Serial Digital Interface Reference Design for Arria II GX Devices

The Serial Digital Interface (SDI) reference design shows how you can transmit and receive video data using the Altera® SDI MegaCore® function and the Arria® II GX video development board. This reference design uses three instances of the SDI MegaCore function. The triple standard SDI MegaCore function comprises of a standard definition (SD-SDI), high definition (HD-SDI), and a 3 gigabits per second (3G-SDI) standards.

This application note describes how the serial digital interface is used with the Arria II GX video development board for different variants. The Arria II GX video development board consists of the Arria II GX FPGA board that is available in the Arria II GX FPGA development kit and an SDI high-speed mezzanine card (HSMC) that is purchased separately.

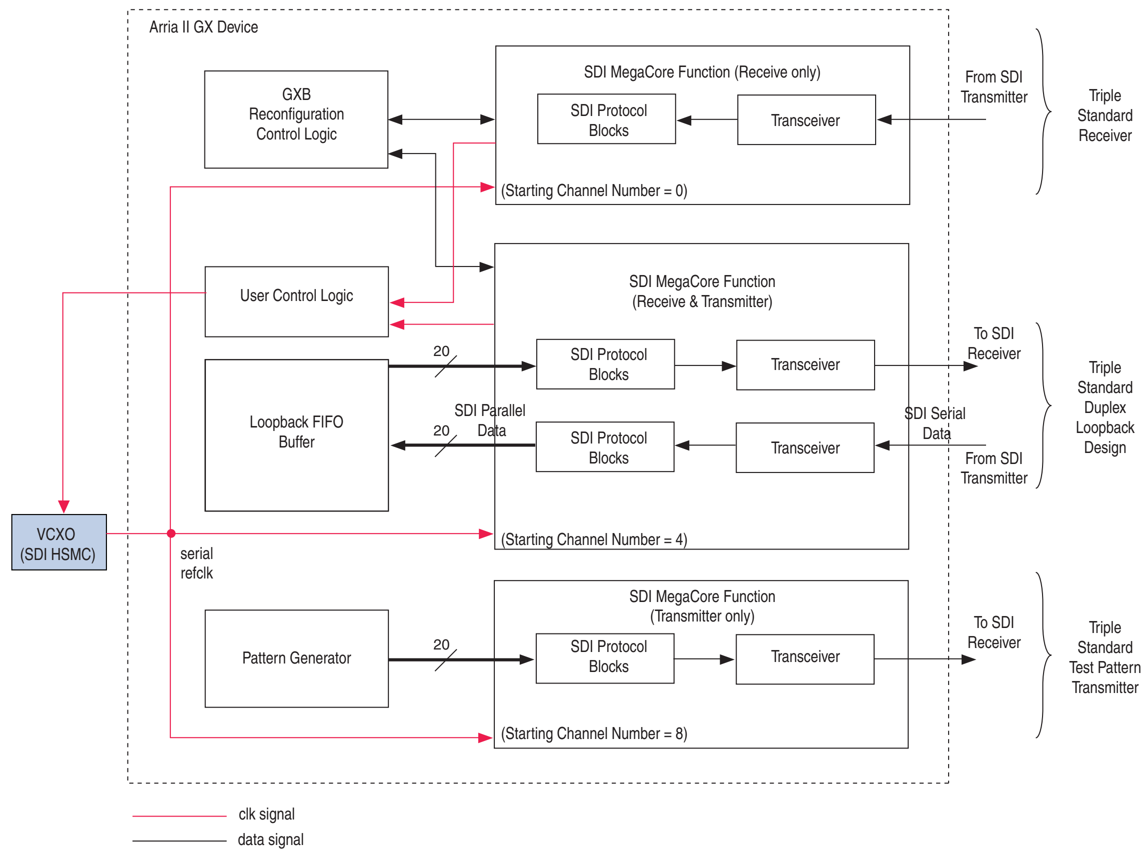
For more information about the Arria II GX FPGA development kit, refer to the *Arria II GX FPGA Development Kit User Guide*. For more information about the Arria II GX FPGA board, refer to the *Arria II GX FPGA Development Board Reference Manual*; for more information about the SDI HSMC, refer to the *SDI HSMC Reference Manual*. For more information about the SDI MegaCore function, refer to the *Serial Digital Interface MegaCore Function User Guide* or contact your Altera representative.

