3D Depth Mapping Demonstration

User Guide

FPGA-UG-02089-1.0

May 2019

Contents

Figures

Tables

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CSI	Camera Serial Interface
EVDK	Embedded Vision Development Kit
GPIO	General Purpose Input/Output
HDMI	High Definition Multimedia Interface
I^2C	Inter-Integrated Circuit
MIPI	Mobile Industry Processing Interface
SGBM	Semi-Global Block Matching
VIP	Video Interface Platform
USB	Universal Serial Bus

1. Introduction

This document describes the design and setup procedure for the Lattice Embedded Vision Development Kit (EVDK) to demonstrate 3D Depth Mapping using the Lattice Semiconductor's Semi-Global Block Matching (SGBM) algorithm. The SGBM algorithm is a 3D depth map generation block that generates a dense depth map of a scene from stereo camera inputs in real time. The semi-global optimization approach of SGBM is hardware-efficient and shows high-quality depth result. The algorithm can be used in various applications, such as distance measurement, object detection, and navigation with obstacle avoidance.

The key features of the SGBM algorithm include:

- Implementation of Semi-Global Block Matching algorithm with four scan-line optimization.
- Input:
  - Two side by side 320 x 240 stereo video streams
  - 8-bit RGB color
- Output:
  - 320 x 240 depth map (true disparity-displaying portion: 256 x 240)
  - 64-level depth output
- Up to 60 fps operation

Figure 1.1. 3D Depth Mapping Demonstration

2. Functional Description

The 3D Depth Mapping demonstration processes video from two Sony IMX214 camera sensors and displays the disparity between the images over HDMI as shown in Figure 2.1. The disparity is calculated using the Lattice SGBM algorithm. The maximum disparity of a centered window is also outputted to the LEDs on the ECP5™ VIP Processor Board.

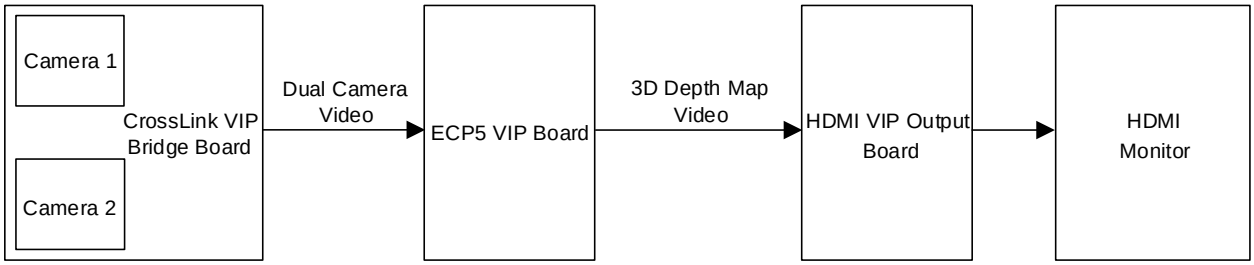

Figure 2.1. 3D Depth Mapping Design Block Diagram

The demonstration uses the Embedded Vision Development Kit, which consists of three boards:

- CrossLink™ Video Interface Platform (VIP) Input Bridge Board
- ECP5 VIP Processor Board
- HDMI VIP Output Bridge Board

For more information on the Embedded Vision Development Kit, visit www.latticesemi.com/en/Products/DevelopmentBoardsAndKits/EmbeddedVisionDevelopmentKit.aspx.

2.1. CrossLink

The CrossLink design receives source-synchronous MIPI CSI-2 data from two IMX214 camera sensors, reserializes the serial data into bytes, and extracts the control signal from MIPI data packets. The byte data is sent to the Byte to Pixel module which converts the byte data into RAW10 data. The RAW10 data from each camera is sent to a separate debayer which converts the RAW10 data into parallel RGB and downscales from 1920 x 1080 to 480 x 270. Each 4 x 4 pixel block is converted into a four 10-bit pixel packet, consisting of Green, Blue, Red and Dummy as shown Figure 2.3. The two streams are sent to an aggregator, which interlaces the lines of video from each stream while also inserting a blank line between the two as shown in Figure 2.4. This interlaced video is sent to the ECP5 VIP Processor Board for 3D depth mapping.

Figure 2.2 shows the CrossLink functional block diagram.

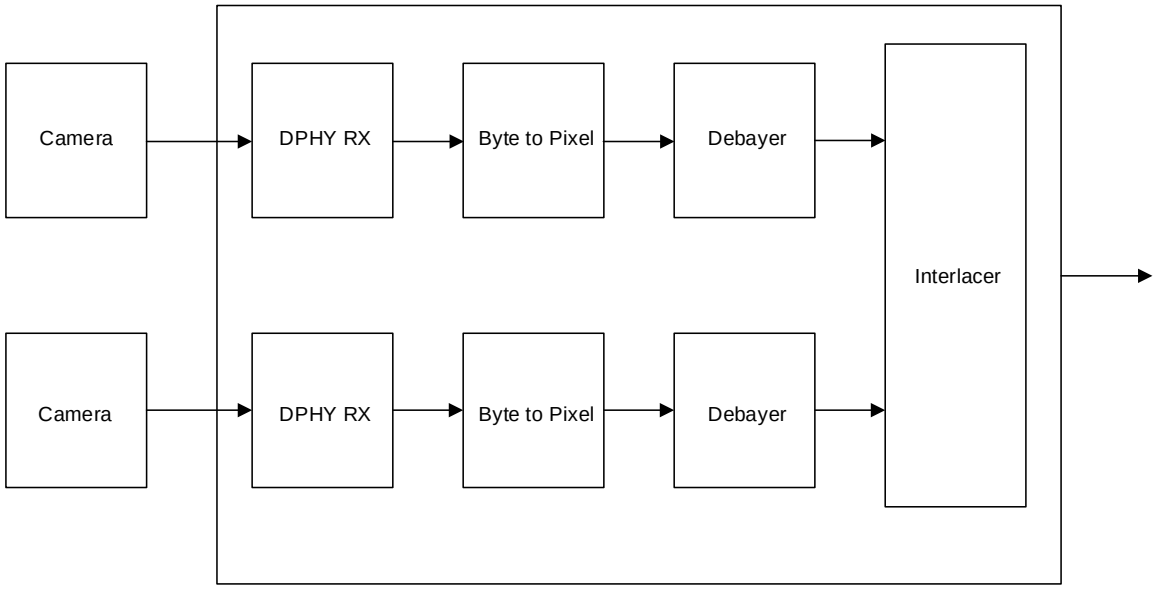

Figure 2.2. CrossLink Functional Block Diagram

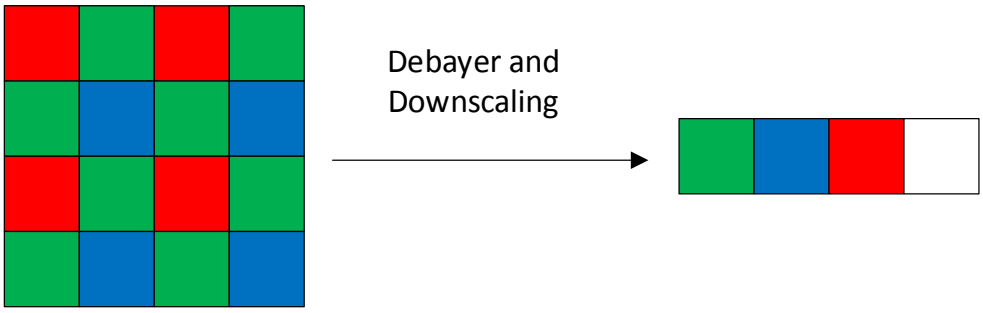

Figure 2.3. Debayer and Downscaling

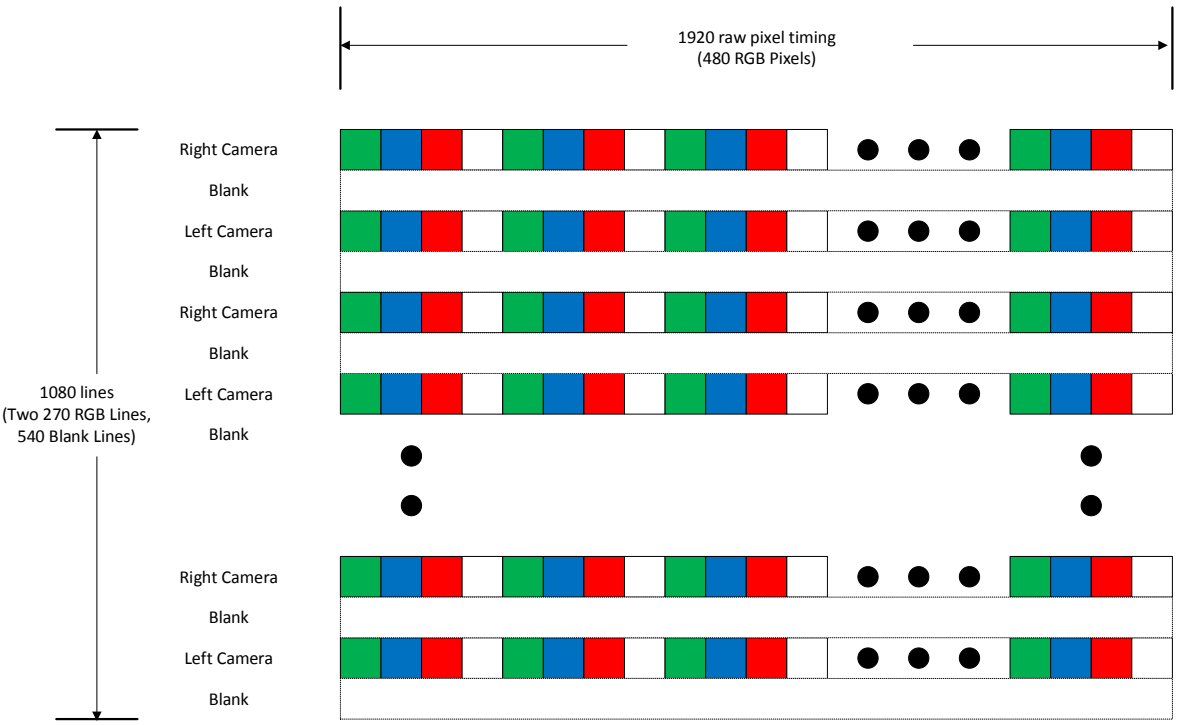

Figure 2.4. Video Output Format

2.2. ECP5

The ECP5 FPGA receives and buffers the interlaced video from the CrossLink VIP Bridge Board. The Rectification block separates the two 480 x 270 images and rectifies both images to 320 x 240 to ensure they are on a common plane. The rectification table is stored on the SD Card and brought over to the DDR memory by the Table Loader on power-up. The two rectified images are then passed to the Semi-Global Block Matching IP, which matches the common points between both images and calculates the disparity. This disparity map is sent to the Object Detection, which filters the disparity map and provides the maximum disparity in the center window. The Video Output block converts the 320 x 240 disparity to 640 x 480 at 60 fps video output, which is then sent to the HDMI VIP Output Board.

