2:1 HDMI Switch Demo

User Guide

FPGA-UG-02036-A

October 2017

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
EBR	Embedded Block RAM
GPIO	General Purpose Input/Output
HDMI	High Definition Multimedia Interface
I^2C	Inter-Integrated Circuit
RGB	Red Green Blue
SCDT	SYNC Detect
SRAM	Static Random Access Memory
VIP	Video Interface Platform
USB	Universal Serial Bus

1. Introduction

This document describes the design and setup procedure for the Lattice Semiconductor 2:1 HDMI Switch demo design which demonstrates the features of SiI1127A, ECP5 and SiI1136 devices. The 2:1 HDMI Switch demo design allows you to switch between 2 HDMI input ports and output the video onto a single HDMI output port.

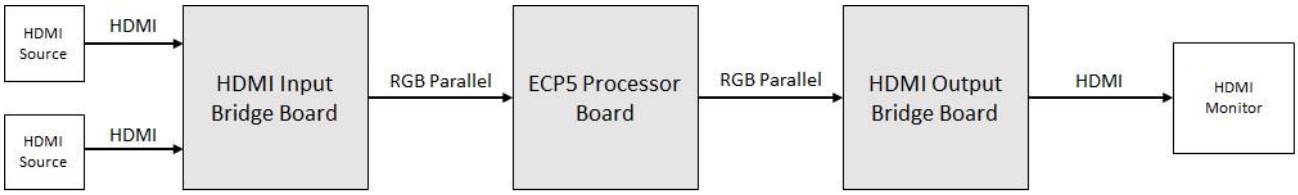

Figure 1.1. 2:1 MIPI CSI-2 to HDMI Bridge System Diagram

The demo consists of three boards:

- HDMI Input Bridge Board
- ECP5 Processor Board
- HDMI Output Bridge Board

Figure 1.2 shows the three boards.

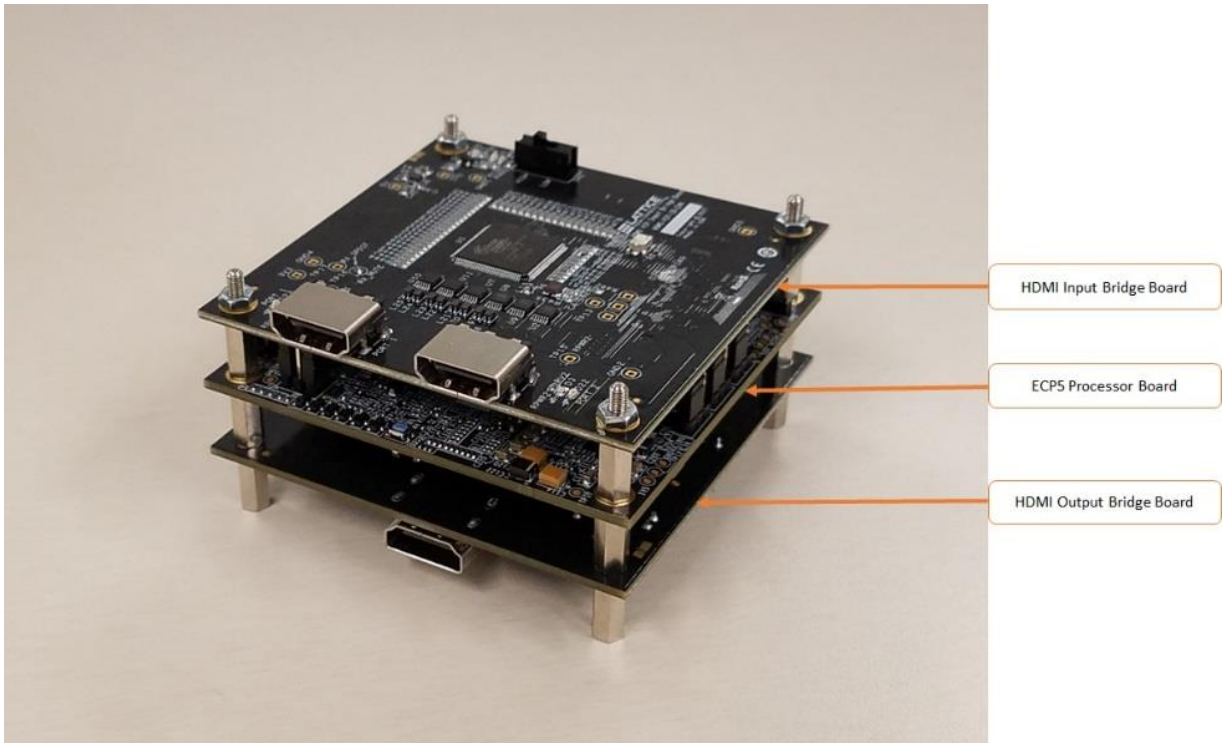

Figure 1.2. 2:1 HDMI Switch Demo Board Stackup

2. Functional Description

2.1. Overview

The 2:1 HDMI Switch demo design allows you to switch between 2 HDMI input ports onto a single HDMI output port.

