Serial Digital Interface Reference Design for Cyclone & Stratix Devices

August 2004, ver 1.1 Application Note

Introduction

The Society of Motion Picture and Television Engineers (SMPTE) have defined a serial digital interface (SDI) that video system designers widely use as an interconnect between equipment in video production facilities.

The SDI reference designs for Altera® Cyclone™ and Stratix® devices provide a 270 megabits per second (Mbps) SDI, as defined by *SMPTE259M-1997 10-Bit 4:2:2 Component Serial Digital Interface.*

This application note describes the following topics:

- The reference design elements
- How to simulate the reference design (with the ModelSim-Altera simulator)
- How to compile the reference design (with the Quartus® II software)
- How to demonstrate SDI with the Cyclone video demonstration board and the Stratix GX video demonstration board.

For more information on the Cyclone video demonstration board, refer to the Cyclone video demonstration board data sheet.

For more information on the Stratix GX video demonstration board, refer to the Stratix GX video demonstration board data sheet.

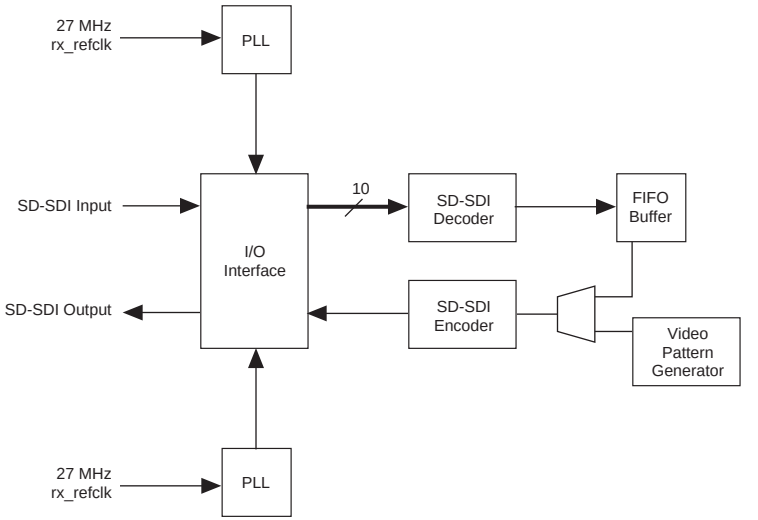
Altera supplies the reference designs as Verilog HDL source code. The reference designs include testbenches that allows you to test the Verilog HDL source code.

The purpose of these reference designs is to demonstrate to video system designers how to implement SDI using Cyclone or Stratix devices.

Functional Description

Figure 1 shows the block diagram.

Figure 1. Block Diagram



The reference design consists of the following elements:

- Stratix or Cyclone low-voltage differential signaling (LVDS) inputs and outputs (I/Os) for the receiver and transmitter
- I/O interface
- Decoder
- Encoder
- Video pattern generator
- Two PLLs for frequency multiplication: one for the transmitter, one for the receiver

The two reference clocks can be either asynchronous to each other or the two PLLs can be driven from the same clock source. The design uses two PLLs. The first PLL generates the following clocks needed for the receiver:

- For the 5× oversample scheme a 337.5-MHz clock (27 MHz × 50/4) and a 337.5-MHz clock phase shifted by 90°
- For the 3× oversample scheme a 207.5-MHz clock (27 MHz ×30/ 4) and a 207.5-MHz clock phase shifted by 90°

The second PLL generates a transmitter serial clock at 270 MHz.

General purpose LVDS I/Os are used for the serial transmitter and receiver interfaces as Cyclone and Stratix devices do not feature any dedicated SERDES ports. Serializing, deserializing, and data recovery are all achieved using logic elements in the FPGA.

If "pattern" mode is set, the transmit and receive paths are independent, i.e., the design can transmit and receive at the same time. If the "loop-back" mode is set, the transmit path is coupled to the receive path through a rate matching first-in first-out (FIFO) buffer.

