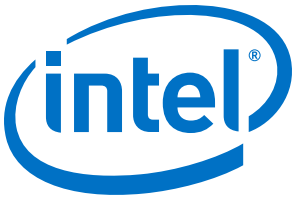


SerialLite III Streaming IP Core User Guide

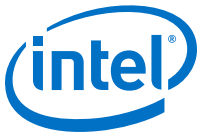
UG-01126
2017.05.08

Last updated for Intel® Quartus® Prime Design Suite: 17.0


Subscribe


Send Feedback

Contents



1 SerialLite III Streaming IP Core Quick Reference

Note: Stratix® 10 devices are not supported in the 17.0 Quartus Prime software. For more information, contact Intel.

The SerialLite III Streaming MegaCore® function is a lightweight protocol suitable for high bandwidth streaming data in chip-to-chip, board-to-board, and backplane applications.

Table 1. SerialLite III Streaming IP Core

Item		Description
Release Information	Version	17.0 (Arria 10, Stratix V and Arria V GZ devices) Quartus® Prime Pro Edition (Stratix 10 devices)
	Release Date	May 2017
	IP Catalog Name	• SerialLite III Streaming (Stratix 10, Stratix V, and Arria V GZ devices) • Arria 10 SerialLite III Streaming (Arria 10 devices)
	Ordering Code	IP-SLITE3/ST
	Product ID	010A
	Vendor ID	6AF7
IP Core Information	Core Features	• Up to 17.4 Gbps lane data rates for Stratix 10 and Arria 10 devices. • Supports 1–24 serial lanes in configurations that provide nominal bandwidths from 3.125 gigabits per second (Gbps) to over 300 Gbps.
	Protocol Features	• Simplex and duplex operations • Support for single or multiple lanes • 64B/67B physical layer encoding • Payload and idle scrambling • Error detection • Low overhead framing • Low point-to-point transfer latency
	Typical Application	• High resolution video • Radar processing • Medical imaging • Baseband processing in wireless infrastructure
	Device Family Support	Stratix 10 (Advance support), Arria® 10 (Final support) , Arria V GZ (Final support), and Stratix® V (Final support) FPGA devices. **Advance support** - The IP core is available for simulation and compilation for this device family. FPGA programming file (.pof) support is not available for Quartus Prime Pro Stratix 10 Edition Beta software and as such IP timing closure cannot be guaranteed. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies,

continued...

Item		Description
		simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs). **Final support** - The IP core is verified with final timimhg models for this device family. The IP core meets all the functional and timing requirements for the device family and can be used in production designs.
	Design Tools	• IP parameter editor in the Quartus Prime software for IP design instantiation and compilation • TimeQuest timing analyzer in the Quartus Prime software for timing analysis • ModelSim-Intel FPGA Edition, MATLAB, or third-party tool using NativeLink for design simulation or synthesis

Related Links

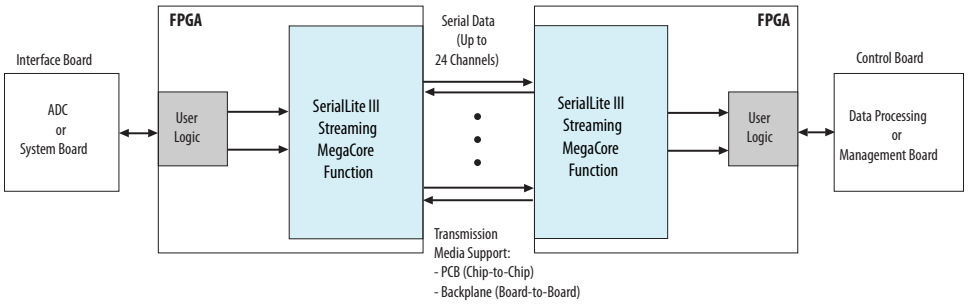
- Intel Software Installation and Licensing
- What's New in Intel IP
- SerialLite III Streaming IP Core Release Notes
- Errata for SerialLite III Streaming IP core in the Knowledge Base
- SerialLite III Streaming IP Core Design Example User Guide
- SerialLite III Streaming IP Core User Guide Archives on page 95
  Provides a list of user guides for previous versions of the SerialLite III Streaming IP core.
- Timing and Power Models
  Reports the default device support levels in the current version of the Quartus Prime Pro Edition software.
- Timing and Power Models
  Reports the default device support levels in the current version of the Quartus Prime Standard Edition software.

2 About the SerialLite III Streaming IP Core

The SerialLite III Streaming IP core is a high-speed serial communication protocol for chip-to-chip, board-to-board, and backplane application data transfers. This protocol offers high-bandwidth, low overhead frames, low I/O count, and supports scalability in both number of lanes and lane speed.

The SerialLite III Streaming IP core incorporates a physical coding sublayer (PCS), a physical media attachment (PMA), and a media access control (MAC) block. The IP core transmits and receives streaming data through the Avalon-ST interface on its FPGA fabric interface.

Figure 1. Typical System Application



2.1 SerialLite III Streaming Protocol

The SerialLite III Streaming IP core implements a protocol that supports the transfer of high bandwidth streaming data over a unidirectional or bidirectional, high-speed serial link.

The SerialLite III Streaming IP core has the following protocol features:

- Simplex and duplex operations
- Support for single or multiple lanes
- 64B/67B physical layer encoding
- Payload and idle scrambling
- Error detection
- Low protocol overhead
- Low point-to-point transfer latency
- Reduces soft logic resource utilization using hardened Native PHY IP core (Arria 10 and Stratix 10 devices) or Interlaken PHY IP core (Stratix V and Arria V GZ devices)

2.2 SerialLite III Streaming Protocol Operating Modes

The protocol defines two operating modes for different applications: continuous and burst mode.

The IP core that you generate can be in either mode. No parameter option is available to select between continuous and burst modes. The selection depends on how you provide data at the Avalon-ST TX interface.

2.2.1 Continuous Mode

A SerialLite III Streaming link operating in continuous mode accepts and transmits user data over the link, and presents it on the user interface at the receiving link at the same rate and without gaps in the stream, if user logic does not de-assert data valid signal or insert control words as part of the stream. However, if user logic de-asserts the data valid signal or insert control words in the middle of data transfer, the streaming interface will no longer operate in continuous mode and there is no guarantee the end-point sink is able to replicate the exact data valid pattern of the source. When operating in this mode, a link implementing the protocol looks like a data pipe that can transparently forward all data presented on the user interface to the far end of the link.

Continuous mode is appropriate for applications that require a simple interface to transmit a single, high bandwidth data stream. An example of this application is sensor data links for radar and wireless infrastructure. With this mode, data converters can connect to either end of the link with minimal interface logic. This mode requires both ends of the link to operate from a common transceiver reference clock.

Note: Continuous mode is only applicable in Standard Clocking Mode. It is not possible to operate in this mode when asynchronous clocking is implemented in your design. This is because the data valid signal needs to be de-asserted to break the data stream to avoid the PPM difference at the sink from overflowing the sink adaptation module's FIFO buffer.

2.2.2 Burst Mode

A SerialLite III Streaming link operating in burst mode accepts bursts of data across the user interface and transmits each burst across the link as a discrete data burst.

Burst mode is appropriate for applications where the data stream is divided into bursts of data. An example of this application is uncompressed digital video where the data stream is divided into lines of display raster. This mode provides more flexibility to the clocking and also supports multiplexing of multiple data streams across the link.

Note: The minimum required gap between bursts is 1 user clock cycle in standard and advanced clocking modes on the transmit side. Therefore,you must provide one extra user clock cycle between an end of burst and the start of the next burst. The SerialLite III Streaming IP core allows you to select between 1 or 2 burst gap. To connect the IP core of version 15.1 to IP core of the previous version, you must select a burst gap of 2 for backward compatability.

Related Links

- Standard Clocking Mode on page 37

- Advanced Clocking Mode on page 39

2.3 Performance and Resource Utilization

This typical resources and expected performance for different SerialLite III Streaming IP core variations are obtained using the Quartus Prime software targeting the Stratix V GX (5SGXMA7H2F35C2), the Arria V GZ (5AGZME7K2F40I3L), and the Arria 10 (10AX115S1F45I1SGES) FPGA device.

Note: The numbers of ALMs and logic registers in the following table are rounded up to the nearest 100.

Table 2. SerialLite III Streaming IP Core FPGA Performance and Resource Utilization

Device	Direction	Clocking Mode	Parameters			ALMs	Logic Registers		M20K
			Number of Lanes	Per-Lane Data Rate (Mbps)	ECC		Primary	Secondary	
Arria 10	Source	Standard	24	17400	Disabled	2613	5049	780	39
		Standard	24	17400	Enabled	5961	9680	525	72
		Advanced	24	17400	Disabled	3009	5240	570	39
		Advanced	24	17400	Enabled	6065	9659	552	72
	Sink	Standard	24	17400	Disabled	3974	7550	1750	49
		Standard	24	17400	Enabled	4065	7570	1632	50
		Advanced	24	17400	Disabled	3297	5815	1580	0
		Advanced	24	17400	Enabled	3275	5524	1870	0
	Duplex	Standard	24	17400	Disabled	6152	12511	2000	88
		Standard	24	17400	Enabled	9313	16606	2193	122
		Advanced	24	17400	Disabled	5833	10462	2146	39
		Advanced	24	17400	Enabled	8868	14853	2112	72
Stratix V GX and Arria V GZ	Source	Standard	24	10312.50	Disabled	5684	6114	46	39
		Standard	24	10312.50	Enabled	11122	13422	271	72
		Advanced	24	10312.50	Disabled	5680	6104	43	39
		Advanced	24	10312.50	Enabled	11015	13418	239	72
	Sink	Standard	24	10312.50	Disabled	5499	9601	93	49
		Standard	24	10312.50	Enabled	5517	9510	91	50
		Advanced	24	10312.50	Disabled	4356	7757	43	0
		Advanced	24	10312.50	Enabled	4356	7757	43	0
	Duplex	Standard	24	10312.50	Disabled	8742	15024	165	88
		Standard	24	10312.50	Enabled	14045	22279	337	122
		Advanced	24	10312.50	Disabled	7550	13211	74	39
		Advanced	24	10312.50	Enabled	12606	20534	293	72

continued...

Device	Direction	Clocking Mode	Parameters			ALMs	Logic Registers		M20K
			Number of Lanes	Per-Lane Data Rate (Mbps)	ECC		Primary	Secondary	
Stratix 10	Source	Standard	16	17400	Disabled	2427	3840	653	26
		Standard	16	17400	Enabled	7161	8304	339	33
		Advanced	16	17400	Disabled	2424	3892	599	26
		Advanced	16	17400	Enabled	7155	8326	318	33
	Sink	Standard	16	17400	Disabled	3651	5211	1112	26
		Standard	16	17400	Enabled	3948	5166	1135	33
		Advanced	16	17400	Disabled	3076	4673	1466	0
		Advanced	16	17400	Enabled	3076	4673	1466	0
	Duplex	Standard	16	17400	Disabled	5290	7980	1743	52
		Standard	16	17400	Enabled	9986	12388	1473	66
		Advanced	16	17400	Disabled	4725	7578	1967	26
		Advanced	16	17400	Enabled	9375	11910	1793	33

Related Links

Fitter Resources Reports

More information about Quartus Prime resource utilization reporting.

3 Getting Started

Related Links

- Introduction to Intel IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

3.1 Installing and Licensing IP Cores

The Intel® Quartus Prime software installation includes the Intel FPGA IP library. This library provides useful IP core functions for your production use without the need for an additional license. Some IP cores in the library require that you purchase a separate license for production use. The OpenCore® feature allows evaluation of any Intel FPGA IP core in simulation and compilation in the Quartus Prime software. Upon satisfaction with functionality and performance, visit the Self Service Licensing Center to obtain a license number for any Intel FPGA product.

The Quartus Prime software installs IP cores in the following locations by default:

Figure 2. IP Core Installation Path

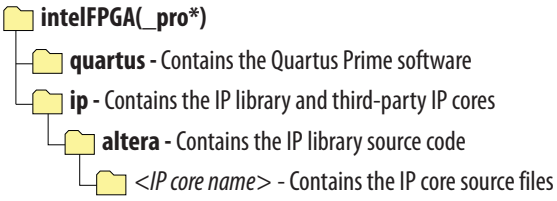


Table 3. IP Core Installation Locations

Location	Software	Platform
`<drive>:\intelFPGA_pro\quartus\ip\altera`	Quartus Prime Pro Edition	Windows*
`<drive>:\intelFPGA\quartus\ip\altera`	Quartus Prime Standard Edition	Windows
`<home directory>:/intelFPGA_pro/quartus/ip/altera`	Quartus Prime Pro Edition	Linux*
`<home directory>:/intelFPGA/quartus/ip/altera`	Quartus Prime Standard Edition	Linux

3.2 OpenCore Plus IP Evaluation

The free OpenCore Plus feature allows you to evaluate licensed MegaCore IP cores in simulation and hardware before purchase. Purchase a license for MegaCore IP cores if you decide to take your design to production. OpenCore Plus supports the following evaluations:

- Simulate the behavior of a licensed IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

OpenCore Plus evaluation supports the following two operation modes:

- Untethered—run the design containing the licensed IP for a limited time.
- Tethered—run the design containing the licensed IP for a longer time or indefinitely. This operation requires a connection between your board and the host computer.

Note: All IP cores that use OpenCore Plus time out simultaneously when any IP core in the design times out.

Related Links

- Quartus Prime Licensing Site
- Quartus Prime Installation and Licensing

3.2.1 IP Core OpenCore Plus Timeout Behavior

All IP cores in a device time out simultaneously when the most restrictive evaluation time is reached. If there is more than one IP core in a design, the time-out behavior of the other IP cores may mask the time-out behavior of a specific IP core .

For IP cores, the untethered time-out is 1 hour; the tethered time-out value is indefinite. Your design stops working after the hardware evaluation time expires. The Quartus Prime software uses OpenCore Plus Files (`.ocp`) in your project directory to identify your use of the OpenCore Plus evaluation program. After you activate the feature, do not delete these files..

When the evaluation time expires the `link_up` signal goes low .

Related Links

AN 320: OpenCore Plus Evaluation of Megafunctions

3.3 Specifying IP Core Parameters and Options

Follow these steps to specify IP core parameters and options.

1. In the Qsys IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. This name identifies the IP core variation files in your project. If prompted, also specify the target Intel FPGA device family and output file HDL preference. Click **OK**.
3. Specify parameters and options for your IP variation:
   - Optionally select preset parameter values. Presets specify all initial parameter values for specific applications (where provided).
   - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
   - Specify options for generation of a timing netlist, simulation model, testbench, or example design (where applicable).
   - Specify options for processing the IP core files in other EDA tools.
4. Click **Finish** to generate synthesis and other optional files matching your IP variation specifications. The parameter editor generates the top-level `.qsys` IP variation file and HDL files for synthesis and simulation. Some IP cores also simultaneously generate a testbench or example design for hardware testing.

The top-level IP variation is added to the current Quartus Prime project. Click **Project ➤ Add/Remove Files in Project** to manually add a `.qsys` or `.ip` file to a project. Make appropriate pin assignments to connect ports.

3.3.1 SerialLite III Parameter Editor

Based on the values you set, the SerialLite III streaming parameter editor automatically calculates the rest of the parameters, and provides you with the following values or information:

- Input data rate per lane
- Transceiver data rate per lane
- A list of feasible transceiver reference clock frequencies, one of which you select to provide to the core
- Information related to the core overheads

Important: If your design targets Stratix V or Arria V GZ devices, you cannot migrate your design to Arria 10 and Stratix 10 devices automatically. For Arria 10 devices, the transceiver reconfiguration functionality is embedded inside the transceivers. Therefore, you must re-instantiate the IP core to target Arria 10 devices. For Stratix 10 devices, you must re-instantiate the IP core to target Stratix 10 devices due to the transceiver architecture differences.

Related Links

SerialLite III Streaming IP Core Parameters on page 13

3.3.2 Arria 10 Designs

If your design targets the Arria 10 devices:

- The parameter editor displays a message about the required output clock frequency of the external TX PLL IP clock. For source or duplex modes, connect the Transceiver PHY Reset Controller to the TX PLL to ensure the appropriate HSSI power-up sequence.
- For source only Arria 10 implementations, the parameter editor does not provide the transceiver reference clock frequency because the user is expected to provide the transmit serial clock. If you use an on-chip PLL to generate the transmit serial clock, you can use the same PLL reference clock frequency that you provide to the core in the sink direction, operating at the same user clock frequency (or equivalent transceiver lane data rate).
- The SerialLite III Streaming IP core expects the user to provide the transmitter's serial clock. If you compile the IP without the proper serial clock, the Quartus Prime Compiler issues a compilation error.
- When generating the example testbench, the SerialLite III Streaming IP core instantiates an external transceiver ATX PLL for the transmit serial clock based on the required user clock only when configured in sink or duplex mode. The Arria 10 simulation testbench uses the external transceiver ATX PLL. The transceiver ATX PLL core is configured with the transceiver reference clock specified in the parameter editor and transmit serial clock.
- To generate the SerialLite III Streaming Arria 10 example testbench using the parameter editor, select **Generate Example Designs > <*directory_name*>** . Intel recommends that you generate the Arria 10 simulation testbench for the sink or duplex direction.

Related Links

- SerialLite III Streaming IP Core Parameters on page 13
- Arria 10 versus Stratix V and Arria V GZ Variations on page 35
- SerialLite III Streaming IP Core Design Example User Guide

3.4 SerialLite III Streaming IP Core Parameters

Table 4. SerialLite III Streaming IP Core Parameters

Parameter	Value	Default	Description
General Design Options			
Direction	Source, Sink, Duplex	Duplex	Supports simplex transmitter, simplex receiver, or full duplex transmissions.
Number of lanes	1–24	2 (Arria 10, Stratix V, and Arria V GZ) 6 (Stratix 10)	Specifies the number of lanes (equal to physical transceiver links) that are used to transfer the streaming data.
Device speed grade	1–4	2	Specifies the device speed grade (Stratix V and Arria V GZ devices only).
PLL type	ATX, CMU	CMU	Selects the transceiver PLL type. (Stratix V and Arria V GZ devices only)

continued...

Parameter	Value	Default	Description
Transceiver reference clock frequency	*<Range supported by the transceiver PLLs>*	644.53125 MHz (Arria 10, Stratix V, and Arria V GZ) 312.5 MHz (Stratix 10)	Supports multiple transceiver reference clock frequencies for flexibility in the oscillator and PLL choices. This transceiver reference clock frequency must match the external PLL reference clock frequency.
Meta frame length in words	200–8191	200	Specifies the metaframe length.
Enable Transceiver Native PHY ADME	Yes/No	No	Enables ADME and Optional Reconfiguration Logic parameters of the Transceiver Native PHY. (Arria 10 devices only)
Enable M20K ECC support	Yes/No	No	Select to use error correcting code (ECC) protection to strengthen the FIFO buffers from single-event upset (SEU) changes. Enables built-in error correcting code (ECC) support on the M20K embedded block memory for single-error correction, double-adjacent-error correction, and triple-adjacent-error detection.
User Interface			
Streaming Mode	Basic, Full	Full	Specifies the streaming mode. • Basic: This is a pure streaming mode where data is sent without burst, sync, empty cycle, and frame delimiter to increase bandwidth. • Full: This mode sends a burst and sync cycle at the start of frame and a burst and empty cycle at the end of frame. Provide a gap of one empty cycle between two data frames. (Stratix 10 devices only) *Note:* Basic streaming mode is not supported in Quartus Prime Pro Edition.
Required idle cycles between bursts	1, 2	2	Supports two values to optimize for bandwidth efficiency or maintain backward compatibility with existing SerialLite III Streaming IP cores (legacy). • 1: Recommended for high bandwidth streaming. The same Burst Gap setting must be set for both source and sink IP core. • 2: For backward compatibility with Quartus II version 15.1 and older sink IP core.
Adaptation FIFO partial full threshold	8 - 18	15	Specifies the partial full threshold of the transmit FIFO. `ready_tx` signal will de-assert when data reaches this level in the FIFO. (Stratix 10 devices only)
Clocking mode	Standard clocking mode, Advanced clocking mode	Standard clocking mode	Specifies the clocking mode. Refer to SerialLite III Streaming IP Core Clocking Guidelines on page 69 for more information.
User input	User clock frequency, Transceiver data rate	User clock frequency	Select **User clock frequency** to specify the user clock input and allow the IP core to determine the transceiver data rate. Select **Transceiver data rate** to specify the desired data rate and allow the IP core to determine the user clock frequency.
User clock frequency required	Minimum: 50 MHz Maximum: Limited by the supported transceiver data rates	146.484375 MHz (Stratix V and Arria V GZ) 150 MHz (Arria 10)	Specifies the desired frequency for the user clock input for the transmit (Standard Clocking Mode and Advanced Clocking Mode) and receive user interface (Standard Clocking Mode).

continued...

Parameter	Value	Default	Description
		177.556818 MHz (Stratix 10)	This frequency in turn determines the required transceiver data rate to support the calculated transmit and receive bandwidths.
User clock frequency output	Minimum: 50 MHz Maximum: Limited by the supported transceiver data rates	146.484375 MHz (Stratix V and Arria V GZ) 150 MHz (Arria 10)	Specifies the actual user clock frequency as produced by the fPLL or I/O PLL and is ideally the same as the required clock frequency. In certain very high precision situations where the desired user clock is provided up to higher decimal places, this value can vary slightly due to the fPLL or I/O PLL constraints. Change the required clock frequency to correct the issue if the minute variation is intolerable. (Arria 10, Stratix V and Arria V GZ devices only)
Transceiver data rate	required user clock frequency * overheads * 64	10.3125 Gbps (Stratix V and Arria V GZ) 10.312499 Gbps (Arria 10) 12.5 Gbps (Stratix 10)	The effective data rate at the output of the transceivers, incorporating transmission and other overheads. The parameter editor automatically calculates this value by adding the input data rate with transmission overheads to provide you with a selection of user clock frequency.
Aggregate user bandwidth	number of lanes * required user clock frequency * 64	18.3125 Gbps (Stratix V and Arria V GZ) 18.75 Gbps (Arria 10) 68.18 Gpbs (Stratix 10)	This value is derived by multiplying the number of lanes and user interface data rate.

Related Links

SerialLite III Parameter Editor on page 12

3.5 Transceiver Reconfiguration Controller for Stratix V and Arria V GZ Designs

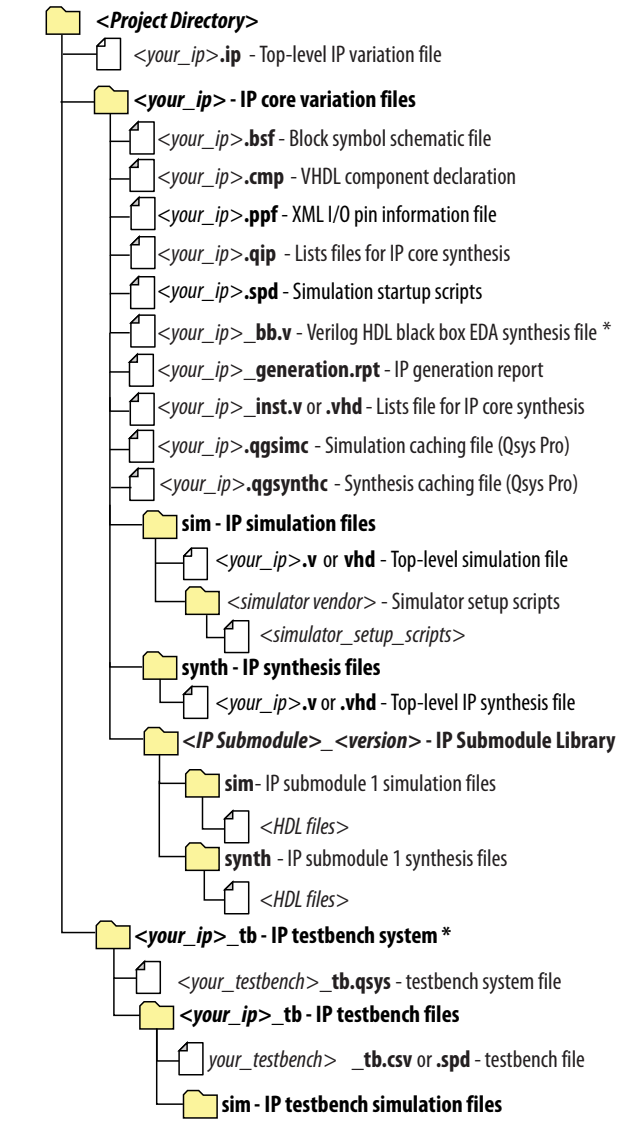
If your design targets Stratix V or Arria V GZ devices, the transceiver reconfiguration controller is not included in the generated IP core. To create a complete system, refer to the design example block diagram on how to connect the transceiver reconfiguration controller.

