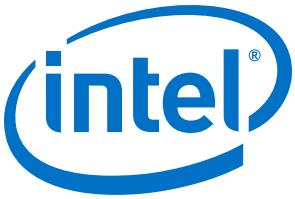


SDI II Intel® Arria 10 FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **18.1**

Subscribe
Send Feedback

UG-20076 | 2018.11.20
Latest document on the web: **PDF** | **HTML**

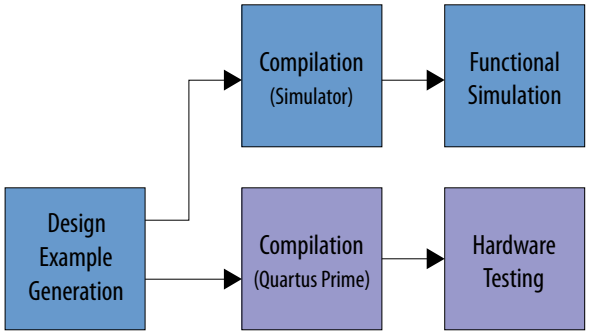
Contents

1. SDI II Intel® FPGA IP Design Example Quick Start Guide for Intel® Arria® 10 Devices

The Serial Digital Interface (SDI) II Intel FPGA IP design examples for Intel® Arria® 10 devices feature a simulation testbench and a hardware design that supports compilation and hardware testing.

When you generate a design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware.

Figure 1. Development Steps



Related Information

- SDI II Intel FPGA IP User Guide
- SDI II Intel Arria 10 FPGA IP Design Example User Guide Archives on page 46
  Provides a list of user guides for previous versions of the SDI II Intel FPGA IP design examples.

1.1. Directory Structure

The directories contain the generated files for the design examples.

Figure 2. Directory Structure for the Design Examples

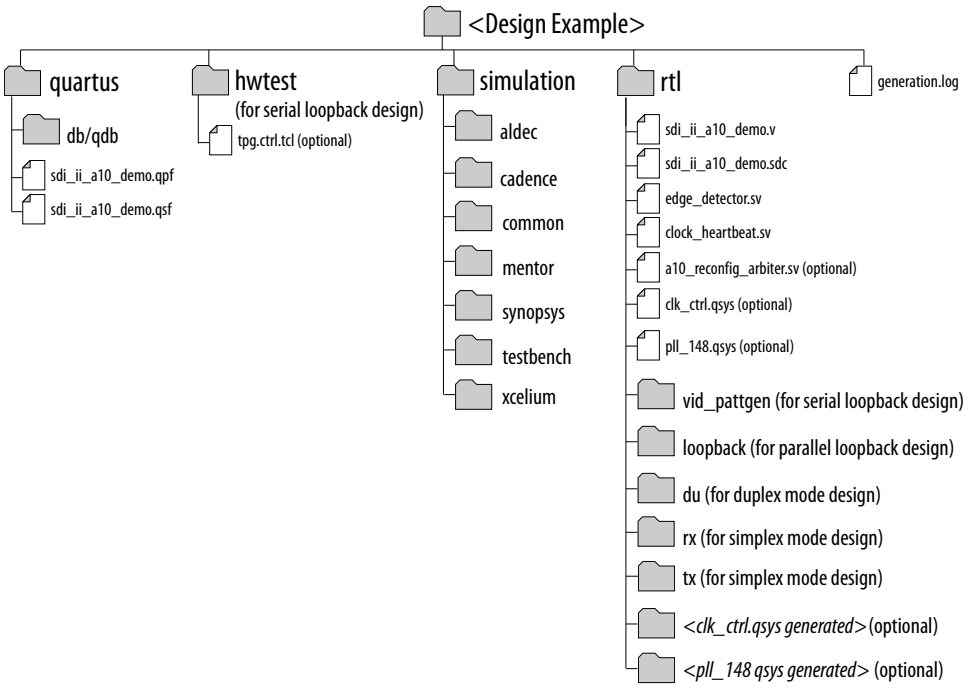


Table 1. Other Generated Files in RTL Folder

Folders	Files
vid_pattgen	/sdi_ii_colorbar_gen.v
	/sdi_ii_ed_vid_pattgen.v
	/sdi_ii_makeframe.v
	/sdi_ii_patho_gen.v
	/jtag.sdc
	/pattgen_ctrl.qsys
	<qsys generated folder>
loopback	/loopback_top.v
	/fifo/sdi_ii_ed_loopback.sdc
	/fifo/sdi_ii_ed_loopback.v
	/pfd/clock_crossing.v (optional)
	/pfd/pfd.sdc (optional)
	/pfd/pfd.v (optional)
	/reclock/sdi_reclock.v (optional)
	/reclock/pid_controller.v (optional)
	/reclock/rcfg_pll_frac.v

continued...

Folders	Files
du	/du_top.v
	/sdi_ii_rx_rcfg_a10.sv (optional)
	/rcfg_sdi_cdr.sv (optional)
	/rcfg_pll_sw.sv (optional)
	/rcfg_refclk_sw.sv (optional)
	/sdi_ii_tx_rcfg_a10.sv (optional)
	/sdi_du_sys.qsys
	• /sdi_rx_phy.qsys (Intel Quartus® Prime Standard Edition) • /sdi_rx_phy.ip (Intel Quartus Prime Pro Edition)
	• /tx_pll.qsys (Intel Quartus Prime Standard Edition) • /tx_pll.ip (Intel Quartus Prime Pro Edition)
	• /tx_pll_alt.qsys (Intel Quartus Prime Standard Edition) • /tx_pll_alt.ip (Intel Quartus Prime Pro Edition) (optional)
	<qsys generated folder>
rx	/rx_top.v
	/sdi_ii_rx_rcfg_a10.sv (optional)
	/rcfg_sdi_cdr.sv (optional)
	/sdi_rx_sys.qsys
	<qsys generated folder>
tx	/tx_top.v
	/rcfg_pll_sw.sv (optional)
	/rcfg_refclk_sw.sv (optional)
	/sdi_ii_tx_rcfg_a10.sv (optional)
	/sdi_tx_sys.qsys
	• /tx_pll.qsys (Intel Quartus Prime Standard Edition) • /tx_pll.ip (Intel Quartus Prime Pro Edition)
	• /tx_pll_alt.qsys (Intel Quartus Prime Standard Edition) • /tx_pll_alt.ip (Intel Quartus Prime Pro Edition) (optional)
	<qsys generated folder>

Table 2. Other Generated Files in Simulation Folder

Folders	Files
aldec	/aldec.do
	/rivierapro_setup.tcl
cadence	/cds.lib

continued...

Folders	Files
	/hdl.var
	/ncsim.sh
	/ncsim_setup.sh
	<cds_libs folder>
common	/modelsim_files.tcl
	/ncsim_files.tcl
	/riviera_files.tcl
	/vcs_files.tcl
	/vcsmx_files.tcl
	/xcelium_files.tcl
mentor	/mentor.do
	/msim_setup.tcl
synopsys	/vcs/filelist.f
	/vcs/vcs_setup.sh
	/vcs/vcs_sim.sh
	/vcsmx/synopsys_sim_setup
	/vcsmx/vcsmx_setup.sh
	/vcsmx/vcsmx_sim.sh
testbench	tb_top.v
	rx_checker/sdi_ii_tb_rx_checker.v
	rx_checker/tb_data_compare.v
	rx_checker/tb_dual_link_sync.v
	rx_checker/tb_fifo_line_test.v
	rx_checker/tb_frame_locked_test.sv
	rx_checker/tb_ln_check.v
	rx_checker/tb_rxsample_test.v
	rx_checker/tb_trs_locked_test.sv
	rx_checker/tb_txpll_test.sv
	rx_checker/tb_vpid_check.v
	tb_control/sdi_ii_tb_control.v
	tb_control/tb_clk_rst.v
	tb_control/tb_data_delay.v
	tb_control/tb_serial_delay.sv
	tb_control/tb_tasks.v

continued...

Folders	Files
	tb_checker/sdi_ii_tb_tx_checker.v
	tb_checker/tb_serial_check_counter.v
	tb_checker/tb_serial_descrambler.v
	tb_checker/tb_tx_clkout_check.v
	vid_pattgen/sdi_ii_colorbar_gen.v
	vid_pattgen/sdi_ii_ed_vid_pattgen.v
	vid_pattgen/sdi_ii_makeframe.v
	vid_pattgen/sdi_ii_patho_gen.v
xcelium	/cds.lib
	/hdl.var
	/xcelium_setup.sh
	/xcelium_sim.sh
	<cds_libs folder>

1.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design examples.

Hardware

- Intel Arria 10 GX FPGA Development Kit
- SDI Signal Generator
- SDI Signal Analyzer
- SubMiniature version B (SMB) to BNC cables for single-rate and triple-rate designs, or BNC to BNC cables for multi-rate designs
- VIDIO* FMC Development Module VIDIO-12G-A (Nextera 12G SDI FMC daughter card) for multi-rate designs

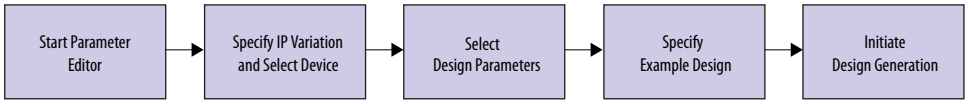
Software

- Intel Quartus Prime (for hardware testing)
- ModelSim* - Intel FPGA Edition, ModelSim - Intel FPGA Starter Edition, NCSim, Riviera-PRO*, VCS* (Verilog HDL only)/VCS MX, or Xcelium* Parallel simulator

1.3. Generating the Design

Configure the SDI II Intel FPGA IP parameter editor in the Intel Quartus Prime software to generate the design examples.

Figure 3. Generating the Design Flow



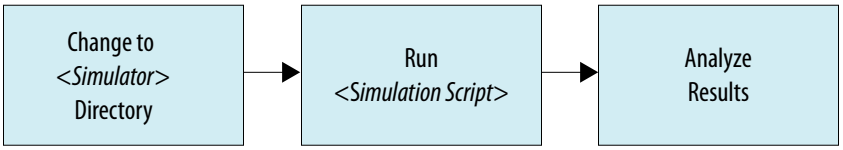
1. Create a project targeting the Intel Arria 10 device family and select the desired device.
2. In the IP Catalog, locate and double-click **SDI II Intel FPGA IP**. The **New IP Variant** or **New IP Variation** window appears.
3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>*.ip /*<your_ip>*.qsys.
4. Click **OK**. The parameter editor appears.
5. On the **IP** tab, select your desired IP settings. The generated design example is based on your settings.
6. On the **Design Example** tab, select **Simulation** to generate the testbench, and select **Synthesis** to generate the hardware design example.

   You must select at least one of these options to generate the design example files.
7. For **Generate File Format**, select **Verilog** or **VHDL**.
8. For **Target Development Kit**, select the relevant FPGA development kit. You may change the target device using the **Change Target Device** parameter if your board revision does not match the grade of the default targeted device.
9. Click **Generate Example Design**.

1.4. Simulating the Design

The SDI II Intel FPGA IP design example testbench simulates one channel serial loopback design with TX instance connected to an internal video pattern generator. The serial output from the TX instance connects to the RX instance in the testbench. The testbench also includes checkers and control mechanisms.

Figure 4. Design Simulation Flow



1. Navigate to the simulation folder of your choice.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator.
3. Analyze the results.

Table 3. Steps to Run Simulation

Simulator	Working Directory	Instructions
Riviera-PRO	/simulation/aldec	In the GUI, type: `do aldec.do`
NCSim	/simulation/cadence	In the command line, type: `source ncsim.sh`
Xcelium	/simulation/xcelium	In the command line, type: `source xcelium_sim.sh`

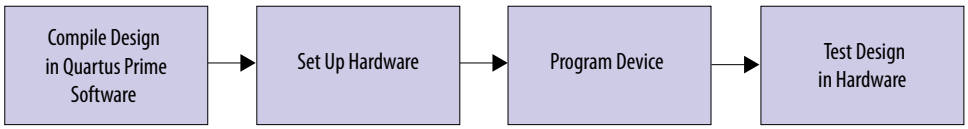
continued...

Simulator	Working Directory	Instructions
ModelSim	/simulation/mentor	In the GUI, type: `do mentor.do`
VCS	/simulation/synopsys/vcs	In the command line, type: `source vcs_sim.sh`
VCS MX	/simulation/synopsys/vcsmx	In the command line, type: `source vcsmx_sim.sh`

A successful simulation ends with the following message:

```
#### TRANSMIT TEST COMPLETED SUCCESSFULLY! ####
#
#### Channel 1: RECEIVE TEST COMPLETED SUCCESSFULLY! ####
```

1.5. Compiling and Testing the Design



To compile and run a demonstration test on the hardware design example, follow these steps:

1. Ensure that the hardware design example generation is complete.
2. Open `quartus/sdi_ii_a10_demo.qpf`.
3. Click **Processing ➤ Start Compilation**.
4. If you set the **Rx core clock (rx_coreclk) Frequency** parameter to 297.0/296.70 MHz, set the frequency for CLK1 in the **Si5338 (U14)** tab of the Clock Control GUI to **297 MHz**.
5. If you enable the **Dynamic Tx clock switching** feature in the **Design Example** parameter editor, set the frequency for CLK2 or CLK3 in the **Si5338 (U14)** tab of the Clock Control GUI.
   - For HD/3G-SDI single-rate and triple-rate designs, set CLK3 to **148.3516 MHz**.
   - For multi-rate designs, set CLK2 to **296.7033 MHz**.
6. After successful compilation, the Intel Quartus Prime software generates a `.sof` file in your specified directory.
7. Configure the selected Intel Arria 10 device on the development board using the generated `.sof` file (**Tools ➤ Programmer**).
8. For serial loopback designs, open the System Console to control the internal video pattern generator. Click **Tools ➤ System Debugging Tools ➤ System Console**.

