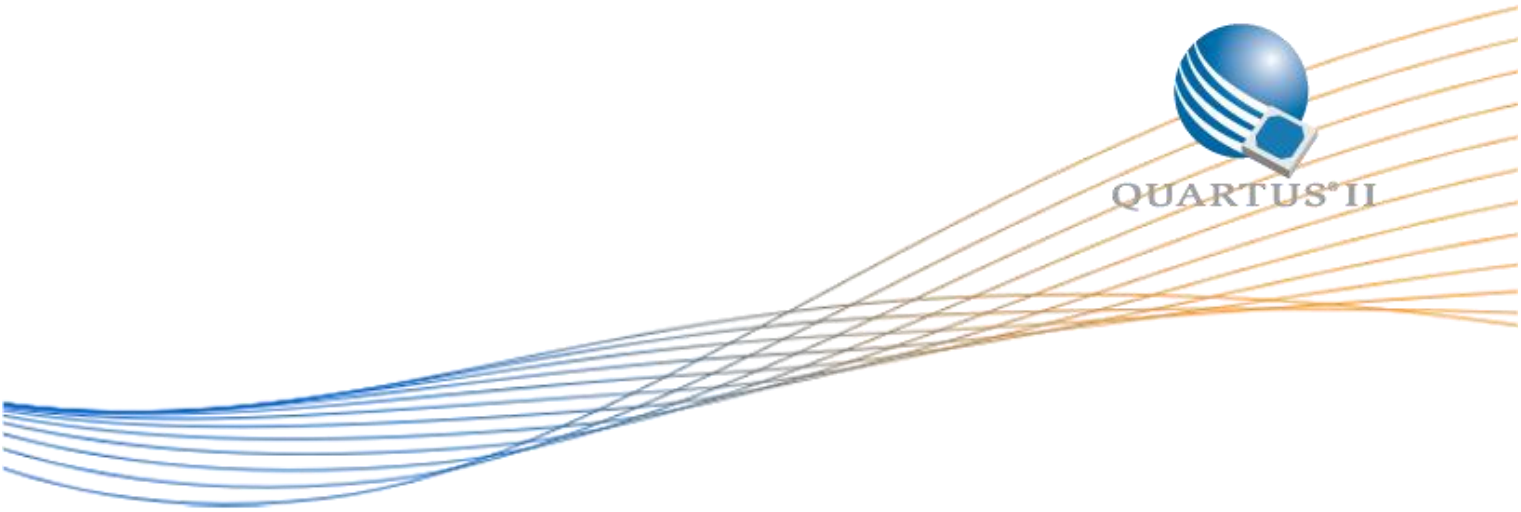

DisplayPort Design Example 15.1 (Tx-Only)

Date: 7/12/16

Revision: 1.1

Contents

Introduction

This document describes an example design that demonstrates Altera DisplayPort Source (TX) functions using a video loop-through system. DisplayPort is a next-generation video interface display technology. The Video Electronics Standards Association (VESA) developed the standard as an open digital communication interface for internal chip-to-chip and external box-to-box digital display connections such as:

- Interfaces within a PC or monitor
- Interfaces between a PC and monitor or projector
- Interfaces between a PC and TV
- Interfaces between a device (e.g. DVD player) and TV

DisplayPort uses packetized data transmission and embeds the clock signal in the serial data stream. It can transmit audio, video, or both simultaneously. It also includes a bidirectional, half-duplex auxiliary (AUX) channel for link and device management. The Hot Plug Detect (HPD) causes the DisplayPort source to initiate the link via AUX channel.

Getting Started

The DisplayPort example design supports the following FPGA development boards and requires using a DisplayPort daughter card.

FPGA Board	Daughter Card Type	Supported DP Link Rates
Stratix V GX Development Board	Bitec HSMC Daughter Card	1.62Gbps, 2.7Gbps, 5.4Gbps
Arria V GX Development Board	Bitec HSMC Daughter Card	1.62Gbps, 2.7Gbps, 5.4Gbps
Cyclone V GT Development Board	Bitec HSMC Daughter Card	1.62Gbps, 2.7Gbps
Arria 10 FPGA Development Board	Bitec FMC Daughter Card	5.4Gbps, 8.1 Gbps

The main changes in this 15.1 design compared with the 14.0 design are:

- DisplayPort IP Core has a new input “clk_cal”. (This will be discussed in detail in the “Clocks” section).
- VIP (Video and Image Processing) Suite components (Test Pattern Generator and Clocked Video Output) are upgraded to the latest version (Gen II).