In loop-back mode, the transmit side reference clock needs to come from an external source genlocked to the house synchronization or from a "cleaned-up" version of the recovered data valid flag. You must frequency lock the transmit clocks to the incoming data to avoid FIFO buffer underflow or overflow.

This section describes the following reference design elements:

- I/O Interface Logic
- Encoder
- Decoder
- Video Pattern Generator

I/O Interface Logic

The I/O interface logic comprises the following functions:

- Deserializer
- Frequency Divider
- Oversampling Interface

The deserializer and oversampling interface are part of the oversampling scheme that recovers data from a self-clocked serial data stream. The oversampling scheme is required because Cyclone and Stratix devices do not feature any dedicated serializer/deserializer (SERDES) or clock data recovery (CDR). The Altera soft data recovery solution for SD-SDI recovers the correct data by first converting the serial data into a parallel format and then extracting the correct bits based on transitions detected in these parallel words. Figure 2 shows the 5× oversampling scheme.

Figure 2. Oversampling Scheme

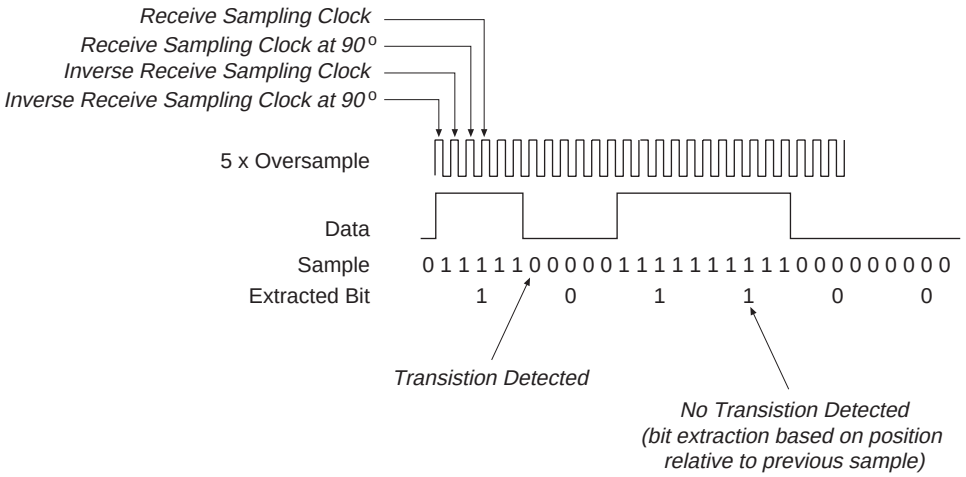
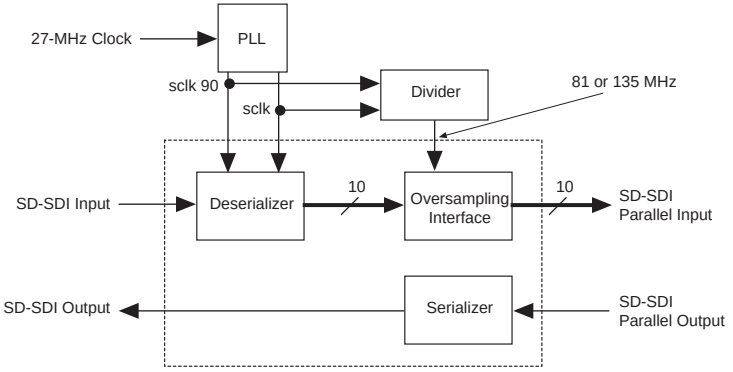


Figure 3 shows the I/O interface logic block diagram.

Figure 3. I/O Interface Logic



Deserializer

The serial data stream from the LVDS input buffer is sampled using four different clocks phase-shifted by 90° from each other. Internally, two out of these four clocks are created from an on-chip PLL; the other two are created by inverting the PLL clock outputs.

