UG0863
User Guide
HDMI RX

Microsemi
a Microchip company

a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200863. 2.0 8/21

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 2.0

The following is a summary of the changes made in this revision.

- Added sections Key Features, page 2 and Supported Families, page 2.
- Added Table 2, page 8 and Table 4, page 12.
- Updated Table 3, page 12.
- Replaced Figure 6, page 9.

1.2 Revision 1.0

This is the first publication of this document.

2 Introduction

Microsemi's HDMI RX IP supports receiving video data according to the HDMI standard specification. HDMI RX IP is specifically designed for PolarFire devices supporting HDMI 2.0 at resolutions of 1920x1080 up to 60 frames per second with maximum bandwidths up to 4.5 Gbps. RX IP supports Hot Plug Detect (HPD) for monitoring power on/off and unplug/plug events to indicate communication between HDMI source and HDMI sink. The HDMI Source uses the DDC to read the Sink's Extended Display Identification Data (EDID) to discover the Sink's configuration and/or capabilities. The HDMI RX IP has pre-programmed EDID that can be read by a HDMI source through standard I2C channel.

PolarFire device Transceiver is used along with RX IP to de-serialize serial data into 10-bit data. The data channels in HDMI are allowed to have a considerable skew between them. The HDMI RX IP removes the skew among the data channels by using FIFOs.

The IP converts the Transition Minimized Differential Signaling (TMDS) data received from HDMI source through transceiver into 24-bit RGB pixel data and control signals. The 4 standard control tokens specified in HDMI protocol are used to phase align the data during deserialization.

2.1 Key Features

- Supports HDMI 2.0
- Supports 8-bits color depth
- Supports resolutions up to 1920x1080 at 60 Hz
- Detects Hot-Plug
- Supports Decoding Scheme – TMDS
- Supports Display Data Channel (DDC) and Enhanced Display Data Channel (E-DDC)
- Supports Native and AXI4 Stream Video Interface for video data transfer