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Functional Description

The reference design provides a general platform to control, test, and monitor different speeds of the SDI operations. Figure 1 shows a high level block diagram of the SDI reference design.

Figure 1. Block Diagram



The following sections describe the various elements of the block diagram.

Triple-Standard Receiver

The triple-standard SDI receiver MegaCore function provides an SD-SDI, HD-SDI, and 3G-SDI receiver interface.

Triple-Standard Transmitter

The triple-standard SDI transmitter MegaCore function outputs a 2.970-Gbps 1080p, 1.485-Gbps 1080i, or 270-Mbps data stream. The transmitter takes its input from the pattern generator.

Triple-Standard Duplex Loopback

The triple standard-SDI duplex MegaCore function provides a full duplex, SD-SDI, HD-SDI, and 3G-SDI, and demonstrates receiver-to-transmitter loopback. The received data is decoded, buffered, recoded, and then transmitted. The interface is configured for 2.970-Gbps, 1.485-Gbps or 270-Mbps rates.

Loopback FIFO Buffer

The decoded receiver data is connected to the transmitter input through a FIFO buffer. When the receiver is locked, the receiver data is written to the FIFO buffer. When the FIFO buffer is half full, the transmitter starts reading, encoding, and transmitting the data.

Pattern Generator

The pattern generator IP core outputs a 2.970-Gbps 1080p, 1.485-Gbps 1080i or 270-Mbps test pattern. The test pattern can be a 100% color bar, a 75% amplitude color bar, or an SDI pathological checkfield frame.

GXB Reconfiguration Control Logic

The reconfiguration control logic handles the reconfiguration of the receiver part of the duplex core and the separate receiver in the design.

The reconfiguration control logic comprises of the following sub blocks:

- sdi_tr_reconfig_multi

  This top-level design contains the arbitration logic for up to four receiver ports. This block also has a state machine to control the ALTGX_RECONFIG megafunction.

- altgx_reconfig

  This block is an ALTGX_RECONFIG instance that is required for the dynamic partial reconfigurable I/O (DPRIO). Only this ALTGX_RECONFIG instance can be used to reprogram the ALTGX transceivers.

- ROMs

  The ROMs hold the ALTGX setting information for each of the video standards. Four ROMs are included, which allows a maximum of four channels to be reconfigured.

- Sdi_mif_intercept

  This block intercepts the read data from the ROMs. If reprogramming to HD is requested, this block modifies the data out of the ROM before sending it to the ALTGX reconfiguration block. This block removes the need to have a ROM for the HD setup.

For more information about the ALTGX_RECONFIG instance, refer to the *Arria II GX Device Handbook*. For more information about DPRIO, refer to the DPRIO section in the *SDI MegaCore Function User Guide*, *AN587: DPRIO and Multiple Instances SDI Application*, and *AN558: Implementing Dynamic Reconfiguration in Arria II GX Devices*.