Functional Description

Overview

The design example transmits a 1,920 x 1,200 color bar image over a DisplayPort TX link. The design example is implemented using Altera's Qsys tool and standalone HDL modules.

Figure 1 shows a block level diagram of the design example.

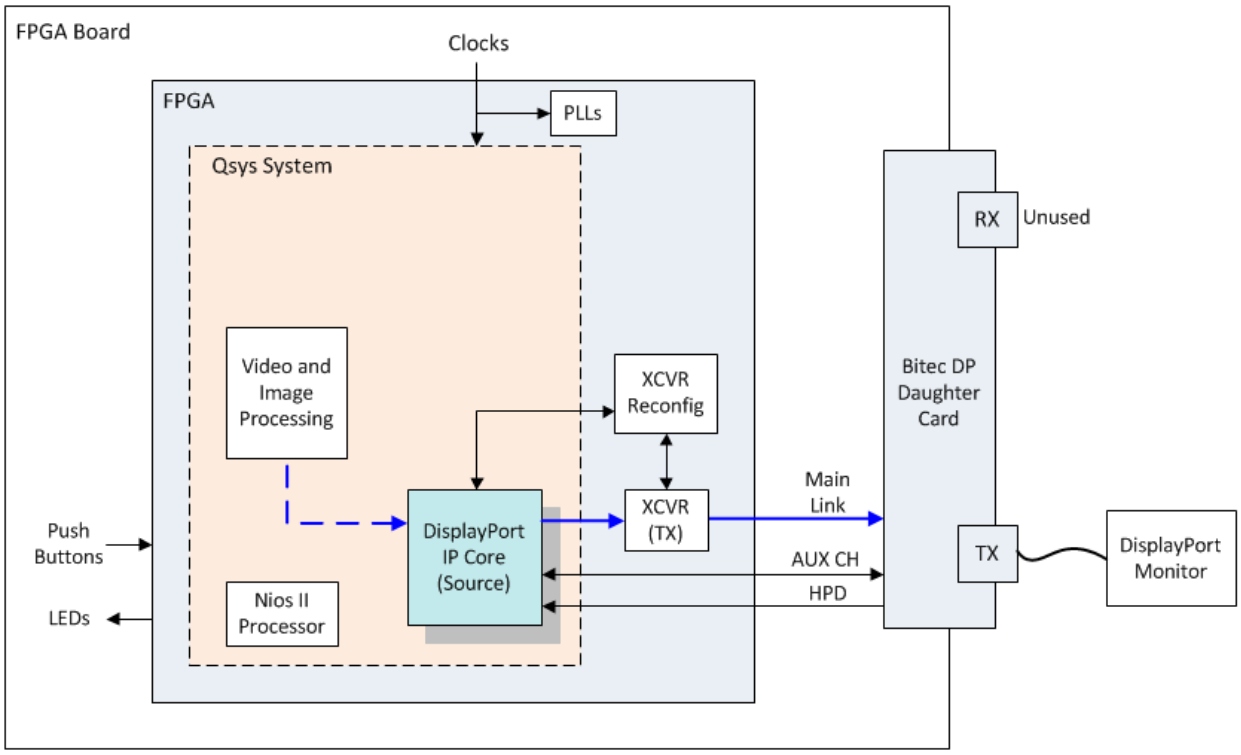

Figure 1 Block Diagram of Design Example

Clocks

FPGA development boards have various programmable oscillators. The example design uses the default outputs and doesn't require any programming.

The table below lists the clock signals for the example design.

