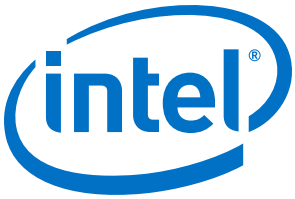


AN 900: Intel® Arria 10 DisplayPort 8K RX-only Design

Subscribe
Send Feedback

AN-900 | 2019.12.16
Latest document on the web: PDF | HTML

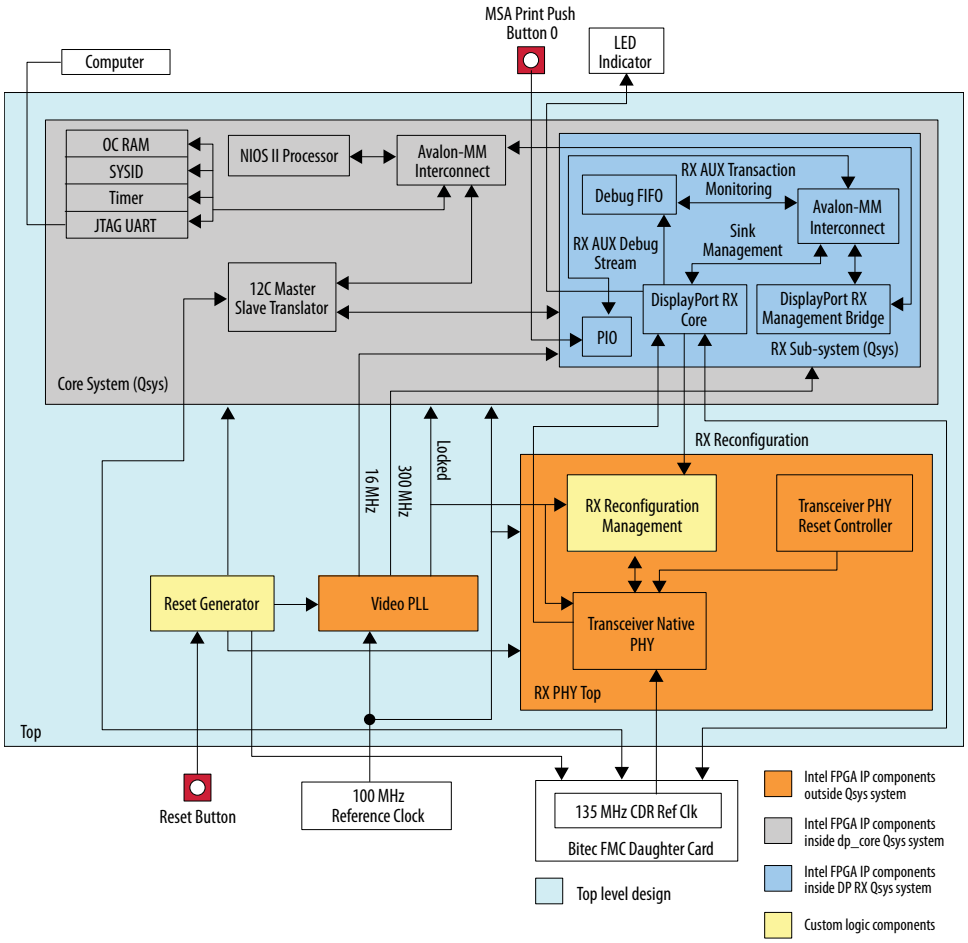
Contents

1. Intel® Arria® 10 DisplayPort 8K RX-only Design

The Intel® Arria® 10 8K DisplayPort RX-only design demonstrates how the DisplayPort sink (RX) receives video input generated by the video source through the Bitec FMC daughter card.

This design uses local Extended Display Identification Data (EDID) information to inform the source device its capabilities during Link Training process. The design enables the DisplayPort sink to receive a wide range of input video resolution from GPU, up to a maximum of 8K, 30Hz video resolution.

Figure 1. Intel Arria 10 DisplayPort 8K RX-only Design Block Diagram



Related Information

- Intel Design Store
  Provides the design files.
- DisplayPort Intel Arria 10 FPGA IP Design Example User Guide
  Provides more information about the Intel Arria 10 design examples.

1.1. Design Components

The DisplayPort Intel FPGA IP design example requires these components.

Table 1. Core System Components

Module	Description
Core System (Platform Designer)	The core system consists of the Nios® II processor and its necessary components, and the DisplayPort RX core sub-systems. This system provides the infrastructure to interconnect the Nios II processor with the DisplayPort Intel FPGA IP (RX instance) through Avalon® memory-mapped interface within a single Platform Designer system to ease the software build flow. This system consists of: • CPU Sub-system • RX Sub-system
RX Sub-system (Platform Designer)	The RX sub-system consists of: • Clock Source—The clock source to the DisplayPort RX core. This sub-system has two clock integrated sources: 300 MHz and 16 MHz. • Reset Bridge—The bridge that connects the external signal to the sub-system. This bridge synchronizes to a respective clock source before it is used. • DisplayPort RX core—DisplayPort sink core, *VESA DisplayPort Standard version 1.4*. • Debug FIFO—This FIFO captures all DisplayPort RX auxiliary cycles, and prints out in the Nios II Debug terminal. • PIO—The parallel IO that triggers the Main Stream Attribute (MSA) captured and prints out when you press the onboard push button.. • Avalon Memory-Mapped Pipeline Bridge—This bridge interconnects the Avalon memory-mapped interface between components within the RX sub-system to the Nios II processor in the Core sub-system. • EDID—The EDID RAM stores the desired EDID values in the RAM and connects to DisplayPort sink core. The design uses this component only when you turn off the **Enable GPU Control** option in the RX core.

