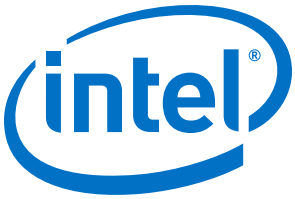


# Intel® FPGA HDMI Design Example User Guide for Intel® Arria 10 Devices

Updated for Intel® Quartus® Prime Design Suite: **17.1**



UG-20077 | 2017.11.06
Latest document on the web: PDF | HTML

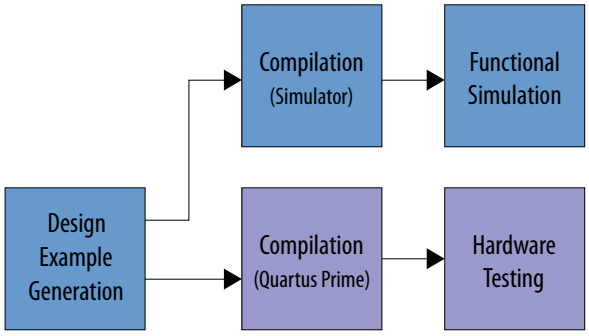
# Contents

# 1 Intel® FPGA HDMI Design Example Quick Start Guide for Intel® Arria® 10 Devices

The Intel® FPGA HDMI IP core design example for Intel Arria® 10 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 Links**

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**

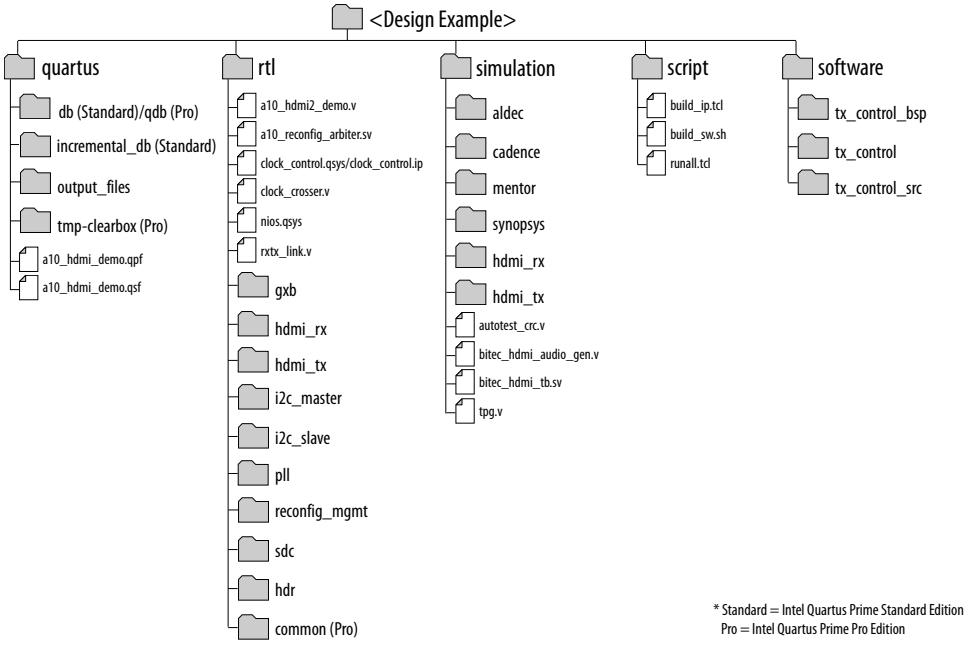
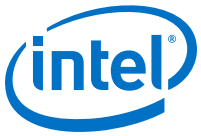



**Table 1. Generated RTL Files**

| Folders | Files |
|---|---|
| gxb | • /gxb_rx.qsys (Intel Quartus Prime Standard Edition)<br>• /gxb_rx.ip (Intel Quartus Prime Pro Edition) |
| | • /gxb_rx_reset.qsys (Intel Quartus Prime Standard Edition)<br>• /gxb_rx_reset.ip (Intel Quartus Prime Pro Edition) |
| | • /gxb_tx.qsys (Intel Quartus Prime Standard Edition)<br>• /gxb_tx.ip (Intel Quartus Prime Pro Edition) |
| | • /gxb_tx_fpll.qsys (Intel Quartus Prime Standard Edition)<br>• /gxb_tx_fpll.ip (Intel Quartus Prime Pro Edition) |
| | • /gxb_tx_reset.qsys (Intel Quartus Prime Standard Edition)<br>• /gxb_tx_reset.ip (Intel Quartus Prime Pro Edition) |
| hdmi_rx | • /hdmi_rx.qsys (Intel Quartus Prime Standard Edition)<br>• /hdmi_rx.ip (Intel Quartus Prime Pro Edition) |
| | /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.qsys (Intel Quartus Prime Standard Edition)<br>• /hdmi_tx.ip (Intel Quartus Prime Pro Edition) |
| | /hdmi_tx_top.v |

*continued...*



| Folders | Files |
|---|---|
| | /mr_ce.v |
| | /mr_hdmi_tx_core_top.v |
| | /mr_tx_oversample.v |
| 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.qsys (Intel Quartus Prime Standard Edition)<br>• /edid_ram.ip (Intel Quartus Prime Pro Edition) |
| | /I2Cslave.v |
| | • /output_buf_i2c.qsys (Intel Quartus Prime Standard Edition)<br>• /output_buf_i2c.ip (Intel Quartus Prime Pro Edition) |
| | /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.qsys (Intel Quartus Prime Standard Edition)<br>• /pll_hdmi.ip (Intel Quartus Prime Pro Edition) |
| | • /pll_hdmi_reconfig.qsys (Intel Quartus Prime Standard Edition)<br>• /pll_hdmi_reconfig.ip (Intel Quartus Prime Pro Edition) |
| common | /reset_controller (Intel Quartus Prime Pro Edition) |
| hdr | /altera_hdmi_aux_hdr.v |
| | /altera_hdmi_aux_snk.v |
| | /altera_hdmi_aux_src.v |
| | /altera_hdmi_hdr_infoframe.v |
| | /avalon_st_mutiplexer.v |

***continued...***

| Folders | Files |
|---|---|
| reconfig_mgmt | /mr_compare_pll.v |
| | /mr_compare_rx.v |
| | /mr_rate_detect.v |
| | /mr_reconfig_master_pll.v |
| | /mr_reconfig_master_rx.v |
| | /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 | /a10_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 |

*continued...*

| Folders | Files |
|---|---|
| hdmi_rx | • /hdmi_rx.qsys (Intel Quartus Prime Standard Edition)<br>• /hdmi_rx.ip (Intel Quartus Prime Pro Edition) |
| | /hdmi_rx.sopcinfo (Intel Quartus Prime Standard Edition) |
| hdmi_tx | • /hdmi_tx.qsys (Intel Quartus Prime Standard Edition)<br>• /hdmi_tx.ip (Intel Quartus Prime Pro Edition) |
| | /hdmi_tx.sopcinfo (Intel Quartus Prime Standard Edition) |