The HDMI Input Bridge Board receives unencrypted HDMI input video through either Port 1 or 2. The SiI1127A device converts the incoming HDMI video to parallel RGB video. The parallel RGB video is passed through a FIFO on the ECP5 part and sent to the SiI1136 device where it is converted back to HDMI video. The Mico32 system monitors SW2 on the HDMI Input Bridge Board to determine which input port should be transmitted. The Mico32 system on the ECP5 device configures and monitors the status of the SiI1127A and SiI1136 devices through the I^2C interface.

Figure 2.1 shows the functional block diagram of the 2:1 HDMI Switch demo design.

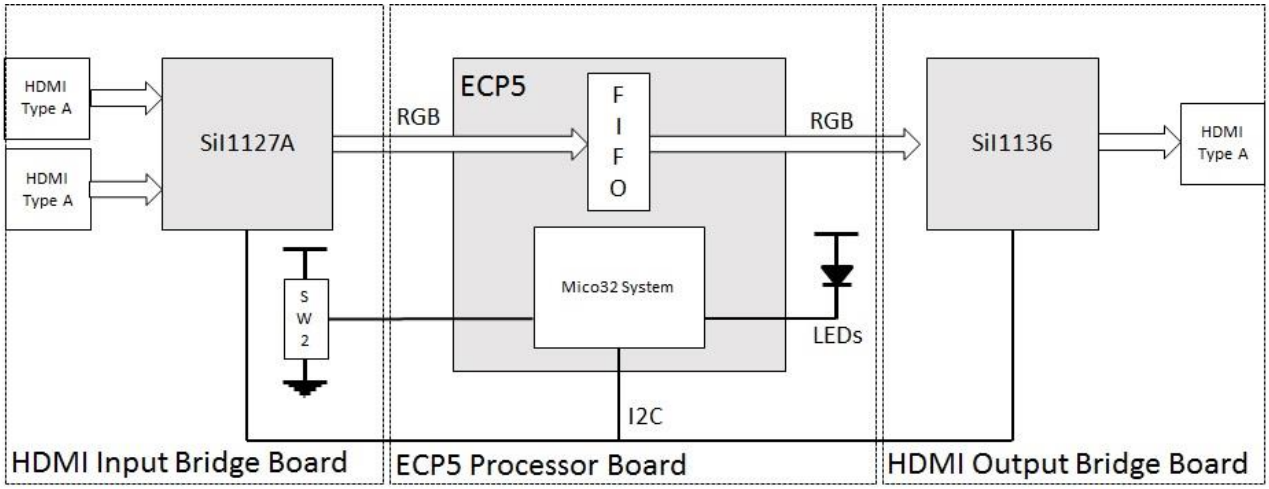

Figure 2.1. 2:1 HDMI Switch Functional Block Diagram

2.2. SiI1127A

The SiI1127A device receives up to 1080p @ 60 Hz HDMI-compliant digital audio and video from either HDMI Type A Connector Port 1 or Port 2 and transmits RGB or YCbCr parallel video and I^2S or SPDIF audio to connectors J1 and J2. The SiI1127A device does not support HDCP decryption, and therefore can only receive unencrypted video.

2.3. SiI1136

The SiI1136 device receives RGB or YCbCr parallel video data and I^2S audio from connectors J1 and J2. Based on the configuration, the SiI1136 device converts the incoming data to HDMI or DVI and outputs to the HDMI Type-A connector.

2.4. ECP5 Mico32 Design

The Mico32 soft processor configures and monitors the SiI1127A and SiI1136 through the I^2C interface. The Mico32 also monitors SW2 on the HDMI Input Bridge to select which port should be transmitted. The LEDs provide a visual indicator of SW2 position. Figure 2.2 on the next page shows the Mico32 System Block diagram. The Mico32 soft processor executes code stored in the EBR.