Figure 2.5 shows the ECP5 3D Depth Mapping Block Diagram, which contains the following blocks:

- **Buffering** – Buffers the incoming video into DDR memory.
- **Rectification** – Separates the video into right and left images, and rectifies them so they are both on a common plane.
- **Semi-Global Block Matching** – Provides 64 levels of disparity between two 320 x 240 images at 60 fps.
- **Object Detection** – Performs a 5 x 5 filtering on the disparity and outputs the highest disparity per frame.
- **Video Retiming** – Converts the 320 x 240 disparity map to 640 x 480 video.
- **SDRAM Controller/Arbiter** – Manages the read/write to the DDR memory.
- **Table Loader** – Reads the Rectification Table from the SD Card and loads data into DDR Memory. It also reads the camera offset and provides it to the I²C Controller.
- **I²C Controller** – Programs the Camera Sensors on the CrossLink VIP Bridge Board and the Sil1136 HDMI Transmitter on the HDMI VIP Output Board.

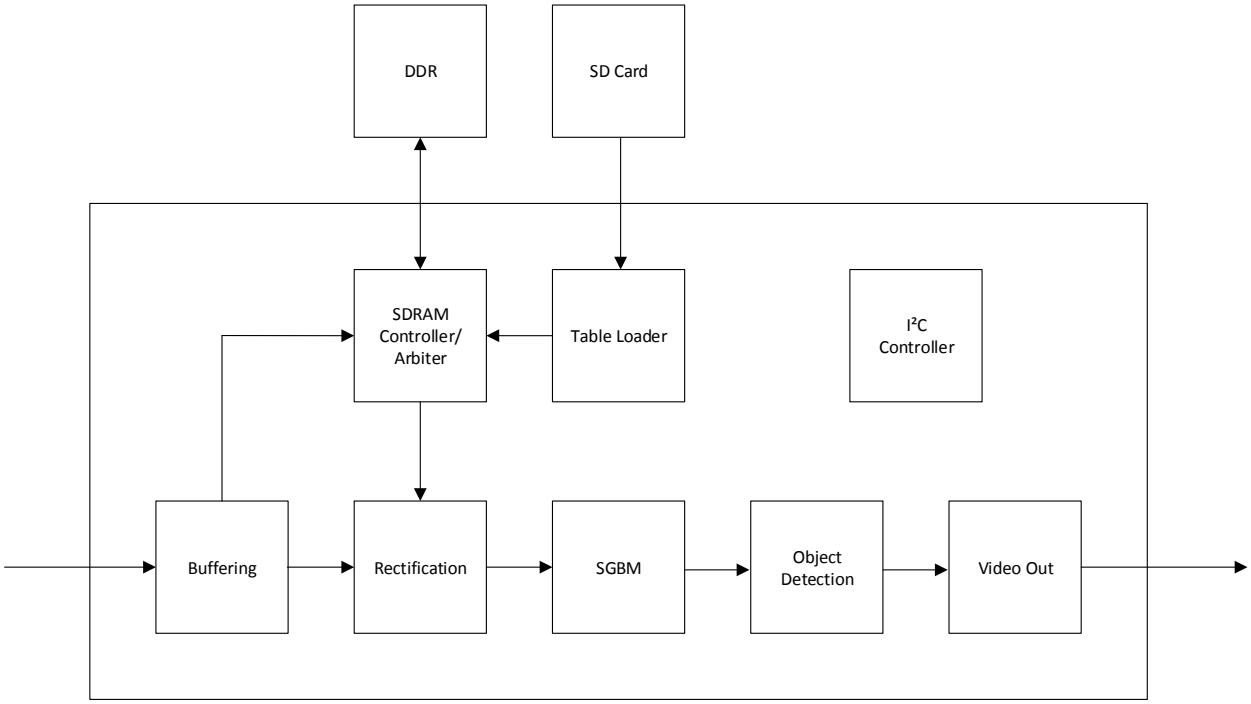

Figure 2.5. ECP5 Functional Block Diagram

2.3. Sil1136

The Sil1136 HDMI transmitter receives 36-bit RGB data, controls signals from ECP5, and converts them to HDMI format that is displayed on the HDMI monitor. This transmitter device is configured to output 640 x 480 at 60 fps through the ECP5 I²C Master interface on the ECP5 VIP.

3. Rectification Table

The Rectification Table must be created to properly rectify the images before the 3D Depth Mapping design can be used. Rectification ensures both images are on a common virtual plane. Minor differences between board and sensor require that the Rectification Table be created for each individual board. There are four steps in creating the Rectification Table:

1. Horizontally aligning the two camera sensors.
2. Capturing test pattern Images in different orientations.
3. Running images through rectification table software.
4. Loading the Rectification Table onto the SD Card.

Creating the Rectification Table requires capturing and processing images on a PC. The USB3-GbE VIP I/O Board is used to provide video capture to the PC. For more information on USB3-GbE VIP I/O, visit http://www.latticesemi.com/Products/DevelopmentBoardsAndKits/USB3GBeVIPIOBoard.

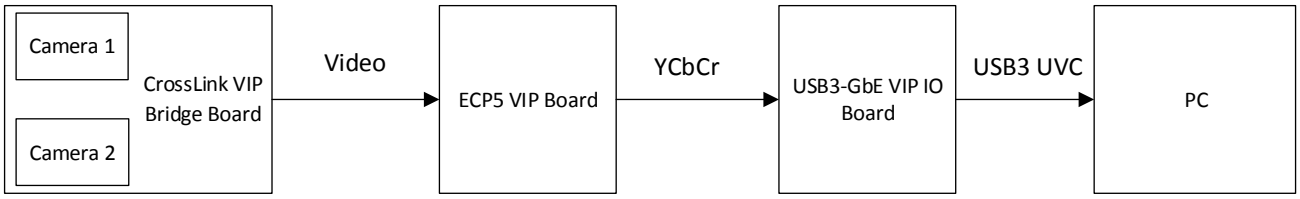

Figure 3.1. Rectification Table Design Block Diagram

The following sections describe how to build the Rectification Table. For step-by-step instructions, refer to the Rectification Procedure section.

3.1. ECP5 Design

The ECP5 FPGA Rectification Design receives the interlaced video from the CrossLink VIP Bridge Board and converts it to 1080p YCbCr to be sent to the USB3-GbE VIP I/O Board. The FX3 device on the USB3-GbE VIP I/O Board transmits the video as a UVC (USB Video Class) device to the PC. Figure 3.2 shows the block diagram of the Rectification Design.

- **Video Retiming** – Receives the 480 x 270 aggregated video from CrossLink and upscales it to 1080p parallel RGB.
- **USB Interface** – Converts the RGB to YCbCr to interface with FX3 on USB3-GbE VIP I/O Board.
- **I²C Controller** – Programs the Camera Sensors on the CrossLink VIP Bridge Board.

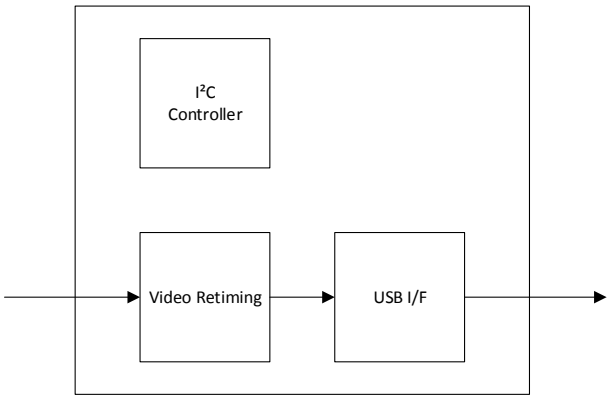

Figure 3.2. ECP5 Functional Block Diagram

3.2. Horizontally Aligning the Camera Sensors

To horizontally align the cameras, the Y-axis start address and end address of the left camera are adjusted through I²C commands. To find the correct offset, the ECP5_rectification_USB3 design is used to capture video of *offset.pdf*, which is nothing more than a single square as shown in Figure 3.3. The offset needed can be calculated using the difference in the horizontal position of the top side of left and right square. Each line of difference equates to an 8-pixel offset (see Figure 3.4). When the left square is higher than the right square, add 8 pixels per line of difference to the start address and end address. When the left square is lower than the right square, subtract 8 pixels per line of difference to the start address and end address.