Table 2. DisplayPort RX PHY Top Components

Module	Description
RX PHY Top	The RX PHY top level consists of the components related to the receiver PHY layer. • Transceiver Native PHY(RX)—The hard transceiver block that receives the serial data from an external video and deserializes it to 20-bit or 40-bit parallel data for the DisplayPort Intel FPGA IP sink core. • Transceiver PHY Reset Controller—The RX Reconfiguration Management module triggers the reset input of this controller to generate the corresponding analog and digital reset signals to the Transceiver Native PHY block according to the reset sequencing. • RX Reconfiguration Management—This block reconfigures and recalibrates the Transceiver Native PHY to receive serial data in the supported data rates (RBR, HBR, HBR2, and HBR3). *Note:* 8.1 Gbps is available only in the Intel Quartus® Prime Pro Edition software.

Table 3. Top-Level Common Block

Module	Description
IOPLL	IOPLL generates two common source clocks: • 300 MHz—Used as DisplayPort RX sink video clock source. • 16 MHz—Used as DisplayPort RX auxiliary clock.

1.2. Clocking Scheme

The clocking scheme illustrates the clock domains in the DisplayPort Intel FPGA IP design example.

Table 4. Clocking Scheme Signals

Clock	Signal Name in Design	Description
RX PLL Refclock	`rx_cdr_refclk`	135 MHz transceiver clock data recovery (CDR) reference clock that is divisible by the transceiver for all DisplayPort data rates (1.62 Gbps, 2.7 Gbps, 5.4 Gbps, and 8.1 Gbps). *Note:* The reference clock source of the RX refclock is located at the HSSI `refclk` pin.
RX Transceiver Clockout	`gxb_rx_clkout`	RX clock recovered from the transceiver, and the frequency varies depending on the data rate and symbols per clock.

Data Rate	Symbols per Clock	Frequency (MHz)
RBR (1.62 Gbps)	2 (dual)	81
	4 (quad)	40.5
HBR (2.7 Gbps)	2 (dual)	135
	4 (quad)	67.5
HBR2 (5.4 Gbps)	2 (dual)	270
	4 (quad)	135
HBR3 (8.1 Gbps)	4 (quad)	202.5

Clock	Signal Name in Design	Description
Management Clock	`rx_rcfg_mgmt_clk`	A free running 100 MHz clock for both Avalon memory-mapped interfaces for reconfiguration and PHY reset controller for transceiver reset sequence.

Component	Required Frequency (MHz)
Avalon memory-mapped reconfiguration	100 – 125
Transceiver PHY reset controller	1 – 500

16 MHz Clock	`dp_rx_clk_16_in_clk`	16 MHz clock used to encode and decode auxiliary channel in the DisplayPort Intel FPGA IP sink core.
Calibration Clock	`dp_rx_clk_cal`	A 50 MHz calibration clock input that must be synchronous to the Transceiver Reconfiguration module's clock. This clock is used in the DisplayPort Intel FPGA IP 's reconfiguration logic.
RX Video Clock	`dp_rx_dp_sink_rx_vid_clk`	A 300 MHz video clock for DisplayPort sink to clock video data stream.

1.3. Top Level Interface Signals

The tables list the signals for the RX-only design example.

Table 5. On-board Oscillator Signal

Signal	Direction	Width	Description
refclk1_p	Input	1	100 MHz clock source used as IOPLL reference clock and Avalon memory-mapped management clock

Table 6. User Push Buttons and LEDs

Signal	Direction	Width	Description
cpu_resetn	Input	1	Global reset
user_pb	Input	3	MSA prints out at user_pb[0]
user_led_g	Output	8	User LED (data rate information)

Table 7. DisplayPort FMC Daughter Card Pins on FMC Port A

Signal	Direction	Width	Description
fmca_gbtclk_m2c_p	Input	2	135 MHz dedicated transceiver reference clock
fmca_dp_m2c_p	Input	4	DisplayPort RX serial data
fmca_la_rx_p_6	Output	1	DisplayPort RX HPD • 1 = HPD asserted • 0 = HPD deasserted
fmca_la_tx_n_9	Input	1	DisplayPort RX Aux In
fmca_la_rx_n_6	Output	1	DisplayPort RX Aux Out
fmca_la_tx_p_9	Output	1	DisplayPort RX Aux OE
fmca_la_rx_n_8	Input	1	RX power detect (inverted)
fmca_la_tx_p_10	Input	1	RX cable detect