User Control Logic

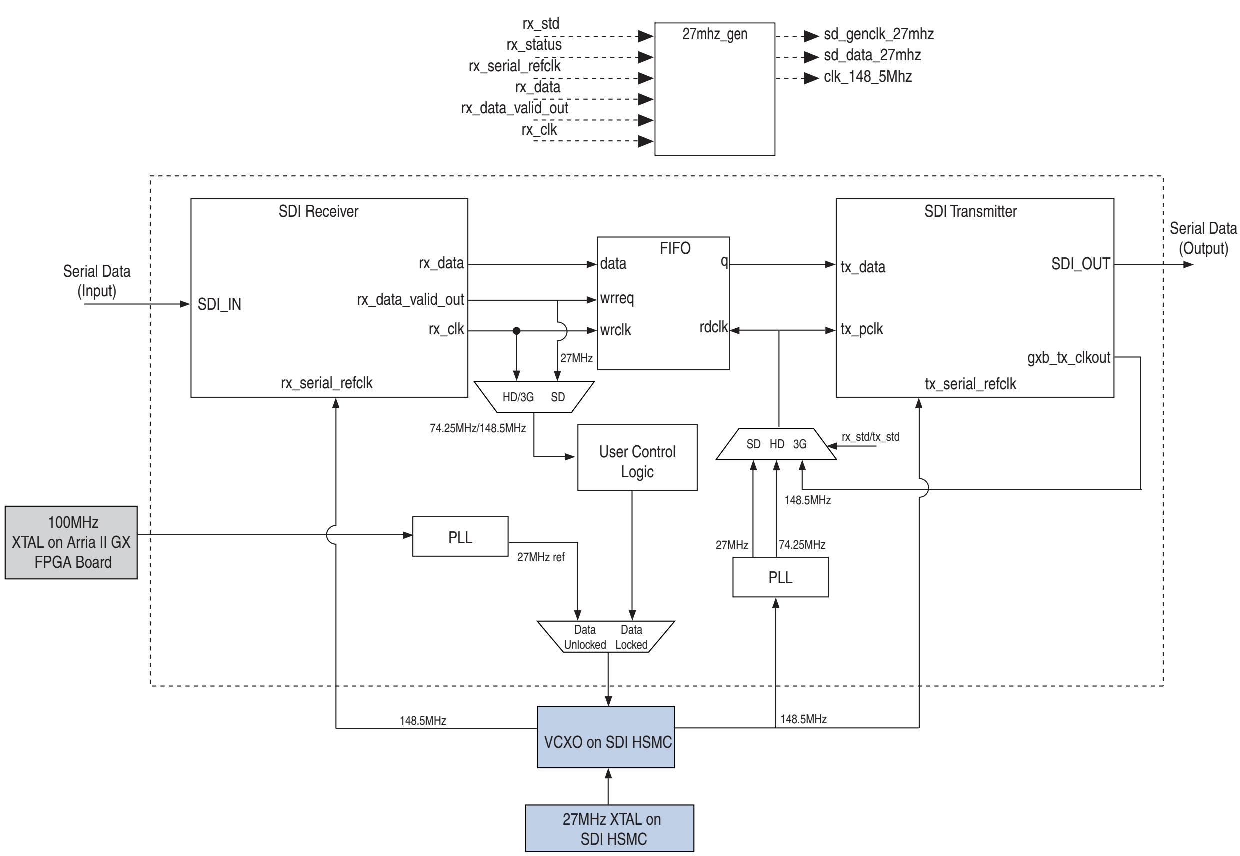
This user control logic receives the CDR receiver clock, `rx_clk`, from the SDI receiver only and the SDI duplex instances, and then sends the receiver clock with the control bits to the VCXO device.

Voltage Controlled Crystal Oscillator (VCXO)

The VCXO device is a phase-locked loop (PLL) based synchronous clock generator (ICS810001) that is located on the SDI HSMC. This device contains two internal frequency multiplication stages that are cascaded in series. The first stage is a VCXO PLL that is optimized to provide reference clock jitter attenuation and to support the complex PLL multiplication ratios needed for video rate conversion. The second stage is a FemtoClock™ frequency multiplier that provides the low jitter, high frequency video output clock. The 148.5-MHz VCXO output clock is connected to the `rx_serial_ref_clk` and `tx_serial_ref_clk` clocks of all the three SDI instances.

Figure 2 shows the block diagram for the duplex loopback FIFO design and the VXCO device.

Figure 2. Block Diagram for Duplex Loopback FIFO Design and VXCO Device



27mhz_gen

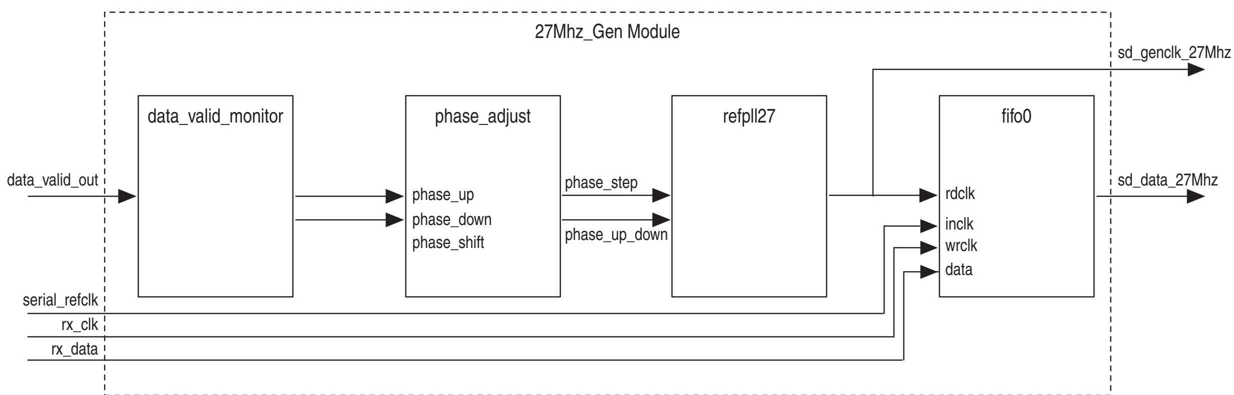
This module generates a 27-MHz parallel clock to receive the SD-SDI data. Use the `sd_genclk_27mhz` output clock to clock the `sd_data_27mhz` parallel data for SD-SDI.