   Note: Close the Clock Control GUI and the Programmer window before you open the System Console.
9. After the initialization, type `source ../hwtest/tpg_ctrl.tcl` in the System Console to open the pattern generator control user interface. Select your desired video format.

Related Information

Setting Up Environment Variables
Provides information about setting up the Clock Control application.

1.5.1. Connection and Settings Guidelines

Before programing with the `.sof` file, ensure that the connections and settings are correct.

Connections and Settings for HD/3G-SDI Single Rate and Triple Rate Designs

- For parallel loopback design, the on-board SMB RX connector (J20) connects to an external video source and the on-board SMB TX connector (J21) connects to a video analyzer.
- For serial loopback design, the on-board SMB TX connector (J21) connects to an on-board SMB RX connector (J20) or a video analyzer.
- Ensure all switches on the development board are in default position.
- The SDI video analyzer displays the video generated from the source.

Note: For parallel loopback designs, you may need to switch the Si516_FS (SW6.3) at the back of the board if you are switching between fractional frame rate and integer frame rate video format.

Figure 5. Switch Settings on the Arria 10 Development Board

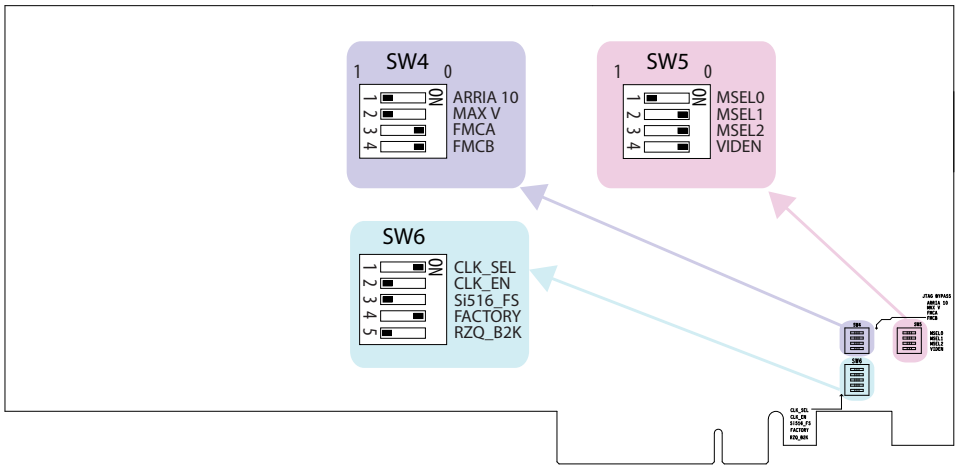


Table 4. SW6 DIP Switch Default Settings (Bottom of the Board)

Switch	Board Label	Description
1	CLK_SEL	• ON for 100 MHz on-board clock oscillator selection (Default position) • OFF for SMA input clock selection
2	CLK_EN	OFF for setting CLK_ENABLE high to the MAX V

continued...

Switch	Board Label	Description
3	SI516_FS	• ON for setting the SDI REFCLK frequency to 148.35 MHz • OFF for setting the SDI REFCLK frequency to 148.5 MHz (Default position)
4	FACTORY	• ON to load factory from flash (Default position) • OFF to load user hardware from flash
5	RZQ_B2K	• ON for setting RZQ resistor of Bank 2K to 99.17 ohm • OFF for setting RZQ resistor of Bank 2K to 240 ohm (Default position)

Connections and Settings for Multi Rate Design

- A VIDIO FMC Development Module VIDIO-12G-A (Nextera 12G SDI FMC daughter card) connects to the FMC Port B on the development board.
- For parallel loopback design, the BNC RX connector (J1/12G In) connects to an external video source and the TX connector (J2/12G Out) connects to a video analyzer.
- For serial loopback design, the BNC TX connector (J2/12G Out) connects to the BNC RX connector (J1/12G In) or a video analyzer.
- Ensure all switches on the development board are in default position.
- The SDI video analyzer displays the video generated from the source.

Note: Change the jumper (J8) position before switching between fractional frame rate and integer frame rate video formats. Press the push button (PB0) to trigger a device (LMK03328) power cycling through the PDN pin every time you change the jumper (J8) position.

Figure 6. Jumper Settings on Nextera 12G-SDI FMC Daughter Card

Refer to these settings to change the jumper (J8) position.

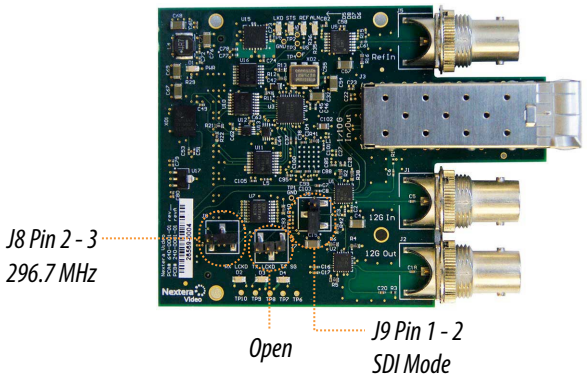


Table 5. Jumper Settings

Jumper Block	Description
J7	Programming header
J8	To switch the generated clock frequency for the TX channel:

continued...

Jumper Block	Description
	• Pin 1–2 = 297 MHz • Pin 2–3 = 297/1.001 MHz
J9	To select SDI or IP mode: • Pin 1–2 = SDI mode • Pin 2–3 = IP mode

Related Information

Intel Arria 10 FPGA Development Kit User Guide

1.5.2. Design Considerations

You need to consider certain issues when instantiating the SDI II Intel FPGA IP design examples.

- For designs using simplex receivers:
  - The simplex receiver channel may not calibrate correctly if its corresponding unused transmitter is preserved. Simplex receiver channels with the corresponding transmitters not preserved calibrate correctly. To overcome this issue, make the following changes in the QSF file.

    Remove the global preservation QSF assignment

    ```
    set_global_assignment –name PRESERVE_UNUSED_XCVR_CHANNEL ON
    ```

    Add per-pin preservation QSF assignment

    ```
    set_instance_assignment –name PRESERVE_UNUSED_XCVR_CHANNEL ON –to <pin name>
    ```

- Serial loopback designs:
  - The serial loopback design is mainly for image and TX clock switching demonstrations only. To get a more accurate jitter performance with the daughter card components, use the parallel loopback design and connect it to a clean video source.
  - To allow segmented frame video format (1080sF30, 1080sF25) and interlaced video format (1080i60, 1080i50) to be correctly differentiated in the external analyzer, Payload ID must be inserted in the serial loopback design.
  - The Omnitek Ultra 4K Analyzer (software version 2.1) may not detect 12G-SDI 2160p59.94 in the serial loopback design. If you encounter such problem, upgrade the Omnitek Ultra 4K analyzer to a later version.

Related Information

High BER Due to Calibration Error in Simplex Receiver Designs
Provides more information about the calibration error when using simplex receivers.

1.6. SDI II Intel FPGA Design Example Parameters

Table 6. SDI II Intel FPGA Design Example Parameters for Intel Arria 10 Devices

Parameter	Value	Description
Available Design Example		
Select Design	Parallel loopback with external VCXO Parallel loopback without external VCXO Serial loopback	Select the design example to be generated. • Parallel loopback with external VCXO: Parallel loopback design with an external VCXO to synchronize the clock between RX and TX. • Parallel loopback without external VCXO: Parallel loopback design without an external VCXO. *Note:* This option is not available for multi-rate designs and older Intel Arria 10 ES devices. • Serial loopback: A serial loopback design to enables a simple demonstration when you do not have a video source available and to highlight the **Dynamic Tx clock switching** feature. The IP core generates an internal video pattern generator along with the TX to be transmitted to RX.
Design Example Options		
Tx PLL type	CMU fPLL	Select the transceiver PLL type. • CMU PLL only supports data rates up to 3G-SDI. • fPLL supports all data rates up to 12G-SDI. *Note:* Only fPLL is available for multi-rate designs or when you generate a parallel loopback design without an external VCXO.
Dynamic Tx clock switching	Off Tx PLL switching Tx PLL reference clock switching	• Off: Disable dynamic switching. • Tx PLL switching: Instantiates two PLLs, each with its own reference input clock. • Tx PLL reference clock switching: Instantiates one PLL with two reference input clocks. Turn on this option to allow dynamic switching between 1 and 1/1.001 data rates. This option is only available when you select **Serial loopback**.
Design Example Files		
Simulation	On Off	Turn on this option to generate the necessary files for the simulation testbench.
Synthesis	On Off	Turn on this option to generate the necessary files for Intel Quartus Prime compilation and hardware demonstration.
Generated HDL Format		
Generate File Format	Verilog VHDL	Select your preferred HDL format for the generated design example fileset. *Note:* This option only determines the format for the generated top level IP files. All other files (e.g. example testbenches and top level files for hardware demonstration) are in Verilog HDL format.
Target Development Kit		
Select Board	No Development Kit Intel Arria 10 FPGA Development Kit Custom Development Kit	Select the board for the targeted design example.

Target Development Kit		
		• No Development Kit: This option excludes all hardware aspects for the design example. The IP core sets all pin assignments to virtual pins. • Intel Arria 10 FPGA Development Kit: This option automatically selects the project's target device to match the device on this development kit. You may change the target device using the **Change Target Device** parameter if your board revision has a different device variant. The IP core sets all pin assignments according to the development kit. *Note:* This option is not available if you select **Multi rate (up to 12G-SDI)** with **Bidirectional** mode. The Nextera daughter card pins are only compatible with simplex mode. • Custom Development Kit: This option allows the design example to be tested on a third party development kit with an Intel FPGA. You may need to set the pin assignments on your own.

Target Device		
Change Target Device	On Off	Turn on this option and select the preferred device variant for the development kit.

2. SDI II Intel FPGA Design Example Detailed Description

The SDI II Intel FPGA IP core includes these design examples for Intel Arria 10 devices.

- Parallel loopback with external VCXO
- Parallel loopback without external VCXO
- Serial loopback

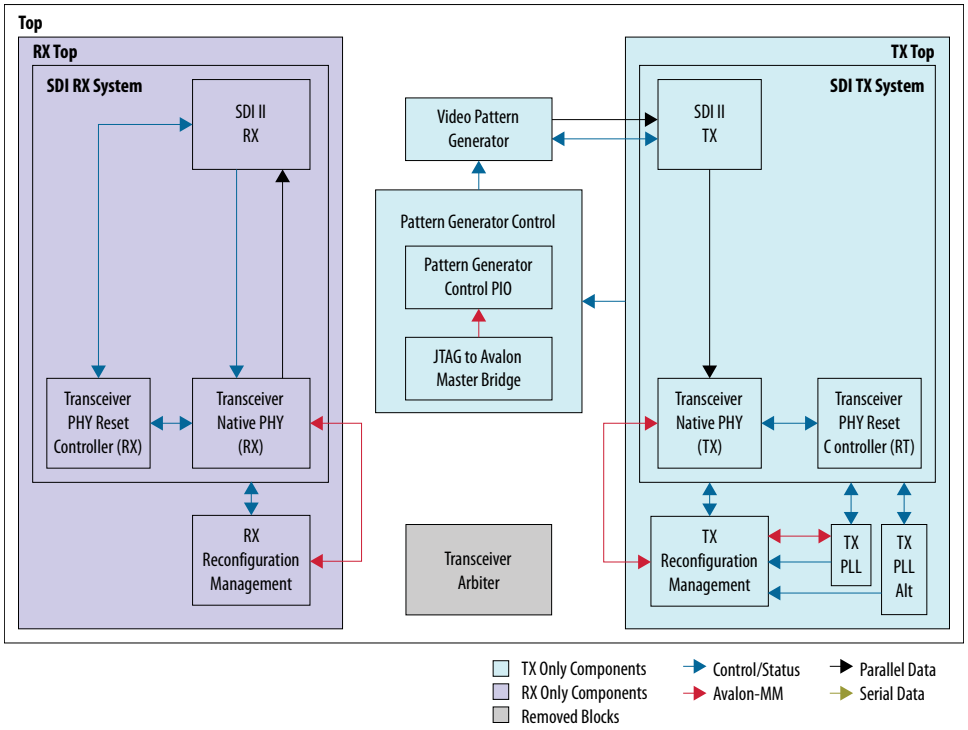
Features

- For HD/3G-SDI single rate and triple rate designs, you can choose either CMU or fPLL as the TX PLL.
- All designs use LED status for early debugging stage.
- The simplex serial loopback designs include RX and TX options. To use RX or TX only components, remove the irrelevant blocks from the designs.