**Table 3. Generated Software Files**

| Folders | Files |
|---|---|
| tx_control_src<br>*Note:* The tx_control folder will also contain duplicates of these files. | /i2c.c |
| | /i2c.h |
| | /main.c |
| | /xcvr_gpll_rcfg.c |
| | /xcvr_gpll_rcfg.h |

## 1.2 Hardware and Software Requirements

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

### Hardware

- Intel Arria 10 GX FPGA Development Kit
- HDMI Source (Graphics Processor Unit (GPU))
- HDMI Sink (Monitor)
- Bitec HDMI 2.0 FMC daughter card (Revision 4.0)
- HDMI cables

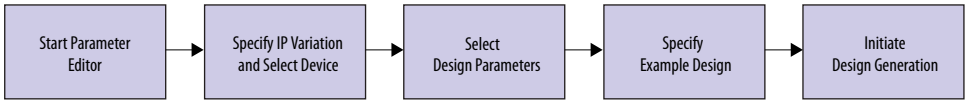
### Software

- Intel Quartus Prime version 17.1 (for hardware testing)
- ModelSim* - Intel FPGA Edition, ModelSim - Intel FPGA Edition Starter Edition, NCSim (Verilog HDL only), Riviera-Pro, or VCS/VCS-MX simulator

## 1.3 Generating the Design

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

**Figure 3. Generating the Design Flow**



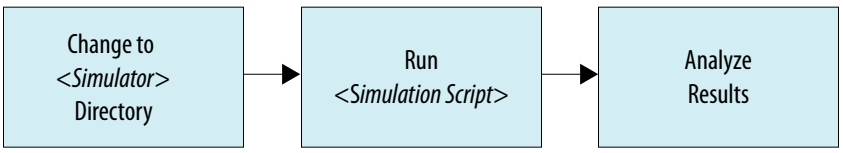
1. Create a project targeting Intel Arria 10 device family and select the desired device.
2. In the IP Catalog, locate and double-click **Intel FPGA HDMI IP Core**. The **New IP Variant** or **New IP Variation** window appears.
3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>.ip or <*your_ip*>.qsys.
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 **Arria 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 Arria 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 Arria 10 GX FPGA Development Kit**, the default device is 10AX115S2F4I1SG.
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<br>`vsim -c -do aldec.do` |
| NCSim | /simulation/cadence | In the command line, type<br>`source ncsim.sh` |

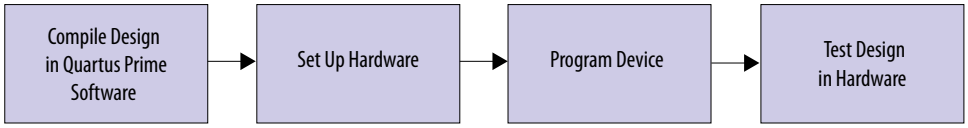
*continued...*

| Simulator | Working Directory | Instructions |
|---|---|---|
| ModelSim | `/simulation/mentor` | In the command line, type<br>`vsim -c -do mentor.do` |
| VCS | `/simulation/synopsys/vcs` | In the command line, type<br>`source vcs_sim.sh` |
| VCS-MX | `/simulation/synopsys/vcsmx` | In the command line, type<br>`source vcsmx_sim.sh` |

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 software and open **project directory/quartus/a10_hdmi2_demo.qpf**.
3. Click **Processing ➤ Start Compilation**.
4. After successful compilation, a `.sof` file will be generated in your specified directory.
5. Connect to the on-board FMCB (J2) Bitec HDMI 2.0 FMC Daughter Card Rev 4.
6. Connect TX (P1) of the Bitec HDMI 2.0 FMC Daughter Card (Revision 4) to an external video source.
7. Connect RX (P2) of the Bitec HDMI 2.0 FMC Daughter Card (Revision 4) 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 Arria 10 device on the development board using the generated `.sof` file (**Tools ➤ Programmer** ).
10. The analyzer should display the video generated from the source.

### Related Links

Intel Arria 10 FPGA Development Kit User Guide

## 1.6 Design Limitation

You may encounter timing violation on the maximum skew constraints required for the designs that use TX PMA and PCS bonding.

Choose the transceiver channels farther away from the Hard IP (HIP) block to meet the maximum skew tolerance constraints requirement.

## 1.7 Intel FPGA HDMI Design Example Parameters

### Table 5. Intel FPGA HDMI Design Example Parameters for Intel Arria 10 Devices

These options are available for Intel Arria 10 devices only.

| Parameter | Value | Description |
|---|---|---|
| **Available Design Example** | | |
| Select Design | Arria 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. |
| **Design Example Files** | | |
| 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. |
| **Generated HDL Format** | | |
| Generate File Format | Verilog, VHDL | Select your preferred HDL format for the generated design example fileset.<br>*Note:* 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. |
| **Target Development Kit** | | |
| Select Board | No Development Kit, Arria 10 GX FPGA Development Kit, Custom Development Kit | Select the board for the targeted design example.<br>• No Development Kit: This option excludes all hardware aspects for the design example. The IP core sets all pin assignments to virtual pins.<br>• Arria 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 **Change Target Device** parameter if your board revision has a different device variant. The IP core sets all pin assignments according to the development kit.<br>• 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. |
| **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 Design Example Detailed Description

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

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

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

### Features

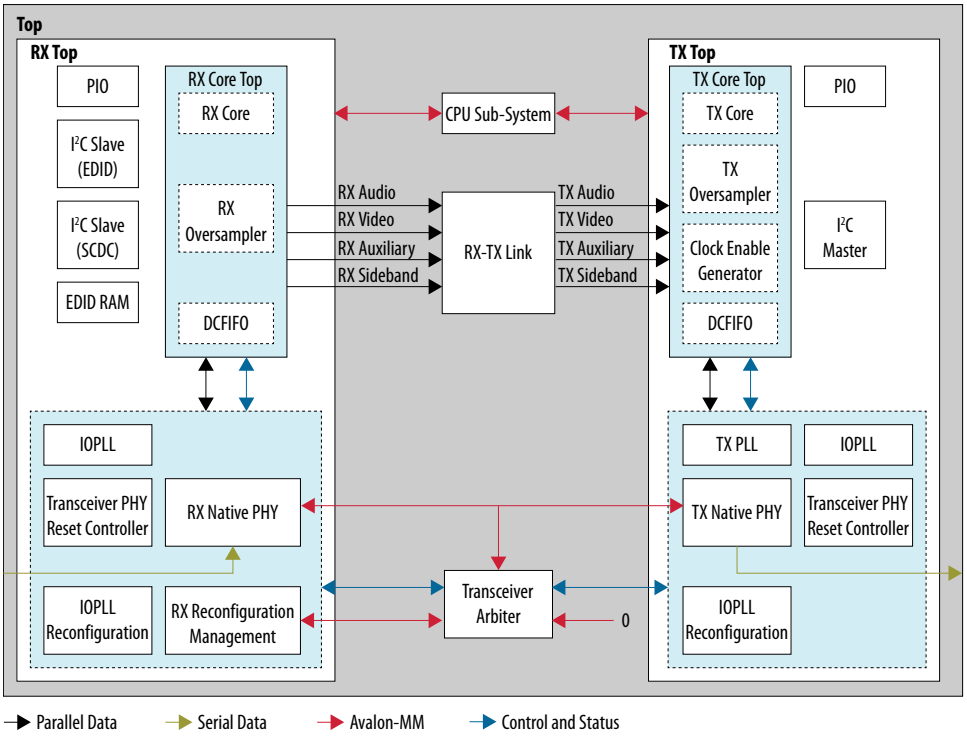
- The design instantiates FIFO buffers to perform a direct HDMI video stream passthrough between the HDMI sink and source.
- The design uses LED status for early debugging stage.
- The design comes with RX and TX only options.
- The design demonstrates the insertion and filtering of Dynamic Range and 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 Example

