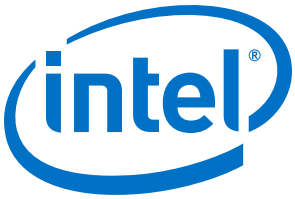


SerialLite III Streaming IP Core Design Example User Guide for Intel Stratix 10 Devices

Updated for Intel® Quartus® Prime Design Suite: **17.1 Stratix 10 ES Editions**

Subscribe
Send Feedback

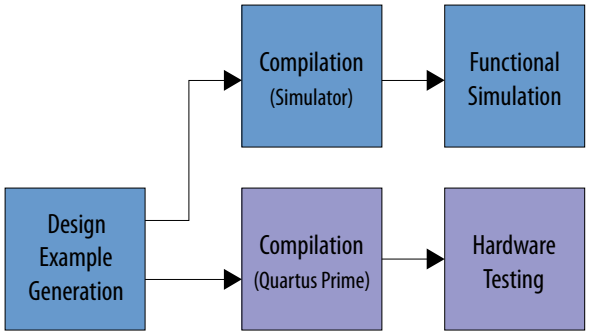
UG-20065 | 2017.11.06
Latest document on the web: **PDF** | **HTML**

Contents

1 Quick Start Guide

The SerialLite III Streaming IP core provides the capability of generating design examples for selected configurations.

Figure 1. Development Stages for the Design Example



1.1 Directory Structure

The Intel Quartus Prime software generates the design example files in the following folders:

- `<user_defined_design_example_directory>/ed_sim`
- `<user_defined_design_example_directory>/ed_synth`
- `<user_defined_design_example_directory>/ed_hwtest`

The following diagrams show the directories that contain the generated files for the design examples.

Figure 2. Directory Structure for Intel Stratix® 10 SerialLite III Streaming Design Example

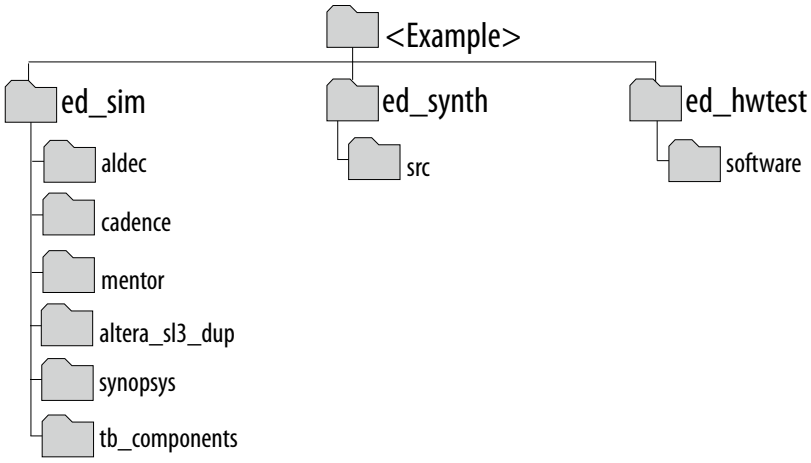
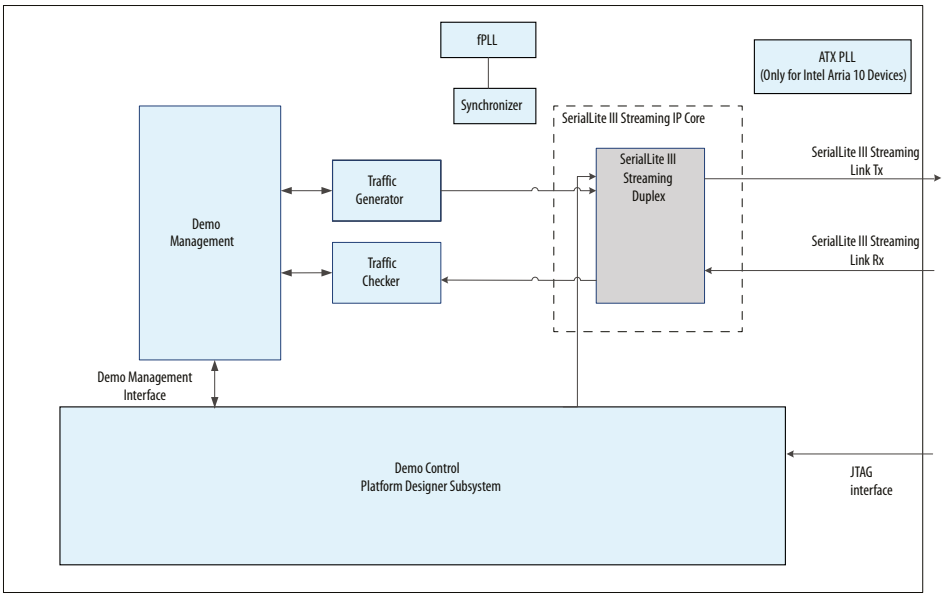


Table 1. Directory and File Description for Design Example Folder

Directory/File	Description
ed_sim/tb_components	The folder that contains the testbench files.
ed_sim/cadence ed_sim/mentor ed_sim/aldec ed_sim/synopsys/vcs or ed_sim/synopsys/vcsmx	The folder that contains the simulation script. It also serves as a working area for the simulator.
ed_sim/altera_sl3_dup	The folder that contains the design example simulation source files.
ed_synth/seriallite_iii_streaming_demo.qpf	Quartus project file.
ed_synth/seriallite_iii_streaming_demo.qsf	Quartus settings file.
ed_synth/seriallite_iii_streaming_demo.sdc	Synopsys Design Constraints (SDC) file.
ed_synth/src	The folder that contains the design example synthesizable components.
ed_synth/src/seriallite_iii_streaming_demo.v	Design example top-level HDL.
ed_synth/altera_sl3_dup/synth/altera_sl3_dup.v	Design example DUT top-level files.
ed_synth/demo_control	The folder for each synthesizable component including Platform Designer generated IPs, such as demo_mgmt and demo_control.
ed_hwtest	The folder that contains the design example hardware setup files.
ed_hwtest/Readme.txt	Instruction file to download the generated design example on the development kit.
ed_hwtest/master_export.v	User interface Verilog design file.
ed_hwtest/master_export_hw.tcl	Component description file for master export custom IP.
ed_hwtest/software	The folder that contains scripts to download the demo_control program into Nios® II processor and open an interactive terminal to run the design example.

1.2 Design Example Components

Figure 3. Design Example Block Diagram



1.3 Generating the Design

You can use the SerialLite III Streaming IP core parameter editor in the Intel Quartus Prime software to generate the design example.