Samples are then all converted to the same clock domain and deserialized into a 10-bit parallel word. The serial clock that samples the bit stream depends on the oversample scheme:

- For the 5× oversample scheme, the serial clock has to be 5/4 of the incoming bit (i.e., 270-bit rate × 5/4 × 4 sample per clock = 1,350 Mbps)
- For the 3× oversample scheme, the serial clock has to be 3/4 of the incoming bit rate (i.e., 270-bit rate × 3/4 × 4 sample per clock = 810 Mbps)

The parallel clock that extracts data from the deserializer is running at 81 MHz for a 3× oversampling scheme and 135 MHz for a 5× oversampling scheme.

Frequency Divider

The parallel clock (81 or 135 MHz) is derived from the serial sampling clock by a soft fequency divider to minimize PLL clock output usage. Edge counters on the 5/4 or 3/4 serial clock and its 90° phase shift create a frequency dividing ratio of 2.5.

Oversampling Interface

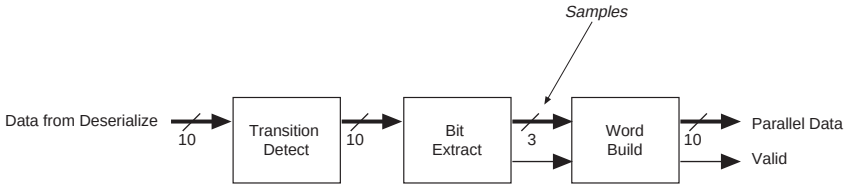
A 3× or 5× oversampling scheme implements data recovery and bit synchronization, which corresponds to a sampling rate of 810 or 1,350 Mbps, respectively.

The deserializer provides a fixed frequency sampling of the serial data. Approximately 3 or 5 samples are taken for each bit. These samples are accumulated by the deserializer and passed to the oversampling interface in a parallel format. Logic extracts the data from the sets of samples generated by the deserializer.

Firstly, the transition points within the received word are determined. These transition points determine the best sample to extract for each data bit. The logic continuously realigns to the transition points in the incoming data, and can therefore adapt to a frequency mismatch between the sampling clock and the incoming data rate. The extracted samples for each data bit are accumulated into a parallel word for processing by the SDI decoder.

Figure 4 shows the oversampling interface.

Figure 4. Oversampling Interface



Encoder

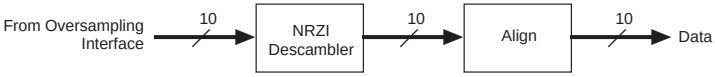
The encoder provides the channel coding functions. Figure 5 shows the SD encoder. For SD, the 10-bit data input is interleaved chroma and luma.

Figure 5. Encoder—SD

Decoder

The decoder provides the channel decoding functions. It also ensures that the received parallel data is word aligned. Figure 6 shows the decoder.

Figure 6. Decoder



The decoder contains the descrambling and NRZI decoding algorithm.

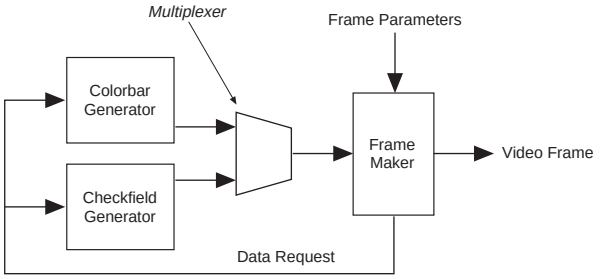
Word alignment is achieved by identifying the unique start of active video (SAV) and EAV tokens within the received data. Once the same bit alignment has been seen for three consecutive tokens, the alignment is locked. If two consecutive tokens are subsequently seen in a different bit position, the alignment is adjusted. The design can be forced to adjust immediately during the serial switch interval.

The 10-bit parallel output provides the interleaved chroma and luma.

Video Pattern Generator

For demonstration and verification, Altera provides a basic video pattern generator, which creates standard colorbar patterns or an SDI checkfield. Figure 7 shows the video pattern generator.

Figure 7. Video Pattern Generator



Testbench

The `tb_sdsdi_rxtx` testbench instantiates two SD-SDI reference designs: one for transmitting; one for receiving.