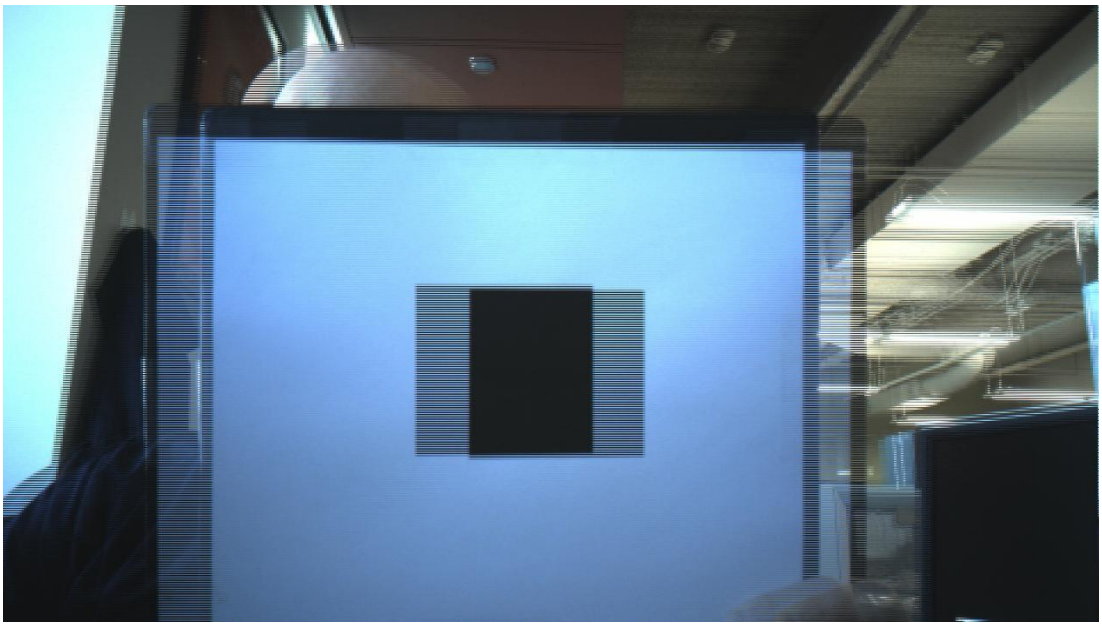

Figure 3.3. Camera Offset Capture

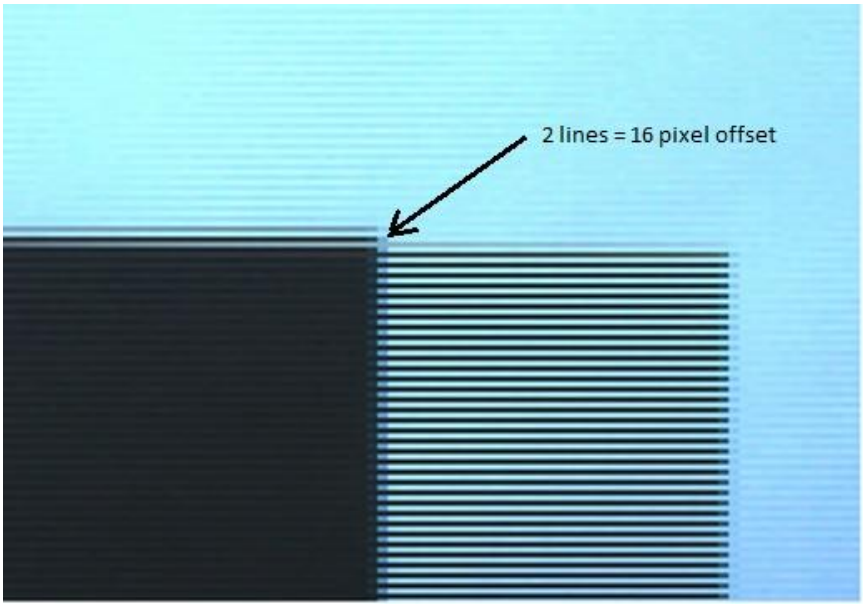

Figure 3.4. Camera Offset Calculation

3.3. Calibration Images

The Rectification Table is created using captured images of a checkerboard pattern in different orientations of the field on view. The field of view is divided into six regions, as shown in Figure 3.5. The checkerboard pattern should take up approximately 25% of the image:

- Upper Left
- Upper Center
- Upper Right
- Lower Left
- Lower Center
- Lower Right

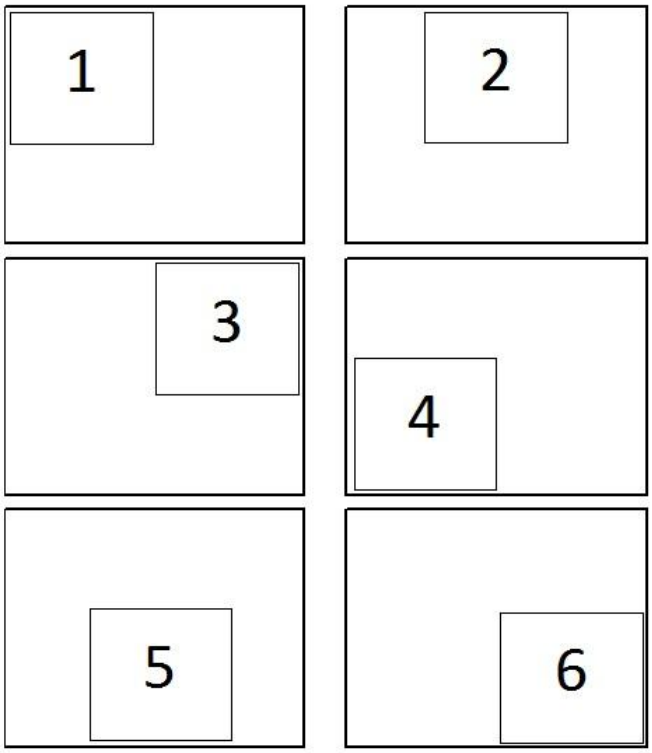

Figure 3.5. Calibration Image Regions

For each region, the checkerboard pattern should be captured in seven different orientations for a total of 42 captured images, as shown in Figure 3.6:

- Landscape
- Vertical tilt left
- Vertical tilt right
- Horizontal tilt upwards
- Horizontal tilt downwards
- Rotated clockwise
- Rotated counter-clockwise

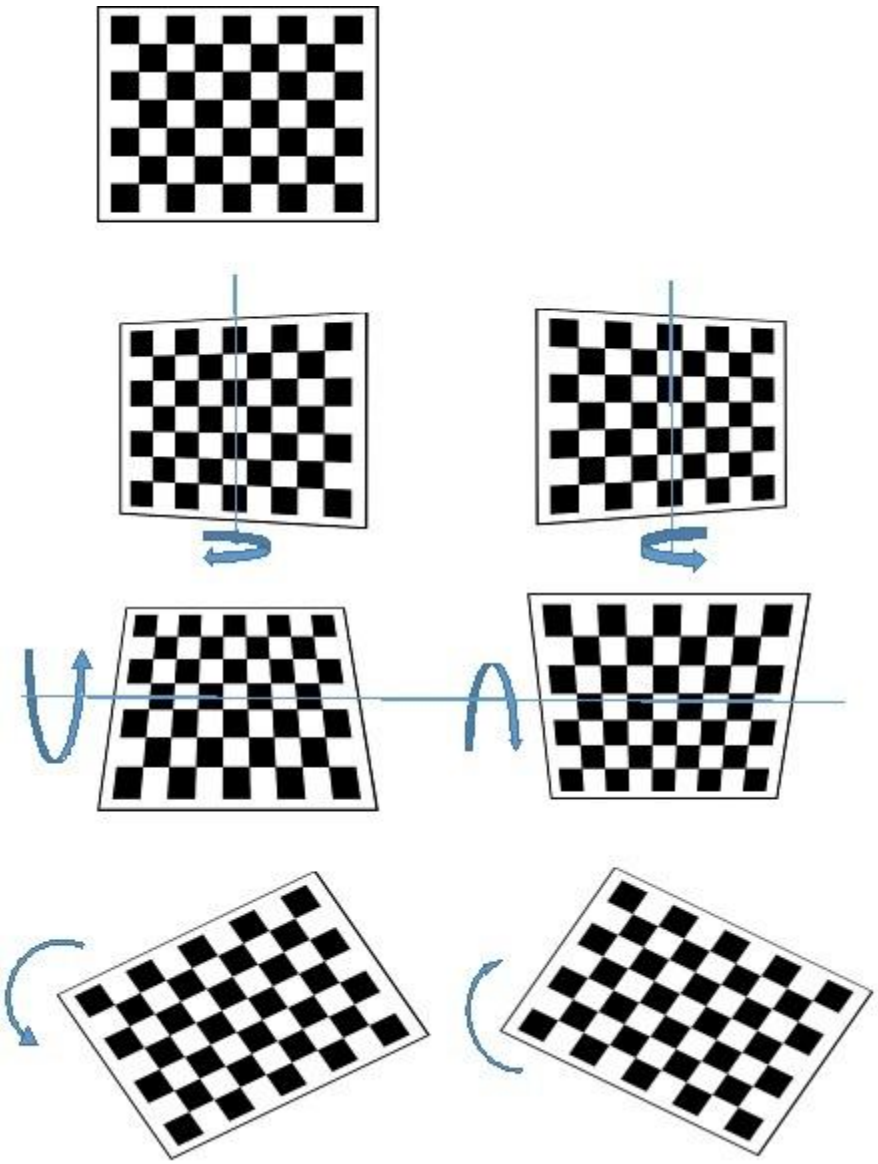

Figure 3.6. Calibration Image Orientations

3.4. Rectification Table Software

The Rectification Table software uses the OpenCV library to analyze images and create the Rectification Table. These executables are called from the command prompt and located in the <Installation Directory>\Rectification_SW directory.