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

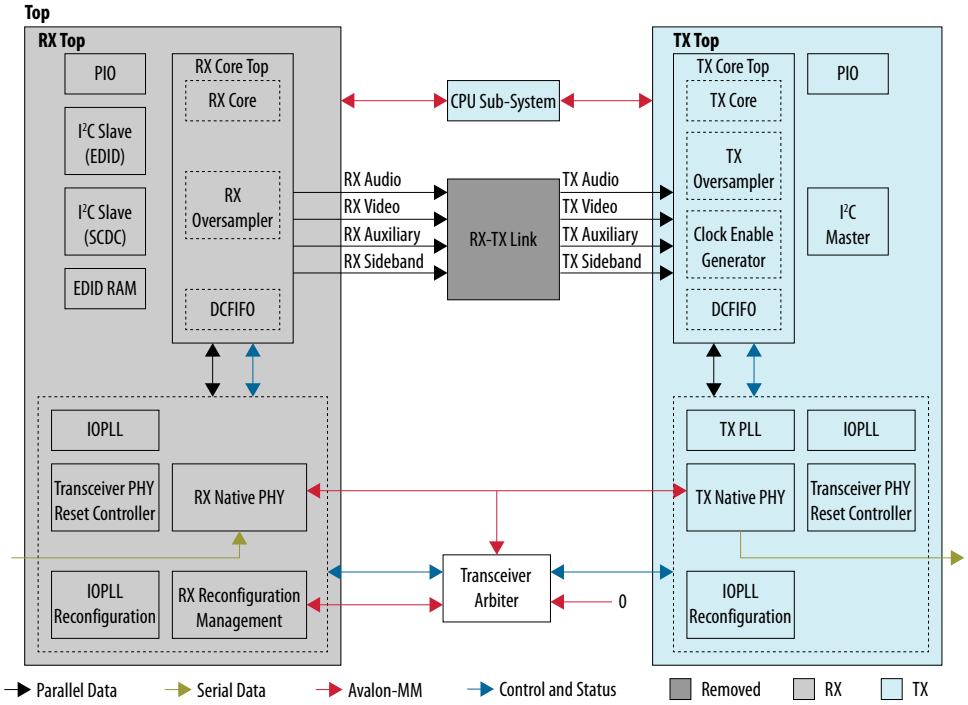
**Figure 5. HDMI RX-TX Retransmit**




To use RX or TX only components, remove the irrelevant blocks from the design

| User Requirement | Preserve | Remove | Add |
|---|---|---|---|
| HDMI RX Only | RX Top | • TX Top<br>• RX-TX Link<br>• CPU Sub-System<br>• Transceiver Arbiter | — |
| HDMI TX Only | TX Top, CPU Sub-System | • RX Top<br>• RX-TX Link<br>• Transceiver Arbiter | Video Pattern Generator (custom module or generated from the VIP Suite IP core) |

**Figure 6. Components Required for RX or TX Only Design**




*Note:* If your design only requires HDMI TX or retransmitting HDMI stream from RX to TX through video frame buffer, supply the TX PLL reference clock directly from an external programmable oscillator. Intel recommends that you do not cascade IOPLL to TX PLL.

### Related Links

Jitter of PLL Cascading or Non-Dedicated Clock Path for Arria 10 PLL Reference Clock
Refer to this solution for workaround if your design clocks experience additional jitter.

## 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 | The RX Core top level consists of: |
| | ***continued...*** |

| Module | Description |
|---|---|
| | • 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.<br>• 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 Arria 10 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.<br>• DCFIFO —The DCFIFO transfers data from the RX transceiver recovered clock domain to the RX link speed clock domain. |
| $I^2C$ Slave | $I^2C$ 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.<br>• The 8-bit $I^2C$ 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 $I^2C$ slave responds to E-EDID data by reading from the RAM.<br>• The $I^2C$ slave-only controller also supports SCDC for HDMI 2.0 operations. The 9-bit $I^2C$ slave address for the SCDC are 0xA8 and 0xA9. When an HPD event occurs, the $I^2C$ slave performs write or read transaction to or from SCDC interface of the HDMI RX core.<br>*Note:* This $I^2C$ slave-only controller for SCDC is not required if HDMI 2.0 is not intended. |
| EDID RAM | The design stores the EDID information using the RAM 1-port IP core. A standard two-wire (clock and data) serial bus protocol ($I^2C$ slave-only controller) transfers the CEA-861-D Compliant E-EDID data structure. This EDID RAM stores the E-EDID information. |
| IOPLL | The IOPLL generates the RX CDR reference clock, link speed clock, and video clock for the incoming TMDS clock.<br>• Output clock 0 (CDR reference clock)<br>• Output clock 1 (Link speed clock)<br>• Output clock 2 (Video clock)<br>*Note:* The default IOPLL configuration is not valid for any HDMI resolution. The IOPLL is reconfigured to the appropriate settings upon power up. |
| Transceiver PHY Reset Controller | 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. |
| RX Native PHY | 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. |
| RX Reconfiguration Management | 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.<br>Refer to the *Multi-Rate Reconfiguration Sequence Flow* figure below. |
| IOPLL Reconfiguration | 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 blocks runs at 100 MHz in Intel Arria 10 devices.<br>Due to IOPLL reconfiguration limitation, apply the `Quartus INI permit_nf_pll_reconfig_out_of_lock=on` during the IOPLL reconfiguration IP generation. |

***continued...***

| Module | Description |
|---|---|
| | To apply the `Quartus INI`, include "`permit_nf_pll_reconfig_out_of_lock=on`" in the `quartus.ini` 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 (pll_hdmi_reconfig) in the Quartus Prime software with the `INI`.<br>*Note:* Without this `Quartus INI`, IOPLL reconfiguration cannot be completed if the IOPLL loses lock during reconfiguration. |
| PIO | The parallel input/output (PIO) block functions as control, status and reset interfaces to or from the CPU sub-system. |