Figure 4. Procedure

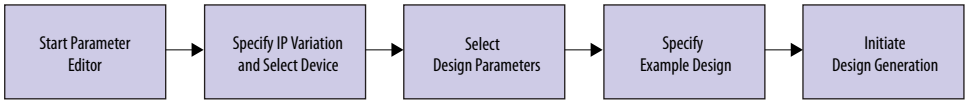
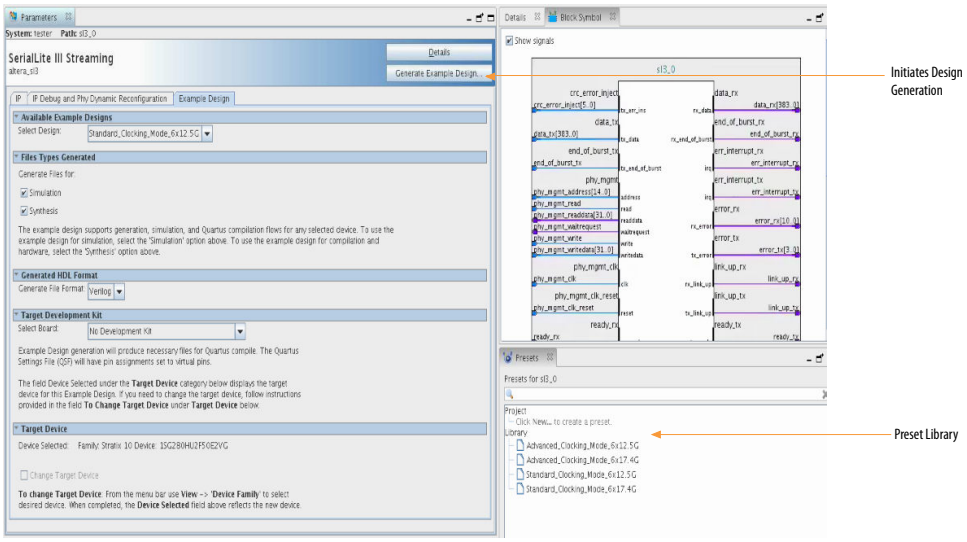


Figure 5. Example Design Tab



1.3.1 Procedure

This is a general procedure on how to generate the design example.

To generate the design example from the IP parameter editor:

1. In the IP Catalog (`Tools > IP Catalog`), locate and select **SerialLite III Streaming**. The IP parameter editor appears.
2. Specify a top-level name and the folder for your custom IP variation. Click **OK**.
3. Select a design from the **Presets** library and click **Apply**. When you select a design, the system automatically populates the IP parameters for the design.

   Note: If you select another design, the settings of the IP parameters change accordingly.
4. Specify the parameters for your design.
5. Click the **Generate Example Design** button.

The software generates all design files in the sub-directories. These files are required to run simulation, compilation, and hardware testing.

1.3.2 Design Example Parameters

The SerialLite III Streaming IP parameter editor includes an *Example Design* tab for you to specify certain parameters before generating the design example.

Table 2. Parameters in the Example Design Tab

Parameter	Description
Select Design	Available example designs for the IP parameter settings. When you select a design from the Preset library, this field shows the selected design.
Generate Files for	The files to generate for different development phases.

continued...

Parameter	Description
	Simulation—when selected, the necessary files for simulating the design example are generated. Synthesis—when selected, the synthesis files are generated. Use these files to compile the design in the Intel Quartus Prime software for hardware testing.
Generate File Format	The format of the RTL files for simulation—Verilog or VHDL.
Select Board	Supported hardware for design implementation. When you select an Intel FPGA development board, the *Target Device* is the one that matches the device on the Development Kit. If this menu is grayed out, there is no supported board for the options that you select. **Intel Stratix® 10 GX Signal Integrity Development Kit:** This option allows you to test the design example on selected Intel FPGA IP development kit. This selection automatically selects the *Target Device* to match the device on the Intel FPGA IP development kit. If your board revision has a different device grade, you can change the target device. **Custom Development Kit:** This option allows you to test the design example on a third party development kit with Intel FPGA IP device, a custom designed board with Intel FPGA IP device, or a standard Intel FPGA IP development kit not available for selection. You can also select a custom device for the custom development kit. **No Development Kit**: This option excludes the hardware aspects for the design example.
Change Target Device	Select a different device grade for Intel FPGA IP development kit. For device-specific details, refer to the device datasheet on the Intel FPGA website.

1.3.3 Presets

Standard presets allow instant entry of pre-selected parameter values in the **IP** and **Example Design** tabs. You can select the presets at the lower right window in the parameter editor.

The parameter values chosen for the presets belong to the group of supported SerialLite III Streaming IP configurations for design example generation. You can select one of the presets available for your target device to quickly generate a design example without having to manually set each parameter in the **IP** tab and verifying that the parameter matches the supported configurations set. There are four preset settings available in the library that support Duplex, Sink and Source modes:

- Advanced Clocking Mode 6x12.5G
- Advanced Clocking Mode 6x17.4G
- Standard Clocking Mode 6x12.5G
- Standard Clocking Mode 6x17.4G

Table 3. Parameter Settings for Intel Stratix 10 Design Example Presets

Intel Stratix 10 SerialLite III Streaming IP core design examples are only available in Intel Quartus Prime Pro Edition. Please contact your local Intel representative for more information.

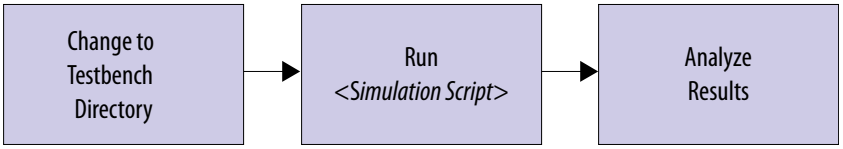
Presets	Advanced Clocking Mode 6x12.5G	Advanced Clocking Mode 6x17.4G	Standard Clocking Mode 6x12.5G	Standard Clocking Mode 6x17.4G
Direction	Duplex	Duplex	Duplex	Duplex
Number of lanes	6	6	6	6

continued...

Presets	Advanced Clocking Mode 6x12.5G	Advanced Clocking Mode 6x17.4G	Standard Clocking Mode 6x12.5G	Standard Clocking Mode 6x17.4G
Meta frame length in words	200	200	200	200
Transceiver reference clock frequency (MHz)	312.5	600.0	312.5	600.0
Enable M20K ECC support	ON/OFF The default value is OFF.	ON/OFF The default value is OFF.	ON/OFF The default value is OFF.	ON/OFF The default value is OFF.
Clocking Mode	Advanced clocking mode	Advanced clocking mode	Standard clocking mode	Standard clocking mode
Required user clock frequency (MHz)	182.835821	254.507463	177.556818	247.159091
Transceiver data rate (Gbps)	12.5	17.4	12.5	17.4

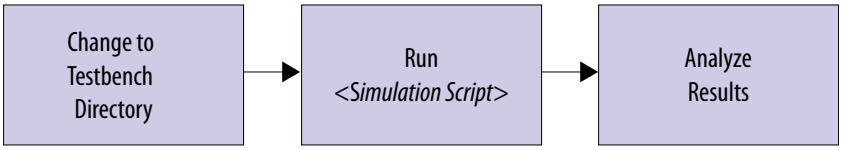
1.4 Simulating the Design

These general steps describe how to compile and run the design example simulation. For specific commands for each design example variant, refer to its respective section.



1.4.1 Procedure

To compile and simulate the design:



1. Change the working directory to <*example_design_directory*>**example/ed_sim/**<*simulator*>.
2. Run the simulation script for the simulator of your choice.