Table 8. FMC Onboard Retimer Reconfiguration Interface Signals

Signal	Direction	Width	Description	
			Bitec FMC Revision 10	Bitec FMC Revision 11
fmca_la_tx_p_0	Inout	1	PS8460_SDA	MCDP6000_SDA
fmca_la_tx_n_0	Inout	1	PS8460_SCL	MCDP6000_SDL
fmca_la_rx_p_0	Output	1	PS8460_EQ0	Unused
fmca_la_rx_n_0	Output	1	PS8460_EQ1	Unused
fmca_la_tx_p_1	Output	1	PS8460_PDN	Unused
fmca_la_tx_n_1	Output	1	PS8460_CFG0	Unused
fmca_la_tx_p_2	Output	1	PS8460_CFG1	Unused
fmca_la_tx_n_2	Output	1	PS8460_CFG2	Unused

1.4. Quick Start Guide

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

1.4.1. Hardware and Software Requirements

To test the design, ensure that you have the appropriate hardware and software.

Hardware

- Intel Arria 10 GX FPGA Development Kit (10AX115S2F45I1SG)
- Bitec FMC daughter card revision 10 or 11
- GPU with 8K DisplayPort output as video source
- DisplayPort cables

Software

- Intel Quartus Prime Pro Edition version 19.2 (for hardware testing)

1.4.2. Directory Structure

The directory structure lists the folders and files needed for the design.

Table 9. Intel Arria 10 DisplayPort 8K RX-only Design Directory Structure

The 8K DisplayPort RX-only design file (`A10_DP_RX_FMC_PRO.par`) has the `Additional_Files.zip` file that contains the master image and other software files. Unzip the `Additional_Files.zip` file and restructure the folders accordingly based on the table below.

Folder	File/Folder
Main	`top.qpf`
	`top.qsf`
	`rtl/`
	`script/` (extracted from `Additional_Files.zip`)
	`software/` (extracted from `Additional_Files.zip`)
	`Additional_Files/` (extracted from `Additional_Files.zip`)
	`ReadMe.txt`
	`DP_RX.stp`
`rtl/`	`a10_dp_demo.v`
	`reset_gen.sv`
	`bitec_reconfig_alt_a10.v`
	`video_pll_a10.ip`
	`reset_sync.sv`
	`example.sdc`
`rtl/core/`	`dp_rx.qsys`

continued...

Folder	File/Folder
	dp_core.qsys
	<Platform Designer (Standard)-generated files and folder>
rtl/rx_phy/	gxb_rx.ip
	gxb_rx_reset.ip
	rx_phy_top.v
	<Platform Designer (Standard)-generated files and folder>
rtl/video_pll_a10/	video_pll_a10.qip
	<Platform Designer (Standard)-generated files and folder>
script	build_ip.tcl
	build_sw.sh
software/	*<Other software files and folder>*
software/dp_demo/	dp_demo.elf
	main.c
	config.h
	rx_utils.c
	debug.c
	<Other software files and folder>
software/dp_demo_bsp/	alt_sys_init.c
	linker.h
	system.h
	<Other software files and folder>
software/dp_demo_bsp/	*<Other software files and folder>*
software/dp_demo_bsp/	*<Other software files and folder>*
software/dp_demo_bsp/	*<Other software files and folder>*
Additional_Files/Master_Image/Rev_11	a10_dp_demo.sof
	dp_demo.elf

1.4.3. Compiling the Design

You can download the DisplayPort 8K RX-only design file (A10_DP_RX_FMC_PRO.par) from the Intel Design Store. To compile and run a demonstration test on the hardware design example, follow these steps.

The .par file includes pre-compiled .sof files that you can run to test the design.

1. Unzip the Additional_Files.zip file from the A10_DP_RX_FMC_PRO.par file, and move the Script and Software folder to the main project directory.
2. Launch the Intel Quartus Prime Pro Edition software and open *<project directory>*/top.qpf.

Note: Bitec DisplayPort FMC daughter card revision 9 and later includes a retimer component at RX to support HBR3 rate. To receive an 8K video resolution, change the local parameter, `BITEC_DP_CARD_REV`, in the top-level RTL file of the design example at <*project directory*>`/rtl/top.v` file to revision 9 or later. Similarly, make the same changes in the <*project directory*>`/software/dp_demo/config.h`.

```
localparam BITEC_DP_CARD_REV = 2;

// 0 = Bitec FMC DP card rev.4 - 8,

// 1 = rev.9 - 10

// 2 = rev.11
```

3. Open Nios II Command Shell and navigate to the `Script` folder.
4. Run the `build_sw_sh` script in the Nios II terminal to build the software.
5. In the Intel Quartus Prime Pro Edition software, click **Processing ➤ Start Compilation**.
6. After successful compilation, the Intel Quartus Prime Pro Edition software generates a `.sof` file in your specified directory.

Related Information

Intel Design Store
Provides the design files.

1.4.3.1. Regenerating and Downloading ELF File

By default, the ELF file is generated when you generate the dynamic design example.

Note: In some cases, you need to regenerate the ELF file if you modify a software file such as the `main.c` or `config.h` files or modify and regenerate the `dp_core.qsys` file. Regenerating the `dp_core.qsys` file updates the `.sopcinfo` file, which requires you to regenerate the ELF file.