Note: If your design targets Arria 10 and Stratix 10 devices, the transceiver reconfiguration functionality is embedded inside the transceivers. The `phy_mgmt` bus interface connects directly to the Avalon Memory-Mapped (Avalon-MM) dynamic reconfiguration interface of the embedded Arria 10 and Stratix 10 Native PHY IP core. This interface is provided at the top level. For Quartus compilation design, create clock constraints for the `phy_mgmt_clk` and `reconfig_to_xcvr[0]` (for Stratix V and Arria V GZ) signals to avoid unconstrained clock warnings.

3.6 IP Core Generation Output (Quartus Prime Pro Edition)

The Quartus Prime software generates the following output file structure for individual IP cores that are not part of a Qsys Pro system.

Figure 3. Individual IP Core Generation Output (Quartus Prime Pro Edition)



* If supported and enabled for your IP core variation.

Table 5. Files Generated for IP Cores

File Name	Description
<your_ip>.ip	Top-level IP variation file that contains the parameterization of an IP core in your project. If the IP variation is part of a Qsys Pro system, the parameter editor also generates a .qsys file.
<your_ip>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you use in VHDL design files.
<your_ip>_generation.rpt	IP or Qsys Pro generation log file. Displays a summary of the messages during IP generation.

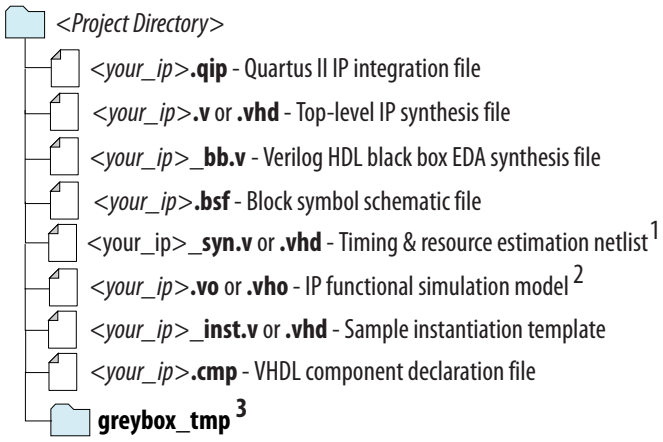
continued...

File Name	Description
<*your_ip*>.qgsimc (Qsys Pro systems only)	Simulation caching file that compares the .qsys and .ip files with the current parameterization of the Qsys Pro system and IP core. This comparison determines if Qsys Pro can skip regeneration of the HDL.
<*your_ip*>.qgsynth (Qsys Pro systems only)	Synthesis caching file that compares the .qsys and .ip files with the current parameterization of the Qsys Pro system and IP core. This comparison determines if Qsys Pro can skip regeneration of the HDL.
<*your_ip*>.qip	Contains all information to integrate and compile the IP component.
<*your_ip*>.csv	Contains information about the upgrade status of the IP component.
<*your_ip*>.bsf	A symbol representation of the IP variation for use in Block Diagram Files (.bdf).
<*your_ip*>.spd	Required input file for ip-make-simscript to generate simulation scripts for supported simulators. The .spd file contains a list of files you generate for simulation, along with information about memories that you initialize.
<*your_ip*>.ppf	The Pin Planner File (.ppf) stores the port and node assignments for IP components you create for use with the Pin Planner.
<*your_ip*>_bb.v	Use the Verilog blackbox (_bb.v) file as an empty module declaration for use as a blackbox.
<*your_ip*>_inst.v or _inst.vhd	HDL example instantiation template. Copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*your_ip*>.regmap	If the IP contains register information, the Quartus Prime software generates the .regmap file. The .regmap file describes the register map information of master and slave interfaces. This file complements the .sopcinfo file by providing more detailed register information about the system. This file enables register display views and user customizable statistics in System Console.
<*your_ip*>.svd	Allows HPS System Debug tools to view the register maps of peripherals that connect to HPS within a Qsys Pro system. During synthesis, the Quartus Prime software stores the .svd files for slave interface visible to the System Console masters in the .sof file in the debug session. System Console reads this section, which Qsys Pro queries for register map information. For system slaves, Qsys Pro accesses the registers by name.
<*your_ip*>.v <*your_ip*>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim™ LNL script msim_setup.tcl to set up and run a simulation.
aldec/	Contains a Riviera*-PRO script rivierapro_setup.tcl to setup and run a simulation.
/synopsys/vcs /synopsys/vcsmx	Contains a shell script vcs_setup.sh to set up and run a VCS* simulation. Contains a shell script vcsmx_setup.sh and synopsys_sim.setup file to set up and run a VCS MX* simulation.
/cadence	Contains a shell script ncsim_setup.sh and other setup files to set up and run an NCSIM simulation.
/submodules	Contains HDL files for the IP core submodule.
<*IP submodule*>/	For each generated IP submodule directory, Qsys Pro generates /synth and /sim sub-directories.

3.7 Files Generated for Intel IP Cores (Legacy Parameter Editor)

The Quartus Prime generates the following output for IP cores that use the legacy MegaWizard parameter editor.

Figure 4. IP Core Generated Files



Notes:
1. If supported and enabled for your IP variation
2. If functional simulation models are generated
3. Ignore this directory

3.8 Simulating

3.8.1 Simulating Intel FPGA IP Cores

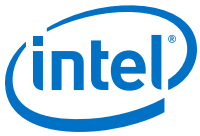
The Quartus Prime software supports IP core RTL simulation in specific EDA simulators. IP generation creates simulation files, including the functional simulation model, any testbench (or example design), and vendor-specific simulator setup scripts for each IP core. Use the functional simulation model and any testbench or example design for simulation. IP generation output may also include scripts to compile and run any testbench. The scripts list all models or libraries you require to simulate your IP core.

The Quartus Prime software provides integration with many simulators and supports multiple simulation flows, including your own scripted and custom simulation flows. Whichever flow you choose, IP core simulation involves the following steps:

1. Generate simulation model, testbench (or example design), and simulator setup script files.
2. Set up your simulator environment and any simulation script(s).
3. Compile simulation model libraries.
4. Run your simulator.

3.8.2 Simulation Parameters

After design generation, simulation files are available for you to simulate your design. To simulate your design, ensure that the SerialLite III Streaming IP core source and sink cores are both generated with the same parameters or are duplex cores.



- Stratix V and Arria V GZ files are located in the <*variation name*>**_sim** directory
- Arria 10 and Stratix 10 files are located in the <*variation name*> directory

The example testbench simulates the core using the user-specified configuration.

Table 6. Stratix V and Arria V GZ Testbench Default Simulation Parameters

Parameter	Default Value	Comments
user clock frequency output (`user_clock_frequency`)	Standard clocking: 145.98375 MHz Advanced clocking: 146.484375 MHz	—
Number of lanes (`lanes`)	2	—
Transceiver reference clock frequency (`pll_ref_freq`)	644.53125 MHz	—
Transceiver data rate (`data_rate`)	10312.5 Mbps	—
Meta frame length in words (`meta_frame_length`)	200	—
Simulation-specific parameters		
Total samples to transfer (`total_samples_to_transfer`)	2000	Total samples to transfer during simulation.
Mode (`mode`)	Continuous/burst	The testbench environment may automatically choose one of the modes depending on the random seed with which it is provided.
Skew insertion enable (`skew_insertion_enable`)	Yes	Skew testing is enabled. The testbench environment randomly inserts skew in the lanes within the range 0 - 107 UI.
Enable M20K ECC support (`ecc_enable`)	0	When set, the core is simulated with the ECC-enabled variant. Use the ECC-enabled variant in the test environment. When ECC mode is disabled, the two most significant bits of the error buses in the source or sink direction are *Don't Care*.

Table 7. Arria 10 Testbench Default Simulation Parameters

Parameter	Default Value	Comments
user clock frequency output (`user_clock_frequency`)	Standard clocking: 146.484375 MHz	—
Number of lanes (`lanes`)	2	—
Transceiver reference clock frequency (`pll_ref_freq`)	644.531187 MHz	—
Transceiver data rate (`data_rate`)	10.312499 Gbps	—
Meta frame length in words (`meta_frame_length`)	200	—
Simulation-specific parameters		
Total samples to transfer (`total_samples_to_transfer`)	2000	Total samples to transfer during simulation.

continued...

Parameter	Default Value	Comments
Mode (mode)	Continuous/burst	The testbench environment may automatically choose one of the modes depending on the random seed with which it is provided.
Skew insertion enable (skew_insertion_enable)	Yes	Skew testing is enabled. The testbench environment randomly inserts skew in the lanes within the range 0 - 107 UI.
Enable M20K ECC support (ecc_enable)	0	When set, the core is simulated with the ECC-enabled variant. Use the ECC enabled variant in the test environment. When ECC mode is disabled, the two most significant bits of the error buses in the source or sink direction are Don't Care.

Table 8. Stratix 10 Testbench Default Simulation Parameters

Parameter	Default Value	Comments
user clock frequency output (user_clock_frequency)	Standard clocking: 177.556818 MHz	—
Number of lanes (lanes)	6	—
Transceiver reference clock frequency (pll_ref_freq)	312.5 MHz	—
Transceiver data rate (data_rate)	12.5 Gbps	—
Meta frame length in words (meta_frame_length)	200	—
Simulation-specific parameters		
Total samples to transfer (total_samples_to_transfer)	2000	Total samples to transfer during simulation.
Mode (mode)	Continuous/burst	The testbench environment may automatically choose one of the modes depending on the random seed with which it is provided.
Skew insertion enable (skew_insertion_enable)	Yes	Skew testing is enabled. The testbench environment randomly inserts skew in the lanes within the range 0 - 107 UI.
Enable M20K ECC support (ecc_enable)	0	When set, the core is simulated with the ECC-enabled variant. Use the ECC enabled variant in the test environment. When ECC mode is disabled, the two most significant bits of the error buses in the source or sink direction are Don't Care.

For more information about Intel FPGA simulation models, refer to the Volume 3 of the Quartus Prime Handbook.

Related Links

Simulating Altera Designs

3.8.3 Simulating and Verifying the Design

By default, the parameter editor generates simulator-specific scripts containing commands to compile, elaborate, and simulate Intel FPGA IP models and simulation model library files. You can copy the commands into your simulation testbench script, or edit these files to add commands for compiling, elaborating, and simulating your design and testbench.

Table 9. Intel FPGA IP Core Simulation Scripts

Simulator	File Directory	Device Family	Script
ModelSim - Intel FPGA Edition ModelSim - Intel FPGA Starter Edition	*<variation name>*_**sim/mentor**	Stratix V Arria V GZ	**msim_setup.tcl** [1]
	<variation name>/**sim/mentor**	Arria 10 Stratix 10	
VCS	*<variation name>*_**sim/synopsys/vcs**	Stratix V Arria V GZ	**vcs_setup.sh**
	<variation name>/**sim/synopsys/vcs**	Arria 10 Stratix 10	
VCS MX	*<variation name>*_**sim/synopsys/vcsmx**	Stratix V Arria V GZ	**vcsmx_setup.sh** **synopsys_sim.setup**
	<variation name>/**sim/synopsys/vcsmx**	Arria 10 Stratix 10	
NCSim	*<variation name>*_**sim/cadence**	Stratix V Arria V GZ	**ncsim_setup.sh**
	<variation name>/**sim/cadence**	Arria 10 Stratix 10	
Aldec Riviera	*<variation name>*_**sim/aldec**	Stratix V Arria V GZ	**rivierapro_set.tcl**
	<variation name>/**sim/aldec**	Arria 10 Stratix 10	

Related Links

- Simulating Altera Designs
  For more information about Altera simulation models.
- Getting Started with Quartus Prime Simulation Using the ModelSim-Altera Software

1 If you did not set up the EDA tool option— which enables you to start third-party EDA simulators from the Quartus Prime software—run this script in the ModelSim-Intel FPGA Simulator Tcl console (not in the Quartus Prime software Tcl console) to avoid any errors.

4 SerialLite III Streaming IP Core Functional Description

The SerialLite III Streaming IP core implements a protocol that defines streaming data encapsulation at the link layer and data encoding at the physical layer. This protocol integrates transparently with existing hardware and provides a reliable data transfer mechanism in applications that do not need additional layers between the data link and application.

4.1 IP Core Architecture

The SerialLite III Streaming IP core has three variations:

- *Source*—formats streaming data from the user application and transmits the data over serial links.
- *Sink*—receives the serial stream data from serial links, removes any formatting information, and delivers the data to the user application.
- *Duplex*—composed of both the source and sink cores. The streaming data can be transmitted and received in both directions.

All three variations include the Intel Transceiver Native PHY IP core (Arria 10 and Stratix 10 devices) or Interlaken PHY IP core (Stratix V and Arria V GZ devices) that utilizes hardened PCS and PMA modules. The source and sink cores use the Native PHY or Interlaken PHY IP core in simplex mode, and the duplex core uses the Native PHY or Interlaken PHY IP core in duplex mode.

Table 10. IP Core Variant and Function

Core	Function
Source	• Data encapsulation • Generation and insertion of Idle Control Words • Lane striping for multi-lane link • User synchronization and burst marker insertion
Sink	• Multi-lane alignment • Data encapsulation removal • Idle Control Words removal • Lane de-striping • User synchronization and burst marker demultiplexing
Duplex	• Data encapsulation and decapsulation • Generation and removal of Idle Control Words • User synchronization and burst marker insertion and deletion

The simplex and duplex cores support the following clocking schemes:

- *Standard clocking*—this mode is for pure streaming designs in which the core provides input/output clocks to drive the user logic.
- *Advanced clocking*—this mode allows the core's input interface to be clocked with the user-preferred clock by trading-off pure streaming operation.

Figure 5. SerialLite III Streaming Simplex Core (Standard Clocking)

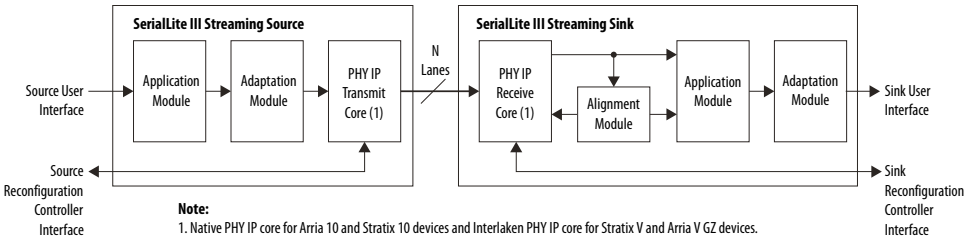
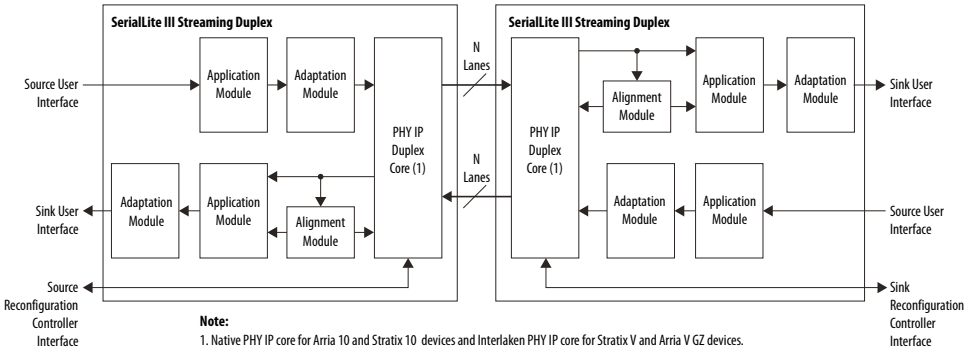


Figure 6. SerialLite III Streaming Duplex Core (Standard Clocking)



The block diagram for advanced clocking is similar to standard clocking, except that it does not include the adaptation module at the sink user interface.

Related Links

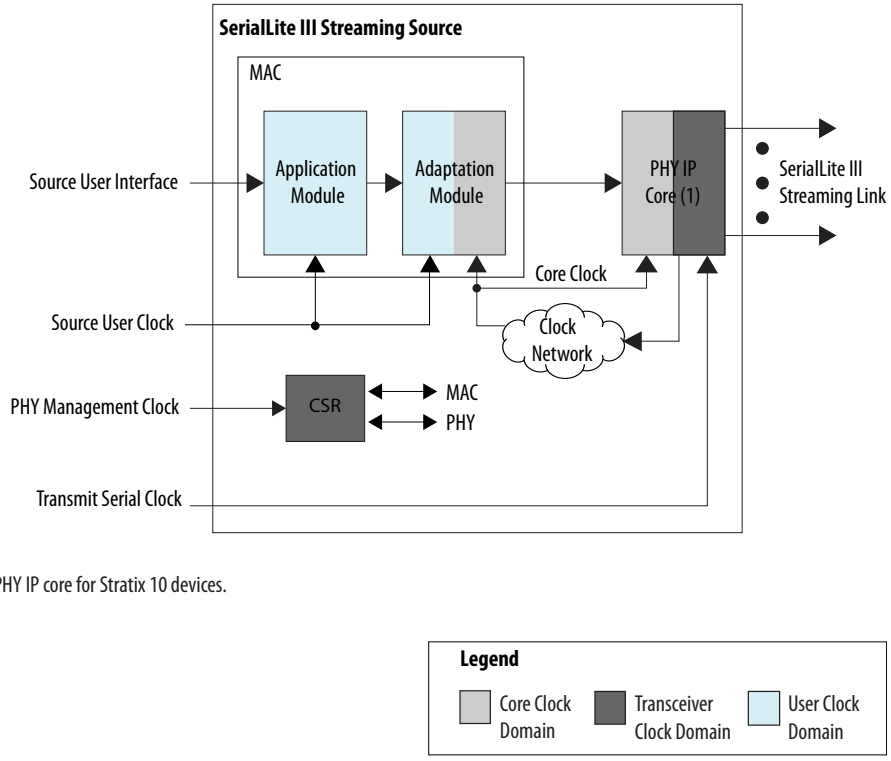
- Arria 10 Transceiver PHY User Guide
  For more information about the Arria 10 Native PHY IP core.
- Stratix 10 H-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 H-Tile Native PHY IP core.
- Stratix 10 L-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 L-Tile Native PHY IP core.
- Altera Transceiver PHY IP Core User Guide

4.1.1 SerialLite III Streaming Source Core

The source core consists of four major functional blocks (the implementation varies depending on the clocking mode):

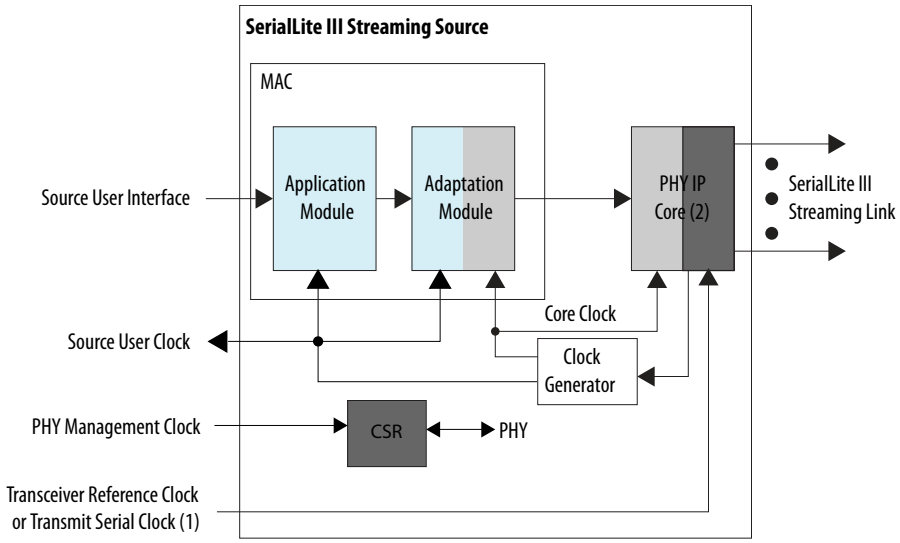
- Clock generator (in the standard clocking mode for Arria 10, Stratix V, and Arria V GZ devices)
- Source application module
- Source adaptation module
- Native PHY IP TX core - Interlaken mode (Arria 10 and Stratix 10 devices)
- Interlaken PHY IP TX core (Stratix V and Arria V GZ devices)

Figure 7. Stratix 10 SerialLite III Streaming Source Core (Standard Clocking and Advanced Clocking Modes)



Notes:
1. Native PHY IP core for Stratix 10 devices.

Figure 8. SerialLite III Streaming Source Core (Standard Clocking Mode for Arria 10, Stratix V, Arria V GZ devices)



Notes:

1. Transceiver reference clock for Stratix V and Arria V GZ devices; transmit serial clock for Arria 10 devices.
2. Native PHY IP core for Arria 10 devices and Interlaken PHY IP core for Stratix V and Arria V GZ devices.