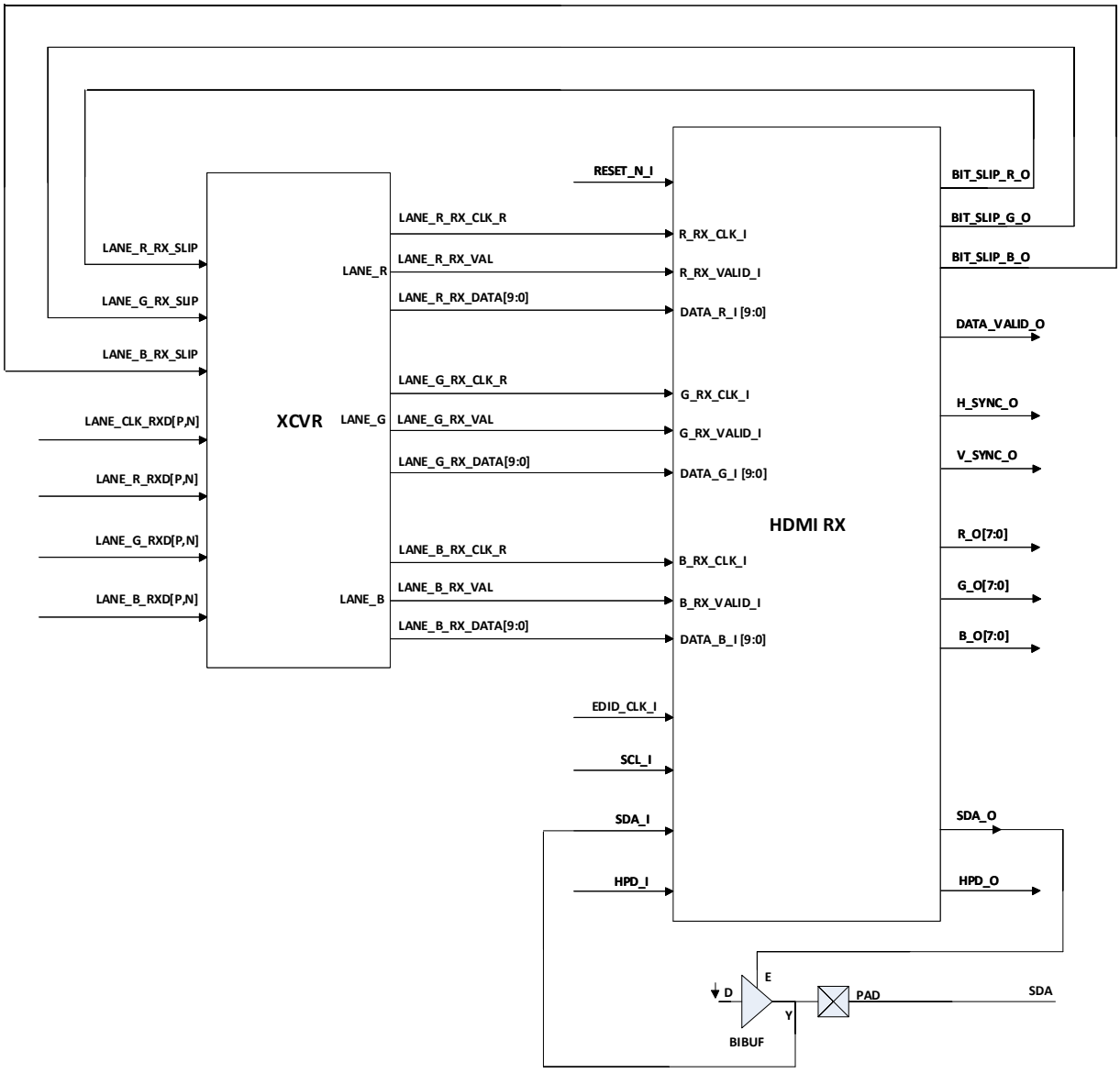
2.2 Supported Families

- PolarFire® SoC
- PolarFire®
- RTG4™
- IGLOO®2
- SmartFusion®2

3 Hardware Implementation

The following figure describes the HDMI RX IP interface with XCVR.