1. Unzip the `Additional_Files.zip` file.
2. Go to <*project directory*>`/software` and edit the code if necessary.
3. Go to <*project directory*>`/script` and execute the following build script:

   `source build_sw.sh`

   - On Windows, search and open Nios II Command Shell. In the Nios II Command Shell, go to <*project directory*>`/script` and execute `source build_sw.sh`.
   - On Linux, launch the Platform Designer, and open **Tools ➤ Nios II Command Shell**. In the Nios II Command Shell, go to <*project directory*>`/script` and execute `source build_sw.sh`.
4. Make sure an `.elf` file is generated in <*project directory*>`/software/dp_demo`.
5. Download the generated `.elf` file into the FPGA without recompiling the `.sof` file by running the following script:

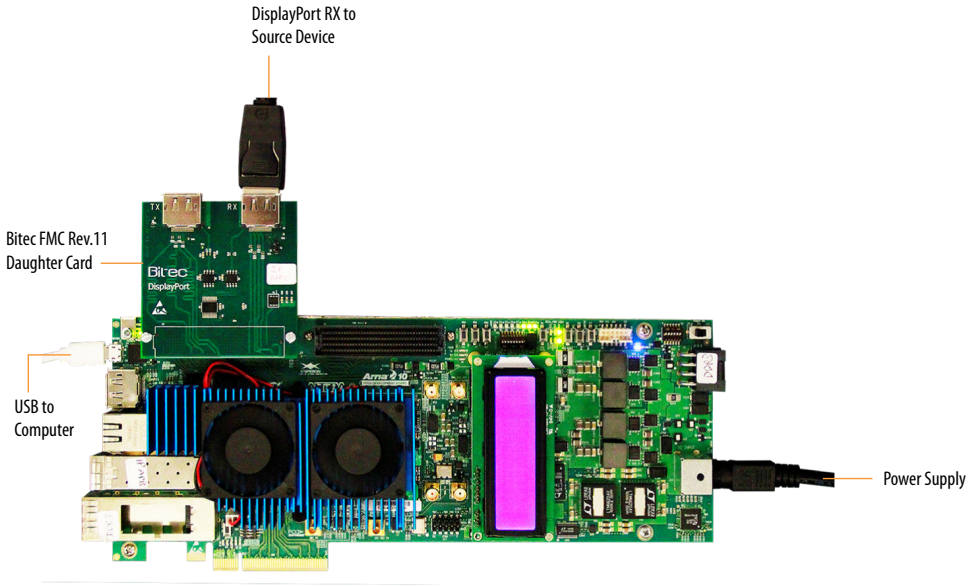
```
nios2-download <project directory>/software/dp_demo/*.elf
```

6. Push the reset button on the FPGA board for the new software to take effect.

1.4.4. Running the Design on Hardware

Set up the hardware before you run the design on the Intel Arria 10 development kit.

Figure 2. Intel Arria 10 Development Kit Setup



1. Install the Bitec FMC daughter card at the FMC port A on the Intel Arria 10 development kit.
2. Connect the DisplayPort RX connector on the Bitec FMC daughter card to a video source such as GPU.
3. Ensure all switches on the development board are in default position.
4. Power up and connect the development board to your PC using a micro USB cable.
5. Download the `.sof` file into the FPGA device using Intel Quartus Prime Programmer.

   Note: If you regenerate the `.elf` file without recompiling the Intel Quartus Prime project, you need to download the `.elf` file after download the `.sof` file. The reference design contains a pre-compiled the `.sof` file that you can use to program the Intel Arria 10 development kit. You need to unzip the `Additional_Files.zip` and locate the `a10_dp_demo.sof` in the `Master_Image/Rev11` directory.
6. Push the **Reset** button on the Intel Arria 10 development kit.
7. Run the command below in the Nios II Command Shell to interface with in-design Nios II core.

```
Nios2-terminal
```

8. To view the MSA information, type 'S' on the keyboard while in the Nios II terminal. You can also view the MSA information by pressing the push button 0 on the Intel Arria 10 development kit.