User Requirement	Preserve	Remove
RX Only	RX Top	— TX Top — Transceiver Arbiter
TX Only	TX Top	— RX Top — Transceiver Arbiter

Note: You can directly connect the Avalon-MM pins at the RX or TX Top as shown in the diagram below.

Figure 7. Components Required for Intel Arria 10 TX or RX Only Design



2.1. Parallel Loopback Design Examples

The parallel loopback design examples demonstrate simplex and duplex channel modes with and without external VCXO.

Note: For parallel loopback duplex designs, do not share the TX PLL reference clock with the RX transceiver reference clock. The design logic tunes the TX PLL clock to match the RX recovered clock frequency. For the parallel loopback with external VCXO designs (single-rate and triple-rate), use the only 148.5 MHz on-board oscillator as the TX PLL reference clock. For the RX reference clock, use a 270 MHz clock from another on-board oscillator.

Figure 8. Parallel Loopback with Simplex Mode Block Diagram

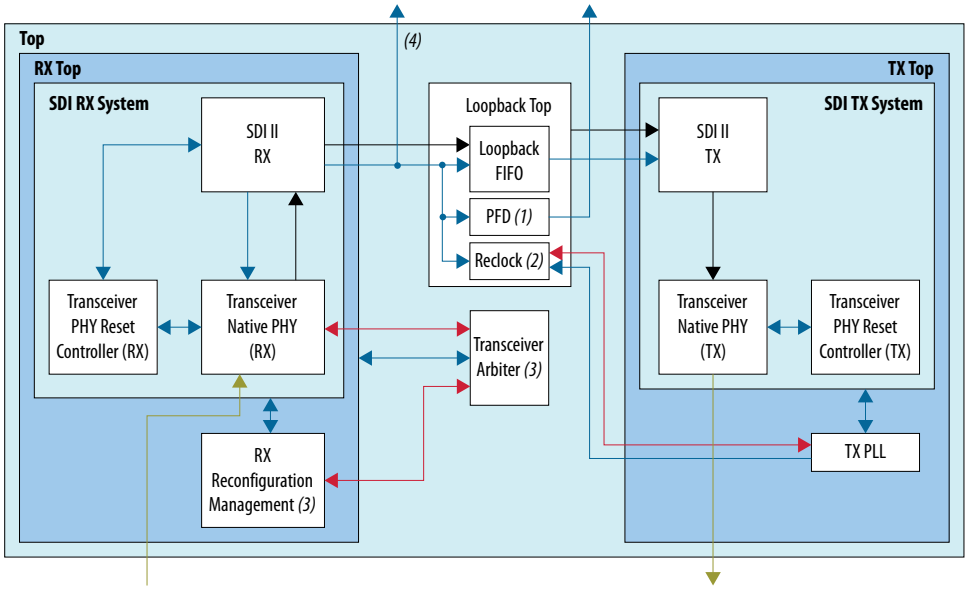


Figure 9. Parallel Loopback with Simplex Mode Clocking Scheme

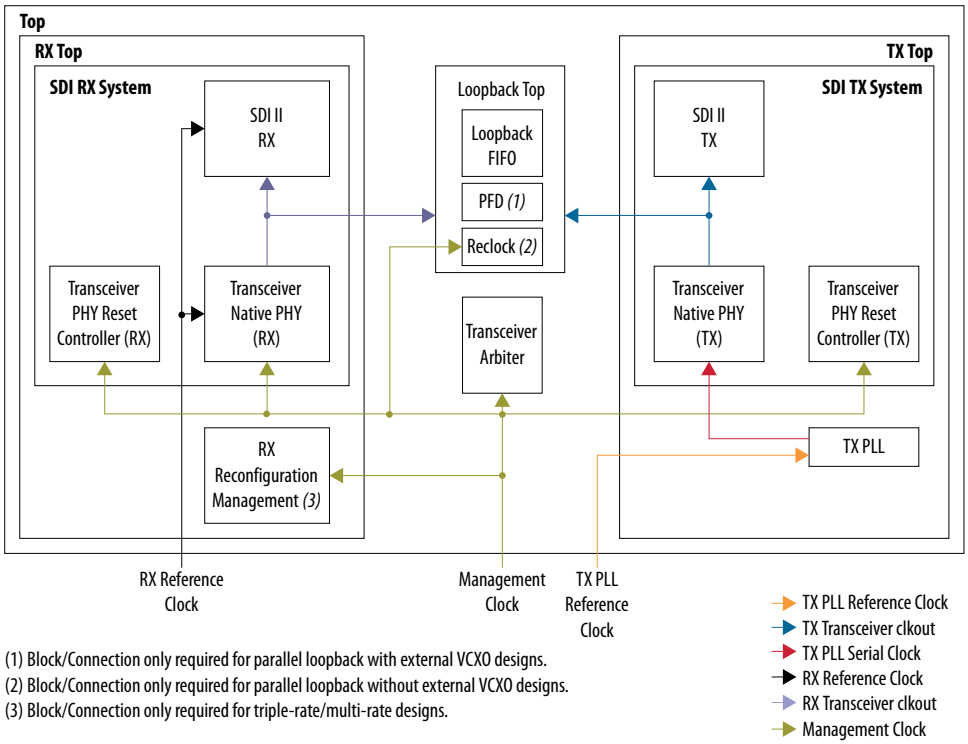
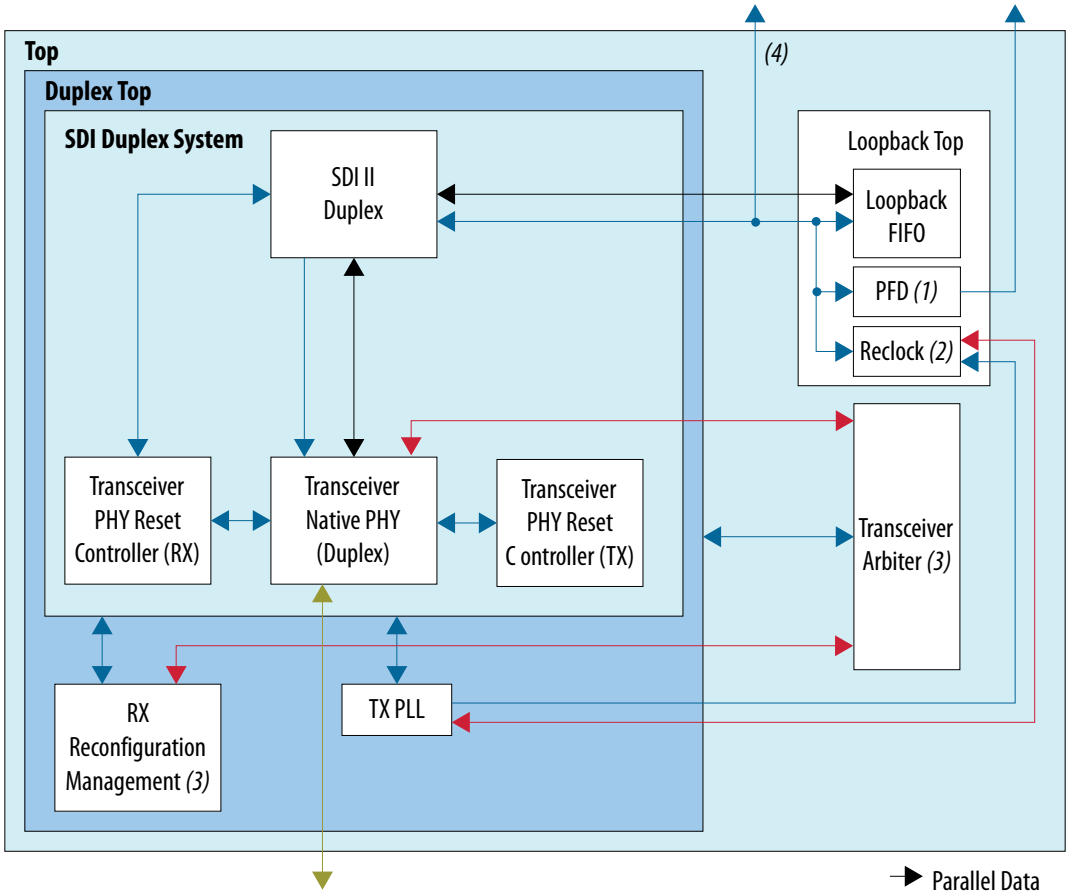
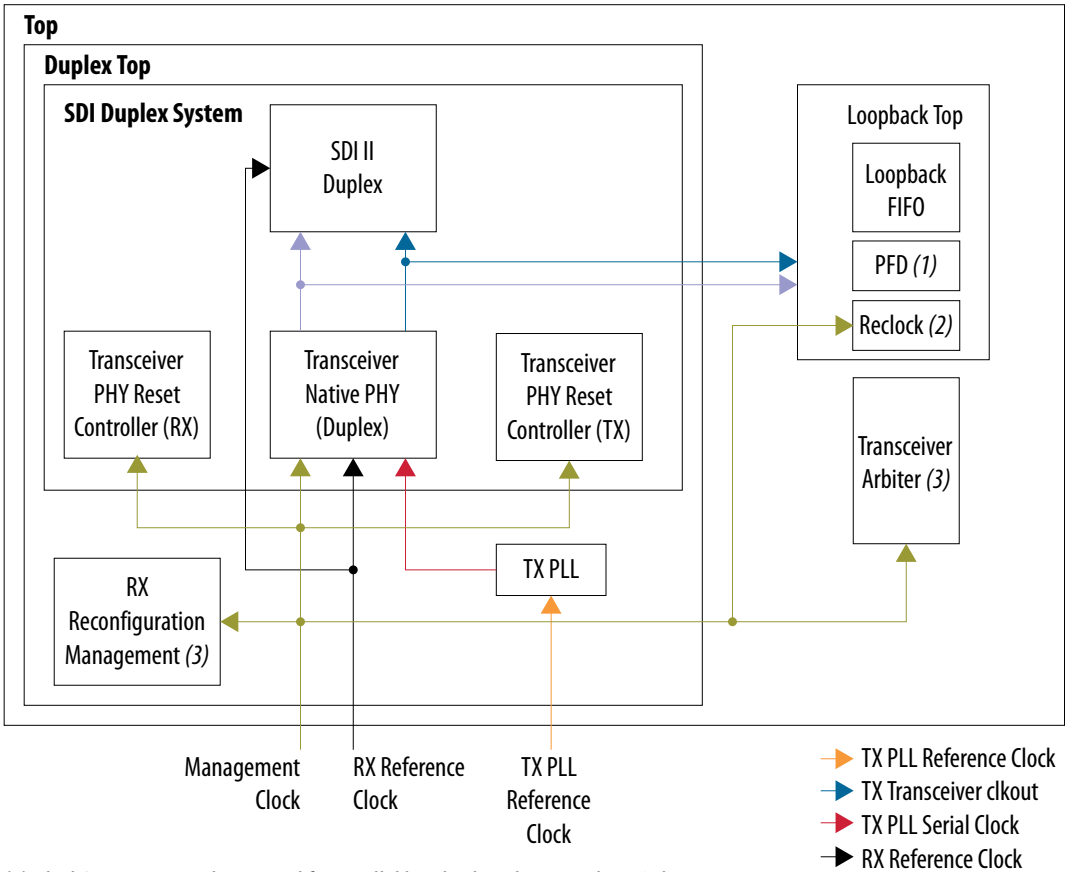


Figure 10. Parallel Loopback with Duplex Mode Block Diagram



(1) Generate up/down control signal to on-board Si516 for clock synchronization purpose.
(2) Block/Connection only required for parallel loopback without external VCXO designs.
(3) Block/Connection only required for triple-rate/multi-rate designs.
(4) FVH video sync signals to LMH1983 for clock synchronization purpose.

Figure 11. Parallel Loopback with Duplex Mode Clocking Scheme



(1) Block/Connection only required for parallel loopback with external VCXO designs.
(2) Block/Connection only required for parallel loopback without external VCXO designs.
(3) Block/Connection only required for triple-rate/multi-rate designs.

2.2. Serial Loopback Design Examples

The serial loopback design examples demonstrate simplex and duplex channel modes.