**Figure 7. Multi-Rate Reconfiguration Sequence Flow**

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

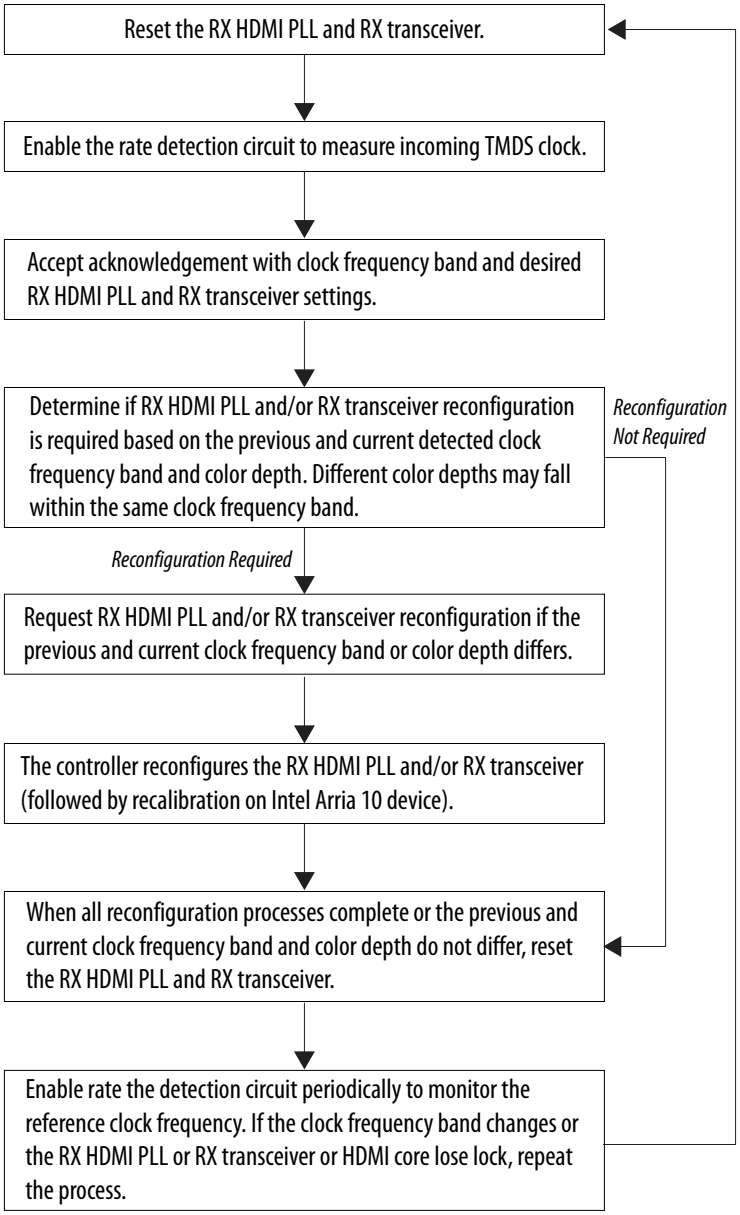



**Table 8. HDMI TX Top Components**

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

*continued...*

| Module | Description |
|---|---|
| | • 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.<br>• 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.<br>(Refer to Table 9 on page 18.)<br>• 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.<br>• 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. |
| $I^2C$ Master | $I^2C$ 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.<br>• As DDC, $I^2C$ Master reads the EDID from the external sink to configure the EDID information EDID RAM in the HDMI RX Top or for video processing.<br>• As SCDC, $I^2C$ master transfers the SCDC data structure from the FPGA source to the external sink for HDMI 2.0 operation. For example, if the outgoing data stream is above 3,400 Mbps, the Nios II processor commands the $I^2C$ master to update the `TMDS_BIT_CLOCK_RATIO` and `SCRAMBLER_ENABLE` bits of the sink SCDC configuration register to 1. |
| IOPLL | The IOPLL supplies the link speed clock and video clock from the incoming TMDS clock.<br>• Output clock 1 (Link speed clock)<br>• Output clock 2 (Video clock)<br>*Note:* The default IOPLL configuration is not valid for any HDMI resolution. The IOPLL is reconfigured to the appropriate settings upon power up. |
| Transceiver PHY Reset Controller | 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.<br>The `tx_ready` 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. |
| Transceiver Native PHY | Hard transceiver block that receives the parallel data from the HDMI TX core and serializes the data from transmitting it.<br>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.<br>*Note:* To meet the HDMI TX inter-channel skew requirement, set the TX channel bonding mode option in the Intel Arria 10 Transceiver Native PHY parameter editor to **PMA and PCS bonding**. You also need to add the maximum skew (`set_max_skew`) constraint requirement to the digital reset signal from the transceiver reset controller (`tx_digitalreset`) as recommended in the *Intel Arria 10 Transceiver PHY User Guide.* |

***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 | 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 blocks runs at 100 MHz in Intel Arria 10 devices.<br>Due to IOPLL reconfiguration limitation, apply the `Quartus INI permit_nf_pll_reconfig_out_of_lock=on` during the IOPLL reconfiguration IP generation.<br>To apply the `Quartus INI`, include `"permit_nf_pll_reconfig_out_of_lock=on"` in the `quartus.ini` 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 (pll_hdmi_reconfig) in the Quartus Prime software with the `INI`.<br>*Note:* Without this `Quartus INI`, IOPLL reconfiguration cannot be completed if the IOPLL loses lock during reconfiguration. |
| 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

