



# Intel<sup>®</sup> FPGA HDMI Design Example User Guide for Intel<sup>®</sup> Cyclone 10 GX Devices

Updated for Intel<sup>®</sup> Quartus<sup>®</sup> Prime Design Suite: **18.1**



**Subscribe**

**Send Feedback**

**UG-20125 | 2018.10.25**

Latest document on the web: [PDF](#) | [HTML](#)



## Contents

---

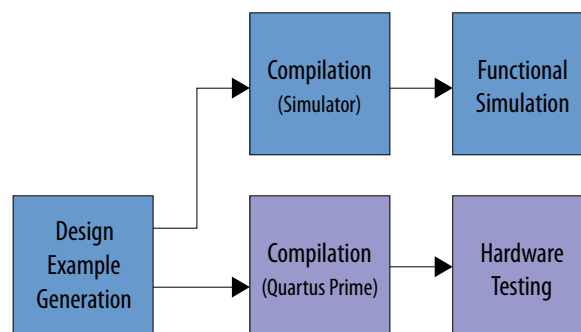
<b>1. Intel® FPGA HDMI IP Design Example Quick Start Guide for Intel® Cyclone® 10 GX Devices.....</b>	<b>3</b>
1.1. Directory Structure.....	3
1.2. Hardware and Software Requirements.....	7
1.3. Generating the Design.....	7
1.4. Simulating the Design.....	8
1.5. Compiling and Testing the Design .....	9
1.6. Design Limitation.....	10
1.7. Intel FPGA HDMI IP Design Example Parameters.....	10
<b>2. Intel FPGA HDMI IP Design Example Detailed Description.....</b>	<b>12</b>
2.1. HDMI RX-TX Retransmit Design Block Diagram.....	12
2.2. Design Components.....	13
2.3. Dynamic Range and Mastering (HDR) InfoFrame Insertion and Filtering.....	19
2.4. Clocking Scheme.....	22
2.5. Interface Signals.....	25
2.6. Design RTL Parameters.....	35
2.7. Hardware Setup.....	36
2.8. Simulation Testbench.....	37
2.9. Upgrading Your Design.....	39
<b>A. Revision History for Intel FPGA HDMI Design Example User Guide for Intel Cyclone 10 GX Devices.....</b>	<b>43</b>

# 1. Intel® FPGA HDMI IP Design Example Quick Start Guide for Intel® Cyclone® 10 GX Devices

The Intel® FPGA HDMI IP design example for Intel Cyclone® 10 GX devices features a simulating testbench and a hardware design that supports compilation and hardware testing.

When you generate a design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware.

**Figure 1. Development Steps**



## Related Information

[Intel FPGA HDMI IP Core User Guide](#)

## 1.1. Directory Structure

The directories contain the generated files for the Intel FPGA HDMI design example.

Figure 2. Directory Structure for the Design Example

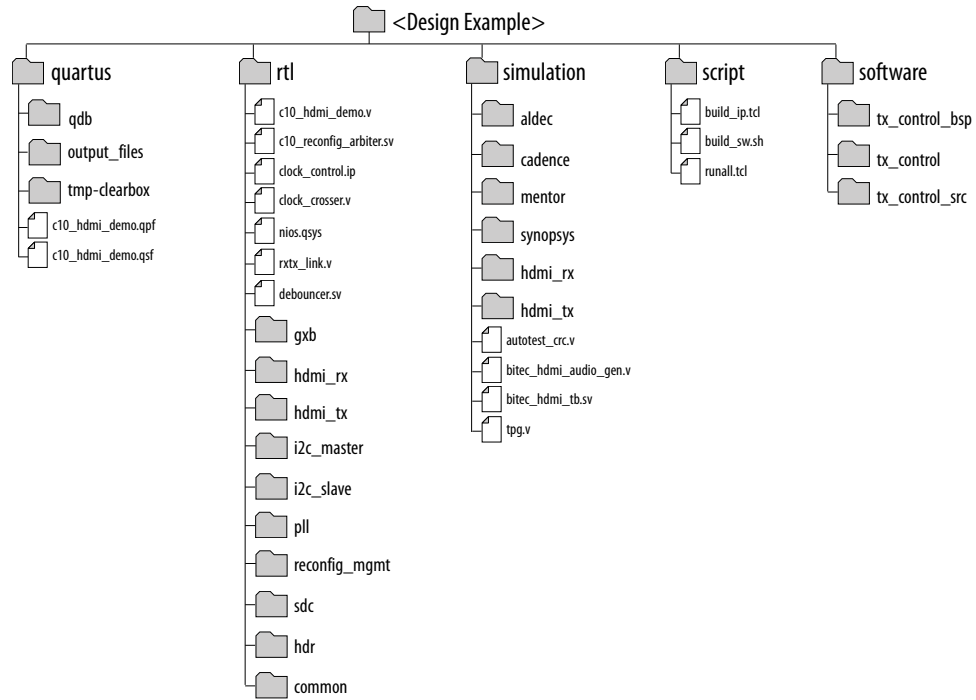


Table 1. Generated RTL Files

Folders	Files
gxb	/gxb_rx.ip
	/gxb_rx_reset.ip
	/gxb_tx.ip
	/gxb_tx_fpll.ip
	/gxb_tx_reset.ip
hdmi_rx	/hdmi_rx.ip
	/hdmi_rx_top.v
	/mr_clock_sync.v
	/mr_hdmi_rx_core_top.v
	/mr_rx_oversample.v
	/symbol_aligner.v
hdmi_tx	/hdmi_tx.ip
	/hdmi_tx_top.v
	/mr_ce.v
	/mr_hdmi_tx_core_top.v
	/mr_tx_oversample.v

*continued...*



Folders	Files
i2c_master	/i2c_master_bit_ctrl.v
	/i2c_master_byte_ctrl.v
	/i2c_master_defines.v
	/i2c_master_top.v
	/oc_i2c_master.v
	/oc_i2c_master_hw.tcl
	/timescale.v
i2c_slave	/edid_ram.ip
	/I2Cslave.v
	/output_buf_i2c.ip
	/Panasonic.hex
	/i2c_avl_mst_intf_gen.v
	/i2c_clk_cnt.v
	/i2c_condt_det.v
	/i2c_databuffer.v
	/i2c_rxshifter.v
	/i2c_slvfsm.v
	/i2c_spksupp.v
	/i2c_txout.v
	/i2c_txshifter.v
	/i2cslave_to_avlmm_bridge.v
pll	/pll_hdmi.ip
	/pll_hdmi_reconfig.ip
common	/reset_controller
hdr	/altera_hdmi_aux_hdr.v
	/altera_hdmi_aux_snk.v
	/altera_hdmi_aux_src.v
	/altera_hdmi_hdr_infotrame.v
	/avalon_st_muxplexer.v
reconfig_mgmt	/mr_compare_pll.v
	/mr_compare_rx.v
	/mr_rate_detect.v
	/mr_reconfig_master_pll.v
	/mr_reconfig_master_rx.v

*continued...*



Folders	Files
	/mr_reconfig_mgmt.v
	/mr_rom_pll_dprioaddr.v
	/mr_rom_pll_valuemask_8bpc.v
	/mr_rom_pll_valuemask_10bpc.v
	/mr_rom_pll_valuemask_12bpc.v
	/mr_rom_pll_valuemask_16bpc.v
	/mr_rom_rx_dprioaddr_bitmask.v
	/mr_rom_rx_valuemask.v
	/mr_state_machine.v
sdc	/c10_hdmi2.sdc
	/mr.sdc
	/jtag.sdc

**Table 2. Generated Simulation Files**

Folders	Files
aldec	/aldec.do
	/rivierapro_setup.tcl
cadence	/cds.lib
	/hdl.var
	/ncsim.sh
	/ncsim_setup.sh
	<cds_libs folder>
mentor	/mentor.do
	/msim_setup.tcl
synopsys	/vcs/filelist.f
	/vcs/vcs_setup.sh
	/vcs/vcs_sim.sh
	/vcsmx/vcsmx_setup.sh
	/vcsmx/vcsmx_sim.sh
	/vcsmx/synopsys_sim_setup
hdmi_rx	/hdmi_rx.ip
hdmi_tx	/hdmi_tx.ip



**Table 3. Generated Software Files**

Folders	Files
tx_control_src	/i2c.c
	/i2c.h
	/ti_i2c.c
	/ti_i2c.h
	/main.c
	/xcvr_gpll_rcfg.c
	/xcvr_gpll_rcfg.h

*Note:* The tx\_control and tx\_control\_bsp folders contain auto-generated files. The tx\_control folder will also contain duplicates of the tx\_control\_src files.

## 1.2. Hardware and Software Requirements

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

### Hardware

- Intel Cyclone 10 GX FPGA Development Kit
- HDMI Source (Graphics Processor Unit (GPU))
- HDMI Sink (Monitor)
- Bitec HDMI FMC 2.0 daughter card (Revision 6.0)
- HDMI cables

*Note:* You can select the revision of the Bitec HDMI daughter card by setting the local parameter BITEC\_DAUGHTER\_CARD\_REV to 4, 6, or 11 in the top-level file (c10\_hdmi2\_demo.v). When you change the revision, the design may swap the transceiver channels and invert the polarity according to the Bitec HDMI daughter card requirements. If you set the BITEC\_DAUGHTER\_CARD\_REV parameter to 0, the design will not make any changes to the transceiver channels and the polarity.

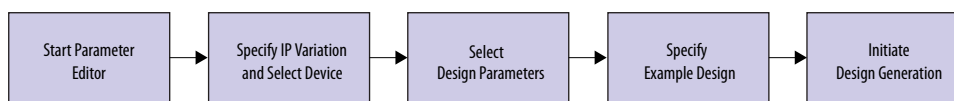
### Software

- Intel Quartus® Prime Pro Edition version 18.1 (for hardware testing)
- ModelSim\* - Intel FPGA Edition, ModelSim - Intel FPGA Starter Edition, NCSim, Riviera-PRO\*, VCS\* (Verilog HDL only)/VCS MX, or Xcelium\* Parallel simulator

## 1.3. Generating the Design

Use the Intel FPGA HDMI parameter editor in the Intel Quartus Prime Pro Edition software to generate the design examples.

**Figure 3. Generating the Design Flow**



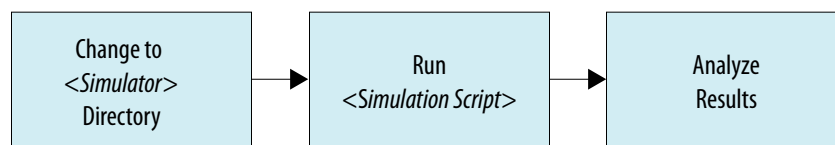


1. Create a project targeting Intel Cyclone 10 GX device family and select the desired device.
2. In the IP Catalog, locate and double-click **Intel FPGA HDMI IP Core**. The **New IP Variant** or **New IP Variation** window appears.
3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named `<your_ip>.ip`.
4. Click **OK**. The parameter editor appears.
5. On the **IP** tab, configure the desired parameters for both TX and RX.
6. On the **Design Example** tab, select **Cyclone 10 HDMI RX-TX Retransmit**.
7. Select **Simulation** to generate the testbench, and select **Synthesis** to generate the hardware design example.  
You must select at least one of these options to generate the design example files. If you select both, the generation time is longer.
8. For **Generate File Format**, select **Verilog** or **VHDL**.
9. For **Target Development Kit**, select **Intel Cyclone 10 GX FPGA Development Kit**. If you select a development kit, then the target device (selected in **step 4**) changes to match the device on target board. For **Intel Cyclone 10 GX FPGA Development Kit**, the default device is 10CX220YF780E5G.
10. Click **Generate Example Design**.

## 1.4. Simulating the Design

The HDMI testbench simulates a serial loopback design from a TX instance to an RX instance. Internal video pattern generator and audio pattern generator modules drive the HDMI TX instance and the serial output from the TX instance connects to the RX instance in the testbench.

