



---

## Serial Lite III Streaming Intel Arria 10 FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**



**Subscribe**

**Send Feedback**

**UG-20030 | 2019.05.13**

Latest document on the web: [PDF](#) | [HTML](#)



# Contents

---

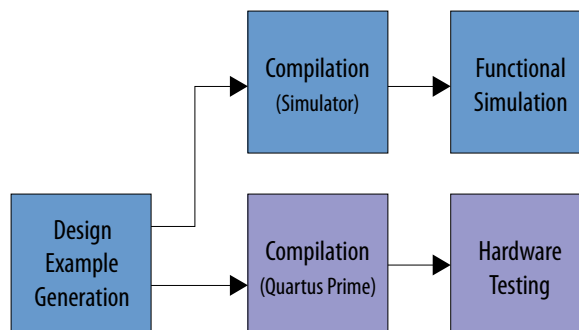
- 1. Quick Start Guide..... 3**
  - 1.1. Directory Structure..... 3
  - 1.2. Design Example Components..... 5
  - 1.3. Generating the Design.....5
    - 1.3.1. Procedure..... 6
    - 1.3.2. Design Example Parameters..... 6
    - 1.3.3. Presets..... 7
  - 1.4. Simulating the Design..... 8
    - 1.4.1. Procedure..... 8
  - 1.5. Compiling and Testing the Design..... 9
- 2. Detailed Description for Intel Arria 10 Serial Lite III Streaming Standard Clocking Mode..... 10**
  - 2.1. Features..... 10
  - 2.2. Hardware and Software Requirements..... 10
  - 2.3. Functional Description..... 11
    - 2.3.1. Design Example Components.....12
    - 2.3.2. Reset Scheme..... 14
    - 2.3.3. Clocking Scheme..... 15
  - 2.4. Simulation..... 17
    - 2.4.1. Testbench..... 17
  - 2.5. Hardware Testing..... 18
    - 2.5.1. Design Setup.....19
    - 2.5.2. Error Details.....19
  - 2.6. Signals..... 20
- 3. Detailed Description for Intel Arria 10 Serial Lite III Streaming Advanced Clocking Mode..... 22**
  - 3.1. Features..... 22
  - 3.2. Hardware and Software Requirements..... 22
  - 3.3. Functional Description..... 22
    - 3.3.1. Design Example Components.....24
    - 3.3.2. Reset Scheme..... 27
    - 3.3.3. Clocking Scheme..... 28
  - 3.4. Simulation..... 30
    - 3.4.1. Testbench..... 31
  - 3.5. Hardware Testing..... 32
    - 3.5.1. Design Setup.....32
    - 3.5.2. Error Details.....32
  - 3.6. Signals..... 33
- 4. Intel Arria 10 SerialLite III Streaming IP Core Design Example User Guide Document Archives..... 35**
- 5. Document Revision History for Serial Lite III Streaming IP Core Design Example User Guide for Intel Arria 10..... 36**

## 1. Quick Start Guide

---

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

**Figure 1. Development Stages for the Design Example**



### Related Information

[Intel Arria 10 SerialLite III Streaming IP Core Design Example User Guide Document Archives](#) on page 35

Provides a list of user guides for previous versions of the Serial Lite III Streaming IP Core Design Example.

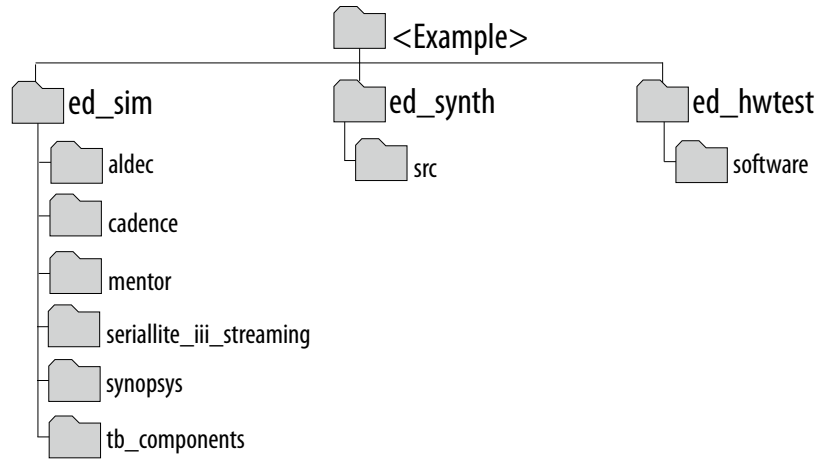
### 1.1. Directory Structure

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 Arria® 10 Serial Lite III Streaming Design Example**



**Table 1. Directory and File Description for Design Example Folder**

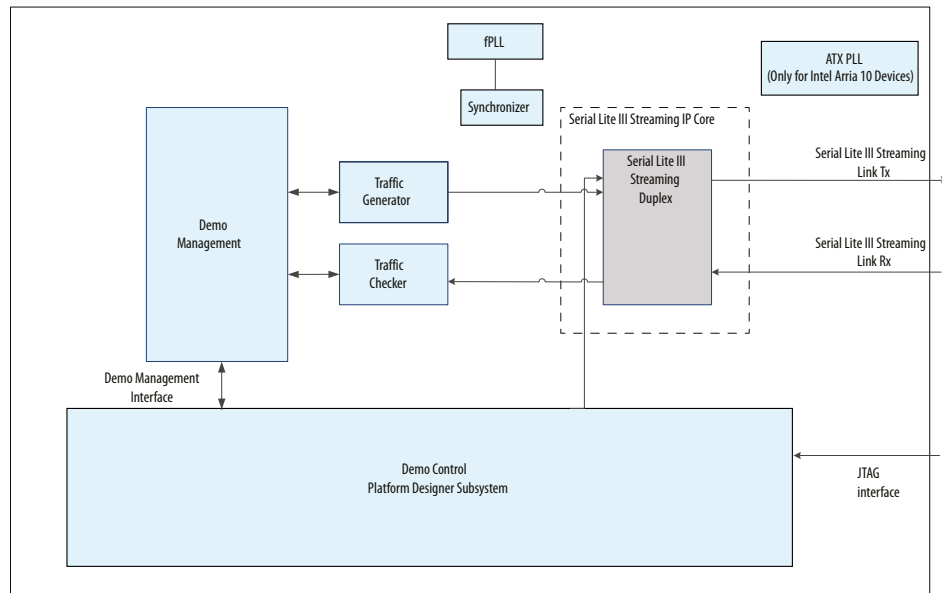
Directory/File	Description
ed_sim/tb_components	The folder that contains the testbench files.
ed_sim/cadence ed_sim/mentor ed_sim/aldec ed_sim/synopsys/vcs or ed_sim/synopsys/vcsmx	The folder that contains the simulation script. It also serves as a working area for the simulator.
ed_sim/seriallite_iii_streaming	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/src/seriallite_iii_streaming/synth/seriallite_iii_streaming.v	Design example DUT top-level files.
ed_synth/src/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/build_demo_control.sh	A script to generate demo control and PLL IPs with NIOS II processor.
<i>continued...</i>	



Directory/File	Description
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 Serial Lite III Streaming IP core parameter editor in the Intel Quartus Prime software to generate the design example.

Figure 4. Procedure

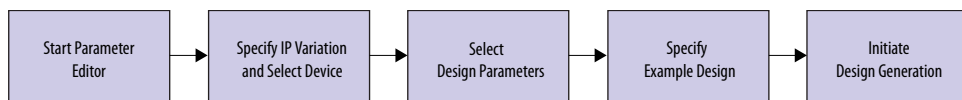
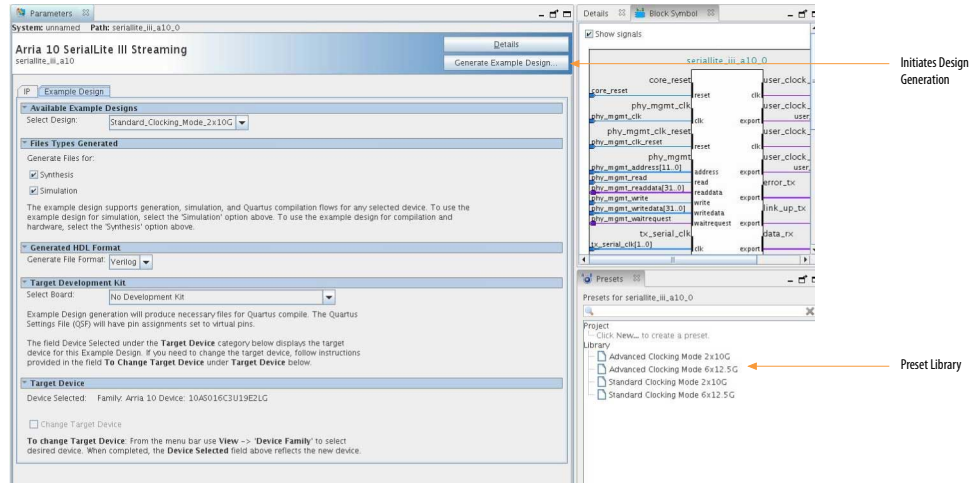


Figure 5. Example Design Tab



### 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 **Serial Lite III Streaming**. The IP parameter editor appears.
2. Specify a top-level name and the folder for your custom IP variation, and the target device. Click **OK**.
3. Select a design from the **Presets** library. 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. Only designs generated from the presets support hardware design examples.