Figure 12. Serial Loopback with Simplex Mode Block Diagram

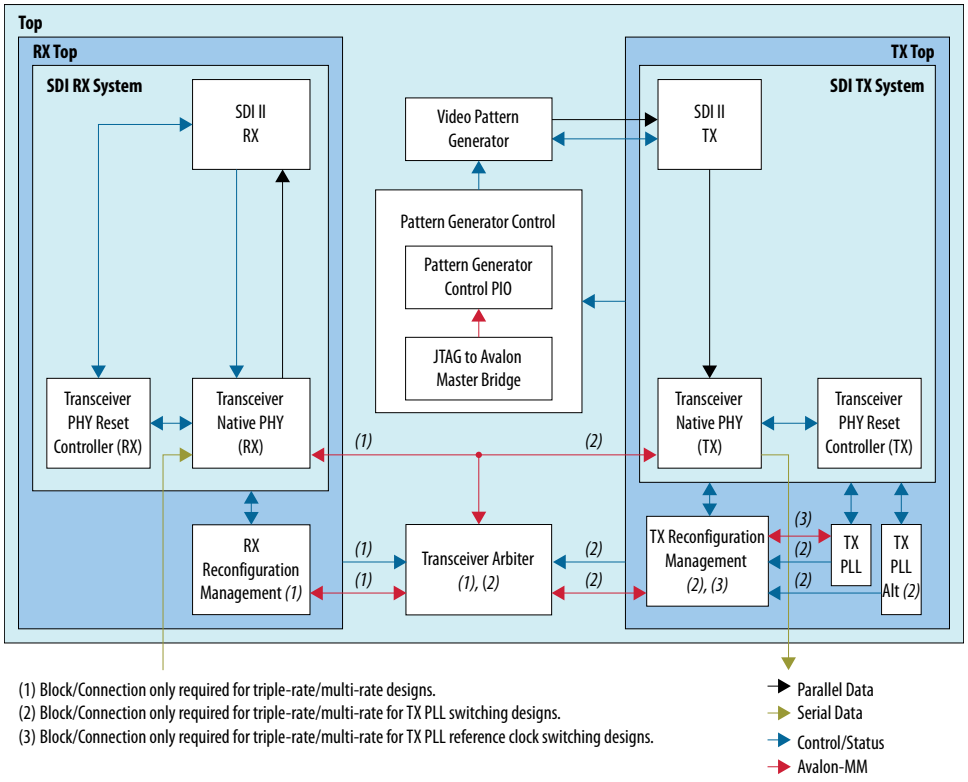
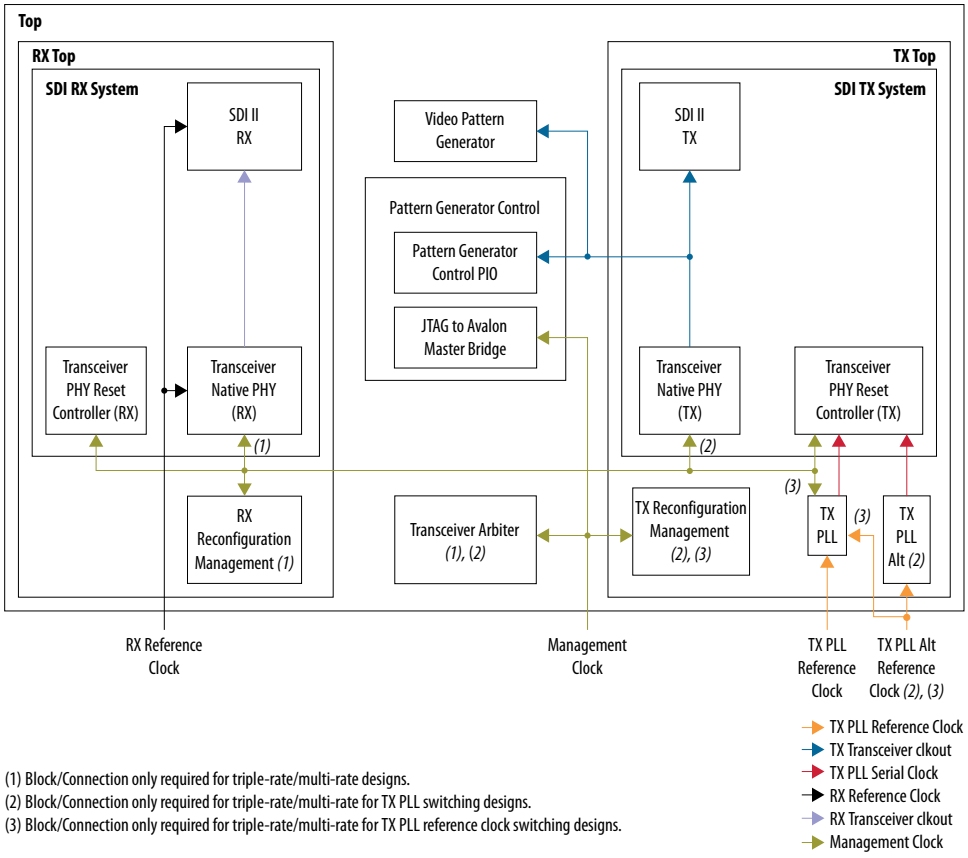
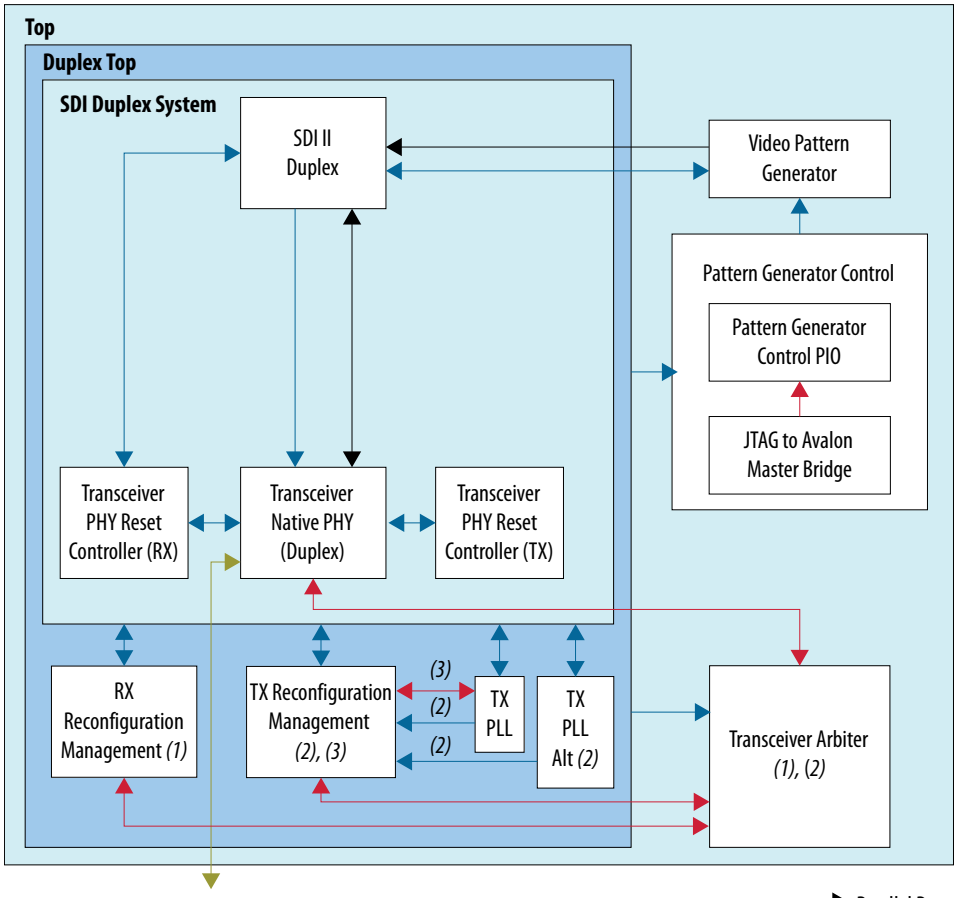


Figure 13. Serial Loopback with Simplex Mode Clocking Scheme



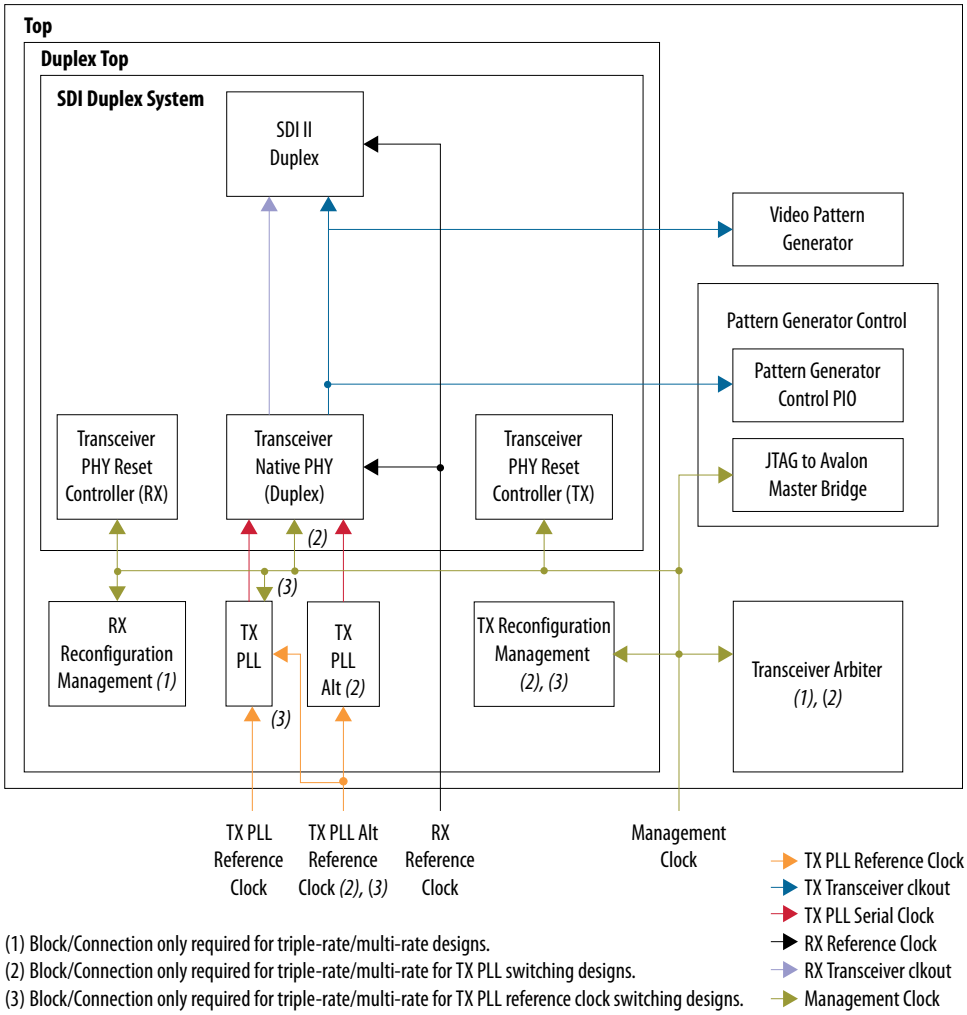
(1) Block/Connection only required for triple-rate/multi-rate designs.
(2) Block/Connection only required for triple-rate/multi-rate for TX PLL switching designs.
(3) Block/Connection only required for triple-rate/multi-rate for TX PLL reference clock switching designs.

Figure 14. Serial Loopback with Duplex Mode Block Diagram



(1) Block/Connection only required for triple-rate/multi-rate designs.
(2) Block/Connection only required for triple-rate/multi-rate for TX PLL switching designs.
(3) Block/Connection only required for triple-rate/multi-rate for TX PLL reference clock switching designs.

Figure 15. Serial Loopback with Duplex Mode Clocking Scheme



(1) Block/Connection only required for triple-rate/multi-rate designs.
(2) Block/Connection only required for triple-rate/multi-rate for TX PLL switching designs.
(3) Block/Connection only required for triple-rate/multi-rate for TX PLL reference clock switching designs.

2.3. Design Components

The SDI II Intel FPGA IP core design examples require the following components.

Table 7. Device Under Test (DUT) Components

Design Component	Description
SDI II Intel FPGA IP	• TX — The TX core receives the video data from the top level and encodes the necessary information, (e.g. line number (LN), cyclical redundancy check (CRC), payload ID), into the data stream(s). — In a multi-rate design, the TX core oversamples the received data up to 11.88 Gbps data rate for every video standard. — Specify the assignment of the parallel data interface (`tx_parallel_data`) to the transceiver based on the 11.88 Gbps data rate settings. • RX — The RX core receives the parallel data from the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP core and decodes information. This information includes descrambling, realigning data, and extracting the necessary information for user. — For a multi-rate design, due to the difference in data widths recovered for different video standards, rearrange `rx_parallel_data` from the transceiver before passing the data back to the protocol block.
Transceiver Native PHY Intel Arria 10/ Cyclone 10 FPGA IP	• TX Native PHY IP block that receives parallel data from the SDI II Intel FPGA IP core and serializes the data before transmission. — For HD/3G-SDI single-rate and triple-rate designs, enable the simplified data interface option to connect parallel data directly to the `tx_dataout` signal of the SDI II Intel FPGA IP core. — For a multi-rate design, disable this option due to the limitation in the 12G-SDI transceiver PHY settings. • RX Native PHY block that receives serial data from an external video source. — For HD/3G-SDI single-rate and triple-rate designs, enable the simplified data interface option to connect parallel data directly to the `rx_datain` signal of SDI II Intel FPGA IP core. — For a multi-rate design, disable this option due to the limitation in the 12G-SDI transceiver PHY settings. *Note:* You must connect the `rx_analogreset_ack` output signal from this block to the RX Reconfiguration Management module to indicate that the transceiver is in reset. For the duplex mode transceiver (SDI triple-rate parallel loopback with external VCXO design example), generate a dummy RX only PHY (`sdi_rx_phy.qsys/ sdi_rx_phy.ip`) to get the transceiver configuration files (`*_CFG0.sv`, `*_CFG1.sv`, …) for RX reconfiguration. The generated configuration files from the duplex mode transceiver may contain some TX registers. You need not reconfigure the registers because only the SDI RX core requires transceiver reconfiguration.
Transceiver PHY Reset Controller Intel FPGA IP	• TX — The reset input of this controller is triggered from the top level. — The controller generates the corresponding analog and digital reset signal to the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP block, according to the reset sequencing inside the block. — Use the `tx_ready` output signal from the block as a reset signal to the TX core to indicate that the transceiver is up and running, and ready to receive data from the core. • RX — The reset input of this controller is triggered by the SDI II Intel FPGA IP core. — The controller generates the corresponding analog and digital reset signal to the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP block according to the reset sequencing inside the block.

continued...