Signal Name	Description	Use
clk	100MHz external source	• DisplayPort IP Core tx_mgmt interface • Nios II processor and peripherals • DisplayPort IP core transceiver reconfiguration mgmt logic • Transceiver reconfiguration controller • Link training state machine controller

		• Video PLL input reference clock
xcvr_pll_refclk	board-dependent external source	• Used to synthesize single 135MHz (Stratix V, Arria V) or dual 162MHz and 270MHz(Cyclone V) transceiver reference clock(s) (Note: Using REFCLK pins in the transceiver bank gives the best jitter performance. If the REFCLK pins are to be used, reprogram the oscillator clock output to 135MHz using Clock Control GUI.)
xcvr_refclk	board-dependent external source	• Transceiver reference clock for Arria 10
clk135	135MHz transceiver PLL output	• Transceiver reference clock for Arria V and Stratix V
clk162, clk270	162MHz and 270MHz transceiver PLL outputs	• Transceiver reference clocks for Cyclone V
video_clk, clk_vid	154MHz video PLL outclk0	• Video clock for Avalon Streaming (Avalon-ST) video datapath for 1920x1200 @ 60Hz • DisplayPort IP core Video Input Interface • Test Pattern Generator IP core • Clocked Video Output IP core
aux_clk, clk_16	16MHz video PLL outclk1	• Clock for 1Mbps AUX channel interface • DisplayPort IP core AUX interface • TX AUX Debug FIFO
clk_cal	50MHz calibration clock for DisplayPort	• This clock must be synchronous to the clock used for the transceiver reconfiguration block (100MHz)

DisplayPort IP Core

The example design uses the following parameter settings for the DisplayPort Source.

- Maximum video color depth = 8 bpc
- Maximum link rate = 5.4Gbps for Stratix V, Arria V and Arria 10, 2.7Gbps for Cyclone V
- Maximum lane count = 4
- Symbol mode = Quad for Stratix V, Arria V and Arria 10, Dual for Cyclone V
- Pixel mode = Single
- Enable AUX debug stream = Enabled
- Support CTS test automation = Enabled

Symbol mode affects the transceiver parallel bus width and the DisplayPort IP core clock frequency. DisplayPort IP core is synchronized with transceiver parallel clock output whose frequency is link rate / transceiver parallel bus width. The following table shows possible IP core clock frequencies for 5.4Gbps and 2.7Gbps link rates.

Symbol Mode (Transceiver Parallel Bus Width)	Link Rate	IP Core Clock
Dual (20-bit)	5.4Gbps	270MHz
Quad (40-bit)	5.4Gbps	135MHz
Dual (20-bit)	2.7Gbps	135MHz
Quad (40-bit)	2.7Gbps	67.5MHz

Pixel mode affects user video clock frequency and video port width of the IP core. The following table shows an example for 1920x1200@60Hz with color depth 8 bpc.

Pixel Mode	Video Clock	Maximum Color Depth (bpc)	Video Port Width (pixel*bpc*3)
Single	154MHz	8	24
Dual	154MHz / 2 = 77MHz	8	48
Quad	154MHz / 4 = 38.5MHz	8	96

Nios II Processor

DisplayPort Source requires a processor such as a Nios II processor to act as link policy maker. Enabling the processor control for DisplayPort Sink is optional.

The Nios II processor performs the following functions in the design:

- Initializes the IP core components
- Runs software that acts as DP link policy maker
- Programs the DP RX redriver EQ settings of Bitec DP daughter card
- Provides access to the IP core status, debug registers

Push Buttons

The following table lists the push button functions and the corresponding board references in each FPGA board.

Function	Cyclone V GT Development Board Reference	Arria V GX Starter Board Reference	Stratix V GX Development Board Reference	Arria 10 FPGA Development Board Reference	Description
Reset	S1	S7	S7	S3	Resets the demo design
Display configuration status	S3	S5	S5	S2	Displays the current TX MSA values and link configuration in the Nios II terminal

LEDs

The following table lists the LEDs used in the design and the corresponding board references in each FPGA board.

Function	Cyclone V GT Development Board Reference	Arria V GX Starter Board Reference	Stratix V GX Development Board Reference	Arria 10 FPGA Development Board	Description (when ON)
PLL lock status	D9	D21	D8	D4	Video clock, AUX clock, and transceiver reference clock have achieved lock
FPGA reset status	D8	D20	D7	D3	FPGA reset is asserted

Video and Image Processing

The Video and Image Processing block is part of the Qsys system in the design. It generates a test pattern and outputs processed video image to the DisplayPort IP core (Source). The following Video and Image Processing (VIP) suite IP cores are used:

- Test Pattern Generator II – Generates color bar pattern for background image
- Clocked Video Output II – Converts Avalon-ST Video protocol to DP Source video input format

Figure 2 shows the video IP connection in the Qsys system.