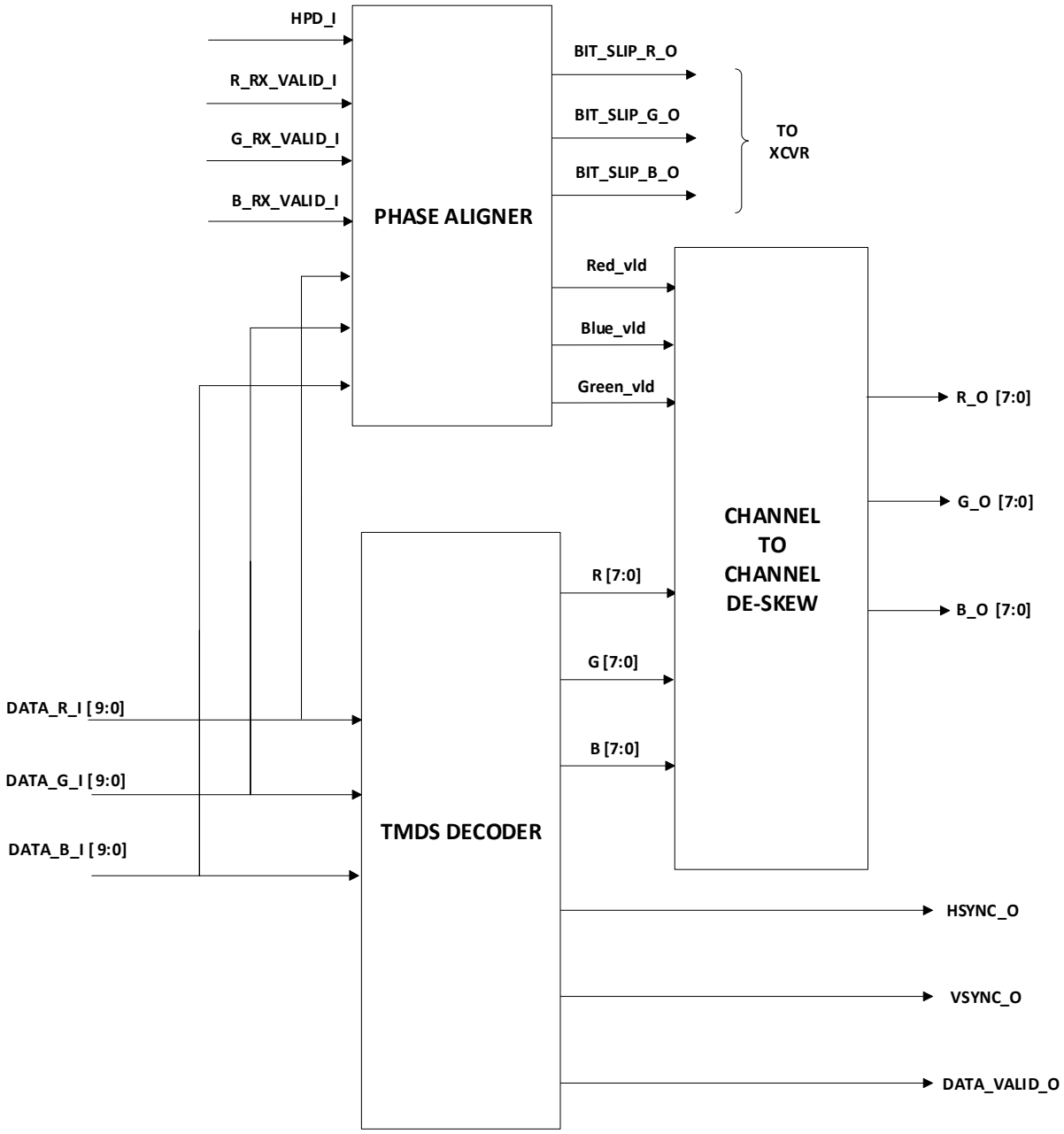
Figure 1 • **HDMI RX Block Diagram**



HDMI RX consists of three stages. In the first stage, the phase aligner aligns the parallel data with respect to control token boundaries using transceiver bit slip. In the second stage, TMDS decoder converts the 10 bit encoded data into 8 bit video pixel data and 2 bit control signals.

In the third stage, the skew between the clocks of R,G and B lanes is removed using FIFOs.

Figure 2 • **Receiver Detailed Block Diagram**



3.1 Phase Aligner

The 10 bit parallel data from the XCVR is not aligned with respect to the TMDS encoded word boundaries. The parallel data needs to be bit shifted and aligned in order to decode the data. Phase aligner correctly aligns the incoming parallel data to word boundaries using bit-slip technique. XCVR in PMA mode allows bit-slip feature, where it adjusts the alignment of 10-bit deserialized word by 1-bit. Every time, after adjusting the 10-bit word by 1-bit slip position, it is compared with any one of the 4 control tokens to lock the position during control period. The 10-bit word is correctly aligned and considered valid for the next stages.

Each color channel has its own phase aligner, the TMDS decoder starts decoding only when all the phase aligners are locked to correct word boundaries.