Design Component	Description
RX Reconfiguration Management	RX transceiver reconfiguration management block that reconfigures the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP block to receive different data rates from SD-SDI to 12G-SDI standards. To indicate the status of the transceiver, connect `rx_cal_busy` and `rx_analogreset_ack` from the transceiver to this block. *Note:* If you want to use the reconfiguration management block in your own design, you need to make some assignments in the QSF file. For guidelines about how to make the QSF assignments, refer to the *Using Generated Reconfiguration Management for Triple and Multi Rates* section in the *SDI II Intel FPGA IP User Guide.*
TX Reconfiguration Management	TX PLL or transceiver reconfiguration management block that reconfigures the TX PLL or Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP block to change the TX clock dynamically for switching between integer and fractional frame rates. The block requires `tx_cal_busy`, `pll_cal_busy`, and `tx_analogreset_ack` from the transceiver, and the PLLs to indicate the status of the transceiver in a TX PLL switching design.
TX PLL/TX PLL Alt	Transmitter PLL block that provides the serial fast clock to Transceiver Native PHY. • For TX PLL switching design, TX PLL is always configured to generate integer frame rate while TX PLL Alt is configured to generate fractional frame rate. • For TX PLL reference clock switching design, TX PLL is configured to have reference clock 0 to generate integer frame rate and reference clock 1 to generate fractional frame rate. • For single-rate and triple-rate designs, this PLL can be either CMU PLL or fPLL. • For multi-rate designs, CMU PLL is not recommended for 12G data rate. Use fPLL instead. Move the TX PLL out from the TX top if you want to merge the PLL between multiple channels.

Table 8. Loopback Components

Component	Description
Loopback FIFO	This block contains a dual-clock FIFO (DCFIFO) buffer to handle the data transmission across asynchronous clock domains—the receiver recovered clock and transmitter clock out. • The receiver sends the decoded RX data to the transmitter through this FIFO buffer. • When the receiver locks, the RX data is written to the FIFO buffer. • The transmitter starts reading, encoding, and transmitting the data when half of the FIFO buffer is filled.
Phase Frequency Detector (PFD)	You require this soft PFD block when you use the Intel Arria 10 FPGA development kit on-board Si516 VCXO for a parallel loopback design. • This block compares the phase between the receiver and transmitter parallel clocks, and generates an up or down signal, that connects to the Si516 VCXO. • These up/down signals control the voltage of the VCXO, so that the frequencies of both clock domains can be tuned as close as possible to each other. *Note:* Applicable only for single-rate and triple-rate parallel loopback with external VCXO designs.
Reclock	The parallel loopback without external VCXO design requires this module. This block compares the phase between the receiver and transmitter parallel clocks. The output interfaces of this block connect to the reconfiguration Avalon Memory-Mapped (Avalon-MM) interfaces of an fPLL. If there is any difference in the frequencies between the clock domains, this module generates the necessary signals to reconfigure the fPLL to match the clock frequencies as close as possible. *Note:* Applicable only for parallel loopback without external VCXO designs.

Table 9. Video Pattern Generator Components

Component	Description
Video Pattern Generator	Basic video pattern generator which supports SD-SDI up to 12G-SDI video formats with 4:2:2 YCbCr. The generator enables you to select static video with colorbar pattern or pathological pattern.
Pattern Gen Control PIO	Provides a memory-mapped interface for controlling the video pattern generator.
JTAG to Avalon Master Bridge	Provides System Console host access to the Parallel I/O (PIO) IP core in the design through the JTAG interface.

Table 10. Common Block

Component	Description
Transceiver Arbiter	This generic functional block prevents transceivers from recalibrating simultaneously when either RX or TX transceivers within the same physical channel require reconfiguration. The simultaneous recalibration impacts applications where RX and TX transceivers within the same channel are assigned to independent IP implementations. This transceiver arbiter is an extension to the resolution recommended for merging simplex TX and simplex RX into the same physical channel. This transceiver arbiter also assists in merging and arbitrating the Avalon-MM RX and TX reconfiguration requests targeting simplex RX and TX transceivers within a channel as the reconfiguration interface port of the transceivers can only be accessed sequentially. The transceiver arbiter is not required when only either RX or TX transceiver is used in a channel. The transceiver arbiter identifies the requester of a reconfiguration through its Avalon-MM reconfiguration interfaces and ensures that the corresponding `tx_reconfig_cal_busy` or `rx_reconfig_cal_busy` is gated accordingly.

Related Information

SDI II Intel FPGA IP User Guide
Provides guidelines about how to make the QSF assignments to use the reconfiguration management block in your own design.

2.4. Clocking Scheme Signals

The table lists the clocking scheme signals for the SDI II Intel FPGA IP core design examples.

Table 11. Clocking Scheme Signals

Clock	Signal Name in Design	Description
TX PLL Refclock	`tx_pll_refclk`	TX PLL reference clock, of any frequency that is divisible by the transceiver for that data rate.

continued...

Clock	Signal Name in Design	Description
		Note: You must connect this clock to a dedicated transceiver reference clock pin. • Parallel loopback with external VCXO — Use a minimum clock frequency of 148.5 MHz (single-rate/triple-rate) or 297 MHz (multi-rate) to meet jitter performance specification. — If you use a higher clock frequency, you would need to modify the TX PLL reference clock value in the TX PLL parameter editor. • Parallel loopback without external VCXO — The recommended frequency is 100 MHz. • Serial loopback — For this design, the TX PLL refclock is configured to generate clock for integer frame rate. — The minimum clock frequency is 148.5 MHz (single-rate/triple-rate) or 297 MHz (multi-rate) to meet jitter performance specification. — If you use a higher clock frequency, you would need to modify the TX PLL reference clock value in the TX PLL parameter editor.
TX PLL Alt Refclock	`tx_pll_refclk_alt`	Second TX PLL reference clock which can be any clock frequency that is divisible by transceiver for that data rate. This clock must be connected to a dedicated transceiver reference clock pin. • Serial loopback — For this design example, TX PLL alt refclock is configured to generate clock for fractional frame rate. — The minimum clock frequency is 148.35 MHz (single-rate/triple-rate) or 296.7 MHz (multi-rate) to meet jitter performance specification. — If you use a higher clock frequency, you would need to modify the TX PLL reference clock value in the TX PLL parameter editor.
TX Transceiver Clockout	`tx_vid_clkout`	Recovered clock from the transceiver. • HD-SDI single rate — 74.25 MHz (default) — 74.1758 MHz (for the Dynamic TX clock switching feature when you transmit video format with fractional frame rate) • 3G-SDI single rate, triple rate or multi rate — 148.5 MHz (default) — 148.35 MHz (for the Dynamic TX clock switching feature when you transmit video format with fractional frame rate)
TX PLL Serial Clock	`tx_serial_clk`	Serial fast clock generated by TX PLL. The clock frequency is set based on the data rate.
RX Refclock	`rx_cdr_refclk`	Transceiver clock data recovery (CDR) reference clock, of any frequency divisible by the transceiver for that data rate. Only a single reference clock frequency is required to support both integer and fractional frame rates. It must be a free running clock connected to the transceiver clock pin. For the Intel Arria 10 design example, the multi-rate designs use the same clock frequency as the Rx core clock frequency—148.5 or 297 MHz based on what you specify for the **Rx core clock (rx_coreclk) frequency** parameter. The single-rate and triple-rate designs use a higher reference clock (270 MHz) instead. Using a higher clock frequency would require a modification of the RX CDR reference clock value in the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP core parameter editor. For triple or multi-rate modes, you need to modify the reference clock value for every profile. *Note:* Do not share the TX PLL reference clock with the RX transceiver reference clock for a parallel loopback design. In parallel loopback designs, the TX PLL clock is tuned to match the RX recovered clock frequency.

continued...

<table>
<tr><th>Clock</th><th>Signal Name in Design</th><th>Description</th></tr>
<tr><td></td><td>rx_core_refclk</td><td>SDI RX core reference clock.
The required frequency is 148.5/148.35 MHz or 297/296.7 MHz depending on what you specify for the Rx core clock (rx_coreclk) frequency parameter. This clock must be a free-running clock.</td></tr>
<tr><td>RX Transceiver Clkout</td><td>rx_vid_clkout</td><td>Recovered clock from the transceiver.
• SD-SDI
— 148.5 MHz (default)
• HD-SDI
— 74.25 MHz when receiving integer frame rate
— 74.1758 MHz when receiving fractional frame rate
• 3G/6G/12-SDI
— 148.5 MHz when receiving integer frame rate
— 148.35 MHz when receiving fractional frame rate</td></tr>
<tr><td rowspan="4">Management Clock</td><td rowspan="2">rx_rcfg_mgmt_clk</td><td>A free-running 100 MHz RX clock used by Avalon-MM interfaces for reconfiguration and by the PHY reset controller for transceiver reset sequence. 100 MHz frequency is used if you set the Rx core clock (rx_coreclk) frequency parameter to 297 MHz. Otherwise, 148.5 MHz is used to share between this clock and rx_coreclk.</td></tr>
<tr><td><table><tr><th>Component</th><th>Required Frequency (MHz)</th></tr><tr><td>Avalon-MM reconfiguration</td><td>100 – 125</td></tr><tr><td>Transceiver PHY reset controller</td><td>1 – 500</td></tr></table></td></tr>
<tr><td rowspan="2">tx_rcfg_mgmt_clk</td><td>A free-running 100 MHz TX clock used by Avalon-MM interfaces for reconfiguration and by the PHY reset controller for transceiver reset sequence. 100 MHz frequency is used if you set the Rx core clock (rx_coreclk) frequency parameter to 297 MHz. Otherwise, 148.5 MHz is used to share between this clock and rx_coreclk. Otherwise, 148.5 MHz is used to share between this clock and rx_coreclk.</td></tr>
<tr><td><table><tr><th>Component</th><th>Required Frequency (MHz)</th></tr><tr><td>Avalon-MM reconfiguration</td><td>100 – 125</td></tr><tr><td>Transceiver PHY reset controller</td><td>1 – 500</td></tr></table></td></tr>
</table>

2.5. Interface Signals

The tables list the signals for the SDI II IP core design examples.

Table 12. Top-Level Signals

Signal	Direction	Width	Description
On-board Oscillator Signals			
clk_fpga_b2_p	Input	1	100 MHz clock for reconfiguration Avalon-MM interfaces.
pcie_ob_refclk_p	Input	1	100 MHz dedicated transceiver reference clock.
refclk_dp_p	Input	1	270 MHz dedicated transceiver reference clock.
refclk_sdi_p	Input	1	148.5 or 148.35 MHz dedicated transceiver reference clock.

continued...

Signal	Direction	Width	Description
On-board Oscillator Signals			
refclk_sma_p	Input	1	302 MHz dedicated transceiver reference clock. Programmable to 148.3516 MHz from the Clock Control GUI.
refclk_fmca_p	Input		625 MHz dedicated transceiver reference clock. Programmable to 297.0/296.7 MHz from the Clock Control GUI.
refclk_fmcb_p	Input	1	625 MHz dedicated transceiver reference clock. Programmable to 296.7033 MHz from the Clock Control GUI.

User Push Buttons and LEDs			
user_pb0	Input	1	Push button to power down LMK03328 after switching the jumper settings.
cpu_resetn	Input	1	Global reset.
user_led_g	Output	8	Green LED display.
user_led_r	Output	8	Red LED display.

On-board Si516, SDI Cable Driver and Equalizer Related Pins			
sdi_rx_p	Input	1	On-board SDI RX serial data.
sdi_tx_p	Output	1	On-board SDI TX serial data.
sdi_clk148_up	Output	1	Voltage control for Si516.
sdi_clk148_down	Output	1	Voltage control for Si516.
sdi_mf0_bypass	Output	1	On-board SDI RX Equalizer Bypass.
sdi_mf1_auto_sleep	Output	1	On-board SDI RX Equalizer Auto Sleep.
sdi_mf1_mute	Output	1	On-board SDI RX Equalizer Mute.
sdi_tx_sd_hdn	Output	1	On-board SDI TX cable driver slew rate control.

