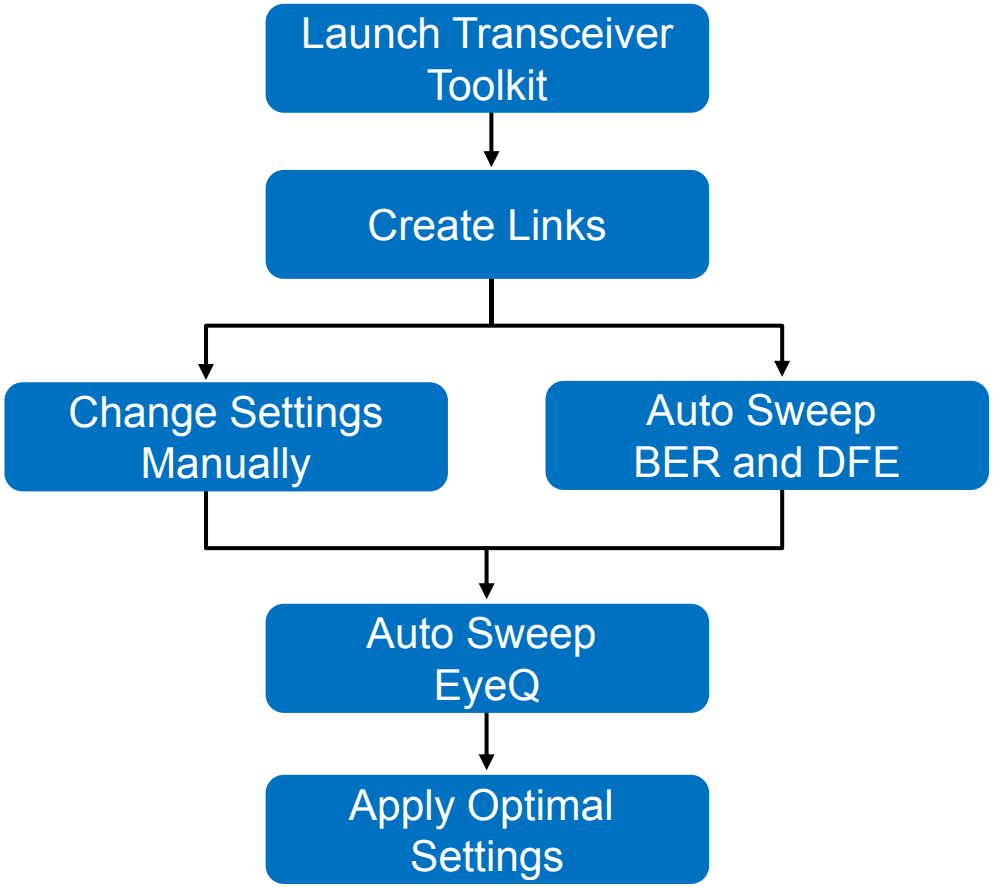
Features	Description
Transceiver channels	• Consist of full-duplex transmitter (TX) and receiver (RX) channels • Enable and disable each transceiver channel
Dynamic reconfiguration	• Change configuration (e.g. data rate, differential output voltage (V_{OD}), pre- emphasis, decision feedback equalizer (DFE), continuous-time linear equalizer (CTLE), and EyeQ) at run time
EyeQ	• Draw BER bathtub curve and eye contour – Stratix V only
DFE	• Enable decision feedback equalization (DFE) – Stratix V only
Data pattern generator and checker	• Change test patterns (e.g. pseudo-random binary sequence (PRBS) 7, 15, 23, and 31) at run time
Auto sweep	• Run sweep tests to converge optimal pre-emphasis and EyeQ settings • Provide target BER for error check
Error insertion	• Insert error on serial data in transceiver channel
Status	• Indicate link lock, # of errors, # of transmitted data, and BER
Reporting	• Report settings converged for different BER, equalization, and pre-emphasis
Diagnostics	• Perform serial loopback and reverse serial loopback tests

Getting Started

Recommended User Flow (1/2)

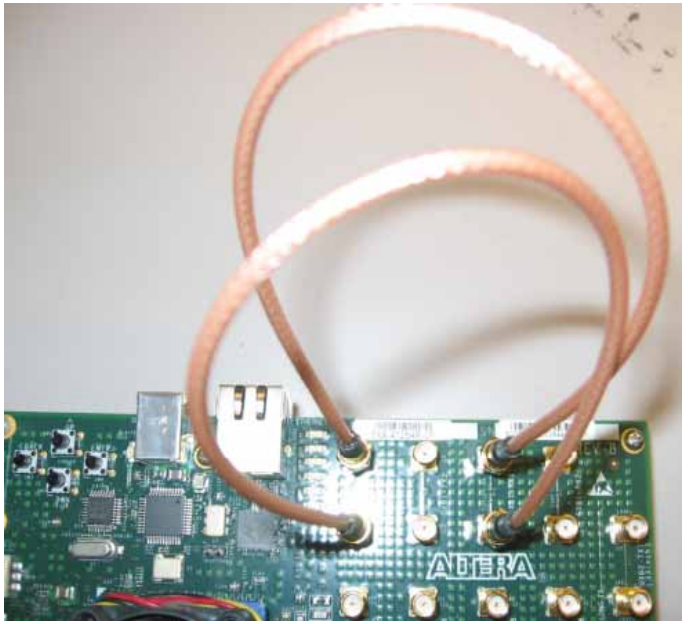


Recommended User Flow (2/2)