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 Serial Lite 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 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 development board, the <i>Target Device</i> 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. <b>Intel Arria® 10 GX Transceiver Signal Integrity Development Kit:</b> This option allows you to test the design example on selected Intel development kit. This selection automatically selects the <i>Target Device</i> to match the device on the Intel development kit. If your board revision has a different device grade, you can change the target device. <b>Custom Development Kit:</b> This option allows you to test the design example on a third party development kit with Intel device, a custom designed board with Intel device, or a standard Intel development kit not available for selection. You can also select a custom device for the custom development kit. <b>No Development Kit:</b> This option excludes the hardware aspects for the design example.
Change Target Device	Select a different device grade for Intel 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 Serial Lite 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.

*Note:* Only designs generated from the presets support hardware design examples.

There are four preset settings available in the library that support Duplex, Sink and Source modes:

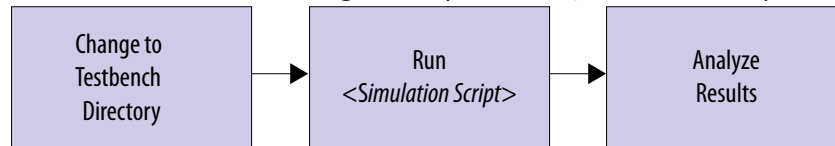
- Advanced Clocking Mode 2x10G
- Advanced Clocking Mode 6x12.5G
- Standard Clocking Mode 2x10G
- Standard Clocking Mode 6x12.5G

**Table 3. Parameter Settings for Intel Arria 10 Design Example Presets**

Presets	Advanced Clocking Mode 2x10G	Advanced Clocking Mode 6x12.5G	Standard Clocking Mode 2x10G	Standard Clocking Mode 6x12.5G
Direction	Duplex, Sink, and Source	Duplex, Sink, and Source	Duplex, Sink, and Source	Duplex, Sink, and Source
Number of lanes	2	6	2	6
Meta frame length in words	200	8191	200	8191
Transceiver reference clock frequency (MHz)	644.53125	312.5	644.531187	312.5
Enable M20K ECC support	ON and OFF	ON and OFF	ON and OFF	ON and OFF
Clocking Mode	Advanced clocking mode	Advanced clocking mode	Standard clocking mode	Standard clocking mode
Required user clock frequency (MHz)	150.8395522	186.4760558	146.484375	177.556818
Transceiver data rate (Gbps)	10.3125	12.5	10.312499	12.5

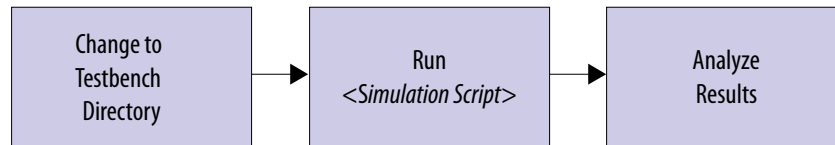
## 1.4. Simulating the Design

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



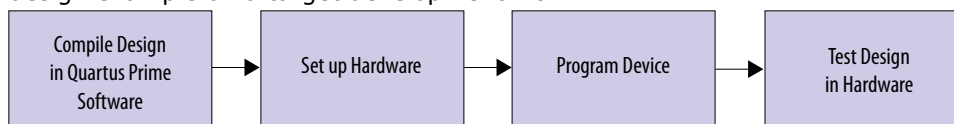


A successful simulation ends with the following message, "Test Passed."

After successful completion, you can analyze the results.

## 1.5. Compiling and Testing the Design

The Serial Lite 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 Intel Quartus Prime version 15.1 only supports programming file generation for Arria 10 engineering sample devices. For more information on support for Arria 10 production devices, contact your local Intel representative or use the support link from Intel website.

The design examples target the Intel Arria 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.

You must use correct pin constraints when using the core in simplex mode or when using more than one reconfiguration controller. The synthesized design typically includes a reconfiguration interface for at least three channels because three channels share an Avalon<sup>®</sup> Memory Mapped (Avalon-MM) slave interface, which connects to the Transceiver Reconfiguration Controller IP core. Conversely, you cannot connect three channels that share an Avalon-MM interface to different Transceiver Reconfiguration Controller IP cores or you will receive a Fitter error.

## 2. Detailed Description for Intel Arria 10 Serial Lite III Streaming Standard Clocking Mode

---

These design examples demonstrate the functionalities of data streaming using standard clocking mode.

To generate the design examples, select the following preset(s):

- Standard Clocking Mode 2x10G
- Standard Clocking Mode 6x12.5G

By default, the design examples are generated as duplex core. To generate the design examples in simplex core, select **Simplex** for the **Direction** parameter.

### Related Information

- [Testbench](#) on page 17
- [Serial Lite III Errata](#)

### 2.1. Features

Features for Standard Clocking Mode 2x10G design example includes:

- Support 2 lanes with 10Gpbs transceiver data rate
- Support simplex and duplex transmission modes
- Traffic checker for data verification and lane de-skew verification
- Support CRC error injection using Nios II processor

Features for Standard Clocking Mode 6x12.5G design example includes:

- Support 6 lanes with 12.5Gpbs transceiver data rate
- Support simplex and duplex transmission modes
- Traffic checker for data verification and lane de-skew verification
- Support CRC error injection using Nios II processor

### 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 Arria 10 GX Transceiver Signal Integrity Development Kit for hardware testing



## 2.3. Functional Description

The design examples consist of various components. The following block diagrams show the design components and the top level connections of the design examples.

**Figure 6. Design Example for Simplex Core in Standard Clocking Mode**

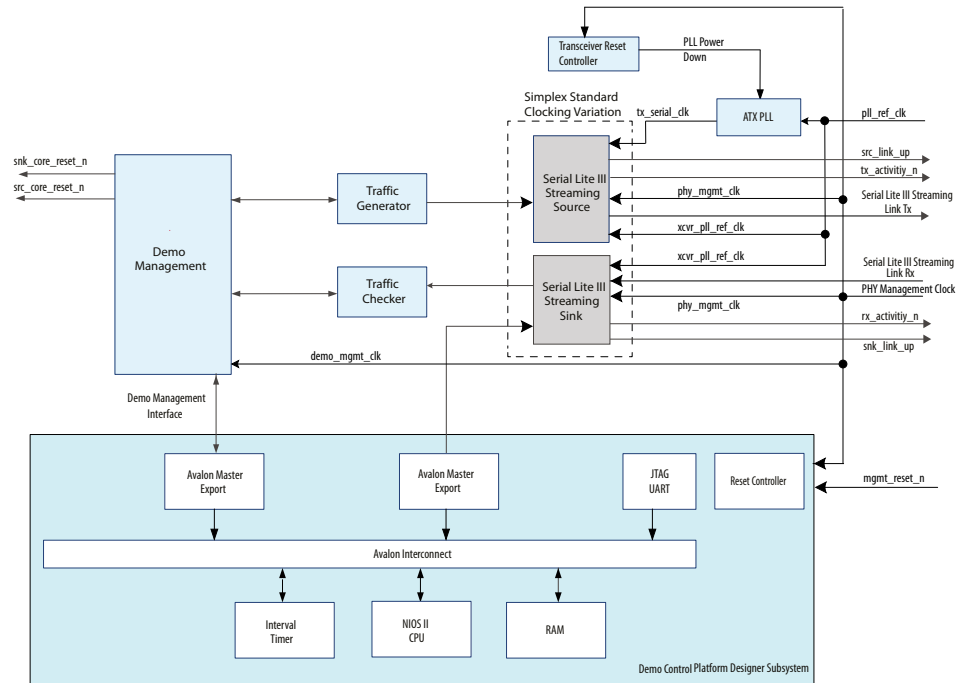
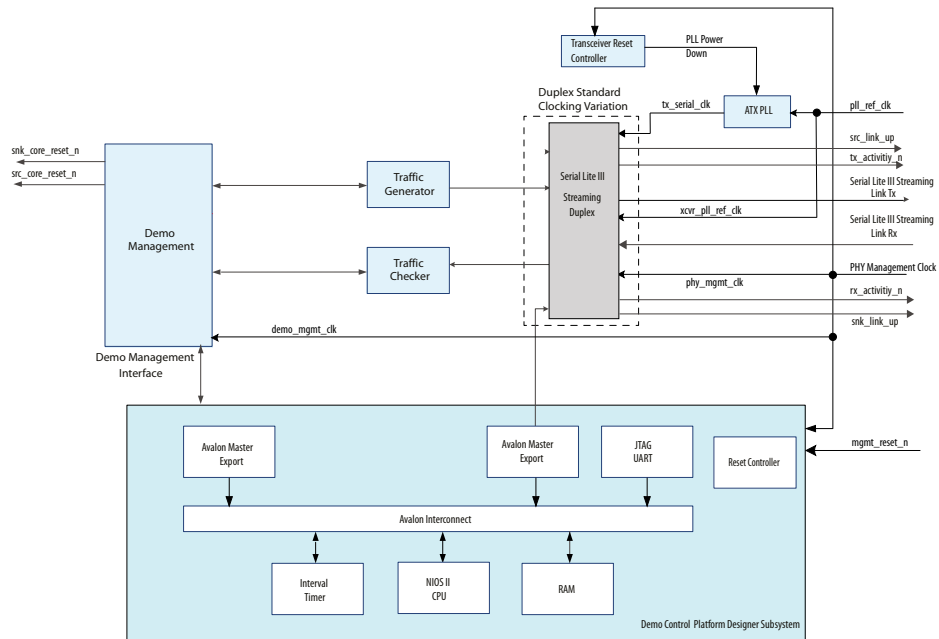


