4 to 1 Image Aggregation with CrossLink-NX VIP Sensor Input Board

User Guide

FPGA-UG-02095-1.1

June 2021

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.

Contents

Figures

Tables

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
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 quad CSI-2 camera to High Definition Multimedia Interface (HDMI®) bridging that features the CrossLink-NX™ FPGA, ECP5™ FPGA and SiI1136 transmitter devices.

Figure 2.1 shows the Lattice Embedded Vision Development Kit that is designed as a stackable modular architecture with 80 mm × 80 mm form factor. The Lattice Embedded Vision Development Kit consists of three boards:

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

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

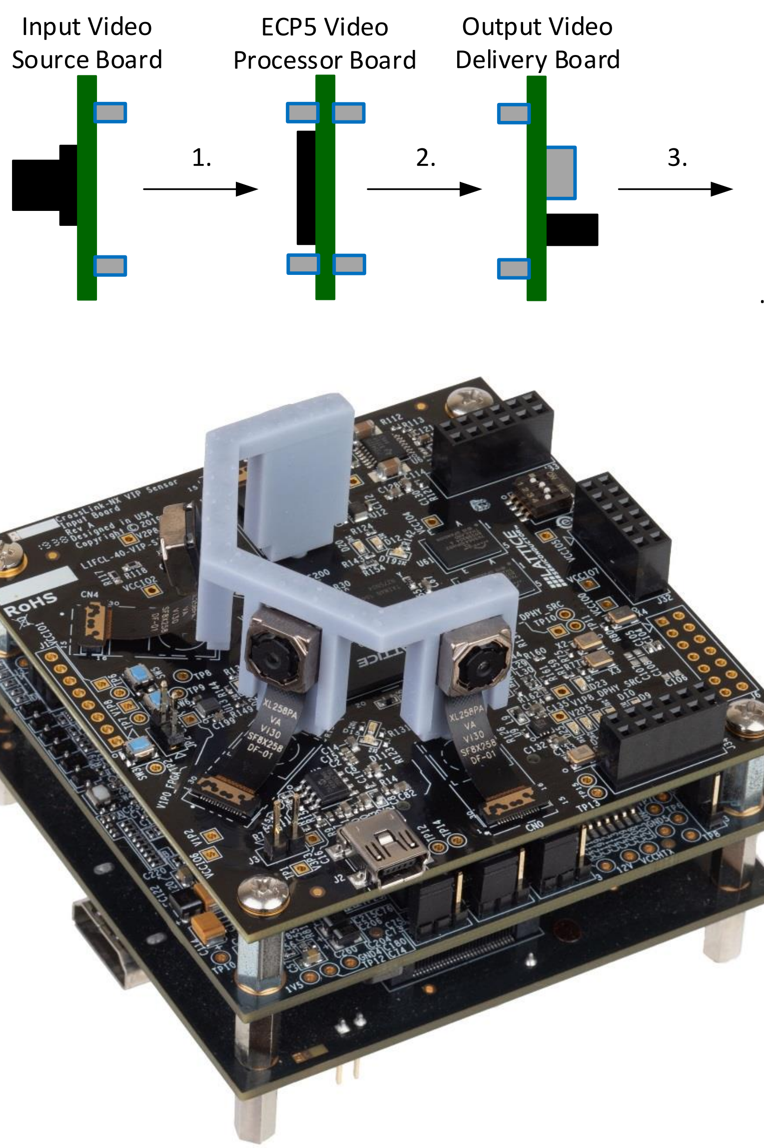

Figure 1.1. 4:1 MIPI CSI-2 to HDMI Bridge

2. Functional Description

The quad camera Mobile Industry Processing Interface (MIPI®) CSI-2 to HDMI demo uses a Sony IMX258 camera to output 1080p video over four MIPI data lanes, each running at 371.25 Mb/s. The CrossLink-NX VIP Sensor Input board receives the MIPI video stream from the onboard camera sensors and extracts the video pixels. These video pixels from four cameras are merged after cropping and the combined image data is transmitted to ECP5 in the form of parallel CMOS interface on the ECP5 video processor board through board-to-board connectors.

The ECP5 FPGA processes the merged sensor image and sends processed parallel image data to the Sil1136 HDMI transmitter on the HDMI VIP output bridge board through board to board connectors. The Sil1136 chip transmits the video data via HDMI to the 1080p display.