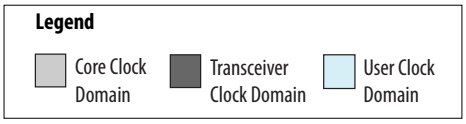
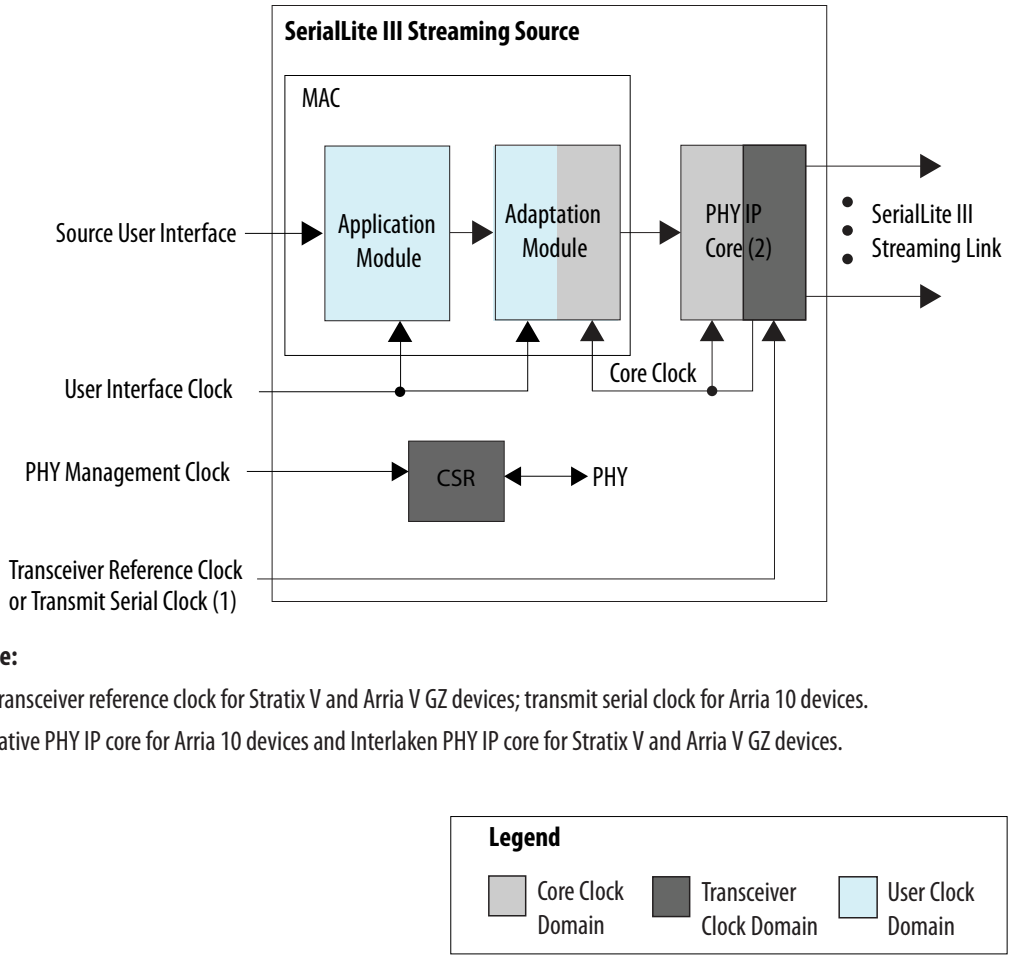


Figure 9. SerialLite III Streaming Source Core (Advanced Clocking Mode for Arria 10, Stratix V, Arria V GZ devices)



Note:

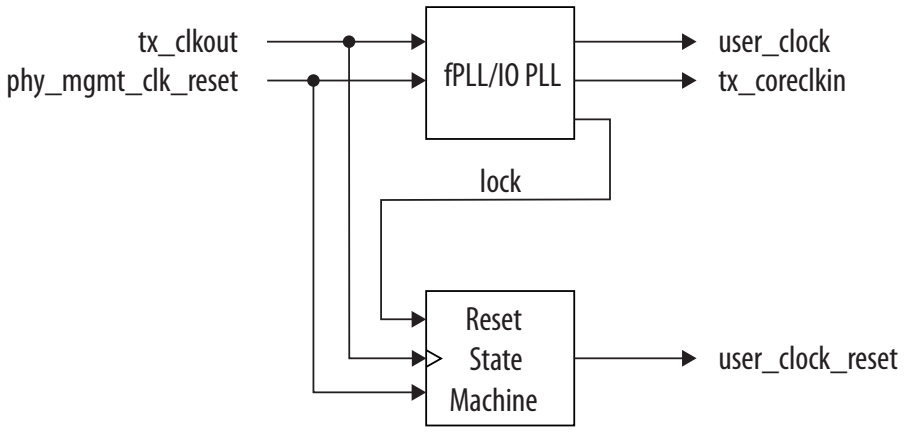
1. Transceiver reference clock for Stratix V and Arria V GZ devices; transmit serial clock for Arria 10 devices.
2. Native PHY IP core for Arria 10 devices and Interlaken PHY IP core for Stratix V and Arria V GZ devices.

4.1.1.1 Source Clock Generator

The clock generator in the source core synthesizes the user clock (`user_clock`) and core clock signals (`tx_coreclockin`) from the Native PHY IP core (Arria 10 devices) or Interlaken PHY IP (Stratix V and Arria V GZ devices) core's output clock signal (`tx_clkout`). This clock generator consists of a fPLL (Stratix V and Arria V GZ) or I/O PLL (Arria 10) and a state machine responsible for clocks generation and reset sequencing. The `user_clock_reset` is not released until the fPLL or I/O PLL is locked. The module is used in the standard clocking mode only.

Note: For Stratix 10 devices, the `tx_clkout` signal provides the clock for core clock signal (`tx_coreclockin`) because there is no clock generator module in the IP core.

Figure 10. Clock Generator Block Diagram



- For all Stratix V and Arria V GZ devices, the fPLL generates the `user_clock/user_clock_tx` and `tx_coreclkin` based on fixed ratios determined by the SerialLite III Streaming parameter editor.
- For Arria 10 devices, the I/O PLL generates the `user_clock/user_clock_tx` based on a fixed ratio, however, the `tx_coreclkin` operates at the same frequency as `tx_clkout`.

Related Links

Sink Clock Generator on page 32

4.1.1.2 Source Application Module

The application module performs the following functions:

- *Burst encapsulation*—inserts burst control words into the data stream to define the beginning and the end of streaming data bursts.
- *Idle insertion*—inserts idle control words (in the standard clocking mode) into all lanes of the data stream interface.

4.1.1.3 Source Adaptation Module

This module provides adaptation logic between the application module and the Native PHY IP core (Arria 10 and Stratix 10 devices) or Interlaken PHY IP (Stratix V and Arria V GZ devices) core. The adaptation module performs the following functions:

- *Rate adaptation*—includes a dual-clock FIFO buffer to cushion the Interlaken PHY IP core's bursty read requests and to provide a streaming user write interface. The FIFO also transfers streaming data between the `user_clock` and `tx_coreclkin` clock domains.
- *Control signal translation*—include state machines that map the control signal semantics on the framing interface [2] to the semantics of the Native PHY or Interlaken PHY IP core TX interface.
- *Non-user idle insertion*—inserts non-user idle control words in the absence of user data to manage the minimum data rate requirements of the Interlaken protocol. The control words are removed by the sink adaptation module in the SerialLite III link partner.
- *ECC correction and ECC fatal error detection*– for Arria 10, Stratix V and Arria V GZ devices only.

4.1.1.4 Interlaken PHY IP TX Core or Native PHY IP TX Core - Interlaken Mode

For Arria 10 and Stratix 10 devices, this block is an instance of the Native PHY IP core configured for Interlaken - TX only operation. The PMA width for Interlaken mode is 64 bits.

For Stratix V and Arria V GZ devices, the Interlaken PHY IP TX core is an instance of the Interlaken PHY IP core configured for TX only operation. The PMA width for Interlaken mode is 40 bits. The core requires a Transceiver Reconfiguration Controller for transceiver calibration. The number of channels programmed for configuration in the Transceiver Reconfiguration Controller depends on the IP core's operation mode. For example,

- if the design is a simplex RX only design, the reconfiguration interfaces is equal to the number of lanes.
- if the design is a simplex TX only design or a duplex design, the reconfiguration interfaces is equal to the number of lanes x 2.

Related Links

- Arria 10 Transceiver PHY User Guide
  For more information about the Arria 10 Native PHY IP core.
- Stratix 10 H-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 H-Tile Native PHY IP core.
- Stratix 10 L-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 L-Tile Native PHY IP core.
- Altera Transceiver PHY IP Core User Guide
  For more information about the Interlaken PHY IP core and how to dynamically reconfigure the PHY.

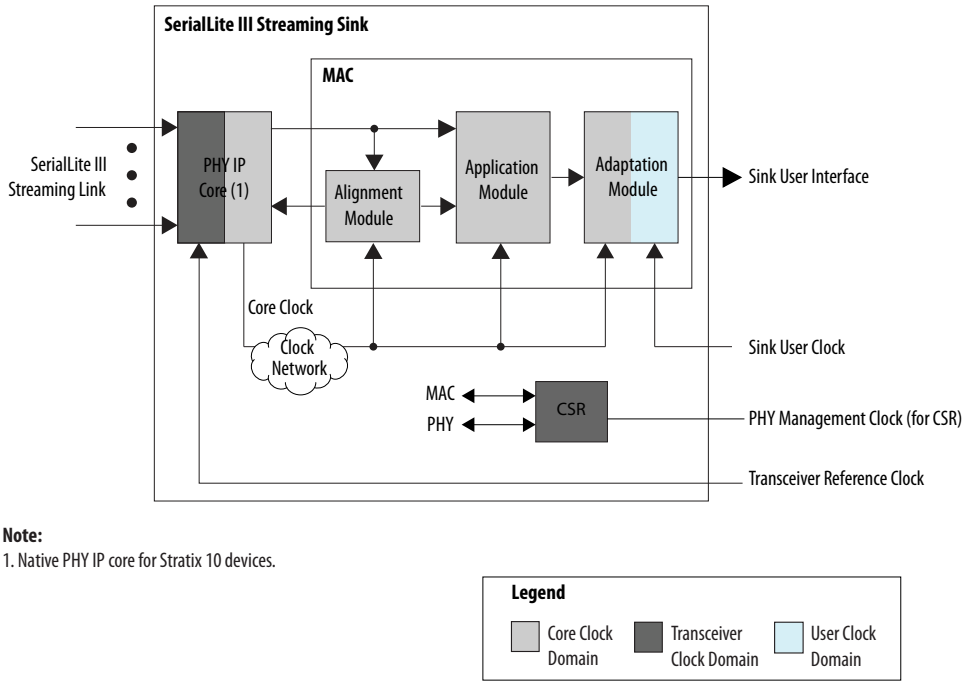
2 The framing interface is to frame every data burst with the Start of Burst, Sync, and End of Burst, and sequence them to the PHY interface.

4.1.2 SerialLite III Streaming Sink Core

The sink core consists of five major functional blocks:

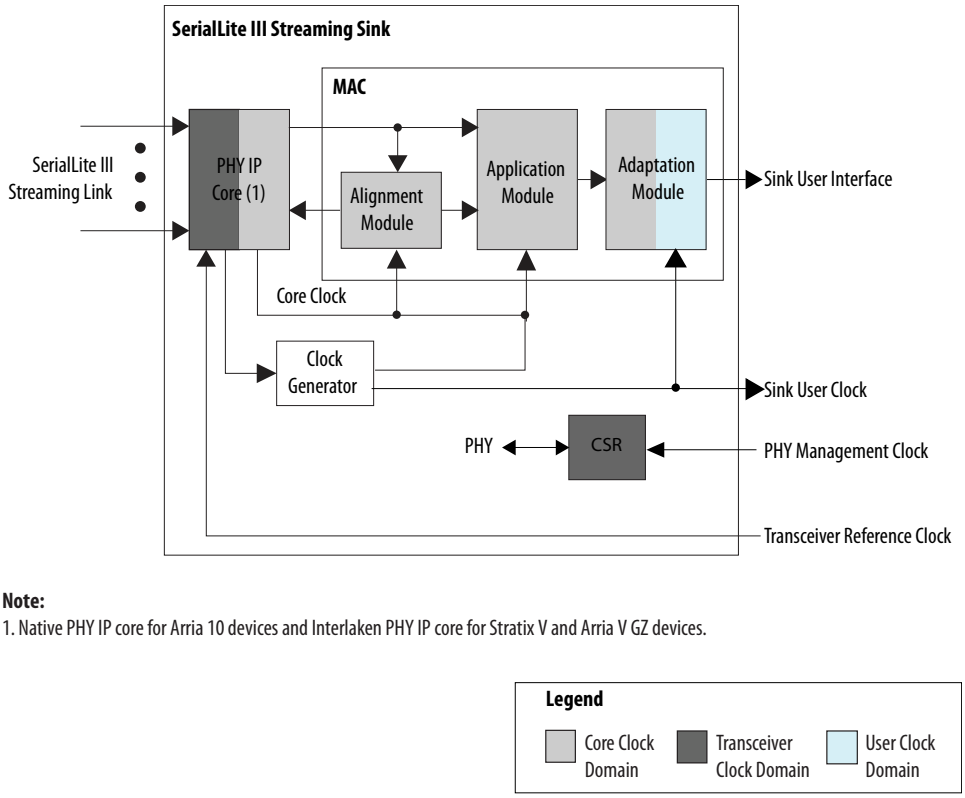
- Native PHY IP RX core - Interlaken mode (Arria 10 and Stratix 10 devices)
- Interlaken PHY IP RX core (Stratix V or Arria V GZ devices)
- Lane alignment module
- Clock generator (in the standard clocking mode for Arria 10, Stratix V, and Arria V GZ devices)
- Sink adaptation module (standard clocking mode only)
- Sink application module

Figure 11. Stratix 10 SerialLite III Streaming Sink Core (Standard Clocking Mode)



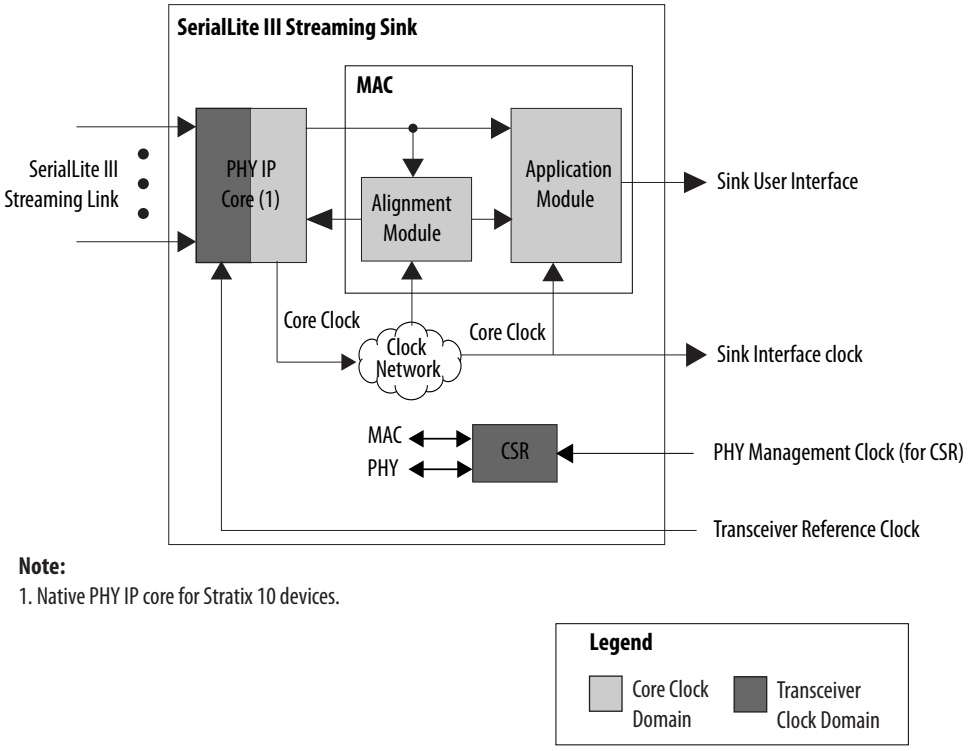
Note:
1. Native PHY IP core for Stratix 10 devices.

Figure 12. SerialLite III Streaming Sink Core (Standard Clocking Mode for Arria 10,Stratix V, Arria V GZ devices)



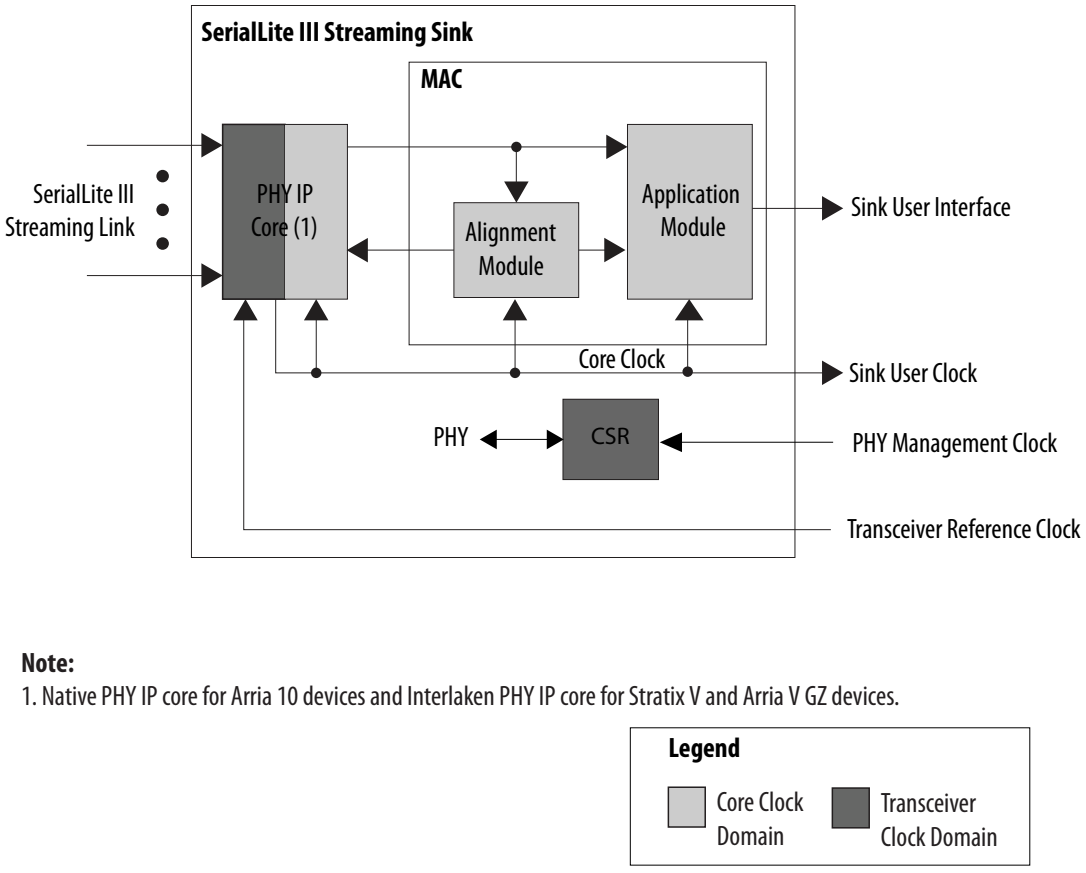
Note:
1. Native PHY IP core for Arria 10 devices and Interlaken PHY IP core for Stratix V and Arria V GZ devices.

Figure 13. Stratix 10 SerialLite III Streaming Sink Core (Advanced Clocking Mode)



Note:
1. Native PHY IP core for Stratix 10 devices.

Figure 14. SerialLite III Streaming Sink Core (Advanced Clocking Mode for Arria 10,Stratix V, Arria V GZ devices)



Note:
1. Native PHY IP core for Arria 10 devices and Interlaken PHY IP core for Stratix V and Arria V GZ devices.

4.1.2.1 Sink Clock Generator

The clock generator is similar to the clock generator in the source core, and is only instantiated in standard clocking mode. The clock generator synthesizes the user clock (`user_clock`) and core clock (`rx_coreclkin`) signals from the Native PHY IP core (Arria 10 devices) or Interlaken PHY IP (Stratix V and Arria V GZ devices) core's output clock signal. The clock generator consists of a fPLL or I/O PLL and a state machine responsible for clock generation and reset sequencing.

Note: For Stratix 10 devices, the `rx_clkout` signal provides the clock for core clock signal (`rx_coreclockin`) because there is no clock generator module in the IP core.

- For all Stratix V and Arria V GZ devices, the I/O PLL generates the `user_clock/user_clock_rx` and `rx_coreclkin` based on fixed ratios determined by the IP core's parameter editor.
- For Arria 10 devices, the I/O PLL generates the `user_clock/user_clock_rx` based on a fixed ratio, however, the `rx_coreclkin` operates at the same frequency as `rx_clkout`.

Related Links

Source Clock Generator on page 26

4.1.2.2 Sink Application Module

The sink application module performs the following functions:

- Strips the Interlaken protocol bursts encapsulation from the received serial data stream and sends the data to the sink adaptation module.
- Decodes idle control words inserted by the source application module when the data stream is not available and indicates the data unavailability at the source by deasserting the output valid signal at the user interface.
- Strips Interlaken framing layer symbols and diagnostic control words from the data stream (*Interlaken framing layer stripping*).

The encapsulation stripping process removes burst control words that define the beginning and the end of streaming data bursts from the data stream. This process adjusts the received data stream to repack the data words into a contiguous sequence.

- In the standard clocking mode (pure streaming), the decoding process checks the received data stream to detect idle control words that the source application module inserts. When the sink application module detects the idle control words, it deasserts the valid signal on the user interface until it receives valid user streaming data.
- In the advanced clocking mode, the sink application module does not insert or delete any idle words. Instead, the sink application module deasserts the output valid signal to indicate an absence of data coming from the sink adaptation module.

4.1.2.3 Sink Adaptation Module

The sink adaptation module provides rate adaptation logic between the application module and the streaming interface. The adaptation module implements the following functions:

- In standard clocking mode, the FIFO buffers help transfer data between the `rx_coreclkin` and `user_clock` domains.
- *Interlaken framing layer stripping*—strips Interlaken framing layer symbols and diagnostic control words from the data stream.

4.1.2.4 Lane Alignment Module

The lane alignment module interfaces with the Native PHY or Interlaken PHY IP core to access incoming data. This module removes lane skew from the incoming serial data streams and aligns various lanes using the Interlaken's synchronization marker. After alignment is achieved, the module continuously monitors the synchronization markers in the Interlaken metaframes for any loss of alignment.

4.1.2.5 Interlaken PHY IP RX Core or Native PHY IP RX Core - Interlaken Mode

For Arria 10 and Stratix 10 devices, this block is an instance of the Native PHY IP core configured for Interlaken - RX only operation. The PMA width for Interlaken mode is 64 bits.

For Stratix V and Arria V GZ devices, the Interlaken module is an instance of the Interlaken PHY IP core configured for RX only operation, and is generated by the Quartus Prime parameter editor. The core requires a Stratix V Transceiver Reconfiguration Controller for transceiver calibration. The interface size is initially equal to the number of transceiver channels that the sink core uses, which is the number of lanes. The PMA width is 40 bits.

Related Links

- Arria 10 Transceiver PHY User Guide
  For more information about the Arria 10 Native PHY IP core.
- Stratix 10 H-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 H-Tile Native PHY IP core.
- Stratix 10 L-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 L-Tile Native PHY IP core.
- Altera Transceiver PHY IP Core User Guide
  For more information about the Interlaken PHY IP core.

4.1.3 SerialLite III Streaming Duplex Core

For Arria 10 and Stratix 10 devices, the duplex core consists of source and sink cores interfaced with the Native PHY IP core in Interlaken mode.

For Stratix V and Arria V GZ devices, the duplex core is composed of source and sink cores interfaced with the Interlaken PHY IP in duplex mode.

4.1.4 Interlaken PHY IP Duplex Core or Native PHY IP Duplex Core - Interlaken Mode

For Arria 10 and Stratix 10 devices, this block is an instance of the Native PHY IP core configured for duplex Interlaken operation. The PMA width for Interlaken mode is 64 bits.

For Stratix V and Arria V GZ devices, the Interlaken module is an instance of the Interlaken PHY IP core configured for duplex operation, and is generated by the Quartus Prime parameter editor. The core requires a Stratix V/Arria V GZ Transceiver Reconfiguration Controller for transceiver calibration. The duplex core initially requires as many reconfiguration interfaces as the number of lanes that the IP core uses plus one for the TX PLL. The PMA width is 40 bits

Related Links

- Arria 10 Transceiver PHY User Guide
  For more information about the Arria 10 Native PHY IP core.
- Stratix 10 H-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 H-Tile Native PHY IP core.
- Stratix 10 L-Tile Transceiver PHY User Guide
  For more information about the Stratix 10 L-Tile Native PHY IP core.
- Altera Transceiver PHY IP Core User Guide
  For more information about the Interlaken PHY IP Core.

4.1.5 Stratix 10 versus Arria 10, Stratix V, and Arria V GZ Variations

The table summarizes the differences between the SerialLite III Streaming IP core in Stratix 10 devices and Arria 10, Stratix V, and Arria V GZ devices.

Table 11. Differences between Stratix 10 and Arria 10, Stratix V, or Arria V GZ

Implementation	Stratix 10	Arria 10, Stratix V or Arria V GZ
Internal clock generator for Standard Clocking Mode	Not included. User clock is provided by users. Use HSSI refclk to drive FPLL to generate the user clock. You must share the HSSI refclk with the transceiver TX PLL refclk when implementing asynchronous clocking in your design.	Included. For Stratix V or Arria V GZ devices, the IP core uses FPLL to generator the user clock. For Arria 10 devices, the IP core uses IOPLL to generate the user clock.
Control Status Registers (CSR) for MAC	Included.	Not included. Only CSR for transceiver is available.
Interrupts	Included.	Not included.