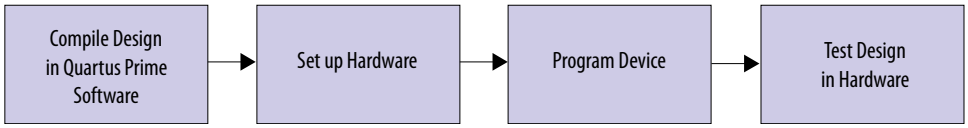
Simulator	Command
ModelSim*	`do run_tb.tcl`
VCS®/VCS MX	`sh run_tb.sh`
Aldec™	`do run_tb.tcl`
NC-Sim	`sh run_tb.sh`

A successful simulation ends with the following message, "Test Passed."

After successful completion, you can analyze the results.

1.5 Compiling and Testing the Design

The SerialLite III Streaming IP Core parameter editor allows you to compile and run the design example on a target development kit.



Follow these steps to compile and test the design in hardware:

1. Launch the Intel Quartus Prime software and change the directory to /**ed_synth**/ and open the **seriallite_iii_streaming_demo.qpf** file.
2. Click **Processing**> **Start Compilation** to compile the design.

   The timing constraints for the design example and the design components are automatically loaded during compilation.
3. Connect the development board to the host computer.
4. Configure the FPGA on the development board using the generated **.sof** file (**Tools**> **Programmer**).

The design examples target the Intel Stratix 10 GX Signal Integrity Development Kit.

The design includes an SDC script as well as a QSF with verified constraints in loopback mode. If you use the design example with another device or development board, you may need to update the device setting and constraints in the QSF file.

You must use correct pin constraints when using the core in simplex mode.

Related Links

SerialLite III Streaming IP Core User Guide

2 Detailed Description for Intel Stratix 10 SerialLite III Streaming Standard Clocking Mode

This design example demonstrates the functionalities of data streaming using standard clocking mode.

To generate the design example, select the following presets:

- Standard Clocking Mode 6x12.5G
- Standard Clocking Mode 6x17.4G

By default, the design examples are generated as duplex core.

Note: Intel Stratix 10 SerialLite III Streaming IP core design examples are only available in Intel Quartus Prime Pro Edition. Please contact your local Intel representative for more information.

Related Links

- Testbench on page 15
- SerialLite III Errata

2.1 Features

Features for Standard Clocking Mode design example includes:

- Support 6 lanes with up to 17.4 Gpbs transceiver data rate
- Support duplex transmission mode
- Traffic checker for data verification and lane de-skew verification
- Support CRC error injection using Nios II processor
- Support slave test mode for master and slave testing

2.2 Hardware and Software Requirements

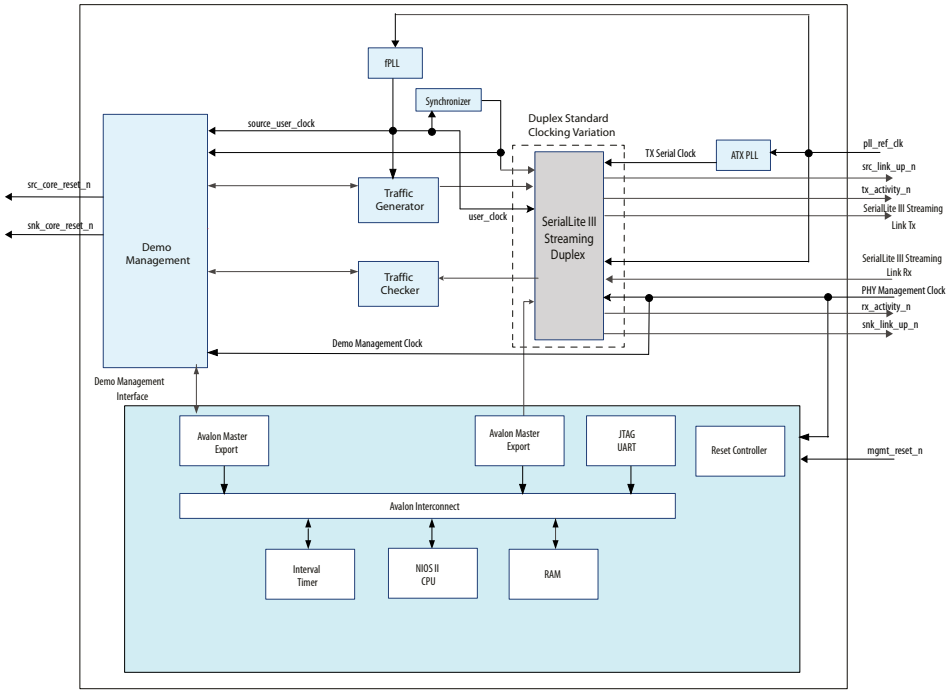
Intel uses the following hardware and software to test the example designs in a Linux system:

- Intel Quartus Prime software
- ModelSim - Intel FPGA Edition, ModelSim-SE, NC-Sim (Verilog only), or VCS simulator
- Intel Stratix 10 GX Signal Integrity Development Kit for hardware testing

2.3 Functional Description

The design examples consist of various components. The following block diagram shows the design components and the top level connections of the design example.

Figure 6. Design Example for Duplex Core in Standard Clocking Mode



2.3.1 Design Example Components

The design example consists of following components:

- SerialLite III Streaming IP core variation
- Source and sink user clock—fPLL
- ATX PLL
- Traffic generator
- Traffic checker
- Demo control
- Demo management

2.3.1.1 SerialLite III Streaming IP Core

The SerialLite III Streaming IP core variation accepts data from the traffic generator and formats the data for transmission. It also receives data from the link, strips the headers, and presents it to the traffic checker for analysis. The core is generated using the parameter editor in the Intel Quartus Prime software.

2.3.1.2 User Clock

The fPLL generates a user clock for sourcing and sinking data into the SerialLite III Streaming IP core.

2.3.1.3 Traffic Generator

The traffic generator generates traffic in a deterministic format to verify that data is transmitted correctly across the link. Traffic consists of sets of sample words, one for each lane on the link, that are presented to the source user interface.

Figure 7. Traffic Generator Sample Word Format

This figure shows the format of the sample words generated for each lane.

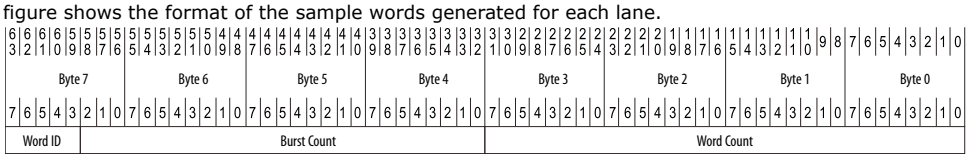


Table 4. Traffic Generator Sample Word Fields

Field	Bits	Description
Word ID	63–59	Contains a static value to distinguish which 64-bit word on the user interface that this sample was presented on. The Word ID value ranges from 0 to (lanes – 1).
Burst Count	58–32	Tracks the number of bursts used to transfer the sample data. This field value starts with one after reset and is incremented each time the `start_of_burst` signal is asserted on the source user interface.
Word Count	31–0	Tracks the number of valid sample words that have been transferred, across all bursts, to the source user interface.

2.3.1.4 Traffic Checker

The traffic checker performs the following inspections to verify that the received data conforms to the expected format:

- Checks each sample word to verify that the expected word ID was received.
- Checks each sample word to verify that the word count value is higher than the word count value from the last valid sample word.
- Verifies that lane de-skew has been properly performed by validating that the word count and burst count values from the sample word are the same as the values received from the adjacent lane.
- If the `start_of_burst` signal is asserted on the user interface, verifies that the burst count value in the current sample word is higher than the burst count value from the last valid sample word. Otherwise, it verifies that the burst count value has not changed.