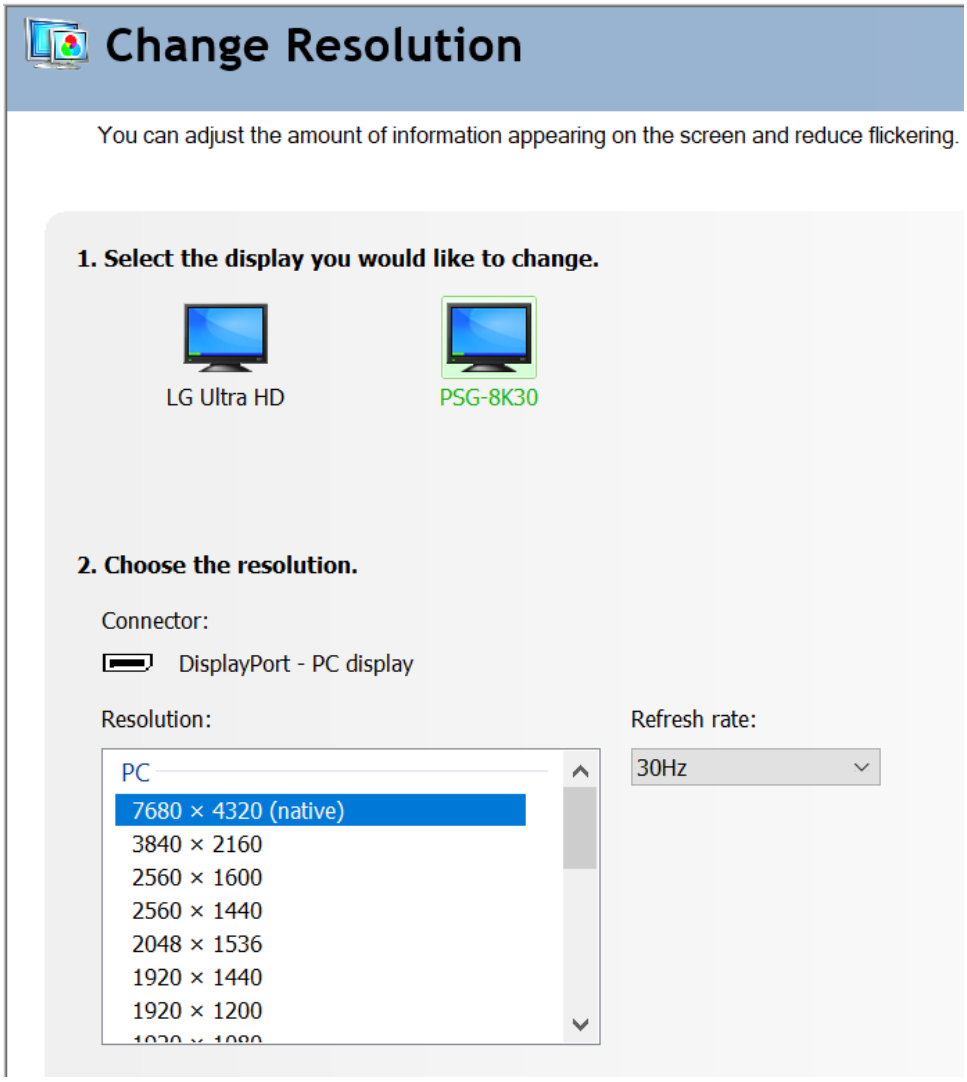
Figure 3. DisplayPort RX-only Design MSA Information

The MSA information tells you that the DisplayPort RX has successfully received the video data without any bit error rate (BER). The MSA information also tells you that the DisplayPort RX receives 7680x4320 video resolution using 4 channels, 8100 Mbps (HBR3) per channel with 8 bpc RGB colorimetry (MISC0 = 20). MSA lock = 1 indicates that the video stream received by the sink is a valid video stream.

```
----------------------------------------
------    RX Main stream attributes    -----
----------------------------------------
--- Stream 0 ---
VB-ID lock : 1   MSA lock : 1
VB-ID : 00  MISC0 : 20  MISC1 : 00
Mvid    : A166     Nvid     : 8000
Htotal  : 7760     Vtotal   : 4381
HSP     : 0000     HSW      : 0032
Hstart  : 0072     Vstart   : 0014
VSP     : 0001     VSW      : 0008
Hwidth  : 7680     Vheight  : 4320
CRC R : 3889  CRC G : 3608  CRC B : fbda
--- 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: F
Lane count : 4
Link rate  : 8100 Mbps
BER0    : 0000     BER1     : 0000
BER2    : 0000     BER3     : 0000
```

Figure 4. Sink Device Detected by the GPU Control Panel

This design used the Nvidia GeForce GTX 1080 GPU for verification. This figure below is a screen-shot of the Nvidia GPU control panel environment.



1.4.5. Design Debug Features

This design also offers debugging features that are useful for debugging link up and no video output issues.

1.4.5.1. Main Stream Attribute (MSA) Information

This debug feature enables you to check the MSA information.

This feature is a part of the DisplayPort RX-only design example. To display the (MSA of the DisplayPort RX core, type 'S' in the Nios II terminal. The RX stream MSA values will appear on the Nios II terminal.

1.4.5.2. Auxiliary Channel Traffic Monitor

This debug feature enables you to check the auxiliary channel transaction.

This feature is also a part of the DisplayPort RX-only design example. To display the auxiliary channel transaction on the Nios II terminal, set the BITEC_AUX_DEBUG flag in the `config.h` file 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 and download the ELF image into the FPGA.

Note: This design also has the **Enable AUX Debug Stream** parameter, in the DisplayPort IP parameter editor, turned on to enable the AUX channel traffic monitor feature.

1.4.5.3. Signal Tap Logic Analyzer

You can view the MSA information and other DisplayPort signals without accessing the Nios II terminal, using the Signal Tap Logic Analyzer.

Click on the **Tools ➤ Signal Tap Logic Analyzer** in the Intel Quartus Prime Pro Edition software and navigate to the preloaded Signal Tap file that comes together with the design example.

Perform a full compilation and view the signals on the **Signal Tap Logic Analyzer** window.

1.4.5.4. Onboard User LED Functions

This debug feature allows you to observe the link rate, number of channels used, and link training status through the onboard LEDs of the development kit.