Figure 7. Design Example for Duplex Core in Standard Clocking Mode



### 2.3.1. Design Example Components

The design example consists of following components:

- Serial Lite III Streaming IP core variation
- ATX PLL
- Traffic generator
- Traffic checker
- Demo control
- Demo management
- Nios® II processor code

#### 2.3.1.1. Serial Lite III Streaming IP Core

The Serial Lite III Streaming IP core in this variant can either accept data from the traffic generator and format it for transmission or receive data from the link, strip the headers, and present it to the traffic checker for analysis. The core is generated with the parameter settings you select using the parameter editor in the Intel Quartus Prime software.

#### 2.3.1.2. ATX PLL

The ATX PLL generates transmit transceiver clock to the Serial Lite III Streaming IP core.



### Related Information

How to compensate the jitter of PLL cascading or non-dedicated clock path for Intel Arria 10 reference clock?

In previous Quartus Prime versions, reference clock from a cascaded PLL output, global clock, or core clock causes additional jitter in Intel Arria 10 Serial Lite III Streaming IP core design example. If you are migrating your design from earlier versions to Quartus Prime 17.1, refer to the KDB link.

### 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 8. 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 <code>start_of_burst</code> 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 occurrence, Avalon-MM interface to access demo management and the Serial Lite III Streaming Intel FPGA IP PHY interface, a reset controller, a 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 Serial Lite III Streaming IP core such as enable and disable traffic generator and traffic checker, enable CRC error insertion, and provide user clock reset for Serial Lite 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 error injection support in 10G PCS is based on groups of three channels or triplets. Setting the corresponding bit for a given channel in the triplet enables CRC error injection for all of the lanes that use any channel in the given triplet.

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.

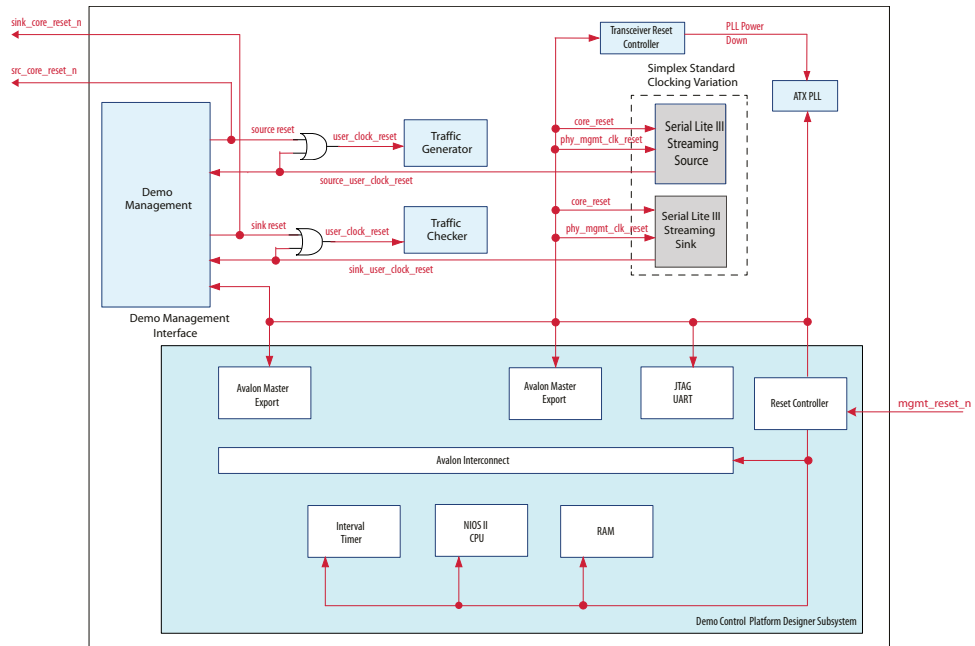
### 2.3.2. Reset Scheme

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 Serial Lite III Streaming IP core. The traffic generator and traffic checker modules get reset through the demo management and the Serial Lite III Streaming IP core.

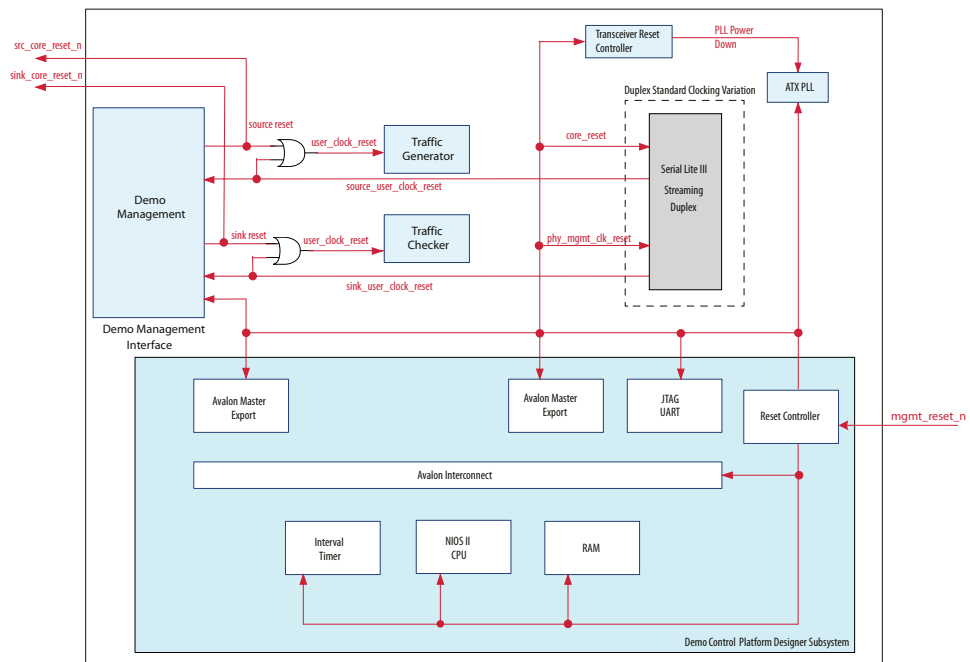
The following diagrams show the reset scheme implemented in the design examples.