- **capture_images.exe -** Captures images from the EVDK with USB3-GbE VIP I/O Board. Pressing the spacebar captures an image, which is placed into the image directory. Once all the required images are captured, press the **ESC** key to exit.
  - Usage: capture_images <cam> <image directory>
- **separate_images.exe** - Separates the captured interlaced images into left/right images
  - Usage: separate_images <image directory> <number of images>
- **calibrate.exe** - Using the captured images, determines the left or right camera's intrinsic and extrinsic parameters. Calls the OpenCV library function calibrateCamera().
  - Usage: calibrate <image directory> <number of images> <left/right>

- **calibrate_stereo.exe** - Using the intrinsic and extrinsic parameters of both cameras, computes the rotation matrices for each camera needed to place on same virtual plane. Calls the OpenCV library function stereoCalibrate() and stereoRectify().
  - Usage: calibrate_stereo <image directory> <number of images>
- **undistort_rectify.exe** - Computes the rectification map and outputs rectified images. Calls the OpenCV library functions initUndistortRectifyMap() and remap().
  - Usage: undistort_rectifiy <image directory> <left image> <right image>
- **table_map.exe** - Builds the SD Card image with the Rectification Table and left camera y-start address.
  - Usage: table_map <image directory> <offset>

4. Rectification Procedure

The following equipment is required for the rectification procedure:

- LF-EVDK1-EVN Rev C or later demonstration kit
- USB3-Gb3 VIP I/O Board
- DC power adapter (12 V)
- Windows Laptop/PC with USB 3.0 port
- USB 2.0 Type A to Mini-B cable
- USB 3.0 Type A to Micro-B cable
- Lattice Diamond® Programmer version 3.10 or higher
- Raw disk image writer (such as win32diskimager)

Note: The software used in the rectification procedure requires Microsoft Visual C++ 2017 Redistributable. If you receive a system error while running the software, download and install from Microsoft Support.

To create the rectification table:

1. Print paper **Offset.pdf** and **Checkerboard.pdf** on 8.5 x 11 (select **Shrink oversized pages**). Attach printouts to rigid backing (that is by binder or cardboard) to prevent bending. For the checkboard pattern, the squares should be 25.2 mm x 25.2 mm in size.
2. Ensure that power is disconnected from the EVDK.
3. Replace the HDMI VIP Output Bridge Board with the USB3-GbE VIP I/O Board. The USB3-GbE VIP I/O Board comes preloaded with the FX3 firmware loaded into the I²C boot ROM required for this demo. If this is changed, update the firmware by referring to Appendix B of the USB3-GbE VIP IO Board Demo User Guide (FPGA-UG-02054) for instructions on how to load the boot image.

Figure 4.1. EVDK with USB3-GbE VIP I/O Board Installed

4. Connect power to the EVDK.
5. Program the SPI Flash on the CrossLink VIP Input Bridge Board with **<Installation Directory>\Crosslink_3D_dual_cams\bitstream\crosslink_3D_dual_cameras.bit**. Refer to Appendix B of the Lattice Embedded Vision Development Kit User Guide (FPGA-UG-02015) for detailed instructions.
6. Program the SPI Flash on the ECP5 VIP Processor Board with **<Installation Directory>\ECP5_Rectification_USB3\bitstream\depth_map_rectification.bit**. Refer to Appendix B of the Lattice Embedded Vision Development Kit User Guide (FPGA-UG-02015) for detailed instructions.
7. Connect the USB3 cable from the USB 3.0 Micro B connector, J4, on the USB3-GbE VIP I/O Board to USB 3.0 port of your PC.
8. Cycle power on board.
9. Press **System Reset** (SW3 on the CrossLink VIP Input Bridge Board).
10. Calculate Offset.
    a. Open Command Prompt, and change directory to **<Installation Directory>\Rectification_SW**.
       - **>>mkdir .\offset_image**
    b. Run capture_images.exe.
       - **>>capture_images <camera> .\offset_image**

       Note: <camera> is system-dependent. Recommended starting value is *0*.
    c. Press **Spacebar** to capture the image (make sure video capture window is active). Position printout of **Offset.pdf** at the center.
    d. When complete, press **ESC** to close video capture window.
    e. Using the captured image, determine the needed Y start address and end address of the left camera. The default setting is 0x01E0 for start address and 0x0A4F for end address.
       - If left square is higher, add 8 pixels per line from start address and end address.
       - If left square is lower, subtract 8 pixels per line from start address and end address.

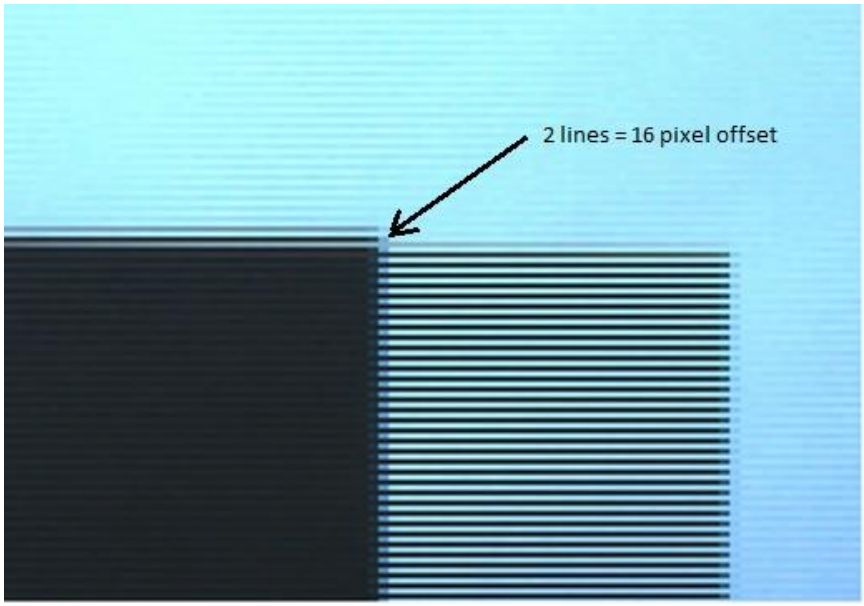

Figure 4.2. Offset Calculation

11. Open <Installation Directory>\ECP5_Rectification_USB3\depth_map_rectification.ldf in Diamond Software.
    a. Open **source /i2c/ic2_ctrl_s.v**.
    b. Edit I²C states 16 and 17 with Y start address and 20 and 21 with Y end address.
    c. Save edits and recompile design.
    d. Repeat steps 6 to 10 to verify proper offset.

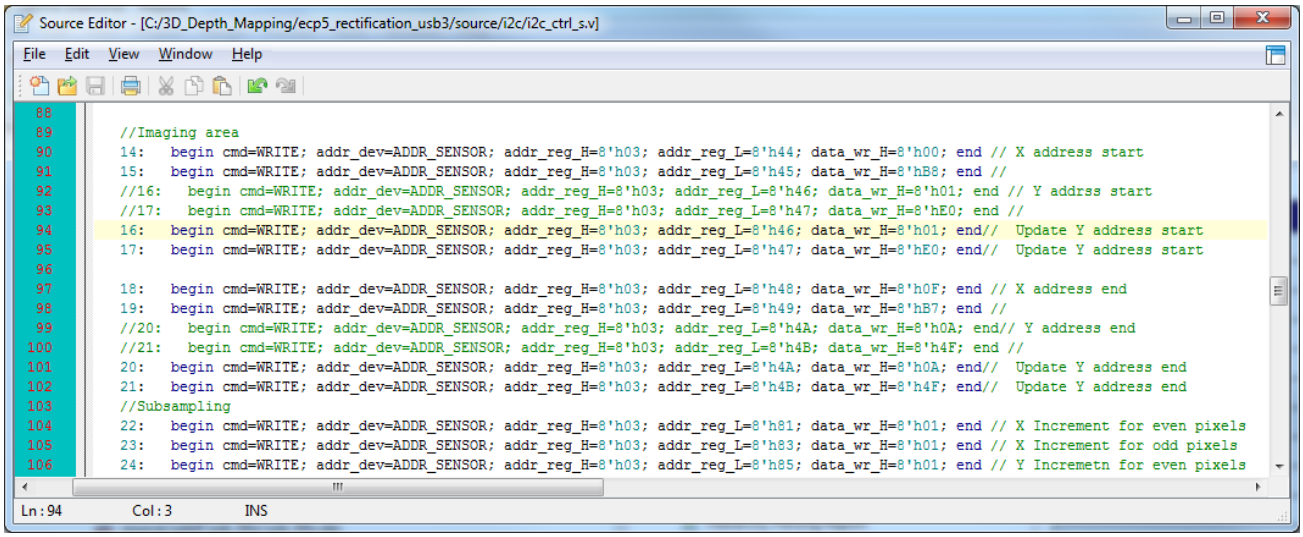

Figure 4.3. Update Y Offset in i2c_ctrls.v

12. Program the ECP5 with updated <rectification bitstream> over JTAG. Refer to Appendix B for detailed instructions.
13. Press **System Reset** (SW3 on CrossLink VIP Input Bridge Board).
14. Capture calibration images.
    a. Open Command Prompt.
    b. Change directory to **<Installation Directory>\Rectification_SW**.
    c. Create an image directory. For this example, *rect_images*.
       - **>>mkdir .\rect_images**
    d. Run **capture_images.exe**.
       - **>>capture_images <camera> .\rect_images**

       Note: <camera> is system dependent. Recommended starting value is *0*.
    e. Press **Spacebar** to capture the images (make sure video capture window is active). Position the Checkerboard pattern in each of the six sections of the display in each of the seven orientation, for a minimum of 42 images. For more information, refer to Calibration Images.
    f. Once completed, press **ESC** to close video capture window.