Table 10. LED Functions

LEDs	Functions
USER_LED[0]	This LED indicates that the source is lane-trained successfully. At this point, the IP asserts the `rx_vid_locked` signal.
USER_LED[5:1]	These LEDs illuminate the design lane counts. • 4'b0001 = 1 lane • 4'b0010 = 2 lanes • 4'b0100 = 4 lanes
USER_LED[7:6]	These LEDs indicate the RX link rates. • 2'b00 = RBR • 2'b01 = HBR • 2'b10 = HBR2 • 2'b11 = HBR3

1.5. Creating the RX-only Design

You can create the DisplayPort RX-only design by making certain software and hardware modifications to the already provided DisplayPort SST parallel loopback with PCR design example.

1.5.1. Generating the Design

Before you make the modifications, first you need to generate the DisplayPort SST Parallel Loopback design example in the Intel Quartus Prime Pro Edition software.

1. Instantiate the DisplayPort Intel FPGA IP and specify the parameters as listed below.

Table 11. DisplayPort SST Parallel Loopback Parameters

Parameters	Value	Description
Maximum video output color depth (TX)	16 bpc	This design supports GPU and monitors up to a maximum of 16 bit-per-color depth. However, the design depends highly on the GPU's capability to transmit the color depth. *Note:* Intel source devices are capable of transmitting 8Kp30 with 8 bpc settings.
Maximum link rate	8.1 Gbps	The bandwidth requirement for 8Kp30 and 8 bpc video stream through serial link: • Active video resolution = 7680 × 4320 pixels per frame • Total resolution (including reduced blanking) = 7760 × 4381 pixels per frame • Refresh rate = 30 Hz or 30 frames per second • Bits per pixel = 8 bpc × 3 colors = 24 bits per pixel • Total bandwidth = (7760 × 4381) pixels per frame × 30 frames per second × 24 bits per pixel = 24.477 Gbps With 8B/10B encoding scheme, the actual bandwidth required = 24.477 Gbps × 10/8 = 30.59 Gbps. With 4 lanes at 8.1 Gbps, the aggregated bandwidth of 32.4 Gbps is sufficient to support the 8K video stream at 30 Hz refresh rate.
Maximum lane count	4	
Symbol output mode (Source)	Quad	Symbol mode affects the transceiver parallel bus width and the DisplayPort IP clock frequency. The DisplayPort IP synchronizes with the transceiver parallel clock. The parallel clock frequency is link rate/transceiver parallel bus width. Frequency for HBR3 (8.1 Gbps) is 8100/40 or 202.5 MHz for quad (40 bits) mode.
Symbol input mode (Sink)		
Pixel input mode (Source)	Quad	Pixel mode affects the video clock frequency and video port width of the IP core. For 8Kp30 video stream, the bandwidth requirement is 7760 (H-total) × 4381 (V-total) × 30 frames per second = 1019896800 pixels per second. Because of the high bandwidth requirement, the design requires quad pixel mode for timing closure. • Single (1 pixel/clock): 1019.89 MHz • Dual (2 pixels/clock) 509.95 MHz • Quad (4 pixels/clock) 254.97 MHz *Note:* This design uses 300 MHz for the video clock generated from PLL.
Pixel output mode (Sink)		

continued...

Parameters	Value	Description
Support analog reconfiguration	On	Enable analog reconfiguration interface. Used to reconfigure vod and pre-emphasis value.
Enable AUX debug stream	On	Enable AUX source traffic output to the Avalon streaming port
DisplayPort SST Parallel Loopback With PCR	On	Enable Pixel Clock Recovery in the design.

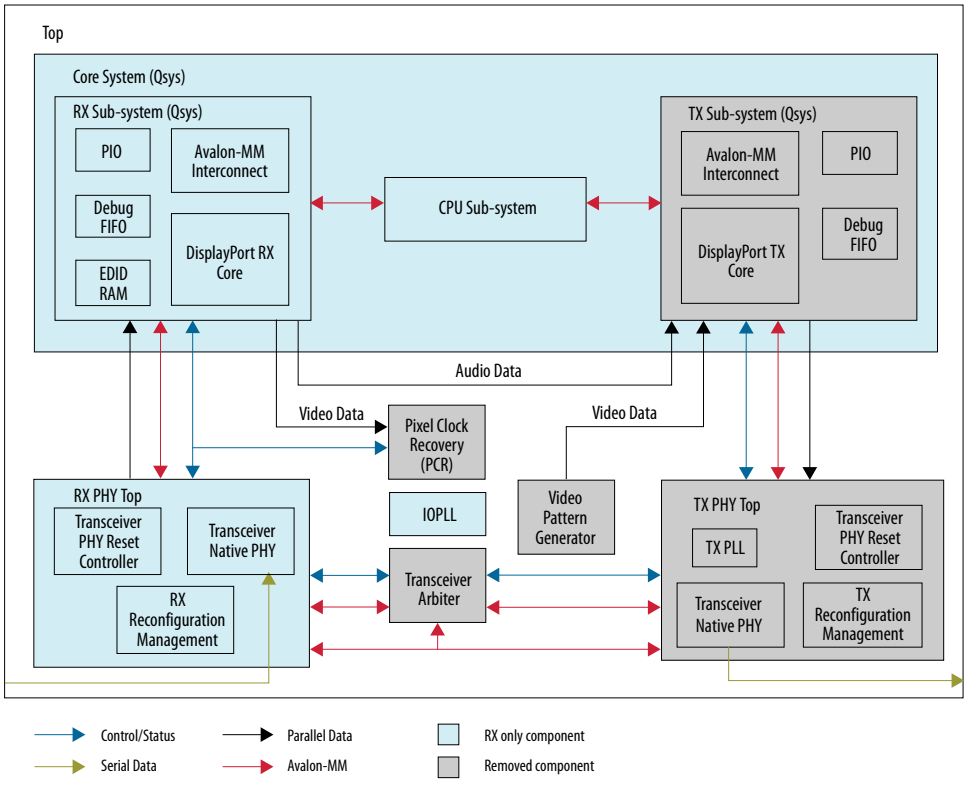
2. Click **Generate Example Design**.