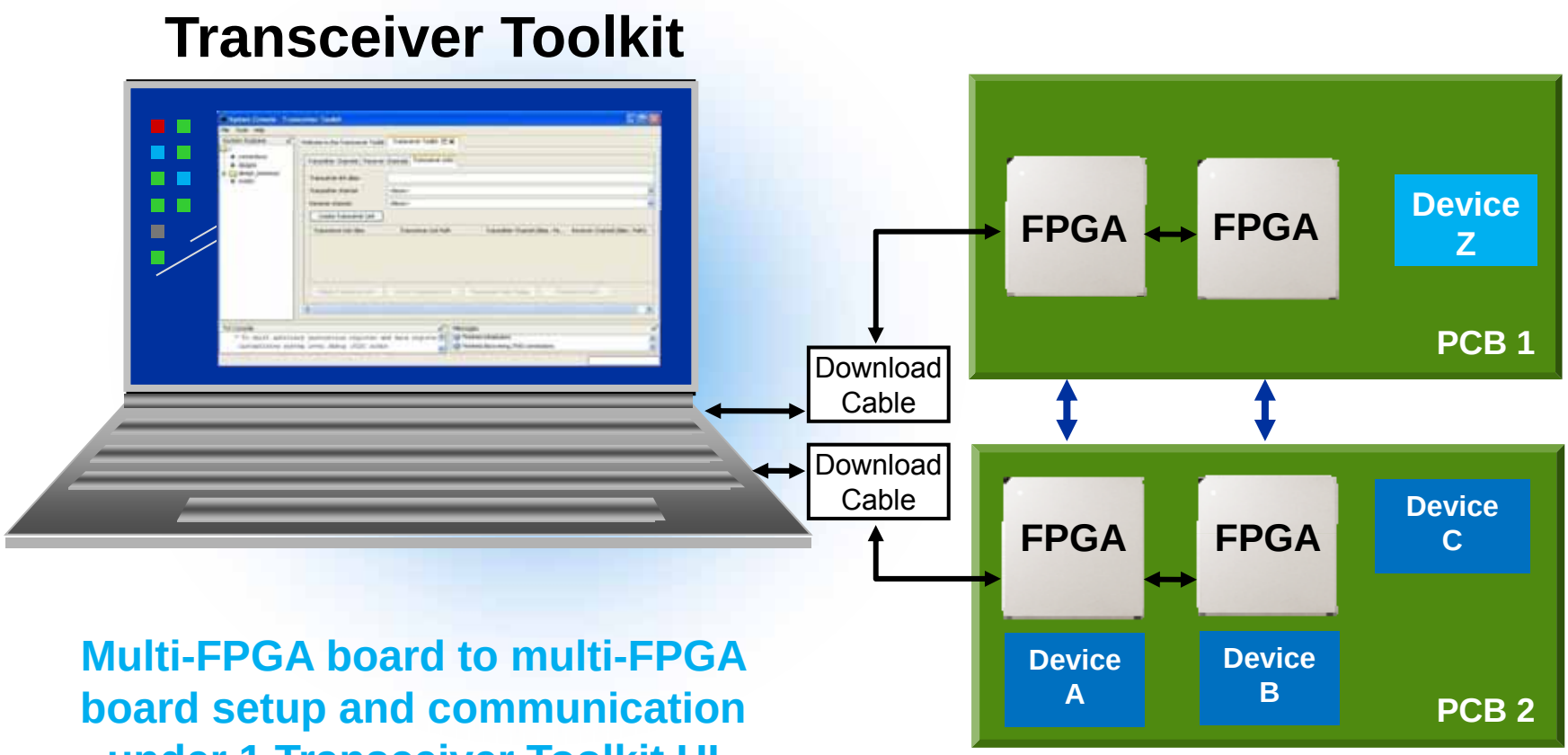


PCB Set-Up

- Power up PCB
- Check if FPGA was configured successfully
- Make sure the links you want to test are connected correctly (e.g. connect cables for external loopback shown below)