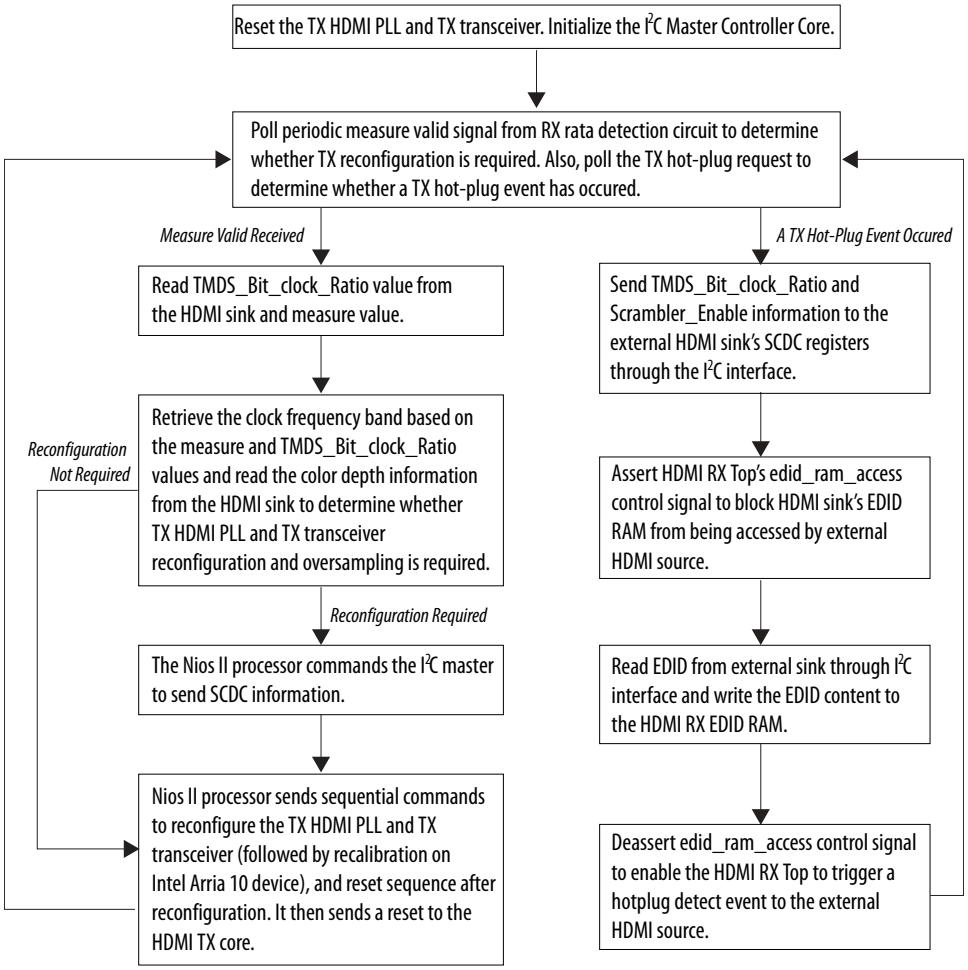
| Module | Description |
|---|---|
| Transceiver Arbiter | This generic functional block prevents transceivers from recalibrating simultaneously when either RX or TX transceivers within the same physical channel require reconfiguration. The simultaneous recalibration impacts applications where RX and TX transceivers within the same channel are assigned to independent IP implementations.<br>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.<br>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.<br>The transceiver arbiter identifies the requester of a reconfiguration through its Avalon-MM reconfiguration interfaces and ensures that the corresponding `tx_reconfig_cal_busy` or `rx_reconfig_cal_busy` is gated accordingly. |

*continued...*

| Module | Description |
|---|---|
| | 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 `tx_reconfig_cal_busy` from asserting and allows `rx_reconfig_cal_busy` to assert. The gating prevents the TX transceiver from being moved to calibration mode unintentionally.<br>*Note:* Because HDMI only requires RX reconfiguration, the `tx_reconfig_mgmt_*` 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. |
| RX-TX Link | • The video data output and synchronization signals from HDMI RX core loop through a DCFIFO across the RX and TX video clock domains.<br>• 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.<br>• 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.<br>• This block also performs external filtering:<br>— Filters the audio data and audio clock regeneration packet from the auxiliary data stream before transmitting to the HDMI TX core auxiliary data port.<br>*Note:* To disable this filtering, press `user_pb[2]`. Enable this filtering to ensure there is no duplication of audio data and audio clock regeneration packet in the retransmitted auxiliary data stream.<br>— 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. |
| CPU Sub-System | The CPU sub-system functions as SCDC and DDC controllers, and source reconfiguration controller.<br>• The source SCDC controller contains the $I^2C$ master controller. The $I^2C$ master controller transfers the SCDC data structure from the FPGA source to the external sink for HDMI 2.0 operation. For example, if the outgoing data stream is 6,000 Mbps, the Nios II processor commands the $I^2C$ master controller to update the `TMDS_BIT_CLOCK_RATIO` and `SCRAMBLER_ENABLE` bits of the sink TMDS configuration register to 1.<br>• The same $I^2C$ master also transfers the DDC data structure (E-EDID) between the HDMI source and external sink.<br>• 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.<br>• 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 Figure 8 on page 20. |

**Figure 8. Reconfiguration Sequence Flow**

The figure illustrates the Nios II software flow that involves the controls for I²C master and HDMI source.




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

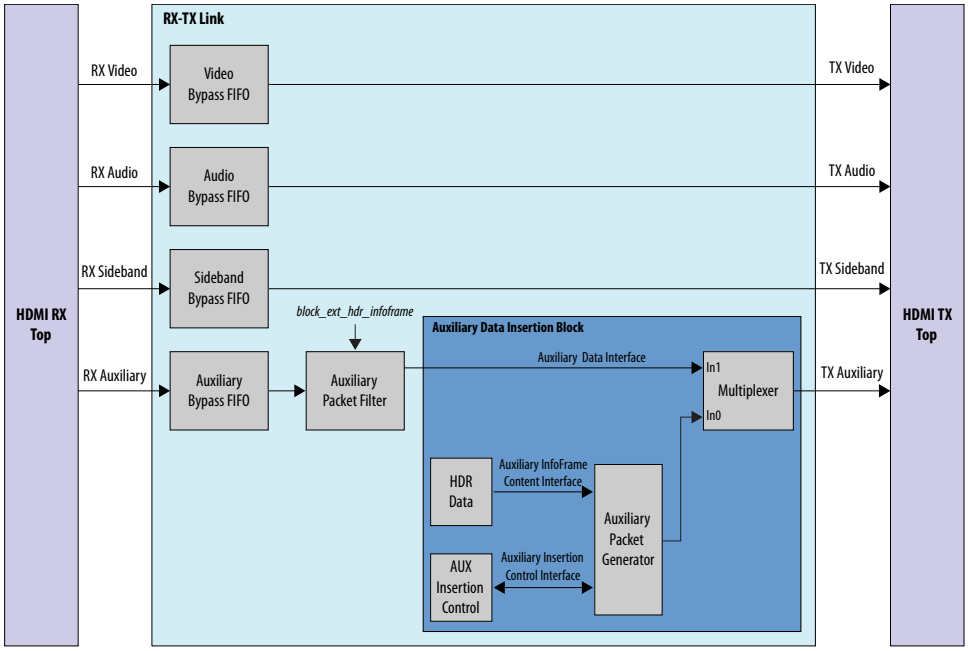
The Intel FPGA HDMI design example includes a demonstration of HDR Infoframe insertion in a RX-TX loopback system.

*HDMI Specification version 2.0a* 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 only need 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 9. 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 |
|---|---|---|---|
| **Clock and Reset** | | | |
| clk_clk | Input | 1 | Clock input. This clock should be connected to the link speed clock. |
| reset_reset_n | Input | 1 | Reset input. |

| **Auxiliary Packet Generator and Multiplexer Signals** | | | |
|---|---|---|---|
| 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_infoframe) 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.