1.5.2. Removing Irrelevant Blocks

The generated DisplayPort SST Parallel Loopback design example consists of TX and RX components. Modify the generated design example by removing the irrelevant blocks from the top-level design and from the `dp_core.qsys` file.

Remove the TX sub-system and TX PHY top components, and the Pixel Clock Recovery (PCR) and Transceiver Arbiter blocks (in gray), as shown in the diagram below. These blocks are not needed for the RX-only design.

Figure 5. Components Required for the DisplayPort RX-only Design



1.5.3. Making a Direct Connection to the RX Transceiver Block

The existing dynamic DisplayPort parallel SST loopback with PCR design example uses the Transceiver Arbiter block to share between an RX and TX Native PHY transceiver within the same channel. As the RX-only design only requires the RX transceiver, you need to remove the Transceiver Arbiter and make a direct connection to the RX transceiver.

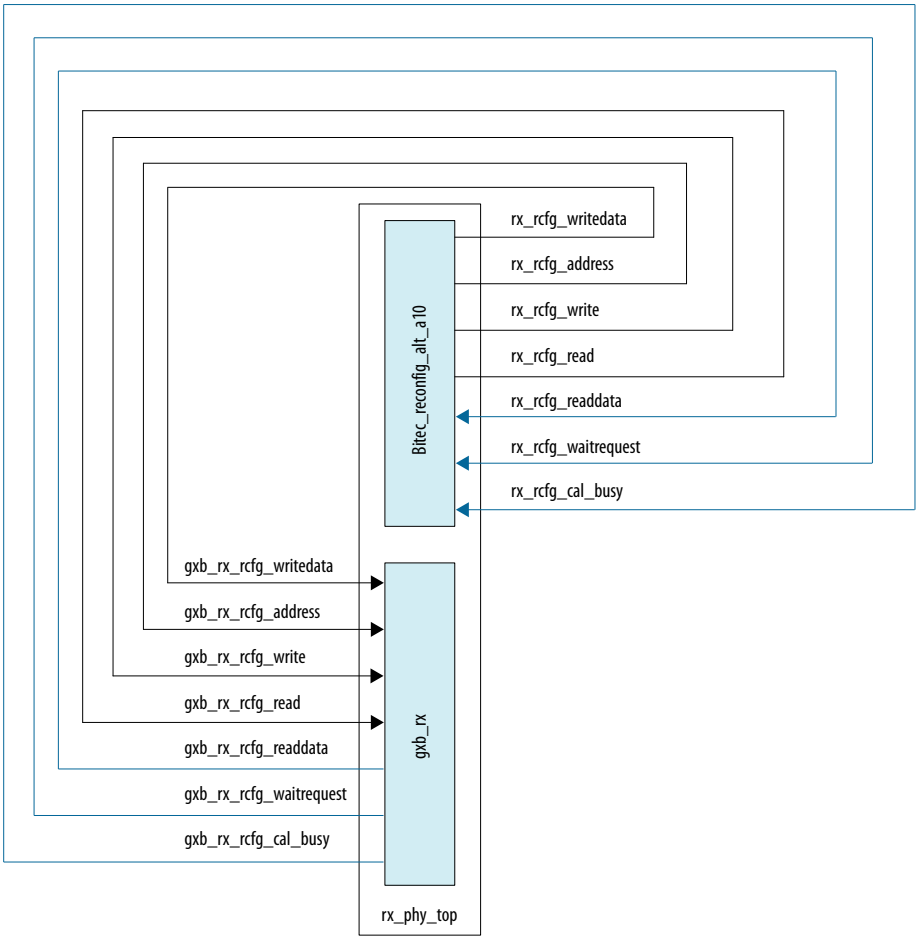
1. Before you make the connection, in the Platform Designer turn on the **Shared Reconfiguration Interface** parameter in the Transceiver Native PHY block to allow for single Avalon memory-mapped slave interface for dynamic reconfiguration of all channels.
2. Update the width of the transceiver signals as shown below in the design top-level and the `rx_phy_top.v` files.