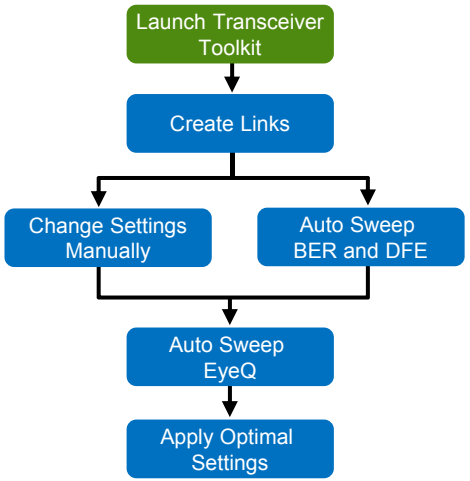
Board to Board

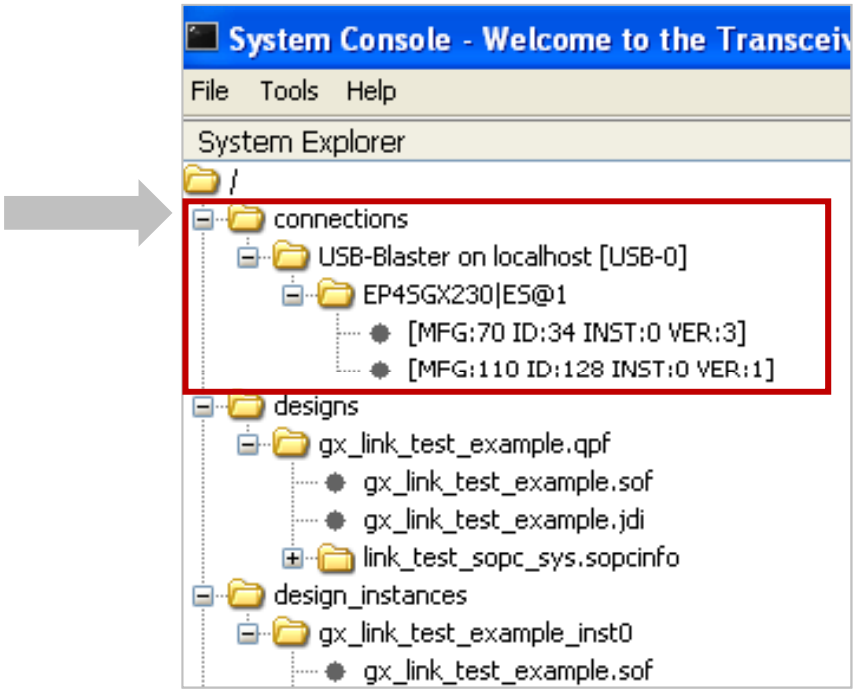


Multi-FPGA board to multi-FPGA board setup and communication under 1 Transceiver Toolkit UI

Launch Transceiver Toolkit

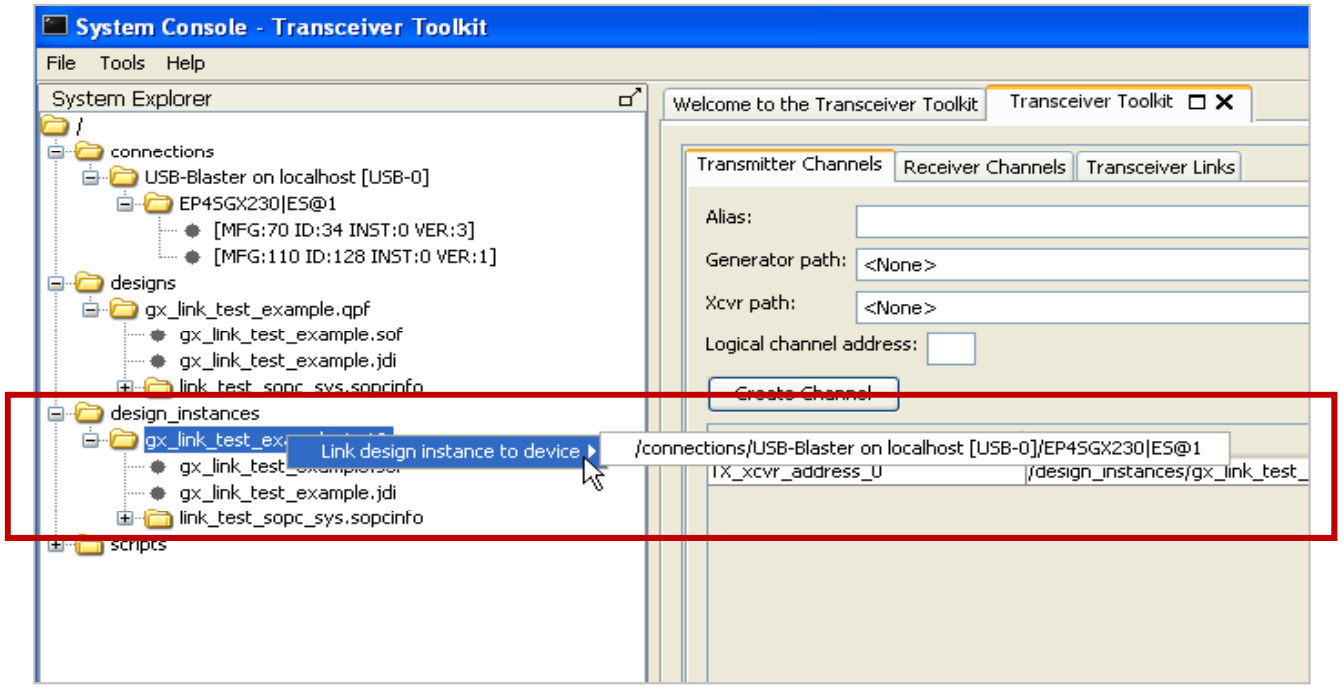
- Click **Transceiver Toolkit** from **Tools** menu in Quartus II software
- From System Explorer window, under "**connections**", check list for your JTAG device
  - If device name does not appear, make sure the device is powered on and connected to the machine
- Load Quartus II project containing transceiver design
  - File → Load Project
- Open Transceiver Toolkit tab
  - Tools → Transceiver Toolkit