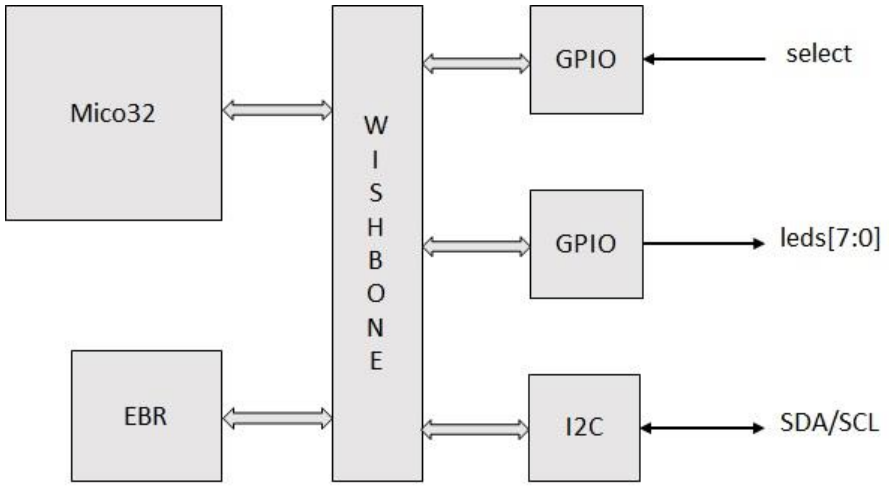

Figure 2.2. Mico32 System

Figure 2.3 shows the software flow of the Mico32 code.

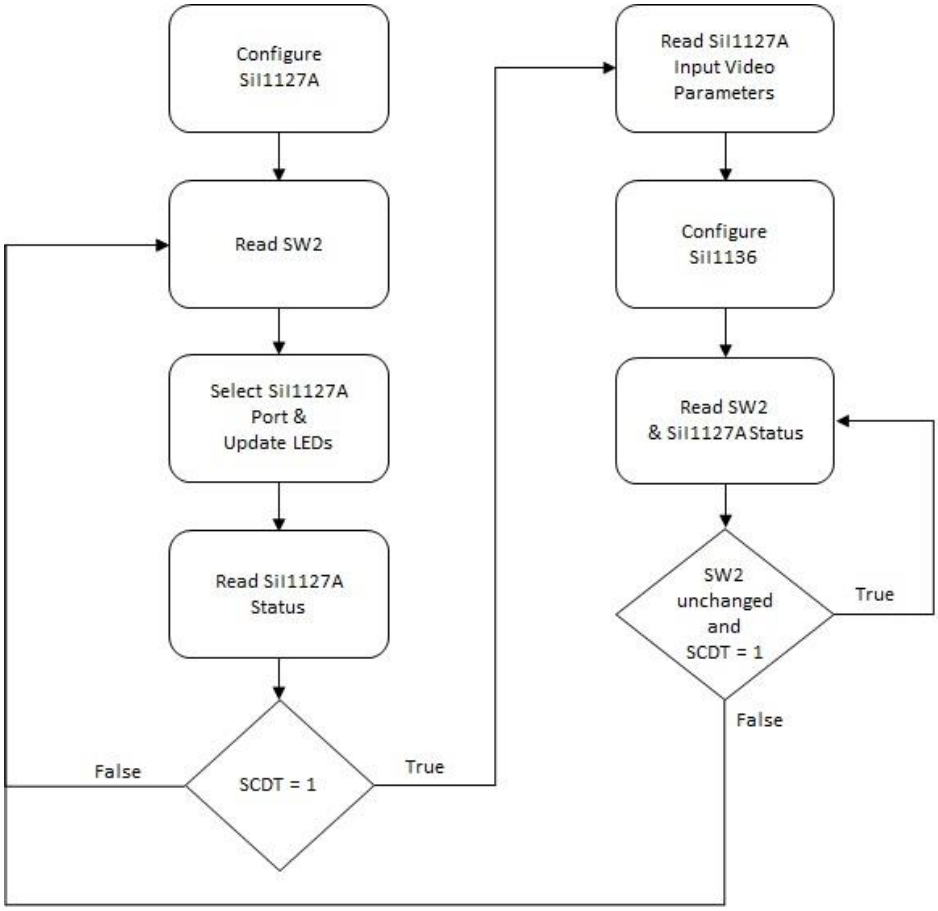

Figure 2.3. Mico32 Software Flow Chart

3. Demo Kit Requirements

- HDMI VIP Input Bridge Board, HDMI-VIP-IB-EVN
- ECP5 VIP Processor Board, LFE5-VIP-P-EVN
- HDMI VIP Output Bridge Board, HDMI-VIP-OB-EVN
- HDMI monitor
- 2 non-encrypted HDMI sources (i.e. laptop with HDMI output, HDMI Media Player for USB drives)
- 3 HDMI cables
- Power adapter (12 V)
- PC or laptop*
- Bit file*
- USB 2.0 Type A to Mini-B cable*
- Lattice Diamond Programmer version 3.7 or higher*