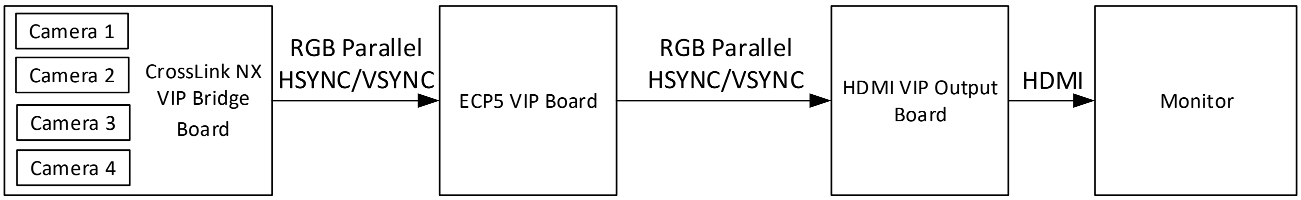

Figure 2.1. 4:1 MIPI CSI-2 to HDMI Bridge System Diagram

2.1. CrossLink-NX

The quad-camera-to-parallel design receives the serial, source-synchronous MIPI data from four MPI CSI-2 cameras, deserializes the serial data into bytes and extracts the control signal from MIPI data packets. This is done using two Hard (dedicated hardware) DPHY IP blocks and two Soft (using FPGA fabric) DPHY IP blocks. The byte data is sent to Byte to Pixel module which converts the byte data into RAW10 data. RAW10 data are converted to RGB888 in Debayer module and sent out to ECP5 on VIP Processor Board along with control signals. The onboard CSI-2 cameras are configured through the I^{2}C master interface on CrossLink-NX. Figure 2.2 shows the CrossLink-NX functional block diagram.