2.3.1.5 Demo Control

The demo control module is a Nios II processor system, generated in Platform Designer (Standard), to control the demo hardware.

Demo control module also consists of a timer to track interrupt occurence, Avalon-MM interface to access demo management and the SerialLite III Streaming IP PHY interface, a reset controller, an UART interface, and an Avalon Streaming (Avalon-ST) interface.

2.3.1.6 Demo Management

The demo management module controls the user modules interaction with the SerialLite III Streaming IP core such as enable and disable traffic generator and traffic checker, enable CRC error insertion, and provide user clock reset for SerialLite III Streaming IP core. The module also implements CSRs to control and monitor the design operation. This includes CSRs to monitor and log errors that occur during the operation.

2.3.1.7 Nios II Processor Code

The Nios II processor controls the options exercised in the design example. The code also enables CRAM bits for CRC-32 error injection support.

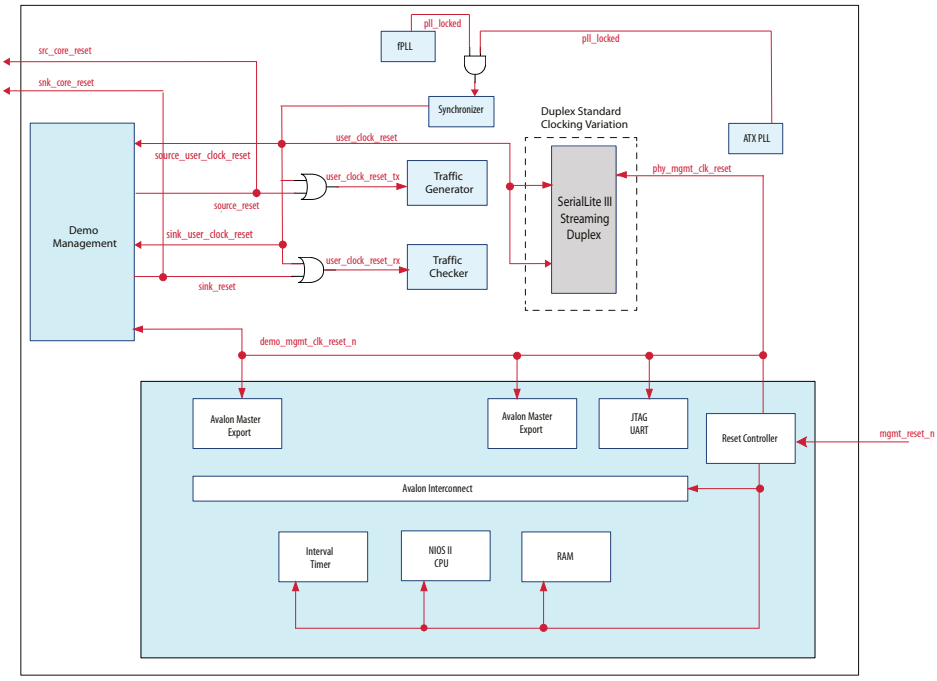
The design example sets the bit for channel 0 that connects to lane 0 in the design example. Therefore, CRC error injection is exercisable for lane 0 only. Refer to the Nios II processor source code (**demo_control.c**) for information on setting bits for other channels.

2.3.2 Reset Scheme

The `mgmt_reset_n` reset signal controls the overall reset structure for the design example. This is an asynchronous and active-low signal. Asserting this signal resets the demo control module and the SerialLite III Streaming IP core. The traffic generator and traffic checker modules get reset through the demo management and the reset synchronizer.

The following diagram show the reset scheme implemented in the design example.

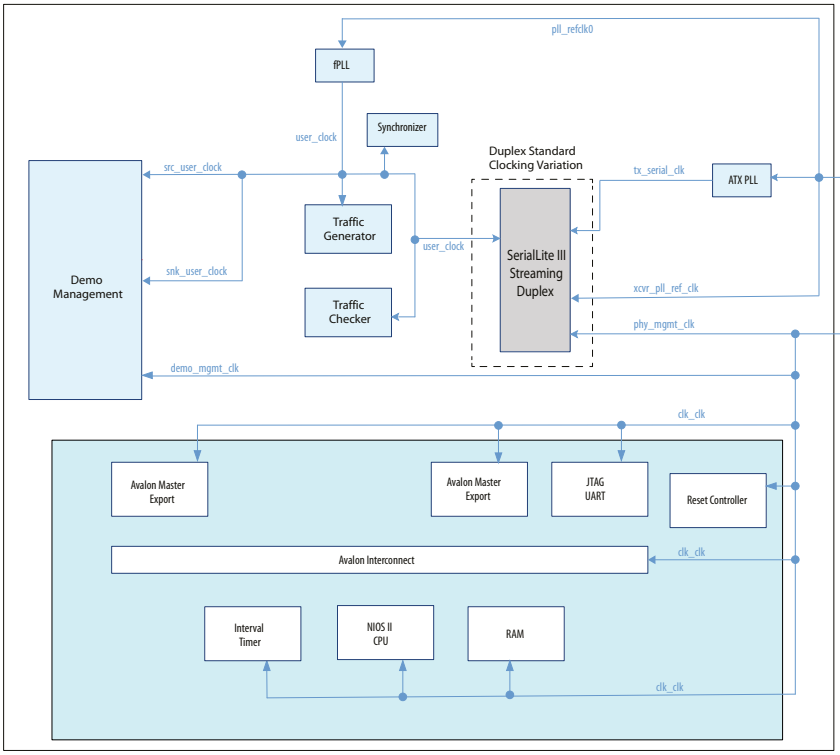
Figure 8. Reset Scheme for Intel Stratix 10 SerialLite III Streaming Duplex Core in Standard Clocking Mode



2.3.3 Clocking Scheme

The following diagrams show the clocking scheme for the design example.

Figure 9. Clocking Scheme for Intel Stratix 10 SerialLite III Streaming Duplex Core in Standard Clocking Mode



2.4 Simulation

The simulation test cases demonstrate continuous streaming of 2000 sample data for 6 lanes with 12.5 Gbps transceiver data rate from traffic generator to the SerialLite III Streaming source core and externally loopback to the sink core in standard clocking mode.

The simulation test case performs the following steps:

1. Initialize and configures SerialLite III Streaming IP core, traffic generator and traffic checker.
2. Traffic generator generates data and starts data transmission.
3. Logs and display link up status and burst information.
4. Traffic checker verifies received data and stop transmission.
5. Testbench logs and displays test result and test information.

2.4.1 Testbench

The generated example testbench is dynamic and has the same configuration as the IP. An external transceiver ATX PLL will be generated for both duplex and simplex directions.

Note: The Intel Stratix 10 example testbench includes the external transceiver PLL; the IP core does not include the transceiver PLL for these devices.