The 27mhz_gen module consists of the following components:

- data_valid_monitor module—a user logic to control the pll
- phase_adjust module—module that controls the pll based `data_validout` signal
- `refpll27`—pll that generates `sd_genclock_27mhz` clock to clock the `sd_data_27mhz` data that comes from the FIFO buffer
- FIFO buffer

Figure 3 shows the block diagram of the 27mhz_gen module.

Figure 3. Block Diagram of the 27mhz_gen Module



Getting Started

This section discusses the requirements and related procedures to demonstrate the SDI reference design with the Stratix IV GX audio video development board. This section contains the following topics:

- Hardware and Software Requirements
- Obtaining the Design
- Hardware Setup
- Running the Reference Design
- Using the Reference Design

Hardware and Software Requirements

The demonstration requires the following hardware and software:

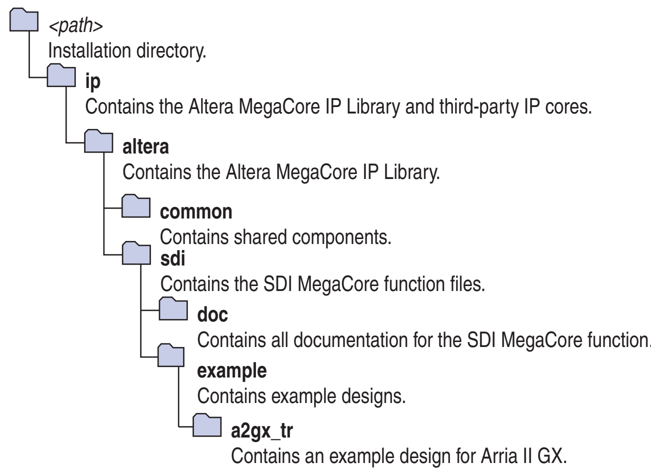
- Arria II GX video development board—Arria II GX FPGA development board and SDI HSMC
- SDI MegaCore function

- Quartus® II software, version 10.0 SP1

Obtaining the Design

Figure 4 shows the directory structure of the reference design.

