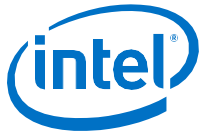


# **Intel® Cyclone® 10 GX DisplayPort 4Kp60 with Video and Image Processing Pipeline Retransmit Reference Design**

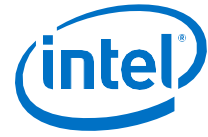
*2018.8*



## Contents

---

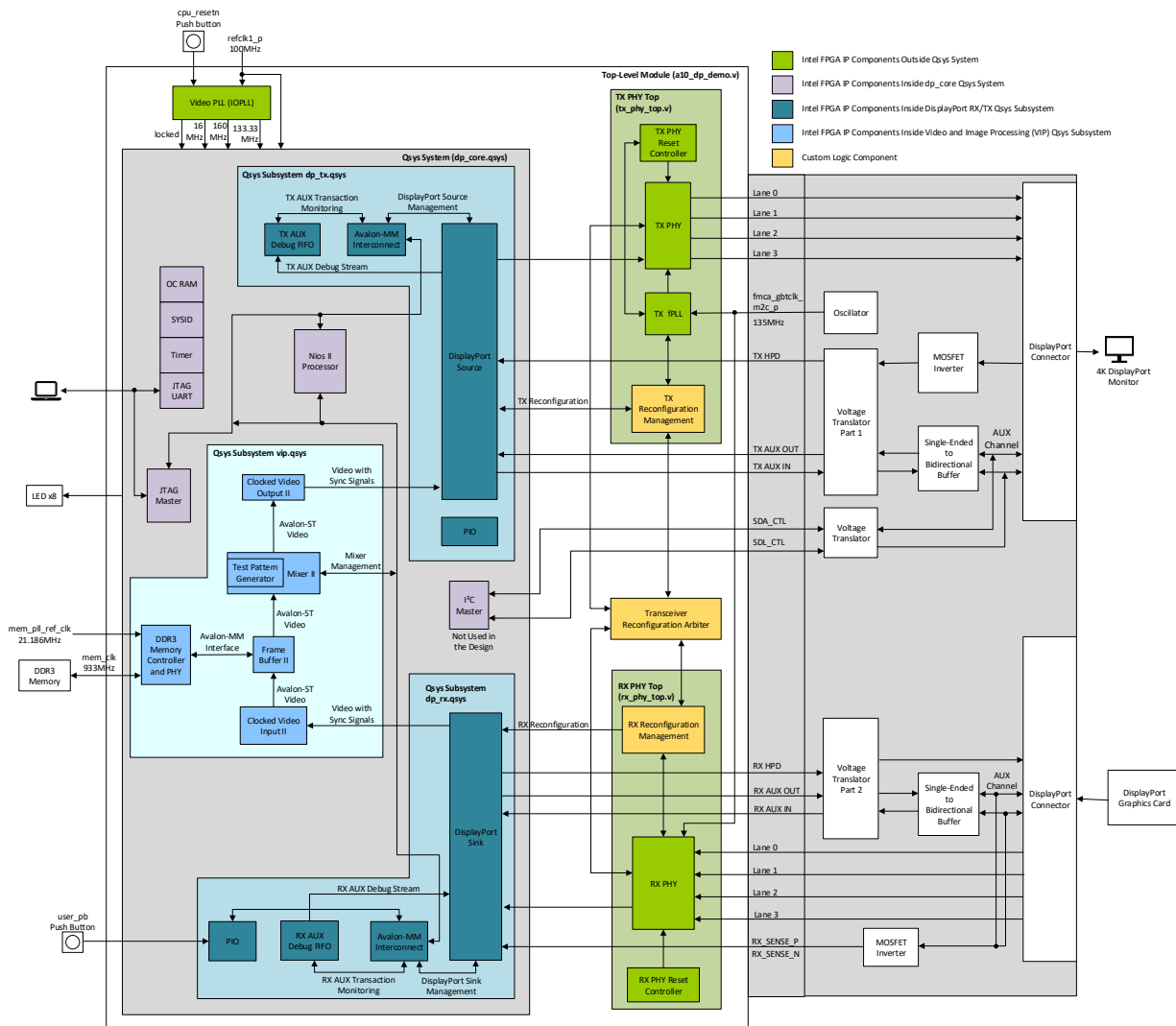
<b>Cyclone® 10 GX DisplayPort 4Kp60 with Video and Image Processing Pipeline Retransmit Reference Design</b> .....	<b>3</b>
1.1 Reference Design Components .....	5
1.1.1 Clocking Scheme .....	6
1.1.2 DisplayPort IP Core .....	7
1.1.3 Video and Image Processing Block.....	8
1.1.4 External Memory Interface.....	10
1.1.5 Nios II Processor .....	10
1.1.6 Transceiver.....	10
1.1.7 Push Buttons and LEDs.....	11
1.2 Reference Design Folders and Files.....	12
1.3 Quick Start Guide.....	14
1.3.1 Hardware and Software Requirements .....	14
1.3.2 Compiling and Running the Reference Design.....	14
1.3.3 Viewing the Expected Result.....	15
1.4 Rebuilding the Nios II Software.....	17
1.5 Reference Design Debug Features.....	17
1.6 Revision History .....	20



# Cyclone® 10 GX DisplayPort 4Kp60 with Video and Image Processing Pipeline Retransmit Reference Design

The Cyclone 10 GX DisplayPort 4Kp60 with Video and Image Processing Pipeline Retransmit reference design demonstrates Intel's video connectivity, the DisplayPort Sink (RX) and Source (TX) functions using a video loop-through system.