Figure 10. SerialLite III Streaming Example Testbench (Duplex) for Intel Stratix 10 Devices

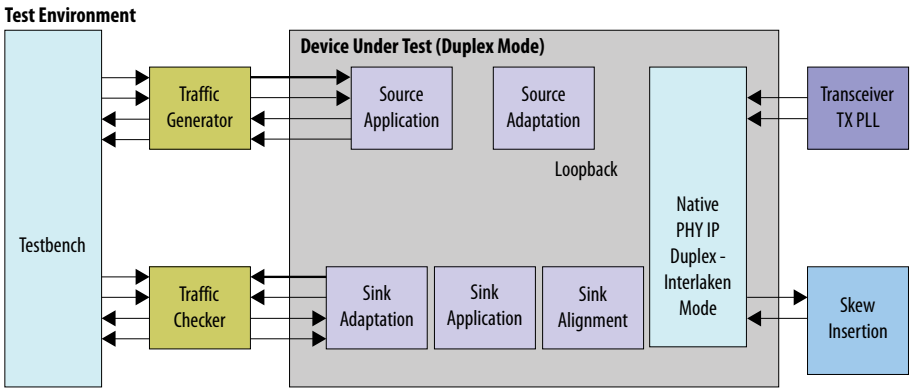
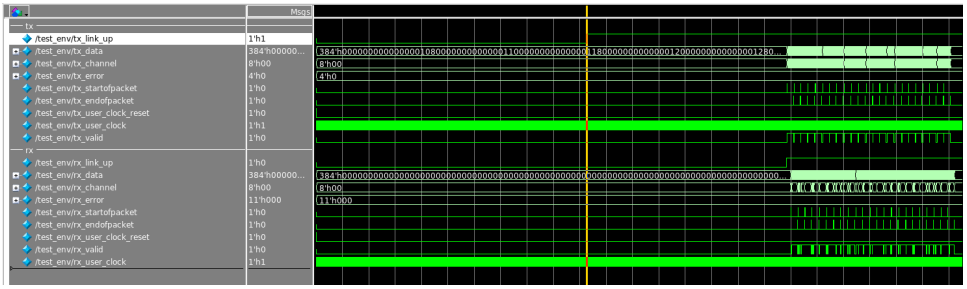


Figure 11. Simulation Waveform

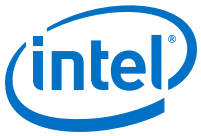


2.5 Hardware Testing

Once you download the design and accompanying software into the FPGA, you can test the design operation through the interactive session. The interactive session provides helpful statistics, as well as controls for controlling various aspects of the design.

You can control the following operations through the interactive session by entering the option numbers listed below:

- **0) Toggle Loopback Mode** - Switch between external loopback mode and internal loopback mode. By default, the design example is set to external loopback mode. The loopback mode is specified in interactive session. Disable the traffic generator/checker before switching loopback modes to avoid transmission error.
- **1) Enable Data Generator/Checker** - Enables the traffic generator and start sending out data.
- **2) Disable Data Generator/Checker** - Disables traffic generation.
- **3) Reset Source Core** - Resets the source core and traffic generator.
- **4) Reset Sink Core** - Resets the sink core and traffic checker.



- **5) Display Error Details** - Displays the error statistics.
- **6) Toggle Burst/Continuous Mode** - Resets the source and sink MACs and switches the traffic generator to generate a burst or continuous traffic stream. By default, the design example is set to burst mode. When in continuous mode, the burst count will always show 1. Disable the data generator/checker before switching mode to avoid transmission error.
- **7) Toggle CRC Error Insertion** - Turns CRC error injection off or on. By default, the design example has CRC error injection turned off.
- **8) Enable Slave Test Mode** - This option disables the traffic generator/checker and enables the traffic to flow from sink to source. This option is only available for hardware setup with master and slave configuration using two different development kits.
- **9) Disable Slave Test Mode** - This option disable data flow from sink to source. Select option 1 to enable the data generator and data checker.

Figure 12. Intel Stratix 10 SerialLite III Streaming Design Example Interactive Session

2.5.1 Design Setup

The design example targets the Intel Stratix 10 Transceiver Signal Integrity Development Kit.

The design includes an SDC script as well as a QSF with verified constraints in loopback mode. If you use the design example with another device or development board, you may need to update the device setting and constraints in the QSF file.

2.5.2 Error Details

These are the list of supported errors in the design example.

Table 5. List of Error Details

Error	Description
Sink Errors:	
Loss of Alignment During Normal Operation	To indicate loss of alignment error (`error_rx[1]`).
Meta Frame CRC Errors	To indicate CRC errors.
Lane Swap Errors	To indicate lane swap errors in traffic checker.
Lane Sequence Errors	To indicate lane sequence error in traffic checker.
Lane Alignment Errors	To indicate lane alignment error in traffic checker.

2.6 Signals

Figure 13. Top-level Signals for Intel Stratix 10 SerialLite III Streaming Standard Clocking Mode Design Example

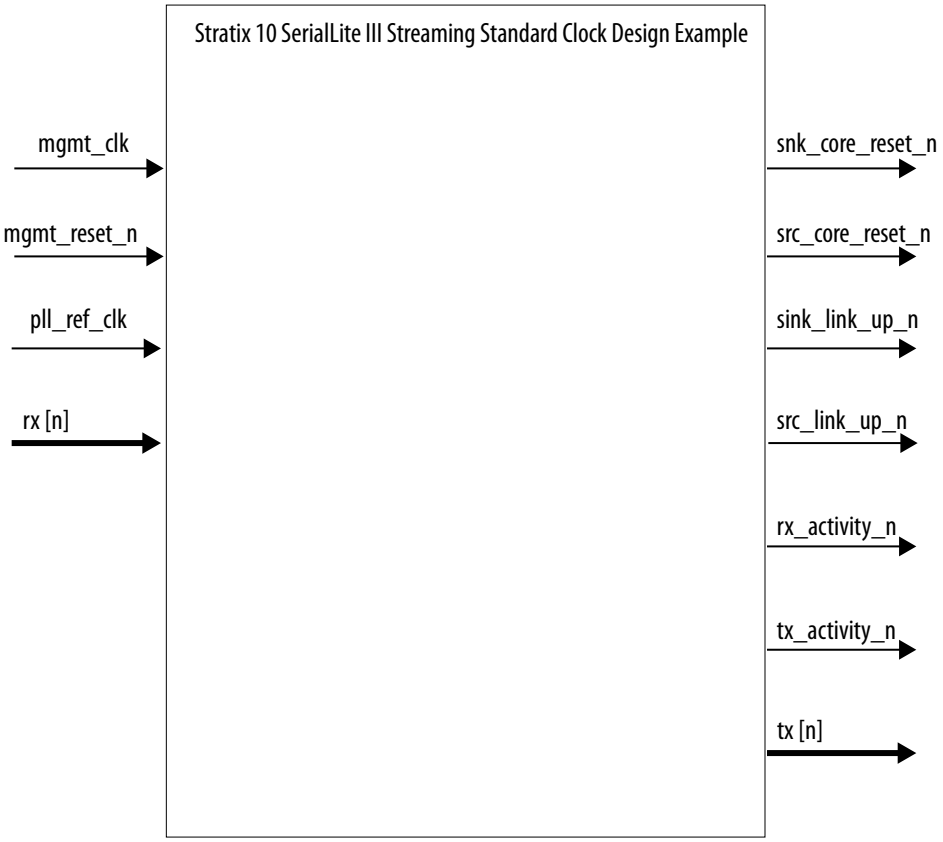


Table 6. Design Example Interface Signals

Signal	Direction	Width	Description
Clock and Reset Signal			
`mgmt_clk`	Input	1	Input clock for:

continued...