4.1.6 Arria 10 versus Stratix V and Arria V GZ Variations

The Arria 10 transceiver is different than the Stratix V or Arria V GZ transceiver. Therefore, the SerialLite III IP core is implemented differently for these device families, and the example testbenches are also different.

Table 12. Differences between Arria 10 and or Arria V GZ Transceivers

Implementation	Arria 10	Stratix V or Arria V GZ
Transceiver PLL	Not included	Included
Transceiver Reconfiguration Controller	Not required	Required
Example Testbench	Generated dynamically (same configuration as the IP core instance except the Burst Gap parameter)	Generated dynamically (same configuration as the IP core instance except the Burst Gap parameter)
Hardware Demonstration Design Example	Included	Included (for Stratix V only)

When you create an instance of the IP core, it dynamically generates an example testbench. This testbench has the same configuration as the IP core instance except for the Burst Gap parameter.

For Arria 10 devices, the Native PHY IP core (Interlaken mode) requires an external transmit PLL. Instantiate the external transceiver PLLs and then connect the transmit serial clock output to the `tx_serial_clk` input. The Seriallite III Streaming IP core uses a transmit serial clock input bus (`tx_serial_clk`) and `tx_pll_locked` input to connect the external transmit PLL to the Arria 10 Native PHY IP core. Refer to the *Arria 10 Transceiver PHY User Guide* for more information.

Related Links

- Signals on page 47
  The following tables list all the input and output signals of the SerialLite III Streaming IP core.
- Arria 10 Transceiver PHY User Guide

For more information about the Arria 10 Native PHY IP core.

- Altera Transceiver PHY IP Core User Guide
  For more information about the Interlaken PHY IP Core.

4.2 Clock Domains

The SerialLite III Streaming IP core contains different clock domains, depending on the clocking mode. In addition to these clock domains, there are another four clock domains in isolation within the transceivers.

Table 13. SerialLite III Streaming IP Core Clock Domains and Signals

Clock Domain		Description	Standard Clocking Mode	Advanced Clocking Mode
Source Core	user_clock	Source user interface clock	Yes	Yes
	phy_mgmt_clk	Source Native PHY or Interlaken PHY IP core reconfiguration interface clock (Arria 10, Stratix V, and Arria V GZ devices). Source Native PHY or Interlaken PHY IP core reconfiguration interface and MAC CSR clock (Stratix 10 devices).	Yes	Yes
	pll_ref_clk	Source transceiver reference clock (Stratix V and Arria V GZ devices only)	Yes	Yes
	tx_coreclkin	Source core clock (Arria 10, Stratix V, and Arria V GZ devices only)	Yes	—
	tx_serial_clk	Transmit transceiver clock (Arria 10 and Stratix 10 devices only)	Yes	Yes
Sink Core	user_clock	Sink user interface clock	Yes	—
	phy_mgmt_clk	Sink Native PHY or Interlaken PHY IP core reconfiguration interface clock (Arria 10, Stratix V, and Arria V GZ devices). Sink Native PHY IP core reconfiguration interface and MAC CSR clock (Stratix 10 devices).	Yes	Yes
	xcvr_pll_ref_clk	Sink transceiver reference clock	Yes	Yes
	rx_cdr_refclk	Transceiver reference clock (Arria 10 and Stratix 10 only)	Yes	Yes
	rx_coreclkin	Sink core clock (Arria 10, Stratix V, and Arria V GZ only)	Yes	—
Duplex Core	user_clock_tx	Source user interface clock For Stratix 10 devices, the IP core use this clock to derive the maximum user clock frequency.	Yes	Yes
	user_clock_rx	Sink user interface clock	Yes	—
	phy_mgmt_clk	Native PHY or Interlaken PHY IP core reconfiguration interface clock (Arria 10, Stratix V, and Arria V GZ devices). Native PHY IP core reconfiguration interface and MAC CSR clock (Stratix 10 devices).	Yes	Yes
	xcvr_pll_ref_clk	Transceiver reference clock	Yes	Yes
	rx_cdr_refclk	Transceiver reference clock (Arria 10 and Stratix 10 devices only)	Yes	Yes

continued...

Clock Domain		Description	Standard Clocking Mode	Advanced Clocking Mode
	tx_coreclkin	Source core clock (Arria 10, Stratix V, and Arria V GZ devices only)	Yes	—
	rx_coreclkin	Sink core clock (Arria 10, Stratix V, and Arria V GZ devices only)	Yes	—
	tx_serial_clk	Transmit transceiver clock (Arria 10 and Stratix 10 devices only)	Yes	Yes

4.2.1 Core Clocking

The SerialLite III Streaming IP core comes with standard and advanced clocking modes; which you can select in the parameter editor.

Table 14. Comparing Standard and Advanced Clocking Modes

Resource	Standard Mode	Advanced Mode	Description
Source user clocking	For Arria 10, Stratix V, and Arria V devices, this clock is generated by the core. For Stratix 10 devices, this clock is provided by user.	Provided by user	If the PPM difference between the generated and user clocks is not acceptable, use the advanced clocking mode.
MAC fPLL	Uses one fPLL or I/O PLL per direction	Does not use fPLLs or I/O PLLs (Arria 10, Stratix V and Arria V GZ only)	If the design uses many fPLLs or I/O PLLs and clock crossing is an issue in the user environment, use the advanced clocking mode.
Transmission overhead	1.1 x *<input data rate>*	*<Interlaken Overhead>* x *<input data rate>*	The advanced clocking mode overhead is less than the standard clocking mode overhead.

4.2.1.1 Standard Clocking Mode

In the standard clocking mode, the SerialLite III Streaming IP core sink user interface will continue to stream as long as the source does not de-assert the data valid signal.

For Arria 10, Stratix V and Arria V GZ devices, the SerialLite III Streaming IP core generates the user clock at both the source and sink to drive the user interface.

For Stratix 10 devices, you are required to provide the user clock at both the source and sink to drive the user interface. The user clock is no longer generated as an output from the IP core.

In this mode, you specify the user clock frequency through the SerialLite III Streaming parameter editor. The Quartus Prime software then automatically calculates the reference clock coming from the Native PHY or Interlaken PHY IP core.

For Stratix 10 devices, this reference clock is connected to a global clock network which generates the core clock for the SerialLite III Streaming IP core. For Arria 10, Stratix V and Arria V GZ devices, the Quartus Prime software then automatically calculates the reference clock coming from the Native PHY or Interlaken PHY IP core and the two clock outputs from the fPLL in the clock generator module.

After the calculation, the Quartus Prime software provides a list of transceiver reference clock values for you to select. For Arria 10, Stratix V and Arria V GZ devices, the SerialLite III Streaming IP core generates the user clock output with value

identical to the user clock frequency that you specify depending on the clock constraints. This output clock signal is not generated for Stratix 10 devices. The Quartus Prime software shows the generated user clock value as well as transceiver reference clock values.

Figure 15. Stratix 10 SerialLite III Streaming IP Core Block Diagram in Standard Clocking Mode

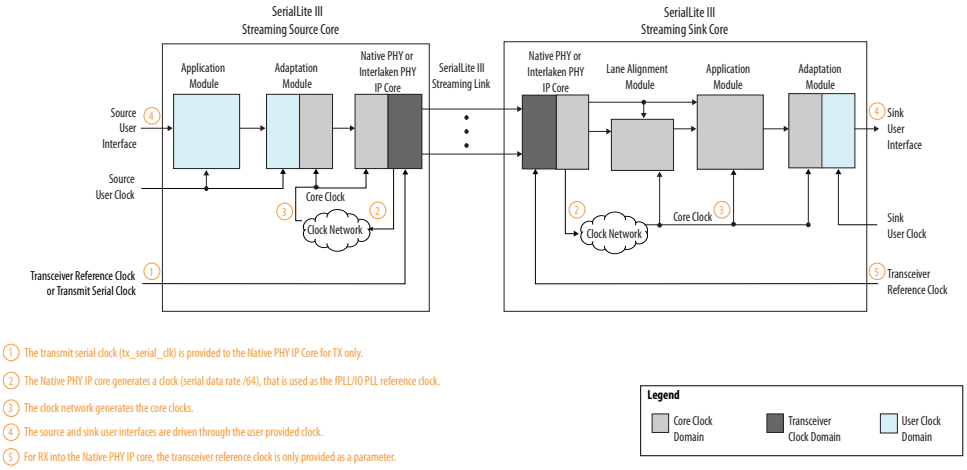
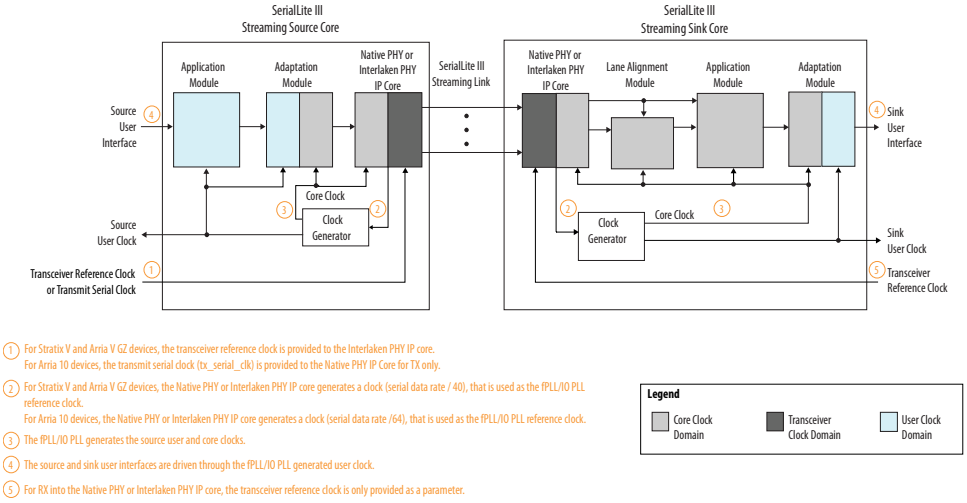


Figure 16. SerialLite III Streaming IP Core Block Diagram in Standard Clocking Mode



Note: The SerialLite III Streaming IP core uses the transmit serial clock bus (`tx_serial_clk`) and the `tx_pll_locked` signal to connect the external transmit PLL to the Arria 10 and Stratix 10 Native PHY IP core.

Related Links

Transmission Overheads and Lane Rate Calculations on page 41

4.2.1.2 Advanced Clocking Mode

The advanced clocking mode allows the user to use a user-specified clock to interface with the source core. For devices prior to Stratix 10, this mode is useful when PPM differences between the user clock (generated by the fPLL or I/O PLL) and the user's interface clock are intolerable.

Similar to the standard clocking mode, you must specify the user clock frequency through the SerialLite III Streaming parameter editor. Based on the user clock frequency value, the Quartus Prime software automatically calculates the lane rate and the core clock.

The parameter editor provides guidance in selecting a source user clock frequency that meets the transceiver data rate constraints. For more information about the lane rate calculation, refer to the "Transmission Overheads and Lane Rate Calculations" section.

The core operates at higher clock rates in Advanced Clocking Mode. Therefore, when operating in this mode, it may be difficult to close timing at higher data rates (for example, 12 to 15 G) and/or number of lanes. You can implement the following qsf assignment when seeing timing recovery violations from sink coreclkin reset synchronizer to the sink transfer paths:

```
set_instance_assignment -name GLOBAL_SIGNAL OFF -to
*seriallite_iii_streaming*clock_gen:sink_clock_gen|dp_sync:coreclkin_reset_sync|
dp_sync_regstage:dp_sync_stage_2*o*
```

Figure 17. Stratix 10 SerialLite III Streaming IP Core Block Diagram in Advanced Clocking Mode

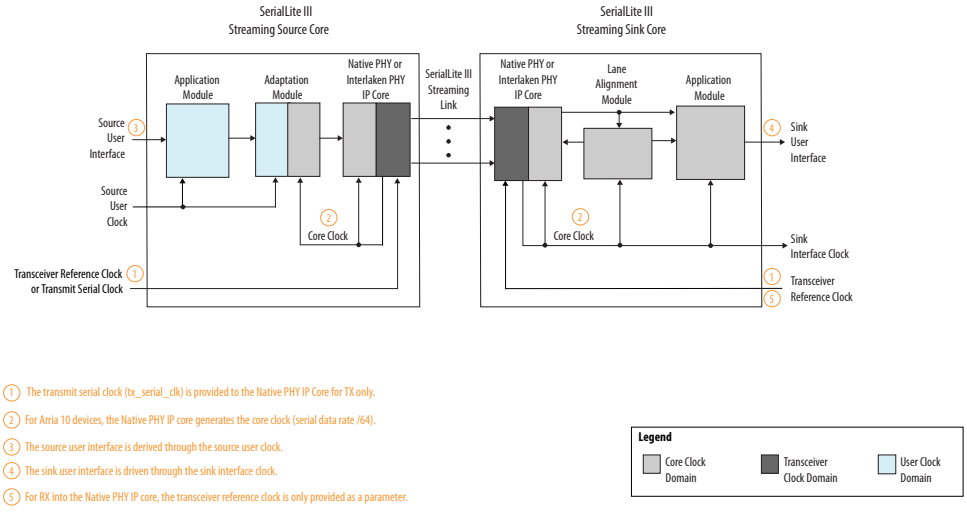
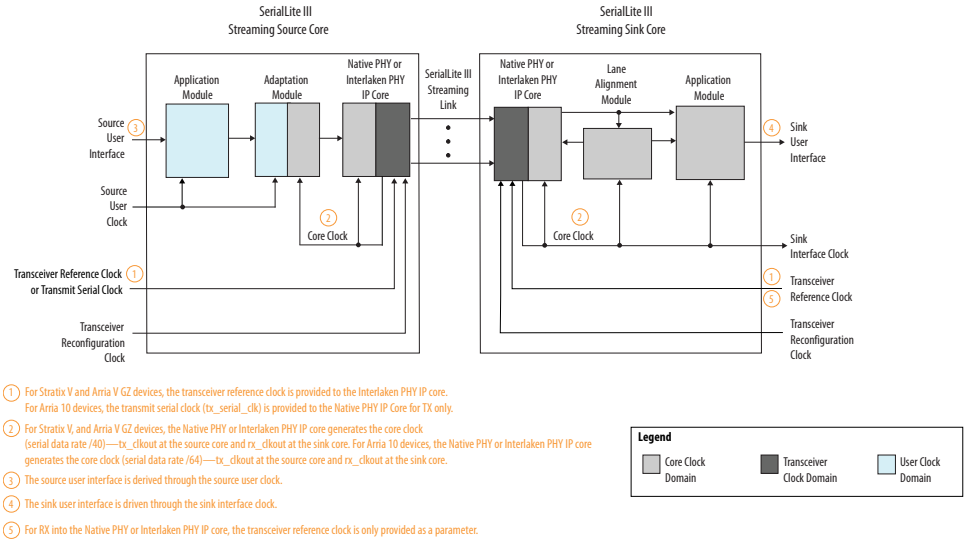


Figure 18. SerialLite III Streaming IP Core Block Diagram in Advanced Clocking Mode



Note: The SerialLite III Streaming IP core uses the transmit serial clock bus (`tx_serial_clk`) and the `tx_pll_locked` signal to connect the external transmit PLL to the Arria 10 and Stratix 10 Native PHY IP core.

4.2.2 Core Latency

The table below lists the latency measurement for the SerialLite III Streaming duplex core in standard and advanced clocking mode. An average value is taken from a set of samples during hardware testing.

For a loopback scenario, the core latency measurement is based on the round trip latency from the TX core input to RX core output.

Table 15. Latency Measurement for Duplex Core

Device	Clocking Mode	Parameters		Latency (ns)
		Number of Lanes	Per-Lane Data Rate (Mbps)	
Arria 10	Standard	5	17,400	174.064
	Advanced	5	17,400	154.996
Stratix V, Arria V GZ	Standard	5	10,312.50	320.964
	Advanced	5	10,312.50	292.712
Stratix 10	Standard	6	12,500	304.128
	Advanced	6	12,500	272.810

Note: To calculate the latency for 17,400 Mbps per lane data rate, an average value was taken from a set of samples. For duplex advanced clocking mode, the latencies varied more in simulation.

4.3 Transmission Overheads and Lane Rate Calculations

The SerialLite III Streaming IP core lane data rate (transceiver data rate) is composed of the input data rate and transmission overheads.

```
Lane Rate = Input Data Rate × Transmission Overheads
```

The parameter editor uses the above equation to ensure that the lane rate is within the maximum supported transceiver lane rates. This puts an upper limit on the input data rate or the user clock frequency, where the user clock frequency equates to:

```
User Clock Frequency = Input Data Rate/64
```

The SerialLite III Streaming IP core uses the Interlaken protocol for transferring data and therefore incurs encoding and metaframe overheads. In the standard clocking mode, the IP core employs an fPLL or I/O PLL for clock generation. To ensure that the fPLL or I/O PLL generates the clock as close as possible to the user clock that you have specified, the fPLL or I/O PLL incurs additional overheads. The transmission overheads can thus be derived in the following functions:

```
Transmission Overheads = Maximum (Interlaken Overheads, fPLL or I/O PLL Overheads)
```

where,

```
Interlaken Overheads = 67/64 × (MetaFrame Length) / (MetaFrame length - 4)
```

To ensure the Interlaken interoperability as well as user clocking requirements, the fPLL or I/O PLL overheads in the standard clocking mode are chosen to be slightly higher than the Interlaken overheads.

The 40-bit PMA interface supports Stratix V and Arria V GZ devices:

```
Lane Data Rate in Standard Clocking Mode = User Clock Frequency × 1.76 × 40 >
Input Data Rate * Interlaken Overheads
```

The 64-bit PMA interface supports Arria 10 and Stratix 10 devices:

```
Lane Data Rate in Standard Clocking Mode = User Clock Frequency × 1.1 × 64 >
Input Data Rate * Interlaken Overheads
```

Note: Calculations with 40 and 64 for the lane data rate in standard clocking mode are for the PMA width interfaces.

Using these calculations, the following overhead can be derived:

```
Transmission Overheads in standard clocking mode = 1.1
```

Note: Assuming maximum metaframe overhead with a metaframe size of 200, the standard clocking mode overheads are independent of Interlaken overheads. For more details, refer to the SerialLite III data efficiency calculator.

Tip: You can obtain the SerialLite III Streaming MegaCore Function Data Efficiency Calculator for 28 nm Intel FPGA devices from your local Intel sales representative or by emailing SLIII_support@altera.com.

Therefore, the lane rate in the standard clocking mode equals:

```
Lane Rate = Input Data Rate × 1.1
```

In the advanced clocking mode, the transmission overheads equals the Interlaken overheads because no fPLL is present. Therefore, the lane rate in advanced clocking mode equals:

```
Lane Rate = Input Data Rate × Interlaken overheads
```

4.4 Reset

Arria 10, Stratix V and Arria V GZ Reset Scheme

Each core has a separate active high reset signal, `core_reset`, that asynchronously resets all logic in the core.

Each core also includes the Native PHY or Interlaken PHY IP reset signal, `phy_mgmt_clk_reset`. This reset signal must be on the same clock domain as the clock used to drive the reconfiguration controllers, `phy_mgmt_clk`. The Native PHY or Interlaken PHY IP core requires the assertion of this reset signal to synchronize with the reconfiguration controller reset signal.

Note: Intel recommends using the same reset signals for both the Native PHY or Interlaken PHY IP core and the reconfiguration controller.

If the `phy_mgmt_clk_reset` or `core_reset` signal is asserted on the source core, the sink will deassert the `link_up_rx` signal. However, there is no additional indication on the sink core whether the last transmitted burst has bad data. The source core reinitializes the internal reset sequence when the `phy_mgmt_clk_reset` or `core_reset` signal is deasserted. Once the internal reset sequence is complete, the core asserts the `link_up_tx` signal to indicate that the core initialization is complete and is ready to transmit user data.

Note: Intel recommends that you wait for an additional 30 µs on the source core before sending any valid Avalon-ST data cycle. This is to ensure that the sink core has sufficient time to assert the `link_up_rx` signal.

Stratix 10 Reset Scheme

For Stratix 10 devices, the IP core uses the `phy_mgmt_clk_reset` signal to reset all the modules in the IP core and `user_clock_reset` signal to reset the user clock domain modules e.g. transmit and receive FIFO.

You may also trigger a reset to the IP core by writing into the reset controller register in the PHY:

- Writing 1 to CSR address 0x02E2 bit 3 to initiates a TX digital reset and bit 1 to initiates a RX digital reset
- Writing 1 to CSR address 0x02E2 bit 2 to initiates a TX analog reset and bit 0 to initiates a RX analog reset

Use the following guidelines to provide a proper reset to the IP core:

- Use the same reset signals for both the source and sink user clock domain modules.
- Synchronize the `user_clock_reset` signals with `phy_mgmt_clock_reset` signal assertion.
- Use the `phy_mgmt_clk_reset` signal to reset the configuration and status registers.
- Ensure all clocks are toggling in a correct rate before de-asserting any reset signals.

4.5 Link-Up Sequence

Refer to the topics on source and sink core link debugging for information about the transmit and receive core link-up sequence.

Related Links

- Source Core Link Debugging (Arria 10, Stratix 10, Arria V GZ, and Stratix V) on page 85
- Sink Core Link Debugging (Arria 10, Stratix 10, Arria V GZ, and Stratix V) on page 86

4.6 CRC-32 Error Injection

In the Quartus Prime software version 13.1 and later, the SerialLite III Streaming IP core supports CRC error injection with the 10G PCS CRC-32 generator. This feature enables corruption of the CRC-32 value of the CRC-32 generator.

To insert CRC errors for a given lane, the IP interface includes a CRC error injection control signal. Asserting this control signal inserts CRC errors for all the lanes and transceivers that have enabled support for error injection. You can enable the CRC error injection for a specific transceiver channel (SerialLite III lane) by programming the appropriate transceiver PCS CRAM bit. The provided example design demonstrates how set the respective CRAM bits using the Nios II processor.

This feature is supported in both burst and continuous modes of the SerialLite III Streaming IP core. For proper functionality of this feature, you must verify that you are adhering to the following steps:

1. Verify both `link_up_tx` and `link_up_rx` are asserted.
2. Verify the user can send/receive normal traffic.
3. After link up, program the transceiver register and verify the register bit (PCS CRAM bit) by reading.
4. With link up, toggle `crc_error_inject` port to high on the transmitter.
5. Monitor the `error_rx` output on the receiver.

Related Links

SerialLite III Streaming IP Core Design Example User Guide

4.7 FIFO ECC Protection

In the Quartus II software version 13.1 and later, the SerialLite III Streaming IP core can be protected from Single-Event Upset (SEU) changes using error correcting code (ECC) protection. You can enable this feature using the ECC protection option in the parameter editor. The ECC protection provides additional error status bits that tell you if the ECC was able to perform a correction from the SEU change or if an uncorrectable error has occurred.

Note: Enabling ECC protection incurs additional logic and latency overhead.

4.8 User Data Interface Waveforms

The following waveforms apply to the SerialLite III Streaming IP core source user interface in source-only and duplex cores.