Figure 1. Reference Design Block Diagram



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

ISO  
9001:2008  
Registered



1. The reference design receives video data (up to a resolution of 3840 pixels × 2160 lines) over the DisplayPort RX link.
2. The design then converts the received video to Avalon Streaming (Avalon-ST) image stream and stores into the external memory.
3. The design mixes the buffered image with a 3840 × 2160 color bar background and sends the combined image to the DisplayPort Source.
4. The DisplayPort Source transmits the combined image to a DisplayPort capable monitor over the DisplayPort TX link.
5. The DisplayPort interface supports dynamic scaling between 1, 2 and 4 lanes:
  - Reduced Bit Rate (RBR) @ 1.62 Gbps/lane
  - High Bit Rate (HBR) @ 2.7 Gbps/lane
  - High Bit Rate 2 (HBR2) @ 5.4 Gbps/lane

The TX and RX physical layer (PHY) are independent of each other although they are placed at the same transceiver channels; the DisplayPort Sink may run at 1 lane @ 2.7 Gbps while the DisplayPort Source runs at 4 lanes @ 5.4 Gbps concurrently. There is no audio or secondary stream being retransmitted in this reference design.

This reference design is implemented using Intel's Platform Designer integration tool and standalone HDL modules.

**Note:** The I<sup>2</sup>C interface is not used in this design.

#### **Related Links**

- [DisplayPort IP Core User Guide](#)
- [Video and Image Processing Suite User Guide](#)
- [Cyclone 10 GX FPGA Development Kit](#)
- [FMC DisplayPort Daughter Card](#)
- [Arria 10 DisplayPort 4Kp60 with Video and Image Processing Pipeline Retransmit Reference Design \(Quartus Prime Pro Edition\)](#)



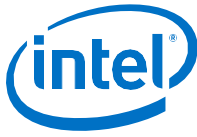
## 1.1 Reference Design Components

The reference design consists of Platform Designer, PHY, and Clock subsystems.

**Table 1. Reference Design Components**

The table lists the major components in the reference design.

System	Components
Platform Designer subsystem	DisplayPort Source and Sink cores
	Video and Image Processing IP cores <ul style="list-style-type: none"> <li>• Clocked Video Input II (CVI)</li> <li>• Clocked Video Output II (CVO)</li> <li>• Frame Buffer II</li> <li>• Mixer II</li> </ul>
	Nios II processor
	DDR3 External Memory Interface
	Avalon Memory-Mapped (Avalon-MM) FIFO Memory
	JTAG to Avalon-MM master bridge
PHY subsystem	Simplex TX and RX Native PHY
	TX fPLL
	Altera PHY reset controller
	TX and RX Bitec reconfiguration management
	Transceiver reconfiguration arbiter
Clock subsystem	IO PLL for video data path



### 1.1.1 Clocking Scheme

The reference design requires several clock sources from the FPGA development kit and the FMC daughter card for proper operation. The reference design uses the default clock frequency from the oscillator; no programming is required through the Clock Control application.

**Table 2. Clock Signals**

Signal	Description	Pin Number	I/O Standard	Usage
refclk1_p	External 100 MHz clock from U64 programmable Si5332A OUT7 on the FPGA development kit.	F23/G23	1.8V	<ul style="list-style-type: none"> <li>Avalon-MM interface for DisplayPort and VIP IP cores</li> <li>Transceiver reconfiguration interface</li> <li>TX and RX Bitec reconfiguration management</li> <li>Transceiver reconfiguration arbiter</li> <li>Intel PHY reset controller</li> <li>TX PLL and TX/RX transceiver channel reconfiguration interfaces</li> <li>Nios II CPU and peripherals</li> <li>Video PLL (IO PLL) input reference clock</li> </ul>
fmca_gbtclk_m2c_p	External 135 MHz clock from Bitec FMC daughter card. The oscillator is not programmable.	W24/W23	LVDS	<ul style="list-style-type: none"> <li>TX PLL input reference clock</li> <li>RX CDR input reference clock</li> </ul>
mem_pll_ref_clk	External 21.186 MHz clock from U64 programmable oscillator on the FPGA development kit.	AA18/AA19	LVDS	DDR3 external memory interface input reference clock
dp_tx_vid_clk	Generated 133.33 MHz clock from video PLL.	-	-	DisplayPort Source-Clocked Video Output interface
dp_rx_vid_clk	Generated 160 MHz clock from video PLL.	-	-	<ul style="list-style-type: none"> <li>DisplayPort Sink-Clocked Video Output interface</li> <li>Avalon-ST video data path</li> </ul>
clk_16	Generated 16MHz clock from video PLL	-	-	<ul style="list-style-type: none"> <li>DisplayPort Sink and Source 1 Mbps AUX channel interface</li> <li>DisplayPort Sink and Source AUX debug FIFO</li> </ul>
clk_cal	Derived 50MHz clock from refclk1	-	-	DisplayPort Sink and Source calibration. This clock must be synchronous to the clock used for the transceiver reconfiguration block (100 MHz)



The 160 MHz clock output from the video PLL drives the DisplayPort Sink and Clocked Video Input interface. This interface runs at input video pixel clock domain; this clock frequency must be equal or greater than the required pixel clock frequency of the input video stream.