Colorbar video patterns are sent from the instantiation of the SDI transmitter that is connected up to the instantiation of the SDI receiver. A random serial data delay generator is inserted on the way to mimic random jitter on the link. The transmitter and receiver are clocked with asynchronous clock sources—the frequencies differ by 200 ppm, which maximizes the stress that the SDI receiver sees.

During a successful simulation, the `align_locked` flag goes active after three consecutive TRSs. The testbench automatically report a successful simulation by displaying the following message:

```
# -- Aligner : 3 consecutive TRS seen with same alignment
# -- Aligner : Alignment set to 1/ 0
#
# **** TEST COMPLETED SUCCESSFULLY! ****
```

Getting Started

This section involves the following steps:

- System Requirements
- Install the Design
- Simulate using the Testbench
- Set the Quartus II Constraints
- Compile the Design

- Use the Demonstrations

System Requirements

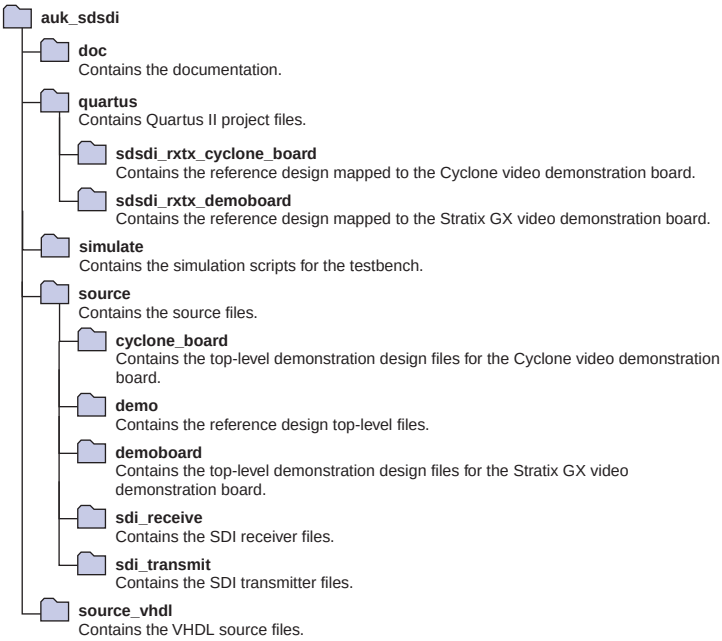
The demonstrations require the following hardware: and software

- A PC running the Windows 98/2000/XP operating system
- Quartus II version 4.0
- ModelSim-Altera version 5.7e

Install the Design

Figure 8 shows the directory structure.

Figure 8. Directory Structure



Simulate using the Testbench

The reference design includes a testbench, **tb_sdsdi_rxtx**, and a script for simulating with the ModelSim-Altera simulator. To use the testbench, follow these steps:

1. Modify the simulation batch file, **run.bat**, to point to your installation of the ModelSim-Altera simulator, by editing line 4 of the file.

2. Execute **run.bat** in the **\simulate** directory.

The waveform viewer displays a selection of signals. The simulation runs automatically and provides a pass or fail indication on completion.

Set the Quartus II Constraints

For this reference design to work reliably, you must implement the following Quartus II constraints:

- Minimize the timing skew among the paths from I/O pins to the four sampling registers
- Set the oversampling clock that is used by the oversampling interface (for 3× set to 81 MHz; or for 5× set to 135 MHz) as an independent clock domain
- To facilitate timing closure of the receiver design's front-end, constrain logic within a small region
- Choose the oversampling ratio

The reference design that is mapped to a Cyclone video demonstration board in the **quartus\sdsdi_rxtx_cyclone_board** directory contains an example of the Quartus II constraints.