Figure 19. Source Waveform for Burst Mode

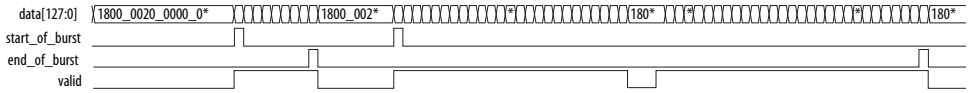


Figure 20. Source Waveform for Burst Mode (Sync)

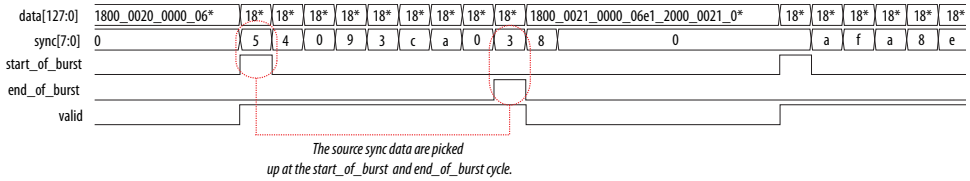
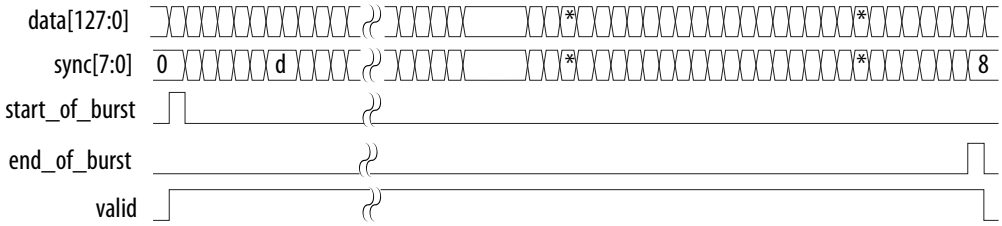
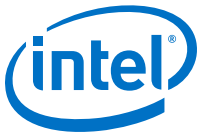


Figure 21. Source Waveform for Continuous Mode





- `start_of_burst` pulses for one clock cycle, indicating that the data burst starts at that clock cycle.
- `end_of_burst` pulses for one clock cycle, indicating that the data burst ends at that clock cycle.
- The `valid` signal indicates valid data. It should be turned off between two data bursts that are between the current data burst's `end_of_burst` clock cycle and next data burst's `start_of_burst` clock cycle. The `valid` signal can be pulled low in the middle of a data burst transferring between the same data burst's `start_of_burst` and `end_of_burst`, indicating non-valid data at that clock cycle.
- The `sync` vector is used in burst mode. It is valid only when `start_of_burst` and `valid` are high. Multiple logical channel is time-multiplexed into physical channels. Sync vector can be used to store the logical channel number that the burst targets. The logical channel number is multiplexed into the sync vector during the `start_of_burst`. The value is embedded into the data and sent over to the receiving party. The sink can extract the channel number from `start_of_burst` data bus to output on the sync vector of the sink. The sync vector can also be used to include empty information which indicates invalid data at the `end_of_burst`. In this case, the empty value is multiplexed into the sync vector during `end_of_burst`. The data is again embedded inside and sent over to the receiving party. The sink extracts the information and output on the sync vector of the sink.

The following waveforms apply to the sink user interface in sink-only and duplex cores.

Figure 22. Sink Waveform for Burst Mode

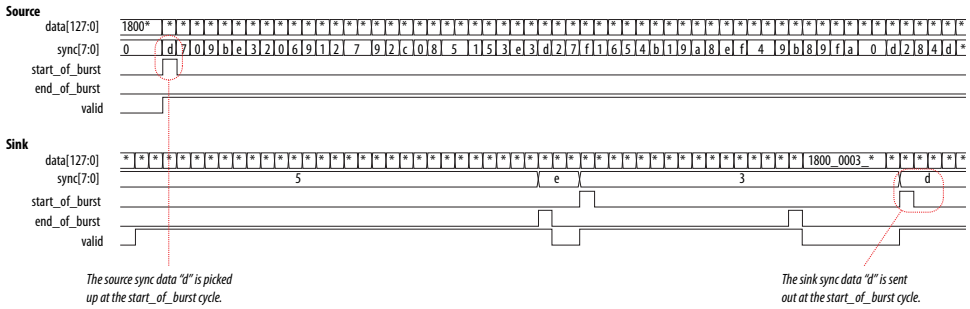
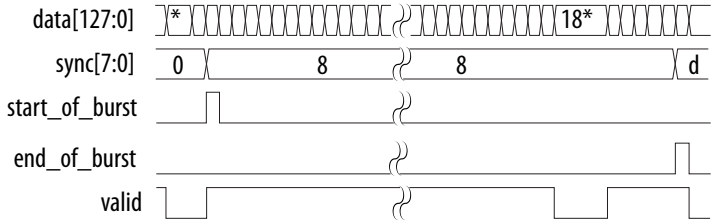


Figure 23. Sink Waveform for Continuous Mode



- `start_of_burst` pulses for one clock cycle, indicating that the data burst starts at that clock cycle.
- `end_of_burst` pulses for one clock cycle, indicating that the data burst ends at that clock cycle.
- The `valid` signal indicates valid data. It is turned off between two data bursts that are between the current data burst's `end_of_burst` clock cycle and the next data burst's `start_of_burst` clock cycle. The valid signal can be pulled low in the middle of a data burst after a data burst's `start_of_burst` and before the data burst's `end_of_burst`, indicating non-valid data at that clock cycle.
- The `sync` vector is used in burst mode. The sync data picked up at the source's `start_of_burst` high cycle is sent out at the sink as shown in the waveform. Multiple logical channel is time-multiplexed into physical channels. Sync vector can be used to store the logical channel number that the burst targets. The logical channel number is multiplexed into the sync vector during the `start_of_burst`. The value is embedded into the data and sent over to the receiving party. The sink can extract the channel number from `start_of_burst` data bus to output on the sync vector of the sink. The sync vector can also be used to include empty information which indicates invalid data at the `end_of_burst`. In this case, the empty value is multiplexed into the sync vector during `end_of_burst`. The data is again embedded inside and sent over to the receiving party. The sink extracts the information and output on the sync vector of the sink.

4.9 Signals

The following tables list all the input and output signals of the SerialLite III Streaming IP core.

Related Links

- Altera Transceiver PHY IP Core User Guide
  More information about the Interlaken PHY IP core signals.
- Loopback Modes
  More information about pre- and post-CDR rx to tx serial loopback modes in the Transceiver Reconfiguration Controller IP Core.

4.9.1 Signals for Stratix V and Arria V GZ Devices

Table 16. SerialLite III Streaming IP Core Source Core Signals

Signal	Width	Clock Domain	Direction	Description
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL that is available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	This signal is the reference clock for the transceivers.
user_clock	1	N/A	Input/ Output	Clock for data transfers across the source core interface. • Input: Using advanced clocking mode • Output: Using standard clocking mode
user_clock_reset	1	user_clock	Input/ Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. In the advanced clocking mode, asserts this signal to reset the adaptation module FIFO buffer. • Input: Using advanced clocking mode • Output: Using standard clocking mode
link_up	1	user_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data	64x*N*	user_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync	8	user_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and the end of a burst are captured and transported across the link. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.

continued...

Signal	Width	Clock Domain	Direction	Description
valid	1	user_clock	Input	This single bit signal indicates that the transmitted streaming data is valid.
start_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error	4	user_clock	Output	This vector indicates an error or overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow • Bit 1: An SEU error occurred and was corrected (ECC enabled) Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled) Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
crc_error_inject	1	user_clock	Input	This signal forces CRC-32 errors when CRC-32 error injection is enabled in the transceiver channels. The CRC-32 error injection is enabled via the transceiver reconfiguration controller.

Table 17. SerialLite III Streaming IP Core Sink Core Signals

Signal	Width	Clock Domain	Direction	Description
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL or I/O PLL that us available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset	1	user_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete in the standard clocking mode.
interface_clock	1	core_clock	Output	Clock for data transfer across the sink core interface in the advanced clocking mode.
interface_clock_reset	1	core_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete in the advanced clocking mode.

continued...

Signal	Width	Clock Domain	Direction	Description
link_up	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data. When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync	8	Standard clocking: user_clock Advanced clocking: core_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This single bit signal indicates that the data is valid.
start_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode, the core asserts this signal only once at the start of the data.
end_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.
error	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error

Table 18. SerialLite III Streaming IP Core Duplex Core Signals

Signal	Width	Clock Domain	Direction	Description
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL or I/O PLL that is available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock_tx	1	N/A	Input/ Output	Clock for data transfers across the transmit interface. • Input: Using advanced clocking mode • Output: Using standard clocking mode
user_clock_reset_tx	1	user_clock_tx	Input/ Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. In the advanced clocking mode, asserts this signal to reset the adaptation module FIFO buffer. • Input: Using advanced clocking mode • Output: Using standard clocking mode
interface_clock_reset_tx	1	core_clock	Output	In the advanced clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.
link_up_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data_tx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync_tx	8	Standard clocking: user_clock Advanced clocking: core_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and at the end of a burst are captured and transported across the link. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector indicates that the data is valid.
start_of_burst_tx	1	Standard clocking: user_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst.

continued...

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock		Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.
error_tx	4	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates an overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow • Bit 1: An SEU error occurred and was corrected (ECC enabled). Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled). Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
user_clock_rx	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset_rx	1	user_clock_rx	Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.
interface_clock_rx	1	core_clock	Output	Clock for data transfers across the sink core interface in the advanced clocking mode.
interface_clock_reset_rx	1	core_clock	Output	In the advanced clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.
link_up_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data. When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data_rx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync_rx	8	Standard clocking: user_clock Advanced clocking: core_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0.

continued...

Signal	Width	Clock Domain	Direction	Description
				Note: This vector is not associated with Interlaken channelization or flow control schemes.
valid_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates that the data is valid.
start_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error_rx	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error
crc_error_inject	1	Standard clocking: user_clock_tx Advanced clocking: core_clock_tx	Input	This signal is used for CRC-32 error injection.

Table 19. Interlaken PHY IP Core Signals and Native PHY IP Core Signals (Interlaken Mode)

Signal	Width	Clock Domain	Direction	Description
phy_mgmt_clk	1	N/A	Input	Clock input for the Avalon-MM PHY management interface within the Interlaken PHY IP core or Native PHY IP core. This signal also clocks the transceiver reconfiguration interface

continued...

Signal	Width	Clock Domain	Direction	Description
				and sequences the reset state machine in the clock generation logic.
phy_mgmt_clk_reset	1	phy_mgmt_clk	Input	Global reset signal that resets the entire IP including MAC, fPLL or I/O PLL (available in standard clocking mode), and Interlaken PHY IP core or Native PHY IP core. This signal is active high and level sensitive.
phy_mgmt_addr[8:0]	9	phy_mgmt_clk	Input	Control and status register (CSR) address for Stratix V and Arria V GZ devices.
phy_mgmt_writedata[31:0]	32	phy_mgmt_clk	Input	CSR write data.
phy_mgmt_readdata[31:0]	32	phy_mgmt_clk	Output	CSR read data.
phy_mgmt_write	1	phy_mgmt_clk	Input	Active high CSR write signal.
phy_mgmt_read	1	phy_mgmt_clk	Input	Active high CSR read signal.
phy_mgmt_waitrequest	1	phy_mgmt_clk	Output	CSR read or write request signal. When asserted, this signal indicates that the Avalon-MM slave interface is unable to respond to a read or write request.
reconfig_busy	1	phy_mgmt_clk	Input	For Stratix V and Arria V GZ devices, when asserted, this signal indicates that a reconfiguration operation is in progress and no further reconfiguration operations should be performed. You can monitor this signal to determine the status of the Transceiver Reconfiguration Controller.
reconfig_to_ xcvr	• Source core: 140x*N* • Sink core: 70x*N* • Duplex core: 140x*N*	phy_mgmt_clk	Input	Dynamic reconfiguration input for the Interlaken PHY IP. *N* represents the number of lanes.
reconfig_from_ xcvr	• Source core: 92x*N* • Sink core: 46x*N* • Duplex core: 92x*N*	phy_mgmt_clk	Output	Dynamic reconfiguration output for the Interlaken PHY IP. *N* represents the number of lanes.
tx_serial_data	*N*	—	Output	The serial output data from the core. *N* represents the number of lanes.
rx_serial_data	*N*	—	Input	The serial input data to the core. *N* represents the number of lanes.

4.9.2 Signals for Arria 10 Devices

Note: For Arria 10 devices, the phy_mgmt bus interface connects to the reconfiguration interface of the instantiated Native PHY IP core.

Table 20. SerialLite III Streaming IP Core Source Core Signals

Signal	Width	Clock Domain	Direction	Description
tx_serial_clk	*N*	N/A	Input	This signal is a high-speed serial clock input from the external transceiver PLL. The width is the same as the number of lanes specified in the parameter editor. Each bit of the vector corresponds to serial clock of the transmit channel. *N* represents the number of lanes.
tx_pll_locked	1	N/A	Input	This signal indicates that all external transceiver PLLs are locked. If more than one external transceiver PLL is required for higher lanes, each instantiation outputs a bit that indicates whether the PLL providing the high-speed clock for a corresponding transceiver has achieved its lock status. The pll_locked output signal from the external transceiver PLLs should be ANDed together before being input to the IP core.
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL that is available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	This signal is present but unused in source-only variations; tie this signal to 1'b0.
user_clock	1	N/A	Input/ Output	Clock for data transfers across the source core interface. • Input: Using advanced clocking mode • Output: Using standard clocking mode
user_clock_reset	1	user_clock	Input/ Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. In the advanced clocking mode, asserts this signal to reset the adaptation module FIFO buffer. • Input: Using advanced clocking mode • Output: Using standard clocking mode
interface_clock_reset	1	user_clock	Output	Clock for data transfer across the source core interface in the advanced clocking mode. Available only in Advanced Clocking Mode.
link_up	1	user_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data	64x*N*	user_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync	8	user_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and the end of a burst are captured and transported across the link. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.

continued...

Signal	Width	Clock Domain	Direction	Description
valid	1	user_clock	Input	This single bit signal indicates that the transmitted streaming data is valid.
start_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error	4	user_clock	Output	This vector indicates an error or overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow • Bit 1: An SEU error occurred and was corrected (ECC enabled) Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled) Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
crc_error_inject	1	user_clock	Input	This signal forces CRC-32 errors when CRC-32 error injection is enabled in the transceiver channels. The CRC-32 error injection is enabled via the transceiver reconfiguration controller.

Table 21. SerialLite III Streaming IP Core Sink Core Signals

Signal	Width	Clock Domain	Direction	Description
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL or I/O PLL that us available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset	1	user_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete in the standard clocking mode.
interface_clock	1	core_clock	Output	Clock for data transfer across the sink core interface in the advanced clocking mode.
interface_clock_reset	1	core_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.

continued...

Signal	Width	Clock Domain	Direction	Description
				Available only in Advanced Clocking Mode.
link_up	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data. When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync	8	Standard clocking: user_clock Advanced clocking: core_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This single bit signal indicates that the data is valid.
start_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode, the core asserts this signal only once at the start of the data.
end_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.
error	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error

Table 22. SerialLite III Streaming IP Core Duplex Core Signals

Signal	Width	Clock Domain	Direction	Description
tx_serial_clk	*N*	N/A	Input	This high-speed serial clock input from the external transceiver PLL. The width is the same as the number of lanes specified in the parameter editor. Each bit of the vector corresponds to serial clock of the transmit channel. *N* represents the number of lanes.
tx_pll_locked	1	N/A	Input	This signal indicates that all external transceiver PLLs are locked. If more than one external transceiver PLL is required for higher lanes, each instantiation outputs a bit that indicates whether the PLL providing the high-speed clock for a corresponding transceiver has achieved its lock status. The pll_locked output signal from the external transceiver PLLs should be ANDed together before being input to the IP core.
core_reset	1	N/A	Input	Asynchronous master reset for the core. Assert this signal high to reset the MAC layer, except for the fPLL or I/O PLL that is available in standard clocking mode. Intel recommends that you tie this signal to the phy_mgmt_clk_reset signal. When these two signals are tied together, the digital core, analog core, and the PLL core will be reset.
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock_tx	1	N/A	Input/ Output	Clock for data transfers across the transmit interface. • Input: Using advanced clocking mode • Output: Using standard clocking mode
user_clock_reset_tx	1	user_clock_tx	Input/ Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. In the advanced clocking mode, asserts this signal to reset the adaptation module FIFO buffer. • Input: Using advanced clocking mode • Output: Using standard clocking mode
interface_clock_reset_tx	1	core_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. Available only in Advanced Clocking Mode.
link_up_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data_tx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync_tx	8	Standard clocking: user_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and at the end of a burst are captured and transported across the link.

continued...

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock		The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector indicates that the data is valid.
start_of_burst_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.
error_tx	4	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates an overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow • Bit 1: An SEU error occurred and was corrected (ECC enabled). Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled). Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
user_clock_rx	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset_rx	1	user_clock_rx	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. Available only in Standard Clocking Mode.
interface_clock_rx	1	core_clock	Output	Clock for data transfers across the sink core interface in the advanced clocking mode.
interface_clock_reset_rx	1	core_clock	Output	The core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete. Available only in Advanced Clocking Mode.
link_up_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.

continued...

Signal	Width	Clock Domain	Direction	Description
				When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data_rx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync_rx	8	Standard clocking: user_clock Advanced clocking: core_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates that the data is valid.
start_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error_rx	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error
crc_error_inject	1	Standard clocking: user_clock_tx	Input	This signal is used for CRC-32 error injection.

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock_tx		

Table 23. Native PHY IP Core Signals (Interlaken Mode)

Signal	Width	Clock Domain	Direction	Description
phy_mgmt_clk	1	N/A	Input	Clock input for the Avalon-MM PHY management interface within the Interlaken PHY IP core or Native PHY IP core. This signal also clocks the transceiver reconfiguration interface and sequences the reset state machine in the clock generation logic.
phy_mgmt_clk_reset	1	phy_mgmt_clk	Input	Global reset signal that resets the entire IP including MAC, fPLL or I/O PLL (available in standard clocking mode), and Interlaken PHY IP core or Native PHY IP core. This signal is active high and level sensitive.
phy_mgmt_addr	10 + log2N], N=number of lanes	phy_mgmt_clk	Input	Control and status register (CSR) address for Arria 10 devices. The width depends on the number of lanes. The parameter editor determines the required width for you. You have to manually tie this extra bit [3]. • phy_mgmt_addr[msb] = 1: for Transceiver reconfiguration usage. • phy_mgmt_addr[msb] = 0: for soft CSR (the transceiver reset and loopback control CSR)
phy_mgmt_writedata[31:0]	32	phy_mgmt_clk	Input	CSR write data.
phy_mgmt_readdata[31:0]	32	phy_mgmt_clk	Output	CSR read data.
phy_mgmt_write	1	phy_mgmt_clk	Input	Active high CSR write signal.
phy_mgmt_read	1	phy_mgmt_clk	Input	Active high CSR read signal.
phy_mgmt_waitrequest	1	phy_mgmt_clk	Output	CSR read or write request signal. When asserted, this signal indicates that the Avalon-MM slave interface is unable to respond to a read or write request.
reconfig_busy	1	phy_mgmt_clk	Input	For Arria 10 devices, this signal is present but unused; tie this signal to 1'b0.
tx_serial_data	*N*	—	Output	The serial output data from the core. *N* represents the number of lanes.
rx_serial_data	*N*	—	Input	The serial input data to the core. *N* represents the number of lanes.

3 For more information about this bit, refer to the Interlaken PHY Registers table in the Altera Transceiver PHY IP Core User Guide.

4.9.3 Signals for Stratix 10 Devices

Table 24. SerialLite III Streaming IP Core Source Core Signals

Signal	Width	Clock Domain	Direction	Description
tx_serial_clk	*N*	N/A	Input	This signal is a high-speed serial clock input from the external transceiver PLL. The width is the same as the number of lanes specified in the parameter editor. Each bit of the vector corresponds to serial clock of the transmit channel. *N* represents the number of lanes.
tx_pll_locked	1	N/A	Input	This signal indicates that all external transceiver PLLs are locked. If more than one external transceiver PLL is required for higher lanes, each instantiation outputs a bit that indicates whether the PLL providing the high-speed clock for a corresponding transceiver has achieved its lock status. The pll_locked output signal from the external transceiver PLLs should be ANDed together before being input to the IP core.
xcvr_pll_ref_clk	1	N/A	Input	This signal is the reference clock for the transceivers.
user_clock	1	N/A	Input/ Output	Clock for data transfers across the source core interface. This is an input signal for standard and advanced clocking mode.
user_clock_reset	1	user_clock	Input/ Output	Asserts this signal to reset all the user clock domain module. Available only in Standard Clocking Mode.
link_up	1	user_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data	64x*N*	user_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync	8	user_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and the end of a burst are captured and transported across the link. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid	1	user_clock	Input	This single bit signal indicates that the transmitted streaming data is valid.
ready	1	user_clock	Output	On interfaces supporting backpressure, the source asserts ready to mark the cycles where transfers may take place. When this signal is asserted on cycle N, cycle (N + readLatency, where readLatency=0) is considered a ready cycle. This signal is only asserted after tx_link_up is asserted Leave unconnected if unused.
start_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst.

continued...

Signal	Width	Clock Domain	Direction	Description
				Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst	1	user_clock	Input	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error	4	user_clock	Output	This vector indicates an error or overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow • Bit 1: An SEU error occurred and was corrected (ECC enabled) Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled) Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
crc_error_inject	1	user_clock	Input	This signal forces CRC-32 errors when CRC-32 error injection is enabled in the transceiver channels. The CRC-32 error injection is enabled via the transceiver reconfiguration controller.
err_interrupt	1	phy_mgmt_clk	Output	This signal indicates if a transmit error occurs in the current transmission. The signal goes to '1' when any error status bit and its associated interrupt enabled bit have been set to '1'. It goes to '0' after all error status bits with interrupt enabled have been cleared. Leave unconnected if unused.

Table 25. SerialLite III Streaming IP Core Sink Core Signals

Signal	Width	Clock Domain	Direction	Description
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset	1	user_clock	Output	Asserts this signal to reset all the user clock domain module. Available only in Standard Clocking Mode.
interface_clock	1	core_clock	Output	Clock for data transfer across the sink core interface in the advanced clocking mode.
interface_clock_reset	1	core_clock	Output	The core asserts this signal when the phy_mgmt_clk_reset signal is high and deasserts this signal when the reset sequence is complete in the advanced clocking mode. Available only in Advanced Clocking Mode.
link_up	1	Standard clocking: user_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.

continued...

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock		When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync	8	Standard clocking: user_clock Advanced clocking: core_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This single bit signal indicates that the data is valid.
start_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode, the core asserts this signal only once at the start of the data.
end_of_burst	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.

continued...

Signal	Width	Clock Domain	Direction	Description
error	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error
ready	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	On interfaces supporting backpressure, the sink asserts ready to mark the cycles where transfers may take place. When this signal is asserted on cycle N, cycle (N + readLatency, where readLatency=0) is considered a ready cycle. If this signal de-asserts in the middle of a data transfer, it is considered an erroneous condition. Leave unconnected if unused.
err_interrupt	1	phy_mgmt_clk	Output	This signal indicates if a receive error as occur in the current transmission. The signal goes to '1' when any error status bit and its associated interrupt enabled bit have been set to '1'. It goes to '0' after all error status bits with interrupt enabled have been cleared. Leave unconnected if unused.

Table 26. SerialLite III Streaming IP Core Duplex Core Signals

Signal	Width	Clock Domain	Direction	Description
tx_serial_clk	*N*	N/A	Input	This high-speed serial clock input from the external transceiver PLL. The width is the same as the number of lanes specified in the parameter editor. Each bit of the vector corresponds to serial clock of the transmit channel. *N* represents the number of lanes.
tx_pll_locked	1	N/A	Input	This signal indicates that all external transceiver PLLs are locked. If more than one external transceiver PLL is required for higher lanes, each instantiation outputs a bit that indicates whether the PLL providing the high-speed clock for a corresponding transceiver has achieved its lock status. The pll_locked output signal from the external transceiver PLLs should be ANDed together before being input to the IP core.
xcvr_pll_ref_clk	1	N/A	Input	Reference clock for the transceivers.
user_clock_tx	1	N/A	Input	Clock for data transfers across the transmit interface.

continued...