**Note:** The ANSI/CEA-861-F standard requires the 3840 × 2160 @ 60 Hz video stream to run at 594.0 MHz pixel clock. This design uses 4 pixels per clock so that the interface runs at 148.5 MHz (quarter rate of 594.0 MHz), but with 4 times video bus width between the DisplayPort Sink and Clocked Video Input. Because 160 MHz is higher than 148.5 MHz, this frequency is sufficient to support 4Kp60 input video stream.

The 133.33 MHz clock output from the video PLL drives the DisplayPort Source and Clocked Video Output interface. The CVT-RB specification states that the 133.33 MHz should be derived from the reduced blanking period of the 4K video output stream.

**Table 3. 4Kp60 Video Stream Timing Information for Normal and Reduced Blanking**

H Active × V Active	H Total	H Blank	V Total	V Blank	Pixel Frequency
Normal	4,400	560	2,250	90	594.00 MHz
Reduced	4,000	160	2,222	62	533.28 MHz

This design uses 4 pixels per clock so that the interface runs at 133.33 MHz (quarter rate of 533.28 MHz), but with 4 times wider video bus between the DisplayPort Source and Clocked Video Output.

### 1.1.2 DisplayPort IP Core

The reference design uses specific DisplayPort IP core parameter settings.

**Table 4. DisplayPort IP Core Settings for the Reference Design**

Parameter	Value	Notes
<b>Maximum video input color depth (TX) / Maximum video output color depth (RX)</b>	8 bpc	This reference design supports GPU and monitor up to a maximum of 8 bit-per-color depth.
<b>Maximum link rate</b>	5.4 Gbps	The bandwidth requirement for 4Kp60, 10 bpc video stream through serial link: Active video resolution = 3840 × 2160 pixels/frame Total resolution (including reduced blanking) = 4000 × 2222 pixels/frame Refresh rate = 60 Hz or 60 frames per second Bits per pixel = 8 bpc × 3 colors = 24 bits per pixel Total bandwidth = (4000 × 2222) pixel/frame × 60 frame/s × 24 bits/pixel = 15.9984 Gbits/s With 8b/10b encoding scheme, the actual bandwidth required = 12.798 × 10/8 = 15.998 Gbps With 4 lanes at 5.4 Gbps, the aggregated bandwidth of 21.6 Gbps is sufficient to support the 4K video stream at 60 Hz refresh rate.
<b>Maximum lane count</b>	4	
<b>Symbol output mode (Source) / Symbol input mode (Sink)</b>	Quad	Symbol mode affects the transceiver parallel bus width and the DisplayPort IP core clock frequency. The DisplayPort IP core synchronizes with the transceiver parallel clock. The parallel clock frequency is link rate / transceiver parallel bus width.

*continued...*



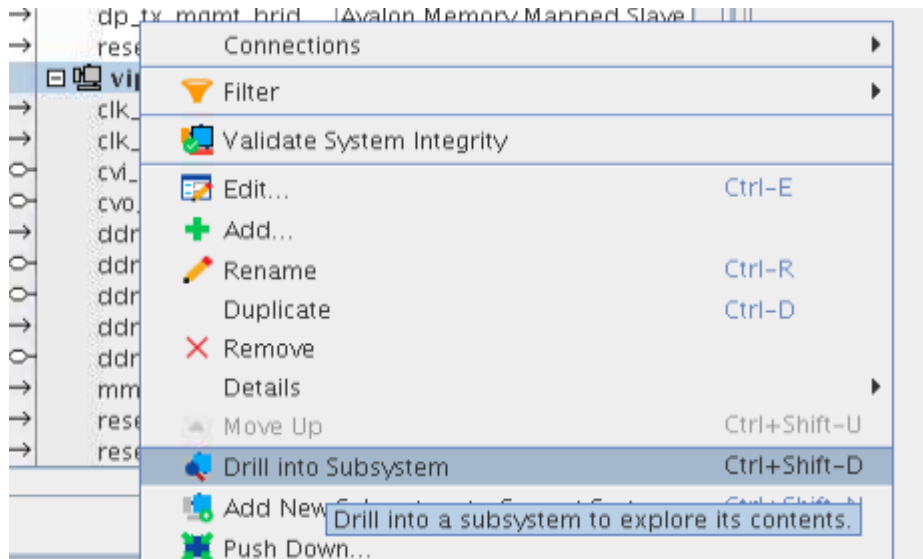
Parameter	Value	Notes	
		Symbol Mode	IP Core Clock
		Dual (20 bits)	5400/20 = 270 MHz
		Quad (40 bits)	5400/40 = 135 MHz
<b>Pixel input mode (Source)/ Pixel output mode (Sink)</b>	Quad	Pixel mode affects the video clock frequency and video port width of the IP core. For 4Kp60 video stream, the bandwidth requirement is $4000 \times 2222 \times 60$ pixel/s = 533280000 pixels/s. Because of the high bandwidth requirement, the design requires dual or quad pixel mode for timing closure.	
		Pixel Mode	Video Clock
		Single (1 pixel/clock)	533.28 MHz
		Dual (2 pixels/clock)	266.64 MHz
		Quad (4 pixels/clock)	133.32 MHz

### 1.1.3 Video and Image Processing Block

The Video and Image Processing (VIP) block is a subsystem within the Platform Designer system in the design.