**Figure 9. Reset Scheme for Intel Arria 10 Serial Lite III Streaming Simplex Core in Standard Clocking Mode**



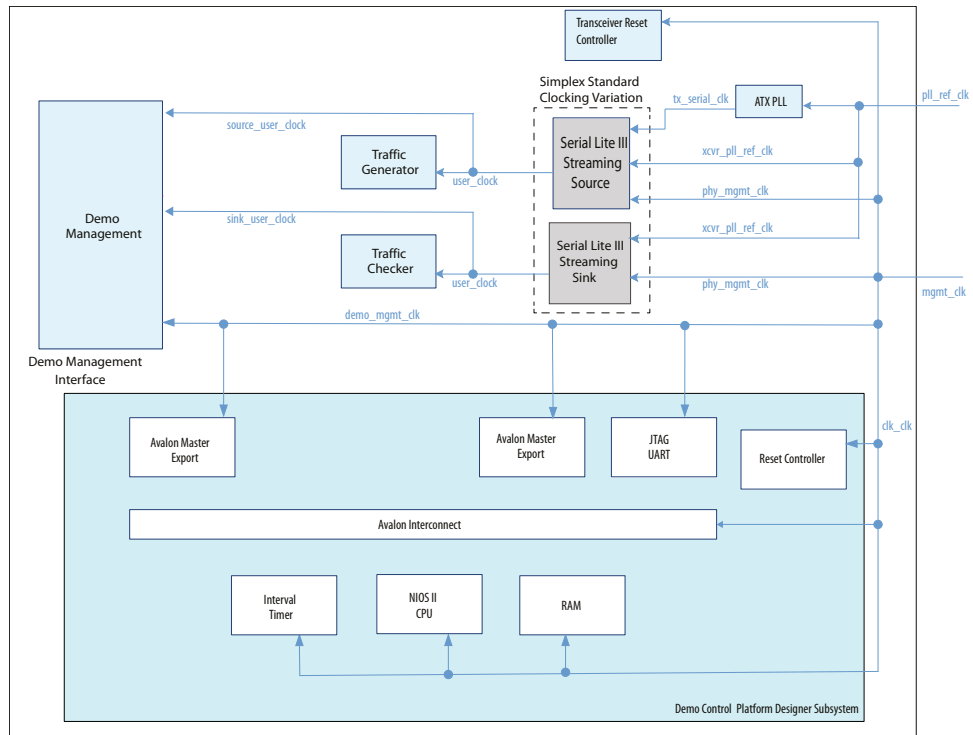
**Figure 10. Reset Scheme for Intel Arria 10 Serial Lite III Streaming Duplex Core in Standard Clocking Mode**



### 2.3.3. Clocking Scheme

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

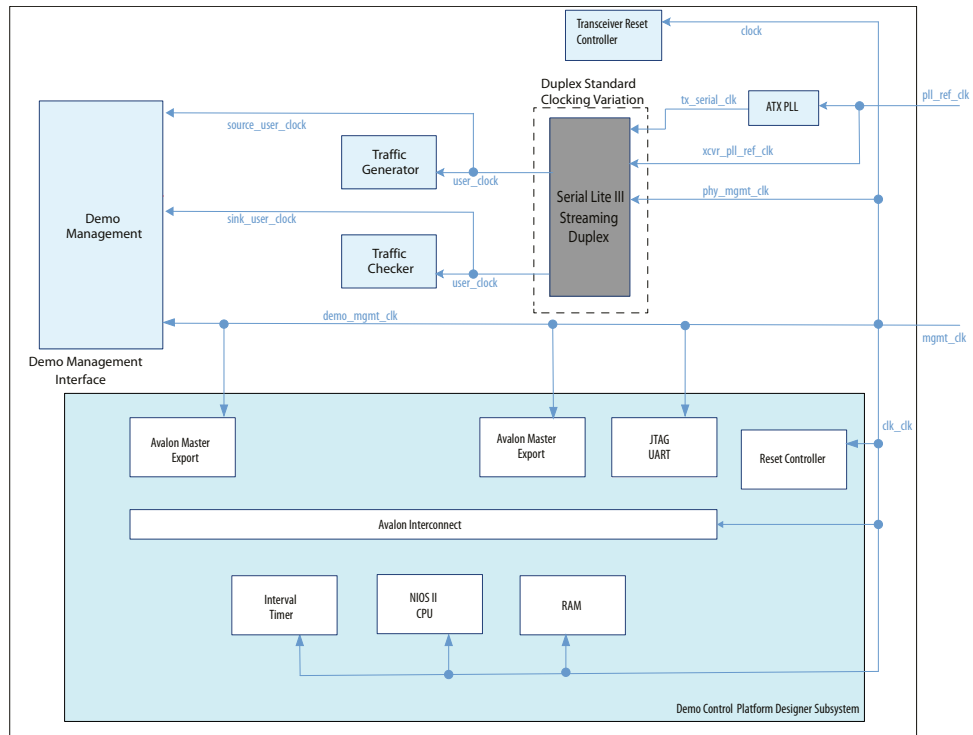
**Figure 11. Clocking Scheme for Intel Arria 10 Serial Lite III Streaming Simplex Core in Standard Clocking Mode**







**Figure 12. Clocking Scheme for Intel Arria 10 Serial Lite III Streaming Duplex Core in Standard Clocking Mode**



## 2.4. Simulation

The simulation test cases demonstrate continuous streaming of 2000 sample data for all lanes from traffic generator to the Serial Lite III Streaming source core and externally loopback to the sink core in standard clocking mode.

The simulation test case performs the following steps:

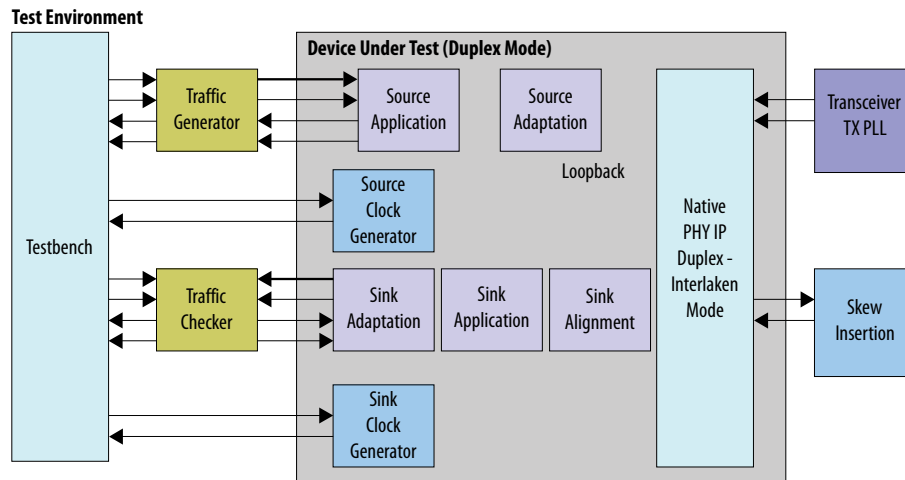
1. Initialize and configures Serial Lite III Streaming IP core, traffic generator and traffic checker.
2. Traffic generator generates data and starts data transmission.
3. Logs and display link up status and burst information.
4. Traffic checker verifies received data and stop transmission.
5. Testbench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

### 2.4.1. Testbench

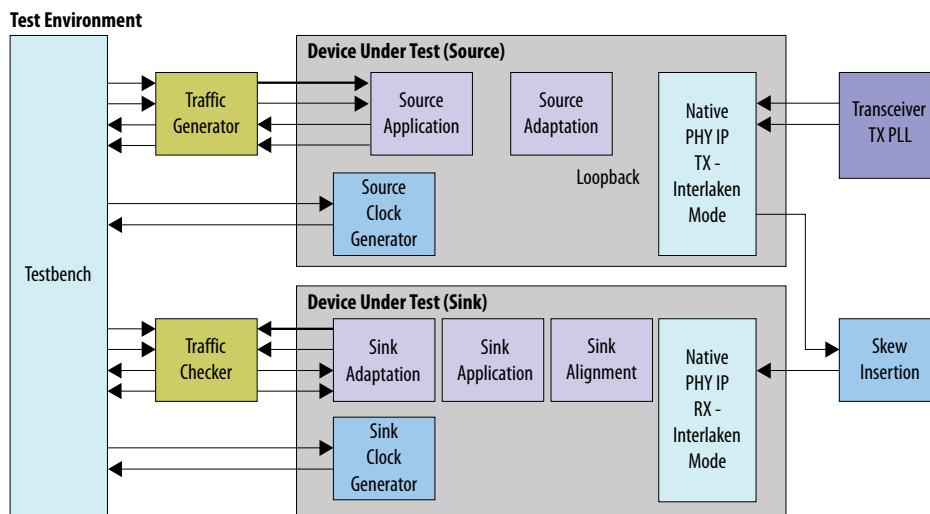
If your design targets Intel Arria 10 devices, the generated example testbench is dynamic and has the same configuration as the IP.

When you choose the sink or duplex direction, the parameter editor generates an external transceiver ATX PLL for use in the Intel Arria 10 testbench.

**Figure 13. Serial Lite III Streaming Example Testbench (Duplex) for Intel Arria 10 Devices**



**Figure 14. Serial Lite III Streaming Example Testbench (Simplex) for Intel Arria 10 Devices**



## 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:

- **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.

### 2.5.1. Design Setup

The design example targets the Intel Arria 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.

You must use correct pin constraints when using the core in simplex mode or when using more than one reconfiguration controller. The synthesized design typically includes a reconfiguration interface for at least three channels because three channels share an Avalon-MM slave interface, which connects to the Transceiver Reconfiguration Controller IP core. Conversely, you cannot connect three channels that share an Avalon-MM interface to different Transceiver Reconfiguration Controller IP cores or you will receive a Fitter error.

#### Related Information

[Altera Transceiver PHY IP Core User Guide](#)

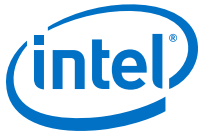
More information about the Interlaken PHY IP core.