**Figure 4. Design Simulation Flow**



1. Go to the desired simulation folder.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator.
3. Analyze the results.

**Table 4. Steps to Run Simulation**

Simulator	Working Directory	Instructions
Riviera-PRO	/simulation/aldec	In the command line, type <pre>vsim -c -do aldec.do</pre>
NCSim	/simulation/cadence	In the command line, type <pre>source ncsim.sh</pre>

*continued...*



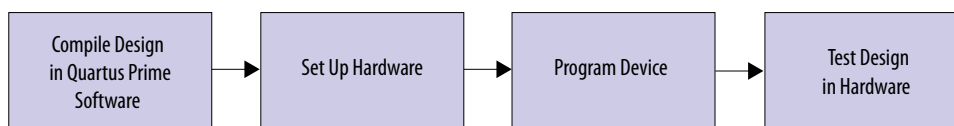


Simulator	Working Directory	Instructions
ModelSim	/simulation/mentor	In the command line, type <pre>vsim -c -do mentor.do</pre>
VCS	/simulation/synopsys/vcs	In the command line, type <pre>source vcs_sim.sh</pre>
VCS MX	/simulation/synopsys/vcsmx	In the command line, type <pre>source vcsmx_sim.sh</pre>
Xcelium Parallel	/simulation/xcelium	In the command line, type <pre>source xcelium_sim.sh</pre>

A successful simulation ends with the following message:

```
# SYMBOLS_PER_CLOCK = 2
# VIC = 0
# AUDIO_CLK_DIVIDE = 800
# TEST_HDMI_6G = 1
# Simulation pass
```

## 1.5. Compiling and Testing the Design



To compile and run a demonstration test on the hardware example design, follow these steps:

1. Ensure hardware example design generation is complete.
2. Launch the Intel Quartus Prime Pro Edition software and open `<project directory>/quartus/c10_hdmi_demo.qpf`.
3. Click **Processing > Start Compilation**.
4. After successful compilation, a `.sof` file will be generated in the `<project directory>/quartus/output_files` folder.
5. Connect to the on-board FMC (J2) Bitec HDMI 2.0 FMC Daughter Card (Revision 4).
6. Connect TX (P1) of the Bitec HDMI 2.0 FMC Daughter Card to an external video source.
7. Connect RX (P2) of the Bitec HDMI 2.0 FMC Daughter Card to an external video sink or video analyzer.
8. Ensure all switches on the development board are in default position.
9. Configure the selected Intel Cyclone 10 GX device on the development board using the generated `.sof` file (**Tools > Programmer**).
10. The analyzer should display the video generated from the source.



## Related Information

Intel Cyclone 10 GX FPGA Development Kit

## 1.6. Design Limitation

You need to consider some limitations when instantiating the Intel FPGA HDMI design examples.

- You may encounter longer lock time using the HDMI RX for HDMI 2.0 resolution. This limitation will be resolved in a future release.
- The HDMI RX core does not perform word alignment for HDMI 2.0 resolutions (data rate > 3.4 Gbps). The designs use the transceiver PCS word aligner (`rtl/hdmi_rx/symbol_aligner.v`) and control logic (`rtl/reconfig_mgmt/*`) to achieve fast word alignment.

## 1.7. Intel FPGA HDMI IP Design Example Parameters

**Table 5. Intel FPGA HDMI IP Design Example Parameters for Intel Cyclone 10 GX Devices**

These options are available for Intel Cyclone 10 GX devices only.

Parameter	Value	Description
<b>Available Design Example</b>		
Select Design	Cyclone 10 HDMI RX-TX Retransmit	Select the design example to be generated. The generated design example has preconfigured parameter settings. It does not follow user settings.
<b>Design Example Files</b>		
Simulation	On, Off	Turn on this option to generate the necessary files for the simulation testbench.
Synthesis	On, Off	Turn on this option to generate the necessary files for Intel Quartus Prime compilation and hardware demonstration.
<b>Generated HDL Format</b>		
Generate File Format	Verilog, VHDL	Select your preferred HDL format for the generated design example fileset. <i>Note:</i> This option only determines the format for the generated top level IP files. All other files (e.g. example testbenches and top level files for hardware demonstration) are in Verilog HDL format.
<b>Target Development Kit</b>		
Select Board	No Development Kit, Cyclone 10 GX FPGA Development Kit, Custom Development Kit	Select the board for the targeted design example.



Target Development Kit		
		<ul style="list-style-type: none"> <li>No Development Kit: This option excludes all hardware aspects for the design example. The IP core sets all pin assignments to virtual pins.</li> <li>Cyclone 10 GX FPGA Development Kit: This option automatically selects the project's target device to match the device on this development kit. You may change the target device using the <b>Change Target Device</b> parameter if your board revision has a different device variant. The IP core sets all pin assignments according to the development kit.</li> <li>Custom Development Kit: This option allows the design example to be tested on a third party development kit with an Intel FPGA. You may need to set the pin assignments on your own.</li> </ul>

Target Device		
Change Target Device	On, Off	Turn on this option and select the preferred device variant for the development kit.

## 2. Intel FPGA HDMI IP Design Example Detailed Description

The Intel FPGA HDMI IP core design example demonstrates one HDMI instance parallel loopback comprising three RX channels and four TX channels.

**Table 6. Intel FPGA HDMI Design Example for Intel Cyclone 10 GX Devices**

Design Example	Data Rate	Channel Mode	Loopback Type
Cyclone 10 HDMI RX-TX Retransmit	< 6,000 Mbps	Simplex	Parallel with FIFO buffer

### Features

- The design instantiates FIFO buffers to perform a direct HDMI video stream passthrough between the HDMI sink and source.
- The design uses LED status for early debugging stage.
- The design comes with RX and TX only options.
- The design demonstrates the insertion and filtering of Dynamic Range and Mastering (HDR) InfoFrame in RX-TX link module.
- The design demonstrates the management of EDID passthrough from an external HDMI sink to an external HDMI source when triggered by a TX hot-plug event.
- The design uses push-button controlled HDMI TX core signals:
  - mode signal to select DVI or HDMI encoded video frame
  - `info_avi[47]`, `info_vsi[61]`, and `audio_info_ai[48]` signals to select auxiliary packet transmission through sidebands or auxiliary data ports

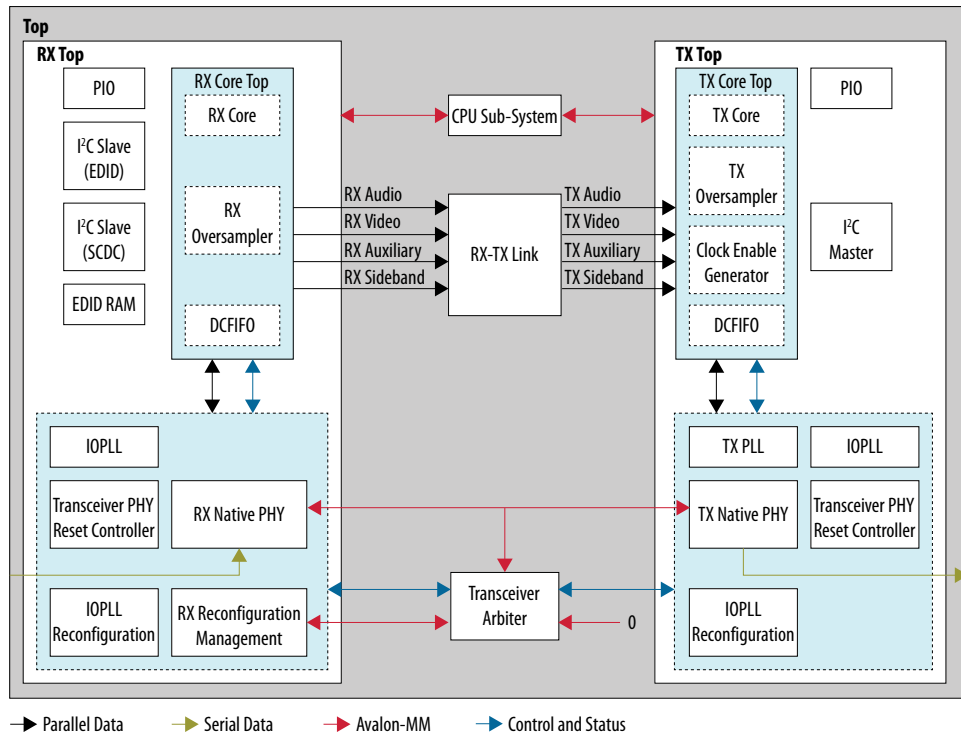
The RX instance receives a video source from the external video generator, and the data then goes through a loopback FIFO before it is transmitted to the TX instance. You need to connect an external video analyzer, monitor, or a television with HDMI connection to the TX core to verify the functionality.

### 2.1. HDMI RX-TX Retransmit Design Block Diagram

The HDMI RX-TX retransmit design example demonstrates parallel loopback on simplex channel mode for Intel FPGA HDMI IP core.



Figure 5. HDMI RX-TX Retransmit Block Diagram



## 2.2. Design Components

The Intel FPGA HDMI IP core design example requires these components.

Table 7. HDMI RX Top Components

Module	Description
RX Core Top	<p>The RX Core top level consists of:</p> <ul style="list-style-type: none"> <li>HDMI RX Core—The IP core receives the serial data from the Transceiver Native PHY and performs data alignment, channel deskew, TMDS decoding, auxiliary data decoding, video data decoding, audio data decoding, and descrambling.</li> <li>RX Oversampler—The RX Oversampler module extracts data from the oversampled incoming data stream when the detected clock frequency band is below the transceiver minimum link rate. The oversampling factor is fixed at 5 and you can program the data width to support different number of symbols. For Intel Cyclone 10 GX devices, the supported data width is 20 bits for 2 symbols per clock. The extracted bit is accompanied by a data valid pulse asserted every 5 clock cycles.</li> <li>DCFIFO —The DCFIFO transfers data from the RX transceiver recovered clock domain to the RX link speed clock domain.</li> </ul>
I <sup>2</sup> C Slave	<p>I<sup>2</sup>C is the interface used for Sink Display Data Channel (DDC) and Status and Data Channel (SCDC). The HDMI source uses the DDC to determine the capabilities and characteristics of the sink by reading the Enhanced Extended Display Identification Data (E-EDID) data structure.</p>