Signal	Width	Clock Domain	Direction	Description
user_clock_reset_tx	1	user_clock_tx	Input	In the standard clocking mode, the core asserts this signal when the phy_mgmt_clk_reset signal is high and deasserts this signal when the reset sequence is complete. In the advanced clocking mode, asserts this signal to reset all user clock domain modules.
link_up_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.
data_tx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
sync_tx	8	Standard clocking: user_clock Advanced clocking: core_clock	Input	The sync vector is an 8 bit bus. The data value at the start of a burst and at the end of a burst are captured and transported across the link. The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	This vector indicates that the data is valid.
start_of_burst_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_tx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	When the core is in burst mode operation, assertion of this signal indicates that the information on the data vector is the end of a burst.
error_tx	4	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates an overflow in the source adaptation module's FIFO buffer. • Bit 0: Source adaptation module's FIFO buffer overflow

continued...

Signal	Width	Clock Domain	Direction	Description
				• Bit 1: An SEU error occurred and was corrected (ECC enabled). Don't care (ECC disabled) • Bit 2: An SEU error occurred and cannot be corrected (ECC enabled). Don't care (ECC disabled) • Bit 3: A burst gap error occured due to a mismatch in the *BURST GAP* parameter value and the gap between end of burst and start of burst.
err_interrupt_tx	1	phy_mgmt_clk	Output	This signal indicates if a transmit error occurs in the current transmission. The signal goes to '1' when any error status bit and its associated interrupt enabled bit have been set to '1'. It goes to '0' after all error status bits with interrupt enabled have been cleared. Leave unconnected if unused.
ready_tx	1	user_clock	Output	On interfaces supporting backpressure, the source asserts ready to mark the cycles where transfers may take place. When this signal is asserted on cycle N, cycle (N + readLatency, where readLatency=0) is considered a ready cycle. This signal is only asserted after tx_link_up is asserted Leave unconnected if unused.
user_clock_rx	1	N/A	Output	Clock for data transfers across the sink core interface in the standard clocking mode.
user_clock_reset_rx	1	user_clock_rx	Output	In the standard clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.
interface_clock_rx	1	core_clock	Output	Clock for data transfers across the sink core interface in the advanced clocking mode.
interface_clock_reset_rx	1	core_clock	Output	In the advanced clocking mode, the core asserts this signal when the core_reset signal is high and deasserts this signal when the reset sequence is complete.
link_up_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data. When this signal is deasserted, all values in the data_rx signal is invalid regardless of the valid_rx signal value. This means even when the valid_rx signal is asserted, the data_rx signal should be treated as invalid when link_up_rx is deasserted.
data_rx	64x*N*	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
sync_rx	8	Standard clocking: user_clock	Output	The sync vector is an 8 bit bus that reflrects the SYNC value received from the remote partner.

continued...

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock		The value at the end of a burst is to indicate the number of invalid 64-bit word in the previous data cycle. As such, for single-lane configuration, the value at the end of a burst is expected to be 0. *Note:* This vector is not associated with Interlaken channelization or flow control schemes.
valid_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates that the data is valid.
start_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the beginning of a burst. Because continuous mode is one long burst, in this mode the signal is asserted only once at the start of the data.
end_of_burst_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Output	When the core is in burst mode operation, asserting this signal indicates that the information on the data vector is the end of a burst. You can optionally send an end of burst signal at the end of continuous mode.
error_rx	*N*+5	Standard clocking: user_clock Advanced clocking: core_clock	Output	This vector indicates the state of the sink adaptation module's FIFO buffer. *N* represents the number of lanes: • [*N*+4]: An SEU error occurred and cannot be corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+3]: An SEU error occurred and was corrected (ECC enabled); Don't care (ECC disabled) Don't care (for advanced clocking mode) • [*N*+2]: FIFO buffer overflow Don't care (for advanced clocking mode) • [*N*+1]: Don't care. Tied to zero. • [*N*]: Loss of alignment • [*N*-1:0]: RX CRC 32 error
ready_rx	1	Standard clocking: user_clock Advanced clocking: core_clock	Input	On interfaces supporting backpressure, the sink asserts ready to mark the cycles where transfers may take place. When this signal is asserted on cycle N, cycle (N + readLatency, where readLatency =0) is considered a ready cycle If this signal de-asserts in the middle of a data transfer, it is considered an erroneous condition. Leave unconnected if unused.
err_interrupt_rx	1	phy_mgmt_clk	Output	This signal indicates if a receive error as occur in the current transmission. The signal goes to '1' when any error status bit and its associated interrupt enabled bit have been set to '1'. It goes to '0' after all error status bits with interrupt enabled have been cleared. Leave unconnected if unused.
crc_error_inject	1	Standard clocking: user_clock_tx	Input	This signal is used for CRC-32 error injection.

Signal	Width	Clock Domain	Direction	Description
		Advanced clocking: core_clock_tx		

Table 27. Native PHY IP Core Signals (Interlaken Mode)

Signal	Width	Clock Domain	Direction	Description
phy_mgmt_clk	1	N/A	Input	Clock input for the Avalon-MM PHY management interface within the Interlaken PHY IP core or Native PHY IP core. This signal also clocks the transceiver reconfiguration interface and sequences the reset state machine in the clock generation logic. Frequency ranges from 100 Hz - 150MHz.
phy_mgmt_clk_reset	1	phy_mgmt_clk	Input	Global reset signal that resets the entire IP including MAC, fPLL or I/O PLL (available in standard clocking mode), and Interlaken PHY IP core or Native PHY IP core. This signal is active high and level sensitive.
phy_mgmt_addr	11 + log2N, N=number of lanes	phy_mgmt_clk	Input	Control and status register (CSR) address for Stratix 10 devices.
phy_mgmt_writedata[31:0]	32	phy_mgmt_clk	Input	CSR write data.
phy_mgmt_readdata[31:0]	32	phy_mgmt_clk	Output	CSR read data.
phy_mgmt_write	1	phy_mgmt_clk	Input	Active high CSR write signal.
phy_mgmt_read	1	phy_mgmt_clk	Input	Active high CSR read signal.
phy_mgmt_waitrequest	1	phy_mgmt_clk	Output	CSR read or write request signal. When asserted, this signal indicates that the Avalon-MM slave interface is unable to respond to a read or write request.
tx_serial_data	*N*	—	Output	The serial output data from the core. *N* represents the number of lanes.
rx_serial_data	*N*	—	Input	The serial input data to the core. *N* represents the number of lanes.

4.10 Accessing Configuration and Status Registers

The Avalon-MM PHY management block within the Interlaken PHY IP core or Native PHY IP core includes master and slave interfaces. This component acts as a bridge. It transfers commands received on its Avalon-MM slave interface to its Avalon-MM port. This interface manages PCS and PMA modules, resets, error handling, and serial loopback controls. Refer to Configuration and Status Registers on page 89 for more information of registers that you can access using the Avalon-MM PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.

5 SerialLite III Streaming IP Core Clocking Guidelines

This section describes the SerialLite III Streaming IP core clocking architecture and usage models targeting streaming applications.

The SerialLite III Streaming IP core has two clocking options to support a variety of streaming applications:

- Standard Clocking Mode (SCM)
- Advanced Clocking Mode (ACM)

The following sections describe the clocking architectures for Arria 10, Stratix 10, and Stratix V series device families.

5.1 Clocking Structure for Stratix 10 Devices

Standard Clocking Mode

Unlike previous device generation, you are required to provide a user clock to drive the user interface for both source and sink core in Standard Clocking Mode in Stratix 10 devices. The SerialLite III Streaming IP core no longer generates a user clock for the user interface.

Figure below shows the source and sink variant clocking structure for standard clocking mode in Stratix 10 devices.

Figure 24. Standard Clocking Mode Structure in Stratix 10 Devices

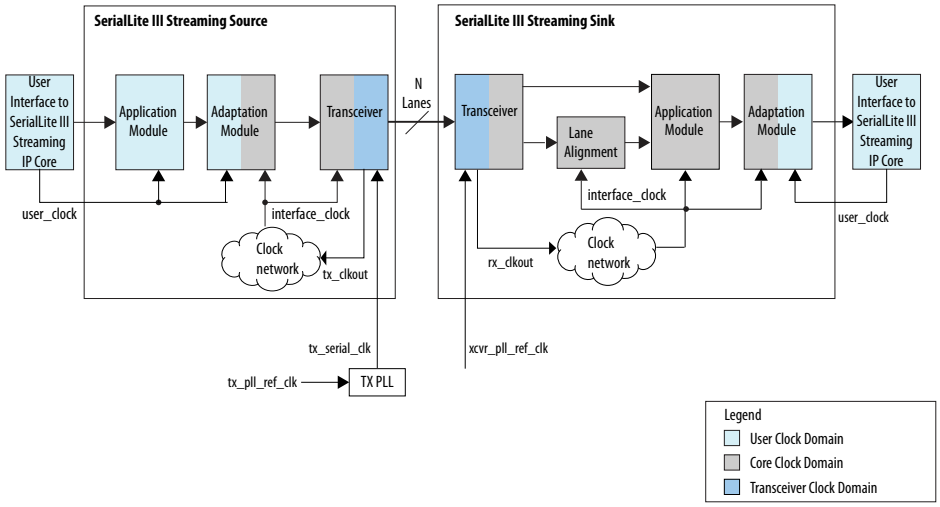


Table 28. Stratix 10 Clocks in Standard Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This is an input clock provided by the user to the IP. This clock should be used to clock the TX user application that drives the TX user interface.
tx_serial_clk	This clock should toggle at one-half the data rate of the transceiver lane. When you enter the `user_clock` frequency in the IP parameter editor, the per lane data rate is calculated. Use that value and divided it by two to determine the `tx_serial_clk`. You are required to instantiate the TX PLL, as shown in the figure above. An example of the TX PLL (ATX PLL) is generated with the IP core and is configured with the required reference clock and `tx_serial_clk`.
tx_clkout	This clock is not exposed to the user. The frequency of `tx_clkout` is the data rate divided by 64.
interface_clock	This clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP parameter editor and is the transceiver data rate divided by transceiver PCS-PMA width (Stratix 10: 64 bits).
Sink	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. The frequency of this clock should match the frequency of the `user_clock` in the Source variant. This is an input clock provided by the user to the IP. This clock should be used to clock the RX user application that drives the RX user interface.
xcvr_pll_ref_clk	This reference clock is used by the Clock Data Recovery (CDR) unit in the transceiver. It serves as a reference for the CDR to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
rx_clkout	This clock is not exposed to the user. The frequency of `rx_clkout` is the data rate divided by 64.
interface_clock	This clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP parameter editor and is the transceiver data rate divided by transceiver PCS-PMA width (Stratix 10: 64 bits).

Advanced Clocking Mode

Figure below shows the source and sink variant clocking structure for advanced clocking mode in Stratix 10 devices.

Figure 25. Advanced Clocking Mode Structure for Stratix 10 Devices

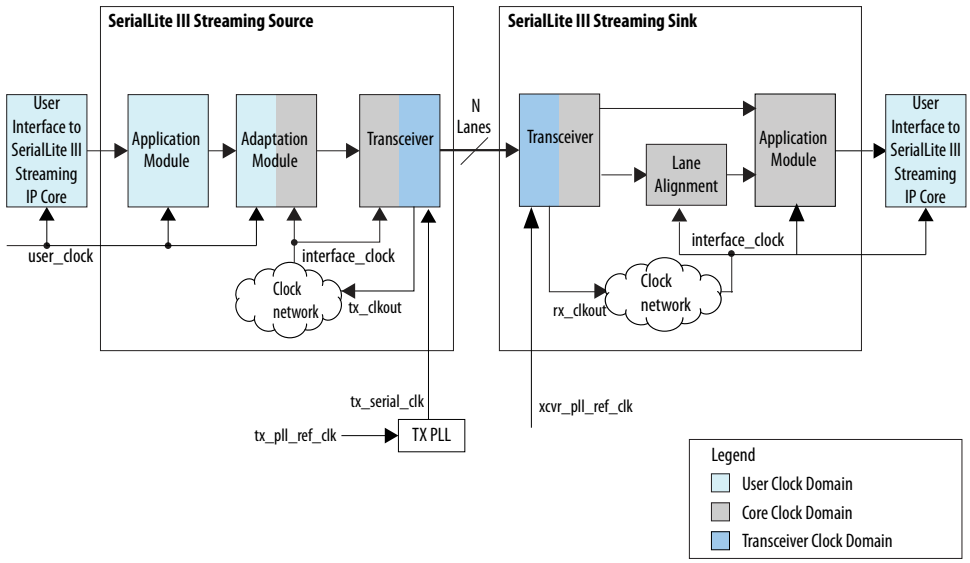


Table 29. Stratix 10 Clocks in Advanced Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This clock is an input to the IP core and you should toggle this at the specified frequency.
tx_serial_clk	This clock should toggle at one-half the data rate of the transceiver lane. When you enter the `user_clock` frequency in the IP parameter editor, the per lane data rate is calculated. Use that value and divided it by two to determine the `tx_serial_clk`. You are required to instantiate the TX PLL. An example of the TX PLL (ATX PLL) is generated with the IP core and is configured with the required reference clock and `tx_serial_clk` frequencies.
tx_clkout	This clock is not exposed to the user. The frequency of `tx_clkout` is the data rate divided by 64.
interface_clock	This is an internal clock and it is not exposed to the user. The frequency of this clock is derived from the transceiver data rate. The frequency is lane data rate divided by 64.
Sink	
xcvr_pll_ref_clk	This reference clock is used by the CDR unit in the transceiver. It serves as a reference for the CDR to be able to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
rx_clkout	This clock is not exposed to the user. The frequency of `rx_clkout` is the data rate divided by 64.
interface_clock	This clock is derived from the transceiver data. It is lane data rate divided by 64.

5.1.1 Choosing TX PLL Type for Stratix 10 Devices

These are the guidelines for choosing the appropriate type of TX PLL for Stratix 10 devices.

Table 30. PLL Data Rate Performance (Preliminary for Stratix 10 devices)

Number of Lanes	10.3125 G	12.5 G	17.4 G
1 – 6	x1 – ATX/FPLL/CMU	x1/x6/xN – ATX/FPLL	x1/x6/xN – ATX
	x6/xN – ATX/FPLL		
7 – 24	x1 – ATX/FPLL/CMU if using multiple PLLs	x1/x6/xN – ATX/FPLL if using 1 or multiple PLLs	x1/x6/xN – ATX if using multiple PLLs (use multiple PLLs as xN with 1.12 V can only support up to 16 G)
	x6/xN – ATX/FPLL if using 1 or multiple PLLs		

For ATX PLL VCO frequencies between 7.2 GHz and 11.4 GHz, when two ATX PLLs operate at the same VCO frequency (within 100 MHz), you must be placed seven ATX PLLs apart (skip 6). For ATX PLL VCO frequencies between 11.4 GHz and 14.4 GHz, when two ATX PLLs operate at the same VCO frequency (within 100 MHz), you must be placed four ATX PLLs apart (skip 3). If these spacing rules are violated, the Quartus Prime issues a critical warning. The maximum channel span of a xN clock network is two transceiver banks above and two transceiver banks below the bank that contains the driving PLL and the master CGB. You can use a maximum of 30 channels in a single-bonded or non-bonded xN group. The maximum data rate supported by the xN clock network while driving channels in either the bonded or non-bonded mode depends on the voltage used to drive the transceiver banks and the transceiver speed grade.

Note: For Quartus Prime Pro Edition, configure the ATX PLL and fPLL parameter VCCR_GXB and VCCT_GXB supply voltage for the Transceiver to 1.1V and re-generate the IP cores when using data rate more then 15 Gbps.

5.2 Clocking Structure For Arria 10 Devices

Standard Clocking Mode

Figure below shows the source and sink variant clocking structure for standard clocking mode in Arria 10 devices.

Figure 26. Clocking Structure for Arria 10 Devices

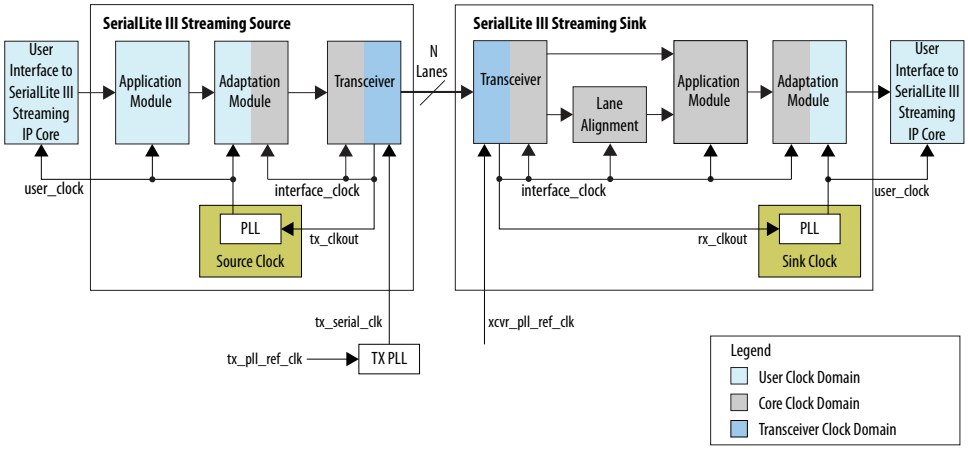


Table 31. Arria 10 Clocks in Standard Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This is an output clock provided by the IP core to the user. This clock should be used to clock the TX user application that drives the TX user interface.
tx_serial_clk	This clock should toggle at one-half the data rate of the transceiver lane. When you enter the `user_clock` frequency in the IP parameter editor, the per lane data rate is calculated. Use that value and divided it by two to determine the `tx_serial_clk`. You are required to instantiate the TX PLL, as shown in the figure above. An example of the TX PLL (ATX PLL) is generated with the IP core and is configured with the required reference clock and `tx_serial_clk`.
tx_clkout	This clock is not exposed to the user. It is used as a reference clock for the internal PLL. The frequency of `tx_clkout` is the data rate divided by 64.
interface_clock	This clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP parameter editor and is the transceiver data rate divided by transceiver PCS-PMA width (Arria 10: 64 bits, Stratix V and Arria V GZ: 40 bits). The internal PLL is configured to generate the required frequency.
Sink	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. The frequency of this clock should match the frequency of the `user_clock` in the Source variant. This is an output clock provided by the IP core to the user. This clock should be used to clock the RX user application that drives the RX user interface.

continued...

Clock Name	Description
xcvr_pll_ref_clk	This reference clock is used by the Clock Data Recovery (CDR) unit in the transceiver. It serves as a reference for the CDR to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
rx_clkout	This clock is not exposed to the user. It is used as a reference clock for the internal PLL in the Sink. The frequency of `rx_clkout` is the data rate divided by 64.
interface_clock	This clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP parameter editor and is the transceiver data rate divided by transceiver PCS-PMA width (Arria 10: 64 bits, Stratix V and Arria V GZ: 40 bits). The internal PLL is configured to generate the required frequency.

Example:

An application requires the SerialLite III Streaming IP core to sustain a frequency of 300 Gbps at the user interface.

user_clock (frequency) × number_of_lanes × 64 bits/lane = 300 Gbps

The data rate for Arria 10 GX transceivers is limited to 17.4 Gbps. Therefore, 300 Gbps / 17.4 Gbps = 18 (rounding up)

Choosing 18 lanes gives:

user_clock (frequency) = 300 / (18 × 64) = 260.42 MHz

A value of 260.40 MHz is out of the supported range for the `user_clock` frequency. Therefore, you need to add one more lane.

user_clock (frequency) = 300 / (19 × 64) = 246.71 MHz

Choosing 246.71 MHz as the user_clock, the IP core provides the following values:

Transceiver data rate: 17.368 Gbps

tx_clkout: 17.368 / 64 = 271.375 MHz

interface_clk: 17.368 / 64 = 271.375 MHz

tx_serial_clock: 17.368 / 2 = 8684 MHz

Advanced Clocking Mode

Figure below shows the source and sink variant clocking structure for advanced clocking mode in Arria 10 devices.

Figure 27. Clocking Structure for Arria 10 Devices

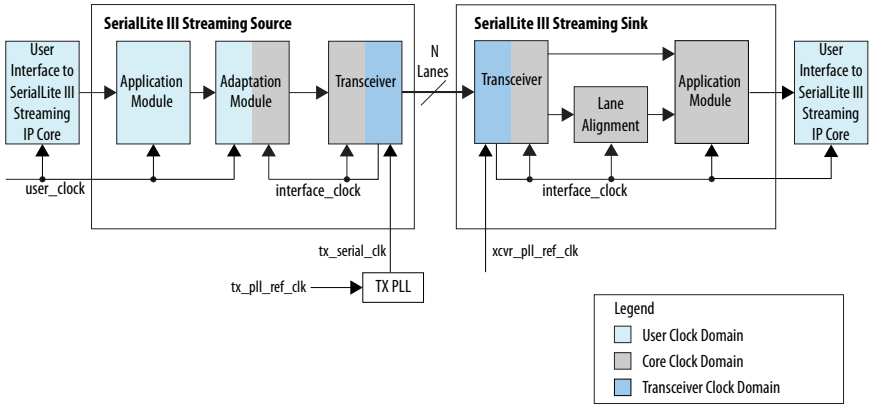


Table 32. Arria 10 Clocks in Advanced Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This clock is an input to the IP core and you should toggle this at the specified frequency.
tx_serial_clk	This clock should toggle at one-half the data rate of the transceiver lane. When you enter the `user_clock` frequency in the IP parameter editor, the per lane data rate is calculated. Use that value and divided it by two to determine the `tx_serial_clk`. You are required to instantiate the TX PLL. An example of the TX PLL (ATX PLL) is generated with the IP core and is configured with the required reference clock and `tx_serial_clk` frequencies.
interface_clock	This is an internal clock and it is not exposed to the user. The frequency of this clock is derived from the transceiver data rate. The frequency is lane data rate divided by 64.
Sink	
xcvr_pll_ref_clk	This reference clock is used by the CDR unit in the transceiver. It serves as a reference for the CDR to be able to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
interface_clock	This clock is derived from the transceiver data. It is lane data rate divided by 64. It is an output of the IP core and should be used to clock the RX user application,

Example:

An application requires the SerialLite III Streaming IP core to sustain a frequency of 240 Gbps at the user interface.

user_clock (frequency) × number_of_lanes × 64 bits/lane = 240 Gbps

The data rate for Arria 10 GX transceivers is limited to 17.4 Gbps. Therefore, 240 Gbps / 17.4 Gbps = 14 (rounding up)

Choosing 14 lanes gives:

user_clock (frequency) = 240 / (14 × 64) = 267.86 MHz

A value of 267.86 MHz is out of the supported range for the `user_clock` frequency. Therefore, you need to add one more lane.

user_clock (frequency) = 240 / (15 × 64) = 250 MHz

Choosing 250 MHz as the `user_clock`, the IP core provides the following values:

Transceiver data rate: 16.78 Gbps

interface_clk: 16.78 / 64 = 262.18 MHz

tx_serial_clock: 16.78 Gbps / 2 = 8390 MHz

5.2.1 Choosing TX PLL Type for Arria 10 Devices

These are the guidelines for choosing the appropriate type of TX PLL for Arria 10 devices.

Table 33. PLL Data Rate Performance

Number of Lanes	10.3125 G	12.5 G	17.4 G
1 – 6	x1 – ATX/FPLL/CMU	x1/x6/xN – ATX/FPLL	x1/x6/xN – ATX
	x6/xN – ATX/FPLL		
7 – 24	x1 – ATX/FPLL/CMU if using multiple PLLs	x1/x6/xN – ATX/FPLL if using 1 or multiple PLLs	x1/x6/xN – ATX if using multiple PLLs (use multiple PLLs as xN with 1.12 V can only support up to 16 G)
	x6/xN – ATX/FPLL if using 1 or multiple PLLs		

For ATX PLL VCO frequencies between 7.2 GHz and 11.4 GHz, when two ATX PLLs operate at the same VCO frequency (within 100 MHz), you must be placed seven ATX PLLs apart (skip 6). For ATX PLL VCO frequencies between 11.4 GHz and 14.4 GHz, when two ATX PLLs operate at the same VCO frequency (within 100 MHz), you must be placed four ATX PLLs apart (skip 3). If these spacing rules are violated, the Quartus Prime issues a critical warning. The maximum channel span of a xN clock network is two transceiver banks above and two transceiver banks below the bank that contains the driving PLL and the master CGB. You can use a maximum of 30 channels in a single-bonded or non-bonded xN group. The maximum data rate supported by the xN clock network while driving channels in either the bonded or non-bonded mode depends on the voltage used to drive the transceiver banks and the transceiver speed grade.