***Note**: Required for programming only.

4. Jumper Settings

Table 4.1. ECP5 Processor Board Jumper Settings

Jumper Name	Description	Comment
J3	Connect 1 and 2, also 5 and 6	Configuration = MSPI
J5	Connect 1 and 2	VCCIO8 = 3.3 V
J6	Connect 1 and 2	VCCIO1 = 3.3 V
J7	Connect 2 and 3	VCCIO0 = 3.3 V
J9	Connect 1 and 2	VCCIO3 = 3.3 V
J50	Connect 1 and 2, also 3 and 5	JTAG ECP5 Only
J51	Connect 2 and 3	VCCIO4 = 3.3 V
J52	Connect 2 and 3	TCK pulled High
J53	Connect 1 and 2	FTDI Resetn pulled High
J55	Connect 2 and 3	VCCIO2 = 3.3 V
—	All other headers should be kept open.	—

5. Demo Procedure

Follow these steps to set up the display demo boards. It is recommended to use the spacers to ensure proper board connection.

1. Connect J1 and J2 connector of HDMI Input Bridge Board to J11 and J10 connector of ECP5 Processor Board.
2. Connect J13 and J12 connector of ECP5 Processor Board to J2 and J1 of HDMI Output Bridge Board.
3. Power up the board and program the ECP5 device.
4. Connect the 12 V wall power adapter cable to J4 of ECP5 Processor Board.
5. Connect Micro USB cable from PC to connector J2 of ECP5 Processor Board.
6. Open Lattice Diamond Programmer.
7. Program the bitstream in to the ECP5 device.

   Note: Optionally, program the SPI Flash with bitstream to allow the demo to be programmed on power up.

   Connect HDMI sources and monitor.
8. Connect one end of HDMI cable to C1 connector of HDMI Output Bridge Board and the other end to monitor.
9. Connect one end of HDMI cable to CN2 and/or CN3 connector of HDMI Input Bridge Board and the other end to a nonencrypted HDMI Source.
10. Position SW2 on HDMI Input Bridge Board to HIGH to select Port 1 (CN2) or LOW to select Port 2 (CN3). The monitor displays the image of the selected port.

Figure 5.1. 2:1 HDMI Switch Demo Setup

6. Demo Package Directory Structure

The key files and directories are list below:

- ECP5_2to1HDMIswitch (Main directory)
  - bitstream
    - hdmi2hdmi.bit (ECP5 bitstream)
  - impl1
  - mico_system
    - HDMI_initialization (Mico32 Software Directory)
      - Opencores_I2C_test.c (Mico32 C Source File)
    - mico32top (Mico32 Project Directory)
      - soc
        - mico32top.msb (Mico32 Platform)
        - mico32top.v (Mico32 System Top Level File)
        - program.mem (Mico32 Executable)
  - video_fifo (Video FIFO Directory)
  - workspace (Lattice Mico System Workspace)
  - hdmi2hdmi.ldf (Diamond Project File)
  - hdmi2hdmi.lpf (Project Settings File)
  - hdmi2hdmi.sty (Project Strategy File)
  - top.v (Top Level HDL File)

7. ECP5 Programing

7.1. Erase the ECP5 Device SRAM Prior to Re-programming

If the ECP5 device is already programmed, either directly or loaded from SPI Flash, follow this procedure to first erase the ECP5 Static Random Access Memory (SRAM), then program the ECP5 SPI Flash using the procedure in the next section, the Program the ECP5 VIP Processor Board section.

Note: If you are erasing the ECP5 SRAM, keep the board powered when re-programming the SPI Flash in the next section, the Program the ECP5 VIP Processor Board section.

1. Connect the 12 V power supply to the barrel plug at J4.
2. Ensure SW2 on ECP5 board is ON to power the board. LEDs should be ON.
3. Connect mini-USB cable from PC to mini-USB connector on ECP5 VIP Processor Board.
4. Launch Diamond Programmer. In the Getting Started dialog box, select “Create a new blank project” and click OK.
5. In the Diamond Programmer main interface, select the device under Device Family and Device as shown in Figure 7.1.