Table 12. RX Transceiver Signals

Signal	Direction	Width (Bit)
`gxb_rx_rcfg_write`	Input	1
`gxb_rx_rcfg_read`	Input	1
`gxb_rx_rcfg_address`	Input	12
`gxb_rx_rcfg_writedata`	Input	32
`gxb_rx_rcfg_readdata`	Output	32
`gxb_rx_rcfg_waitrequest`	Output	1
`gxb_rx_rcfg_cal_busy`	Output	1

3. Make a direct connection from the RX Reconfiguration Management block to the RX Transceiver Native PHY block in the `rx_phy_top.v` file as shown in the diagram below.

Figure 6. Bitec Reconfig and RX Transceiver Block Connection



4. Remove the following `Transceiver Reconfig Group` assignments from the Intel Quartus Prime Settings File (.qsf).
   - `- set_instance_assignment -name XCVR_RECONFIG_GROUP 1 -to fmca_dp_m2c_p[0] -entity a10_dp_demo`
   - `- set_instance_assignment -name XCVR_RECONFIG_GROUP 2 -to fmca_dp_m2c_p[1] -entity a10_dp_demo`
   - `- set_instance_assignment -name XCVR_RECONFIG_GROUP 3 -to fmca_dp_m2c_p[2] -entity a10_dp_demo`
   - `- set_instance_assignment -name XCVR_RECONFIG_GROUP 4 -to fmca_dp_m2c_p[3] -entity a10_dp_demo`

1.5.4. Selecting the Bitec FMC Daughter Card Revision

Make sure that the Bitec daughter card revision is updated accordingly.

To update the Bitec daughter card revision, edit the top-level project file and in the `config.h` software file. This design uses Bitec daughter card revision 11 to support HBR3 data rate.

```
localparam BITEC_DP_CARD_REV = 2;

// 0 = Bitec FMC DP card rev.4 - 8,

// 1 = rev.9 - 10

// 2 = rev.11
```

1.5.5. Modifying the Software

After removing the irrelevant blocks, reconnecting the remaining blocks, and selecting the Bitec FMC daughter card revision, modify the software.

1. First, modify the software's `config.h` file. Navigate to the design example folder and change the values of the following parameter settings in the file.

Table 13. Config.h Parameter Settings

Parameter	Value	Description
BITEC_AUX_DEBUG	0	Set to 1 to enable AUX channel traffic monitoring.
BITEC_STATUS_DEBUG	1	Set to 1 to enable MSA and link status monitoring.
DP_SUPPORT_RX	1	Set to 1 if the DisplayPort supports RX.
BITEC_RX_GPUMODE	1	Set to 1 to enable sink GPU mode.
BITEC_RX_CAPAB_MST	0	Set to 1 to enable MST support.
BITEC_RX_FAST_LT_SUPPORT	0	Set to 1 to enable Fast Link Training support.
BITEC_RX_LQA_SUPPORT	0	Set to 1 to enable Link Quality Analysis support.
BITEC_EDID_800X600_AUDIO	0	Set to 1 to use an EDID with maximum resolution of 800 x 600
BITEC_DP_0_AV_RX_CONTROL_BITEC_CFG_RX_SUPPORT_MST	0	Set to 1 to enable MST support
DP_SUPPORT_TX	0	Set to 1 if DisplayPort supports TX
BITEC_TX_CAPAB_MST	0	Set to 1 to enable MST support
TX_VIDEO_IM_ENABLE	0	Set to 1 to enable TX Video IM interface
DP_SUPPORT_EDID_PASSTHRU	0	Set to 1 to enable EDID passthrough from sink to source.
BITEC_DP_CARD_REV	2	• Set to 0 = Bitec FMC DisplayPort daughter card revision 4 – 8 (without Paradetech Retimer) • Set to 1 = Bitec FMC DisplayPort daughter card revision 10 (with Paradetech Retimer) • Set to 2 = Bitec FMC DisplayPort daughter card revision 11 (with Megachip Retimer)

continued...

Parameter	Value	Description
MST_RX_STREAMS	0	RX MST number of streams
MST_TX_STREAMS	0	TX MST number of streams
PSG_8K_EDID	1	Set to 1 is sink supports 8K video.

2. Next, open the `main.c` file located in the `software/dp_demo` folder, and remove any TX-related components, such as the following:
   - `#include “tx_utils.h”`
   - `bitec_dptx_init();`
   - `bitec_dp_dump_source_msa(btc_dptx_baseaddr(0));`
   - `bitec_dp_dump_source_config(btc_dptx_baseaddr(0));`

   Note: Remove the `tx_utils.c` and `tx_utils.h` files from the software folder to avoid potential software build errors.
3. Next, for debugging purposes, modify the `debug.c` file located in the `software/dp_demo` folder. Open the `debug.c` file and remove the `void bitec_dp_dump_source_msa()` and `void bitec_dp_dump_source_config()` functions.
4. After modifying the software, rebuild the software as instructed in Regenerating and Downloading ELF File on page 9.
5. Finally, compile the project as described in Compiling the Design on page 8.

1.6. Document Revision History for AN 900: Intel Arria 10 DisplayPort 8K RX-only Design

Document Version	Changes
2019.12.16	Initial release.