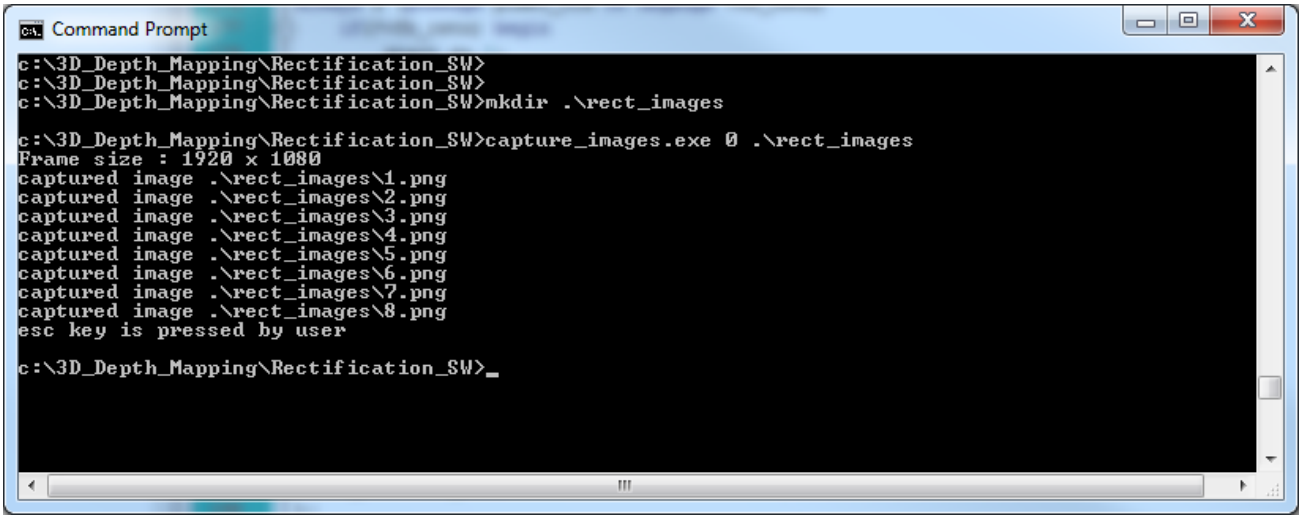

Figure 4.4. Captured Images

15. Run the Rectification Calibration software
    a. Separate images into left and right.
       - **>separate_images.exe .\rect_images [# of images]**
    b. Calibrate left camera and right camera.
       - **>calibrate .\rect_images [# of images] left**
       - **>calibrate .\rect_images [# of images] right**
    c. Calibrate stereo image.
       - **>calibrate_stereo .\rect_images [# of images]**
    d. Create Rectification Table.
       - **>undistort_rectify.exe .\rect_images left1 right1**
    e. Build SD Card image, adding Y stat address calculated in step 10.
       - **>table_map.exe .\rect_images [-offset]**

       Note: If you receive an *ERROR!: out-of-bound* message, your rectified image does not fit in the required boundaries of the SGBM IP. Check the following:
       - Horizontal offset between the cameras is correct.
       - Captured images clearly show the checkerboard pattern.
       - Checkerboard pattern adhered to a rigid backing.

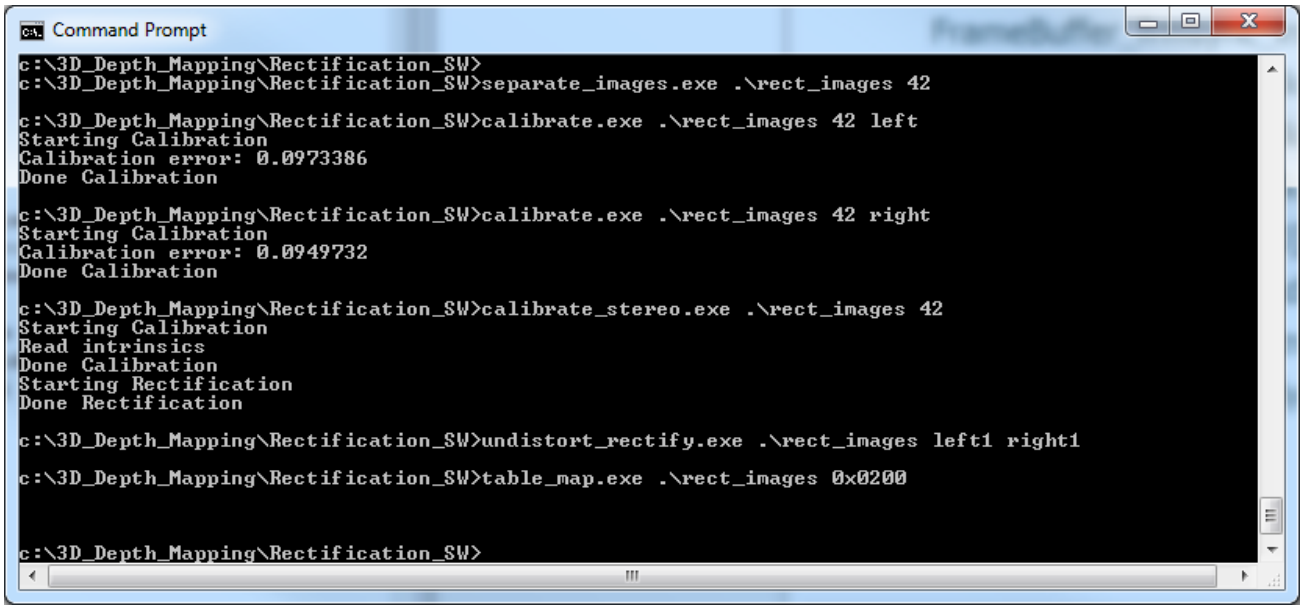

Figure 4.5. Rectification Table Software

16. Load **<Installation Directory>\Rectification_SW\rect_images\rect_table.bin** on to an SD Card using a raw disk image writer (such as win32diskimager). Do not format the SD Card.

5. Demo Procedure

The following equipment is required for the demo:

- LF-EVDK1-EVN Rev C or later demo kit
- MicroSD Card Adapter
- HDMI monitor
- HDMI cable
- DC power adapter (12 V)
- Windows Laptop/PC with USB 3.0 port
- USB 2.0 Type A to Mini-B cable
- Lattice Diamond Programmer version 3.10 or higher

To run the demonstration:

1. Ensure that power is disconnected from the EVDK.
2. Replace the USB3-GbE VIP I/O Board with the HDMI VIP Output Board.
3. Connect the SD Card Adapter Board to Jumper J14 of the ECP5 VIP Processor Board. Ensure that the SD Card is inserted and that it contains the Rectification Table built in the Rectification Procedure section.

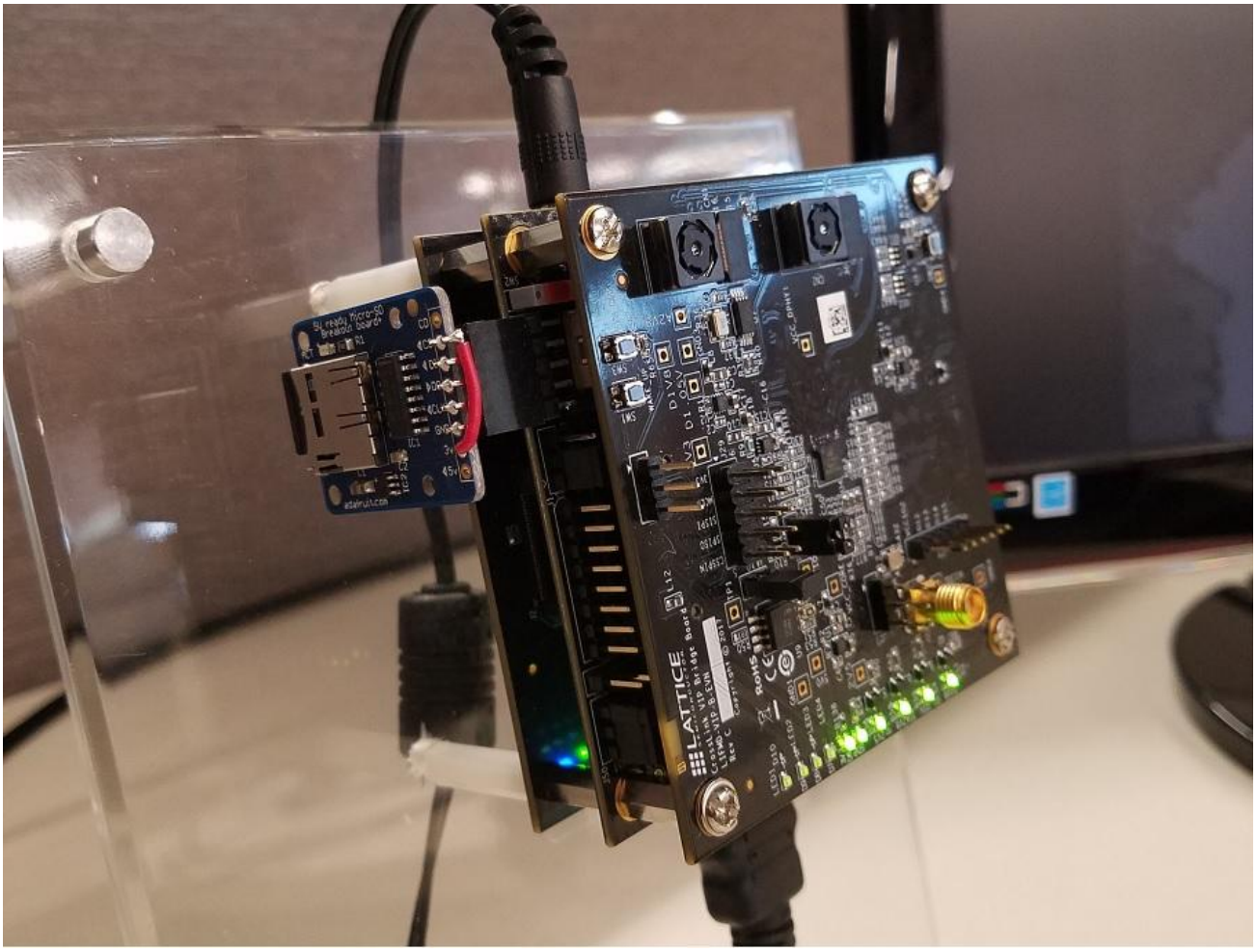

Figure 5.1. SD Card Adapter

4. Connect the ECP5 VIP Processor Board to the wall socket using the 12 V power adapter.
5. Power up the demo kit by turning on SW2 on the ECP5 VIP Processor Board.
6. Ensure that the SPI Flash on the CrossLink VIP Input Bridge Board is programmed with **<Installation Directory>\Crosslink_3D_dual_cams\bitstream\crosslink_3D_dual_cameras.bit**. This should be performed in the Rectification Procedure section. Refer to Appendix B of the Lattice Embedded Vision Development Kit User Guide (FPGA-UG-02015) for detailed instructions.
7. Program the SPI Flash on the ECP5 VIP Processor Board with **<Installation Directory>\ECP5_3D_DepthMapping\bitstream\ECP5_DepthMap.bit**. Refer to Appendix B of the Lattice Embedded Vision Development Kit User Guide (FPGA-UG-02015) for detailed instructions.
8. Power cycle the EVDK.
9. Connect the HDMI cable from CN1 of HDMI VIP output board to the HDMI display/monitor. The monitor displays the dual camera merged image as shown in Figure 5.2.