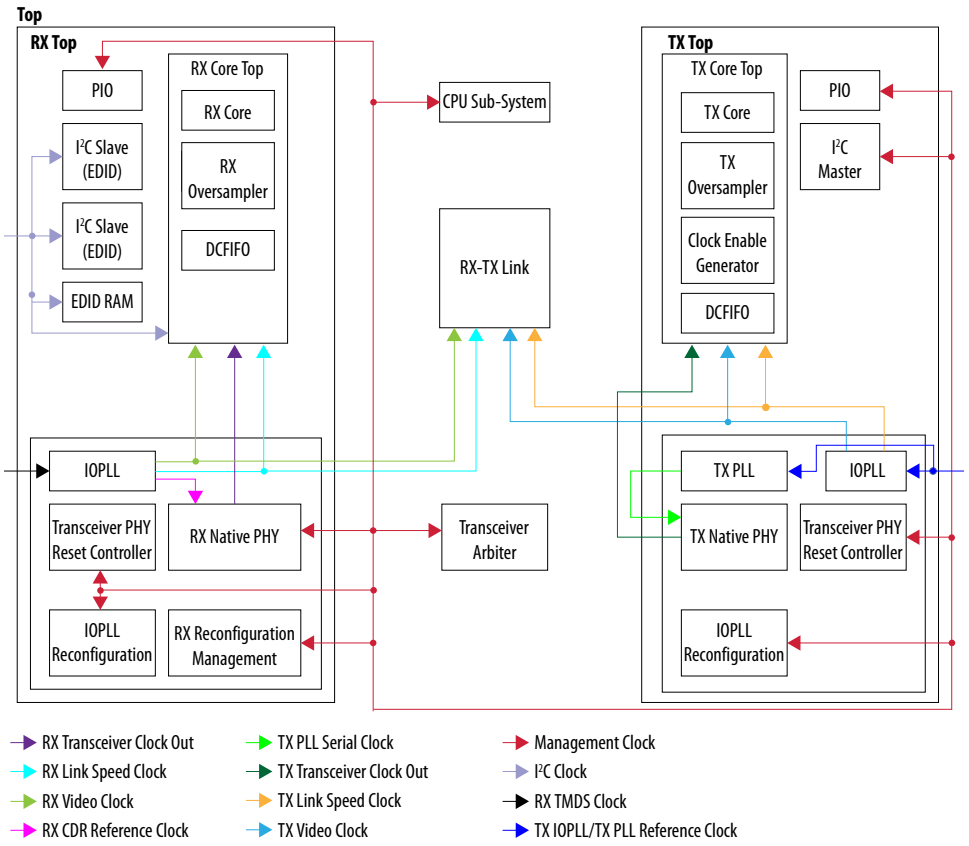
To disable HDR Infoframe insertion and filtering:

1. Set `block_ext_hdr_infoframe` 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 10. 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.<br>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.<br>*Note:* 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.<br>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...*

<table>
<tr><th>Clock</th><th>Signal Name in Design</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>TMDS Bit Clock Ratio</th><th>Link Speed Clock Frequency</th></tr>
<tr><td>0</td><td>TMDS clock frequency/ Symbol per clock</td></tr>
<tr><td>1</td><td>TMDS clock frequency *4 / Symbol per clock</td></tr>
</table>
</td></tr>
<tr><td>TX/RX Video Clock</td><td><code>vid_clk</code></td><td>
Video data clock. The video data clock frequency is derived from the TX link speed clock based on the color depth.
<table>
<tr><th>TMDS Bit Clock Ratio</th><th>Video Data Clock Frequency</th></tr>
<tr><td>0</td><td>TMDS clock/ Symbol per clock/ Color depth factor</td></tr>
<tr><td>1</td><td>TMDS clock *4 / Symbol per clock/ Color depth factor</td></tr>
</table>
<table>
<tr><th>Bits per Color</th><th>Color Depth Factor</th></tr>
<tr><td>8</td><td>1</td></tr>
<tr><td>10</td><td>1.25</td></tr>
<tr><td>12</td><td>1.5</td></tr>
<tr><td>16</td><td>2.0</td></tr>
</table>
</td></tr>
<tr><td>RX TMDS Clock</td><td><code>tmds_clk_in</code></td><td>TMDS clock channel from the HDMI RX and connects to the reference clock to the IOPLL.</td></tr>
<tr><td>RX CDR Reference Clock</td><td><code>iopll_outclk0</code></td><td>
Reference clock to the RX CDR of RX transceiver.
<table>
<tr><th>Data Rate</th><th>RX Reference Clock Frequency</th></tr>
<tr><td>Data rate <1 Gbps</td><td>5× TMDS clock frequency</td></tr>
<tr><td>1 Gbps< Data rate <3.4 Gbps</td><td>TMDS clock frequency</td></tr>
<tr><td>Data rate >3.4 Gbps</td><td>4× TMDS clock frequency</td></tr>
</table>
<ul>
<li>Data Rate <1 Gbps: For oversampling to meet transceiver minimum data rate requirement.</li>
<li>Data Rate >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>
Note: 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.
</td></tr>
<tr><td>RX Transceiver Clock Out</td><td><code>rx_clk</code></td><td>Clock out recovered from the transceiver, and the frequency varies depending on the data rate and symbols per clock.<br>RX transceiver clock out frequency = Transceiver data rate/ (Symbol per clock*10)</td></tr>
<tr><td>Management Clock</td><td><code>mgmt_clk</code></td><td>A free running 100 MHz clock for these components:</td></tr>
</table>

***continued...***

| Clock | Signal Name in Design | Description |
|---|---|---|
| | | • AVMM interfaces for reconfiguration<br>The frequency range requirement is between 100–125 MHz.<br>• PHY reset controller for transceiver reset sequence<br>The frequency range requirement is between 1–500 MHz.<br>• IOPLL Reconfiguration<br>Maximum clock frequency is 100 MHz.<br>• RX Reconfiguration for management<br>• CPU<br>• I²C Master |
| I²C Clock | i2c_clk | A 50 MHz clock input that clocks I²C slave, SCDC registers in the HDMI RX core, and EDID RAM. |

**Related Links**

- 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 |
|---|---|---|---|
| **On-board Oscillator Signal** | | | |
| clk_fpga_b3_p | Input | 1 | 100 MHz free running clock |
| clk_50 | Input | 1 | 50 MHz free running clock |
| **User Push Buttons and LEDs** | | | |
| user_pb | Input | 1 | Push button to control the Intel FPGA HDMI design functionality |
| cpu_resetn | Input | 1 | Global reset |
| user_led_g | Output | 8 | Green LED display |
| user_led_r | Output | 8 | Red LED display |
| **HDMI FMC Daughter Card Pins on FMC Port B** | | | |
| fmcb_gbtclk_m2c_p_0 | Input | 1 | HDMI RX TMDS clock |
| fmcb_dp_m2c_p | Input | 3 | HDMI RX red, green, and blue data channels |
| fmcb_dp_c2m_p | Output | 4 | HDMI TX clock, red, green, and blue data channels |
| fmcb_la_rx_p_9 | Input | 1 | HDMI RX +5V power detect |
| fmcb_la_rx_p_8 | Inout | 1 | HDMI RX hot plug detect |
| fmcb_la_rx_n_8 | Inout | 1 | HDMI RX I²C SDA |