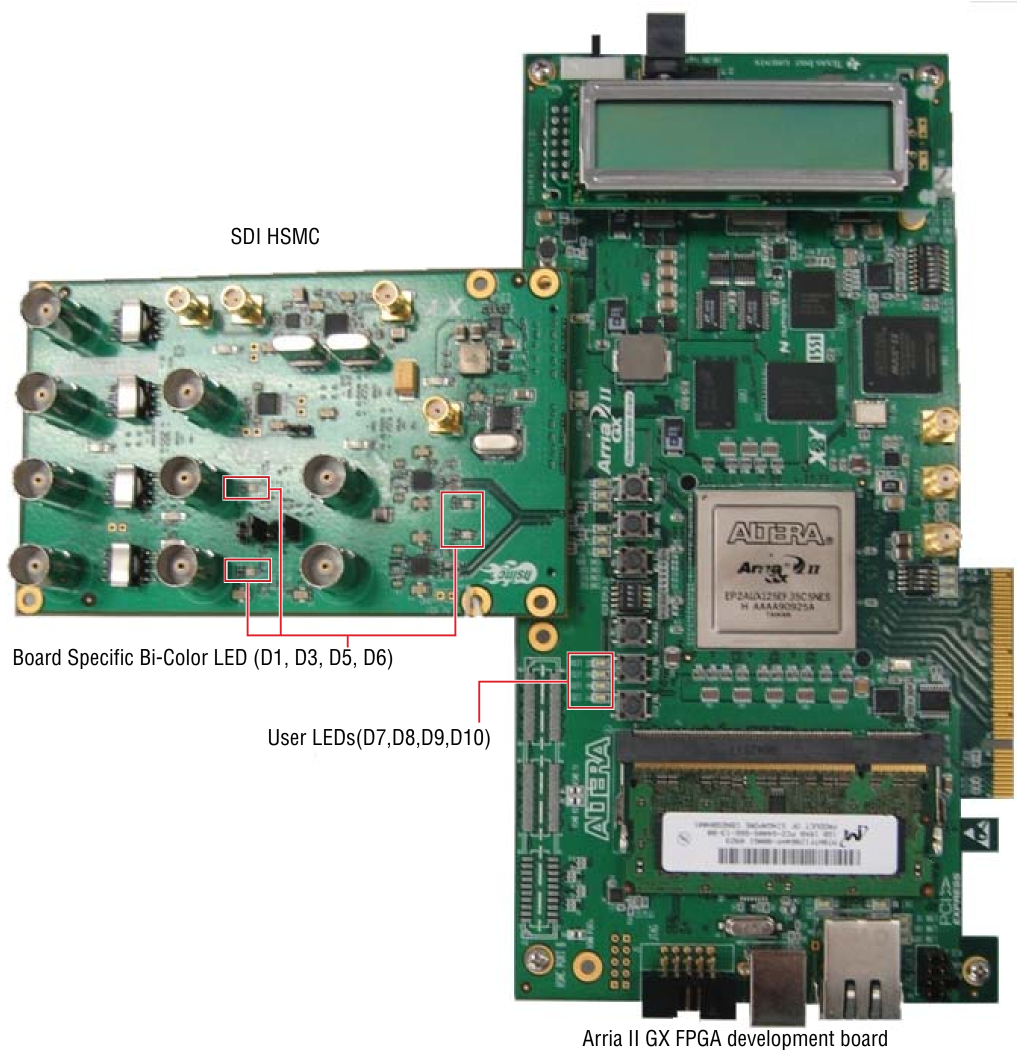
Figure 4. Directory Structure



Hardware Setup

Figure 5 shows how the Arria II GX FPGA development board is connected to the SDI HSMC.

Figure 5. Hardware Setup



For more information about the Arria II GX FPGA development board, refer to the *Arria II GX FPGA Development Board Reference Manual*; for more information about the SDI HSMC, refer to the *SDI HSMC Reference Manual*.

Table 1 describes the function of each LED on the Arria II GX FPGA development board and the corresponding dual in-line package (DIP) switch settings.

Table 1. LEDs on Arria II GX FPGA Development Board

<table>
<tr><th>DIP Switch Setting</th><th>LED</th><th>Description</th></tr>
<tr><td rowspan="4">USER_DIP[3:2] = 2'b00</td><td>D7</td><td>SDI IN 1 in reset</td></tr>
<tr><td>D8</td><td>SDI IN 1 frame lock</td></tr>
<tr><td>D9</td><td>SDI IN 1 TRS lock</td></tr>
<tr><td>D10</td><td>SDI IN 1 alignment lock</td></tr>
<tr><td rowspan="4">USER_DIP[3:2] = 2'b01</td><td>D7</td><td>SDI IN 2 in reset</td></tr>
<tr><td>D8</td><td>SDI IN 2 frame lock</td></tr>
<tr><td>D9</td><td>SDI IN 2 TRS lock</td></tr>
<tr><td>D10</td><td>SDI IN 2 alignment lock</td></tr>
<tr><td rowspan="4">USER_DIP[3:2] = 2'b10</td><td>D7</td><td rowspan="2">SDI IN 2 received signal standard
[D7, D8]: 00 = SD-SDI, 01 = HD-SDI, 11= 3G-SDI</td></tr>
<tr><td>D8</td></tr>
<tr><td>D9</td><td rowspan="2">SDI IN 1 received signal standard
[D9, D10]: 00 = SD-SDI, 01= HD-SDI,11 = 3G-SDI</td></tr>
<tr><td>D10</td></tr>
<tr><td rowspan="4">USER_DIP[3:2] = 2'b11</td><td>D7</td><td>Not used</td></tr>
<tr><td>D8</td><td>Not used</td></tr>
<tr><td>D9</td><td rowspan="2">Internal pattern generator signal standard
[D9, D10]: 00 = SD-SDI, 01 = HD-SDI, 11 = 3G-SDI</td></tr>
<tr><td>D10</td></tr>
</table>

Table 2 describes the function of each user-defined DIP switch control (SW2). When the switch is in the OFF position, a logic 1 is selected. When the switch is in the ON position, a logic 0 is selected.