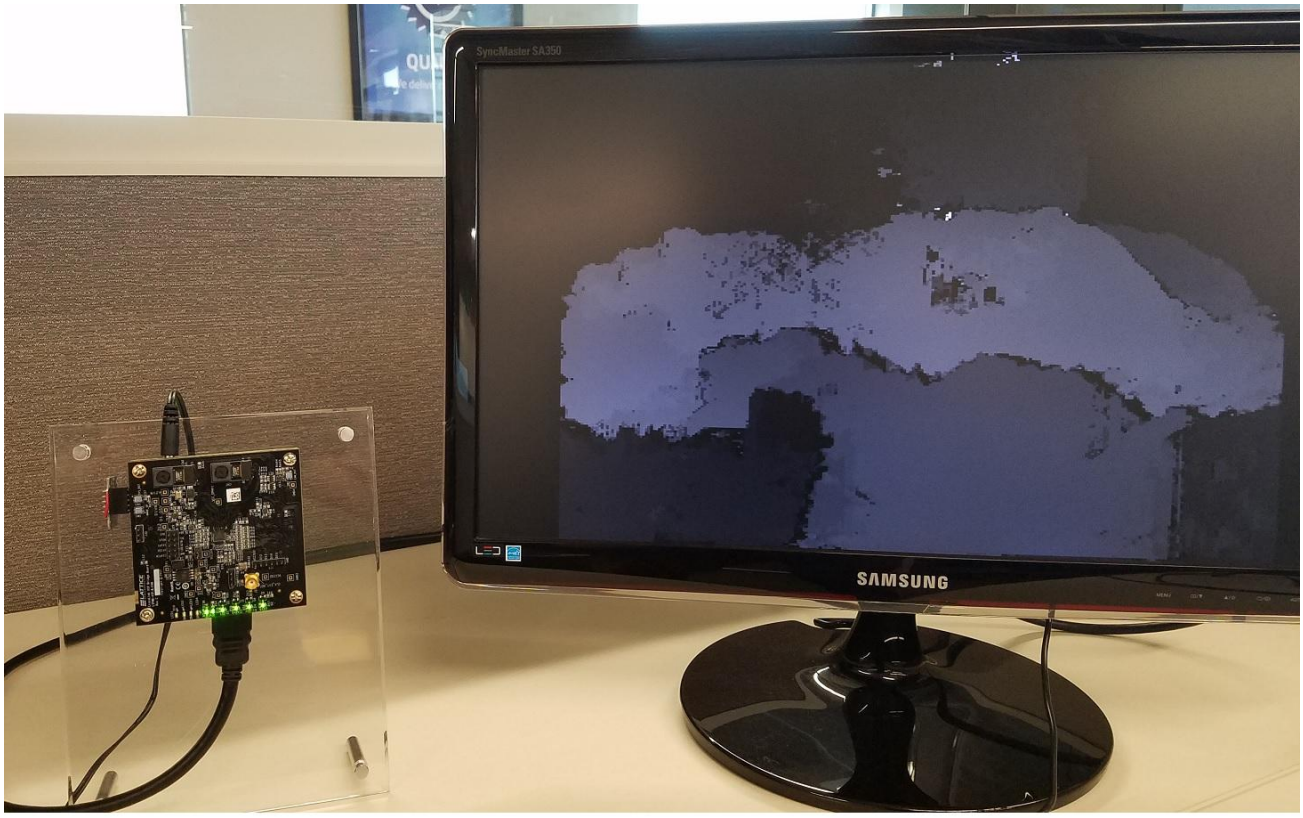

Figure 5.2. 3D Depth Mapping Demonstration

6. Jumper Settings

Table 6.1. CrossLink VIP Input Bridge Board

Jumper	Description	Default
J2	CrossLink SPI Chip Select	Short
J4	SPI Flash Chip Select	Short
J30	CRESETB selection	Open
—	—	All other headers should be kept open.

Table 6.2. ECP5 VIP Processor Board

Jumper	Description	Default
J3	ECP5 Configuration Selection	Connect 1 and 2 and connect 5 and 6 (Master SPI)
J5	Bank 3 Voltage Selection	Connect 1 and 2 (3.3 V)
J6	Bank 1 Voltage Selection	Connect 1 and 2 (3.3 V)
J7	Bank 0 Voltage Selection	Connect 2 and 3 (3.3 V)
J9	Bank 8 Voltage Selection	Connect 1 and 2 (3.3 V)
J50	JTAG Daisy Chain	Connect 1 and 2 and connect 3 and 5 (ECP5 only)
J51	Bank 4 Voltage Selection	Connect 1 and 2 (3.3 V)
J52	FTDI TCK Pull Up/Down	Connect 2 and 3 (JTAG)
J53	FTDI Reset	Connect 1 and 2 (Pulled High)
J55	Bank 2 Voltage	Connect 2 and 3 (3.3 V)
—	—	All other headers should be kept open.

Table 6.3. USB3-GbE VIP I/O Board

Jumper	Description	Default
P2	VCCIO Selection	Connect 2 and 3 (3.3 V)
P3	FX3 Boot Mode	Open
P4	FX3 Boot Mode	Connect 1 and 2
P5	FX3 Boot Mode	Open

7. Demo Package Directory Structure

The key files and directories are listed below:

- SGBM_DepthMapping (Main directory)
  - Crosslink_Dual_Camera (CrossLink design directory)
    - bitstream
      - Crosslink_Dual_Camera.bit (CrossLink bitstream)
    - source (CrossLink source files)
    - Crosslink_Dual_Camera.ldf (CrossLink Diamond Project file)
    - Crosslink_Dual_Camera.lpf (CrossLink Project Settings file)
    - Crosslink_Dual_Camera1.sty (CrossLink Project Strategy file)
  - ECP5_SGBM_DepthMapping (ECP5 design directory)
    - bitstream
      - ECP5_SGBM_DepthMapping.bit (ECP5 bitstream)
    - source (ECP5 source files)
    - ECP5_SGBM_DepthMapping.ldf (ECP5 Diamond Project file)
    - ECP5_SGBM_DepthMapping.lpf (ECP5 Project Settings file)
    - ECP5_SGBM_DepthMapping1.sty (ECP5 Project Strategy file)
    - sgbm.ngo (Encrypted SGBM IP file)
  - ECP5_Rectification_USB3 (ECP5 design directory)
    - bitstream
      - ECP5_Rectification_USB3.bit (ECP5 bitstream)
    - source (ECP5 source files)
    - ECP5_Rectification_USB3.ldf (ECP5 Diamond Project file)
    - ECP5_Rectification_USB3.lpf (ECP5 Project Settings file)
    - ECP5_Rectification_USB31.sty (ECP5 Project Strategy file)
  - Rectification_SW (Rectification Software directory)
    - Visual_Studio_Projects (SW Project files)
    - test_images (Set of test images)
    - calibrate.exe (Executable file)
    - calibrate_stereo.exe (Executable file)
    - capture_images.exe (Executable file)
    - checkerboard.pdf (Test file)
    - offset.pdf (Test file)
    - opencv_world341.dll (OpenCV library)
    - separate_images.exe (Executable file)
    - table_map_offset.exe (Executable file)
    - undistort_rectify.exe (Executable file)

8. Pinout Information

8.1. ECP5 3D Depth Mapping Pinout

Table 8.1 lists the ECP5 pinouts used in the demo.

Table 8.1. ECP5 Pinouts

Port Name	Pin	Bank	Buffer Type	Site	Properties
clk_100	C5	7	LVDS_IN	PL11A	Clamp: On
reset_n	AH1	8	LVCMOS33_IN	PB4B	Pull: Down, Clamp: On, Hysteresis: On
CrossLink Interface					
CSI2_sens_clk	P27	2	LVCMOS33_IN	PR44C	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _fv	K27	2	LVCMOS33_IN	PR38A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _lv	K26	2	LVCMOS33_IN	PR38B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[0]	A13	0	LVCMOS33_IN	PT42B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[1]	A8	0	LVCMOS33_IN	PT20B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[2]	F9	0	LVCMOS33_IN	PT22A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[3]	D9	0	LVCMOS33_IN	PT22B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[4]	C9	0	LVCMOS33_IN	PT24A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[5]	A9	0	LVCMOS33_IN	PT24B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[6]	C10	0	LVCMOS33_IN	PT29B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[7]	B10	0	LVCMOS33_IN	PT31A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[8]	A10	0	LVCMOS33_IN	PT31B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens _data[9]	E11	0	LVCMOS33_IN	PT33B	Pull: Down, Clamp: On, Hysteresis: On
reset_crosslink	D13	0	LVCMOS33_OUT	PT40B	Drive:8 mA, Clamp: On, Slew: Slow
Camera Sensor Interface					
scl	D15	0	LVCMOS33_OUT	PT51B	Drive:8 mA, Clamp: On, Slew: Slow
scl2	A14	0	LVCMOS33_OUT	PT49B	Drive:8 mA, Clamp: On, Slew: Slow
sda	F15	0	LVCMOS33_OUT	PT51A	Drive:8 mA, Clamp: On, Slew: Slow
sda2	B14	0	LVCMOS33_OUT	PT49A	Drive:8 mA, Clamp: On, Slew: Slow
reset_sensor	B4	0	LVCMOS33_OUT	PT4B	Drive:8 mA, Clamp: On, Slew: Slow
SiI1136 Interface					
HDMI_scl	AG1	8	LVCMOS33_OUT	PB4A	Drive:8 mA, Clamp: On, Slew: Slow
HDMI_sda	AJ1	8	LVCMOS33_OUT	PB6A	Drive:8 mA, Clamp: On, Slew: Slow
reset_hdmi	Y30	3	LVCMOS33_OUT	—	—
pixclk_out	E25	1	LVCMOS33_OUT	PT110A	Drive:8 mA, Clamp: On, Slew: Slow
data_enable	C25	1	LVCMOS33_OUT	PT107A	Drive:8 mA, Clamp: On, Slew: Slow
hsync	D25	1	LVCMOS33_OUT	PT107B	Drive:8 mA, Clamp: On, Slew: Slow
vsync	A25	1	LVCMOS33_OUT	PT105A	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[0]	T31	3	LVCMOS33_OUT	PR65B	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[1]	R32	3	LVCMOS33_OUT	PR65A	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[2]	Y32	3	LVCMOS33_OUT	PR86B	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[3]	W31	3	LVCMOS33_OUT	PR86A	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[4]	T29	3	LVCMOS33_OUT	PR53C	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[5]	U28	3	LVCMOS33_OUT	PR53D	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[6]	V27	3	LVCMOS33_OUT	PR56C	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[7]	V26	3	LVCMOS33_OUT	PR56D	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[8]	AC31	3	LVCMOS33_OUT	PR89C	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[9]	AB32	3	LVCMOS33_OUT	PR92A	Drive:8 mA, Clamp: On, Slew: Slow