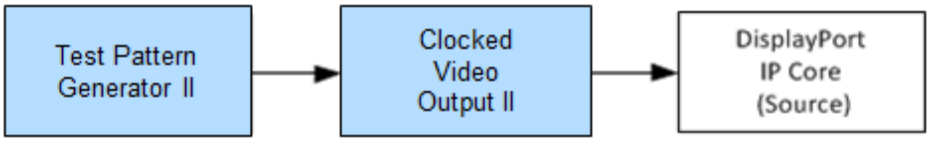

Figure 2 Video IP Connection in Qsys

This example design supports 2K resolution. The parameters for Test Pattern Generator II are set as follow:

- Maximum Frame Width: 1920
- Maximum Frame Height: 1200
- Pixels in Parallel: 1
- Interlacing: Progressive output

The output image data format of Test Pattern Generator II is set to “Progressive output”. Therefore, the Clocked Video Output II uses separate synchronization signals (vertical syncs and horizontal syncs). According to VESA and Industry Standards and Guidelines for Computer Display Monitor Timing (DMT), the timing parameters of sync signals for Clocked Video Output II are set as follow:

- Image Width: 1920
- Image Height: 1200
- Number of Pixels in Parallel: 1
- Horizontal Sync: 32 pixels
- Horizontal Front Porch: 48 pixels
- Horizontal Back Porch: 80 pixels
- Vertical Sync: 6 lines
- Vertical Front Porch: 3 lines
- Vertical Back Porch: 26 lines
- Pixel FIFO Size: 1920
- FIFO level at which to start output: 1919

TX Transceivers

The Quartus IP Catalog provides a transceiver Native PHY library that includes the transceivers configured for DisplayPort application. DisplayPort transceiver Native PHYs in the IP Catalog are based on the maximum link rate and transceiver parallel bus width.

Maximum link rate:

- HBR2 (5.4Gbps)
- HBR (2.7Gbps)
- RBR (1.62Gbps)

Transceiver parallel bus width:

- Dual Symbol (20-bit wide)
- Quad Symbol (40-bit wide)

In Quartus 14.0 and later releases, a single reference clock frequency is supported to generate all link rates. The reference clock frequency used in the DisplayPort transceiver Native PHY libraries is 135MHz.

Transceiver Reconfiguration

TX transceivers are reconfigured when DP Source and Sink auto-negotiate the link configuration during link training:

- TX link rate
- TX output voltage swing (VOD) and Pre-emphasis level

The following is a list of RTL modules for transceiver reconfiguration. All RTL modules except for the transceiver reconfiguration controller IP are provided in clear-text.

- **<device>_xcvr_reconfig.v** – Transceiver reconfiguration controller IP. <device>: sv for Stratix V, av for Arria V, cv for Cyclone V
- **reconfig_mgmt_hw_ctrl.v** – Top level reconfiguration management FSM that handles reconfiguration request from the DisplayPort IP core
- **reconfig_mgmt_write.v** – Generates Avalon-MM write cycles to the transceiver reconfiguration controller
- **dp_mif_mappings.v** – Maps DP link rate to the transceiver PLL settings
- **dp_analog_mappings.v** – Maps DP VOD and Pre-emphasis levels to the transmitter analog settings

For Arria 10 example design, the following modules are used for transceiver reconfiguration:

- **bitec_reconfig_alt_a10.v**
- **a10_reconfig_arbiter.sv**
- **gxb_tx.qip**
- **gxb_tx_fpll.qip**
- **gxb_tx_reset.qip**

Using the Design Example

Hardware Requirements

This design is intended for one of the following boards:

- Stratix V GX Development Board
- Arria V GX Development Board
- Cyclone V GT Development Board
- Arria 10 FPGA Development Board

The design also requires the use of a Bitec HSMC DisplayPort daughter card (FMC daughter card for Arria 10).

The hardware setup is as follows:

- Connect the FPGA board to your PC using a USB cable. If your board doesn't have On-Board USB-Blaster II connection, you can use an external USB-Blaster cable.
- Attach the Bitec DisplayPort daughter card to the connector on your board.
- Connect a DisplayPort monitor to the TX port on the daughter card using a DisplayPort cable.