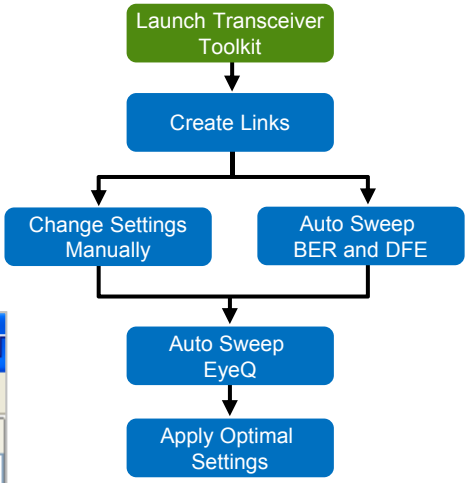


Connect to Targeted Device

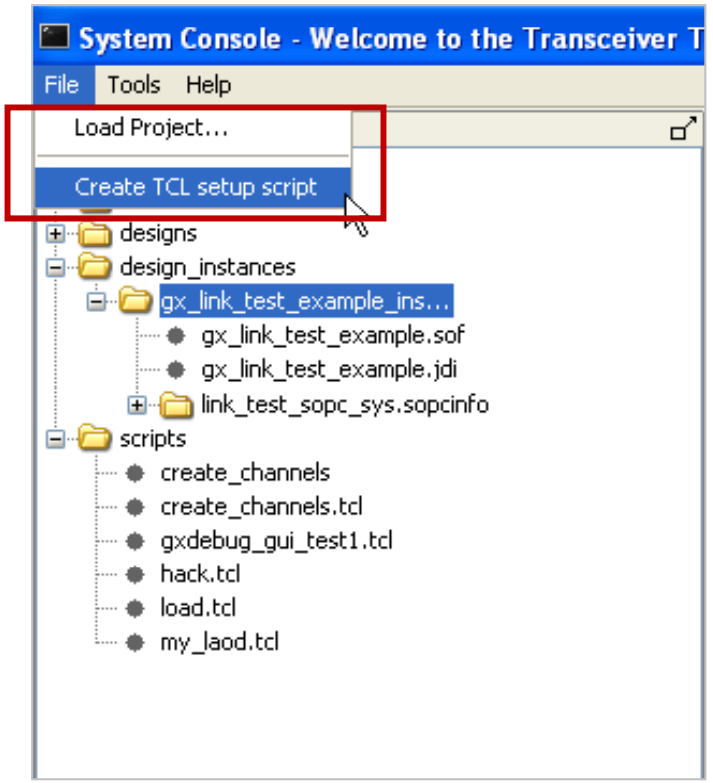
- Linking design instance to device:
  - From System Explorer window, navigate to “**design_instances**”
  - Right click on targeted instance
  - Select your connected device



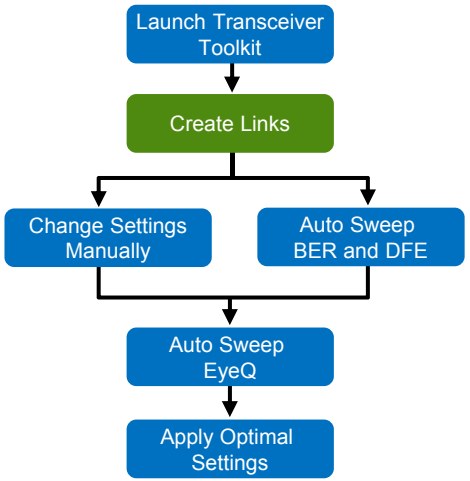
Create Tcl Set-Up Script



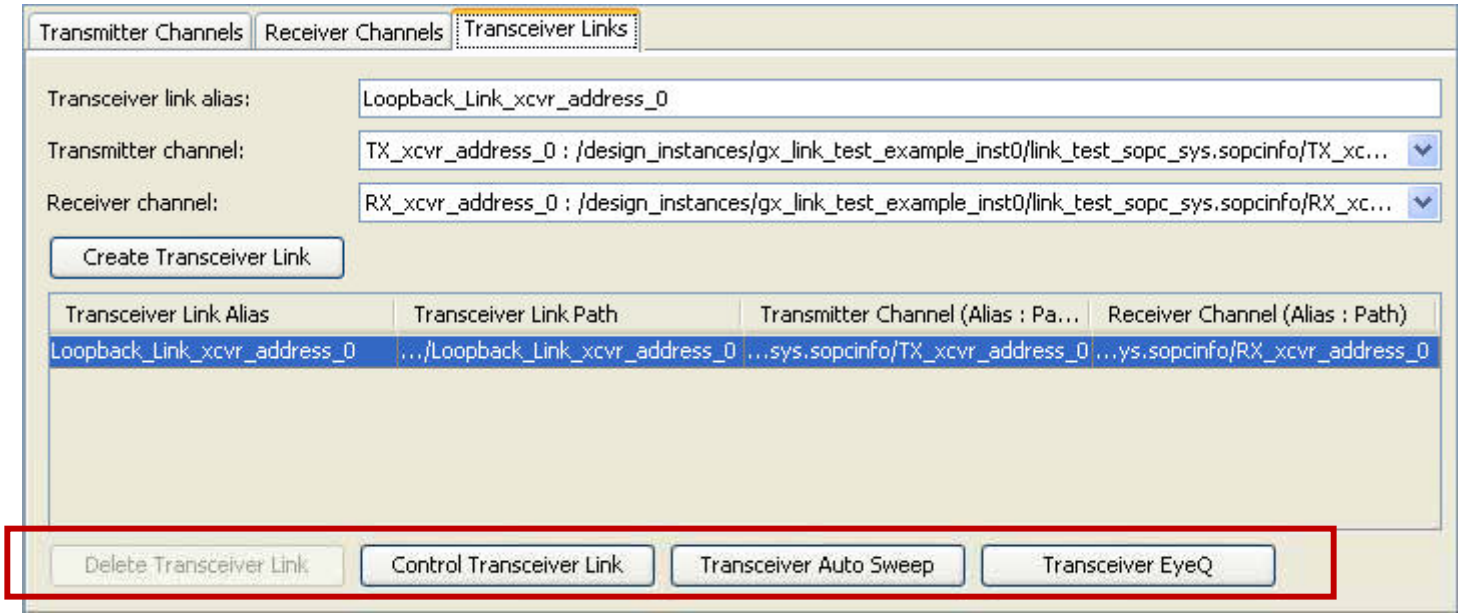
- Automate set-up using Tcl script
  - Click **Save As…** from **File** menu
  - All saved scripts are shown under "**scripts**"
  - Click on a script