### 2.5.2. Error Details

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

**Table 5. List of Error Details**

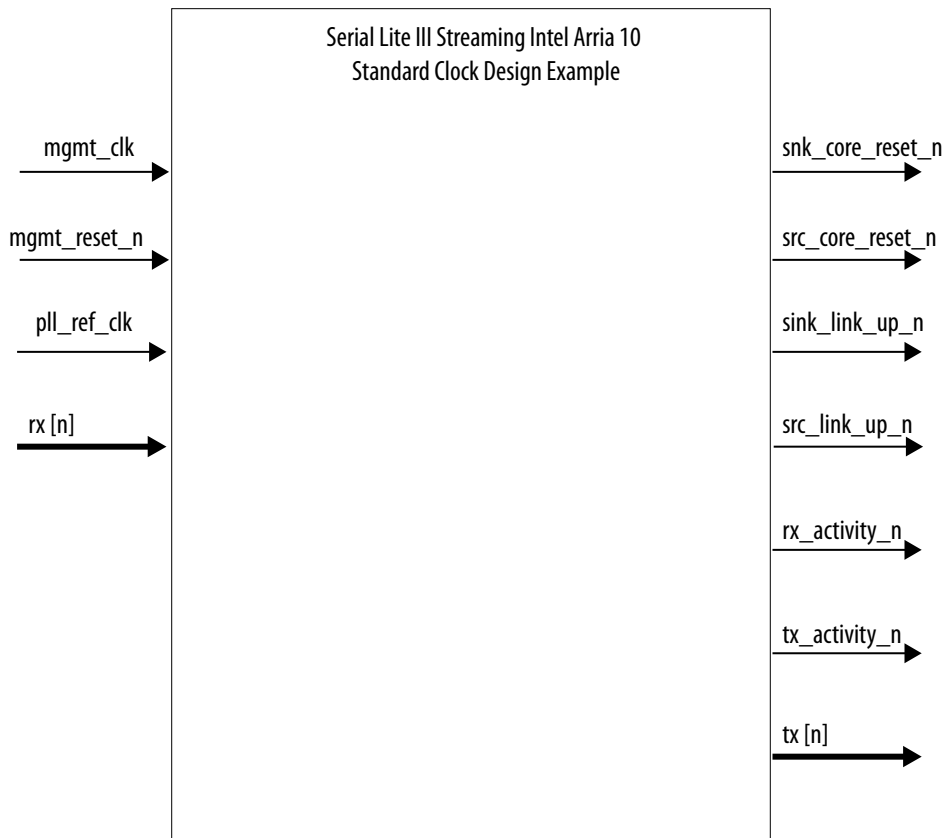
Error	Description
<b>Source Error:</b>	
Adaptation FIFO Overflow	To indicate source adaptation FIFO overflow error.
<b>Sink Errors:</b>	
Adaptation FIFO Overflow	To indicate sink adaptation FIFO overflow error.
Loss of Alignment During Normal Operation	To indicate loss of alignment error ( <code>error_rx[1]</code> ).
<i>continued...</i>	



Error	Description
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.

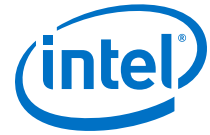
## 2.6. Signals

**Figure 15. Top-level Signals for Intel Arria 10 Serial Lite III Streaming Standard Clocking Mode Design Example**



**Table 6. Design Example Interface Signals**

Signal	Direction	Width	Description
Clock and Reset Signal			
<code>mgmt_clk</code>	Input	1	Input clock for: <ul style="list-style-type: none"> <li>Avalon-MM PHY management interface for Serial Lite III Streaming IP core</li> <li>Demo management module</li> <li>Demo control module</li> <li>Transceiver reset controller</li> </ul>
<i>continued...</i>			



Signal	Direction	Width	Description
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 asynchronous 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 <b>Number of Lanes</b> value	This vector carries the transmitted streaming data from the core. N represents the number of lanes.
tx[n]	Output	Based on <b>Number of Lanes</b> 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 Arria 10 Serial Lite III Streaming Advanced Clocking Mode

---

These design examples demonstrate the functionalities of data streaming using advanced clocking mode.

To generate the design examples, select the following preset(s):

- Advanced Clocking Mode 2x10G
- Advanced Clocking Mode 6x12.5G

By default, the design examples are generated as duplex core. To generate the design examples in simplex core, select **Simplex** for the **Direction** parameter.

### 3.1. Features

Features for Advanced Clocking Mode 2x10G design example includes:

- Support 2 lanes with 10Gpbs transceiver data rate
- Support simplex and duplex transmission modes
- Traffic checker for data verification and lane de-skew verification
- Support CRC error injection using Nios II processor

Features for Advanced Clocking Mode 6x12.5G design example includes:

- Support 6 lanes with 12.5Gpbs transceiver data rate
- Support simplex and 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 Arria 10 GX Transceiver Signal Integrity Development Kit for hardware testing

### 3.3. Functional Description

The design examples consist of various components. The following block diagrams show the design components and the top-level connections of the design examples.

---

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

\*Other names and brands may be claimed as the property of others.



Figure 16. Design Example for Simplex Core in Advanced Clocking Mode

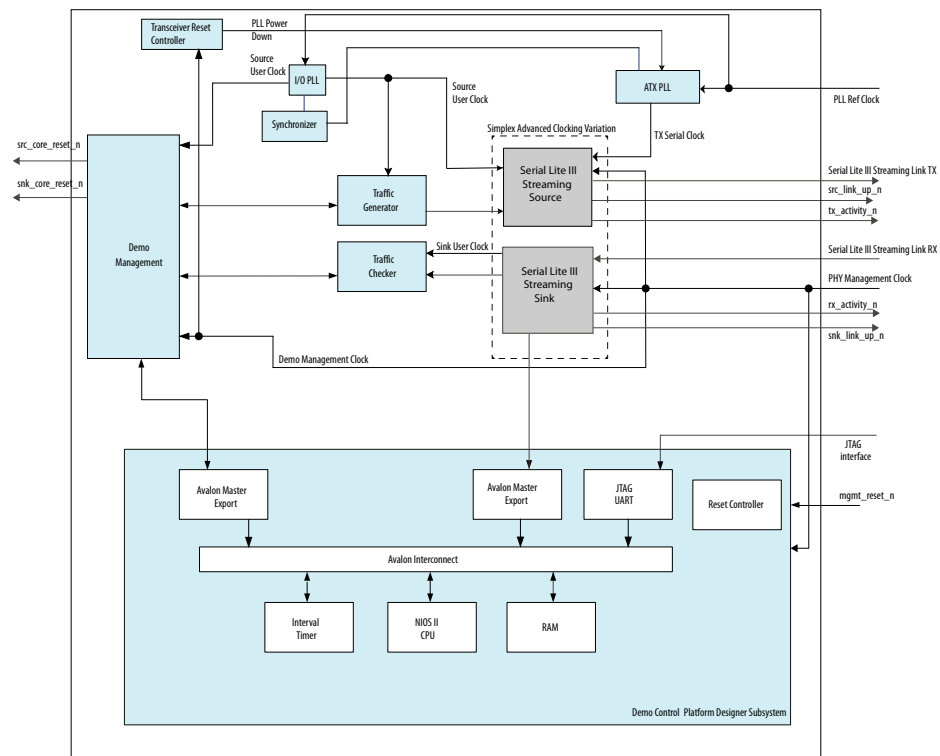
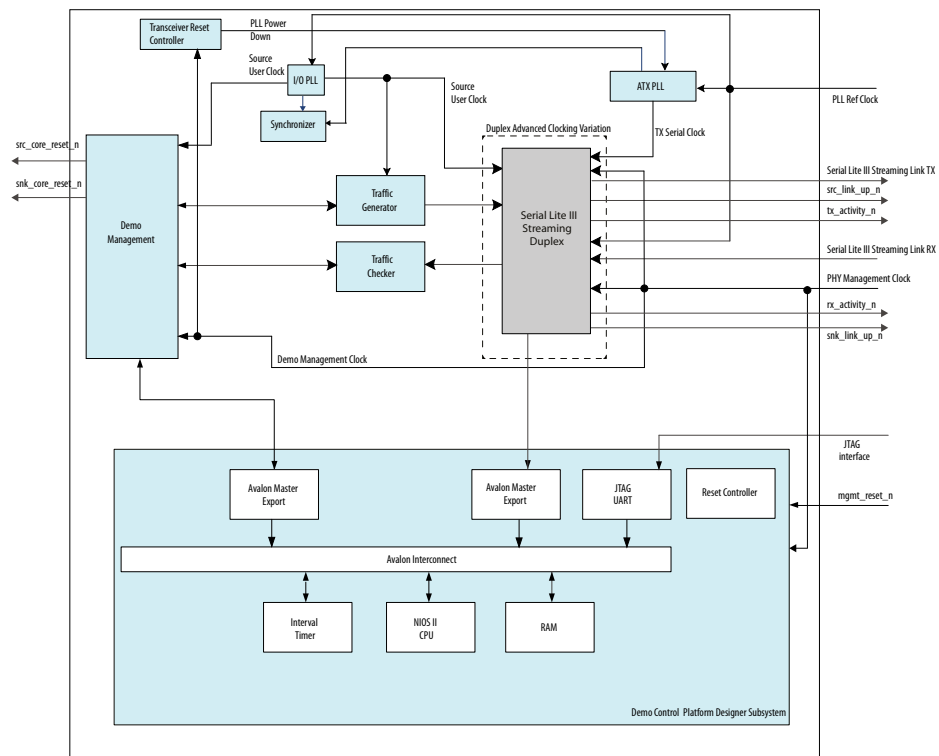