Nextera SDI FMC Daughter Card Pins on FMC Port B			
fmcb_gbtclk_m2c_p0	Input	1	297 or 296.7 MHz dedicated transceiver reference clock from FMC port B.
fmcb_dp_m2c_p2	Input	1	SDI RX serial data from FMC port B.
fmcb_la_tx_p1	Input	1	RX cable equalizer lock status on Nextera daughter card.
fmcb_dp_c2m_p0	Output	1	SDI TX serial data from FMC port B.
fmcb_la_tx_p12	Output	1	Initialize LMH1983 on Nextera daughter card.
fmcb_la_tx_n12	Output	1	F sync signal LMH1983 on Nextera daughter card.
fmcb_la_tx_p14	Output	1	V sync signal LMH1983 on Nextera daughter card.
fmcb_la_tx_n14	Output	1	H sync signal LMH1983 on Nextera daughter card.
fmcb_la_tx_p15	Output	1	Power-down signal LMH1983 on Nextera daughter card.

Table 13. RX/TX/DU Top Signals

Signal	Direction	Width	Description
Clocks			
rx_cdr_refclk	Input	1	RX transceiver reference clock. This clock must be a free-running clock.
rx_core_refclk	Input	1	SDI RX core clock. This clock must be a free-running clock.
tx_pll_refclk	Input	1	TX PLL reference clock. This clock must be a free-running clock.
tx_pll_refclk_alt	Input	1	Secondary TX PLL reference clock. This clock must be a free-running clock.
rx_rcfg_mgmt_clk	Input	1	RX reconfiguration management clock, Avalon-MM interface clock, and PHY reset control input clock. This clock must be a free-running clock.
tx_rcfg_mgmt_clk	Input	1	TX reconfiguration management clock, and Avalon-MM interface clock, and PHY reset control input clock. This clock must be a free-running clock.
rx_vid_clkout	Output	1	RX transceiver recovered parallel clock for video data.
tx_vid_clkout	Output	1	TX transceiver recovered parallel clock for video data.
Reset			
tx_resetn	Input	1	TX core and PHY reset signal.
rx_resetn	Input	1	RX core and PHY reset signal.
tx_rcfg_mgmt_resetn	Input	1	TX reconfiguration reset signal.
rx_rcfg_mgmt_resetn	Input	1	RX reconfiguration reset signal.
sdi_rx_rst_proto_out	Output	1	Reset signal generated to reset the receiver downstream protocol logic. This generated reset signal is synchronous to rx_vid_clkout clock domain.
Video Signal Interfaces (Interface with Video Image and Processing (VIP) Components)			
rx_vid_data	Output	20*N	Receiver parallel video data out. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
rx_vid_datavalid	Output	1	Data valid signal generated from SDI RX core. The timing must be synchronous to rx_vid_clkout and has the following settings: • SD-SDI: 1H 4L 1H 5L • HD/3G/6G/12G-SDI: H
rx_vid_std	Output	3	Received video standard. • 3'b000: SD-SDI • 3'b001: HD-SDI • 3'b011: 3G-SDI Level A • 3'b010 3G-SDI Level B • 3'b101: 6G-SDI 4 Streams Interleaved • 3'b100: 6G-SDI 8 Streams Interleaved • 3'b111: 12G-SDI 8 Streams Interleaved • 3'b110: 12G-SDI16 Streams Interleaved

continued...

Video Signal Interfaces (Interface with Video Image and Processing (VIP) Components)			
rx_vid_locked	Output	1	Frame locked indicates that the IP core has spotted multiple frames with the same timing.
rx_vid_hsync	Output	*N*	Horizontal blanking interval timing signal. The receiver asserts this signal when the horizontal blanking interval is active. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
rx_vid_vsync	Output	*N*	Vertical blanking interval timing signal. The receiver asserts this signal when the vertical blanking interval is active. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
rx_vid_f	Output	*N*	Field bit timing signal. This signal indicates which video field is currently active. For interfaced frame, 0 means first field (F0) while 1 means second field (F1). For progressive frame, the value is always 0. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
rx_vid_trs	Output	*N*	On-board SDI TX cable driver slew rate control. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
tx_vid_data	Output	20**N*	Receiver output signal that indicates current word is timing reference signal (TRS). This signal asserts at the first word of 3FF 000 000 TRS. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
tx_vid_datavalid	Input	1	Transmitter parallel data valid. The timing (H: High, L: Low) must be synchronous to tx_pclk clock domain and has the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H (for single-rate) and 1H 1L (triple-rate/ multi-rate) • 3G/6G/12G-SDI = H
tx_vid_std	Input	3	Indicates the desired transmit video standard. • 3'b000: SD-SDI • 3'b001: HD-SDI • 3'b011: 3G-SDI Level A • 3'b010 3G-SDI Level B • 3'b101: 6G-SDI 4 Streams Interleaved • 3'b100: 6G-SDI 8 Streams Interleaved • 3'b111: 12G-SDI 8 Streams Interleaved • 3'b110: 12G-SDI16 Streams Interleaved
tx_vid_trs	Input	1	Transmitter TRS input. For use in LN, CRC, or payload ID insertion. Assert on the first word of both end of active video (EAV) TRS and start of active video (SAV) TRS.

Other SDI Video Protocol Interfaces			
sdi_tx_enable_crc	Input	1	Enable CRC insertion for all SDI video standards, except SD-SDI.
sdi_tx_enable_ln	Input	1	Enable LN insertion for all SDI video standards, except SD-SDI.
sdi_tx_ln	Input	11**N*	LN insertion in the data stream when sdi_tx_enable_ln = 1. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)

continued...

Other SDI Video Protocol Interfaces			
sdi_tx_ln_b	Input	11*N	LN insertion in the data stream when sdi_tx_enable_ln = 1. Only for 3G level B, 6G 8 streams interleaved, and 12G 16 streams interleaved. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_tx_vpid_overwrite	Input	1	Enable this signal to overwrite the existing payload ID embedded in the data stream.
sdi_tx_line_f0	Input	11*N	Indicates the line number to be inserted with the payload ID.
sdi_tx_line_f1	Input	11*N	
sdi_tx_vpid_byte1	Input	8*N	Payload ID byte to be inserted in the payload ID field.
sdi_tx_vpid_byte2	Input	8*N	
sdi_tx_vpid_byte3	Input	8*N	
sdi_tx_vpid_byte4	Input	8*N	
sdi_tx_vpid_byte1_b	Input	8*N	
sdi_tx_vpid_byte2_b	Input	8*N	
sdi_tx_vpid_byte3_b	Input	8*N	
sdi_tx_vpid_byte4_b	Input	8*N	
sdi_rx_coreclk_is_ntsc_paln	Input	1	To indicate whether rx_coreclk is 148.5 MHz or 148.35 MHz: • 0: 148.5 MHz • 1: 148.35 MHz
sdi_tx_datavalid	Output	1	Data valid signal generated from SDI TX core. The timing (H: High, L: Low) is synchronous to tx_vid_clkout and has the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H (for single-rate) and 1H 1L (triple-rate/ multi-rate) • 3G/6G/12G-SDI = H
sdi_rx_align_locked	Output	1	Alignment locked indicating the IP core has spotted a TRS and word alignment performed.
sdi_rx_trs_locked	Output	*N*	TRS locked indicating the IP core has spotted six consecutive TRS with same timing. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_rx_clkout_is_ntsc_paln	Output	1	Indicates that the receiver is receiving video rate at integer or fractional frame rate: • 0: Integer frame rate • 1: Fractional frame rate
sdi_rx_format	Output	4*N	Received video transport format. Refer to the *SDI II IP User Guide* for the encoding value. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_rx_ap	Output	*N*	Active picture interval timing signal. This signal asserts when the active picture interval is active.
sdi_rx_eav	Output	*N*	Receiver output signal that indicates current TRS is EAV. This signal is asserted at the fourth word of TRS, which is the XYZ word.

continued...

Other SDI Video Protocol Interfaces			
sdi_rx_ln	Output	11*N	Received line number from protocol.
sdi_rx_ln_b	Output	11*N	
sdi_rx_crc_error_c	Output	N	CRC error status signal from protocol.
sdi_rx_crc_error_y	Output	N	
sdi_rx_crc_error_c_b	Output	N	
sdi_rx_crc_error_y_b	Output	N	
sdi_rx_line_f0	Output	11*N	Payload ID status from protocol.
sdi_rx_line_f1	Output	11*N	
sdi_rx_vpid_byte1	Output	8*N	
sdi_rx_vpid_byte2	Output	8*N	
sdi_rx_vpid_byte3	Output	8*N	
sdi_rx_vpid_byte4	Output	8*N	
sdi_rx_vpid_checksum_error	Output	N	
sdi_rx_vpid_valid	Output	N	
sdi_rx_vpid_byte1_b	Output	8*N	
sdi_rx_vpid_byte2_b	Output	8*N	
sdi_rx_vpid_byte3_b	Output	8*N	
sdi_rx_vpid_byte4_b	Output	8*N	
sdi_rx_vpid_checksum_error_b	Output	N	
sdi_rx_vpid_valid_b	Output	N	

Transceiver Interfaces			
tx_pll_refclk_sel	Input	1	Indicate which of pll_locked signals to be monitored for TX PHY reset controller's reset sequencing. Always set to 1'b0 if only one PLL is in use.
tx_rcfg_cal_busy	Input	1	Transceiver calibration status to TX PHY reset controller.
rx_rcfg_cal_busy	Input	1	Transceiver calibration status to RX PHY reset controller and Rx reconfiguration management module.
gxb_rx_serial_data	Input	1	RX transceiver serial data.
gxb_tx_serial_data	Output	1	TX transceiver serial data.
gxb_rx_ready	Output	1	RX transceiver status.
gxb_tx_ready	Output	1	TX transceiver status.
gxb_rx_cal_busy	Output	1	Calibration status signal from RX transceiver.
gxb_tx_cal_busy	Output	1	Calibration status signal from TX transceiver.
tx_pll_locked	Output	1	TX PLL lock status.

continued...

Transceiver Interfaces			
tx_pll_locked_alt	Output	1	TX PLL alt lock status.
cdr_reconfig_busy	Output	1	RX CDR reconfiguration status.
tx_reconfig_busy	Output	1	TX PLL/transceiver reconfiguration status.

Transceiver Reconfiguration Interfaces			
gxb_du_rcfg_write	Input	1	Reconfiguration interface signals from transceiver arbiter to duplex mode transceiver.
gxb_du_rcfg_read	Input	1	
gxb_du_rcfg_address	Input	10	
gxb_du_rcfg_writedata	Input	32	
gxb_du_rcfg_readdata	Output	32	
gxb_du_rcfg_waitrequest	Output	1	
gxb_rx_rcfg_write	Input	1	Reconfiguration interface signals from transceiver arbiter to RX transceiver.
gxb_rx_rcfg_read	Input	1	
gxb_rx_rcfg_address	Input	10	
gxb_rx_rcfg_writedata	Input	32	
gxb_rx_rcfg_readdata	Output	32	
gxb_rx_rcfg_waitrequest	Output	1	
gxb_tx_rcfg_write	Input	1	Reconfiguration interface signals from transceiver arbiter to TX transceiver.
gxb_tx_rcfg_read	Input	1	
gxb_tx_rcfg_address	Input	10	
gxb_tx_rcfg_writedata	Input	32	
gxb_tx_rcfg_readdata	Output	32	
gxb_tx_rcfg_waitrequest	Output	1	
rx_rcfg_readdata	Input	32	Reconfiguration interface signals from RX reconfiguration management module to transceiver arbiter.
rx_rcfg_waitrequest	Input	1	
rx_rcfg_write	Output	1	
rx_rcfg_read	Output	1	
rx_rcfg_address	Output	10	
rx_rcfg_writedata	Output	32	
tx_rcfg_readdata	Input	32	Reconfiguration interface signals from TX reconfiguration management module to transceiver arbiter
tx_rcfg_waitrequest	Input	1	

continued...

Transceiver Reconfiguration Interfaces			
tx_rcfg_write	Output	1	
tx_rcfg_read	Output	1	
tx_rcfg_address	Output	10	
tx_rcfg_writedata	Output	32	
tx_fpll_rcfg_write	Input	1	Reconfiguration interface signals to fPLL Avalon-MM interface.
tx_fpll_rcfg_read	Input	1	
tx_fpll_rcfg_writedata	Input	32	
tx_fpll_rcfg_address	Input	10	
tx_fpll_rcfg_readdata	Output	32	
tx_fpll_rcfg_waitrequest	Output	1	

Table 14. Loopback Top Signals

Signal	Direction	Width	Description
Clocks			
sdi_tx_clkout	Input	1	TX transceiver recovered parallel clock for video data.
sdi_rx_clkout	Input	1	RX transceiver recovered parallel clock for video data.
sdi_reclk_sysclk	Input	1	Input clock for reclock module (without external VCXO solution). This clock should be the same as fPLL reconfig_clk.
Resets			
sdi_rx_rst_proto	Input	1	Reset signal from SDI RX core to indicate that the protocol is currently held in reset.
sdi_reclk_rst	Input	1	Reset signal to reclock module (without external VCXO solution).
SDI Related Signals			
sdi_rx_dataout	Input	20*N	Receiver recovered parallel video data. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_rx_dataout_valid	Input	1	Data valid signal generated from SDI RX core.
sdi_rx_std	Input	3	Received video standard from SDI RX core.
sdi_rx_trs	Input	*N*	Receiver output signal from SDI II IP core that indicates current word is TRS. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_rx_trs_locked	Input	*N*	TRS locked status signal from SDI RX core. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_rx_frame_locked	Input	1	Frame locked status signal from SDI RX core.
sdi_tx_dataout_valid	Input	1	Data valid signal generated from SDI TX core.

continued...