Create Links

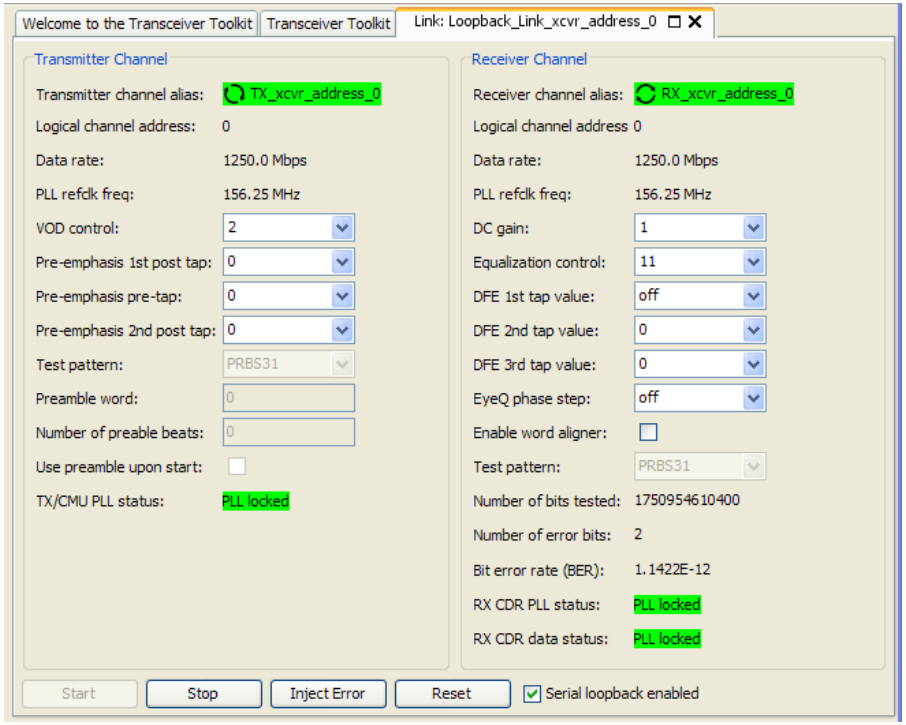


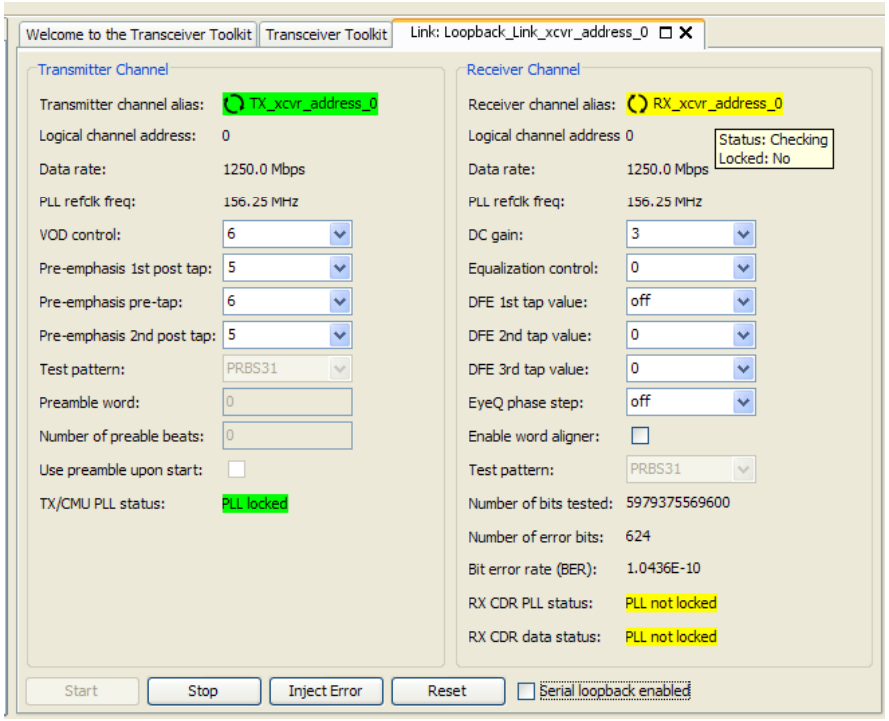
- Transceiver Links tab
  - The channels in your design are auto-populated in the **Transmitter Channels** and **Receiver Channels** tabs
  - By default, a link will be created between the transmitter and receiver of the same channel and shown in **Transceiver Links**



