Figures

Figure 1  HDMI RX Block Diagram ................................................................. 3
Figure 2  Receiver Detailed Block Diagram .................................................. 4
Figure 3  Channel to Channel De-skew .......................................................... 6
Figure 4  Creating SmartDesign Testbench ................................................... 8
Figure 5  Naming SmartDesign Testbench ..................................................... 8
Figure 6  HDMI RX in Libero SoC Catalog ..................................................... 8
Figure 7  Promote to Top Level ................................................................. 9
Figure 8  Generate Component ............................................................... 9
Figure 9  Simulating Testbench ............................................................... 10
Figure 10  ModelSim Tool with HDMI RX Testbench File .......................... 10
Figure 11  Video Data ........................................................................ 11
Figure 12  Horizontal Sync and Vertical Sync Signals ................................. 11
Figure 13  Bit-slip Generation and Data Adjustment ..................................... 11
Figure 14  EDID Signals ....................................................................... 11
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Inputs and Outputs</td>
<td>7</td>
</tr>
<tr>
<td>Table 2</td>
<td>Resource Utilization</td>
<td>11</td>
</tr>
</tbody>
</table>
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 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. The following are the features for HDMI IP:

- 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)

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.
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

PHASE ALIGNER

TMDS DECODER

CHANNEL TO CHANNEL DE-SKEW

HPD_I
R_RX_VALID_I
G_RX_VALID_I
B_RX_VALID_I

BIT_SLIP_R_O
BIT_SLIP_G_O
BIT_SLIP_B_O

Red_vld
Blue_vld
Green_vld

R_O [7:0]
G_O [7:0]
B_O [7:0]

HSYNC_O
VSYNC_O
DATA_VALID_O

DATA_R_I [9:0]
DATA_G_I [9:0]
DATA_B_I [9:0]
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.
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>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESET_N_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Asynchronous active low reset signal.</td>
</tr>
<tr>
<td>R_RX_CLK_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Parallel clock for &quot;R&quot; channel from XCVR.</td>
</tr>
<tr>
<td>G_RX_CLK_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Parallel clock for &quot;G&quot; channel from XCVR.</td>
</tr>
<tr>
<td>B_RX_CLK_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Parallel clock for &quot;B&quot; channel from XCVR.</td>
</tr>
<tr>
<td>R_RX_VALID_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Valid signal from XCVR for “R” channel parallel data.</td>
</tr>
<tr>
<td>G_RX_VALID_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Valid signal from XCVR for “G” channel parallel data.</td>
</tr>
<tr>
<td>B_RX_VALID_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Valid signal from XCVR for “B” channel parallel data.</td>
</tr>
<tr>
<td>DATA_R_I</td>
<td>Input</td>
<td>10 bits</td>
<td>Received “R” channel parallel data from XCVR.</td>
</tr>
<tr>
<td>DATA_G_I</td>
<td>Input</td>
<td>10 bits</td>
<td>Received “G” channel parallel data from XCVR.</td>
</tr>
<tr>
<td>DATA_B_I</td>
<td>Input</td>
<td>10 bits</td>
<td>Received “B” channel parallel data from XCVR.</td>
</tr>
<tr>
<td>SCL_I</td>
<td>Input</td>
<td>1 bit</td>
<td>I2C serial clock input for DDC.</td>
</tr>
<tr>
<td>HPD_I</td>
<td>Input</td>
<td>1 bit</td>
<td>Hot Plug Detect Input signal.</td>
</tr>
<tr>
<td>SDA_I</td>
<td>Input</td>
<td>1 bit</td>
<td>I2C serial data input for DDC.</td>
</tr>
<tr>
<td>EDID_CLK_I</td>
<td>Input</td>
<td>1 bit</td>
<td>System clock for I2C module.</td>
</tr>
<tr>
<td>BIT_SLIP_R_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Bit slip signal to &quot;R&quot; channel of transceiver.</td>
</tr>
<tr>
<td>BIT_SLIP_G_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Bit slip signal to &quot;G&quot; channel of transceiver.</td>
</tr>
<tr>
<td>BIT_SLIP_B_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Bit slip signal to &quot;B&quot; channel of transceiver.</td>
</tr>
<tr>
<td>DATA_VALID_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Data valid output (‘1’, for video data ‘0’, for control data).</td>
</tr>
<tr>
<td>H_SYNC_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Active horizontal sync pulse.</td>
</tr>
<tr>
<td>V_SYNC_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Active vertical sync pulse.</td>
</tr>
<tr>
<td>R_O</td>
<td>Output</td>
<td>8 bits</td>
<td>Decoded “R” data.</td>
</tr>
<tr>
<td>G_O</td>
<td>Output</td>
<td>8 bits</td>
<td>Decoded “G” data.</td>
</tr>
<tr>
<td>B_O</td>
<td>Output</td>
<td>8 bits</td>
<td>Decoded “B” data.</td>
</tr>
<tr>
<td>SDA_O</td>
<td>Output</td>
<td>1 bit</td>
<td>I2C serial data output for DDC.</td>
</tr>
<tr>
<td>HPD_O</td>
<td>Output</td>
<td>1 bit</td>
<td>Hot Plug Detect Output Signal.</td>
</tr>
</tbody>
</table>
4.2 Testbench Simulation

Testbench is provided to check the functionality of HDMI RX core. 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:
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.
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>
<thead>
<tr>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFFs</td>
<td>514</td>
</tr>
<tr>
<td>4LUTs</td>
<td>710</td>
</tr>
</tbody>
</table>

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

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

The following diagram shows EDID part.