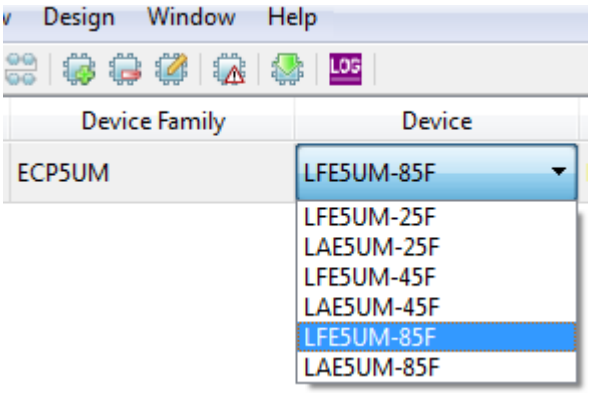

Figure 7.1. Device Selection

6. In the Device Properties dialog box, select “Erase Only” in Operation.

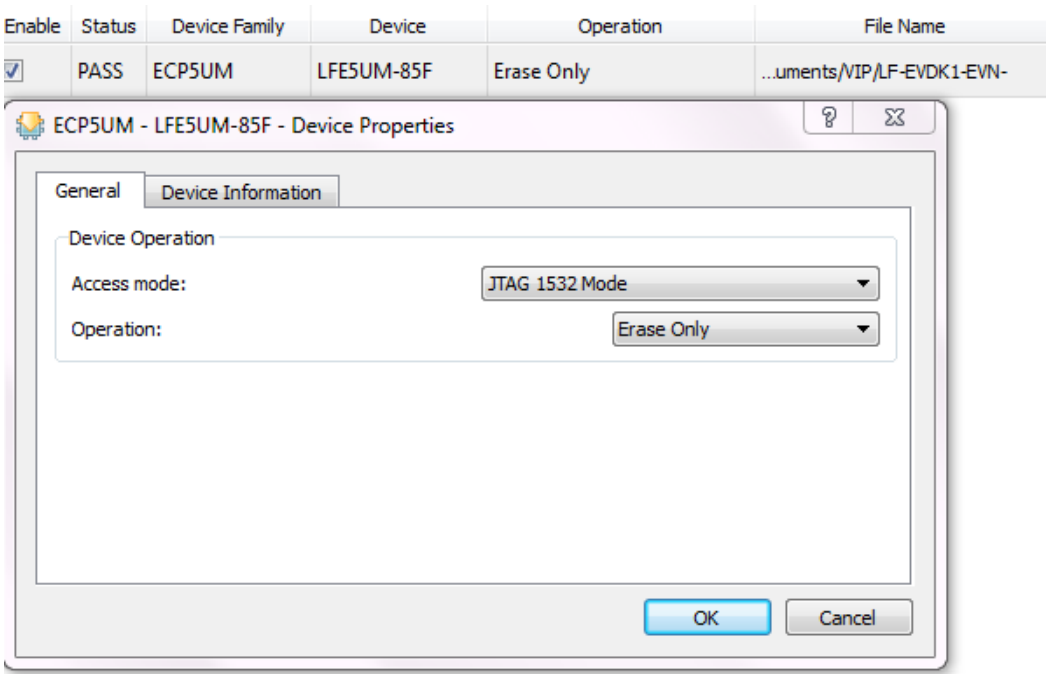

Figure 7.2. Device Operation Options

7. Click the Program button  in Diamond Programmer to start the Erase sequence.

   If you power off/on the board, the SPI Flash will program the ECP5 device again. This requires you to repeat steps 1 through 7.

7.2. Program the ECP5 VIP Processor Board

Before proceeding with the programming described in this section, make sure that the content in the flash memory is properly erased. See the Erase the ECP5 Device SRAM Prior to Re-programming section for more details.

1. Double-click the selection in the Operation box and change “Access mode” to “SPI Flash Background Programming” to bring up the following dialog.
2. Make selections as shown in Figure 7.3. Note that in the “Programming file” field, select the file “hdmi2hdmi.bit”. Click OK.