The VIP block receives video data from the DisplayPort Sink, processes and transmits the processed data to the DisplayPort Source. To navigate to the VIP subsystem through `dp_core.qsys`, right click **vip**, and select **Drill into subsystem**.

**Figure 2. Navigating to the VIP Subsystem**







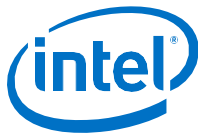
The VIP block comprises the following components:

- Clocked Video Input II IP core: Converts the DisplayPort Sink video output format to Avalon-ST video protocol
- Frame Buffer II IP core: Handles mismatch in RX and TX video data rate through triple-buffering
- Mixer II IP core: Overlays the buffered image on top of the background color bar
- Clocked Video Output II IP core: Converts the Avalon-ST video protocol to the DisplayPort Source video input format

This reference design supports 4K resolution.

**Table 5. VIP IP Core Settings for the Reference Design**

IP Core	Parameter	Value
Clocked Video Input II	Bits per pixel per color plane	8
	Number of color planes	3
	Number of pixels in parallel	4
	Use control port	Off
Frame Buffer II	Maximum frame width	3840
	Maximum frame height	2160
	Bits per pixel per color plane	8
	Number of color planes	3
	Pixels in parallel	4
	Avalon-MM master (s) local ports width	512
	AV-MM burst target write	64
	AV-MM burst target read	64
	Frame dropping	On
	Frame repeating	On
	Drop invalid frames	On
Run-time writer control	Off	
Mixer II	Maximum output frame width	3840
	Maximum output frame height	2160
	Bits per pixel per color plane	8
	Number of pixels in parallel	4
	Colorspace (used for background layer)	RGB
	Pattern	Color bars
	How user packets are handled	Discard all user packets received
Clocked Video Output II	Image width / Active pixels	3840
	Image height / Active lines	2160
	Bits per pixel per color plane	8
	Number of color planes	3
	Number of pixels in parallel	4
	Separate syncs only - Frame/ Field 1 Horizontal sync	32
		<i>continued...</i>



	<b>Separate syncs only - Frame/ Field 1 Horizontal front porch</b>	48
	<b>Separate syncs only - Frame/ Field 1 Horizontal back porch</b>	80
	<b>Separate syncs only - Frame/ Field 1 Vertical sync</b>	5
	<b>Separate syncs only - Frame/ Field 1 Vertical front porch</b>	3
	<b>Separate syncs only - Frame/ Field 1 Vertical back porch</b>	54
	<b>Pixel FIFO size</b>	3840
	<b>FIFO level at which to start output</b>	3839
	<b>Use control port</b>	Off

### 1.1.4 External Memory Interface

The Frame Buffer II IP core uses the external on board DDR3 SDRAM to triple-buffer video frames and handle mismatch in RX and TX video data rates. The IP core writes to the memory to store input pixels and reads from the memory to retrieve video frames and transmit them.

The DDR3 IP is configured with x32 with ECC @ 933 MHz interface.

The Frame Buffer II IP core supports up to a DQ width of x64.

### 1.1.5 Nios II Processor

The DisplayPort Source and Sink require a processor (e.g. Nios II processor) to act as link policy maker.

The design requires the **Enable GPU Control** option for the DisplayPort Sink to be turned on.

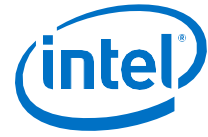
The Nios II processor performs the following functions:

- Runs software that acts as a DisplayPort link policy maker.
- Provides access to the DisplayPort IP core status and debug registers.
- Retrieves AUX channel transaction logs from the DisplayPort Source and Sink AUX debug FIFO.
- Monitors push buttons to print Main Stream Attribute (MSA) values and AUX channel transaction logs to the Nios II terminal.
- Initializes the VIP Suite IP cores.

### 1.1.6 Transceiver

The DisplayPort main link uses the Native PHY IP core for serial communication between the GPU and monitor.

The reference design uses separate simplex TX and RX PHY blocks because the TX and RX channels may run at different data rates based on the link training results.



**Table 6. Native PHY Parameter Settings**

Parameter	Value	Corresponding DisplayPort Source/Sink Parameters
Number of data channels	4	Maximum lane count = 4
Data rate (Mbps)	5,400	TX/RX maximum link rate = 5.4 Gbps
Standard PCS/PMA interface width	20	Symbol input/output mode = Quad
TX/RX byte serializer mode	Disabled	

The design uses two Intel FPGA PHY reset controllers to control the TX and RX PHY blocks independently.

*Note:* The 70  $\mu$ s delay is not applied to the PHY reset controller because the delay is too long for the DisplayPort Source and Sink cores to complete the link training. Instead, the design uses the acknowledgment model.

The Bitec reconfiguration management module controls the reset input of the Intel FPGA PHY reset controllers, and manages the dynamic reconfiguration of the TX PHY, RX PHY and TX PLL blocks for data rate switch and PMA analog settings (TX VOD and pre-emphasis). Because of the simplex TX and RX PHY channel merging requirement, a transceiver arbiter is inserted in between the Bitec reconfiguration management module Avalon-MM master and the PHY reconfiguration Avalon-MM slave interface.

### 1.1.7 Push Buttons and LEDs

The reference design uses the push buttons and LEDs on the Cyclone 10 GX FPGA Development Kit as functional indicators.