Figure 17. Design Example for Duplex Core in Advanced Clocking Mode



### 3.3.1. Design Example Components

The design example consists of following components:

- Serial Lite III Streaming IP core variation
- ATX PLL
- Source user clock—I/O PLL
- Traffic generator
- Traffic checker
- Demo control
- Demo management
- Nios II processor code

#### 3.3.1.1. Serial Lite III Streaming IP Core

The Serial Lite 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. Source User Clock - I/O PLL

The I/O PLL generates a user clock for sourcing data into the Serial Lite III Streaming IP core when configured in Advanced Clocking Mode.

### 3.3.1.3. ATX PLL

The ATX PLL generates transmit transceiver clock to the Serial Lite III Streaming IP core.

#### Related Information

[How to compensate the jitter of PLL cascading or non-dedicated clock path for Intel Arria 10 reference clock?](#)

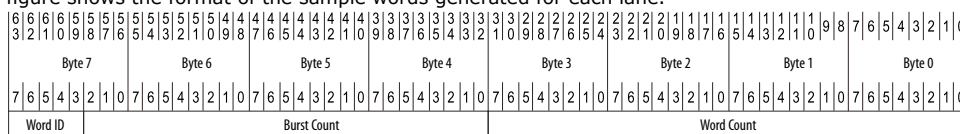
In previous Quartus Prime versions, reference clock from a cascaded PLL output, global clock, or core clock causes additional jitter in Intel Arria 10 Serial Lite III Streaming IP core design example. If you are migrating your design from earlier versions to Quartus Prime 17.1, refer to the KDB link.

### 3.3.1.4. 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 18. 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	58–52	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 <code>start_of_burst</code> 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.



### 3.3.1.5. 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.

### 3.3.1.6. 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 occurrence, Avalon-MM interface to access demo management and the Serial Lite III Streaming Intel FPGA IP PHY interface, a reset controller, a UART interface, and an Avalon Streaming (Avalon-ST) interface.

### 3.3.1.7. Demo Management

The demo management module controls the user modules interaction with the Serial Lite III Streaming IP core such as enable and disable traffic generator and traffic checker, enable CRC error insertion, and provide user clock reset for Serial Lite 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.

### 3.3.1.8. Nios II Processor Code

The Nios II processor controls the options exercised in the design example. The code also enables the configuration RAM (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.

The `demo_control.c` program Intel Stratix® 10 H-tile and L-tile devices uses the control registers to dynamically toggle the `rx_serial1pbken` port on the Transceiver PHY block to change the TX to RX loopback from internal to external.

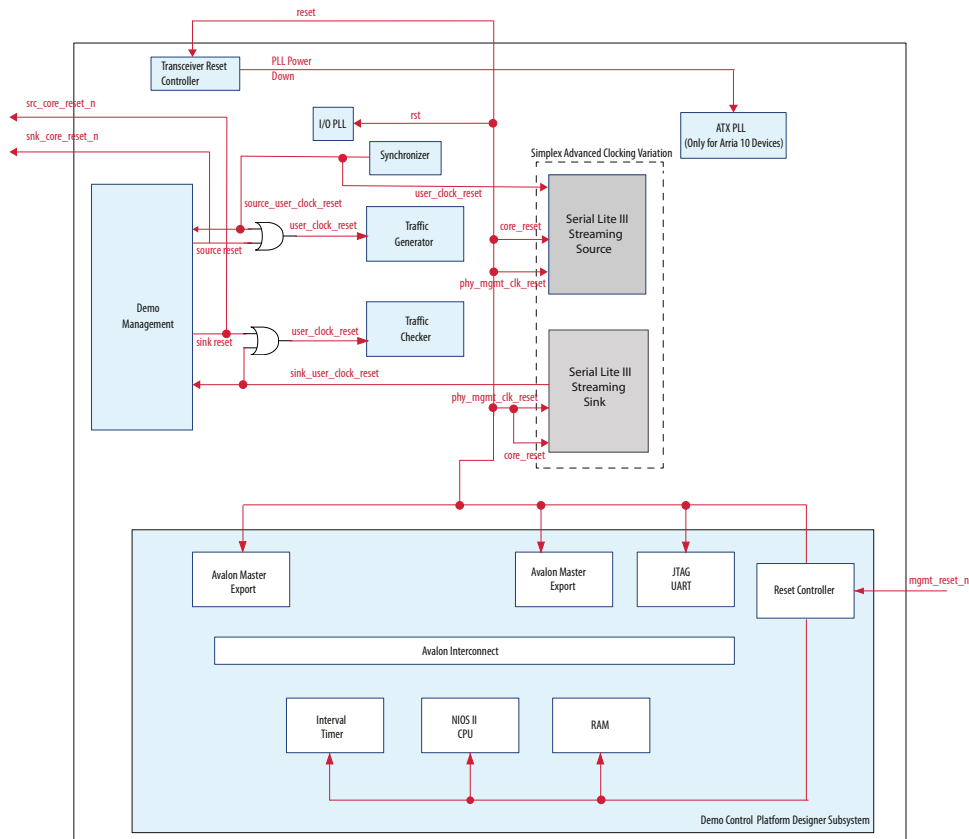


### 3.3.2. Reset Scheme

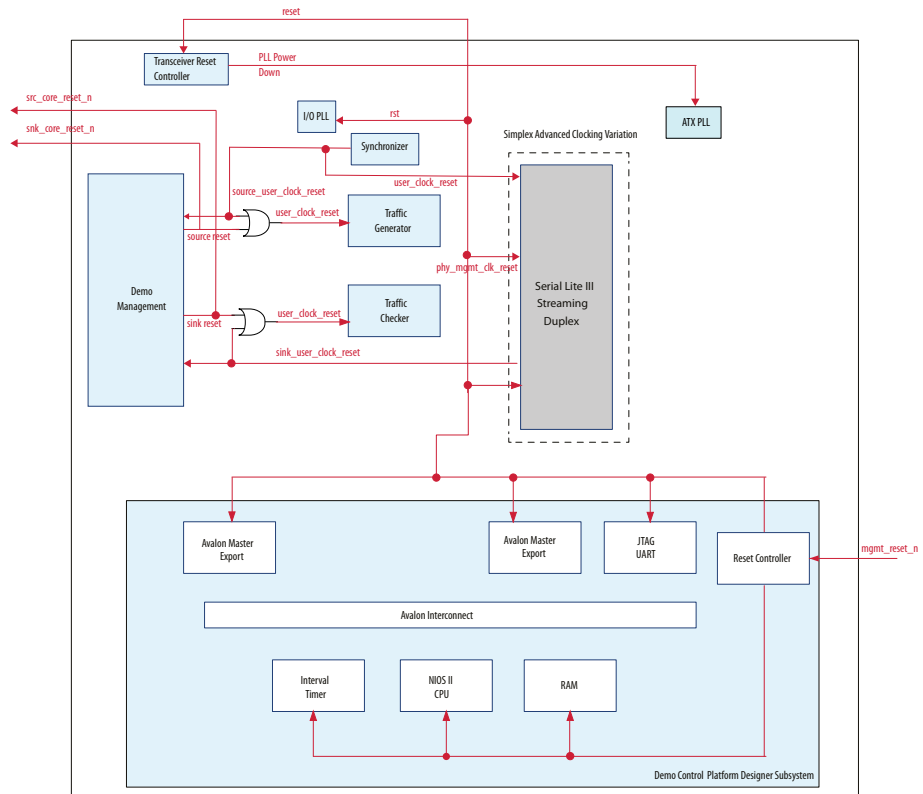
The `mgmt_reset_n` reset signal controls the overall reset structure for the design example. This is an asynchronous and active-low signal. Asserting this signal resets the demo control module and the Serial Lite III Streaming IP core. The traffic generator and traffic checker modules get reset through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