*continued...*

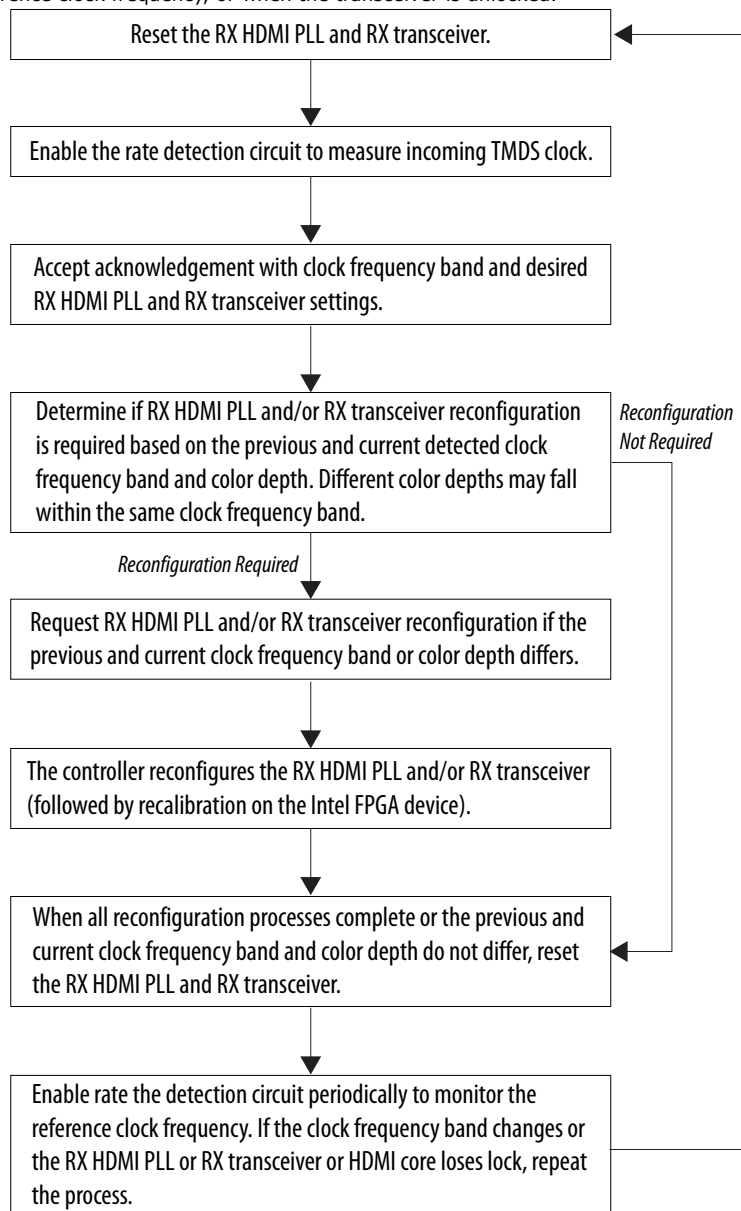


Module	Description
	<ul style="list-style-type: none"> <li>The 8-bit I<sup>2</sup>C slave addresses for E-EDID are 0xA0 and 0xA1. The LSB indicates the access type: 1 for read and 0 for write. When an HPD event occurs, the I<sup>2</sup>C slave responds to E-EDID data by reading from the on-chip RAM.</li> <li>The I<sup>2</sup>C slave-only controller also supports SCDC for HDMI 2.0 operations. The 8-bit I<sup>2</sup>C slave address for the SCDC are 0xA8 and 0xA9. When an HPD event occurs, the I<sup>2</sup>C slave performs write or read transaction to or from SCDC interface of the HDMI RX core.</li> </ul> <p><i>Note:</i> This I<sup>2</sup>C slave-only controller for SCDC is not required if HDMI 2.0b is not intended.</p>
EDID RAM	<p>The design stores the EDID information using the RAM 1-port IP core. A standard two-wire (clock and data) serial bus protocol (I<sup>2</sup>C slave-only controller) transfers the CEA-861-D Compliant E-EDID data structure. This EDID RAM stores the E-EDID information.</p>
IOPLL	<p>The IOPLL generates the RX CDR reference clock, link speed clock, and video clock for the incoming TMDS clock.</p> <ul style="list-style-type: none"> <li>Output clock 0 (CDR reference clock)</li> <li>Output clock 1 (Link speed clock)</li> <li>Output clock 2 (Video clock)</li> </ul> <p><i>Note:</i> The default IOPLL configuration is not valid for any HDMI resolution. The IOPLL is reconfigured to the appropriate settings upon power up.</p>
Transceiver PHY Reset Controller	<p>The Transceiver PHY reset controller ensures a reliable initialization of the RX transceivers. The reset input of this controller is triggered by the RX reconfiguration, and it generates the corresponding analog and digital reset signal to the Transceiver Native PHY block according to the reset sequencing inside the block.</p>
RX Native PHY	<p>Hard transceiver block that receives the serial data from an external video source. It deserializes the serial data to parallel data before passing the data to the HDMI RX core.</p>
RX Reconfiguration Management	<p>RX reconfiguration management that implements rate detection circuitry with the HDMI PLL to drive the RX transceiver to operate at any arbitrary link rates ranging from 250 Mbps to 6,000 Mbps.</p> <p>Refer to <a href="#">Figure 6</a> on page 15 below.</p>
IOPLL Reconfiguration	<p>IOPLL reconfiguration block facilitates dynamic real-time reconfiguration of PLLs in Intel FPGAs. This block updates the output clock frequency and PLL bandwidth in real time, without reconfiguring the entire FPGA. This block runs at 100 MHz in Intel Cyclone 10 GX devices.</p> <p>Due to IOPLL reconfiguration limitation, apply the Quartus INI <code>permit_nf_pll_reconfig_out_of_lock=on</code> during the IOPLL reconfiguration IP generation.</p> <p>To apply the Quartus INI, include <code>"permit_nf_pll_reconfig_out_of_lock=on"</code> in the <code>quartus.ini</code> file and place in the file the Intel Quartus Prime project directory. You should see a warning message when you edit the IOPLL reconfiguration block (<code>pll_hdmi_reconfig</code>) in the Quartus Prime software with the INI.</p> <p><i>Note:</i> Without this Quartus INI, IOPLL reconfiguration cannot be completed if the IOPLL loses lock during reconfiguration.</p>
PIO	<p>The parallel input/output (PIO) block functions as control, status and reset interfaces to or from the CPU sub-system.</p>



**Figure 6. Multi-Rate Reconfiguration Sequence Flow**

The figure illustrates the multi-rate reconfiguration sequence flow of the controller when it receives input data stream and reference clock frequency, or when the transceiver is unlocked.



**Table 8. HDMI TX Top Components**

Module	Description
TX Core Top	The TX Core top level consists of:

*continued...*



Module	Description
	<ul style="list-style-type: none"> <li>HDMI TX Core—The IP core receives video data from the top level and performs TMDS encoding, auxiliary data encoding, audio data encoding, video data encoding, and scrambling.</li> <li>TX Oversampler—The TX Oversampler module transmits data by repeating each bit of the input word a given number of times and constructs the output words. The oversampling factor can be 3, 4, or 5 depending on the TMDS clock frequency. The TX oversampler assumes that the input word is only valid every 3, 4, or 5 clock cycles according to the oversampling factor. This block is enabled when the outgoing data stream is below the minimum link rate of the TX transceiver. When FPLL input reference clock frequency is below 50 MHz, the allowable data rate must be below 1,500 Mbps. (Refer to <a href="#">Table 9</a> on page 17.)</li> <li>DCFIFO —The DCFIFO transfers data from the TX link speed clock domain to the transceiver parallel clock out domain. When the Nios II processor determines the outgoing data stream is below the TX transceiver minimum data rate, the TX transceiver accepts the data from the TX oversampler. Otherwise, the TX transceiver reads the data directly from the DCFIFO with a read request asserted at all times.</li> <li>Clock Enable Generator—A logic that generates a clock enable pulse. This clock enable pulse asserts every 5 clock cycles and serves as a read request signal to clock the data out from the DCFIFO.</li> </ul>
I <sup>2</sup> C Master	<p>I<sup>2</sup>C is the interface used for Sink Display Data Channel (DDC) and Status and Data Channel (SCDC). The HDMI source uses the DDC to determine the capabilities and characteristics of the sink by reading the Enhanced Extended Display Identification Data (E-EDID) data structure.</p> <ul style="list-style-type: none"> <li>As DDC, I<sup>2</sup>C Master reads the EDID from the external sink to configure the EDID information EDID RAM in the HDMI RX Top or for video processing.</li> <li>As SCDC, I<sup>2</sup>C master transfers the SCDC data structure from the FPGA source to the external sink for HDMI 2.0b operation. For example, if the outgoing data stream is above 3,400 Mbps, the Nios II processor commands the I<sup>2</sup>C master to update the TMDS_BIT_CLOCK_RATIO and SCRAMBLER_ENABLE bits of the sink SCDC configuration register to 1.</li> </ul>
IOPLL	<p>The IOPLL supplies the link speed clock and video clock from the incoming TMDS clock.</p> <ul style="list-style-type: none"> <li>Output clock 1 (Link speed clock)</li> <li>Output clock 2 (Video clock)</li> </ul> <p><i>Note:</i> The default IOPLL configuration is not valid for any HDMI resolution. The IOPLL is reconfigured to the appropriate settings upon power up.</p>
Transceiver PHY Reset Controller	<p>The Transceiver PHY reset controller ensures a reliable initialization of the TX transceivers. The reset input of this controller is triggered from the top level, and it generates the corresponding analog and digital reset signal to the Transceiver Native PHY block according to the reset sequencing inside the block.</p> <p>The <code>tx_ready</code> output signal from this block also functions as a reset signal to the Intel FPGA HDMI IP core to indicate the transceiver is up and running, and ready to receive data from the core.</p>
Transceiver Native PHY	<p>Hard transceiver block that receives the parallel data from the HDMI TX core and serializes the data from transmitting it.</p> <p>Reconfiguration interface is enabled in the TX Native PHY block to demonstrate the connection between TX Native PHY and transceiver arbiter. No reconfiguration is performed for TX Native PHY.</p> <p><i>Note:</i> To meet the HDMI TX inter-channel skew requirement, set the TX channel bonding mode option in the Intel Cyclone 10 GX Transceiver Native PHY parameter editor to <b>PMA and PCS bonding</b>. You also need to add the maximum skew (<code>set_max_skew</code>) constraint requirement to the digital reset signal from the transceiver reset controller (<code>tx_digitalreset</code>) as recommended in the <i>Intel Cyclone 10 GX Transceiver PHY User Guide</i>.</p>

*continued...*





Module	Description
TX PLL	The transmitter PLL block provides the serial fast clock to the Transceiver Native PHY block. For this Intel FPGA HDMI design example, fPLL is used as TX PLL.
IOPLL Reconfiguration	<p>IOPLL reconfiguration block facilitates dynamic real-time reconfiguration of PLLs in Intel FPGAs. This block updates the output clock frequency and PLL bandwidth in real time, without reconfiguring the entire FPGA. This block runs at 100 MHz in Intel Cyclone 10 GX devices.</p> <p>Due to IOPLL reconfiguration limitation, apply the Quartus INI <code>permit_nf_pll_reconfig_out_of_lock=on</code> during the IOPLL reconfiguration IP generation.</p> <p>To apply the Quartus INI, include <code>"permit_nf_pll_reconfig_out_of_lock=on"</code> in the <code>quartus.ini</code> file and place in the file the Intel Quartus Prime project directory. You should see a warning message when you edit the IOPLL reconfiguration block (<code>pll_hdmi_reconfig</code>) in the Intel Quartus Prime software with the INI.</p> <p><i>Note:</i> Without this Quartus INI, IOPLL reconfiguration cannot be completed if the IOPLL loses lock during reconfiguration.</p>
PIO	The parallel input/output (PIO) block functions as control, status and reset interfaces to or from the CPU sub-system.