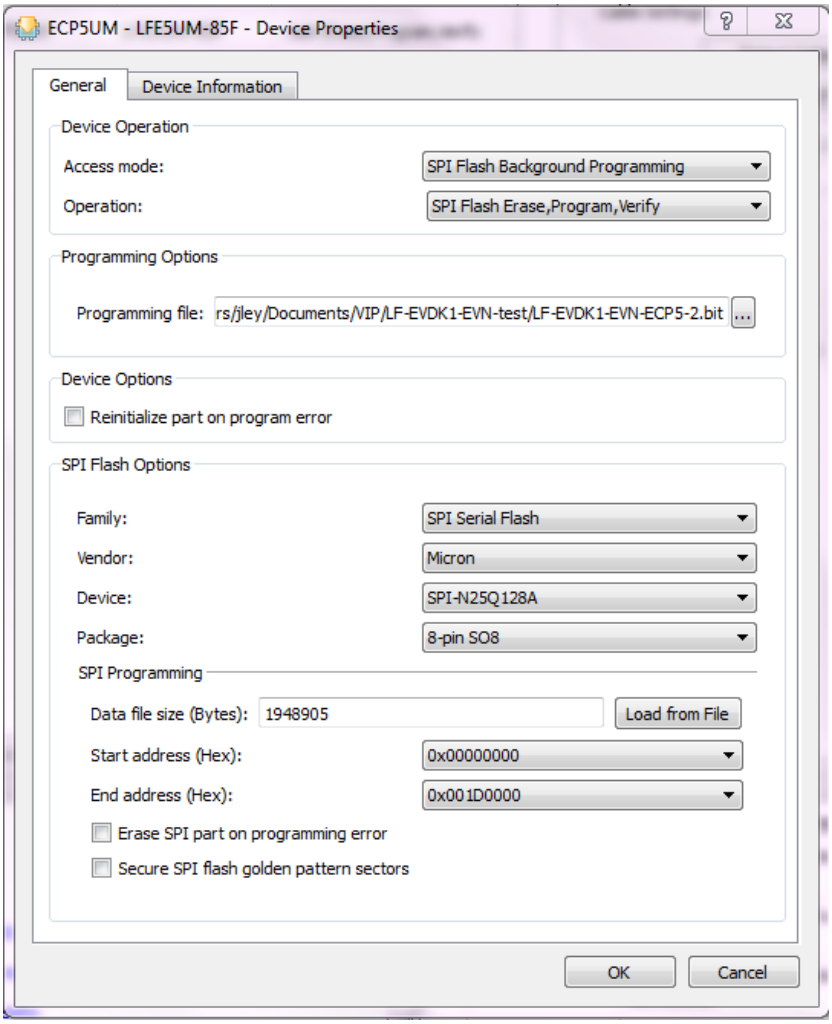

Figure 7.3. Device Properties

Note: Instead of entering the Data file size, just click “Load from File”.

3. Click the Program button  in the main interface to start the programming sequence.
4. Successful programming is displayed in the programmer output console as shown in Figure 7.4.
5. Power cycle the board.

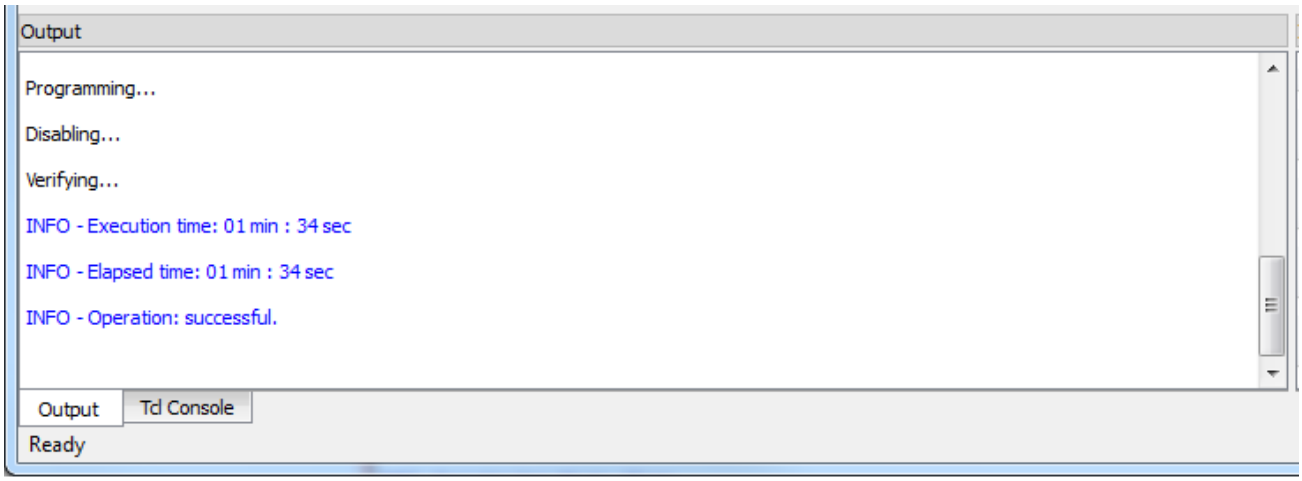

Figure 7.4. Output Console

8. Pinout Information

Table 8.1 lists the ECP5 pinouts used for the demo.