Signal	Direction	Width	Description
			• Avalon-MM PHY management interface for SerialLite III Streaming IP core • Demo management module • Demo control module • Transceiver reset controller
pll_ref_clk	Input	1	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.
mgmt_reset_n	Input	1	Design example asychrounous master reset. Assert this reset signal to reset the overall design example system. This is an active low signal.
snk_core_reset_n	Output	1	Demo management module asserts this signal to reset traffic checker module.
src_core_reset_n	Output	1	Demo management module asserts this signal to reset traffic generator module.
Data Signal			
rx[n]	Input	Based on **Number of Lanes** value	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
tx[n]	Output	Based on **Number of Lanes** value	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
Status Signal			
rx_activity_n	Output	1	This single bit signal indicates that the data is valid.
tx_activity_n	Output	1	This single bit signal indicates that the data is valid.
snk_link_up_n	Output	1	The core asserts this signal to indicate that the core initialization is complete and is ready to receive user data.
src_link_up_n	Output	1	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.

3 Detailed Description for Intel Stratix 10 SerialLite III Streaming Advanced Clocking Mode

This design example demonstrates the functionalities of data streaming using advanced clocking mode.

To generate the design example, select the following presets:

- Advanced Clocking Mode 6x12.5G
- Advanced Clocking Mode 6x17.4G

By default, the design examples are generated as duplex core.

3.1 Features

Features for Advanced Clocking Mode design example includes:

- Support 6 lanes with up to 17.4 Gpbs transceiver data rate
- Support duplex transmission modes
- Traffic checker for data verification and lane de-skew verification
- Support CRC error injection using Nios II processor

3.2 Hardware and Software Requirements

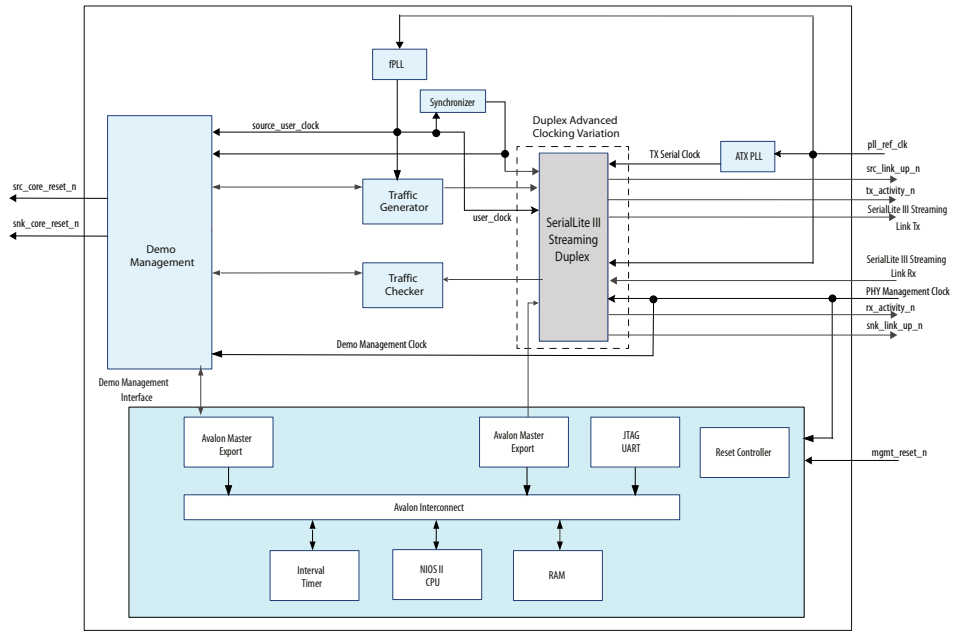
Intel uses the following hardware and software to test the example designs in a Linux system:

- Intel Quartus Prime software
- ModelSim - Intel FPGA Edition, ModelSim-SE, NC-Sim (Verilog only), or VCS simulator
- Intel Stratix 10 GX Signal Integrity Development Kit for hardware testing

3.3 Functional Description

The design example consists of various components. The following block diagram shows the design components and the top level connections of the design example.

Figure 14. Design Example for Duplex Core in Advanced Clocking Mode



3.3.1 Design Example Components

The design example consists of following components:

- SerialLite III Streaming IP core variation
- Source user clock—fPLL
- ATX PLL
- Traffic generator
- Traffic checker
- Demo control
- Demo management

3.3.1.1 SerialLite III Streaming IP Core

The SerialLite III Streaming IP core variation accepts data from the traffic generator and formats the data for transmission. It also receives data from the link, strips the headers, and presents it to the traffic checker for analysis. The core is generated using the parameter editor in the Intel Quartus Prime software.

3.3.1.2 User Clock

The fPLL generates a user clock for sourcing data into the SerialLite III Streaming IP core.

3.3.1.3 Traffic Generator

The traffic generator generates traffic in a deterministic format to verify that data is transmitted correctly across the link. Traffic consists of sets of sample words, one for each lane on the link, that are presented to the source user interface.

Figure 15. Traffic Generator Sample Word Format

This figure shows the format of the sample words generated for each lane.

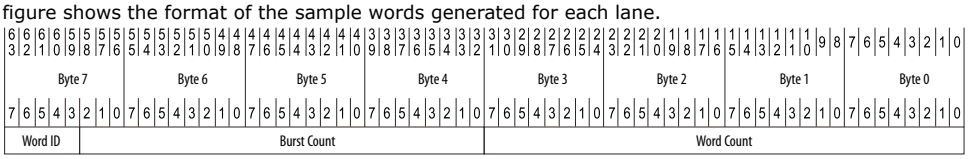


Table 7. Traffic Generator Sample Word Fields

Field	Bits	Description
Word ID	63–59	Contains a static value to distinguish which 64-bit word on the user interface that this sample was presented on. The Word ID value ranges from 0 to (lanes – 1).
Burst Count	58–32	Tracks the number of bursts used to transfer the sample data. This field value starts with one after reset and is incremented each time the `start_of_burst` signal is asserted on the source user interface.
Word Count	31–0	Tracks the number of valid sample words that have been transferred, across all bursts, to the source user interface.

3.3.1.4 Traffic Checker

The traffic checker performs the following inspections to verify that the received data conforms to the expected format:

- Checks each sample word to verify that the expected word ID was received.
- Checks each sample word to verify that the word count value is higher than the word count value from the last valid sample word.
- Verifies that lane de-skew has been properly performed by validating that the word count and burst count values from the sample word are the same as the values received from the adjacent lane.
- If the `start_of_burst` signal is asserted on the user interface, verifies that the burst count value in the current sample word is higher than the burst count value from the last valid sample word. Otherwise, it verifies that the burst count value has not changed.

3.3.1.5 Demo Control

The demo control module is a Nios II processor system, generated in Platform Designer (Standard), to control the demo hardware.

Demo control module also consists of a timer to track interrupt occurence, Avalon-MM interface to access demo management and the SerialLite III Streaming IP PHY interface, a reset controller, an UART interface, and an Avalon Streaming (Avalon-ST) interface.