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

**Figure 19. Reset Scheme for Intel Arria 10 Serial Lite III Streaming Simplex Core in Advanced Clocking Mode**



**Figure 20. Reset Scheme for Intel Arria 10 Serial Lite III Streaming Duplex Core in Advanced Clocking Mode**

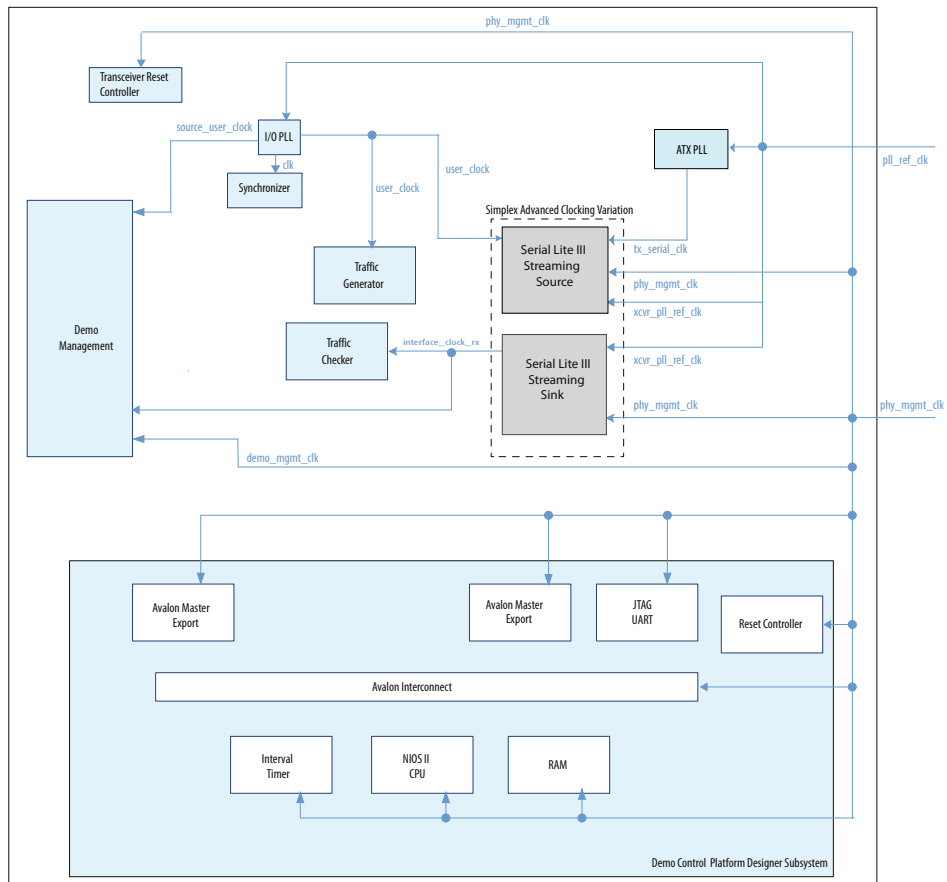


### 3.3.3. Clocking Scheme

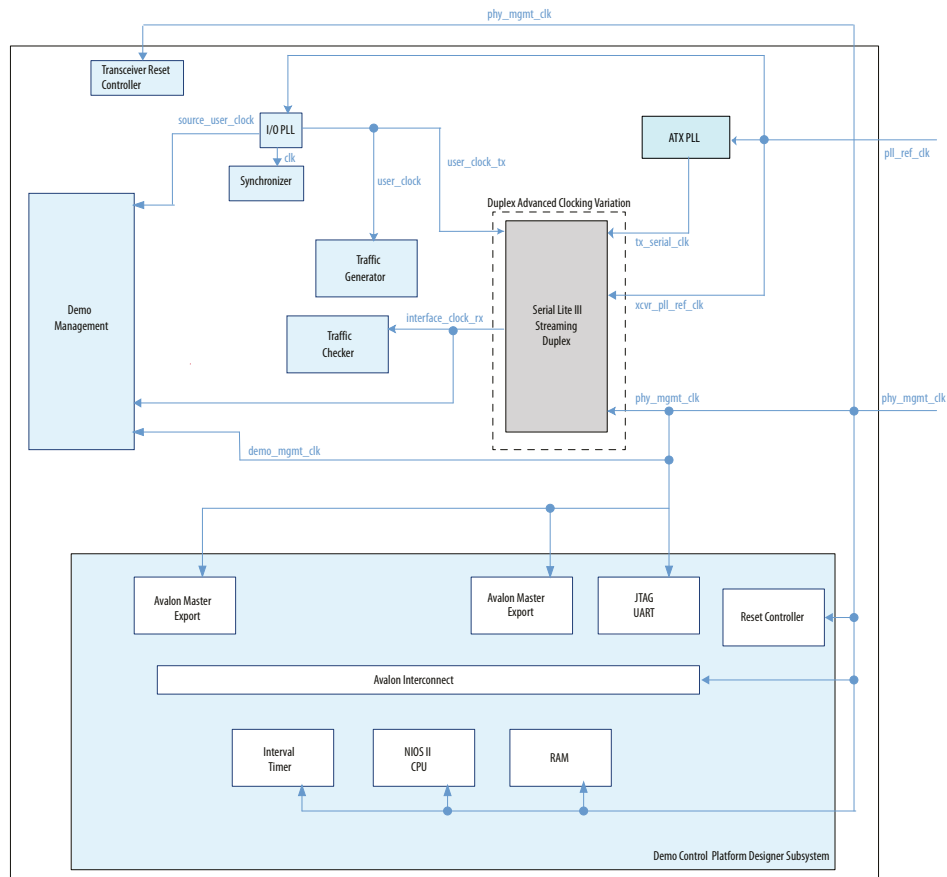
The following diagrams show the clocking scheme for the design examples.



Figure 21. Clocking Scheme for Intel Arria 10 Serial Lite III Streaming Simplex Core in Advanced Clocking Mode



**Figure 22. Clocking Scheme for Intel Arria 10 Serial Lite III Streaming Duplex Core in Advanced Clocking Mode**



### 3.4. Simulation

The simulation test cases demonstrate continuous streaming of 2000 sample data for all lanes from traffic generator to the Serial Lite 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 Serial Lite 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.

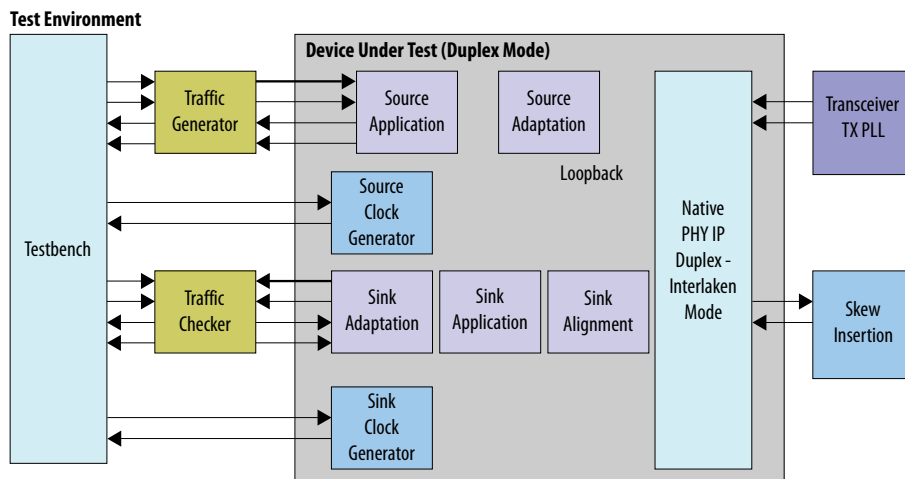


### 3.4.1. Testbench

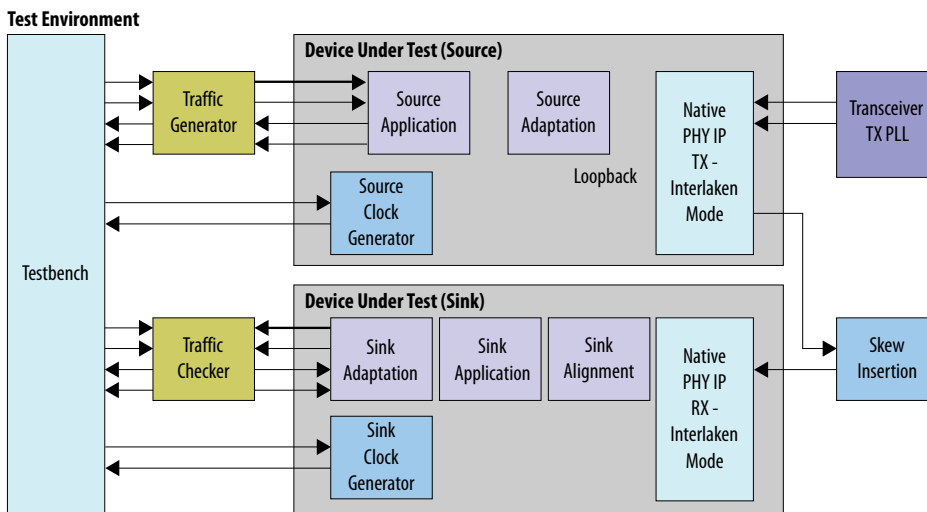
If your design targets Intel Arria 10 devices, the generated example testbench is dynamic and has the same configuration as the IP.