Related Links

Arria 10 Transceiver PHY User Guide
For more information about the PLL types in Arria 10 Transceiver PHY.

5.3 Clocking Structure For Stratix V or Arria V Devices

Standard Clocking Mode

Figure below shows the source and sink variant clocking structure for standard clocking mode for Stratix V or Arria V devices.

Figure 28. Clocking Structure for Stratix V or Arria V Devices

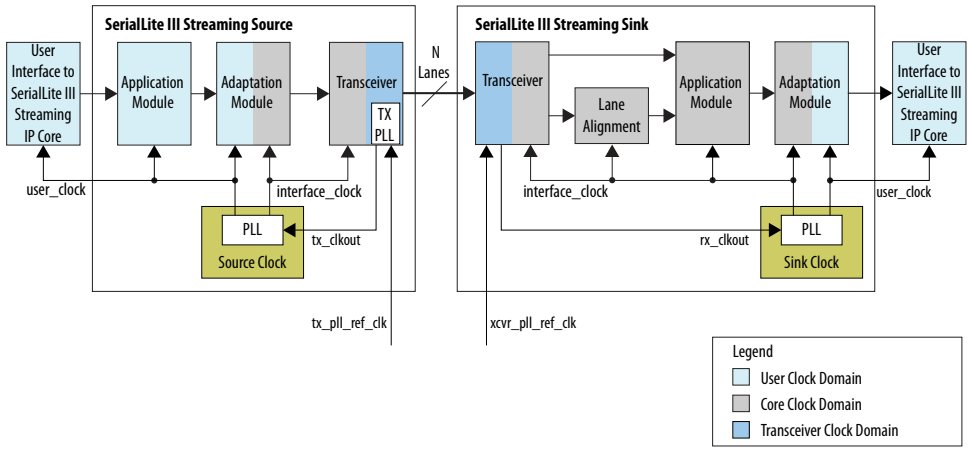


Table 34. Stratix V or Arria V Clocks in Standard Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This clock is provided by the IP core and used to clock the TX user application.
tx_pll_ref_clk	This is the reference clock for the transceiver TX PLL. The frequency is selected from the available values in the IP parameter editor and must match that value.
tx_clkout	This clock is not exposed to the user. It is used as a reference clock for the internal PLL. The frequency of this clcok is data rate divided by 40.
interface_clock	The interface_clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP core and is the transceiver data rate divided by transceiver PCS-PMA width (Arria 10: 64 bits, Stratix V and Arria V GZ: 40 bits). The internal PLL is configured to generate the required frequency.
Sink	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. This is an output clock provided by the IP core to the user and used to clock the RX user application.

continued...

Clock Name	Description
xcvr_pll_ref_clk	This reference clock is used by the CDR unit in the transceiver. It serves as a reference for the CDR to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
rx_clkout	This clock is not exposed to the user. It is used as a reference clock for the internal PLL in the Sink. The frequency of this clock is data rate divided by 40.
interface_clock	This clock is an internal clock and is not exposed to the user. The frequency of this clock is calculated by the IP core and is the transceiver data rate divided by transceiver PCS-PMA width (Arria 10: 64 bits, Stratix V and Arria V GZ: 40 bits). The internal PLL is configured to generate the required frequency.

Advanced Clocking Mode

Figure below shows the source and sink variant clocking structure for advamced clocking mode for Stratix V or Arria V devices.

Figure 29. Clocking Structure for Stratix V or Arria V Devices

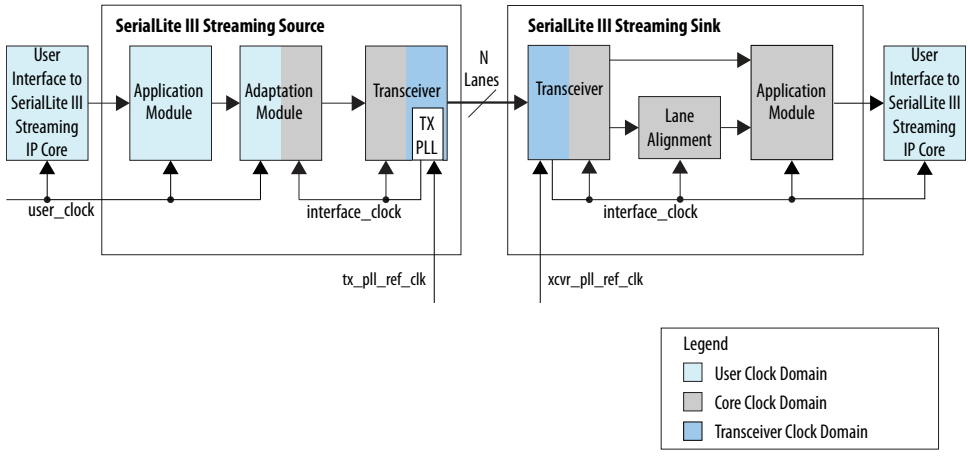


Table 35. Stratix V or Arria V Clocks in Advanced Clocking Mode

Clock Name	Description
Source	
user_clock	User-defined. This clock is determined by the required throughput of the user application. For example, if the user interface is 384-bits wide (6 lanes × 64 bit/lane) and the required throughput is 120 Gbps, the `user_clock` frequency is 312.5 MHz. You are responsible for supplying this clock into the IP core.
tx_pll_ref_clk	This is the reference clock for the transceiver TX PLL. The frequency is selected from the available values in the IP GUI and must match that value.
interface_clock	This is an internal clock and it is not exposed to the user. The frequency of this clock is derived from the transceiver data rate. It is lane data rate divided by 40.

continued...

Clock Name	Description
Sink	
xcvr_pll_ref_clk	This reference clock is used by the CDR unit in the transceiver. It serves as a reference for the CDR to recover the clock from the serial line. The frequency of this clock must match the frequency you select in the IP parameter editor. It should also match the frequency of the `tx_pll_ref_clk` reference clock for the TX PLL at the Source variant.
interface_clock	The frequency of this clock is derived from the transceiver data. It is lane data rate divided by 40. This clock is an output of the IP core. Use this clock to clock the RX user application.

5.3.1 Choosing TX PLL Type for Stratix V or Arria V Devices

The SerialLite III Streaming IP core allows a selection of PLL type for use inside the transmit and receive PMA blocks. The IP parameter editor in Quartus Prime allows you to select either a CMU PLL or an ATX PLL. The CMU PLL is more suitable for lower lane data rates, while the ATX PLL is better for higher lane data rates. The supported data rates for the CMU PLL and ATX PLL are provided in Tables 1 (Stratix V) and 2 (Arria V GZ). These tables list the maximum lane data rates per transceiver speed grade. For example, if your design requires a 14.1 Gbps lane rate, you need to use an ATX PLL and select a Transceiver Speed Grade 1 device.

Table 36. Stratix V CMU and ATX PLL Supported Data Rates

Symbol/ Description	Conditions	Transceiver Speed Grade 1 (Mbps)			Transceiver Speed Grade 2 (Mbps)			Transceiver Speed Grade 3 (Mbps)		
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
CMU PLL Supported Data Range	_	600	_	12500	600	_	12500	600	_	8500
ATX PLL Supported Data Range	VCO Post-divider L=2	8000	_	14100	8000	_	12500	8000	_	8500
	L=4	4000	_	7050	4000	_	6600	4000		6600
	L=8	2000	_	3525	2000	_	3300	2000	_	3300

Table 37. Arria V GZ CMU and ATX PLL Supported Data Rates

Symbol/ Description	Conditions	Transceiver Speed Grade 2 (Mbps)			Transceiver Speed Grade 3 (Mbps)		
		Min	Typ	Max	Min	Typ	Max
CMU PLL Supported Data Range	_	600	_	12500	600	_	10312.5
ATX PLL Supported Data Range	VCO Post-divider L=2	8000	_	12500	8000	_	10312.5
	L=4	4000	_	6600	4000		6600
	L=8	2000	_	3300	2000	_	3300

For more information about the Stratix V and Arria V GZ devices, refer to their respective device datasheets.

Related Links

- Altera Transceiver PHY User Guide
- Stratix V Device Datasheet
- Arria V GZ Device Datasheet

5.4 Clocking Implementation Scenarios

This section describes a few clocking implementation scenarios.

Synchronous Systems

In this scenario, both the Source User Clock and Sink FIFO read clock frequencies are the same. As shown in the figure below, the FIFO read clock is derived from the same crystal oscillator as the Source User Clock. If the Source User Clock requires a PLL, the Sink User Clock will also require a PLL with the same configuration.

Figure 30. Same Source and Sink User Clock Frequencies from Same Crystal Oscillator for Stratix 10 Devices

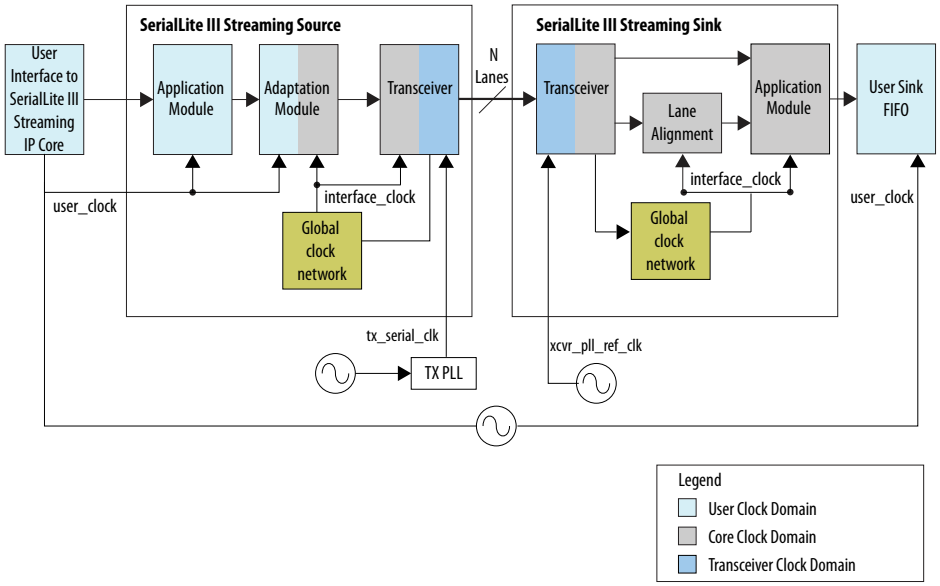
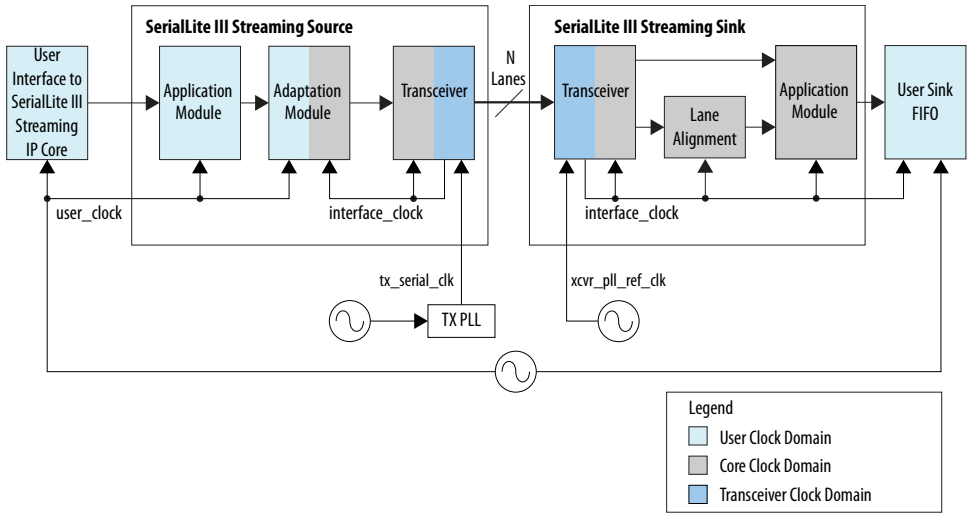


Figure 31. Same Source and Sink User Clock Frequencies from Same Crystal Oscillator for Arria 10, Stratix V and Arria V GZ Devices



Asynchronous Systems

In an asynchronous system, the sink FIFO read clock is derived from a different crystal oscillator, but has the same frequency as the Source User Clock. In this scenario, a PPM difference exists between the Source User Clock and the FIFO read clock. The Source input data rate needs to be reduced to avoid overflowing the Sink FIFO buffer due to the PPM differences. One recommended way is to insert empty cycles in the Source input data stream at Source User Data Interface to reduce the data rate. The Source Application and Adaptation modules absorb these empty data cycles, convert them to idle cells, and insert them into link data stream. These cells are automatically removed at the sink interface and converted back into empty cycles on the sink user interface.

Note: You have to take into consideration the PPM difference and insert enough empty cycles to offset the PPM difference for the worst case scenario.

Figure 32. Same Source and Sink User Frequencies with Different Crystal Oscillators for Stratix 10 Devices

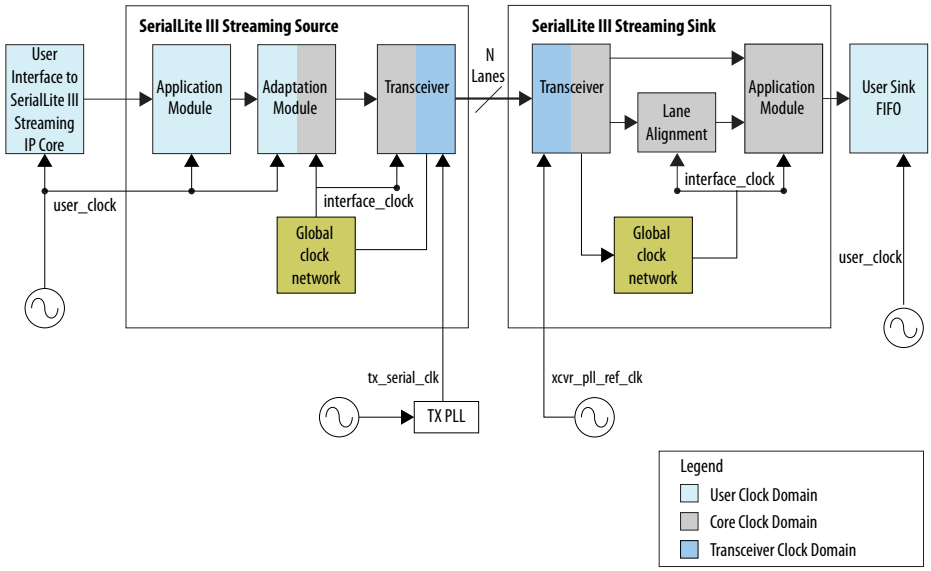
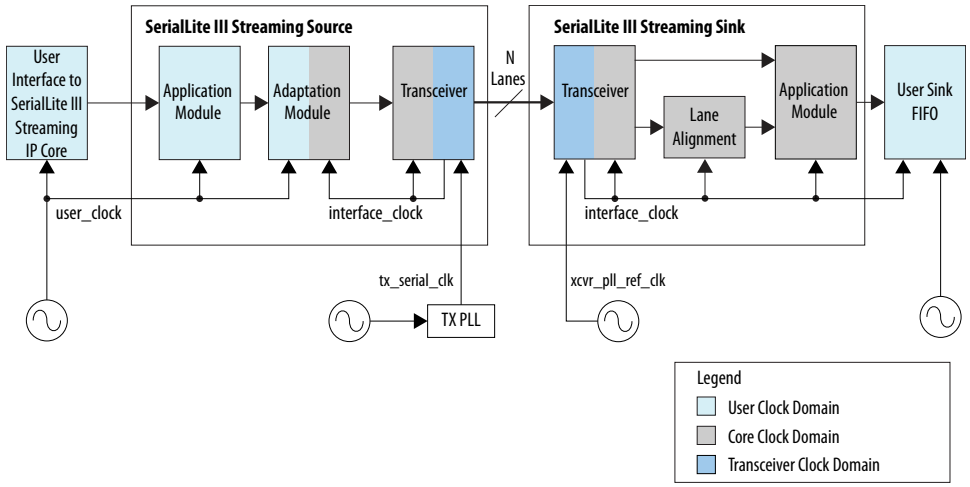


Figure 33. Same Source and Sink User Frequencies, with Different Crystal Oscillators for Arria 10, Stratix V and Arria V GZ Devices

Figure illustrates how two crystal oscillators are used to provide the Source User Clock and the Sink FIFO read clock.



Related Links

SCFIFO and the DCFIFO IP Cores User Guide

For more information on how to add a sink FIFO buffer.

5.5 Standard Clocking Mode vs Advanced Clocking Mode

Table below lists the comparison between two clocking modes that the SerialLite III Streaming IP core supports.

Table 38. Clocking Mode Comparisons

Attribute	Standard Clocking Mode	Advanced Clocking Mode
User clock sourcing	For Arria 10, Stratix V, and Arria V GZ devices, this clock is generated by the SerialLite III Streaming IP core, provided as an output to the user. For Stratix 10 devices, this clock is provided by the user to the IP core.	Generated by the user, provided as an input to the SerialLite III Streaming IP core.
fPLL usage (for Arria 10, Stratix V, and Arria V GZ devices) *Note:* The fPLLs are located in the FPGA core fabric	Simplex: 1 per Source core and 1 per Sink core per SerialLite III Streaming IP core instance. Duplex: 2 per SerialLite III Streaming IP core instance. (Lane number does not factor in the use of fPLLs. Only the SerialLite III Streaming IP core instances are factored in.)	Not used in either simplex or duplex cores.
Transmission overhead	1.1	Interlaken overheads
Resource utilization	Higher	Lower

These considerations apply when selecting a clocking mode:

- If the Source User Interface Clock must come from a crystal oscillator other than the transceiver reference clock crystal, you are recommended to use Advanced Clocking Mode.

6 SerialLite III Streaming IP Core Debugging Guidelines

This section includes guidelines to assist you in debugging the IP core link issues.

6.1 Creating a Signal Tap II Debug File to Match Your Design Hierarchy

For Arria 10 devices, the Quartus Prime Standard Edition software generates two files, `build_stp.tcl` and `<ip_core_name>.xml`. You can use these files to generate a Signal Tap II file with probe points matching your design hierarchy.

The Quartus Prime software stores these files in the `<IP core directory>/synth/debug/stp/` directory.

Synthesize your design using the Quartus Prime software.

1. To open the Tcl console, click **View ➤ Utility Windows ➤ Tcl Console**.
2. Type the following command in the Tcl console:
   `source <IP core directory>/synth/debug/stp/build_stp.tcl`
3. To generate the STP file, type the following command:
   `main -stp_file <output stp file name>.stp -xml_file <input xml_file name>.xml -mode build`
4. To add this Signal Tap II file (**.stp**) to your project, select **Project ➤ Add/Remove Files in Project**. Then, compile your design.
5. To program the FPGA, click **Tools ➤ Programmer**.
6. To start the Signal Tap II Logic Analyzer, click **Quartus Prime ➤ Tools ➤ Signal Tap II Logic Analyzer**.

   The software generation script may not assign the Signal Tap II acquisition clock in `<output stp file name>.stp`. Consequently, the Quartus Prime software automatically creates a clock pin called `auto_stp_external_clock`. You may need to manually substitute the appropriate clock signal as the Signal Tap II sampling clock for each STP instance.
7. Recompile your design.
8. To observe the state of your IP core, click **Run Analysis**.

   You may see signals or Signal Tap II instances that are red, indicating they are not available in your design. In most cases, you can safely ignore these signals and instances. They are present because software generates wider buses and some instances that your design does not include.

6.2 SerialLite III Streaming Link Debugging

The following section describes the link-up sequence that you can use when debugging the SerialLite III Streaming IP core. The internal signals in the charts and tables can be observed by the Signal Tap II.

6.2.1 Source Core Link Debugging (Arria 10, Stratix 10, Arria V GZ, and Stratix V)

Figure 34. Source Core Link Debugging Flow Chart

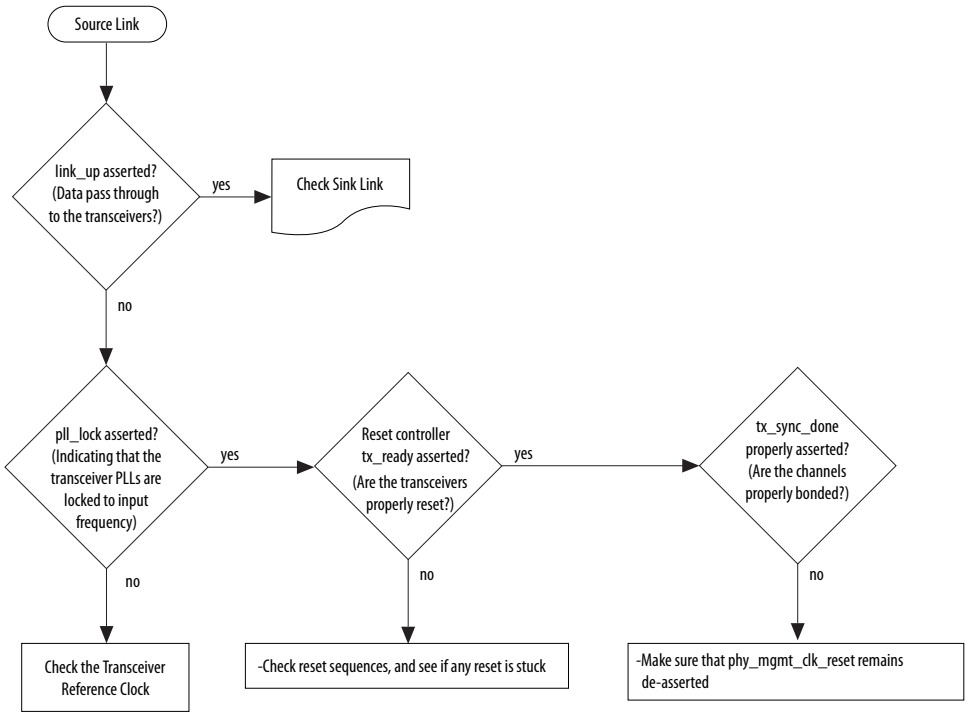


Table 39. Source Link Debugging Signals (Arria 10, Stratix 10, Arria V GZ, and Stratix V devices)

Signal Name	Location (Arria 10, Arria V GZ, and Stratix V)	Location (Stratix 10)	Description
link_up_tx	Top level port	Top level port	The core asserts this signal to indicate that initialization sequence is complete and the core is ready to transmit the data.
tx_pll_locked	Native PHY wrapper port	Top level port	This active high signal indicates that the transceivers are locked to the reference clock.

continued...

Signal Name	Location (Arria 10, Arria V GZ, and Stratix V)	Location (Stratix 10)	Description
tx_pcs_ready	Native PHY wrapper port	(Encrypted) Soft PHY port	This active high signal indicates that the reset sequence for the source PCS is complete and is ready to accept data.
tx_sync_done	Native PHY wrapper port	PHY top port	This active high signal indicates that all the lanes are bonded by the Native PHY or Interlaken PHY IP core. This signal should be properly asserted for normal operation. A rapidly toggling signal indicates that the source FIFO is having either too much or too little data, or the core reset is having issues.
tx_cal_busy	**/source/ Interlaken_phy_ip_tx/ sv_ilk_inst** (for Stratix V devices only)	_	Source transceiver calibration status. This active high signal can be used for debugging if the reconfiguration controller is actively calibrating during the initialization sequence.