Table 8.1. ECP5 Pinouts

Name	Pin	BANK	IO Direction	IO_TYPE	PULLMODE
clk_i	E17	1	Input	LVCMOS33	—
reset_n	AH1	8	Input	LVCMOS33	UP
SW2	C16	0	Input	LVCMOS33	UP
pixel_clk_i	P27	2	Input	LVCMOS33	—
vsync_i	J30	2	Input	LVCMOS33	—
hsync_i	D30	2	Input	LVCMOS33	—
de_i	K32	2	Input	LVCMOS33	—
red_i[11]	C13	0	Input	LVCMOS33	—
red_i[10]	D13	0	Input	LVCMOS33	—
red_i[9]	C30	2	Input	LVCMOS33	—
red_i[8]	C29	2	Input	LVCMOS33	—
red_i[7]	F28	2	Input	LVCMOS33	—
red_i[6]	F29	2	Input	LVCMOS33	—
red_i[5]	C15	0	Input	LVCMOS33	—
red_i[4]	D15	0	Input	LVCMOS33	—
red_i[3]	J27	2	Input	LVCMOS33	—
red_i[2]	J26	2	Input	LVCMOS33	—
red_i[1]	K26	2	Input	LVCMOS33	—
red_i[0]	K27	2	Input	LVCMOS33	—
green_i[11]	F15	0	Input	LVCMOS33	—
green_i[10]	B4	0	Input	LVCMOS33	—
green_i[9]	A14	0	Input	LVCMOS33	—
green_i[8]	B14	0	Input	LVCMOS33	—
green_i[7]	C11	0	Input	LVCMOS33	—
green_i[6]	D11	0	Input	LVCMOS33	—
green_i[5]	E11	0	Input	LVCMOS33	—
green_i[4]	A10	0	Input	LVCMOS33	—
green_i[3]	D31	2	Input	LVCMOS33	—
green_i[2]	K30	2	Input	LVCMOS33	—
green_i[1]	K29	2	Input	LVCMOS33	—
green_i[0]	J29	2	Input	LVCMOS33	—
blue_i[11]	B10	0	Input	LVCMOS33	—
blue_i[10]	C10	0	Input	LVCMOS33	—
blue_i[9]	A9	0	Input	LVCMOS33	—
blue_i[8]	C9	0	Input	LVCMOS33	—
blue_i[7]	D9	0	Input	LVCMOS33	—
blue_i[6]	F9	0	Input	LVCMOS33	—

Name	Pin	BANK	IO Direction	IO_TYPE	PULLMODE
blue_i[5]	A8	0	Input	LVCMOS33	—
blue_i[4]	A13	0	Input	LVCMOS33	—
blue_i[3]	L30	2	Input	LVCMOS33	—
blue_i[2]	L32	2	Input	LVCMOS33	—
blue_i[1]	H32	2	Input	LVCMOS33	—
blue_i[0]	F32	2	Input	LVCMOS33	—
leds[7]	AM29	4	Output	LVCMOS33	—
leds[6]	AM28	4	Output	LVCMOS33	—
leds[5]	AJ29	4	Output	LVCMOS33	—
leds[4]	AG32	4	Output	LVCMOS33	—
leds[3]	AH32	4	Output	LVCMOS33	—
leds[2]	AK30	4	Output	LVCMOS33	—
leds[1]	AK29	4	Output	LVCMOS33	—
leds[0]	AG30	4	Output	LVCMOS33	—
pixel_clk_o	E25	1	Output	LVCMOS33	—
vsync_o	A25	1	Output	LVCMOS33	—
hsync_o	D25	1	Output	LVCMOS33	—
de_o	C25	1	Output	LVCMOS33	—
red_o[11]	F17	1	Output	LVCMOS33	—
red_o[10]	F25	1	Output	LVCMOS33	—
red_o[9]	W28	3	Output	LVCMOS33	—
red_o[8]	D24	1	Output	LVCMOS33	—
red_o[7]	Y27	3	Output	LVCMOS33	—
red_o[6]	AC26	3	Output	LVCMOS33	—
red_o[5]	AB27	3	Output	LVCMOS33	—
red_o[4]	AB28	3	Output	LVCMOS33	—
red_o[3]	AB30	3	Output	LVCMOS33	—
red_o[2]	AB29	3	Output	LVCMOS33	—
red_o[1]	AD27	3	Output	LVCMOS33	—
red_o[0]	AE27	3	Output	LVCMOS33	—
green_o[11]	T30	3	Output	LVCMOS33	—
green_o[10]	W30	3	Output	LVCMOS33	—
green_o[9]	Y26	3	Output	LVCMOS33	—
green_o[8]	W32	3	Output	LVCMOS33	—
green_o[7]	V32	3	Output	LVCMOS33	—
green_o[6]	AB31	3	Output	LVCMOS33	—
green_o[5]	AC30	3	Output	LVCMOS33	—
green_o[4]	T32	3	Output	LVCMOS33	—
green_o[3]	A24	1	Output	LVCMOS33	—
green_o[2]	R26	3	Output	LVCMOS33	—
green_o[1]	T26	3	Output	LVCMOS33	—