Table 2. SW2 DIP Switch Controls

<table>
<tr><th>USER_DIP</th><th>Description</th></tr>
<tr><td>3</td><td rowspan="2">Control signal for the user LED display:
USER_DIP[3:2] = 00: LED displays <code>rx_p0_status</code>
USER_DIP[3:2] = 01: LED displays <code>rx_p1_status</code>
USER_DIP[3:2] = 10: LED displays <code>rx_p0_std</code> and <code>rx_p1_std</code>
USER_DIP[3:2] = 11: LED displays <code>tx_std</code></td></tr>
<tr><td>2</td></tr>
<tr><td>1</td><td rowspan="2">Change internal pattern generator signal standard
USER_DIP[1:0]: 00 = SD-SDI, 01 = HD-SDI, 11 = 3G-SDI</td></tr>
<tr><td>0</td></tr>
</table>

Table 3 describes the function of each board specific bi-color LED on the SDI HSMC.

Table 3. Board Specific Bi-Color LEDs on SDI HSMC

LED	Description
D1	SDI IN 2 receiving SDI signal in the following standards: ■ Green = 3G-SDI ■ Orange = HD-SDI ■ Red = SD-SDI
D3	SDI OUT 2 transmitting SDI signal in the following standards: ■ Green = 3G-SDI ■ Orange = HD-SDI ■ Red = SD-SDI
D5	SDI OUT 1 transmitting SDI signal in the following standards: ■ Green = 3G-SDI ■ Orange = HD-SDI ■ Red = SD-SDI
D6	SDI IN 1 receiving SDI signal in the following standards: ■ Green = 3G-SDI ■ Orange = HD-SDI ■ Red = SD-SDI

Table 4 describes the function of each push button.

Table 4. Push Buttons

Push Button	Description
PB1	Selects 100% color bar output (default color bar output is 75% color bar)
PB2	Selects a pathological SDI checkfield pattern
PB3	Resets the CPU or FPGA logic

Running the Reference Design

To run the reference design, perform the following steps:

1. Set up the board connections. With the power to the board off, perform the following steps:
   a. Connect the SDI HSMC to the FPGA development board. Refer to Figure 5 on page 7.
   b. Specify the following board settings located on the back of the FPGA development board:
      - DIP switch bank (SW4)
      - PCI Express DIP switch bank (SW3)
      - JTAG Chain Header Switch Controls (J9)

      Match the board settings to the switch control settings in Table 5.
   c. Connect the FPGA development board (J4) to the power supply.

Table 5. SW DIP Switch Control Settings

Switch	Schematic Signal Name	Description	Default
SW4			
1	MAX_DIP0	Reserved	OFF
2	MAX_DIP1	Reserved	OFF
3	MAX_DIP2	Reserved	OFF
4	MAX_DIP3	ON: Load user hardware page 1 from flash memory upon power-up OFF: Load factory design from flash memory upon power-up	OFF
5	LCD_PWRMON	ON: LCD driven from the MAX II EPM2210 System Controller (power monitor) OFF: Unused	OFF
6	USB_DISABLEn	ON: Embedded USB-Blaster disable OFF: Embedded USB-Blaster enable	OFF
7	CLK_ENABLE	ON: On-board oscillators enable OFF: On-board oscillators disable	ON
8	CLK_SEL	ON: 100 Mhz clock select OFF: SMA input clock select	ON
SW3			
1	PCIE_LED_x1	ON: Enable x1 presence detect OFF: Disable x1 presence detect	OFF
2	PCIE_LED_x4	ON: Enable x4 presence detect OFF: Disable x4 presence detect	OFF
3	PCIE_LED_x8	ON: Enable x8 presence detect OFF: Disable x8 presence detect	OFF
4	NC	Not used	OFF

Table 5. SW DIP Switch Control Settings

Switch	Schematic Signal Name	Description	Default
J9			
1	`MAX_JTAG_EN`	ON: Bypass MAX II CPLD EPM2210 System Controller OFF: MAX II CPLD EPM2210 System Controller in-chain	ON
2	`HSMA_JTAG_EN`	ON: Bypass HSMA OFF: HSMA in-chain	OFF
3	`HSMB_JTAG_EN`	ON: Bypass HSMB OFF: HSMB in-chain	ON
4	`PCIE_JTAG_EN`	ON: Bypass PCI Express OFF: Reserved	ON