3.2 TMDS Decoder

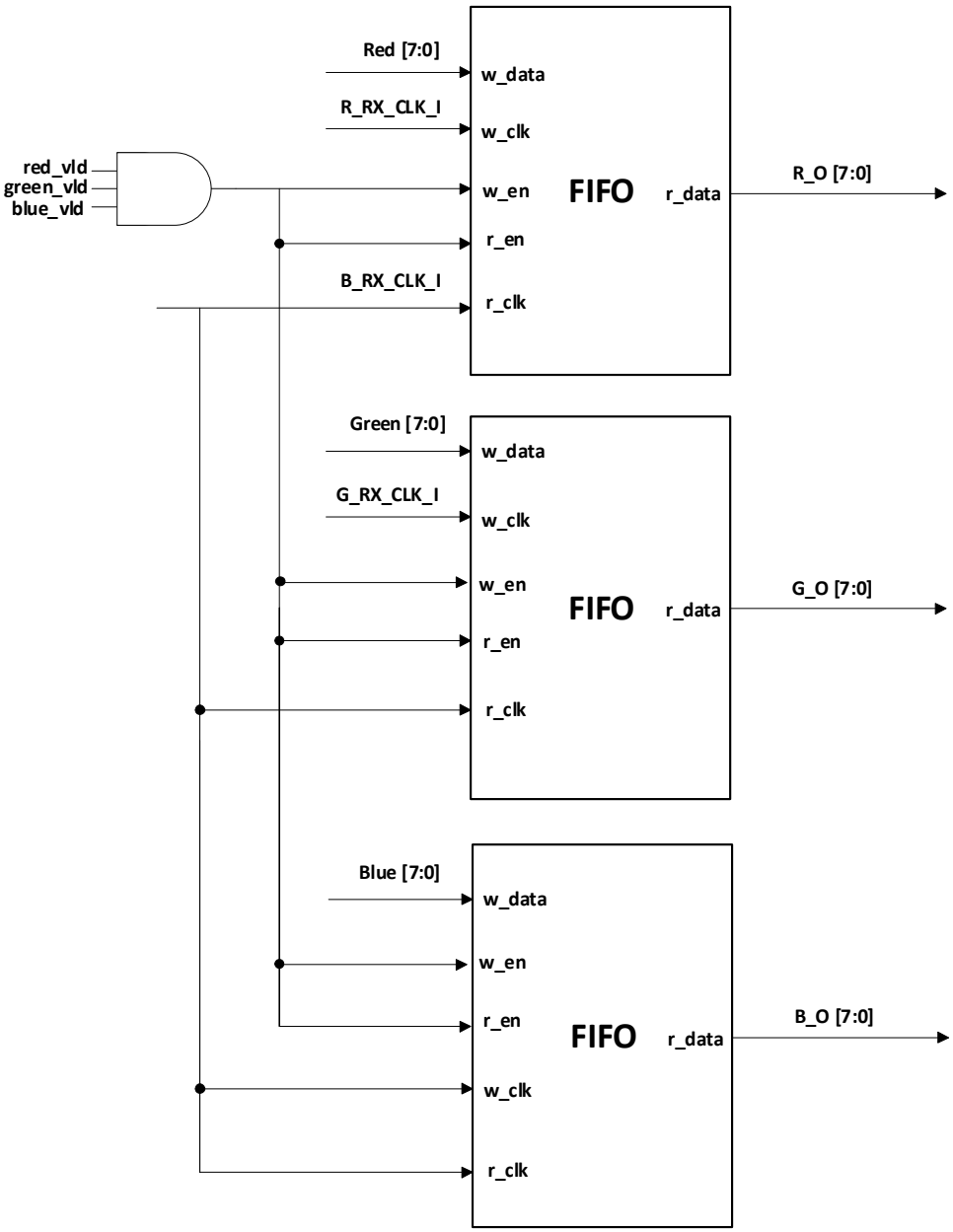
TMDS decoder decodes the 10bit deserialized from transceiver into 8 bit pixel data, HSYNC and VSYNC. The HSYNC and VSYNC signals are generated from the blue channel control signals. The TMDS decoder of each channel operates on its own clock and hence can have a certain skew between the channels.

3.3 Channel to Channel De-skew

A FIFO based deskew logic is used to remove the skew between the channels.

Each channel receives a valid signal from the phase alignment units to indicate, if the incoming 8-bit data from TMDS decoder are valid. If all channels are valid (have achieved phase alignment), FIFO module starts passing data through (continuously writing in and reading out). The dual-clock FIFOs synchronize all three data streams to the blue channel clock to remove the relevant skew. The following diagram describes about channel to channel de-skew technique.

Figure 3 • **Channel to Channel De-skew**



3.4 Display Data Channel (DDC)

The DDC is a communication channel based on the I2C bus specification. The Source will use I2C commands to read information from a Sink's E-EDID with a slave address. The HDMI RX IP uses predefined EDID with multiple resolution support up to 1920x1080 at 60Hz. The EDID represents the display name as **PolarFire display**.

4 Inputs and Outputs

4.1 Ports

The following table describes the input and output ports of HDMI RX IP.