SDI Related Signals			
sdi_rx_h	Input	1	Horizontal blanking interval timing signal extracted from SDI RX core.
sdi_rx_format	Input	4	Received video transport format.
sdi_rx_clkout_is_ntsc_paln	Input	1	Indication from SDI RX core that the receiver is receiving video rate at integer or fractional frame rate.
sdi_tx_datain	Output	20*N	Parallel video data input to SDI TX core. *Note: N* = 4 (multi-rate design) or 1 (triple-rate design)
sdi_tx_datain_valid	Output	1	Data valid for the transmitter parallel data to SDI TX core.
sdi_tx_trs	Output	1	Transmitter TRS input to indicate that the current word is a TRS to SDI TX core.
sdi_tx_std	Output	3	Indicates the desired transmit video standard to SDI TX core.

Voltage Control Signals for On-board Si516			
vcoclk_up	Output	1	Voltage up signal to Si516 to increase the voltage.
vcoclk_down	Output	1	Voltage down signal to Si516 to decrease the voltage.

fPLL Reconfiguration Signals			
pll_locked	Input	1	PLL lock status signal.
pll_reconfig_readdata	Input	32	Reconfiguration interface signals to fPLL Avalon-MM interface.
pll_reconfig_waitrequest	Input	1	
pll_reconfig_write	Output	1	
pll_reconfig_read	Output	1	
pll_reconfig_writedata	Output	32	
pll_reconfig_address	Output	10	

Table 15. Transceiver Arbiter Signals

Signal	Direction	Width	Description
On-board Oscillator Signals			
clk	Input	1	Reconfiguration clock. This clock should be sharing the same clock as reconfiguration management blocks.
reset	Input	1	Reset signal. This reset should be sharing the same reset as reconfiguration management blocks.
rx_rcfg_en	Input	1	RX reconfiguration enable signal.
tx_rcfg_en	Input	1	TX reconfiguration enable signal.
rx_rcfg_ch	Input	2	Indicates which channel to be reconfigured on RX. Always assign to 2'b00 for SDI case.

continued...

Signal	Direction	Width	Description
On-board Oscillator Signals			
tx_rcfg_ch	Input	2	Indicates which channel to be reconfigured on TX. Always assign to 2'b00 for SDI case.
rx_reconfig_mgmt_write	Input	1	Reconfiguration Avalon-MM interfaces from RX reconfiguration management.
rx_reconfig_mgmt_read	Input	1	
rx_reconfig_mgmt_address	Input	10	
rx_reconfig_mgmt_writedata	Input	32	
rx_reconfig_mgmt_readdata	Output	32	
rx_reconfig_mgmt_waitrequest	Output	1	
tx_reconfig_mgmt_write	Input	1	Reconfiguration Avalon-MM interfaces from TX reconfiguration management.
tx_reconfig_mgmt_read	Input	1	
tx_reconfig_mgmt_address	Input	10	
tx_reconfig_mgmt_writedata	Input	32	
tx_reconfig_mgmt_readdata	Output	32	
tx_reconfig_mgmt_waitrequest	Output	1	
reconfig_write	Output	1	Reconfiguration Avalon-MM interfaces to transceiver.
reconfig_read	Output	1	
reconfig_address	Output	10	
reconfig_writedata	Output	32	
rx_reconfig_readdata	Input	32	
rx_reconfig_waitrequest	Input	1	
tx_reconfig_readdata	Input	1	
tx_reconfig_waitrequest	Input	1	
rx_cal_busy	Input	1	Calibration status signal from RX transceiver.
tx_cal_busy	Input	1	Calibration status signal from TX transceiver.
rx_reconfig_cal_busy	Output	1	Calibration status signal to RX transceiver PHY reset control.
tx_reconfig_cal_busy	Output	1	Calibration status signal from TX transceiver PHY reset control.

Video Pattern Generator Signals			
clk	Input	1	Clock signal. This clock must be connected to the tx_vid_clkout input signal on TX/Du top.
rst	Input	1	Reset signal. This reset signal should be synchronized with the tx_vid_clkout clock signal from the TX/Du top.
bar_100_75n	Input	1	Enable this signal to generate 100% color-bar pattern. Disable to generate 75% color-bar pattern.
enable	Input	1	This signal acts as a data valid signal to this module. This signal should be connected to the sdi_tx_datavalid signal from the TX/Du top.
patho	Input	1	Enable this signal to generate pathological pattern.
blank	Input	1	Enable this signal to generate blank signal.
no_color	Input	1	Enable this signal to generate bar with no color.
sgmt_frame	Input	1	Enable this signal to generate payload ID for segmented frame video format when generating 1080i50 or 1080i60 video.
tx_std	Input	3	Indicates the desired transmit video standard. This input signal must match tx_vid_std on the TX/Du top.
tx_format	Input	4	Indicates the desired transmit video format.
dl_mapping	Input	1	Enable this signal to generate data streams with dual-link mapping. *Note:* Applicable only for HD dual link or 3G Level B dual link video standard.
ntsc_paln	Input	1	Enable this signal to generate payload ID for fractional frame rate video format. Disable to generate integer frame rate video format.
dout	Output	20*S*	Data output signal to be connected to the tx_vid_data input signal on the TX/Du top.
dout_valid	Output	1	Data valid output signal to be connected to the tx_vid_datavalid input signal on the TX/Du top.
trs	Output	1	TRS output signal to be connected to the tx_vid_trs input signal on the TX/Du top.
ln	Output	11*S*	Line number output signal to be connected to the sdi_tx_ln input signal on the TX/Du top.
dout_b	Output	20*S*	Data output signal for link B (HD dual link).
dout_valid_b	Output	1	Data valid output signal for link B (HD dual link).
trs_b	Output	1	TRS output signal for link B (HD dual link).
ln_b	Output	11*S*	Line number output signal to be connected to the sdi_tx_ln_b input signal on the TX/Du top.
vpid_byte1	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte1 input signal on TX/Du top.
vpid_byte2	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte2 input signal on TX/Du top.
vpid_byte3	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte3 input signal on TX/Du top.

continued...

Video Pattern Generator Signals			
vpid_byte4	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte4 input signal on TX/Du top.
vpid_byte1_b	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte1_b input signal on TX/Du top.
vpid_byte2_b	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte2_b input signal on TX/Du top.
vpid_byte3_b	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte3_b input signal on TX/Du top.
vpid_byte4_b	Input	8*N*	The payload ID output signal to be connected to sdi_tx_vpid_byte4_b input signal on TX/Du top.
line_f0	Output	11*N*	The line number output signal to be inserted with the payload ID. This signal must connect to sdi_tx_line_f0 input signal on TX/Du top.
line_f1	Output	11*N*	The line number output signal to be inserted with the payload ID. This signal must connect to sdi_tx_line_f1 input signal on TX/Du top.

Pattern Generator Control Module Signals			
avmm_clk_in_clk	Input	1	Clock signal to Avalon-MM interface.
tx_clkout_in_clk	Input	1	Clock signal to Parallel I/O (PIO) IP. This clock must share the same clock as video pattern generator.
avmm_clk_reset_n	Input	1	Reset signal to Avalon-MM interface.
pattgen_rst_reset_in0	Input	1	Input reset signals to a reset synchronizer which synchronizes the reset to the tx_clkout_in_clk clock domain.
pattgen_rst_reset_in1	Input	1	
pattgen_rst_reset_out	Input	1	Output reset from the reset synchronizer. This reset synchronizes to the tx_clkout_in_clk clock domain and connects to the video pattern generator's input reset.
pattgen_ctrl_pio_out_port	Output	12	Output control signal from PIO to control the video pattern generator.

2.6. Video Pattern Generator Parameters

Customize the video pattern generator parameters according to your design.

Table 16. Video Pattern Generator Parameters

Parameter	Valid Value	Default Value	Description
OUTW_MULTP	1, 4	1	Defines the width of the output ports. Select 4 for a multi-rate design, otherwise select 1.
SD_BIT_WIDTH	10, 20	10	Defines the generated SD interface bit width. This value must match with the SD interface bit width parameter of the SDI II TX core in the same design.
TEST_GEN_ANC	0, 1	0	Select 1 to generate the ancillary data packet in output stream. The module inserts the embedded Data ID (DID) packet with 10'h242 if TEST_GEN_VPID is not enabled.
TEST_GEN_VPID	0, 1	0	Select 1 to generate the payload ID packet in output streams. The module inserts the embedded Data ID (DID) packet with 10'h241.

2.7. Hardware Setup

To run the hardware test for parallel loopback designs, connect an SDI video generator to the receiver input pin.

- Connect an external video analyzer to the TX instance to verify full functionality.
- To validate if the RX core locks to the signal and receives the video data correctly, use the on-board LEDs that display the RX status.

To run the hardware test for serial loopback designs, connect the transmitter output pin directly to the receiver input pin.

- To validate if the RX core locks to the signal and receives the video data correctly, use the on-board LEDs that display the RX status.
- You may also connect an SDI signal analyzer to the transmitter output pin to view the generated image.

Figure 16. Intel Arria 10 Development Board User LEDs

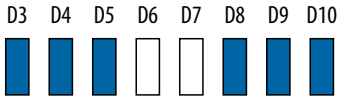


Table 17. On-board User LED Functions

LEDs	Function
D3–D5	Indicates the receiver video standard: • 000: SD-SDI • 001: HD-SDI • 010: 3G Level B • 011: 3G Level A • 100: 6G 8 Streams Interleaved

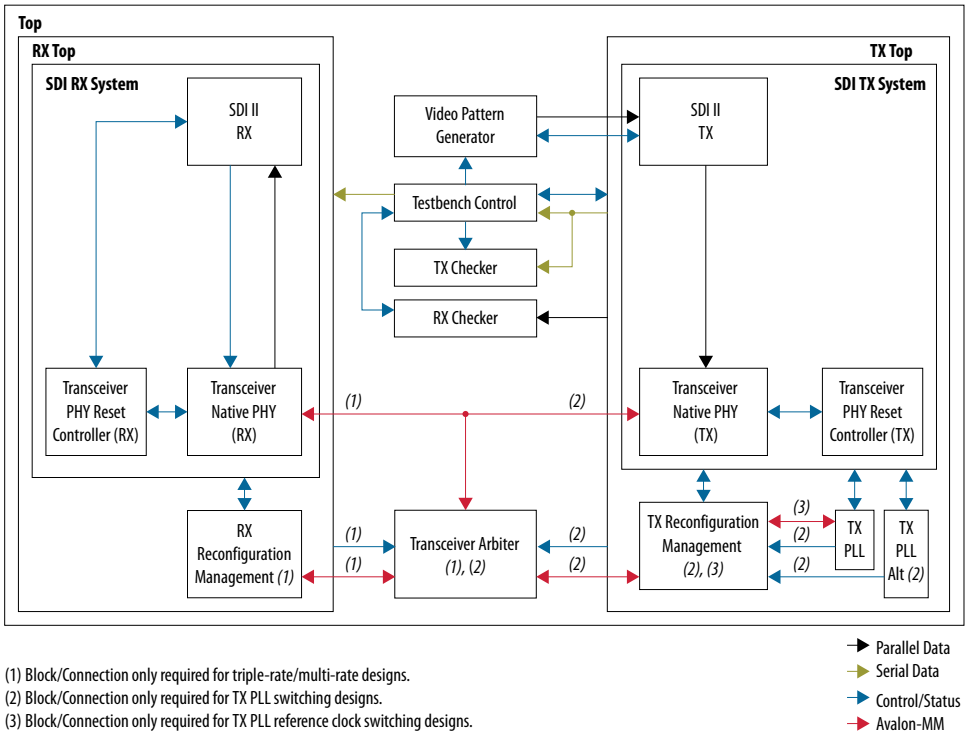
continued...