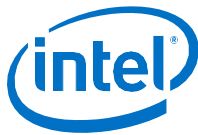
**Table 7. Push Buttons**

Function	Pin Number/I/O Standard	Schematic Net Name	Reference Designator	Description
Reset	AH2/1.8V	USER_PB2	S11	Resets the reference design.
Display MSA values	AE4/1.8V	USER_PB0	S8	Display the current TX/RX MSA values and link configuration on the Nios II terminal.

**Table 8. LEDs**

Function	Pin Number/I/O Standard	Schematic Net Name	Reference Designator	Description
DisplayPort Sink video locked	AF6/1.8V	USER_LED_G0	D19	When illuminated, it indicates that the DisplayPort Sink video

*continued...*



Function	Pin Number/I/O Standard	Schematic Net Name	Reference Designator	Description
				output stream to the Clocked Video Input IP core is stable.
DisplayPort TX PLL Locked	AE6/1.8V	USER_LED_G1	D20	When illuminated, it indicates that the DisplayPort TX PLL is locked
DisplayPort Sink link rate	AC6/1.8V AC7/1.8V	USER_LED_G2 USER_LED_G3	D21 D22	2-bit indicator of the link rate at the DisplayPort Sink. The LED arrangement is {D21, D22}: <ul style="list-style-type: none"> <li>• 00 = 1.62 Gbps (RBR)</li> <li>• 01 = 2.70 Gbps (HBR)</li> <li>• 10 = 5.40 Gbps (HBR2)</li> </ul>

## 1.2 Reference Design Folders and Files

All the relevant design files reside in the project folder.

**Table 9.**

Folder/File Name	Description
rtl/core/altera_avalon_i2c	Contains I <sup>2</sup> C master source files. I <sup>2</sup> C master is not used in this reference design.
rtl/core/dp_core	Contains the generated IP files and subfolders in dp_core.qsys of the Qsys system.
rtl/core/ip/dp_rx (for Quartus Prime Pro version)	Contains the generated IP files and subfolders in dp_rx.qsys of the Qsys subsystem.
rtl/core/ip/dp_tx (for Quartus Prime Pro version)	Contains the generated IP files and subfolders in dp_tx.qsys of the Qsys subsystem.
rtl/core/ip/vip (for Quartus Prime Pro version)	Contains the generated IP files and subfolders in vip.qsys of the Qsys subsystem.
master_image	Contains precompiled .sof and .elf files.
rtl/rx_phy/gxb_rx	Contains generated RX PHY IP files.
rtl/rx_phy/gxb_rx_reset	Contains generated RX PHY reset controller IP files.
rtl/tx_phy/gxb_tx	Contains generated TX PHY IP files.
<b>continued...</b>	



Folder/File Name	Description
rtl/tx_phy/gxb_tx_fpll	Contains generated TX fPLL IP files.
rtl/tx_phy/gxb_tx_reset	Contains generated RX PHY reset controller IP files.
rtl/i2c_gpio_buf	Contains generated IO buffer IP files for the I <sup>2</sup> C master interface. I <sup>2</sup> C master is not used in this reference design.
rtl/video_pll_a10	Contains generated the IO PLL IP files for video PLL.
software	Contains the Nios II software project. The dp_demo.zip file contains the original software project; the dp_demo/mem_init folder contains the .qip and .hex file of the software project.
tcl	Contains the TCL script for debugging purpose.
software/dp_demo_src/main.c software/dp_demo_src/rx_utils.c software/dp_demo_src/tx_utils.c software/dp_demo_src/tx_utils.h software/dp_demo_src/config.h software/dp_demo_src/vip.h	These are the C source code and header files. You can customize these files for your applications. These files will be copied to the software folder when you run the build_sw.sh script.
top.qpf and top.qsf	The Quartus Prime project and setting files for this reference design.
rtl/core/dp_core.qsys, rtl/core/dp_rx.qsys, rtl/core/dp_tx.qsys and rtl/core/vip.qsys	dp_core.qsys file belongs to the top level Platform Designer system while dp_rx.qsys file belongs to the DisplayPort RX Qsys subsystem, dp_tx.qsys file belongs to the DisplayPort TX Qsys subsystem, vip.qsys file belongs to the VIP Qsys subsystem. The design includes the dp_rx.qsys, dp_tx.qsys and vip.qsys file so that dp_core.qsys is loaded correctly into Qsys.
rtl/rx_phy/gxb_rx.ip	RX Native PHY instance variant file.
rtl/rx_phy/gxb_rx_reset.ip	RX Native PHY's transceiver PHY reset controller instance variant file.
rtl/tx_phy/gxb_tx.ip	TX Native PHY instance variant file.
rtl/tx_phy/gxb_tx_reset.ip	TX Native PHY's transceiver PHY reset controller instance variant file.
rtl/tx_phy/gxb_tx_fpll.ip	TX Native PHY's FPLL instance variant file.
rtl/i2c_gpio_buf.ip	I <sup>2</sup> C buffer instance variant file. This buffer is not used in this reference design.
rtl/video_pll_a10.ip	IO PLL instance variant file.
rtl/example.sdc,, rtl/frame_buffer_dcfifo.sdc	Display Port Reference Design SDC timing constraint file.
script/build_sw.sh	Shell script to re-build the NIOS II software.
script/rerun.sh	Shell script to load the FPGA hardware image (.sof) and software image (.elf).
rtl/core/dp_core/dp_core.sopcinfo	The build_sw.sh script uses this file to re-build the Nios II software for the control Qsys system.
Others socpinfo files	These files are not needed to re-build the Nios II software.
simple_debug_new.stp	SignalTap II file for debug purpose.
rtl/c10_reconfig_arbiter.sv	HDL module to arbitrate access to the Avalon-MM interface of the TX and RX Native PHY. This module is needed for merging simplex TX/RX Native PHY into the same physical transceiver channel.
rtl/c10_dp_demo.v	Top-level HDL file for this reference design.