Table 1 • **Input and Output for Native Video Interface**

Signal Name	Direction	Width	Description
RESET_N_I	Input	1 bit	Asynchronous active low reset signal.
R_RX_CLK_I	Input	1 bit	Parallel clock for "R" channel from XCVR.
G_RX_CLK_I	Input	1 bit	Parallel clock for “G” channel from XCVR.
B_RX_CLK_I	Input	1 bit	Parallel clock for "B" channel from XCVR.
R_RX_VALID_I	Input	1 bit	Valid signal from XCVR for “R” channel parallel data.
G_RX_VALID_I	Input	1 bit	Valid signal from XCVR for “G” channel parallel data.
B_RX_VALID_I	Input	1 bit	Valid signal from XCVR for “B” channel parallel data.
DATA_R_I	Input	10 bits	Received “R” channel parallel data from XCVR.
DATA_G_I	Input	10 bits	Received “G” channel parallel data from XCVR.
DATA_B_I	Input	10 bits	Received “B” channel parallel data from XCVR.
SCL_I	Input	1 bit	I2C serial clock input for DDC.
HPD_I	Input	1 bit	Hot Plug Detect Input signal.
SDA_I	Input	1 bit	I2C serial data input for DDC.
EDID_CLK_I	Input	1 bit	System clock for I2C module.
BIT_SLIP_R_O	Output	1 bit	Bit slip signal to “R” channel of transceiver.
BIT_SLIP_G_O	Output	1 bit	Bit slip signal to “G” channel of transceiver.
BIT_SLIP_B_O	Output	1 bit	Bit slip signal to “B” channel of transceiver.
DATA_VALID_O	Output	1 bit	Data valid output (‘1’, for video data ‘0’, for control data).
H_SYNC_O	Output	1 bit	Active horizontal sync pulse.
V_SYNC_O	Output	1 bit	Active vertical sync pulse.
R_O	Output	8 bits	Decoded “R” data.
G_O	Output	8 bits	Decoded “G” data.
B_O	Output	8 bits	Decoded “B” data.
SDA_O	Output	1 bit	I2C serial data output for DDC.
HPD_O	Output	1 bit	Hot Plug Detect Output Signal.

Table 2 • **Input and Output Ports for AXI4 Stream Video Interface**

Port Name	Type	Width	Description
RESET_N_I	Input	1 bit	Active low asynchronous reset signal to design.
SYS_CLK_I	Input	1 bit	System clock
TREADY_O	Output	1 bit	Output target ready signal
TDATA_O	Output	3*G_PIXELS*G_DATA_WIDTH bit	Output Video Data
TVALID_O	Output	1 bit	Output Video Valid
TLAST_O	Output	1 bit	Output frame end signal
TUSER_O	Output	4 bits	bit 0 = End of frame bit 1 = unused bit 2 = unused bit 3 = unused
TSTRB_O	Output	G_DATA_WIDTH/8	Output Video Data strobe
TKEEP_O	Output	G_DATA_WIDTH/8	Output Video Data Keep

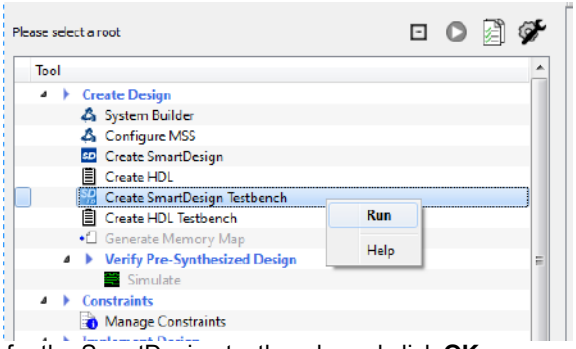
4.2 Testbench Simulation

Testbench is provided to check the functionality of HDMI RX core. Testbench will work only in Native video Interface.

The following steps describe how to simulate the core using the testbench:

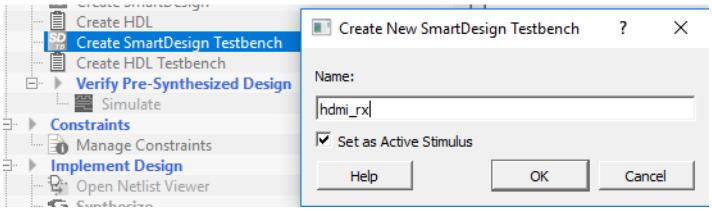
1. In the **Design Flow** window, expand **Create Design**.
2. Right-click **Create SmartDesign testbench** and click **Run**, see the following figure:

Figure 4 • **Creating SmartDesign Testbench**



3. Enter a name for the SmartDesign testbench, and click **OK**.

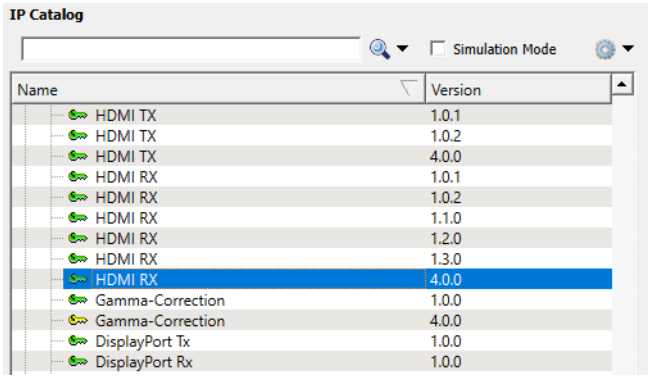
Figure 5 • **Naming SmartDesign Testbench**



SmartDesign testbench is created, and a canvas appears to the right of the Design Flow pane.