LEDs	Function
	• 101: 6G 4 Streams Interleaved • 110: 12G 16 Streams Interleaved • 111: 12G 8 Streams Interleaved
D6	Shows the slower version of the TX transceiver parallel clock
D7	Shows the slower version of the RX transceiver parallel clock
D8	Illuminates when `align_locked` asserts
D9	Illuminates when `trs_locked` asserts
D10	Illuminates when `frame_locked` asserts

2.8. Simulation Testbench

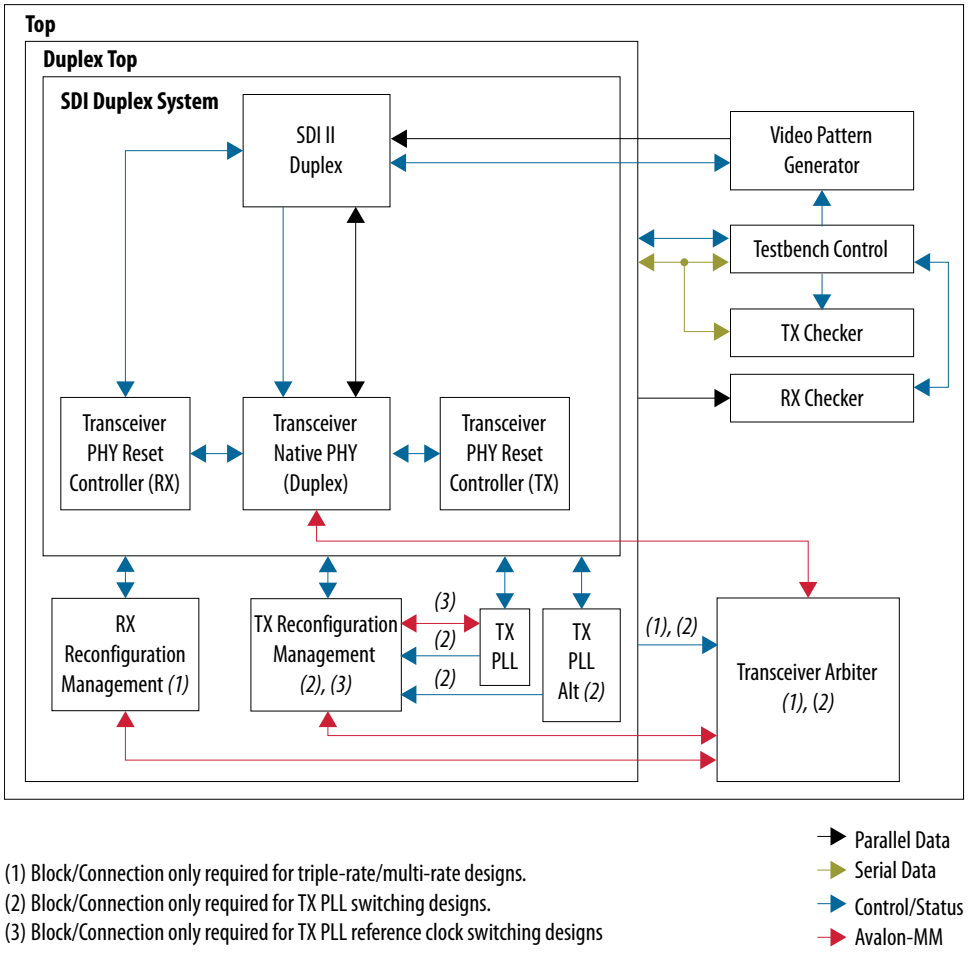
The simulation testbench checks for the assertion of the `trs_locked` signal. The testbench also detects the number of transceiver reconfiguration triggered after every video standard switching.

Figure 17. Simplex Mode Simulation Testbench Block Diagram



(1) Block/Connection only required for triple-rate/multi-rate designs.
(2) Block/Connection only required for TX PLL switching designs.
(3) Block/Connection only required for TX PLL reference clock switching designs.

Figure 18. Duplex Mode Simulation Testbench Block Diagram

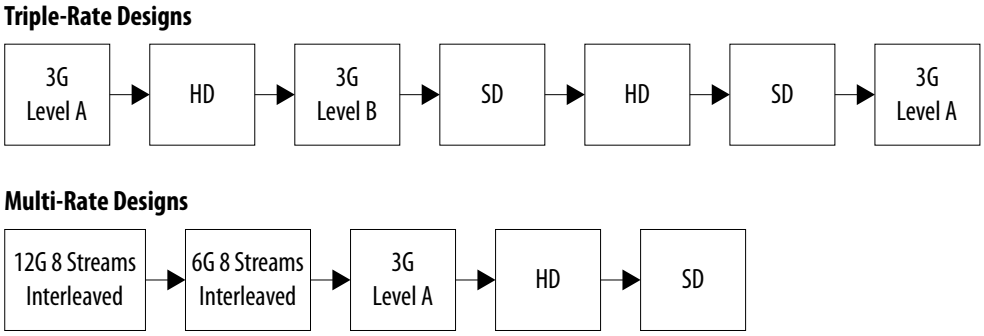


(1) Block/Connection only required for triple-rate/multi-rate designs.
(2) Block/Connection only required for TX PLL switching designs.
(3) Block/Connection only required for TX PLL reference clock switching designs

Table 18. Testbench Components

Component	Description
Testbench Control	This block controls the test sequence of the simulation and generates the necessary stimulus signals to the TX and video pattern generator blocks.
RX Checker	This checker detects the `trs_locked` signal from the RX protocol and compares the actual number of transceiver reconfigurations performed versus the expected number.
TX Checker	This checker verifies if the TX serial data contains a valid TRS signal.

Figure 19. Sequence of Video Standards for Triple-Rate and Multi-Rate Designs



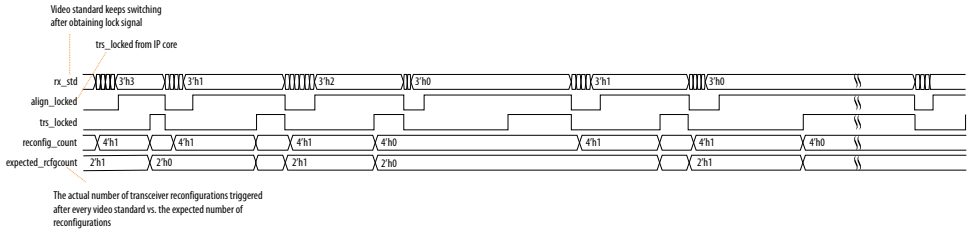
For single-rate designs, only one video standard is tested:

- HD-SDI single-rate—HD
- 3G-SDI single-rate—3G Level A

If you enable the **Dynamic Tx Clock Switching** parameter, only one video standard is being tested with 2 different TX PHY reference clocks to demonstrate the switching:

- HD-SDI single-rate—HD
- 3G-SDI single-rate/triple-rate—3G Level A
- Multi-rate—12G 8 streams interleaved

Figure 20. Simulation Waveform



A successful simulation ends with the following message:

```
#### TRANSMIT TEST COMPLETED SUCCESSFULLY! ####
#
#### Channel 1: RECEIVE TEST COMPLETED SUCCESSFULLY! ####
```

2.9. Upgrading Your Design

When you upgrade your designs to a later version, you may have to add, remove, or edit some of the generated files.

Upgrading from Intel Quartus Prime Pro Edition Version 17.1, 18.0, or 18.0 Update 1 to 18.1

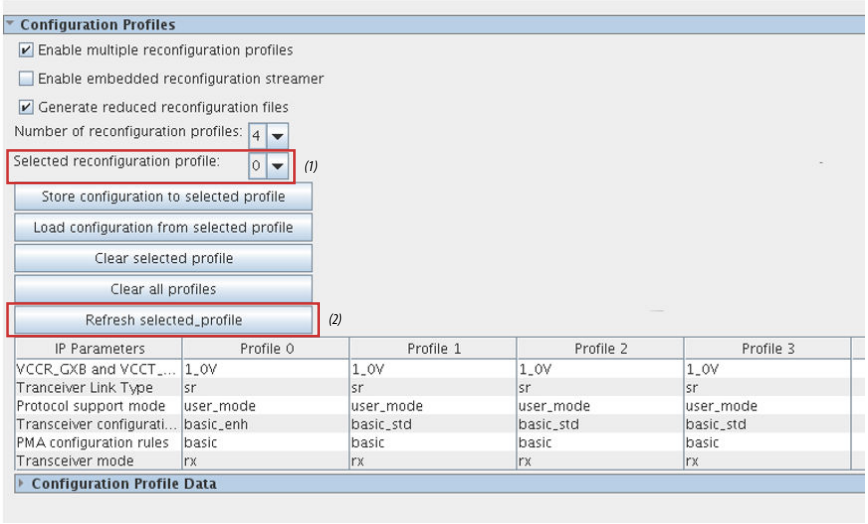
1. Click **IP Upgrade** to upgrade all the IP and Platform Designer files.
2. If you have triple-rate or multi-rate designs, update all the Native PHY config files location to the latest version in the simulation run script. For example, in the `mentor.do` file, update the version as in the following line:

```
vlog -sv \$USER_DEFINED_VERILOG_COMPILE_OPTIONS  "\$QSYS_SIMDIR/../rtl/du/
sdi_rx_phy/altera_xcvr_native_a10_181/
sim/reconfig/altera_xcvr_native_a10_reconfig_parameters_CFG0.sv"
```

   Note: You should be able to find the updated Native PHY library path in the specified folder indicated in the line.
3. Generate the same design example configuration in the new Intel Quartus Prime release version.
4. Compare the whole design example directory; replace the files that have changes with the new files and copy over the new files to your existing design.

Upgrading from Intel Quartus Prime Standard Edition Version 18.0 to Intel Quartus Prime Pro Edition Version 18.1

1. Click **IP Upgrade** to upgrade all the IP files.
2. Manually update the Platform Designer system files from the Platform Designer.
3. Refresh all the Native PHY profiles in the triple-rate or multi-rate RX Native PHY transceiver.
   - Open the `sdi_rx_phy.qsys` and `sdi_rx_sys.qsys` or `sdi_du_sys.qsys` file.
   - At the **Selected reconfiguration profile** parameter, select **0**, and click **Refresh selected profile**. Refer to notes *(1)* and *(2)* in the diagram below.



   - Repeat the previous step for the rest of the profiles accordingly.
   - After you have refreshed all the profiles, switch back to profile 0 and click **Generate HDL**.

4. If you have multi-rate or triple-rate designs, update all the Native PHY config files location to the latest version in the simulation run script. For example, in the `mentor.do` file, update the version as in the following line:

```
vlog -sv \$USER_DEFINED_VERILOG_COMPILE_OPTIONS  "\$QSYS_SIMDIR/../rtl/du/
sdi_rx_phy/altera_xcvr_native_a10_181/
sim/reconfig/altera_xcvr_native_a10_reconfig_parameters_CFG0.sv"
```

   Note: You should be able to find the updated Native PHY library path in the specified folder indicated in the line.

5. Generate the same design example configuration in the new Intel Quartus Prime Pro Edition release version.
6. Compare the whole design example directory; replace the files that have changes with the new files and copy over the new files to your existing design.
7. Update the project QSF file to remove the library switches and modify the SDC_FILE assignment to SDC_ENTITY_FILE assignment for some `.sdc` files.
   - Remove the following Intel Quartus Prime Standard Edition assignments:

```
Set_global_assignment -name VERILOG_FILE ../rtl/loopback/fifo/
sdi_ii_ed_loopback.v
-library sdi_du_sys_sdi_ii_rx_phy_mgmt_180
```

```
Set_global_assignment -name SDC_FILE ../rtl/loopback/fifo/
sdi_ii_ed_loopback.sdc
```

```
Set_global_assignment -name VERILOG_FILE ../rtl/du/du_top.v
-library sdi_du_sys_sdi_ii_rx_phy_mgmt_180
```

   - Add the following Intel Quartus Prime Pro Edition assignments:

```
Set_global_assignment -name VERILOG_FILE ../rtl/loopback/fifo/
sdi_ii_ed_loopback.v
```

```
Set_global_assignment -name SDC_ENTITY_FILE ../rtl/loopback/fifo/
sdi_ii_ed_loopback.sdc
-entity sdi_ii_ed_loopback
```

```
Set_global_assignment -name VERILOG_FILE ../rtl/du/du_top.v
```

3. SDI II Intel Arria 10 FPGA IP Design Example 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
17.0	Intel Arria 10 SDI II IP Core Design Example User Guide
16.1	SDI II IP Core Design Example User Guide

4. Revision History for SDI II Intel Arria 10 FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	Changes
2018.11.20	18.1	• Updated the *Directory Structure* section with new folders and files for loopback design and simulation: — `rcfg_pll_frac.v` — `modelsim_files.tcl` — `ncsim_files.tcl` — `riviera_files.tcl` — `vcs_files.tcl` — `vcsmx_files.tcl` — `xcelium_files.tcl` — `tb_ln_check.v` — `cds.lib` — `hdl.var` — `xcelium_setup.sh` — `xcelium_sim.sh` • Added a note that fPLL is only available when you select the **Parallel loopback without external VCXO** design. • Added information that the multi-rate designs support `rx_coreclk` frequency of 297 MHz. • Added a step in the *Compiling and Testing the Design* section to set the frequency for CLK1 in the **Si5338 (U14)** tab of the Clock Control GUI to **297 MHz** if you set the **Rx core clock (rx_coreclk) Frequency** parameter to 297.0/296.70 MHz. • Added instructions to run simulation using the Xcelium Parallel Simulator in the *Simulating the Design* section. • Edited the *Hardware and Software Requirements* section to include the Xcelium Parallel simulator. • Added a 625-MHz dedicated transceiver clock signal (`refclk_fmca_p`) in the *Interface Signals* section. • Added *Upgrading Your Design* section to provide guidelines about upgrading your existing designs to the latest version. • Added information about the simplex receiver channel calibration issue and a KDB link about the issue in the *Design Considerations* section.

Date	Version	Changes
May 2017	2017.05.08	• Rebranded as Intel. • Changed the part number. • Updated information about the parallel design examples and added new information about the serial design example.

continued...

Date	Version	Changes
		• Added files designated for Intel Quartus Prime Pro Edition. • Added information about video pattern generator interface signals and parameters. • Added link to archived version of the *Arria 10 SDI II IP Core Design Example User Guide*.
October 2016	2016.10.31	Initial release.