**Table 9. Transceiver Data Rate and Oversampling Factor for Each TMDS Clock Frequency Range**

TMDS Clock Frequency (MHz)	TMDS Bit clock Ratio	Oversampling Factor	Transceiver Data Rate (Mbps)
85–150	1	Not applicable	3400–6000
100–340	0	Not applicable	1000–3400
50–100	0	5	2500–5000
35–50	0	3	1050–1500
30–35	0	4	1200–1400
25–30	0	5	1250–1500

**Table 10. Top-Level Common Blocks**

Module	Description
Transceiver Arbiter	<p>This generic functional block prevents transceivers from recalibrating simultaneously when either RX or TX transceivers within the same physical channel require reconfiguration. The simultaneous recalibration impacts applications where RX and TX transceivers within the same channel are assigned to independent IP implementations.</p> <p>This transceiver arbiter is an extension to the resolution recommended for merging simplex TX and simplex RX into the same physical channel. This transceiver arbiter also assists in merging and arbitrating the Avalon-MM RX and TX reconfiguration requests targeting simplex RX and TX transceivers within a channel as the reconfiguration interface port of the transceivers can only be accessed sequentially.</p> <p>The interface connection between the transceiver arbiter and TX/RX Native PHY/PHY Reset Controller blocks in this design example demonstrates a generic mode that apply for any IP combination using the transceiver arbiter. The transceiver arbiter is not required when only either RX or TX transceiver is used in a channel.</p> <p>The transceiver arbiter identifies the requester of a reconfiguration through its Avalon-MM reconfiguration interfaces and ensures that the corresponding <code>tx_reconfig_cal_busy</code> or <code>rx_reconfig_cal_busy</code> is gated accordingly.</p>

*continued...*

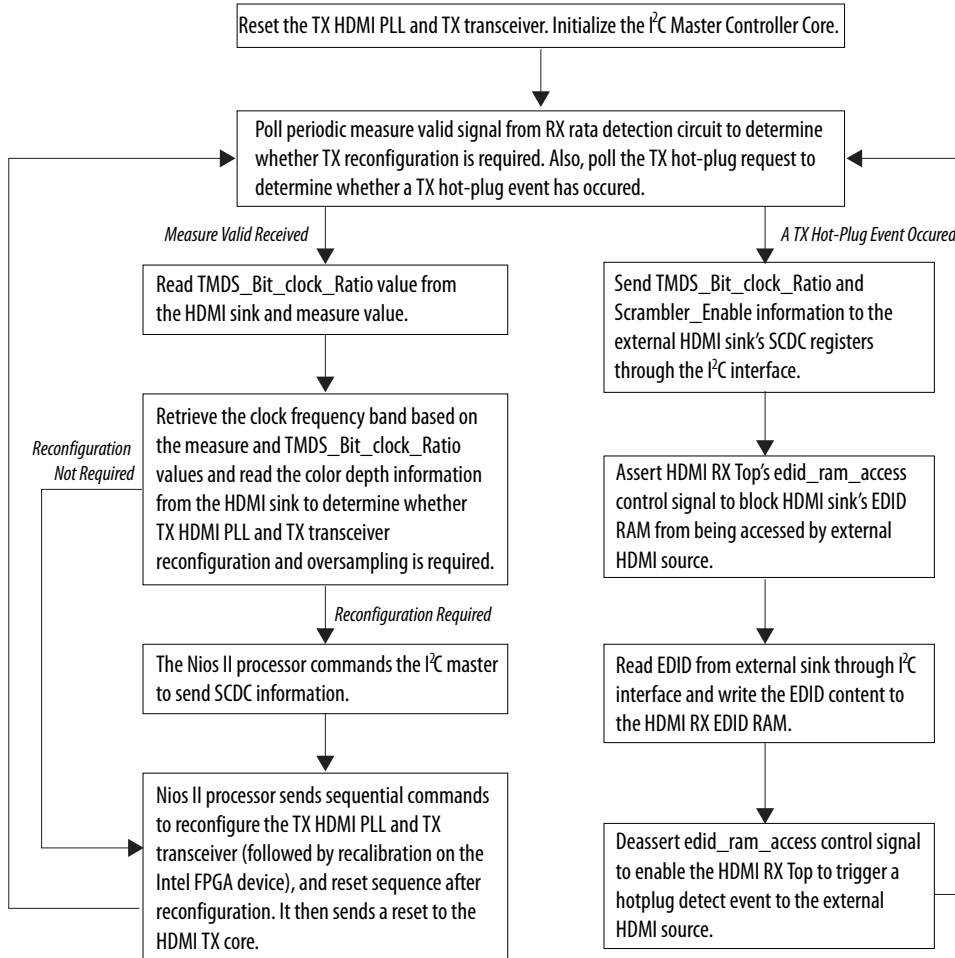


Module	Description
	<p>For HDMI application, only RX initiates reconfiguration. By channeling the Avalon-MM reconfiguration request through the arbiter, the arbiter identifies that the reconfiguration request originates from the RX, which then gates <code>tx_reconfig_cal_busy</code> from asserting and allows <code>rx_reconfig_cal_busy</code> to assert. The gating prevents the TX transceiver from being moved to calibration mode unintentionally.</p> <p><i>Note:</i> Because HDMI only requires RX reconfiguration, the <code>tx_reconfig_mgmt_*</code> signals are tied off. Also, the Avalon-MM interface is not required between the arbiter and the TX Native PHY block. The blocks are assigned to the interface in the design example to demonstrate generic transceiver arbiter connection to TX/RX Native PHY/PHY Reset Controller.</p>
RX-TX Link	<ul style="list-style-type: none"> <li>The video data output and synchronization signals from HDMI RX core loop through a DCFIFO across the RX and TX video clock domains.</li> <li>The General Control Packet (GCP), InfoFrames (AVI, VSI and AI), auxiliary data, and audio data loop through DCFIFOs across the RX and TX link speed clock domains.</li> <li>The auxiliary data port of the HDMI TX core controls the auxiliary data that flow through the DCFIFO through backpressure. The backpressure ensures there is no incomplete auxiliary packet on the auxiliary data port.</li> <li>This block also performs external filtering:               <ul style="list-style-type: none"> <li>Filters the audio data and audio clock regeneration packet from the auxiliary data stream before transmitting to the HDMI TX core auxiliary data port.</li> </ul> <p><i>Note:</i> To disable this filtering, press <code>user_pb[2]</code>. Enable this filtering to ensure there is no duplication of audio data and audio clock regeneration packet in the retransmitted auxiliary data stream.</p> <ul style="list-style-type: none"> <li>Filters the High Dynamic Range (HDR) InfoFrame from the HDMI RX auxiliary data and inserts an example HDR InfoFrame to the auxiliary data of the HDMI TX through the Avalon ST multiplexer.</li> </ul> </li> </ul>
CPU Sub-System	<p>The CPU sub-system functions as SCDC and DDC controllers, and source reconfiguration controller.</p> <ul style="list-style-type: none"> <li>The source SCDC controller contains the I<sup>2</sup>C master controller. The I<sup>2</sup>C master controller transfers the SCDC data structure from the FPGA source to the external sink for HDMI 2.0b operation. For example, if the outgoing data stream is 6,000 Mbps, the Nios II processor commands the I<sup>2</sup>C master controller to update the <code>TMDS_BIT_CLOCK_RATIO</code> and <code>SCRAMBLER_ENABLE</code> bits of the sink TMDS configuration register to 1.</li> <li>The same I<sup>2</sup>C master also transfers the DDC data structure (E-EDID) between the HDMI source and external sink.</li> <li>The Nios II CPU acts as the reconfiguration controller for the HDMI source. The CPU relies on the periodic rate detection from the RX Reconfiguration Management module to determine if the TX requires reconfiguration. The Avalon-MM slave translator provides the interface between the Nios II processor Avalon-MM master interface and the Avalon-MM slave interfaces of the externally instantiated HDMI source's IOPLL and TX Native PHY.</li> <li>The reconfiguration sequence flow for TX is same as RX, except that the PLL and transceiver reconfiguration and the reset sequence is performed sequentially. Refer to <a href="#">Figure 7</a> on page 19.</li> </ul>



**Figure 7. Reconfiguration Sequence Flow**

The figure illustrates the Nios II software flow that involves the controls for I<sup>2</sup>C master and HDMI source.



### 2.3. Dynamic Range and Mastering (HDR) InfoFrame Insertion and Filtering

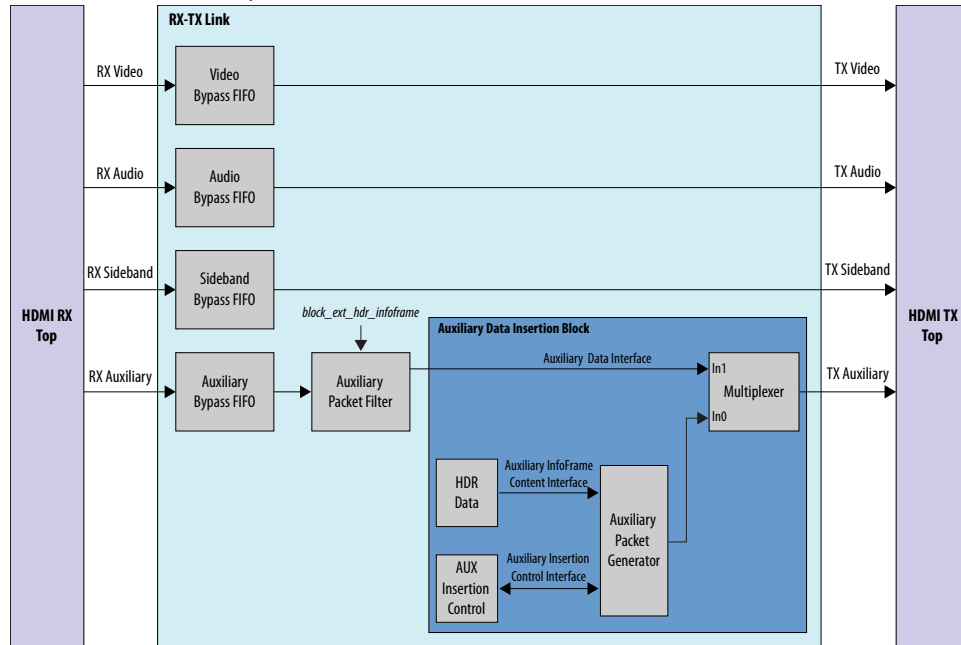
The Intel FPGA HDMI design example includes a demonstration of HDR InfoFrame insertion in a RX-TX loopback system.

*HDMI Specification version 2.0b* allows Dynamic Range and Mastering InfoFrame to be transmitted through HDMI auxiliary stream. In the demonstration, the Auxiliary Data Insertion block supports the HDR insertion. You need only to format the intended HDR InfoFrame packet as specified in the module's signal list table and use the provided AUX Insertion Control module to schedule the insertion of the HDR InfoFrame once every video frame.

In this example configuration, in instances where the incoming auxiliary stream already includes HDR InfoFrame, the streamed HDR content is filtered. The filtering avoids conflicting HDR InfoFrames to be transmitted and ensures that only the values specified in the HDR Sample Data module are used.

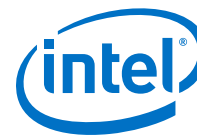
**Figure 8. RX-TX Link with Dynamic Range and Mastering InfoFrame Insertion**

The figure shows the block diagram of RX-TX link including Dynamic Range and Mastering InfoFrame insertion into the HDMI TX core auxiliary stream.



**Table 11. Auxiliary Data Insertion Block (altera\_hdmi\_aux\_hdr) Signals**

Signal	Direction	Width	Description
<b>Clock and Reset</b>			
clk_clk	Input	1	Clock input. This clock should be connected to the link speed clock.
reset_reset_n	Input	1	Reset input.
<b>Auxiliary Packet Generator and Multiplexer Signals</b>			
multiplexer_out_data	Output	72	Avalon streaming output from the multiplexer.
multiplexer_out_valid	Output	1	
multiplexer_out_ready	Output	1	
multiplexer_out_startofpacket	Output	1	
multiplexer_out_endofpacket	Output	1	
multiplexer_out_channel	Output	11	
multiplexer_in_data	Input	72	Avalon streaming input to the In1 port of the multiplexer.
multiplexer_in_valid	Input	1	
multiplexer_in_ready	Input	1	
multiplexer_in_startofpacket	Input	1	
multiplexer_in_endofpacket	Input	1	



Control Signal			
hdmi_tx_vsync	Input	1	HDMI TX Video Vsync. This signal should be synchronized to the link speed clock domain. The core inserts the HDR InfoFrame to the auxiliary stream at the rising edge of this signal.

**Table 12. HDR Data Module (altera\_hdmi\_hdr\_infocore) Signals**

Signal	Direction	Width	Description
hb0	Output	8	Header byte 0 of the Dynamic Range and Mastering InfoFrame: InfoFrame type code.
hb1	Output	8	Header byte 1 of the Dynamic Range and Mastering InfoFrame: InfoFrame version number.
hb2	Output	8	Header byte 2 of the Dynamic Range and Mastering InfoFrame: Length of InfoFrame.
pb	Input	224	Data byte of the Dynamic Range and Mastering InfoFrame.