*continued...*

| HDMI FMC Daughter Card Pins on FMC Port B | | | |
|---|---|---|---|
| fmcb_la_tx_p_10 | Input | 1 | HDMI RX $I^2C$ SCL |
| fmcb_la_tx_p_12 | Input | 1 | HDMI TX hot plug detect |
| fmcb_la_tx_n_12 | Inout | 1 | HDMI $I^2C$ SDA |
| fmcb_la_rx_p_10 | Inout | 1 | HDMI $I^2C$ SCL |

## Table 16. HDMI RX 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 |
| 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 |
| **RX Transceiver and IOPLL Signals** | | | |
| 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 | |
| **RX Reconfiguration Management** | | | |
| 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: |

***continued...***

| RX Reconfiguration Management | | | |
|---|---|---|---|
| | | | • 0: No oversampling<br>• 1: 5× oversampling |
| 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 | | | |
|---|---|---|---|
| TMDS_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 | HDMI RX core auxiliary interfaces |
| 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<br>*Note: N* = symbols per clock |
| vid_vsync | Output | *N* | |
| vid_hsync | Output | *N* | |
| vid_de | Output | *N* | |
| mode | Output | 1 | HDMI RX core control and status ports |

***continued...***

| HDMI RX Core Signals | | | |
|---|---|---|---|
| ctrl | Output | *N**6 | *Note: N* = symbols per clock |
| locked | Output | 3 | |
| vid_lock | Output | 1 | |
| in_5v_power | Input | 1 | HDMI RX 5V detect and hotplug detect |
| hdmi_rx_hpd_n | Inout | 1 | |

| $I^2C$ 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.<br>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 |

***continued...***

| TX Transceiver and IOPLL Signals | | | |
|---|---|---|---|
| gxb_tx_cal_busy_out | Output | 4 | TX Native PHY calibration busy signal to the transceiver arbiter |
| 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:<br>• 0: No oversampling<br>• 1: 3× oversampling<br>• 2: 4× oversampling<br>• 3: 5× oversampling |
| 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<br>*Note: N* = Symbols per clock |
| 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 | |

*continued...*

| HDMI TX Core Signals | | | |
|---|---|---|---|
| audio_data | Input | 256 | |
| 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 | HDMI TX core auxiliary interfaces |
| 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<br>*Note: N* = symbols per clock |
| vid_vsync | Input | *N* | |
| vid_hsync | Input | *N* | |
| vid_de | Input | *N* | |

| $I^2C$ and Hot Plug Detect Signals | | | |
|---|---|---|---|
| tx_i2c_avalon_waitrequest | Output | 1 | $I^2C$ 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_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 | Calibration status signal from the RX transceiver |

*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 | $I^2C$ 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 | *N* | HDMI RX video interfaces<br>*Note: N* = symbols per clock |
| hdmi_rx_hsync | Input | *N* | |
| hdmi_rx_vsync | Input | *N* | |
| hdmi_rx_data | Input | *N*48* | |
| 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<br>*Note: N* = symbols per clock |
| 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. Qsys 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_avalon_anti_slave_0_address | Output | 3 | $I^2C$ Master Avalon-MM interfaces |
| oc_i2c_master_av_slave_translator_avalon_anti_slave_0_write | Output | 1 | |
| oc_i2c_master_av_slave_translator_avalon_anti_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 | |
| edid_ram_access_pio_external_connection_export | Output | 1 | EDID RAM access interfaces.<br>Assert edid_ram_access_pio_external_connection_export when you want to write to or read from the EDID RAM on the RX top. Connect EDID RAM access Avalon-MM slave in Platform Designer to the EDID RAM interface on the top-level RX modules. |
| 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 PLL waitrequest |
| 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 | |
| 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 |

***continued...***

| Signal | Direction | Width | Description |
|---|---|---|---|
| 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:<br>• 0: No oversampling<br>• 1: 3× oversampling<br>• 2: 4× oversampling<br>• 3: 5× oversampling |
| 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 hotplug 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 Design Example parameter editor. You can still change the design example settings you made in the parameter editor through the RTL parameters.

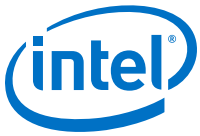


**Table 21. HDMI RX Top Parameters**

| Parameter | Value | Description |
|---|---|---|
| SUPPORT_DEEP_COLOR | • 0: No deep color<br>• 1: Deep color | Determines if the core can encode deep color formats. |
| SUPPORT_AUXILIARY | • 0: No AUX<br>• 1: AUX | Determines if the auxiliary channel encoding is included. |
| SYMBOLS_PER_CLOCK | 2 | Supports only 2 symbols per clock for Intel Arria 10 devices. |
| SUPPORT_AUDIO | • 0: No audio<br>• 1: Audio | Determines if the core can encode audio. |

**Table 22. HDMI TX Top Parameters**

| Parameter | Value | Description |
|---|---|---|
| USE_FPLL | 1 | Supports FPLL as TX PLL only for Intel Arria 10 devices. Always set this parameter to 1. |
| SUPPORT_DEEP_COLOR | • 0: No deep color<br>• 1: Deep color | Determines if the core can encode deep color formats. |
| SUPPORT_AUXILIARY | • 0: No AUX<br>• 1: AUX | Determines if the auxiliary channel encoding is included. |
| SYMBOLS_PER_CLOCK | 2 | Supports only 2 symbols per clock for Intel Arria 10 devices. |
| SUPPORT_AUDIO | • 0: No audio<br>• 1: Audio | Determines if the core can encode audio. |
| POLARITY_INVERSION | • 0: Invert polarity<br>• 1: Do not invert polarity | 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_polinv` 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 Arria 10 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 and User LED Functions**

| LEDs | Function |
|---|---|
| cpu_resetn | Press once to perform system reset. |
| user_pb[0] | Press once to toggle the HPD signal to the standard HDMI source. |
| user_pb[1] | • Press and hold to instruct the TX core to send the DVI encoded signal.<br>• Release to send the HDMI encoded signal. |
| user_pb[2] | • Press and hold to instruct the TX core to stop sending the InfoFrames from the sideband signals.<br>• Release to resume sending the InfoFrames from the sideband signals. |
| USER_LED[0] | RX HDMI PLL lock status.<br>• 0 = Unlocked<br>• 1 = Locked |
| USER_LED[1] | RX transceiver ready status.<br>• 0 = Not ready<br>• 1 = Ready |
| USER_LED[2] | RX HDMI core lock status.<br>• 0 = At least 1 channel unlocked<br>• 1 = All 3 channels locked |
| USER_LED[3] | RX oversampling status.<br>• 0 = Non-oversampled (data rate > 1,000 Mbps in Intel Arria 10 device)<br>• 1 = Oversampled (data rate < 100 Mbps in Intel Arria 10 device) |
| USER_LED[4] | TX HDMI PLL lock status.<br>• 0 = Unlocked<br>• 1 = Locked |
| USER_LED[5] | TX transceiver ready status.<br>• 0 = Not ready<br>• 1 = Ready |
| USER_LED[6] | TX transceiver PLL lock status.<br>• 0 = Unlocked<br>• 1 = Locked |
| USER_LED[7] | TX oversampling status.<br>• 0 = Non-oversampled (data rate > 1,000 Mbps in Intel Arria 10 device)<br>• 1 = Oversampled (data rate < 1,000 Mbps in Intel Arria 10 device) |