Port Name	Pin	Bank	Buffer Type	Site	Properties
pix_blue[10]	AC32	3	LVCMOS33_OUT	PR92B	Drive:8 mA, Clamp: On, Slew: Slow
pix_blue[11]	AD32	3	LVCMOS33_OUT	PR92C	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[0]	AD26	3	LVCMOS33_OUT	PR77D	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[1]	T26	3	LVCMOS33_OUT	PR47D	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[2]	R26	3	LVCMOS33_OUT	PR47C	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[3]	A24	1	LVCMOS33_OUT	PT101A	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[4]	T32	3	LVCMOS33_OUT	PR68A	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[5]	AC30	3	LVCMOS33_OUT	PR89A	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[6]	AB31	3	LVCMOS33_OUT	PR89B	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[7]	V32	3	LVCMOS33_OUT	PR68C	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[8]	W32	3	LVCMOS33_OUT	PR68D	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[9]	Y26	3	LVCMOS33_OUT	PR71A	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[10]	W30	3	LVCMOS33_OUT	PR65C	Drive:8 mA, Clamp: On, Slew: Slow
pix_green[11]	T30	3	LVCMOS33_OUT	PR59D	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[0]	AE27	3	LVCMOS33_OUT	PR80B	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[1]	AD27	3	LVCMOS33_OUT	PR80A	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[2]	AB29	3	LVCMOS33_OUT	PR83B	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[3]	AB30	3	LVCMOS33_OUT	PR83A	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[4]	AB28	3	LVCMOS33_OUT	PR77A	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[5]	AB27	3	LVCMOS33_OUT	PR77B	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[6]	AC26	3	LVCMOS33_OUT	PR77C	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[7]	Y27	3	LVCMOS33_OUT	PR71B	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[8]	D24	1	LVCMOS33_OUT	PT103A	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[9]	W28	3	LVCMOS33_OUT	PR71D	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[10]	F25	1	LVCMOS33_OUT	PT110B	Drive:8 mA, Clamp: On, Slew: Slow
pix_red[11]	F17	1	LVCMOS33_OUT	PT69B	Drive:8 mA, Clamp: On, Slew: Slow
DDR3 Interface					
em_ddr_addr[0]	W4	6	SSTL15_I_OUT	PL74B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[1]	Y5	6	SSTL15_I_OUT	PL71C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[2]	AB6	6	SSTL15_I_OUT	PL77B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[3]	P3	6	SSTL15_I_OUT	PL59B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[4]	AB5	6	SSTL15_I_OUT	PL77A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[5]	W5	6	SSTL15_I_OUT	PL71D	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[6]	AC6	6	SSTL15_I_OUT	PL74D	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[7]	Y7	6	SSTL15_I_OUT	PL71A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[8]	AC7	6	SSTL15_I_OUT	PL77C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[9]	AD6	6	SSTL15_I_OUT	PL80A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[10]	W1	6	SSTL15_I_OUT	PL68D	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[11]	Y6	6	SSTL15_I_OUT	PL71B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[12]	U1	6	SSTL15_I_OUT	PL68B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_addr[13]	AD7	6	SSTL15_I_OUT	PL77D	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_ba[0]	U3	6	SSTL15_I_OUT	PL62D	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_ba[1]	Y4	6	SSTL15_I_OUT	PL74A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_ba[2]	W3	6	SSTL15_I_OUT	PL65C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_cas_n	T1	6	SSTL15_I_OUT	PL68A	DRIVE:8 mA CLAMP:ON SLEW:FAST

Port Name	Pin	Bank	Buffer Type	Site	Properties
em_ddr_cke[0]	T2	6	SSTL15_I_OUT	PL65B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_cke[1]	R1	6	SSTL15_I_OUT	PL65A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_clk[0]	R3	6	SSTL15D_I_OUT	PL59C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_clk[1]	J4	7	SSTL15D_I_OUT	PL32C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_cs_n	P2	6	SSTL15_I_OUT	PL59A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_data[0]	AC5	6	SSTL15_I_BIDI	PL83C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[1]	AC2	6	SSTL15_I_BIDI	PL89C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[2]	AB4	6	SSTL15_I_BIDI	PL83B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[3]	AE3	6	SSTL15_I_BIDI	PL86D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[4]	W2	6	SSTL15_I_BIDI	PL86A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[5]	AD4	6	SSTL15_I_BIDI	PL83D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[6]	Y1	6	SSTL15_I_BIDI	PL86B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[7]	AB1	6	SSTL15_I_BIDI	PL92A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[8]	V6	6	SSTL15_I_BIDI	PL56C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[9]	P4	6	SSTL15_I_BIDI	PL47B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[10]	V7	6	SSTL15_I_BIDI	PL56D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[11]	T7	6	SSTL15_I_BIDI	PL47D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[12]	U6	6	SSTL15_I_BIDI	PL50C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[13]	T4	6	SSTL15_I_BIDI	PL53C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[14]	U7	6	SSTL15_I_BIDI	PL50D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[15]	U4	6	SSTL15_I_BIDI	PL56A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[16]	H2	7	SSTL15_I_BIDI	PL26A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[17]	K1	7	SSTL15_I_BIDI	PL29C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[18]	F1	7	SSTL15_I_BIDI	PL23C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[19]	L2	7	SSTL15_I_BIDI	PL32A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[20]	E1	7	SSTL15_I_BIDI	PL23B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[21]	K3	7	SSTL15_I_BIDI	PL26D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[22]	H3	7	SSTL15_I_BIDI	PL26B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[23]	H1	7	SSTL15_I_BIDI	PL23D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[24]	N7	7	SSTL15_I_BIDI	PL44B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[25]	J6	7	SSTL15_I_BIDI	PL35D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[26]	L4	7	SSTL15_I_BIDI	PL38D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[27]	K7	7	SSTL15_I_BIDI	PL38B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[28]	P7	7	SSTL15_I_BIDI	PL44D	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[29]	L7	7	SSTL15_I_BIDI	PL41C	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[30]	K6	7	SSTL15_I_BIDI	PL38A	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_data[31]	H5	7	SSTL15_I_BIDI	PL35B	DRIVE:8 mA CLAMP:ON SLEW:FAST VREF1_LOAD
em_ddr_dm[0]	AB3	6	SSTL15_I_OUT	PL83A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_dm[1]	R6	6	SSTL15_I_OUT	PL50A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_dm[2]	F2	7	SSTL15_I_OUT	PL23A	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_dm[3]	H6	7	SSTL15_I_OUT	PL35A	DRIVE:8 mA CLAMP:ON SLEW:FAST

Port Name	Pin	Bank	Buffer Type	Site	Properties
em_ddr_dqs[0]	AC3	6	SSTL15D_I_BIDI	PL89A	DRIVE:8 mA CLAMP:ON SLEW:FAST DIFFRESISTOR:100
em_ddr_dqs[1]	R4	6	SSTL15D_I_BIDI	PL53A	DRIVE:8 mA CLAMP:ON SLEW:FAST DIFFRESISTOR:100
em_ddr_dqs[2]	K2	7	SSTL15D_I_BIDI	PL29A	DRIVE:8 mA CLAMP:ON SLEW:FAST DIFFRESISTOR:100
em_ddr_dqs[3]	N3	7	SSTL15D_I_BIDI	PL41A	DRIVE:8 mA CLAMP:ON SLEW:FAST DIFFRESISTOR:100
em_ddr_odt	V1	6	SSTL15D_I_OUT	PL68C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_ras_n	C2	7	SSTL15D_I_OUT	PL17B	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_reset_n	C4	7	SSTL15D_I_OUT	PL11C	DRIVE:8 mA CLAMP:ON SLEW:FAST
em_ddr_we_n	P1	6	SSTL15D_I_OUT	PL62B	DRIVE:8 mA CLAMP:ON SLEW:FAST
Status LEDs					
led[0]	AG30	4	LVCMOS33_OUT	PB114B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[1]	AK29	4	LVCMOS33_OUT	PB116A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[2]	AK30	4	LVCMOS33_OUT	PB116B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[3]	AH32	4	LVCMOS33_OUT	PB119A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[4]	AG32	4	LVCMOS33_OUT	PB119B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[5]	AJ29	4	LVCMOS33_OUT	PB121A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[6]	AM28	4	LVCMOS33_OUT	PB96A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
led[7]	AM29	4	LVCMOS33_OUT	PB96B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
SD Card Interface					
spi_clk	AL32	4	LVCMOS33_OUT	PB107B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
spi_css	AK31	4	LVCMOS33_OUT	PB105A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
spi_miso	AM31	4	LVCMOS33_IN	PB107A	PULL:UP CLAMP:ON HYSTERESIS:ON
spi_mosi	AJ31	4	LVCMOS33_OUT	PB105B	DRIVE:8 mA CLAMP:ON SLEW:SLOW

8.2. CrossLink

Table 8.2 lists the CrossLink pinouts used for the demo.