2. Launch the Quartus II software and compile the reference design:
   a. On the File menu click **Open Project**, navigate to ****<*directory*>**\\a2gxsdi.qpf**, and click **Open**.
   b. On the Processing menu, click **Start Compilation**.
3. Download the Arria II GX**.sof** file:
   a. Connect the USB-Blaster™ download cable to the board's USB Type-B Connector (J6).
   b. On the Tools menu, click **Programmer.** The file is automatically detected by the software during compilation and it appears on the pop-up window**.** Click **Start** to download the Quartus II-generated file to the board. If the file does not appear in the pop-up window, click **Add File**, navigate to ****<*directory*>**\\a2gxsdi.sof**, and click **Open**.

This design is volatile and must be reloaded each time the board is powered on.

After you have set up the board in step 1, run the different variants described in the following sections.

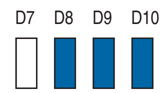
Parallel Loopback

To run the parallel loopback demonstration, perform the following steps:

1. Connect an SDI signal generator to the receiver input of `SDI IN 2` (BNC J2).
2. Connect an SDI signal analyzer to the transmitter output of `SDI OUT 2` (BNC J1).
3. Specify USER_DIP[3:2] = 2'b01. Refer to Table 2 on page 8.

4. The parallel loopback demonstration runs. The LEDs indicate the following conditions:
   - LED D10 illuminates when the receiver is word aligned at port 2.
   - LED D9 illuminates when the received line format is stable at port 2.
   - LED D8 illuminates when the receiver frame format is stable at port 2.
   - LED D7 illuminates when the `SDI IN 2` receiver is resetting.

Table 6. Condition of LEDs for Parallel Loopback Demonstration



Additionally, the LEDs on the SDI HSMC indicate the following conditions:

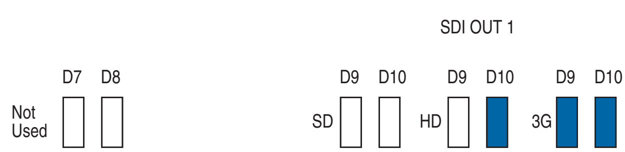
- LED D1 illuminates when the receiver signal standard is detected at port 2.
- LED D3 illuminates when the transmitter signal standard is detected at port 2.

Test Pattern Transmitter

To run the test pattern transmitter demonstration, perform the following steps:

1. Connect an SDI signal analyzer to the transmitter output `SDI OUT 1` (BNC J8).
2. Specify USER_DIP[3:2] = 2’b11. Refer to Table 2 on page 8.
3. The test pattern demonstration runs. The LEDs indicate the following conditions:
   - LED D8 and D7 are not used.
   - LED D10 and D9 indicate the internal pattern generator signal standard that transmits through port 1 in the transmitter. Refer to Table 7.

Table 7. Condition of LEDs for Test Pattern Transmitter Demonstration



Additionally, the LED D5 on the SDI HSMC illuminates when the transmitter signal standard is detected at port 1.

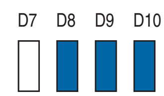
4. Check the result on the SDI signal analyzer.
5. The design has a default output of a 75% color bar test pattern. To change the test pattern, use the push buttons (PB1 and PB2) on the board. For more information about the push buttons, refer to Table 4 on page 9.

Receiver Only

To run the receiver only demonstration, perform the following steps:

1. Connect a SDI signal generator to the receiver input `SDI IN 1` (BNC J9).
2. Specify USER_DIP[3:2] = 2'b00. Refer to Table 2 on page 8.
3. The receiver demonstration runs. The LEDs indicate the following conditions:
   - LED D10 illuminates when the receiver is word aligned at port 1.
   - LED D9 illuminates when the received line format is stable at port 1.
   - LED D8 illuminates when the receiver frame format is stable at port 1.
   - LED D7 illuminates when the `SDI IN 1` receiver is resetting.