## 2.8 Simulation Testbench

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

**Figure 11. Intel FPGA HDMI IP Core Simulation Testbench Block Diagram**

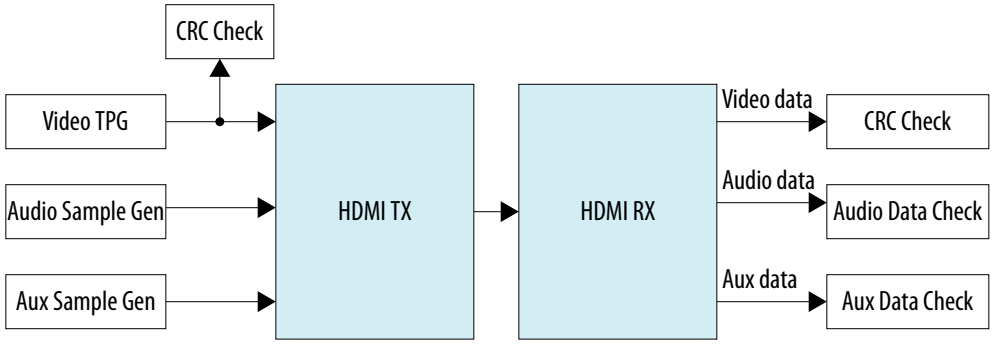



**Table 24. Testbench Components**

| 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 | • The testbench implements CRC checking on the input and output video.<br>• It checks the CRC value of the transmitted data against the CRC calculated in the received video data.<br>• The testbench then performs the checking after detecting 4 stable V-SYNC signals from the receiver. |
| Auxiliary data | • The aux sample generator generates a fixed data to be transmitted from the transmitter.<br>• On the receiver side, the generator compares whether the expected auxiliary data is received and decoded correctly. |
| Audio data | • The audio sample generator generates an incrementing test data pattern to be transmitted through the audio channel.<br>• On the receiver side, the audio data checker checks and compares whether the incrementing test data pattern is received and decoded correctly. |

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 | Yes | Yes |
| VCS/VCS-MX | Yes | Yes |
| Riviera-Pro | Yes | Yes |
| NCSim | Yes | No |

# A Intel FPGA HDMI Design Example User Guide for Intel Arria 10 Devices Archives

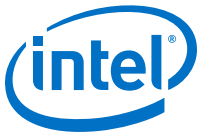
If an IP core version is not listed, the user guide for the previous IP core version applies.

| IP Core Version | User Guide |
|---|---|
| 17.0 | Intel Arria 10 HDMI IP Core Design Example User Guide |
| 16.1 | HDMI IP Core Design Example User Guide |

# B Revision History for Intel FPGA HDMI Design Example User Guide for Intel Arria 10 Devices

| Date | Version | Changes |
|---|---|---|
| November 2017 | 2017.11.06 | • Renamed HDMI IP core to Intel FPGA HDMI as per Intel rebranding.<br>• Changed the term Qsys to Platform Designer.<br>• Added information about Dynamic Range and Mastering InfoFrame (HDR) insertion and filtering feature.<br>• Updated the directory structure:<br>— Added script and sofware folders and files.<br>— Updated common and hdr files.<br>— Removed atx files.<br>— Differentiated files for Intel Quartus Prime Standard Edition and Intel Quartus Prime Pro Edition.<br>• Updated the *Generating the Design* section to add the device used as 10AX115S2F4I1SG.<br>• Edited the transceiver data rate for 50-100 MHz TMDS clock frequency to 2550-5000 Mbps.<br>• Updated the RX-TX link information that you can release the `user_pb[2]` button to disable external filtering.<br>• Updated the Nios II software flow diagram that involves the controls for $I^2C$ master and HDMI source.<br>• Added information about the **Design Example** GUI parameters.<br>• Added HDMI RX and TX Top design parameters.<br>• Added these HDMI RX and TX top-level signals:<br>— `mgmt_clk`<br>— `reset`<br>— `i2c_clk`<br>— `hdmi_clk_in`<br>— Removed these HDMI RX and TX top-level signals:<br>• `version`<br>• `i2c_clk`<br>• Added a note that the transceiver analog setting is tested for the Intel Arria 10 FPGA Development Kit and Bitec HDMI 2.0 Daughter card. You may modify the analog setting for your board. |

***continued...***



| Date | Version | Changes |
|---|---|---|
| | | • Added a link for workaround to avoid jitter of PLL cascading or non-dedicated clock paths for Intel Arria 10 PLL reference clock.<br>• Added a note that you cannot use a transceiver RX pin as a CDR reflk for HDMI RX or as a TX PLL refclk for HDMI TX.<br>• Added a note about how to add `set_max_skew` constraint for designs that use TX PMA and PCS bonding. |
| May 2017 | 2017.05.08 | • Rebranded as Intel.<br>• Changed part number.<br>• Updated the directory structure:<br>— Added hdr files.<br>— Changed `qsys_vip_passthrough.qsys` to `nios.qsys`.<br>— Added files designated for Intel Quartus Prime Pro Edition.<br>• Updated information that the RX-TX Link block also performs external filtering on 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 Avalon ST multiplexer.<br>• Added a note for the Transceiver Native PHY description that to meet the HDMI TX inter-channel skew requirement, you need to set the TX channel bonding mode option in the Arria 10 Transceiver Native PHY parameter editor to **PMA and PCS bonding**.<br>• Updated description for `os` and `measure` signals.<br>• Modified the oversampling factor for different transceiver data rate at each TMDS clock frequency range to support TX FPLL direct clock scheme.<br>• Changed TX IOPLL to TX FPLL cascade clocking scheme to TX FPLL direct scheme.<br>• Added TX PMA reconfiguration signals.<br>• Edited USER_LED[7] oversampling status. 1 indicates oversampled (data rate < 1,000 Mbps in Arria 10 device).<br>• Updated *HDMI Design Example Supported Simulators* table. VHDL not supported for NCSim.<br>• Added link to archived version of the *Arria 10 HDMI IP Core Design Example User Guide*. |
| October 2016 | 2016.10.31 | Initial release. |