Table 8.2. CrossLink Pinouts

Port Name	Pin	Bank	Buffer Type	Site	Properties
reset_n_i	J4	1	LVCMOS33_IN	PB38C	Pull: Up, Clamp: On, Hysteresis: On
Camera Sensor Interface					
clk_p_i	A1	61	DPHY_BIDI	DPHY1_CKP	—
clk_n_i	A2	61	DPHY_BIDI	DPHY1_CKN	—
d0_p_i	B1	61	DPHY_BIDI	DPHY1_DP0	—
d0_n_i	B2	61	DPHY_BIDI	DPHY1_DN0	—
d1_p_i	A3	61	DPHY_BIDI	DPHY1_DP1	—
d1_n_i	B3	61	DPHY_BIDI	DPHY1_DN1	—
d2_p_i	C1	61	DPHY_BIDI	DPHY1_DP2	—
d2_n_i	C2	61	DPHY_BIDI	DPHY1_DN2	—
d3_p_i	A4	61	DPHY_BIDI	DPHY1_DP3	—
d3_n_i	B4	61	DPHY_BIDI	DPHY1_DN3	—
clk_p_i_s	A8	60	DPHY_BIDI	DPHY0_CKP	—
clk_n_i_s	A9	60	DPHY_BIDI	DPHY0_CKN	—
d0_p_i_s	B7	60	DPHY_BIDI	DPHY0_DP0	—
d0_n_i_s	A7	60	DPHY_BIDI	DPHY0_DN0	—
d1_p_i_s	B8	60	DPHY_BIDI	DPHY0_DP1	—
d1_n_i_s	B9	60	DPHY_BIDI	DPHY0_DN1	—
d2_p_i_s	B6	60	DPHY_BIDI	DPHY0_DP2	—
d2_n_i_s	A6	60	DPHY_BIDI	DPHY0_DN2	—
d3_p_i_s	C8	60	DPHY_BIDI	DPHY0_DP3	—
d3_n_i_s	C9	60	DPHY_BIDI	DPHY0_DN3	—
ECP5 Interface					
pixel_clk	J6	1	LVCMOS33_OUT	PB29C	Drive: 6 mA, Clamp: On
fv	J3	1	LVCMOS33_OUT	PB43C	Drive: 6 mA, Clamp: On
lv	H3	1	LVCMOS33_OUT	PB43D	Drive: 6 mA, Clamp: On
pixdata[0]	F9	2	LVCMOS33_OUT	PB2A	Drive: 6 mA, Clamp: On
pixdata[1]	F8	2	LVCMOS33_OUT	PB2B	Drive: 6 mA, Clamp: On
pixdata[2]	G9	2	LVCMOS33_OUT	PB2C	Drive: 6 mA, Clamp: On
pixdata[3]	G8	2	LVCMOS33_OUT	PB2D	Drive: 6 mA, Clamp: On
pixdata[4]	E9	2	LVCMOS33_OUT	PB6A	Drive: 6 mA, Clamp: On
pixdata[5]	E8	2	LVCMOS33_OUT	PB6B	Drive: 6 mA, Clamp: On
pixdata[6]	H9	2	LVCMOS33_OUT	PB6C	Drive: 6 mA, Clamp: On
pixdata[7]	H8	2	LVCMOS33_OUT	PB6D	Drive: 6 mA, Clamp: On
pixdata[8]	F7	2	LVCMOS33_OUT	PB12A	Drive: 6 mA, Clamp: On
pixdata[9]	E7	2	LVCMOS33_OUT	PB12B	Drive: 6 mA, Clamp: On

8.3. ECP5 Rectification Pinout

Table 8.3 lists the ECP5 pinouts used for the rectification table design.

Table 8.3. CrossLink Pinouts

Port Name	Pin	Bank	Buffer Type	Site	Properties
clk_i	E17	1	LVCMOS33_IN		
reset_n	AH1	8	LVCMOS33_IN	PB4B	Pull: Down, Clamp: On, Hysteresis: On
CrossLink Interface					
CSI2_sens_clk	P27	2	LVCMOS33_IN	PR44C	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_fv	K27	2	LVCMOS33_IN	PR38A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_lv	K26	2	LVCMOS33_IN	PR38B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[0]	A13	0	LVCMOS33_IN	PT42B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[1]	A8	0	LVCMOS33_IN	PT20B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[2]	F9	0	LVCMOS33_IN	PT22A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[3]	D9	0	LVCMOS33_IN	PT22B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[4]	C9	0	LVCMOS33_IN	PT24A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[5]	A9	0	LVCMOS33_IN	PT24B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[6]	C10	0	LVCMOS33_IN	PT29B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[7]	B10	0	LVCMOS33_IN	PT31A	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[8]	A10	0	LVCMOS33_IN	PT31B	Pull: Down, Clamp: On, Hysteresis: On
CSI2_sens_data[9]	E11	0	LVCMOS33_IN	PT33B	Pull: Down, Clamp: On, Hysteresis: On
reset_crosslink	D13	0	LVCMOS33_OUT	PT40B	Drive:8 mA, Clamp: On, Slew: Slow
Camera Sensor Interface					
scl	D15	0	LVCMOS33_OUT	PT51B	Drive:8 mA, Clamp: On, Slew: Slow
scl2	A14	0	LVCMOS33_OUT	PT49B	Drive:8 mA, Clamp: On, Slew: Slow
sda	F15	0	LVCMOS33_OUT	PT51A	Drive:8 mA, Clamp: On, Slew: Slow
sda2	B14	0	LVCMOS33_OUT	PT49A	Drive:8 mA, Clamp: On, Slew: Slow
reset_sensor	B4	0	LVCMOS33_OUT	PT4B	Drive:8 mA, Clamp: On, Slew: Slow
FX3 Interface					
PinFx3Clk	V26	3	LVCMOS33_OUT	PR56D	DRIVE:8 mA CLAMP:ON SLEW:FAST
PinFx3Fv	D24	1	LVCMOS33_OUT	PT103A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
PinFx3Lv	C24	1	LVCMOS33_OUT	PT101B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
PinFx3Data[0]	W31	3	LVCMOS33_OUT	PR86A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[1]	W30	3	LVCMOS33_OUT	PR65C	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[2]	Y30	3	LVCMOS33_OUT	PR65D	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[3]	AB30	3	LVCMOS33_OUT	PR83A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[4]	Y32	3	LVCMOS33_OUT	PR86B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[5]	U26	3	LVCMOS33_OUT	PR50D	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[6]	AB29	3	LVCMOS33_OUT	PR83B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[7]	T31	3	LVCMOS33_OUT	PR65B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[8]	AD27	3	LVCMOS33_OUT	PR80A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[9]	AD26	3	LVCMOS33_OUT	PR77D	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[10]	AE27	3	LVCMOS33_OUT	PR80B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[11]	AB27	3	LVCMOS33_OUT	PR77B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[12]	AB28	3	LVCMOS33_OUT	PR77A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[13]	AC26	3	LVCMOS33_OUT	PR77C	DRIVE:4 mA CLAMP:ON SLEW:SLOW

Port Name	Pin	Bank	Buffer Type	Site	Properties
PinFx3Data[14]	V27	3	LVCMOS33_OUT	PR56C	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[15]	R32	3	LVCMOS33_OUT	PR65A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[16]	D25	1	LVCMOS33_OUT	PT107B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[17]	P32	3	LVCMOS33_OUT	PR62B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[18]	C25	1	LVCMOS33_OUT	PT107A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[19]	A25	1	LVCMOS33_OUT	PT105A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[20]	U31	3	LVCMOS33_OUT	PR62C	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[21]	T28	3	LVCMOS33_OUT	PR53B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[22]	R29	3	LVCMOS33_OUT	PR53A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[23]	U27	3	LVCMOS33_OUT	PR50C	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[24]	P29	3	LVCMOS33_OUT	PR47B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[25]	T27	3	LVCMOS33_OUT	PR50B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[26]	P30	3	LVCMOS33_OUT	PR59B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[27]	R27	3	LVCMOS33_OUT	PR50A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[28]	V29	3	LVCMOS33_OUT	PR56B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[29]	F17	1	LVCMOS33_OUT	PT69B	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[30]	V32	3	LVCMOS33_OUT	PR68C	DRIVE:4 mA CLAMP:ON SLEW:SLOW
PinFx3Data[31]	U29	3	LVCMOS33_OUT	PR56A	DRIVE:4 mA CLAMP:ON SLEW:SLOW
Status LEDs					
status_leds[0]	AM29	4	LVCMOS25_OUT	PB96B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[1]	AM28	4	LVCMOS25_OUT	PB96A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[2]	AJ29	4	LVCMOS25_OUT	PB121A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[3]	AG32	4	LVCMOS25_OUT	PB119B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[4]	AH32	4	LVCMOS25_OUT	PB119A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[5]	AK30	4	LVCMOS25_OUT	PB116B	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[6]	AK29	4	LVCMOS25_OUT	PB116A	DRIVE:8 mA CLAMP:ON SLEW:SLOW
status_leds[7]	AG30	4	LVCMOS25_OUT	PB114B	DRIVE:8 mA CLAMP:ON SLEW:SLOW

9. Ordering Information

Table 9.1. Ordering Information

Description	Ordering Part Number
Lattice Embedded Vision Development Kit	LF-EVDK1-EVN
USB3-GbE VIP I/O Board	USB3-VIP-EVN
MicroSD Card Adapter	MICROSD-ADP-EVN

References

For more information, refer to the following documents:

- ECP5 and ECP5-5G Family Data Sheet (FPGA-DS-02012)
- CrossLink Family Data Sheet (FPGA-DS-02007)
- SiI9136-3/SiI1136 HDMI Deep Color Transmitter (SiI-DS-1084)

For schematics, refer to the following documents:

- ECP5 VIP Processor Board Evaluation Board User Guide (FPGA-EB-02001)
- CrossLink VIP Input Bridge Board Evaluation Board User Guide (FPGA-EB-02002)
- HDMI VIP Output Bridge Board Evaluation Board User Guide (FPGA-EB-02003)
- USB3-GigE VIP IO Board User Guide (FPGA-EB-02016)

Technical Support

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

Revision History

Revision 1.0, May 2019

Section	Change Summary
All	Initial release

www.latticesemi.com