3.3.1.6 Demo Management

The demo management module controls the user modules interaction with the SerialLite III Streaming IP core such as enable and disable traffic generator and traffic checker, enable CRC error insertion, and provide user clock reset for SerialLite III Streaming IP core. The module also implements CSRs to control and monitor the design operation. This includes CSRs to monitor and log errors that occur during the operation.

3.3.1.7 Nios II Processor Code

The Nios II processor controls the options exercised in the design example. The code also enables CRAM bits for CRC-32 error injection support.

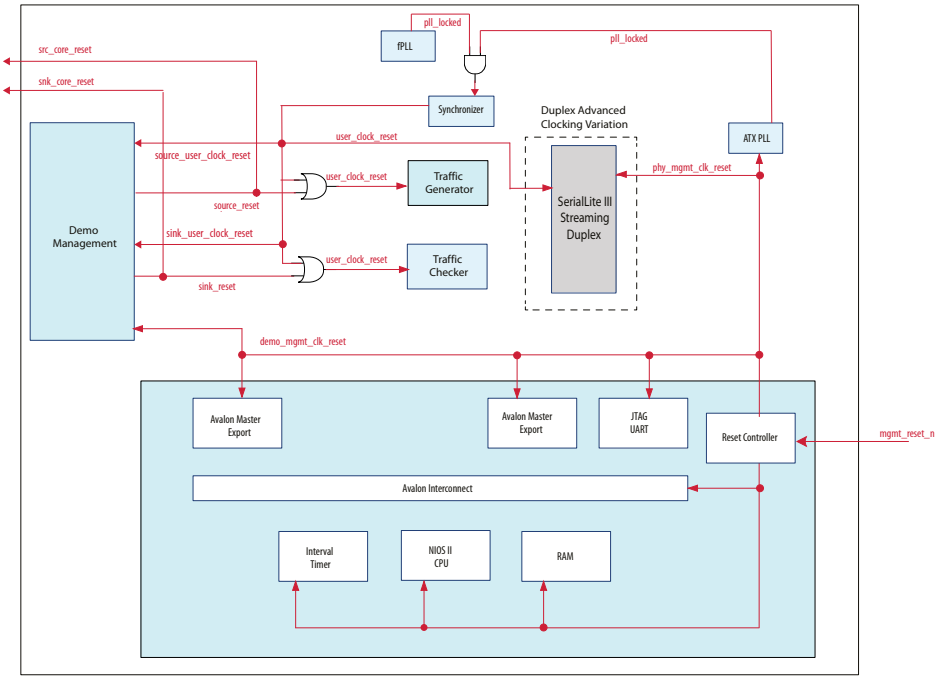
The design example sets the bit for channel 0 that connects to lane 0 in the design example. Therefore, CRC error injection is exercisable for lane 0 only. Refer to the Nios II processor source code (**demo_control.c**) for information on setting bits for other channels.

3.3.2 Reset Scheme

The `mgmt_reset_n` reset signal controls the overall reset structure for the design example. This is an asynchronous and active-low signal. Asserting this signal resets the demo control module and the SerialLite III Streaming IP core. The traffic generator and traffic checker modules get reset through the demo management and the reset synchronizer.

The following diagrams show the reset scheme implemented in the design example.

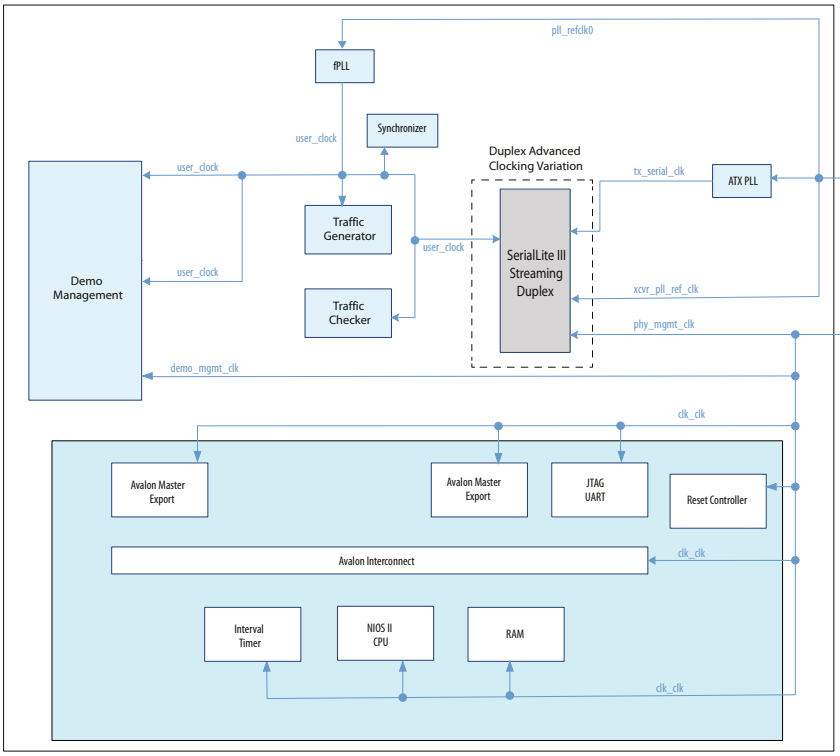
Figure 16. Reset Scheme for Intel Stratix 10 SerialLite III Streaming Duplex Core in Advanced Clocking Mode



3.3.3 Clocking Scheme

The following diagram show the clocking scheme for the design example.

Figure 17. Clocking Scheme for Intel Stratix 10 SerialLite III Streaming Duplex Core in Advanced Clocking Mode



3.4 Simulation

The simulation test cases demonstrate continuous streaming of 2000 sample data for 6 lanes with 12.5 Gbps transceiver data rate from traffic generator to the SerialLite III Streaming source core and externally loopback to the sink core in advanced clocking mode.

The simulation test case performs the following steps:

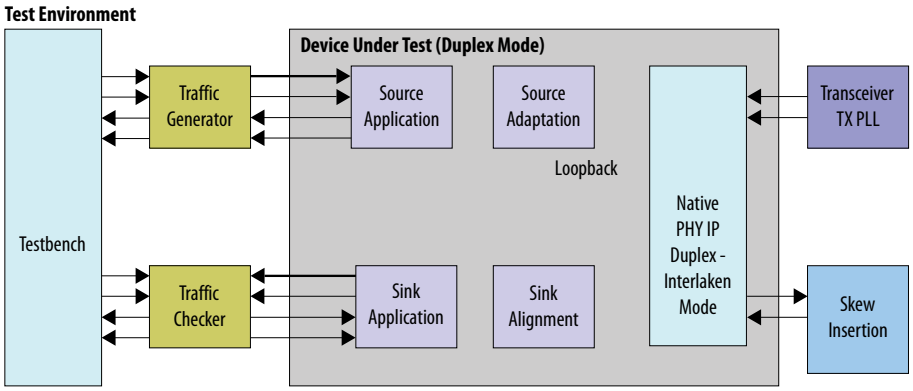
1. Initialize and configures SerialLite III Streaming IP core, traffic generator and traffic checker.
2. Traffic generator generates data and starts data transmission.
3. Logs and display link up status and burst information.
4. Traffic checker verifies received data and stop transmission.
5. Testbench logs and displays test result and test information.