Minimize Timing Skew

You should minimize the timing skew among the paths from I/O pins to the four sampling registers (`sample_a[0]`, `sample_b[0]`, `sample_c[0]`, and `sample_d[0]`). To minimize the timing skew, manually place the sampling registers close to each other and to the serial input pin. Because these four registers are using four different clock domains, place two of the four registers in one LAB and the other two in another LAB. Furthermore, place the 2 chosen LABs within the same row whatever the placement of the serial input. Finally, do not place the four sampling registers at the immediate rows or columns next to the I/O, but the second one next to the I/O bank. This location is because inter-LAB interconnects between I/O banks and their immediate rows or columns are much faster than core interconnect.

The following code is an example of a constraint, which you can set using the Quartus II Assignment Editor:

```
set_location_assignment LC_X32_Y4_N0 -to
"sdsdi_rxtx:u_demo\|sdsdi_io_interface:u_io_interface\|s2p:u_s2p\|sample_a\[0\]"
```

```
set_location_assignment LC_X33_Y4_N0 -to
"sdsdi_rxtx:u_demo\|sdsdi_io_interface:u_io_interface\|s2p:u_s2p\|sample_b\[0\]"
set_location_assignment LC_X32_Y4_N1 -to
"sdsdi_rxtx:u_demo\|sdsdi_io_interface:u_io_interface\|s2p:u_s2p\|sample_c\[0\]"
set_location_assignment LC_X33_Y4_N1 -to
"sdsdi_rxtx:u_demo\|sdsdi_io_interface:u_io_interface\|s2p:u_s2p\|sample_d\[0\]"
```

Set Oversampling Clock

You should set the oversampling clock that the oversampling interface uses (3×: 81 MHz or 5×: 135 MHz) as an independent clock domain, so that timing analyzer generates an accurate timing calculation.

The following code is an example of a constraint, which you can set using the Quartus II Assignment Editor:

```
sdsdi_rxtx:u_demo|freq_trans:u_freq_trans|sclk_out~reg0 Clock Settings 135MHz Yes
```

Define the 135-MHz clock setting in the **Settings** > **Timing Settings** > **Clocks** (Assignment menu), before using it in the Assignment Editor.

Timing Closure

To help with timing closure of the receiver design's front end, constrain logic within a small region. A fixed LogicLock™ region of seven by three LABs located next to the serial input pin that contains the following logic helps timing closure:

```
"sdsdi_rxtx:u_demo|sdsdi_io_interface:u_io_interface|gxb_rxsample:u_rxsample" &
"sdsdi_rxtx:u_demo|sdsdi_io_interface:u_io_interface|s2p:u_s2p"
```

Choose the Oversampling Ratio

By default, the design implements the oversampling interface using a ratio of 5×. However, you can select the 3× oversampling ratio by commenting the following RTL code line in the top level Verilog HDL file, **sdsdi_rxtx.v**:

```
//`define OVERSAMPLE_5X
```

Compile the Design

Quartus II project files are provided for the reference design mapped to the Cyclone video demonstration board and the Stratix GX video demonstration board. The **.qsf** files define the pinout and other compilation directives for the designs. You must compile the designs to produce the device images required for the demonstrations.

To compile any of the demonstration designs, open the relevant project in the Quartus II software and choose **Start Compilation** (Tools menu).

Use the Demonstrations

Altera provide demonstrations for the Cyclone video demonstration board or the Stratix GX video demonstration board.

You can borrow either board from Altera field applications engineers (FAEs) or sales for a short period of time for evaluation. The boards are not for sale.

Cyclone Video Demonstration Board

The SDI receive and transmit demonstration exercises both the SDI receive and transmit ports. The SDI input is received and decoded. If DIP switch 1 is set to 0 (closed), a video pattern (colourbar) is generated and transmitted on the SDI transmit port; if DIP switch 1 is set to 1 (open), the demonstration is in loopback mode—the data that comes into the receive port comes out on the transmit port. There are the following options for the receive and transmit reference clock source:

- On-board 27-MHz crystal oscillator (U7, DIL14)
- Video clock input (BNC J14, only for transmit)
- LVTTL clock input (SMA J13)
- On-board 27-MHz voltage controlled crystal oscillator (VCXO) (needed for SDI loopback mode)

By default, `clk1` transmits; `clk3` receives. To use the on-board VCXO for transmitting, ensure you set jumpers `JP_CLK1` in position 3 (pin 3 and 2 shorted).