Control Transceiver Link

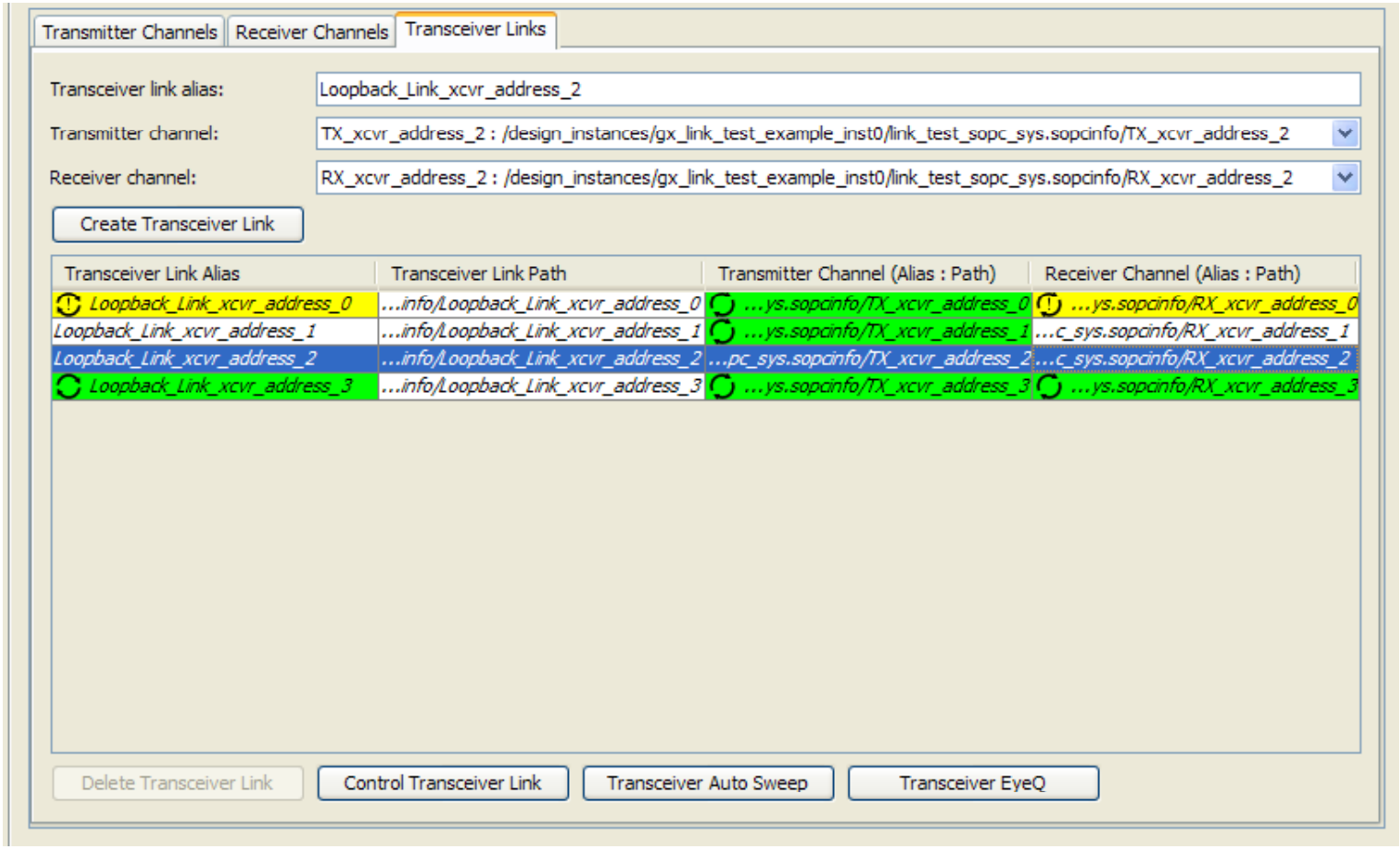
- **New Custom Control Panel (From QII 11.0 onwards)**
  - Intelligent control management for improved ease of use
  - Additional data reported – channel datarate, reference clock, LTD/LTR lock status
  - Serial Loopback Control added
  - Highly visible status indicators – color and animation



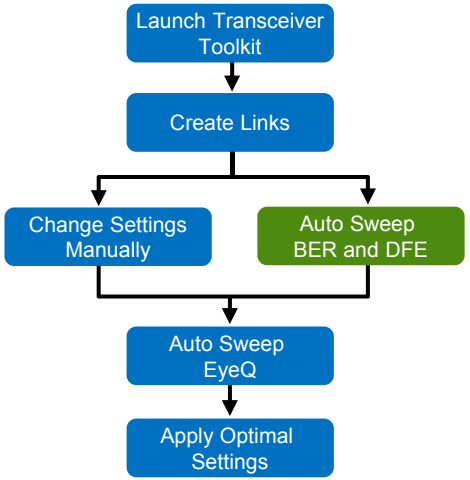


Improved Test Management

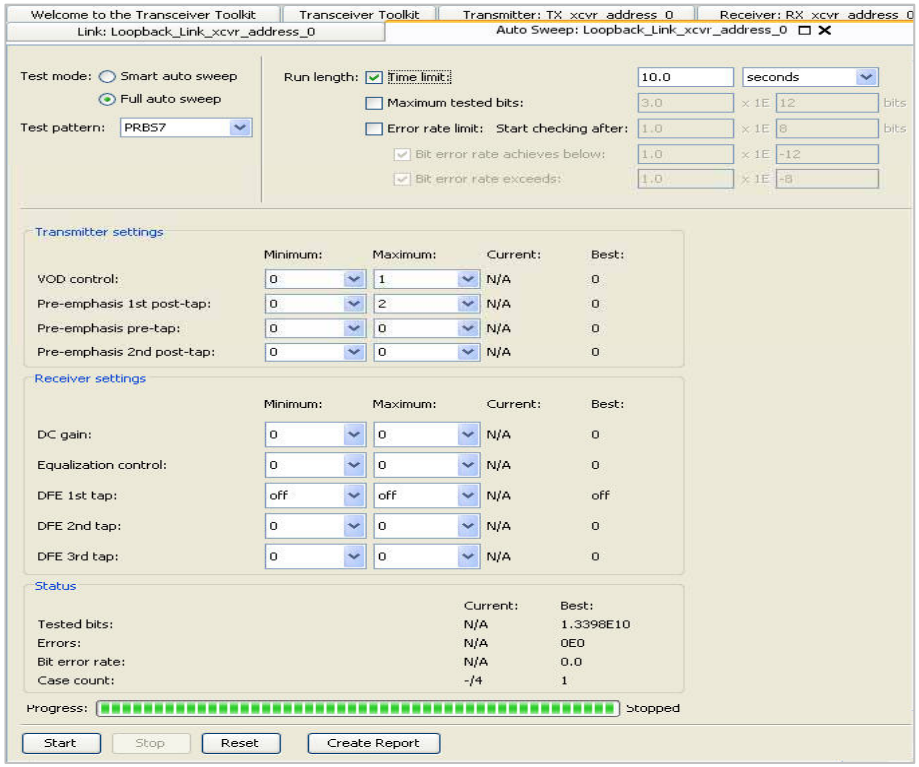
- Easily manage many parallel running BER tests
- Identify which channel resources are in use
- Quickly identify test run status (not started, running, running with problem)