Name	Pin	BANK	IO Direction	IO_TYPE	PULLMODE
green_o[0]	AD26	3	Output	LVCMOS33	—
blue_o[11]	AD32	3	Output	LVCMOS33	—
blue_o[10]	AC32	3	Output	LVCMOS33	—
blue_o[9]	AB32	3	Output	LVCMOS33	—
blue_o[8]	AC31	3	Output	LVCMOS33	—
blue_o[7]	V26	3	Output	LVCMOS33	—
blue_o[6]	V27	3	Output	LVCMOS33	—
blue_o[5]	U28	3	Output	LVCMOS33	—
blue_o[4]	T29	3	Output	LVCMOS33	—
blue_o[3]	W31	3	Output	LVCMOS33	—
blue_o[2]	Y32	3	Output	LVCMOS33	—
blue_o[1]	R32	3	Output	LVCMOS33	—
blue_o[0]	T31	3	Output	LVCMOS33	—
SDA	AJ1	8	Bidir	LVCMOS33	—
SCL	AG1	8	Bidir	LVCMOS33	—

9. Ordering Information

Table 9.1. Ordering Information

Description	Ordering Part Number
HDMI VIP Input Bridge Board	HDMI-VIP-IB-EVN
ECP5 VIP Processor Board	LFE5-VIP-P-EVN
HDMI VIP Output Bridge Board	HDMI-VIP-0B-EVN

10. FAQ

Question: My board was working first. But when I cycle the power, it does not work anymore. Why?

Answer: There is a known warm issue caused by a boot from flash.

Issue description: Some boards fail to configure from SPI Flash after the board has been running for a period of time and has become warmer. Once board cools down, the configuration is successful.

Workaround: The problem is believed to be caused by the loading on the SPI interface. Removing the connection to the downstream board appears to solve the issue. On the ECP5 board, remove pin 27 (SCLK) and pin 31 (MISO) from the inside of connector J12. The yellow circle in the following figure depicts the position of the removed pins.

References

This is a list of the related documents that are available from your Lattice Semiconductor sales representative.

Document	Title
SiI-DS-1059	SiI9127A/SiI1127A HDMI Receiver with Deep Color Output
SiI-PR-1019	SiI9223/9233/9127 HDMI Receivers Programmer's Reference (The Programmer's Reference requires an NDA with Lattice Semiconductor)
FPGA-DS-02012	ECP5 and ECP5-5G Family Data Sheet
SiI-DS-1084	SiI9136-3/SiI1136 HDMI Deep Color Transmitter
SiI-PR-1060	SiI9136-3 and SiI9334 HDMI Transmitter Programmer's Reference (The Programmer's Reference requires an NDA with Lattice Semiconductor)
FPGA-UG-02015	Lattice Embedded Vision Development Kit User Guide
FPGA-EB-02001	ECP5 VIP Processing Board
FPGA-EB-02003	HDMI VIP Output Bridge Board
FPGA-EB-02008	HDMI VIP Input Bridge Board

Technical Support

For assistance, submit a technical support case at www.latticesemi.com/techsupport.

Revision History

Revision A, October 2017

First production release.

7th Floor, 111 SW 5th Avenue
Portland, OR 97204, USA
T 503.268.8000
www.latticesemi.com