**Table 13. Dynamic Range and Mastering InfoFrame Data Byte Bundle Bit-Fields**

Bit-Field	Definition	Static Metadata Type 1
7:0	Data Byte 1: {5'h0, EOTF[2:0]}	
15:8	Data Byte 2: {5'h0, Static_Metadata_Descriptor_ID[2:0]}	
23:16	Data Byte 3: Static_Metadata_Descriptor	display primaries_x[0], LSB
31:24	Data Byte 4: Static_Metadata_Descriptor	display primaries_x[0], MSB
39:32	Data Byte 5: Static_Metadata_Descriptor	display primaries_y[0], LSB
47:40	Data Byte 6: Static_Metadata_Descriptor	display primaries_y[0], MSB
55:48	Data Byte 7: Static_Metadata_Descriptor	display primaries_x[1], LSB
63:56	Data Byte 8: Static_Metadata_Descriptor	display primaries_x[1], MSB
71:64	Data Byte 9: Static_Metadata_Descriptor	display primaries_y[1], LSB
79:72	Data Byte 10: Static_Metadata_Descriptor	display primaries_y[1], MSB
87:80	Data Byte 11: Static_Metadata_Descriptor	display primaries_x[2], LSB
95:88	Data Byte 12: Static_Metadata_Descriptor	display primaries_x[2], MSB
103:96	Data Byte 13: Static_Metadata_Descriptor	display primaries_y[2], LSB
111:104	Data Byte 14: Static_Metadata_Descriptor	display primaries_y[2], MSB
119:112	Data Byte 15: Static_Metadata_Descriptor	white_point_x, LSB
127:120	Data Byte 16: Static_Metadata_Descriptor	white_point_x, MSB
135:128	Data Byte 17: Static_Metadata_Descriptor	white_point_y, LSB

*continued...*



Bit-Field	Definition	Static Metadata Type 1
143:136	Data Byte 18: Static_Metadata_Descriptor	white_point_y, MSB
151:144	Data Byte 19: Static_Metadata_Descriptor	max_display_mastering_luminance, LSB
159:152	Data Byte 20: Static_Metadata_Descriptor	max_display_mastering_luminance, MSB
167:160	Data Byte 21: Static_Metadata_Descriptor	min_display_mastering_luminance, LSB
175:168	Data Byte 22: Static_Metadata_Descriptor	min_display_mastering_luminance, MSB
183:176	Data Byte 23: Static_Metadata_Descriptor	Maximum Content Light Level, LSB
191:184	Data Byte 24: Static_Metadata_Descriptor	Maximum Content Light Level, MSB
199:192	Data Byte 25: Static_Metadata_Descriptor	Maximum Frame-average Light Level, LSB
207:200	Data Byte 26: Static_Metadata_Descriptor	Maximum Frame-average Light Level, MSB
215:208	Reserved	
223:216	Reserved	

### Disabling HDR Insertion and Filtering

Disabling HDR insertion and filter enables you to verify the retransmission of HDR content already available in the source auxiliary stream without any modification in the RX-TX Retransmit design example.

To disable HDR InfoFrame insertion and filtering:

1. Set `block_ext_hdr_infotrame` to `1'b0` in the `rxtx_link.v` file to prevent the filtering of the HDR InfoFrame from the Auxiliary stream.
2. Set `multiplexer_in0_valid` of the `avalon_st_multiplexer` instance in the `altera_hdmi_aux_hdr.v` file to `1'b0` to prevent the Auxiliary Packet Generator from forming and inserting additional HDR Infoframe into the TX Auxiliary stream.

## 2.4. Clocking Scheme

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



Figure 9. Intel FPGA HDMI Design Example Clocking Scheme

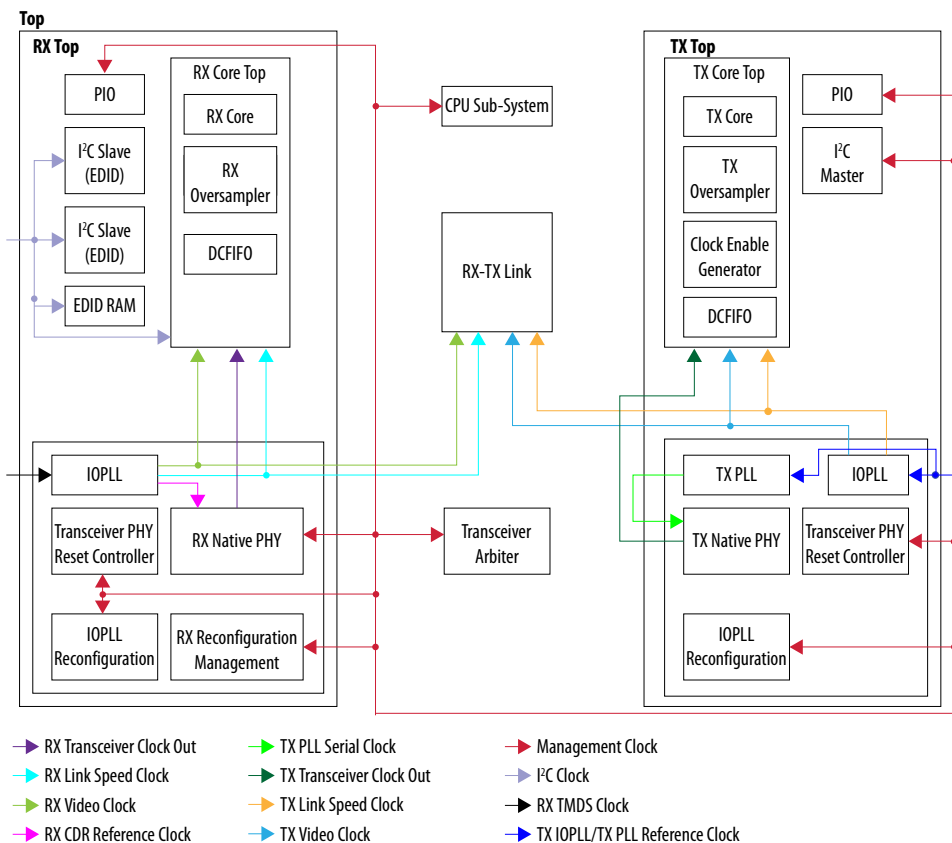


Table 14. Clocking Scheme Signals

Clock	Signal Name in Design	Description
TX IOPLL/ TX PLL Reference Clock	hdmi_clk_in	Reference clock to the TX IOPLL and TX PLL. The clock frequency is the same as the expected TMDS clock frequency from the HDMI TX TMDS clock channel. For this Intel FPGA HDMI design example, this clock is connected to the RX TMDS clock for demonstration purpose. In your application, you need to supply a dedicated clock with TMDS clock frequency from a programmable oscillator for better jitter performance. <i>Note:</i> Do not use a transceiver RX pin as a TX PLL reference clock. Your design will fail to fit if you place the HDMI TX refclk on an RX pin.
TX Transceiver Clock Out	tx_clk	Clock out recovered from the transceiver, and the frequency varies depending on the data rate and symbols per clock. TX transceiver clock out frequency = Transceiver data rate/ (Symbol per clock*10)
TX PLL Serial Clock	tx_bonding_clocks	Serial fast clock generated by TX PLL. The clock frequency is set based on the data rate.
TX/RX Link Speed Clock	ls_clk	Link speed clock. The link speed clock frequency depends on the expected TMDS clock frequency, oversampling factor, symbols per clock, and TMDS bit clock ratio.

continued...



Clock	Signal Name in Design	Description	
		<b>TMDS Bit Clock Ratio</b>	<b>Link Speed Clock Frequency</b>
		0	TMDS clock frequency/ Symbol per clock
		1	TMDS clock frequency *4 / Symbol per clock
TX/RX Video Clock	vid_clk	Video data clock. The video data clock frequency is derived from the TX link speed clock based on the color depth.	
		<b>TMDS Bit Clock Ratio</b>	<b>Video Data Clock Frequency</b>
		0	TMDS clock/ Symbol per clock/ Color depth factor
		1	TMDS clock *4 / Symbol per clock/ Color depth factor
		<b>Bits per Color</b>	<b>Color Depth Factor</b>
		8	1
		10	1.25
		12	1.5
		16	2.0
RX TMDS Clock	tmds_clk_in	TMDS clock channel from the HDMI RX and connects to the reference clock to the IOPLL.	
RX CDR Reference Clock	iopll_outclk0	Reference clock to the RX CDR of RX transceiver.	
		<b>Data Rate</b>	<b>RX Reference Clock Frequency</b>
		Data rate <1 Gbps	5× TMDS clock frequency
		1 Gbps < Data rate <3.4 Gbps	TMDS clock frequency
		Data rate >3.4 Gbps	4× TMDS clock frequency
		<ul style="list-style-type: none"> <li>Data Rate &lt;1 Gbps: For oversampling to meet transceiver minimum data rate requirement.</li> <li>Data Rate &gt;3.4 Gbps: To compensate for the TMDS bit rate to clock ratio of 1/40 to maintain the transceiver data rate to clock ratio at 1/10.</li> </ul> <p><i>Note:</i> Do not use a transceiver RX pin as a CDR reference clock. Your design will fail to fit if you place the HDMI RX refclk on an RX pin.</p>	
RX Transceiver Clock Out	rx_clk	Clock out recovered from the transceiver, and the frequency varies depending on the data rate and symbols per clock. RX transceiver clock out frequency = Transceiver data rate/ (Symbol per clock*10)	
Management Clock	mgmt_clk	A free running 100 MHz clock for these components:	

**continued...**





Clock	Signal Name in Design	Description
		<ul style="list-style-type: none"> <li>Avalon-MM interfaces for reconfiguration               <ul style="list-style-type: none"> <li>The frequency range requirement is between 100–125 MHz.</li> </ul> </li> <li>PHY reset controller for transceiver reset sequence               <ul style="list-style-type: none"> <li>The frequency range requirement is between 1–500 MHz.</li> </ul> </li> <li>IOPLL Reconfiguration               <ul style="list-style-type: none"> <li>The maximum clock frequency is 100 MHz.</li> </ul> </li> <li>RX Reconfiguration for management</li> <li>CPU</li> <li>I<sup>2</sup>C Master</li> </ul>
I <sup>2</sup> C Clock	i2c_clk	A 100 MHz clock input that clocks I <sup>2</sup> C slave, SCDC registers in the HDMI RX core, and EDID RAM.

#### Related Information

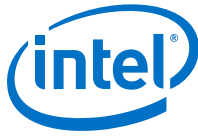
- Using Transceiver RX Pin as CDR Reference Clock
- Using Transceiver RX Pin as TX PLL Reference Clock

## 2.5. Interface Signals

The tables list the signals for the Intel FPGA HDMI IP core design example.