Figure 3 and figure 4 show the hardware setup for Arria V board and Arria 10 board.

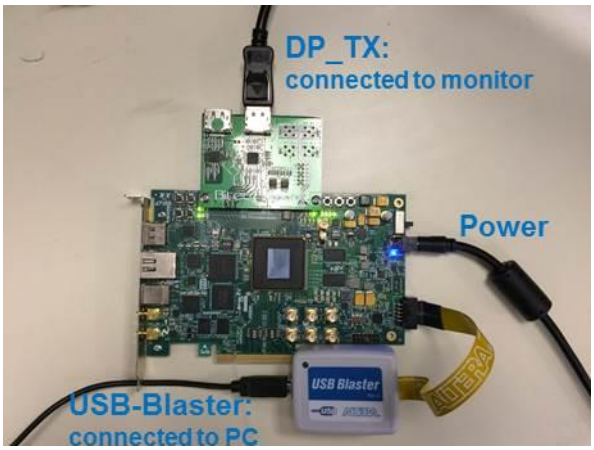

Figure 3 Arria V Hardware Setup

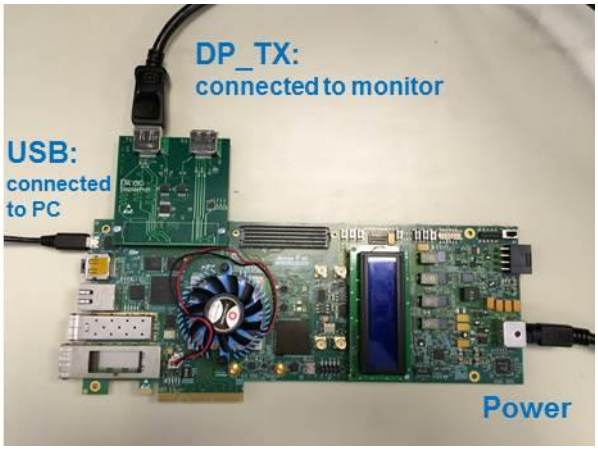

Figure 4 Arria 10 Hardware Setup

Software Requirements

This design was tested using Altera Quartus II software v15.1 release.

Compiling the Design Example

In this step you use a script to build and compile the FPGA design.

Cyclone V, Arria V, and Stratix V

Open a Nios II command shell, navigate within the shell to the project directory, and type the command:

```
quartus_sh –t runall.tcl
```

The script performs the following steps:

- Regenerates the MegaWizard Plug-In Manager Components
  - <device>_video_pll.v
  - <device>_xcvr_pll.v
  - <device>_aux_buffer.v

    - <device>_xcvr_reconfig.v
    - <device>_native_phy_tx.v
    - <device>_native_phy_rx.v
- Regenerates the Qsys system, including the DisplayPort IP core
- Creates the project, overwriting any previous settings files
- Adds the assignments to the project by running assignments.tcl
- Compiles the project

Arria 10

Open a Nios II command shell, navigate within the shell to the project directory, and type the command:

```
./runall.tcl
```

The script performs the following steps:

- Runs build_ip.tcl to regenerate all IP Catalog and Qsys components
- Creates the project, overwriting any previous settings files
- Adds the assignments to the project by running assignments.tcl
- Builds the software for the project by running build_sw.sh
- Compiles the project

Build, Load, and Run the Software

In this step you build the software, load it into the device, and run the software.

Cyclone V, Arria V, and Stratix V

Open a Nios II command shell. Navigate within the shell to the “Software” folder inside the project directory, and type the command:

```
./batch_script.sh <USB cable number>
```

If you do not know the USB cable number, use the command

```
jtagconfig
```

The script performs the following steps:

- Builds the software
- Programs the device with the .sof file from compilation
- Downloads the Nios II program (.elf file) to the device
- Launches a debug terminal.

If you have already built the software but want to program the device and download the software again, you can instead use the command:

```
./rerun.sh
```

This script performs the same function but skips the software-building step.