**continued...**



Folder/File Name	Description
rtl/bitec_reconfig_alt_c10.v	HDL module to dynamically reconfigure the TX/RX Native PHY and TX fPLL for data rate switching.
filelist.txt	A list of all the files consisting in this design.

## 1.3 Quick Start Guide

The reference design features a hardware design that supports compilation and hardware testing.

### 1.3.1 Hardware and Software Requirements

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

#### Hardware

- Cyclone 10 GX FPGA Development Kit (10CX220YF780E5G)
- Bitec FMC daughter card rev8
- 2 DisplayPort cables of 2m length
- Micro USB cable
- PC with Gigabyte Radeon RX560 Graphic Card and Nvidia Quadro K420 Graphic Card that supports 3840 × 2160 resolution
- Dell P2715Q Monitor that supports 3840 × 2160 resolution

#### Software

- Quartus® Prime Pro (for hardware testing)

### 1.3.2 Compiling and Running the Reference Design

Intel also provides a precompiled **C10\_dp\_demo.sof** and **dp\_demo.elf** files as part of the project file in the **master\_image** directory.

Use the provided design file to run the reference design.

Follow these steps to run the reference design:

1. Configure your PC to produce an image at resolution of 1920 × 1080.
2. Install the Bitec FMC daughter card at FMC port of the C10 GX development kit board.
3. Connect the development kit board to your PC using a micro USB cable.
4. Connect the DisplayPort cable from your monitor to the TX DisplayPort connector on the FMC daughter card.
5. Connect the DisplayPort cable from your PC to the RX DisplayPort connector on the FMC daughter card.
6. Power up the development kit board.
7. Extract the reference design to your PC.
8. Use Quartus programmer to configure **C10\_dp\_demo.sof**.

**Note:** Timing failure seen on reference design is expected. User can eliminate Timing Failure by disable the signal\_tap file and then recompile the reference design.

**Figure 3. Cyclone 10 GX FPGA Development Kit Hardware Setup**



### 1.3.3 Viewing the Expected Result

The result shows if your design runs correctly.

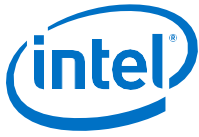
After you run the design, you should see the image from your PC overlapping with the color bar background as shown in [Figure 4](#).

**Note:** Ensure PC graphic card setting maintain the display scaling so that it doesn't always expand the display to full monitor screen despite of the display resolution setting

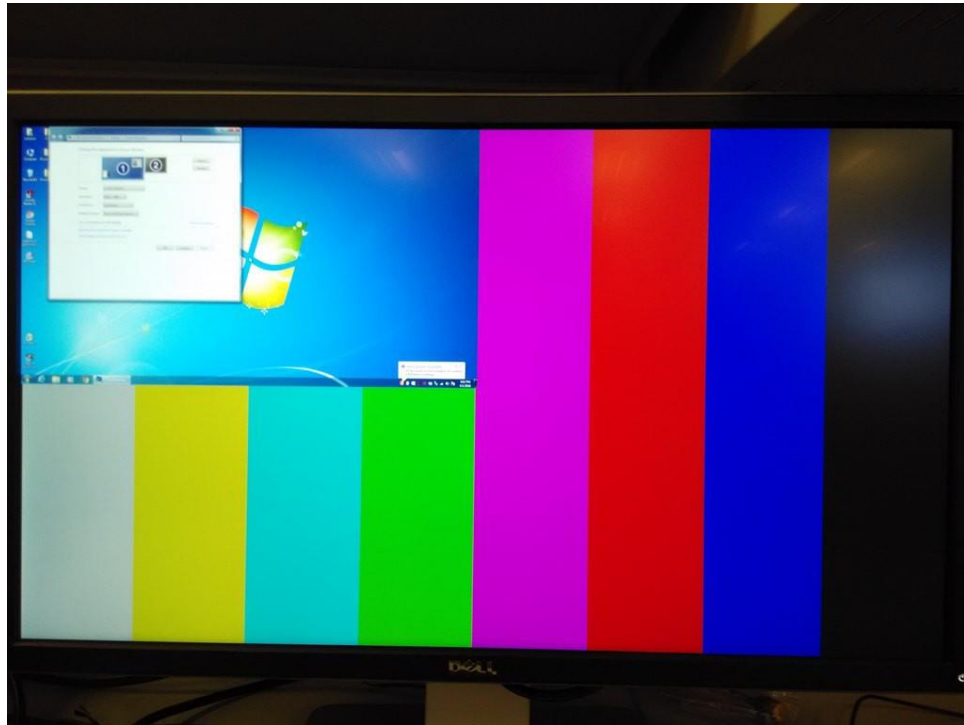
**Note:** If you do not see visible output on the monitor, press push button **USER\_PB2 (S11)** to generate a reset, causing the PC Graphic Card (Display Port Tx Source) to retrain the link.