Table 8. Condition of LEDs for Receiver Only Demonstration



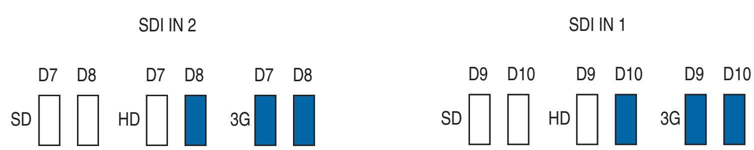
Additionally, the LED D6 on the SDI HSMC illuminates when the receiver signal standard is detected at port 1.

Check for Received Signal Standard rx_p0_std and rx_p1_std

To check the received signal standard, perform the following steps:

1. Connect SDI input source to `SDI IN 1` or `SDI IN 2`.
2. Specify USER_DIP[3:2] = 2'b10. Refer to Table 2 on page 8. The LEDs indicate the following conditions:
   - LED D10 and D9 represent the received signal standard at `SDI IN 1`.
   - LED D8 and D7 represent the received signal standard at `SDI IN 2`.

Table 9. Condition of LEDs for Received Signal Standard rx_p0_std and rx_p1_std

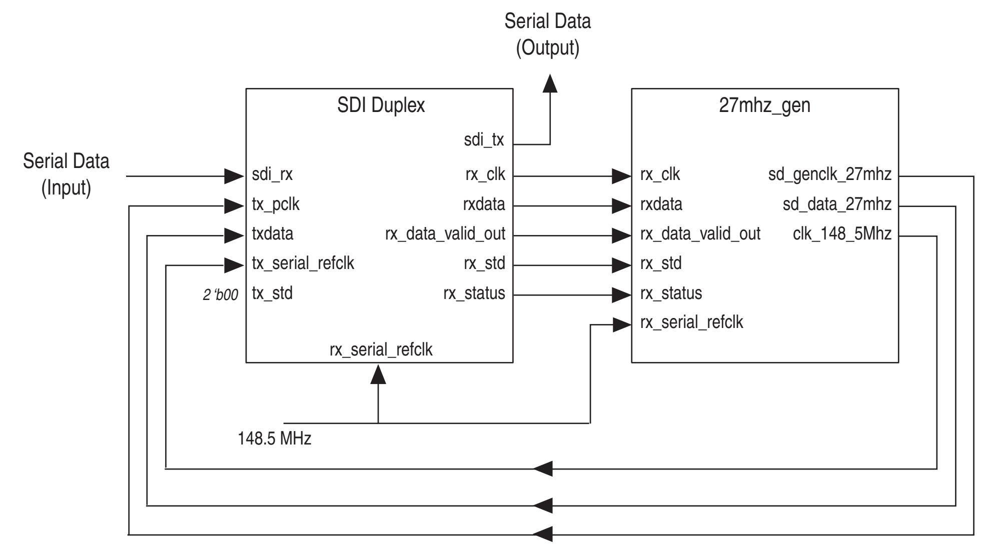


Using the Reference Design

If you are using the SD-SDI standard, use the reference design with the 27mhz_gen module to generate the 27-MHz clock to receive the SD-SDI data.

Figure 6 shows how to use the 27mhz_gen module to generate a 27-MHz clean clock to receive SD-SDI parallel data. The 27-MHz clock and the SD-SDI parallel data from the 27mhz_gen module connects to the transmitter of SDI duplex instance, and transmits to a third party for monitoring.

Figure 6. Using 27mhz_gen Module with the Reference Design



If you are using the SD-SDI standard, type the following code to control the `GENERATE_SD_27MHZ_CLK` parameter:

```
GENERATE_SD_27MHZ_CLK =1'b1
```

If you are using a regular SDI operation, type the following code to control the `GENERATE_SD_27MHZ_CLK` parameter:

```
GENERATE_SD_27MHZ_CLK =1'b0
```

When compiling for a regular SDI operation, remove the back slash from the following line:

```
//define clk_148_p
```

Conclusion

This application note provides ways to use the SDI reference design with the Arria II GX FPGA board and SDI HSMC. You can use the different variants discussed to evaluate the SDI MegaCore function for integration into Altera FPGA designs.

Document Revision History

Table 10 shows the revision history for this application note.

Table 10. Document Revision History

Date	Version	Changes
December 2010	1.3	■ Added information about the 27mhz_gen module. ■ Updated the design files.
May 2010	1.2	Updated Figure 2 on page 4.
February 2010	1.1	Updated Figure 2 on page 4.
December 2009	1.0	Initial release.