6.2.2 Sink Core Link Debugging (Arria 10, Stratix 10, Arria V GZ, and Stratix V)

Figure 35. Sink Core Link Debugging Flow Chart

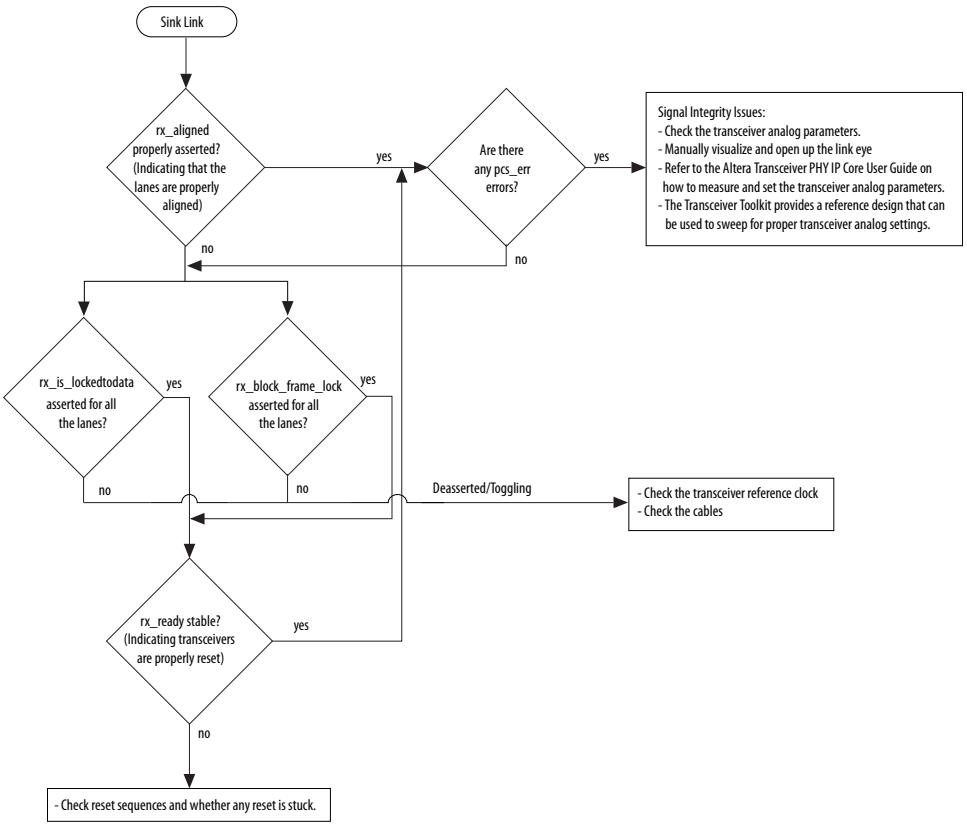


Table 40. Sink Link Debugging Signals (Arria 10, Stratix 10, Arria V GZ, and Stratix V devices)

Signal Name	Location (Arria 10, Arria V GZ, and Stratix V)	Location (Stratix 10)	Description
rx_aligned	Top level port	Top level port	This active high signal indicates that the lanes are properly aligned. This signal should remain asserted for proper operation.
rx_pcs_ready	Native PHY wrapper port	(Encrypted) Soft PHY port	An asserted value for this active high signal indicates that the reset sequence for the sink PCS is complete.
rx_crc32err	Native PHY wrapper port	PHY top internal	This active high signal indicates CRC-32 error from the CRC checker.
rx_pcs_err	_	PHY top port	This active high signal indicates if there is any Sync Header, Meta-frame Length, or CRC32 error. This signal can be used to debug whether there is any data integrity issue on a given meta-frame.
rx_frame_lock [lanes-1:0]	Native PHY wrapper port	PHY top internal	This active high signal indicates that four Interlaken synchronization words are found for a given lane.
rx_block_frame_lock	Native PHY wrapper port	PHY top port	This active high signal indicates whether a link has established both block lock and frame lock link alignment state.
rx_is_lockedtodata [lanes-1:0]	Native PHY wrapper internal	PHY top internal	This active high signal indicates that the transceiver channel PLL has locked itself to the incoming data.
rx_cal_busy	Native PHY wrapper internal	PHY top internal	Sink transceiver calibration status. This active high signal can be used for debugging if the reconfiguration controller is actively calibrating during the initialization sequence.

6.3 Error Handling

Table 41. Error Conditions and Core Behavior

This table lists the error conditions that the core detect and their behavior in response to each condition.

Condition		Error Indication	Core Behavior
Source Core	Rate adaptation FIFO buffer overflow in source interface	The source core asserts the error flag for one clock cycle.	There is an overflow on the rate adaptation FIFO buffer in the source interface. The core behavior depends on the operation mode: • Continuous mode—error is flagged once an overflow is detected. • Burst mode—error is flagged only when an overflow occurs during burst data transfer across the user interface.
Sink Core	Diagnostic code word CRC-32 error	The sink core asserts error[N-1:0][4] flag for one clock cycle.	The sink interface detects a metaframe CRC-32 error on one of the lanes. These errors are reported on a per-lane basis for diagnostic purposes.

continued...

4 N is the number of lanes.

Condition		Error Indication	Core Behavior
	Lane alignment failure during normal operation	The sink core asserts error[N][4] flag for one clock cycle.	The sink interface detects a loss of lane alignment during normal operation.
Sink Core **Duplex**	Error status on `error_rx` signal	One or more errors have occured in a given meta-frame, as determined by Native PHY PCS logic (in Interlaken mode).	The error status such as CRC error can only be checked at the end of meta-frame, and there may be multiple user packets within a given meta-frame. The implication is that the `error_rx` signal could be much delayed (with respect to the user packets received earlier) at the receiving link. *Note:* If data integrity is critical, additional error checksum may be included in the user logic as part of data payload so that the downstream user logic can determine the data integrity at packet level.

A SerialLite III Streaming Configuration and Status Registers

A.1 Register Map

Table 42. Register Map for SerialLite III Streaming MAC

Word Address	Register Name
Source Registers	
0x0090	TX Error Status
0x0091	TX Error Interrupt Enable
Sink Registers	
0x00c2	RX MAC Control
0x00D0	RX Error Status
0x00D1	RX Error Interrupt Enable

A.2 Configuration and Status Registers

Table 43. Source Configuration and Status Registers for MAC

Use the following defnition for register access shown in the table:

- W1C = Write 1 to clear.
- RW = Read Write.

Word Address	Bits	Register Name	Description	Access	Default Value
TX Error Status Register					
0x0090	6	tx_burst_gap_err	TX burst gap error. This bit is set when the gap between two consecutive bursts (or packets) on TX user data interface is less than the required BURST_GAP (a synthesis option). When this error happens, make sure the user interface behavior and BURST_GAP are matching.	W1C	
	5	ecc_err_fatal	TX ECC Error Fatal consolidated status (of all lanes). This bit is set when double bit error is detected and uncorrected.	W1C	0x0
	4	ecc_err_corrected	TX ECC Error Corrected consolidated status (of all lanes). This bit is set when single bit error is detected and corrected.	W1C	0x0

continued...

Word Address	Bits	Register Name	Description	Access	Default Value
	3	adapt_fifo_overflow	TX MAC Adaptation FIFO overflow consolidated status (of all lanes). In normal condition with all clocks running correctly, this bit is set when the user data rate is faster than expected.When this happens, stop further data transfer and check if the clocks are set correctly.	W1C	0x0
	2	tx_sync_done_lost	TX Lost of Lane Alignment consolidated status (of all lanes).	W1C	0x0
	1	phy_fifo_underflow	TX PHY Phase Compensation FIFO underflow consolidated status (of all lanes). This bit is set when the IP core has major error and requires a full IP core reset.	W1C	0x0
	0	phy_fifo_overflow	TX PHY Phase Compensation FIFO overflow consolidated status (of all lanes). This bit is set when the IP core has major error and requires a full IP core reset.	W1C	0x0
TX Error Interrupt Enable Register					
0x0091	6	tx_burst_gap_err_en	Set this bit to 1 to enable the Burst Gap Error Interrupt.	RW	0x1
	5	ecc_err_fatal_en	Set this bit to 1 to enable the ECC Uncorrected Error Interrupt.	RW	0x0
	4	ecc_err_corrected_en	Set this bit to 1 to enable the ECC Corrected Error Interrupt.	RW	0x0
	3	adapt_fifo_overflow_en	Set this bit to 1 to enable the Adaptation FIFO Overflow Interrupt.	RW	0x1
	2	tx_sync_donelost_en	Set this bit to 1 to enable the Loss of Lane Alignment Interrupt.	RW	0x0
	1	phy_fifo_underflow_en	Set this bit to 1 to enable the PHY FIFO Empty Interrupt.	RW	0x0
	0	phy_fifo_overflow_en	Set this bit to 1 to enable the PHY FIFO Error Interrupt.	RW	0x0

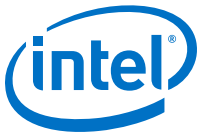
Table 44. Sink Configuration and Status Registers for MAC

Use the following defnition for register access shown in the table:

- W1C = Write 1 to clear.
- RW = Read Write.

Word Address	Bit	Register Name	Description	Access	Default Value
RX MAC Control Register					
0x00C2	0	link_reinit	Set this bit to initiate link re-initialization. When asserted, lane alignment state machine will go to IDLE state and restart the lane alignment process.	RW	0x0
RX Error Status Register					
0x00D0	11	—	Reserved	—	—

continued...



Word Address	Bit	Register Name	Description	Access	Default Value
	10	rx_deskew_fatal	RX Lane Deskew Fatal status (of all lanes). This bit is set, when the lane skews across all lanes have exceeded the hardware de-skew capability. This should not happen under normal conditions. When this bit is set, identify the routing of the lanes (e.g. RX PHY-> board routing -> TX PHY) where large skews are introduced.	W1C	0x0
	9	ecc_err_fatal	RX ECC Error Fatal consolidated status (of all lanes). This bit is set when double bit error detected and uncorrected.	W1C	0x0
	8	ecc_err_corrected	RX ECC Error Corrected consolidated status (of all lanes). This bit is set when single bit error is detected and corrected.	W1C	0x0
	7	—	Reserved	—	—
	6	rx_alignment_lostlock	RX Lane Alignment Lost consolidated status (of all lanes). This bit is set when lost of alignment is detected by the MAC. This error happens when the SYNC control words across all lanes do not appear in the same clock cycle and data corruption could have happened since one or more lanes would be out of alignment with others. The RX MAC will re-establish the alignment to recover from this error. However, if the condition still persists, a full IP reset is required.	W1C	0x0
	5	rx_align_retry_fail	RX Lane Alignment Retry Fail consolidated status (of all lanes). This bit is set to indicate the number of lane alignment retries has exceeded the expected value.	W1C	0x0
	4	rx_pcs_err	RX PCS Error consolidated status (of all lanes). This bit is set when a synchronization error, metaframe error or crc32 error happens.	W1C	0x0
0x00D0	3	rx_crc32err	RX CRC error consolidated status (of all lanes) for data integrity monitoring purpose.	W1C	0x0
	2	rx_frame_lostlock	RX Loss of Frame Lock consolidated status (of all lanes). This bit is set when there is a loss of frame lock in the receive frame due to signal integrity errors on the serial data stream or when the remote partner is being reset. When this happen, check and remove the cause(s) of loss of frame lock to enable the IP core to self-recover to normal state.	W1C	0x0
	1	rx_block_lostlock	RX Loss of Block Lock consolidated status (of all lanes). This bit is set when there is a loss of block lock in the receive frame due to signal integrity errors on the serial data stream or when the remote partner is being reset.	W1C	0x0

continued...

Word Address	Bit	Register Name	Description	Access	Default Value
			When this happen, check and remove the cause(s) of loss of block lock to enable the IP core to self-recover to normal state.		
	0	phy_fifo_overflow	RX PHY Phase Compensation FIFO overflow consolidated status (of all lanes). This bit is set when the IP core has major error and requires a full IP core reset.	W1C	0x0
RX Error Interrupt Enable Register					
0x00D1	11	—	Reserved	—	—
	10	rx_deskew_fatal_int_en	Set this bit to 1 to enable the RX Lane Deskew Fatal Interrupt.	RW	0x0
	9	ecc_err_fatal_int_en	Set this bit to 1 to enable the RX ECC Error Detected and Uncorrected Interrupt.	RW	0x0
	8	ecc_err_corrected_int_en	Set this bit to 1 to enable the RX ECC Error Detected and Corrected Interrupt.	RW	0x0
	7	—	Reserved	—	—
	6	rx_alignment_lostlock_int_en	Set this bit to 1 to enable the RX Loss of Lane Alignment Interrupt.	RW	0x0
	5	rx_align_retry_fail_int_en	Set this bit to 1 to enable the RX Lane Alignment Retry Fail Interrupt.	RW	0x0
	4	rx_pcs_err_int_en	Set this bit to 1 to enable the RX PCS Error Interrupt.	RW	0x0
	3	rx_crc32err_int_en	Set this bit to 1 to enable the RX CRC Error Interrupt.	RW	0x0
	2	rx_frame_lostlock_int_en	Set this bit to 1 to enable the RX Loss of Frame Lock Interrupt.	RW	0x0
	1	rx_block_lostlock_int_en	Set this bit to 1 to enable the RX Loss of Block Lock Interrupt.	RW	0x0
	0	phy_fifo_overflow_int_en	Set this bit to 1 to enable the RX PHY FIFO Overflow Interrupt.	RW	0x0

Table 45. Interlaken PHY Registers

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x022	[<p>-1:0]	RO	pma_tx_pll_is_locked	If <*p*> is the PLL number, Bit[<*p*>] indicates that the TX CMU PLL (<*p*>) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system.
Reset Control Registers-Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel <n> can be reset when bit<*n*> = 1. Channel <*n*> cannot be reset when bit<*n*> = 0.

continued...

Word Addr	Bits	R/W	Register Name	Description
				The Interlaken PHY IP requires the use of the embedded reset controller to initiate the correct the reset sequence. A hard reset to `phy_mgmt_clk_reset` and `mgmt_rst_reset` is required for Interlaken PHY IP. Intel does not recommend use of a soft reset or the use of these reset register bits for Interlaken PHY IP.
0x042	[1:0]	WO	`reset_control` (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the `reset_ch_bitmask`. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the `reset_ch_bitmask`.
		RO	`reset_status`(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
Reset Controls -Manual Mode				
0x044	-	RW	`reset_fine_control`	You can use the `reset_fine_control` register to create your own reset sequence. The reset control module, illustrated in Transceiver PHY Top-Level Modules, performs a standard reset sequence at power on and whenever the `phy_mgmt_clk_reset` is asserted. Bits [31:4, 0] are reserved. The Interlaken PHY IP requires the use of the embedded reset controller to initiate the correct the reset sequence. A hard reset to `phy_mgmt_clk_reset` and `mgmt_rst_reset` is required for Interlaken PHY IP. Intel does not recommend use of a soft reset or the use of these reset register bits for Interlaken PHY IP.
	[3]	RW	`reset_rx_digital`	Writing a 1 causes the RX digital reset signal to be asserted, resetting the RX digital channels enabled in `reset_ch_bitmask`. You must write a 0 to clear the reset condition.
	[2]	RW	`reset_rx_analog`	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX analog logic of all channels enabled in `reset_ch_bitmask`. You must write a 0 to clear the reset condition.
	[1]	RW	`reset_tx_digital`	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in `reset_ch_bitmask`. You must write a 0 to clear the reset condition.
PMA Control and Status Registers				
0x061	[31:0]	RW	`phy_serial_loopback`	Writing a 1 to channel <*n*> puts channel <*n*> in tx to rx serial loopback mode. For information about pre- or post-CDR rx to tx serial loopback modes, refer to Loopback Modes.
0x064	[31:0]	RW	`pma_rx_set_locktodata`	When set, programs the RX CDR PLL to lock to the incoming data. Bit <*n*> corresponds to channel <*n*>. By default, the Interlaken PHY IP configures the CDR PLL in Auto lock Mode. This bit is part of the CDR PLL Manual Lock Mode which is not the recommended usage.

continued...

Word Addr	Bits	R/W	Register Name	Description
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <*n*> corresponds to channel <*n*>. By default, the Interlaken PHY IP configures the CDR PLL in Auto lock Mode. This bit is part of the CDR PLL Manual Lock Mode which is not the recommended usage.
0x066	[31:0]	RO	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <*n*> corresponds to channel <*n*>.
00x067	[31:0]	RO	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit <*n*> corresponds to channel <*n*>.
0x080	[31:0]	WO	indirect_addr	Provides for indirect addressing of all PCS control and status registers. Use this register to specify the logical channel address of the PCS channel you want to access.
Device Registers				
	[27]	RO	rx_crc32_err	Asserted by the CRC32 checker to indicate a CRC error in the corresponding RX lane. **From block**: CRC32 checker.
0x081	[25]	RO	rx_sync_lock	Asserted by the frame synchronizer to indicate that 4 frame synchronization words have been received so that the RX lane is synchronized. **From block**: Frame synchronizer.
	[24]	RO	rx_word_lock	Asserted when the first alignment pattern is found. The RX FIFO generates this synchronous signal. **From block**: The RX FIFO generates this synchronous signal.

For Native PHY IP core configuration and status registers, refer to *Arria 10 Tranceiver Register Map*, *Logical View of the Stratix 10 L-Tile Transceiver Registers*, and *Physical View of the Stratix 10 L-Tile Transceiver Registers* in related links.

Related Links

- Arria 10 Transceiver Register Map
  Information on configuration and status registers in Arria 10 transceiver.
- Logical View of the Stratix 10 L-Tile Transceiver Registers
- Physical View of the Stratix 10 L-Tile Transceiver Registers

B SerialLite III Streaming IP Core User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
16.1	SerialLite III Streaming IP Core User Guide
16.0	SerialLite III Streaming IP Core User Guide
15.1	SerialLite III Streaming IP Core User Guide
15.0	SerialLite III Streaming IP Core User Guide
14.1	SerialLite III Streaming IP Core User Guide

C SerialLite III Streaming IP Core Document Revision History

Date	Version	Changes
May 2017	2017.05.08	• Clarified the device family support for Stratix 10 devices. • Clarified the exact replica of output data support for pure streaming operation in the following locations: — Continuous Mode sub-topic — IP Core Architecture topic — Comparing Standard and Advanced Clocking Modes table — Standard Clocking Mode sub-topic — Sink Adaptation Module sub-topic — Standard Clocking Mode vs Advanced Clocking Mode topic • Removed the Continuous vs. Burst Mode Characteristics table. • Updated the SerialLite III Streaming IP Core FPGA Performance and Resource Utilization table for Arria 10, Stratix V GX and Arria V GZ, and Stratix 10 devices. • Updated the SerialLite III Streaming IP Core Parameters table: — Removed the Streaming Mode parameter. — Updated the description for User input parameter. • Updated the description in the Specifying IP Core Parameters and Options topic. • Updated the Simulation Parameters sub-topic to include testbench default simulation parameters for Arria 10 and Stratix 10 devices. • Updated the Simulator column in the Intel FPGA IP Core Simulation Scripts table. • Updated the Interlaken PHY IP Duplex Core or Native PHY IP Duplex Core - Interlaken Mode topic. • Updated the description of the `phy_mgmt_clk` signal in the SerialLite III Streaming IP Core Clock Domains and Signals table. • Updated the SerialLite III Streaming Sink Core topic. • Updated the Latency Measurement for Duplex Core table in the Core Latency topic to include Stratix 10 device. • Updated the second note in the Reset topic. • Updated the description in the CRC-32 Error Injection topic. • Updated the Clocking Structure for Stratix 10 Devices topic: — Added the Source and Sink descriptions for interface_clock signal in the Stratix 10 Clocks in Standard Clocking Mode table. — Added `tx_clkout` and `rx_clkout` signals in the Stratix 10 Clocks in Advanced Clocking Mode table. — Updated the description for Choosing TX PLL Type for Stratix 10 Devices sub-topic. • Updated the Clocking Structure For Arria 10 Devices topic: — Updated the Clocking Structure for Arria 10 Devices figure. — Updated the Source and Sink descriptions for interface_clock signal in the Arria 10 Clocks in Standard Clocking Mode table. — Updated the description for Choosing TX PLL Type for Arria 10 Devices sub-topic.

continued...

Date	Version	Changes
		• Updated the SerialLite III Streaming Link Debugging topic: — Updated the Source Core Link Debugging Flow Chart figure. — Updated the Source Link Debugging Signals table to include Stratix 10 support. — Updated the Sink Core Link Debugging Flow Chart figure. — Updated the Sink Link Debugging Signals table to include Stratix 10 support. • Updated the Error Handling topic: — Updated the sink core error flag. — Added information on error status on `error_rx` signal condition. • Updated the the Register Map for SerialLite III Streaming MAC table: — Added RX Error Status register. — Updated the source register name for 0x0090 from TX Error to TX Error Status. — Removed RX MAC status. • Updated the Configuration and Status Registers topic. • Minor typographical corrections and stylistic changes.
October 2016	2016.10.28	• Added information about Stratix 10 support. • Update document template.
May 2016	2016.05.02	• Added a new parameter—**Enable Transceiver Native PHY ADME** • Updated the IP core parameter names. • Added new sections: — SerialLite III Streaming IP Core Clocking Guidelines on page 69 — Creating a Signal Tap II Debug File to Match Your Design Hierarchy on page 84 • Revised the `core_reset` signal description—removed 32-cycle reset restriction. This restriction is removed in IP core version 15.1 onwards but is still applicable to prior versions. • Updated the IP Core release information. • Removed the design example chapter. The information is now located in the *Design Examples for SerialLite III Streaming IP Core User Guide*.
November 2015	2015.11.02	• Updated the IP Core Performance and Resource Utilization table. • Added a new topic—IP Core OpenCore Plus Timeout Behavior on page 11 • Added a link to *Introduction to Altera IP Cores*. • Added a note in "Altera IP Core Simulation Scripts" to recommend that you run the **msim_setup.tcl** script in the ModelSim-Altera Simulator Tcl console. • Changed the minimum required gap between bursts to one user clock cycle. • Added information about using I/O PLL to generate the core clock and user clock signals for Arria 10 devices. • Added a new parameter—**Burst Gap**. • Updated the parameter description for **Interface clock frequency**, **Core clock frequency**, and **fPLL reference clock frequency**. • Updated the parameter value for **Core clock frequency**. • Updated the PMA width for Interlaken mode to 64 bits for Arria 10 devices. • Updated all SerialLite III Streaming IP Core block diagrams. • Removed the Source PPM Absorption module from the core. • Changed the bit function and description for `error` (source core) and `error_tx` (duplex core) signals. • Updated the description of `link_up_rx` signal. • Added "Interlaken PHY Register Descriptions" table to specify the registers to access using the Avalon-MM PHY management interface.

continued...

Date	Version	Changes
		• Updated the design example to support Arria 10 devices. • Changed the target development kit to Transceiver Signal Integrity Development Kit, Stratix V GX Edition. • Updated the design operation names in the Design Example Operation topic. • Changed the sink link debug signal from `rx_crc32` to `rx_crc32err`. • Updated the sink core conditions in the Error Handling topic. • Changed instances of *Quartus II* to *Quartus Prime*.
May 2015	2015.05.04	• Updated the IP Core Performance and Resource Utilization table. • Changed the width of `sync_rx` and `sync_tx` signals from 4 to 8 bits in *Signals*. • Added external serial loopback in Testbench and Testbench.
December 2014	2014.12.15	Described Arria 10 support for up to 17.4 Gbps transceiver data rate. Updated core latency numbers. Updated the Transmission Overheads and Lane Rate Calculations on page 41. Minor text changes.
August 2014	2014.08.18	Added information about Arria 10 support.
June 2014	2014.06.30	Replaced references to MegaWizard Plug-In Manager with IP catalog or parameter editor. Minor text changes.
November 2013	2013.11.04	• Added information on CRC-32 error injection. • Added information on the FIFO ECC protection option.
May 2013	2013.05.13	Initial release