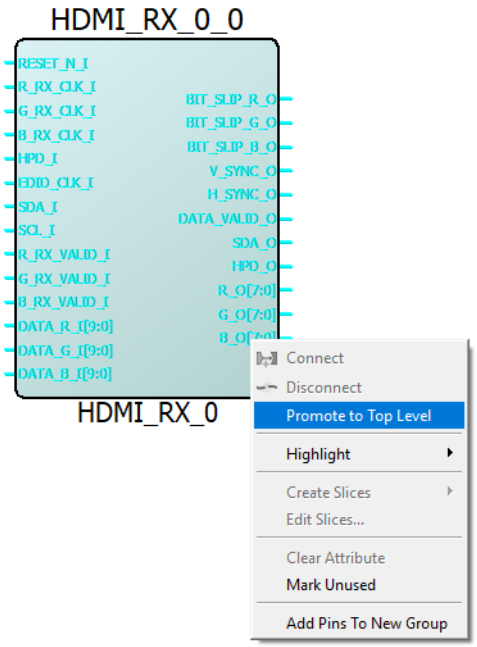
4. In the **Libero SoC Catalog** (**View** > **Windows** > **Catalog**), expand **Solutions-Video**, and drag the **HDMI RX IP** core onto the SmartDesign testbench canvas.

Figure 6 • **HDMI RX in Libero SoC Catalog**



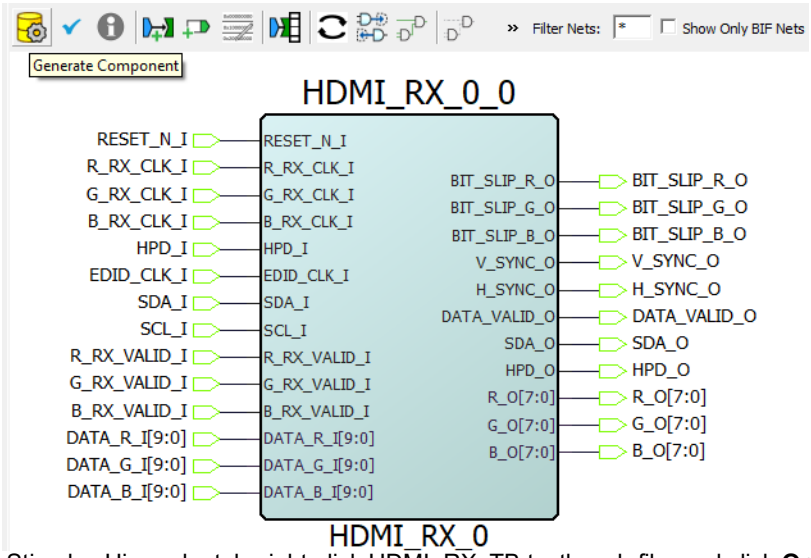
5. Select all the ports, right-click, and select **Promote to Top Level**, as shown in the following figure:

Figure 7 • **Promote to Top Level**



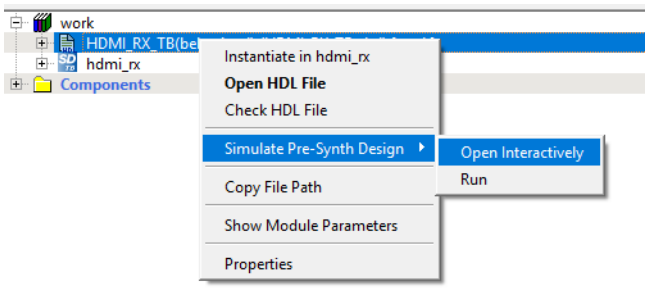
6. Click **Generate Component** from the SmartDesign tool bar, as shown in the following figure:

Figure 8 • **Generate Component**



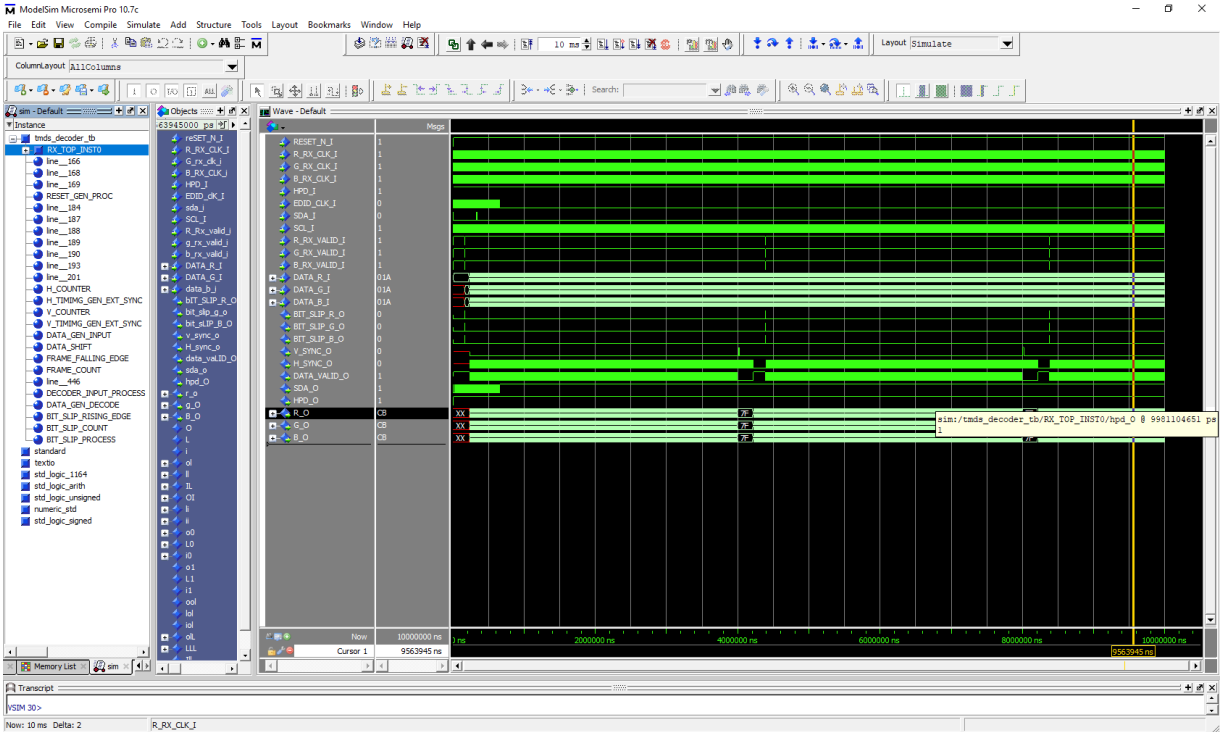
7. On Stimulus Hierarchy tab, right-click HDMI_RX_TB testbench file, and click **Open Interactively** from Simulate Pre-Synth Design.

Figure 9 • **Simulating Testbench**



The ModelSim tool appears with the test bench file loaded as shown in the following figure:

Figure 10 • **ModelSim Tool with HDMI RX Testbench File**

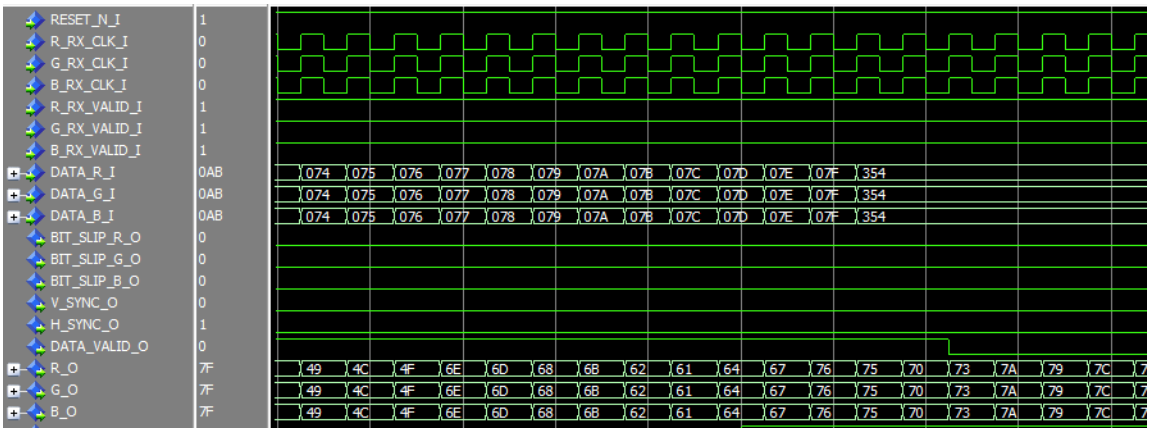


If the simulation is interrupted because of the runtime limit in the DO file, use the run -all command to complete the simulation.

4.3 Simulation Results

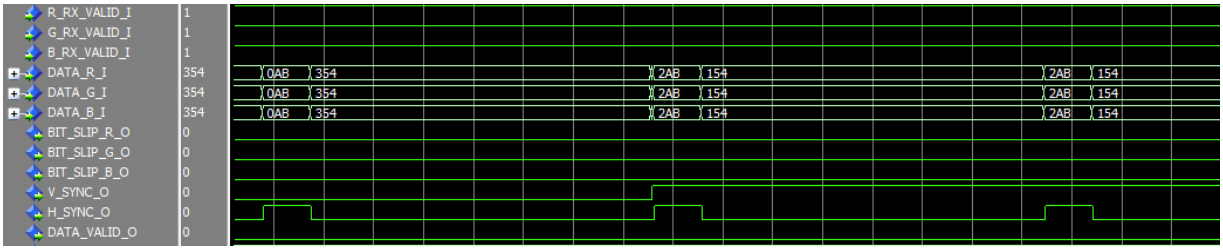
The following timing diagram for HDMI RX IP shows video data and control data periods.

Figure 11 • **Video Data**



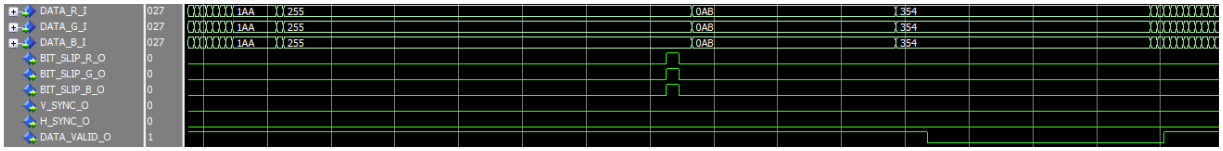
The following diagram shows the hsync and vsync outputs for corresponding control data inputs.

Figure 12 • **Horizontal Sync and Vertical Sync Signals**



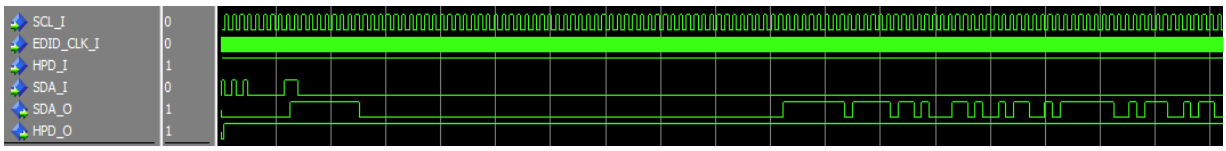
The following diagram shows bit-slip enable and data-adjustment.

Figure 13 • **Bit-slip Generation and Data Adjustment**



The following diagram shows EDID part.

Figure 14 • **EDID Signals**



4.4 Resource Utilization

HDMI RX IP is implemented in PolarFire FPGA (MPF300T - 1FCG1152I Package). The following table describes the resources utilized by the FPGA.

Table 3 • **Resource Utilization for Native Video Interface**

Resource	Usage
DFFs	529
4LUTs	685
LSRAM 18K	6

Table 4 • **Resource Utilization for AXI4 Stream Video Interface**

Resource	Usage
DFFs	532
4LUTs	684
LSRAM 18K	6