**Note:** Another potential issue for no output on monitor is PC Graphic Card is using non-compatible bit per colour (bpc) setting. This C10 Display Port reference design only accept 8bpc setting.

**Note:** Other known debug technique is to reduce the Display Port cable length used in hardware testing. Optimum cable length would be either 1m or 2m.



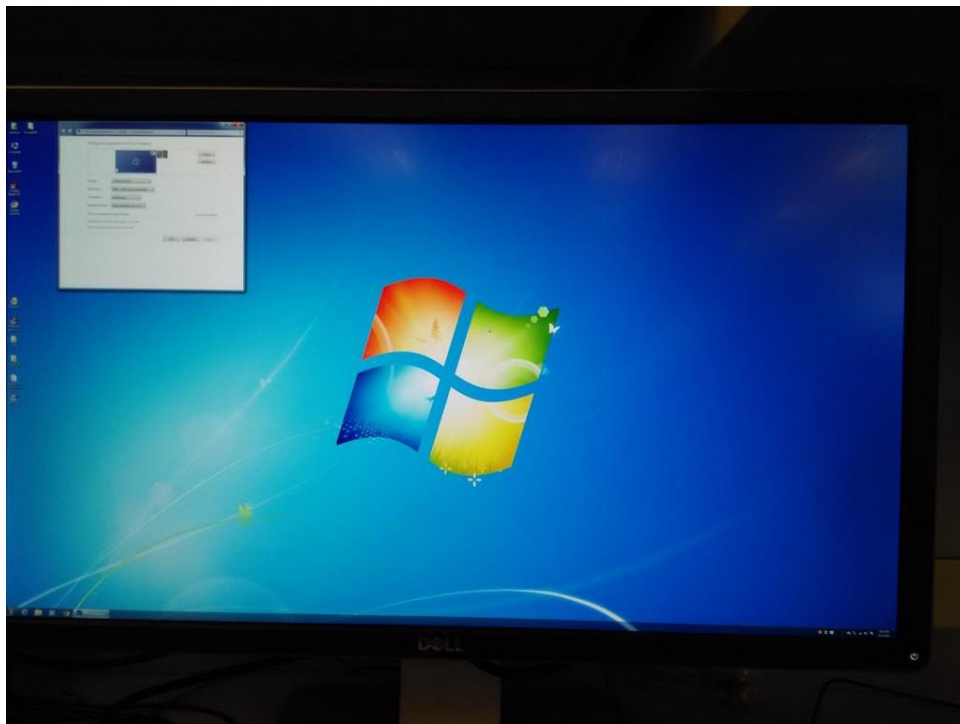
**Figure 4. Image from PC Mixed with Color Bar Background**



You can then scale the image to the maximum resolution of  $3840 \times 2160$ . You should see the updated image getting refreshed on the screen as indicated on **Figure 5**.

At this resolution, the image from the PC covers the entire color bar background.

**Figure 5. No Color Bar Background at  $3840 \times 2160$  Image Resolution**







**Attention:** DO NOT scale the resolution beyond 3840 x 2160 to avoid unexpected side effects at the Mixer II output.

## 1.4 Rebuilding the Nios II Software

If you change the Nios II software, you can rebuild the software using the Nios II command shell.

At the command shell, type `./build_sw.sh` after navigating to the **script** directory in the reference design project folder.

If you make changes to the connection or components in top level Qsys system (`dp_core.qsys`), DisplayPort RX subsystem (`dp_rx.qsys`), DisplayPort TX subsystem (`dp_tx.qsys`) or VIP subsystem (`vip.qsys`), you need to click on **Generate HDL** at the top-level Qsys system and then perform a full compilation.

Because the `sopcinfo` file is updated after the Qsys system is regenerated, you can rebuild the Nios II software while the full compilation is in progress.

## 1.5 Reference Design Debug Features

There are several debug features in this reference design that are useful for debugging link up and no video output issues.

### Main Stream Attribute Info

This debug feature is a part of the DisplayPort IP Core hardware demonstration design example. To display the Main Stream Attribute (MSA) of DisplayPort TX and RX cores, press the **USER\_PBO** push button (**S8**) on the development kit board. The TX and RX stream MSA values appears on the Nios II terminal.

1. Launch Nios II command shell after successful configuration of **C10\_dp\_demo.sof**.
2. Type `nios2-terminal` and hit enter
3. Type `s` and hit enter or simply press **S8** push button on the development kit board.
4. Monitor both Tx and Rx MSA lock is set to 1,
5. Inspect Tx link count, link rate and video resolution is correct
6. Inspect Rx link lane doesn't contains BER, link rate and video resolution is correct
7. Ensure `MISC0` value is set to 2x (indicated 8bpc setting). If `MISC0` shown 4x (indicated 10bpc setting) then this is not supported feature by this reference design

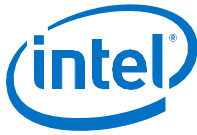
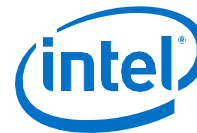