**Table 15. Top-Level Signals**

Signal	Direction	Width	Description
<b>On-board Oscillator Signal</b>			
c10_refclk2_p	Input	1	100 MHz free running clock
<b>User Push Buttons and LEDs</b>			
user_pb	Input	1	Push button to control the Intel FPGA HDMI design functionality
user_dipsw	Input	1	DIP switch to send the DVI or HDMI encoded signal.
user_led_g	Output	4	Green LED display
<b>HDMI FMC Daughter Card Pins on FMC Port B</b>			
fmc_gbtclk_m2c_p_0	Input	1	HDMI RX TMDS clock
fmc_dp_m2c_p	Input	3	HDMI RX red, green, and blue data channels
fmc_dp_c2m_p	Output	4	HDMI TX clock, red, green, and blue data channels
fmc_la_rx_p_9	Input	1	HDMI RX +5V power detect
fmc_la_rx_p_8	Inout	1	HDMI RX hot plug detect
fmc_la_rx_n_8	Inout	1	HDMI RX I <sup>2</sup> C SDA
fmc_la_tx_p_10	Input	1	HDMI RX I <sup>2</sup> C SCL
<i>continued...</i>			



HDMI FMC Daughter Card Pins on FMC Port B			
fmc_la_tx_p_12	Input	1	HDMI TX hot plug detect
fmc_la_tx_n_12	Inout	1	HDMI I <sup>2</sup> C SDA
fmc_la_rx_p_10	Inout	1	HDMI I <sup>2</sup> C SCL

**Table 16. HDMI RX Top-Level Signals**

Signal	Direction	Width	Description
<b>Clock and Reset Signals</b>			
mgmt_clk	Input	1	System clock input (100 MHz)
reset	Input	1	System reset input
tmds_clk_in	Input	1	HDMI RX TMDs clock
i2c_clk	Input	1	Clock input for DDC and SCDC interface
vid_clk_out	Output	1	Video clock output
ls_clk_out	Output	8	Link speed clock output
sys_init	Output	1	System initialization to reset the system upon power-up

<b>RX Transceiver and IOPLL Signals</b>			
rx_serial_data	Input	3	HDMI serial data to the RX Native PHY
gxb_rx_ready	Output	1	Indicates RX Native PHY is ready
gxb_rx_cal_busy_out	Output	3	RX Native PHY calibration busy to the transceiver arbiter
gxb_rx_cal_busy_in	Input	3	Calibration busy signal from the transceiver arbiter to the RX Native PHY
iopll_locked	Output	1	Indicate IOPLL is locked
gxb_reconfig_write	Input	3	Transceiver reconfiguration Avalon-MM interface from the RX Native PHY to the transceiver arbiter
gxb_reconfig_read	Input	3	
gxb_reconfig_address	Input	30	
gxb_reconfig_writedata	Input	96	
gxb_reconfig_readdata	Output	96	
gxb_reconfig_waitrequest	Output	3	

<b>RX Reconfiguration Management</b>			
rx_reconfig_en	Output	1	RX Reconfiguration enables signal
measure	Output	24	HDMI RX TMDs clock frequency measurement (in 10 ms)
measure_valid	Output	1	Indicates the measure signal is valid
os	Output	1	Oversampling factor: <ul style="list-style-type: none"> <li>• 0: No oversampling</li> <li>• 1: 5x oversampling</li> </ul>
<i>continued...</i>			



RX Reconfiguration Management			
reconfig_mgmt_write	Output	1	RX reconfiguration management Avalon-MM interface to transceiver arbiter
reconfig_mgmt_read	Output	1	
reconfig_mgmt_address	Output	12	
reconfig_mgmt_writedata	Output	32	
reconfig_mgmt_readdata	Input	32	
reconfig_mgmt_waitrequest	Input	1	

HDMI RX Core Signals			
TMSD_Bit_clock_Ratio	Output	1	SCDC register interfaces
audio_de	Output	1	HDMI RX core audio interfaces
audio_data	Output	256	
audio_info_ai	Output	48	
audio_N	Output	20	
audio_CTS	Output	20	
audio_metadata	Output	165	
audio_format	Output	5	
aux_pkt_data	Output	72	
aux_pkt_addr	Output	6	
aux_pkt_wr	Output	1	
aux_data	Output	72	
aux_sop	Output	1	
aux_eop	Output	1	
aux_valid	Output	1	
aux_error	Output	1	
gcp	Output	6	HDMI RX core sideband signals
info_avi	Output	112	
info_vsi	Output	61	
colordepth_mgmt_sync	Output	2	
vid_data	Output	$N*48$	HDMI RX core video ports <i>Note: N = symbols per clock</i>
vid_vsync	Output	$N$	
vid_hsync	Output	$N$	
vid_de	Output	$N$	
mode	Output	1	HDMI RX core control and status ports <i>Note: N = symbols per clock</i>
ctrl	Output	$N*6$	
locked	Output	3	

*continued...*



HDMI RX Core Signals			
vid_lock	Output	1	
in_5v_power	Input	1	HDMI RX 5V detect and hotplug detect
hdmi_rx_hpd_n	Inout	1	

I <sup>2</sup> C Signals			
hdmi_rx_i2c_sda	Inout	1	HDMI RX DDC and SCDC interface
hdmi_rx_i2c_scl	Inout	1	

RX EDID RAM Signals			
edid_ram_access	Input	1	HDMI RX EDID RAM access interface. Assert edid_ram_access when you want to write or read from the EDID RAM, else this signal should be kept low.
edid_ram_address	Input	8	
edid_ram_write	Input	1	
edid_ram_read	Input	1	
edid_ram_readdata	Output	8	
edid_ram_writedata	Input	8	
edid_ram_waitrequest	Output	1	

**Table 17. HDMI TX Top-Level Signals**

Signal	Direction	Width	Description
Clock and Reset Signals			
mgmt_clk	Input	1	System clock input (100 MHz)
reset	Input	1	System reset input
hdmi_clk_in	Input	1	Reference clock to TX IOPLL and TX PLL. The clock frequency is the same as the TMDS clock frequency.
vid_clk_out	Output	1	Video clock output
ls_clk_out	Output	8	Link speed clock output
sys_init	Output	1	System initialization to reset the system upon power-up
reset_xcvr	Input	1	Reset to TX transceiver
reset_pll	Input	1	Reset to IOPLL and TX PLL
reset_pll_reconfig	Output	1	Reset to PLL reconfiguration
TX Transceiver and IOPLL Signals			
tx_serial_data	Output	4	HDMI serial data from the TX Native PHY
gxb_tx_ready	Output	1	Indicates TX Native PHY is ready
gxb_tx_cal_busy_out	Output	4	TX Native PHY calibration busy signal to the transceiver arbiter
<i>continued...</i>			



TX Transceiver and IOPLL Signals			
gxb_tx_cal_busy_in	Input	4	Calibration busy signal from the transceiver arbiter to the TX Native PHY
iopll_locked	Output	1	Indicate IOPLL is locked
txpll_locked	Output	1	Indicate TX PLL is locked
gxb_reconfig_write	Input	4	Transceiver reconfiguration Avalon-MM interface from the TX Native PHY to the transceiver arbiter
gxb_reconfig_read	Input	4	
gxb_reconfig_address	Input	40	
gxb_reconfig_writedata	Input	128	
gxb_reconfig_readdata	Output	128	
gxb_reconfig_waitrequest	Output	4	

TX IOPLL and TX PLL Reconfiguration Signals			
pll_reconfig_write/ tx_pll_reconfig_write	Input	1	TX IOPLL/TX PLL reconfiguration Avalon-MM interfaces
pll_reconfig_read/ tx_pll_reconfig_read	Input	1	
pll_reconfig_address/ tx_pll_reconfig_address	Input	10	
pll_reconfig_writedata/ tx_pll_reconfig_writedata	Input	32	
pll_reconfig_readdata/ tx_pll_reconfig_readdata	Output	32	
pll_reconfig_waitrequest/ tx_pll_reconfig_waitrequest	Output	1	
os	Input	2	Oversampling factor: <ul style="list-style-type: none"> <li>• 0: No oversampling</li> <li>• 1: 3× oversampling</li> <li>• 2: 4× oversampling</li> <li>• 3: 5× oversampling</li> </ul>
measure	Input	24	Indicates the TMDS clock frequency of the transmitting video resolution.

HDMI TX Core Signals			
ctrl	Input	6*N	HDMI TX core control interfaces <i>Note: N = Symbols per clock</i>
mode	Input	1	
TMDS_Bit_clock_Ratio	Input	1	SCDC register interfaces
Scrambler_Enable	Input	1	
audio_de	Input	1	HDMI TX core audio interfaces
audio_mute	Input	1	
audio_data	Input	256	

continued...



HDMI TX Core Signals			
audio_info_ai	Input	49	
audio_N	Input	22	
audio_CTS	Input	22	
audio_metadata	Input	166	
audio_format	Input	5	
aux_ready	Output	1	
aux_data	Input	72	
aux_sop	Input	1	
aux_eop	Input	1	
aux_valid	Input	1	
gcp	Input	6	HDMI TX core sideband signals
info_avi	Input	113	
info_vsi	Input	62	
vid_data	Input	$N*48$	HDMI TX core video ports <i>Note: N = symbols per clock</i>
vid_vsync	Input	$N$	
vid_hsync	Input	$N$	
vid_de	Input	$N$	

I <sup>2</sup> C and Hot Plug Detect Signals			
tx_i2c_avalon_waitrequest	Output	1	I <sup>2</sup> C master Avalon-MM interfaces
tx_i2c_avalon_address	Input	3	
tx_i2c_avalon_writedata	Input	32	
tx_i2c_avalon_readdata	Output	32	
tx_i2c_avalon_chipselect	Input	1	
tx_i2c_avalon_write	Input	1	
tx_i2c_avalon_irq	Output	1	
hdmi_tx_i2c_sda	Inout	1	HDMI TX DDC and SCDC interfaces
hdmi_tx_i2c_scl	Inout	1	
hdmi_tx_ti_i2c_sda	Inout	1	I <sup>2</sup> C interfaces for Bitec daughter card revision 11, T1181 control
hdmi_tx_ti_i2c_scl	Inout	1	
hdmi_tx_hpd_n	Input	1	HDMI TX hotplug detect interfaces
tx_hpd_ack	Input	1	
tx_hpd_req	Output	1	



**Table 18. Transceiver Arbiter Signals**

Signal	Direction	Width	Description
clk	Input	1	Reconfiguration clock. This clock must share the same clock with the reconfiguration management blocks.
reset	Input	1	Reset signal. This reset must share the same reset with the reconfiguration management blocks.
rx_rcfg_en	Input	1	RX reconfiguration enable signal
tx_rcfg_en	Input	1	TX reconfiguration enable signal
rx_rcfg_ch	Input	2	Indicates which channel to be reconfigured on the RX core. This signal must always remain asserted.
tx_rcfg_ch	Input	2	Indicates which channel to be reconfigured on the TX core. This signal must always remain asserted.
rx_reconfig_mgmt_write	Input	1	Reconfiguration Avalon-MM interfaces from the RX reconfiguration management
rx_reconfig_mgmt_read	Input	1	
rx_reconfig_mgmt_address	Input	10	
rx_reconfig_mgmt_writedata	Input	32	
rx_reconfig_mgmt_readdata	Output	32	
rx_reconfig_mgmt_waitrequest	Output	1	
tx_reconfig_mgmt_write	Input	1	Reconfiguration Avalon-MM interfaces from the TX reconfiguration management
tx_reconfig_mgmt_read	Input	1	
tx_reconfig_mgmt_address	Input	10	
tx_reconfig_mgmt_writedata	Input	32	
tx_reconfig_mgmt_readdata	Output	32	
tx_reconfig_mgmt_waitrequest	Output	1	
reconfig_write	Output	1	Reconfiguration Avalon-MM interfaces to the transceiver
reconfig_read	Output	1	
reconfig_address	Output	10	
reconfig_writedata	Output	32	
rx_reconfig_readdata	Input	32	
rx_reconfig_waitrequest	Input	1	
tx_reconfig_readdata	Input	1	
tx_reconfig_waitrequest	Input	1	
rx_cal_busy	Input	1	

*continued...*



Signal	Direction	Width	Description
tx_cal_busy	Input	1	Calibration status signal from the TX transceiver
rx_reconfig_cal_busy	Output	1	Calibration status signal to the RX transceiver PHY reset control
tx_reconfig_cal_busy	Output	1	Calibration status signal from the TX transceiver PHY reset control

**Table 19. RX-TX Link Signals**

Signal	Direction	Width	Description
reset	Input	1	Reset to the video/audio/auxiliary/sidebands FIFO buffer.
mgmt_clk	Input	1	100 MHz clock
i2c_clk	Input	1	I <sup>2</sup> C clock
hdmi_tx_ls_clk	Input	1	HDMI TX link speed clock
hdmi_rx_ls_clk	Input	1	HDMI RX link speed clock
hdmi_tx_vid_clk	Input	1	HDMI TX video clock
hdmi_rx_vid_clk	Input	1	HDMI RX video clock
sys_init	Input	1	System initialization to reset the system upon power-up
wd_reset	Input	1	Watchdog timer reset
hdmi_rx_locked	Input	3	Indicates HDMI RX locked status
hdmi_rx_de	Input	<i>N</i>	HDMI RX video interfaces <i>Note: N = symbols per clock</i>
hdmi_rx_hsync	Input	<i>N</i>	
hdmi_rx_vsync	Input	<i>N</i>	
hdmi_rx_data	Input	<i>N*48</i>	
rx_audio_format	Input	5	HDMI RX audio interfaces
rx_audio_metadata	Input	165	
rx_audio_info_ai	Input	48	
rx_audio_CTS	Input	20	
rx_audio_N	Input	20	
rx_audio_de	Input	1	
rx_audio_data	Input	256	
rx_gcp	Input	6	HDMI RX sideband interfaces
rx_info_avi	Input	112	
rx_info_vsi	Input	61	
rx_aux_eop	Input	1	HDMI RX auxiliary interfaces
rx_aux_sop	Input	1	
rx_aux_valid	Input	1	