Note: If you are using Linux and you encounter an error running either script, it may be because the script is expecting a Windows environment. To correct this, type the commands:

```
dos2unix batch_script.sh

dos2unix rerun.sh
```

To switch back, use the same commands but replace “dos2unix” with “unix2dos”

Arria 10

The software was already built in the previous step, but there is a script to program the device and download the software. Open a Nios II command shell and navigate within the shell to the project directory. Type the command:

```
./rerun.sh
```

The script performs the following steps:

- Programs the device with the .sof file from compilation
- Downloads the Nios II program (.elf file) to the device
- Launches a debug terminal

Note: If you are using Linux and you encounter an error running either script, it may be because the script is expecting a Windows environment. To correct, type the command:

```
dos2unix rerun.sh
```

To switch back, use the same commands but replace “dos2unix” with “unix2dos”

Viewing the Results

The DisplayPort monitor will display the color bars shown below:

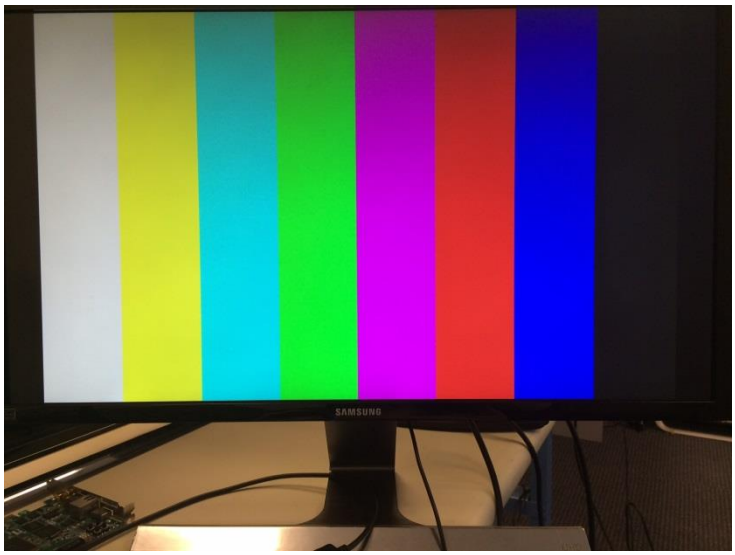

Figure 5 Monitor Display Color bars

The Nios II command shell will capture and display the AUX channel transactions:

```
00 00 00
00000844 [SRC] Reply got  AUX_ACK 00
00000158 [SRC] Req sent   AUX_RD @ 0202 (LANE0_1_STATUS) 90 02 02 01
00000611 [SRC] Reply got  AUX_ACK 00 00 00
00000203 [SRC] Req sent   AUX_RD @ 0206 (ADJUST_REQUEST_LANE0_1) 90 02 06 01
00000611 [SRC] Reply got  AUX_ACK 00 22 22
00000269 [SRC] Req sent   AUX_WR @ 0103 (TRAINING_LANE0_SET) 80 01 03 03 02 02 0
2 02
00000828 [SRC] Reply got  AUX_ACK 00
00000369 [SRC] Req sent   AUX_RD @ 0202 (LANE0_1_STATUS) 90 02 02 01
00000612 [SRC] Reply got  AUX_ACK 00 11 11
00000206 [SRC] Req sent   AUX_RD @ 0206 (ADJUST_REQUEST_LANE0_1) 90 02 06 01
00000611 [SRC] Reply got  AUX_ACK 00 22 22
00000229 [SRC] Req sent   AUX_WR @ 0102 (TRAINING_PATTERN_SET) 80 01 02 00 22
00000788 [SRC] Reply got  AUX_ACK 00
00000161 [SRC] Req sent   AUX_RD @ 0206 (ADJUST_REQUEST_LANE0_1) 90 02 06 01
00000620 [SRC] Reply got  AUX_ACK 00 66 66
00000259 [SRC] Req sent   AUX_WR @ 0103 (TRAINING_LANE0_SET) 80 01 03 03 0A 0A 0
A 0A
00000678 [SRC] Reply got  AUX_ACK 00
00000155 [SRC] Req sent   AUX_RD @ 0202 (LANE0_1_STATUS) 90 02 02 01
00000610 [SRC] Reply got  AUX_ACK 00 77 77
00000204 [SRC] Req sent   AUX_RD @ 0206 (ADJUST_REQUEST_LANE0_1) 90 02 06 01
00000612 [SRC] Reply got  AUX_ACK 00 66 66
00000226 [SRC] Req sent   AUX_WR @ 0102 (TRAINING_PATTERN_SET) 80 01 02 00 00
00000790 [SRC] Reply got  AUX_ACK 00
-15522111 [SRC] Req sent   AUX_RD @ 0201 (DEVICE_SERVICE_IRQ_VECTOR) 90 02 01 00

00000618 [SRC] Reply got  AUX_ACK 00 00
00000177 [SRC] Req sent   AUX_RD @ 0100 (LINK_BW_SET) 90 01 00 01
00000617 [SRC] Reply got  AUX_ACK 00 0A 84
00000204 [SRC] Req sent   AUX_RD @ 0200 (SINK_COUNT) 90 02 00 05
```