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

**Figure 23. Serial Lite III Streaming Example Testbench (Duplex) for Intel Arria 10 Devices**



**Figure 24. Serial Lite III Streaming Example Testbench (Simplex) for Intel Arria 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:

1. **Enable Data Generator/Checker**—Enables the traffic generator and start sending out data. This option enable data streaming in continuous mode.
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.

### 3.5.1. Design Setup

The design example targets the Intel Arria 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.

You must use correct pin constraints when using the core in simplex mode or when using more than one reconfiguration controller. The synthesized design typically includes a reconfiguration interface for at least three channels because three channels share an Avalon-MM slave interface, which connects to the Transceiver Reconfiguration Controller IP core. Conversely, you cannot connect three channels that share an Avalon-MM interface to different Transceiver Reconfiguration Controller IP cores or you will receive a Fitter error.

#### Related Information

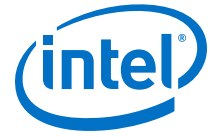
[Altera Transceiver PHY IP Core User Guide](#)

More information about the Interlaken PHY IP core.

### 3.5.2. Error Details

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



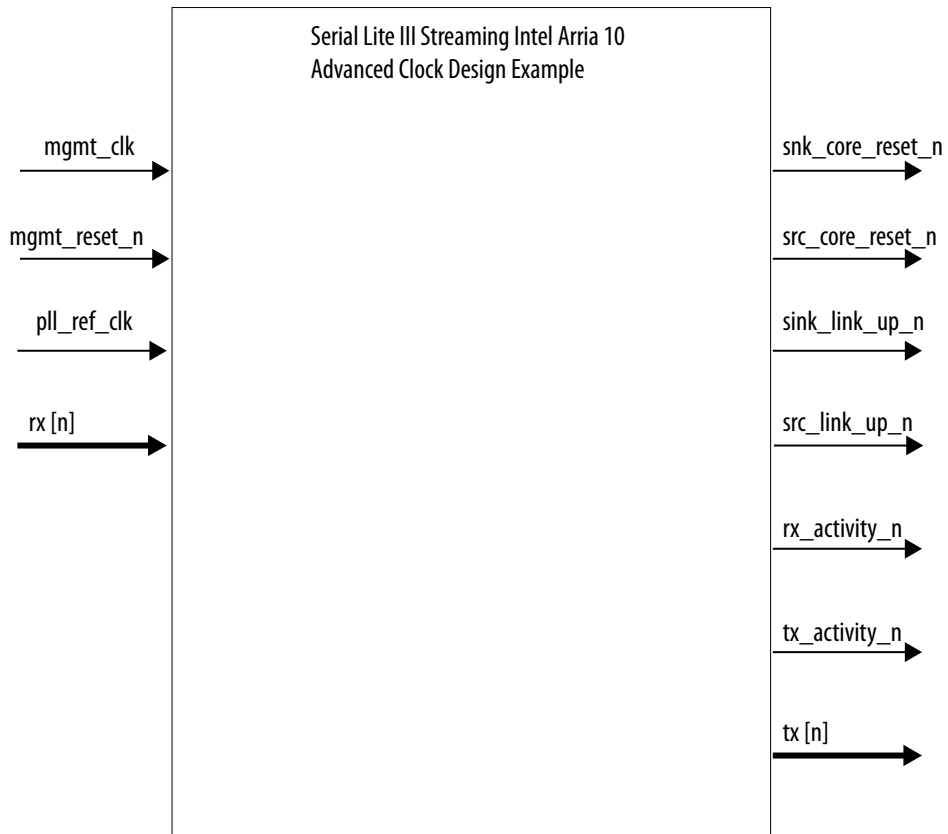


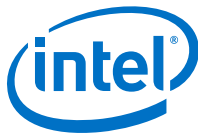
**Table 8. List of Error Details**

Error	Description
<b>Source Error:</b>	
Adaptation FIFO Overflow	To indicate source adaptation FIFO overflow error.
<b>Sink Errors:</b>	
Adaptation FIFO Overflow	To indicate sink adaptation FIFO overflow error.
Loss of Alignment During Normal Operation	To indicate loss of alignment error ( <code>error_rx[1]</code> ).
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 25. Top-level Signals for Intel Arria 10 Serial Lite III Streaming Advanced Clocking Mode Design Example**





**Table 9. Design Example Interface Signals**

Signal	Direction	Width	Description
<b>Clock and Reset Signal</b>			
mgmt_clk	Input	1	Input clock for: <ul style="list-style-type: none"> <li>Avalon-MM PHY management interface for Serial Lite III Streaming IP core</li> <li>Demo management module</li> <li>Demo control module</li> <li>Transceiver reset controller</li> </ul>
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 asynchronous 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.
<b>Data Signal</b>			
rx[n]	Input	Based on <b>Number of Lanes</b> value	This vector carries the transmitted streaming data from the core. <i>N</i> represents the number of lanes.
tx[n]	Output	Based on <b>Number of Lanes</b> value	This vector carries the transmitted streaming data to the core. <i>N</i> represents the number of lanes.
<b>Status Signal</b>			
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. Intel Arria 10 SerialLite III Streaming IP Core Design Example User Guide Document Archives

---

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

IP Core Version	User Guide
17.0	<a href="#">SerialLite III Streaming IP Core Design Example User Guide Document Archives</a>
16.1	<a href="#">SerialLite III Streaming IP Core Design Example User Guide Document Archives</a>
16.0	<a href="#">SerialLite III Streaming IP Core Design Example User Guide Document Archives</a>

## 5. Document Revision History for Serial Lite III Streaming IP Core Design Example User Guide for Intel Arria 10

Document Version	Intel Quartus Prime Version	Changes
2019.05.13	17.1	<ul style="list-style-type: none"> <li>Clarified that hardware design examples are only supported in designs that generated from presets in <i>Procedure</i> and <i>Presets</i> topics. .</li> <li>Rebranded <i>SerialLite III Streaming IP core</i> to <i>Serial Lite III Streaming IP core</i>.</li> </ul>

Date	Version	Changes
November 2017	2017.11.06	<ul style="list-style-type: none"> <li>Renamed the document as <i>SerialLite III Streaming IP Core Design Example User Guide for Intel Arria 10 Devices</i>.</li> <li>Added Knowledge Base link <a href="#">How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock?</a> in <i>ATX PLL</i> on page 12 chapter.</li> <li>Updated for latest branding standards.</li> </ul>
May 2017	2017.05.08	<ul style="list-style-type: none"> <li>Rebranded as Intel.</li> <li>Restructured document to migrate Stratix 10 device support to a new design example user guide.</li> <li>Updated the Compiling and Testing the Design topic.</li> <li>Updated the Directory and File Description for Design Example Folder table.</li> <li>Updated the Testbench sub-topic for Detailed Description for Arria 10 SerialLite III Streaming Standard Clocking Mode chapter.</li> <li>Updated the following figures: <ul style="list-style-type: none"> <li>Design Example for Simplex Core in Advanced Clocking Mode</li> <li>Design Example for Duplex Core in Advanced Clocking Mode</li> <li>Reset Scheme for Arria 10 SerialLite III Streaming Simplex Core in Advanced Clocking Mode</li> <li>Reset Scheme for Arria 10 SerialLite III Streaming Duplex Core in Advanced Clocking Mode</li> <li>Clocking Scheme for Arria 10 SerialLite III Streaming Simplex Core in Advanced Clocking Mode</li> <li>Clocking Scheme for Arria 10 SerialLite III Streaming Duplex Core in Advanced Clocking Mode</li> </ul> </li> <li>Updated the Testbench sub-topic for Detailed Description for Arria 10 SerialLite III Streaming Advanced Clocking Mode chapter.</li> <li>Minor typographical corrections.</li> </ul>
December 2016	2016.12.09	<ul style="list-style-type: none"> <li>Added 6x17.4Gbps presets for Intel Stratix 10 Standard and Advanced Mode design examples.</li> <li>Added error details list.</li> </ul>
October 2016	2016.10.28	<ul style="list-style-type: none"> <li>Added Intel Stratix 10 device support.</li> <li>Restructured document.</li> </ul>
May 2016	2016.05.02	Initial release.

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

\*Other names and brands may be claimed as the property of others.