Auto-Sweep BER Tests

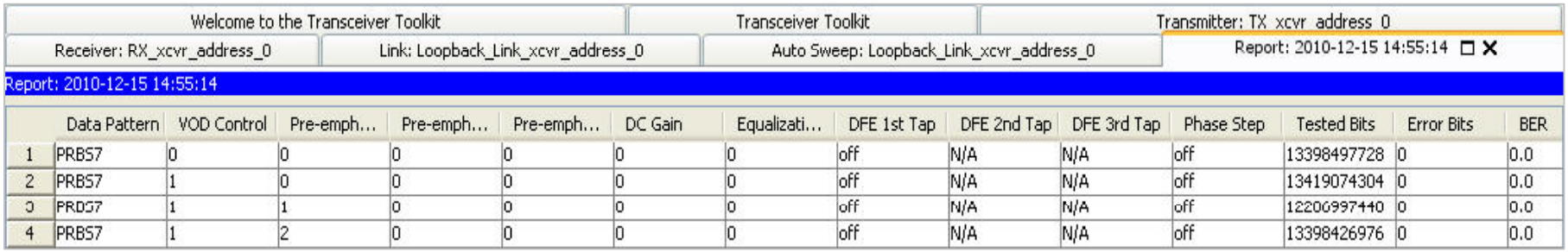


- Click **Transceiver Auto Sweep**
  - Choose the Minimum and Maximum values for each setting (e.g. VOD, pre-emphasis, DC gain, equalization, etc.)
  - Case count = number of different permutations will be performed
  - Run length = conditions for stopping the auto-sweep (e.g. time limit per permutation, error rate, etc.)



Auto-Sweep BER Report

- Create reports by clicking **Create Report** in the middle of the test or when the test completes
- Different columns show various transceiver settings
  - All columns can be sorted
- The report is exportable in a .csv format
  - Right click on report, Select "**Export…**"



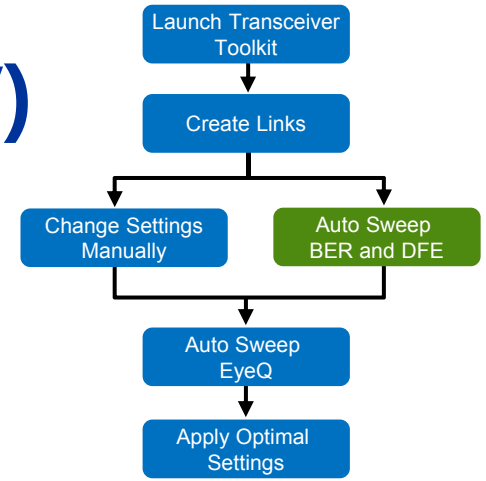
Welcome to the Transceiver Toolkit | Transceiver Toolkit | Transmitter: TX_xcvr_address_0

Receiver: RX_xcvr_address_0 | Link: Loopback_Link_xcvr_address_0 | Auto Sweep: Loopback_Link_xcvr_address_0 | Report: 2010-12-15 14:55:14