*continued...*





Signal	Direction	Width	Description
rx_aux_data	Input	72	
hdmi_tx_de	Output	$N$	HDMI TX video interfaces <i>Note: <math>N</math> = symbols per clock</i>
hdmi_tx_hsync	Output	$N$	
hdmi_tx_vsync	Output	$N$	
hdmi_tx_data	Output	$N*48$	
tx_audio_format	Output	5	HDMI TX audio interfaces
tx_audio_metadata	Output	165	
tx_audio_info_ai	Output	48	
tx_audio_CTS	Output	20	
tx_audio_N	Output	20	
tx_audio_de	Output	1	
tx_audio_data	Output	256	
tx_gcp	Output	6	HDMI TX sideband interfaces
tx_info_avi	Output	112	
tx_info_vsi	Output	61	
tx_aux_eop	Output	1	HDMI TX auxiliary interfaces
tx_aux_sop	Output	1	
tx_aux_valid	Output	1	
tx_aux_data	Output	72	
tx_aux_ready	Output	1	

**Table 20. Platform Designer System Signals**

Signal	Direction	Width	Description
cpu_clk	Input	1	CPU clock
cpu_clk_reset_n	Input	1	CPU reset
tmds_bit_clock_ratio_pio_external_connection_export	Input	1	TMDS bit clock ratio
measure_pio_external_connection_export	Input	24	Expected TMDS clock frequency
measure_valid_pio_external_connection_export	Input	1	Indicates measure PIO is valid
oc_i2c_master_av_slave_translator_avalonанти_slave_0_address	Output	3	I <sup>2</sup> C Master Avalon-MM interfaces for HDMI TX DDC and SCDC
oc_i2c_master_av_slave_translator_avalonанти_slave_0_write	Output	1	
oc_i2c_master_av_slave_translator_avalonанти_slave_0_readdata		32	

*continued...*



Signal	Direction	Width	Description
oc_i2c_master_av_slave_translator_avalon_anti_slave_0_writedata	Output	32	
oc_i2c_master_av_slave_translator_avalon_anti_slave_0_waitrequest	Input	1	
oc_i2c_master_av_slave_translator_avalon_anti_slave_0_chipselect	Output	1	
oc_i2c_master_ti_avalon_anti_slave_address	Output	3	I <sup>2</sup> C Master Avalon-MM interfaces for Bitec daughter card revision 11, T1181 control
oc_i2c_master_ti_avalon_anti_slave_write	Output	1	
oc_i2c_master_ti_avalon_anti_slave_readdata	Input	32	
oc_i2c_master_ti_avalon_anti_slave_writedata	Output	32	
oc_i2c_master_ti_avalon_anti_slave_waitrequest	Input	1	
oc_i2c_master_ti_avalon_anti_slave_chipselect	Output	1	
edid_ram_access_pio_external_connection_export	Output	1	
edid_ram_slave_translator_address	Output	8	
edid_ram_slave_translator_write	Output	1	
edid_ram_slave_translator_read	Output	1	
edid_ram_slave_translator_readdata	Input	8	
edid_ram_slave_translator_writedata	Output	8	
edid_ram_slave_translator_waitrequest	Input	1	
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_waitrequest	Input	1	TX PLL Reconfiguration Avalon-MM interfaces
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_writedata	Output	32	
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_address	Output	10	
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_write	Output	1	
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_read	Output	1	
tx_pll_rcfg_mgmt_translator_avalon_anti_slave_readdata	Input	32	
tx_pll_waitrequest_pio_external_connection_export	Input	1	
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_address	Output	12	TX PMA Reconfiguration Avalon-MM interfaces
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_write	Output	1	

*continued...*



Signal	Direction	Width	Description
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_read	Output	1	
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_readdata	Input	32	
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_writedata	Output	32	
tx_pma_rcfg_mgmt_translator_avalon_anti_slave_waitrequest	Input	1	
tx_pma_waitrequest_pio_external_connection_export	Input	1	TX PMA waitrequest
tx_pma_cal_busy_pio_external_connection_export	Input	1	TX PMA Recalibration Busy
tx_pma_ch_export	Output	2	TX PMA Channels
tx_rcfg_en_pio_external_connection_export			TX PMA Reconfiguration Enable
tx_iopll_rcfg_mgmt_translator_avalon_anti_slave_writedata	Output	32	TX IOPLL Reconfiguration Avalon-MM interfaces
tx_iopll_rcfg_mgmt_translator_avalon_anti_slave_address	Output	9	
tx_iopll_rcfg_mgmt_translator_avalon_anti_slave_write	Output	1	
tx_iopll_rcfg_mgmt_translator_avalon_anti_slave_read	Output	1	
tx_iopll_rcfg_mgmt_translator_avalon_anti_slave_readdata	Input	32	
tx_os_pio_external_connection_export	Output	2	Oversampling factor: <ul style="list-style-type: none"> <li>• 0: No oversampling</li> <li>• 1: 3× oversampling</li> <li>• 2: 4× oversampling</li> <li>• 3: 5× oversampling</li> </ul>
tx_rst_pll_pio_external_connection_export	Output	1	Reset to IOPLL and TX PLL
tx_rst_xcvr_pio_external_connection_export	Output	1	Reset to TX Native PHY
wd_timer_resetrequest_reset	Output	1	Watchdog timer reset
color_depth_pio_external_connection_export	Input	2	Color depth
tx_hpd_ack_pio_external_connection_export	Output	1	For TX hot-plug detect handshaking
tx_hpd_req_pio_external_connection_export	Input	1	

## 2.6. Design RTL Parameters

Use the HDMI TX and RX Top RTL parameters to customize the design example.

Most of the design parameters are available in the Intel FPGA HDMI IP Design Example parameter editor. You can still change the design example settings you made in the parameter editor through the RTL parameters.



**Table 21. HDMI RX Top Parameters**

Parameter	Value	Description
SUPPORT_DEEP_COLOR	<ul style="list-style-type: none"> <li>0: No deep color</li> <li>1: Deep color</li> </ul>	Determines if the core can encode deep color formats.
SUPPORT_AUXILIARY	<ul style="list-style-type: none"> <li>0: No AUX</li> <li>1: AUX</li> </ul>	Determines if the auxiliary channel encoding is included.
SYMBOLS_PER_CLOCK	2	Supports only 2 symbols per clock for Intel Cyclone 10 GX devices.
SUPPORT_AUDIO	<ul style="list-style-type: none"> <li>0: No audio</li> <li>1: Audio</li> </ul>	Determines if the core can encode audio.
BITEC_DAUGHTER_CARD_REV	<ul style="list-style-type: none"> <li>0: Not targeting any Bitec HDMI daughter card</li> <li>4: Targeting Bitec HDMI daughter card revision 4</li> <li>6: Targeting Bitec HDMI daughter card revision 6</li> <li>11: Targeting Bitec HDMI daughter card revision 11</li> </ul>	Specifies the revision of the Bitec HDMI daughter card used. When you change the revision, the design may swap the transceiver channels and invert the polarity according to the Bitec HDMI daughter card requirements. If you set the BITEC_DAUGHTER_CARD_REV parameter to 0, the design will not make any changes to the transceiver channels and the polarity.

**Table 22. HDMI TX Top Parameters**

Parameter	Value	Description
USE_FPLL	1	Supports fPLL as TX PLL only for Intel Cyclone 10 GX devices. Always set this parameter to 1.
SUPPORT_DEEP_COLOR	<ul style="list-style-type: none"> <li>0: No deep color</li> <li>1: Deep color</li> </ul>	Determines if the core can encode deep color formats.
SUPPORT_AUXILIARY	<ul style="list-style-type: none"> <li>0: No AUX</li> <li>1: AUX</li> </ul>	Determines if the auxiliary channel encoding is included.
SYMBOLS_PER_CLOCK	2	Supports only 2 symbols per clock for Intel Cyclone 10 GX devices.
SUPPORT_AUDIO	<ul style="list-style-type: none"> <li>0: No audio</li> <li>1: Audio</li> </ul>	Determines if the core can encode audio.
BITEC_DAUGHTER_CARD_REV	<ul style="list-style-type: none"> <li>0: Not targeting any Bitec HDMI daughter card</li> <li>4: Supports Bitec HDMI daughter card revision 4</li> <li>6: Supports Bitec HDMI daughter card revision 6</li> <li>11: Supports Bitec HDMI daughter card revision 11</li> </ul>	Specifies the revision of the Bitec HDMI daughter card used. When you change the revision, the design may swap the transceiver channels and invert the polarity according to the Bitec HDMI daughter card requirements. If you set the BITEC_DAUGHTER_CARD_REV parameter to 0, the design will not make any changes to the transceiver channels and the polarity.
POLARITY_INVERSION	<ul style="list-style-type: none"> <li>0: Invert polarity</li> <li>1: Do not invert polarity</li> </ul>	Set this parameter to 1 to invert the value of each bit of the input data. Setting this parameter to 1 assigns 4'b1111 to the tx_polinvt port of the TX transceiver.

## 2.7. Hardware Setup

The Intel FPGA HDMI design example is HDMI 2.0 capable and performs a loop-through demonstration for a standard HDMI video stream.



To run the hardware test, connect an HDMI-enabled device—such as a graphics card with HDMI interface—to the Transceiver Native PHY RX block, and the HDMI sink input.

1. The HDMI sink decodes the port into a standard video stream and sends it to the clock recovery core.
2. The HDMI RX core decodes the video, auxiliary, and audio data to be looped back in parallel to the HDMI TX core through the DCFIFO.
3. The HDMI source port of the FMC daughter card transmits the image to a monitor.

**Note:** If you use another Intel FPGA development board, you must change the device assignments and the pin assignments. The transceiver analog setting is tested for the Intel Cyclone 10 GX FPGA development kit and Bitec HDMI 2.0 daughter card. You may modify the settings for your own board.

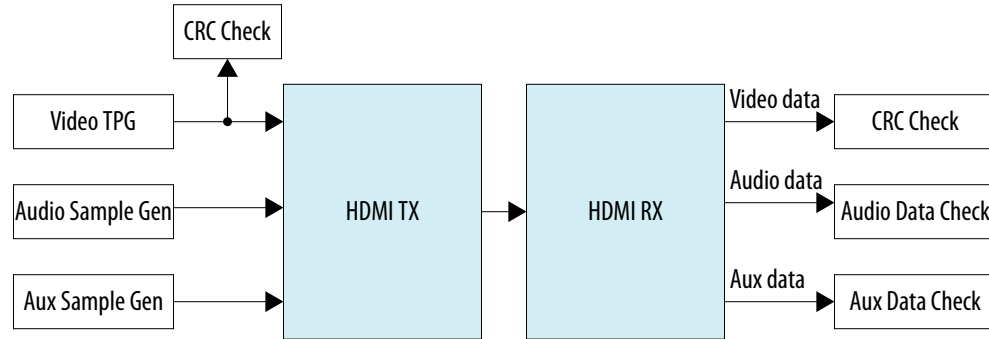
**Table 23. On-board Push Button, DIP Switch, and User LED Functions**

LEDs	Function
user_dipsw[0]	<ul style="list-style-type: none"> <li>• Set 0 (in the design example) and push the board switch to <b>ON</b> = HDMI TX core to send the DVI encoded signal</li> <li>• Set 1 (in the design example) and push the board switch to <b>OFF</b> = HDMI TX core to send the HDMI encoded signal</li> </ul>
user_pb[0]	Press once to perform system reset.
user_pb[1]	Press once to toggle the HPD signal to the standard HDMI source.
user_pb[2]	<ul style="list-style-type: none"> <li>• Press and hold to instruct the TX core to stop sending the InfoFrames from the sideband signals.</li> <li>• Release to resume sending the InfoFrames from the sideband signals.</li> </ul>
USER_LED[0]	RX HDMI PLL lock status or RX transceiver ready status. <ul style="list-style-type: none"> <li>• 0 = RX HDMI PLL unlocked or RX transceiver is not ready</li> <li>• 1 = RX HDMI PLL locked or RX transceiver is ready</li> </ul>
USER_LED[1]	RX HDMI core lock status. <ul style="list-style-type: none"> <li>• 0 = At least 1 channel unlocked</li> <li>• 1 = All 3 channels locked</li> </ul>
USER_LED[2]	TX HDMI PLL lock status, TX transceiver PLL lock status, or TX transceiver ready status. <ul style="list-style-type: none"> <li>• 0 = TX HDMI PLL unlocked, TX transceiver PLL unlocked, or TX transceiver is not ready</li> <li>• 1 = TX HDMI PLL locked, TX transceiver PLL locked, or TX transceiver is ready</li> </ul>
USER_LED[3]	TX or RX oversampling status. <ul style="list-style-type: none"> <li>• 0 = Non-oversampled (data rate &gt; 1,000 Mbps in Intel Cyclone 10 GX device)</li> <li>• 1 = Oversampled (data rate &lt; 1,000 Mbps in Intel Cyclone 10 GX device)</li> </ul>

## 2.8. Simulation Testbench

The simulation testbench simulates the HDMI TX serial loopback to the RX core.