Figure 6 Nios II Terminal Print AUX Info

As stated in the “Push Button” section before, you can push the reset button to build link training again. Also, you can push the dump button to display MSA information, as shown in Figure 7.

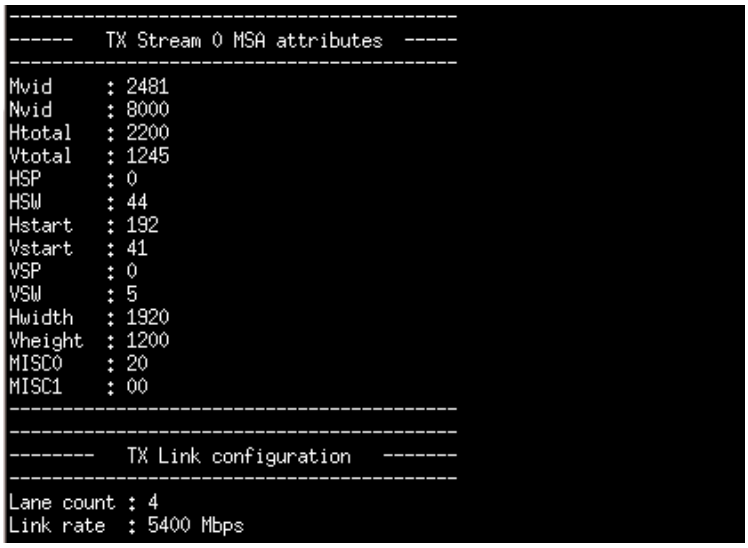

Figure 7 Nios II Terminal Print MSA Info

4K Support

4K (3820x2160) resolution is supported in the Arria V, Stratix V and Arria 10 examples with some minor changes to the 2K system. To guarantee the performance of the system, we change the pixel mode to Quad. Therefore, when running a design at 3840x2160@60Hz, the pixel clock is 533MHz / 4 = 133MHz.

Here are the parameters that need to be changed:

DisplayPort IP Core

- Pixel Input Mode: Quad

Test Pattern Generator II

- Maximum Frame Width: 3840
- Maximum Frame Height: 2160
- Pixels in Parallel: 4

Clocked Video Output II

- Image Width: 3840
- Image Height: 2160
- Number of Pixels in Parallel: 4
- Horizontal Sync: 32 pixels
- Horizontal Front Porch: 48 pixels
- Horizontal Back Porch: 80 pixels

- Vertical Sync: 5 lines
- Vertical Front Porch: 3 lines
- Vertical Back Porch: 54 lines
- Pixel FIFO Size: 3840
- FIFO level at which to start output: 3839

<device>_video_pll

- video_pll_inst.outclk_0 Desired Frequency: 133.333MHz

<device>_dp_demo.v

- wire tx_vid_v_sync must be width 4
- wire tx_vid_h_sync must be width 4
- wire tx_vid_vid_datavalid must be width 4
- wire tx_vid_data must be width 96

Reference

DisplayPort IP Core User Guide

https://www.altera.com/en_US/pdfs/literature/ug/ug_displayport.pdf

Video and Image Processing Suite User Guide

https://www.altera.com/en_US/pdfs/literature/ug/ug_vip.pdf

VESA DisplayPort Standard, Version 1, Revision 2a, May 23, 2012

VESA and Industry Standards and Guidelines Display Monitor Timing (DMT), Version 1.0, Rev.13, February 8, 2013