To use the demonstration, follow these steps:

1. Select the required clock source by editing the **source\sdsdi_cyclone_board\sdsdi_rxtx_cyclone_board.v** file and uncommenting the appropriate define statement.

2. Fit a 27-MHz crystal oscillator (DIL14) in U7.

3. Compile the design using the Quartus II software. The **.qpf** and **.qsf** files are in the **quartus\sdsdi_rxtx_cyclone_board** directory.

4. If in video pattern mode, connect a coaxial cable between BNC J9 and an SDI monitor.

   or

   If in loopback mode, connect an SDI source to BNC J3 and connect an SDI monitor to BNC J9.

5. Connect a 5-V power supply to the board (J15).

6. Connect a ByteBlaster™ II cable on J11 and load the **sdsdi_rxtx_cyclone_board.sof** file.

7. Set switch 1 to open for NTSC format, closed for PAL format (only necessary in video pattern mode).

8. Set switch 2 to open for video pattern mode, closed for loopback mode; in video pattern mode, a colorbar pattern is generated by defaut.

9. Press button S1 to reset the system.

10. Press button S2 to remove the color from the pattern.

11. Press button S3 to switch to an SDI pathological pattern.

LED 0 flashes to indicate that the receive reference clock is running. LED 1 flashes to indicate that the transmit reference clock is running. LED 2 illuminates to indicate that the receiver is in lock. If in loopback mode, LED 11 illuminates to indicate that the VCXO is in lock.

Stratix GX Video Demonstration Board

As for the Cyclone video demonstration board, in pattern mode, the demonstration generates and transmits a test pattern as an SDI output. It also receives and decodes an SDI input. In loopback mode, the demonstration receives, decodes, encodes, and re-transmits an SDI signal. The receive reference clock is provided by the on-board 27-MHz crystal oscillator. The transmit reference clock is provided by the on-board 27-MHz VCXO, which is used as a "clean-up" PLL on the recovered data valid flag.

Because Stratix GX high-speed transceivers are not used, use an external video interface board for the cable interface. LVDS I/Os are used for SDI input and output via the connections to/from spare SMAs.

For more information on the video interface board, contact your Altera representative.

Before you use the demonstration, connect the video interface board to the Stratix GX video demonstration board, by following these steps:

1. Connect SMA `ASI_TX1_N` (J26) to video interface board J4 and `ASI_TX1_P` (J28) to video interface board J6.
2. Connect SMA `ASI_RX1_N` (J27) to video interface board J9 and `ASI_RX1_P` (J29) to video interface board J7.
3. Connect the video interface board ground lead and ground of the Stratix GX video demonstration board to the ground of the power supply.
4. Connect the video interface board power lead to a 3.3-V supply.
5. Connect the Stratix GX video demonstration board board power lead to a 5.0-V supply.
6. Manually set the transmit edge rate using JP1 on the video interface board (link 1 and 2 for SD).

To use the SDI reference design, follow these steps:

1. Connect an SDI monitor or analyzer to the BNC connector J5 on the video interface board.
2. Connect an SDI generator to the BNC connector J8 on the video interface board.
3. Connect the ByteBlaster II cable on JP1 and load the **sdsdi_rxtx_demoboard.sof** file.
4. LED D6 and D7 should flash if the receive (local oscillator) and transmit clock (from external VCXO) are correctly connected to the Stratix GX device.

Button S3 is system reset.

For pattern mode, follow these steps:

1. Set switch 1 to open for NTSC format; closed for PAL format.
2. Set switch 2 to open for pattern mode.

3. The control status of the SDI signal displays on the monitor a colorbar pattern.

4. Press button S4 to remove the color from the pattern.

5. Press button S5 to switch to an SDI pathological pattern.

For loopback mode, set switch 2 to closed for loopback mode.

The control status of the SDI signal displays on the monitor whatever has been generated on the receiver side.