**Figure 10. Intel FPGA HDMI IP Core Simulation Testbench Block Diagram**



**Table 24. Testbench Components**

Component	Description
Video TPG	The video test pattern generator (TPG) provides the video stimulus.
Audio Sample Gen	The audio sample generator provides audio sample stimulus. The generator generates an incrementing test data pattern to be transmitted through the audio channel.
Aux Sample Gen	The aux sample generator provides the auxiliary sample stimulus. The generator generates a fixed data to be transmitted from the transmitter.
CRC Check	This checker verifies if the TX transceiver recovered clock frequency matches the desired data rate.
Audio Data Check	The audio data check compares whether the incrementing test data pattern is received and decoded correctly.
Aux Data Check	The aux data check compares whether the expected aux data is received and decoded correctly on the receiver side.

The HDMI simulation testbench does the following verification tests:

HDMI Feature	Verification
Video data	<ul style="list-style-type: none"> <li>The testbench implements CRC checking on the input and output video.</li> <li>It checks the CRC value of the transmitted data against the CRC calculated in the received video data.</li> <li>The testbench then performs the checking after detecting 4 stable V-SYNC signals from the receiver.</li> </ul>
Auxiliary data	<ul style="list-style-type: none"> <li>The aux sample generator generates a fixed data to be transmitted from the transmitter.</li> <li>On the receiver side, the generator compares whether the expected auxiliary data is received and decoded correctly.</li> </ul>
Audio data	<ul style="list-style-type: none"> <li>The audio sample generator generates an incrementing test data pattern to be transmitted through the audio channel.</li> <li>On the receiver side, the audio data checker checks and compares whether the incrementing test data pattern is received and decoded correctly.</li> </ul>

A successful simulation ends with the following message:

```
# SYMBOLS_PER_CLOCK = 2
# VIC = 0
# AUDIO_CLK_DIVIDE = 800
# TEST_HDMI_6G = 1
# Simulation pass
```



**Table 25. Intel FPGA HDMI Design Example Supported Simulators**

Simulator	Verilog HDL	VHDL
ModelSim - Intel FPGA Edition/ ModelSim - Intel FPGA Starter Edition	Yes	Yes
VCS/VCS MX	Yes	Yes
Riviera-PRO	Yes	Yes
NCSim	Yes	No
Xcelium Parallel	Yes	No

## 2.9. Upgrading Your Design

When you upgrade your designs to a later version, you may have to add, remove, or edit some of the generated files.

### Upgrading from Version 18.0 Update 1 to 18.1

1. Generate a new design example in version 18.1 using the same configurations of your existing design.
2. Compare the whole design example directory; replace the files that have changes with the new files and copy over the new files to your existing design.
3. Click **IP Upgrade** to upgrade all the IP and Platform Designer files.
4. Move the following assignments to the bottom of the QSF file.

```
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/mr_clock_sync.sdc -  
entity mr_clock_sync  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/  
mr_reconfig_mgmt.sdc -entity mr_reconfig_mgmt  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/rxtx_link.sdc -  
entity rxtx_link
```

5. Recompile the design.

### Upgrading from Version 18.0 to 18.1

1. Generate a new design example in version 18.1 using the same configurations of your existing design.
2. Compare the whole design example directory; replace the files that have changes with the new files and copy over the new files to your existing design.
3. In the `/quartus/c10_hdmi2_demo.qsf` file:



- Add the following new file:

```
set_global_assignment -name IP_FILE ../rtl/ip/nios/  
nios_oc_i2c_master_ti.ip
```

- Add new pin assignments for `fmcb_la_tx_p_11` and `fmcb_la_rx_n_9` pins:

```
set_location_assignment PIN_H13 -to fmcb_la_tx_p_11  
set_location_assignment PIN_H18 -to fmcb_la_rx_n_9
```

```
set_instance_assignment -name IO_STANDARD "1.8 V" -to fmcb_la_tx_p_11  
set_instance_assignment -name IO_STANDARD "1.8 V" -to fmcb_la_rx_n_9
```

```
set_instance_assignment -name AUTO_OPEN_DRAIN_PINS ON -to  
fmcb_la_tx_p_11  
set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to  
fmcb_la_tx_p_11  
set_instance_assignment -name AUTO_OPEN_DRAIN_PINS ON -to  
fmcb_la_rx_n_9  
set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to  
fmcb_la_rx_n_9
```

```
set_instance_assignment -name CURRENT_STRENGTH_NEW DEFAULT -to  
fmcb_la_tx_p_11  
set_instance_assignment -name CURRENT_STRENGTH_NEW DEFAULT -to  
fmcb_la_rx_n_9
```

```
set_instance_assignment -name SLEW_RATE 1 -to fmcb_la_tx_p_11  
set_instance_assignment -name SLEW_RATE 1 -to fmcb_la_rx_n_9
```

4. Click **IP Upgrade** to upgrade all the IP and Platform Designer files.
5. Move the following assignments to the bottom of the QSF file.

```
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/mr_clock_sync.sdc -  
entity mr_clock_sync  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/  
mr_reconfig_mgmt.sdc -entity mr_reconfig_mgmt  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/rxtx_link.sdc -  
entity rxtx_link
```

6. Copy over the `software/tx_control_src` folder and locate the script directory. Run `sh script/build_sw.sh` script to rebuild the software.
7. Recompile the design.

### Upgrading from Version 17.1 Update 1 to 18.1

1. Generate a new design example in version 18.1 using the same configurations of your existing design.
2. Compare the whole design example directory; replace the files that have changes with the new files and copy over the new files to your existing design.
3. In the `/quartus/c10_hdmi2_demo.qsf` file:





- Locate and place the `clock_crosser.v` file into `/rtl/common` folder.

```
set_global_assignment -name VERILOG_FILE ../rtl/common/clock_crosser.v
```

- Add the following new files:

```
set_global_assignment -name VERILOG_FILE ../rtl/common/dcfifo_inst.v
```

```
set_global_assignment -name IP_FILE ../rtl/common/fifo.ip
```

```
set_global_assignment -name IP_FILE ../rtl/ip/nios/  
nios_oc_i2c_master_ti.ip
```

- Rename the `a10_reconfig_arbiter.sv` to `xcvr_reconfig_arbiter.sv`.

```
set_global_assignment -name SYSTEMVERILOG_FILE ../rtl/  
xcvr_reconfig_arbiter.sv
```

- Remove `mr.sdc` and add the `mr_clock_sync.sdc`, `mr_reconfig_mgmt.sdc`, and `rxtx_link.sdc` files at the the bottom of the QSF file.

```
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/  
mr_clock_sync.sdc -entity mr_clock_sync  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/  
mr_reconfig_mgmt.sdc -entity mr_reconfig_mgmt  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/rxtx_link.sdc -  
entity rxtx_link
```

- Add new pin assignments for `fmcb_la_tx_p_11` and `fmcb_la_rx_n_9` pins:

```
set_location_assignment PIN_H13 -to fmcb_la_tx_p_11  
set_location_assignment PIN_H18 -to fmcb_la_rx_n_9
```

```
set_instance_assignment -name IO_STANDARD "1.8 V" -to fmcb_la_tx_p_11  
set_instance_assignment -name IO_STANDARD "1.8 V" -to fmcb_la_rx_n_9
```

```
set_instance_assignment -name AUTO_OPEN_DRAIN_PINS ON -to  
fmcb_la_tx_p_11  
set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to  
fmcb_la_tx_p_11  
set_instance_assignment -name AUTO_OPEN_DRAIN_PINS ON -to  
fmcb_la_rx_n_9  
set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to  
fmcb_la_rx_n_9
```

```
set_instance_assignment -name CURRENT_STRENGTH_NEW DEFAULT -to  
fmcb_la_tx_p_11  
set_instance_assignment -name CURRENT_STRENGTH_NEW DEFAULT -to  
fmcb_la_rx_n_9
```

```
set_instance_assignment -name SLEW_RATE 1 -to fmcb_la_tx_p_11  
set_instance_assignment -name SLEW_RATE 1 -to fmcb_la_rx_n_9
```

4. Click **IP Upgrade** to upgrade all the IP and Platform Designer files.
5. Move the following assignments to the bottom of the QSF file.

```
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/mr_clock_sync.sdc -  
entity mr_clock_sync  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/  
mr_reconfig_mgmt.sdc -entity mr_reconfig_mgmt  
set_global_assignment -name SDC_ENTITY_FILE ../rtl/sdc/rxtx_link.sdc -  
entity rxtx_link
```



6. Copy over the `software/tx_control_src` folder and locate the script directory. Run `sh script/build_sw.sh` script to rebuild the software.
7. Recompile the design.



## A. Revision History for Intel FPGA HDMI Design Example User Guide for Intel Cyclone 10 GX Devices

Document Version	Intel Quartus Prime Version	Changes
2018.10.25	18.1	<ul style="list-style-type: none"> <li>• Added generated files for the tx_control_src folder: ti_i2c.c and ti_i2c.h</li> <li>• Added a new design RTL parameter, BITEC_DAUGHTER_CARD_REV, to enable you select the revision of the Bitec HDMI daughter card.</li> <li>• Added limitation that the HDMI RX core does not perform word alignment for HDMI 2.0 resolutions (data rate &gt; 3.4 Gbps). Use Transceiver PCS word aligner and control logic to achieve fast word alignment.</li> <li>• Added the following new signals for Bitec daughter card revision 11. <ul style="list-style-type: none"> <li>– hdmi_tx_ti_i2c_sda</li> <li>– hdmi_tx_ti_i2c_scl</li> <li>– oc_i2c_master_ti_avalon_anti_slave_address</li> <li>– oc_i2c_master_ti_avalon_anti_slave_write</li> <li>– oc_i2c_master_ti_avalon_anti_slave_readdata</li> <li>– oc_i2c_master_ti_avalon_anti_slave_writedata</li> <li>– oc_i2c_master_ti_avalon_anti_slave_waitrequest</li> <li>– oc_i2c_master_ti_avalon_anti_slave_chipselect</li> </ul> </li> <li>• Added a section about <i>Upgrading Your Design</i>.</li> <li>• Edited the errors in the description for user_dipsw[0] and user_pb[1] in the <i>Hardware Setup</i> section.</li> </ul>
2017.12.25	17.1	Initial release.

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 or its subsidiaries in the U.S. and/or other countries. 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.

\*Other names and brands may be claimed as the property of others.

ISO  
9001:2015  
Registered