3.4.1 Testbench

The generated example testbench is dynamic and has the same configuration as the IP. An external transceiver ATX PLL will be generated for both duplex and simplex directions.

Note: The Intel Stratix 10 example testbench includes the external transceiver PLL; the IP core does not include the transceiver PLL for these devices.

Figure 18. SerialLite III Streaming Example Testbench (Duplex) for Intel Stratix 10 Devices



3.5 Hardware Testing

Once you download the design and accompanying software into the FPGA, you can test the design operation through the interactive session. The interactive session provides helpful statistics, as well as controls for controlling various aspects of the design.

You can control the following operations through the interactive session by entering the option numbers listed below:

- **0) Toggle Loopback Mode** - Switch between external loopback mode and internal loopback mode. By default, the design example is set to external loopback mode. The loopback mode is specified in interactive session. Disable the traffic generator/checker before switching loopback modes to avoid transmission error.
- **1) Enable Data Generator/Checker** - Enables the traffic generator and start sending out data.
- **2) Disable Data Generator/Checker** - Disables traffic generation.
- **3) Reset Source Core** - Resets the source core and traffic generator.
- **4) Reset Sink Core** - Resets the sink core and traffic checker.

- **5) Display Error Details** - Displays the error statistics.
- **6) Toggle Burst/Continuous Mode** - Resets the source and sink MACs and switches the traffic generator to generate a burst or continuous traffic stream. By default, the design example is set to burst mode. When in continuous mode, the burst count will always show 1. Disable the data generator/checker before switching mode to avoid transmission error.
- **7) Toggle CRC Error Insertion** - Turns CRC error injection off or on. By default, the design example has CRC error injection turned off.

Figure 19. Intel Stratix 10 SerialLite III Streaming Design Example Interactive Session

Note: Options 8 and 9 only work in standard clocking mode.

3.5.1 Design Setup

The design example targets the Intel Stratix 10 Transceiver Signal Integrity Development Kit.

The design includes an SDC script as well as a QSF with verified constraints in loopback mode. If you use the design example with another device or development board, you may need to update the device setting and constraints in the QSF file.

3.5.2 Error Details

These are the list of supported errors in the design example.

Table 8. List of Error Details

Error	Description
Sink Errors:	
Loss of Alignment During Normal Operation	To indicate loss of alignment error (`error_rx[1]`).
Meta Frame CRC Errors	To indicate CRC errors.
Lane Swap Errors	To indicate lane swap errors in traffic checker.
Lane Sequence Errors	To indicate lane sequence error in traffic checker.
Lane Alignment Errors	To indicate lane alignment error in traffic checker.

3.6 Signals

Figure 20. Top-level Signals for Intel Stratix 10 SerialLite III Streaming Advanced Clocking Mode Design Example

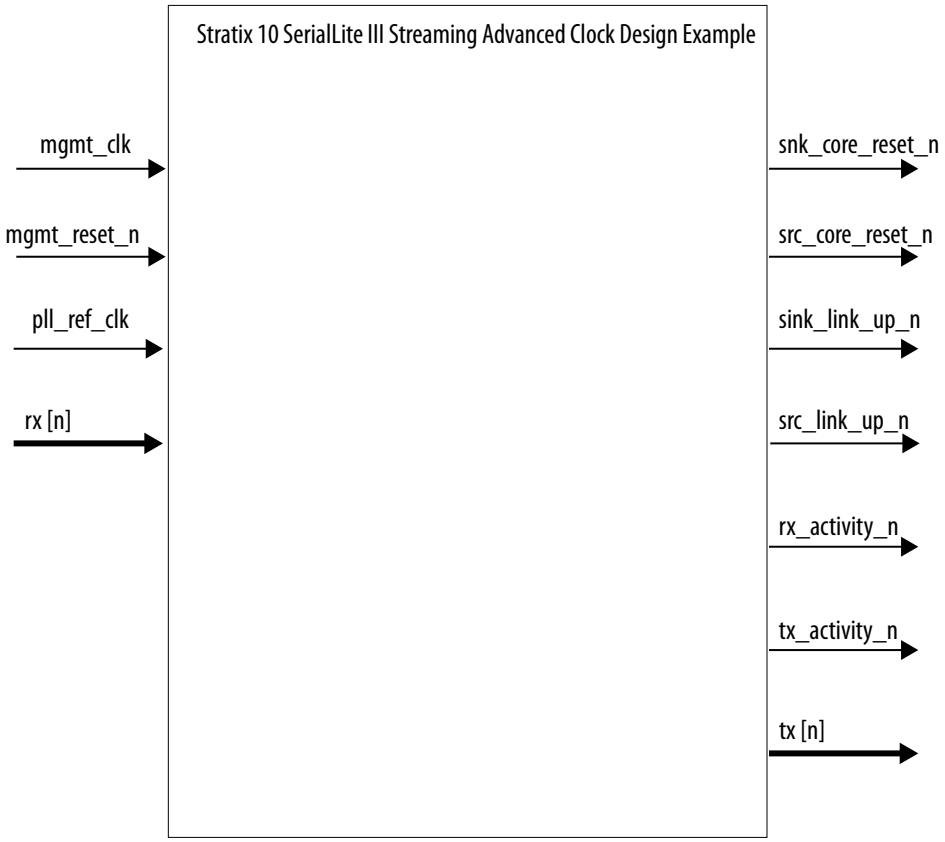


Table 9. Design Example Interface Signals

Signal	Direction	Width	Description
Clock and Reset Signal			
`mgmt_clk`	Input	1	Input clock for: • Avalon-MM PHY management interface for SerialLite III Streaming IP core • Demo management module • Demo control module • Transceiver reset controller
`pll_ref_clk`	Input	1	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

continued...

Signal	Direction	Width	Description
			match the frequency of the tx_pll_ref_clk reference clock for the TX PLL at the Source variant.
mgmt_reset_n	Input	1	Design example asychrounous master reset. Assert this reset signal to reset the overall design example system. This is an active low signal.
snk_core_reset_n	Output	1	Demo management module asserts this signal to reset traffic checker module.
src_core_reset_n	Output	1	Demo management module asserts this signal to reset traffic generator module.
Data Signal			
rx[n]	Input	Based on **Number of Lanes** value	This vector carries the transmitted streaming data from the core. *N* represents the number of lanes.
tx[n]	Output	Based on **Number of Lanes** value	This vector carries the transmitted streaming data to the core. *N* represents the number of lanes.
Status Signal			
rx_activity_n	Output	1	This single bit signal indicates that the data is valid.
tx_activity_n	Output	1	This single bit signal indicates that the data is valid.
snk_link_up_n	Output	1	The core asserts this signal to indicate that the core initialization is complete and is ready to receive user data.
src_link_up_n	Output	1	The core asserts this signal to indicate that the core initialization is complete and is ready to transmit user data.

4 Document Revision History for SerialLite III Streaming IP Core Design Example User Guide for Intel Stratix 10 Devices

Date	Version	Changes
November 2017	2017.11.06	• Rebranded as Intel. • Renamed the document as *SerialLite III Streaming IP Core Design Example User Guide for Intel Stratix 10 Devices*. • Updated the "Parameters in the Example Design Tab" table: Updated the descriptions for **Select Board** parameter.
May 2017	2017.05.08	• Initial release.