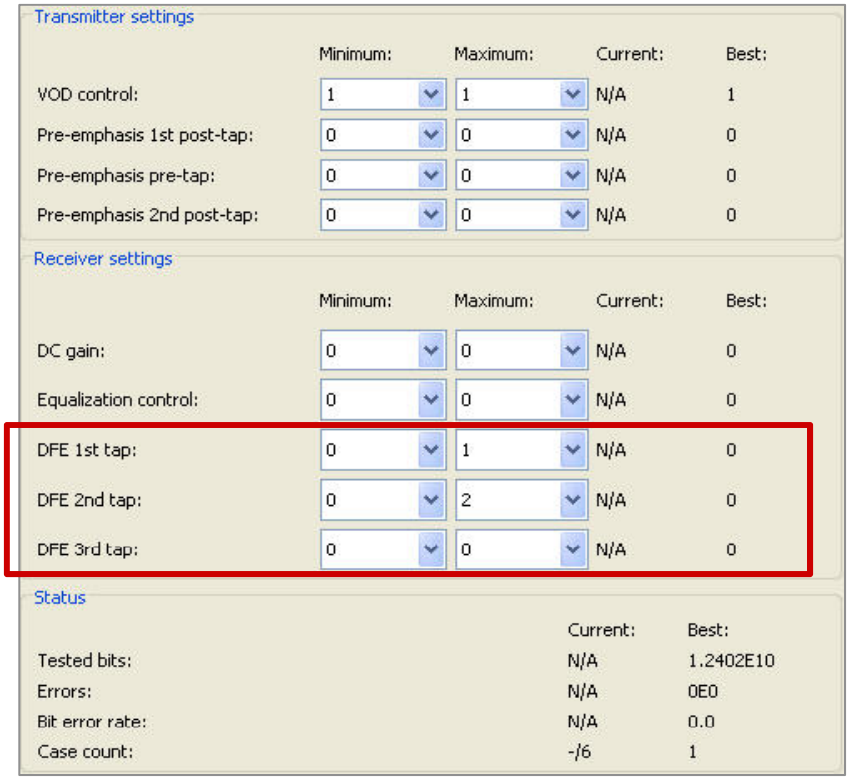
Report: 2010-12-15 14:55:14

	Data Pattern	VOD Control	Pre-emph…	Pre-emph…	Pre-emph…	DC Gain	Equalizati…	DFE 1st Tap	DFE 2nd Tap	DFE 3rd Tap	Phase Step	Tested Bits	Error Bits	BER
1	PRBS7	0	0	0	0	0	0	off	N/A	N/A	off	13398497728	0	0.0
2	PRBS7	1	0	0	0	0	0	off	N/A	N/A	off	13419074304	0	0.0
3	PRBS7	1	1	0	0	0	0	off	N/A	N/A	off	12206997440	0	0.0
4	PRBS7	1	2	0	0	0	0	off	N/A	N/A	off	13398426976	0	0.0

Auto-Sweep DFE Tests (Stratix V)



1. Auto sweep with DFE off
2. With best BER results, lock down settings (VOD, pre-emphasis, DC gain, equalization)
3. Then sweep with DFE settings to find best BER



Transmitter settings

	Minimum:	Maximum:	Current:	Best:
VOD control:	1	1	N/A	1
Pre-emphasis 1st post-tap:	0	0	N/A	0
Pre-emphasis pre-tap:	0	0	N/A	0
Pre-emphasis 2nd post-tap:	0	0	N/A	0

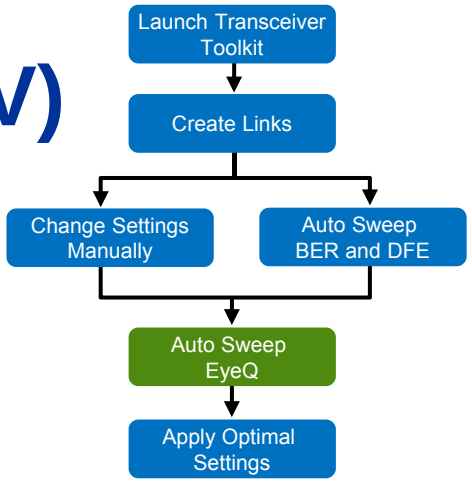
Receiver settings

	Minimum:	Maximum:	Current:	Best:
DC gain:	0	0	N/A	0
Equalization control:	0	0	N/A	0
DFE 1st tap:	0	1	N/A	0
DFE 2nd tap:	0	2	N/A	0
DFE 3rd tap:	0	0	N/A	0

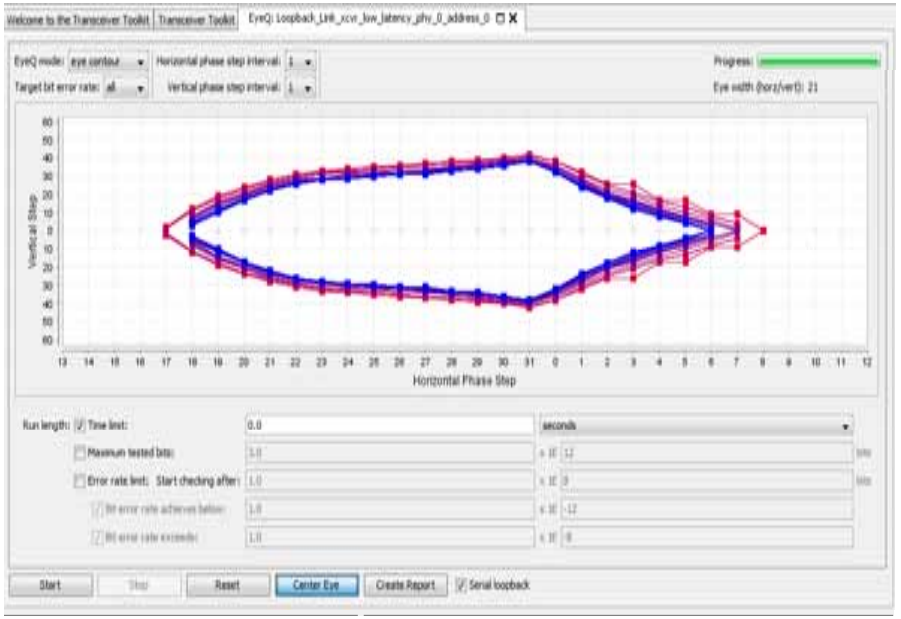
Status

	Current:	Best:
Tested bits:	N/A	1.2402E10
Errors:	N/A	0E0
Bit error rate:	N/A	0.0
Case count:	-/6	1

Auto-Sweep EyeQ Tests (Stratix V)



- Click **Transceiver EyeQ** button on **Transceiver Links** tab to launch the EyeQ window
- Eye Contour automatically generated when sweep completes
- Click **Create Report** to generate EyeQ reports
  - If you sort by BER column, the number of rows with BER = 0 will be considered as the unit width of the eye from the specified physical media attachment (PMA) settings



Demo