Figure 6. MSA Values dump example on Nios II Terminal

```
----- TX Main stream attributes -----
--- Stream 0 ---
MSA lock : 1
VB-ID : 10 MISC0 : 28 MISC1 : 00
Mvid : 7E69 Nvid : 8000
Htotal : 4000 Vtotal : 2222
HSP : 0000 HSW : 0032
Hstart : 0112 Vstart : 0059
VSP : 0000 VSW : 0005
Hwidth : 3840 Vheight : 2160
CRC R : 0000 CRC G : 0000 CRC B : 0000
--- Stream 1 ---
MSA lock : 0
VB-ID : 00 MISC0 : 00 MISC1 : 00
Mvid : 0000 Nvid : 0000
Htotal : 0000 Vtotal : 0000
HSP : 0000 HSW : 0000
Hstart : 0000 Vstart : 0000
VSP : 0000 VSW : 0000
Hwidth : 0000 Vheight : 0000
CRC R : 0000 CRC G : 0000 CRC B : 0000
----- TX Link configuration -----
Lane count : 4
Link rate : 5400 Mbps
----- RX Main stream attributes -----
--- Stream 0 ---
VB-ID lock : 1 MSA lock : 1
VB-ID : 00 MISC0 : 28 MISC1 : 00
Mvid : 2342 Nvid : 8000
Htotal : 2200 Vtotal : 1125
HSP : 0000 HSW : 0044
Hstart : 0192 Vstart : 0043
VSP : 0000 VSW : 0005
Hwidth : 1920 Vheight : 1080
CRC R : 0000 CRC G : 0000 CRC B : 0000
--- Stream 1 ---
VB-ID lock : 0 MSA lock : 0
VB-ID : 00 MISC0 : 00 MISC1 : 00
Mvid : 0000 Nvid : 0000
Htotal : 0000 Vtotal : 0000
HSP : 0000 HSW : 0000
Hstart : 0000 Vstart : 0000
VSP : 0000 VSW : 0000
Hwidth : 0000 Vheight : 0000
CRC R : 0000 CRC G : 0000 CRC B : 0000
----- RX Link configuration -----
CR Done: F SYM Done: 1
Lane count : 1
Link rate : 5400 Mbps
BER0 : 0000 BER1 : 7FFF
BER2 : 7FFF BER3 : 7FFF
```



## Auxiliary Channel Traffic Monitor

This debug feature is also a part of the DisplayPort IP Core hardware demonstration design example. To display the auxiliary channel transaction on the Nios II terminal, set the `BITEC_AUX_DEBUG` flag in `config.h` in the project folder to 1.

```
#define BITEC_AUX_DEBUG 1 // Set to 1 to enable AUX CH traffic monitoring
```

Rebuild the Nios II software, recompile whole Quartus design and then configure new sof file into FPGA.

Open NIOS II terminal and the AUX channel log dump should begin automatically.

## Logic Analyzer

If you want to monitor the basic debug signals of this reference design, then just launch `simple_debug_new.stp` after configured C10 GX FPGA with **C10\_dp\_demo.sof**.

## System Console

The top level Qsys includes a JTAG to Avalon Master Bridge.

- The TCL script in the `tcl` folder allows you to control the operation of the VIP IP cores.
- The `main.tcl` script contains the procedures to access the Clocked Video Input II, Clocked Video Output II, Mixer II, and Frame Buffer II control and status registers.
- The `vip_csr_offset.tcl` file contains the CSR offset of the Clocked Video Input II, Clocked Video Output II, Mixer II, and Frame Buffer II IP cores.
- The `system_base_addr_map.tcl` contains the Qsys base address of these VIP IP cores.

In this reference design, the system console accesses the Mixer CSR through the Avalon-MM interface. To access the CSR of other VIP IP cores:

1. Drill into the VIP Qsys subsystem and enable the CSR Avalon-MM interface of the IP core.
2. Connect the control port (Avalon-MM slave) to the m0 port of the Avalon-MM Pipeline Bridge component, `mm_bridge_vip`.

When assigning the base address in the VIP Qsys subsystem of the Avalon-MM slave of the VIP IP cores, follow the address map in the `system_base_addr_map.tcl` file:

- Clocked Video Input II: 0x0000
- Mixer II: 0x0200
- Clocked Video Output II: 0x0400
- Frame Buffer II: 0x0800

In `dp_core.qsys`, open the Address Map tab, ensure the base address for connection between `master_0.master` and `vip.mm_bridge_vip_s0` starts from 0x0000.

Move up to the top-level Qsys from VIP subsystem, regenerate the Qsys and perform a full compilation. To have full control of the VIP IP cores CSR, (e.g. to debug no video output issue), stop the Nios II processor from accessing the CSR.

1. Clear the `ALT_VIP` flag in `config.h`:

```
#define ALT_VIP 0 // Set to 1 if a VIP IP core (e.g. Mixer II) is used
```

2. Rebuild the Nios II software and program the updated ELF file.



3. Launch the system console and type these commands to load the `main.tcl` file and display the Mixer II output:

```
cd tcl
source main.tcl
go
```

Note: You may need to modify the variable `master_index` in `main.tcl` to access to the correct service path.

4. To stop the Mixer II from displaying video stream at its input 0 port and display the color bar only, type

```
mixer_input0_stop
```

## 1.6 Revision History

Table 10.

Date	Version	Changes
Aug 2018	2018.8.13	Initial release.
Aug 2018	2018.8.17	Add in more debug tips
Nov 2018	2018.11.1	Add in explanation for design timing failure due to <code>signal_tap</code>