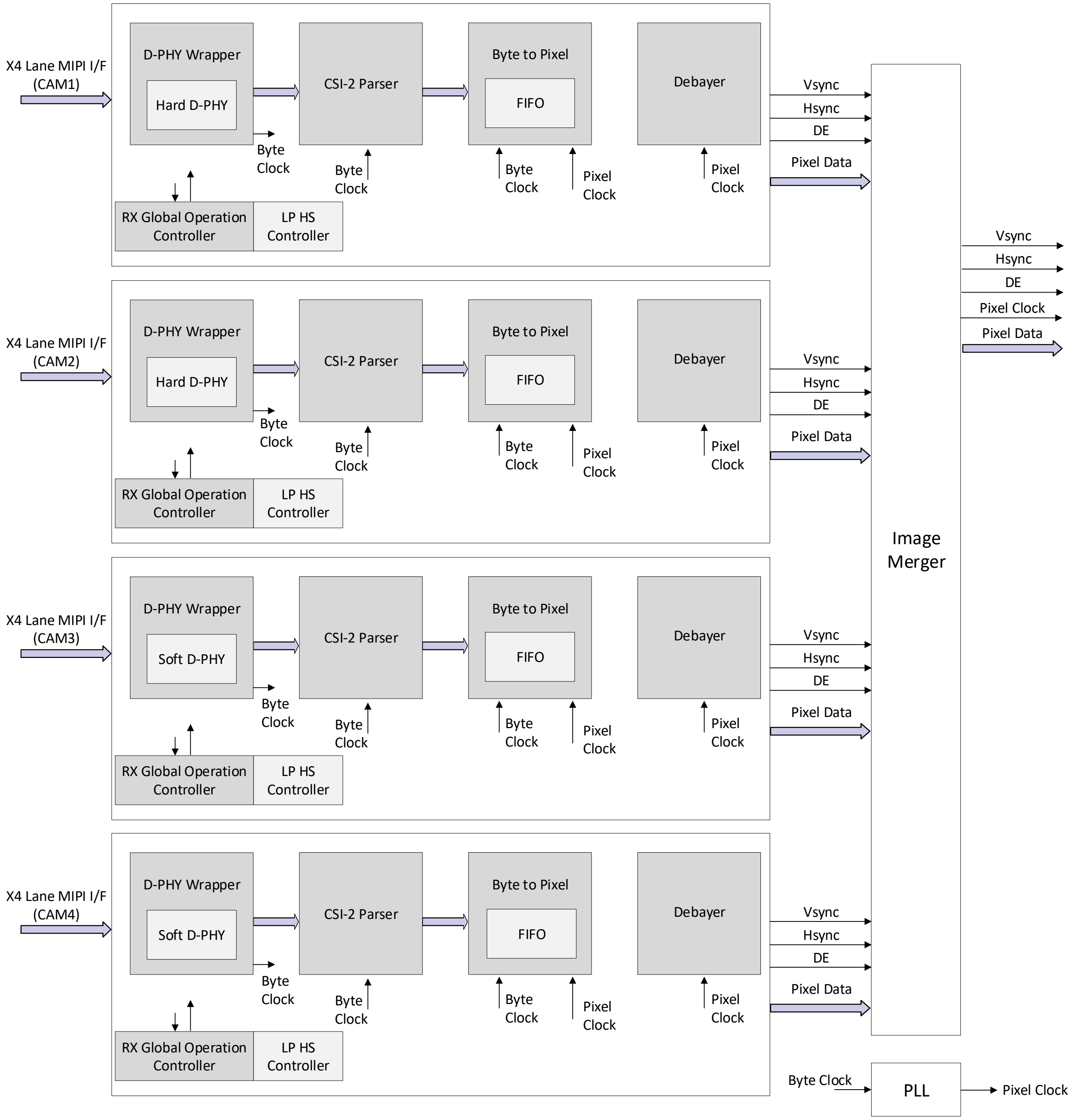

Figure 2.2. CrossLink-NX Functional Block Diagram

2.2. ECP5

The ECP5 FPGA receives image processed RGB888 data from CrossLink-NX, frame buffers the data, converts it to 36-bit RGB and sends it to the HDMI board. Figure 2.3 shows the signal flow in the ECP5.

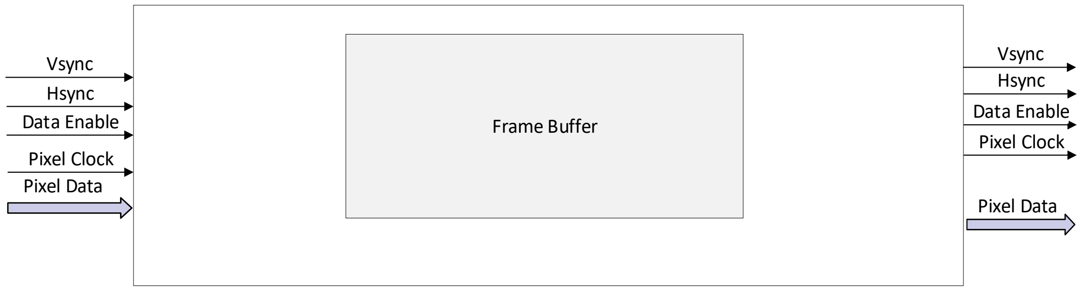

Figure 2.3. ECP5 Functional Block Diagram

2.3. SiI1136

Figure 2.4 shows the functional block diagram of the SiI1136 HDMI transmitter. This transmitter device is configured to output 1080p60 through the ECP5 I^{2}C Master interface on ECP5 VIP processor board. It receives 36-bit RGB data and control signals from ECP5 and converts it to HDMI format that is displayed on the HDMI monitor.

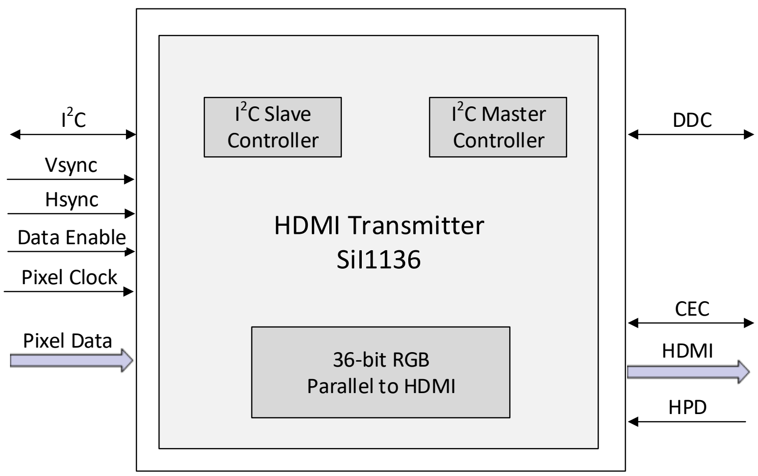

Figure 2.4. SiI1136 Functional Block Diagram

3. Demo Requirements

The following equipment is required for the demo:

- LFCL-VIP-SI-EVN Demo Kit
- HDMI monitor
- HDMI cable
- DC power adapter (12 V)
- Laptop/PC
- Bit/JED file
- USB 2.0 Type A to Mini-B cable*
- Lattice Radiant® Programmer version 3.0 or higher*

***Note**: Required only in re-programming.

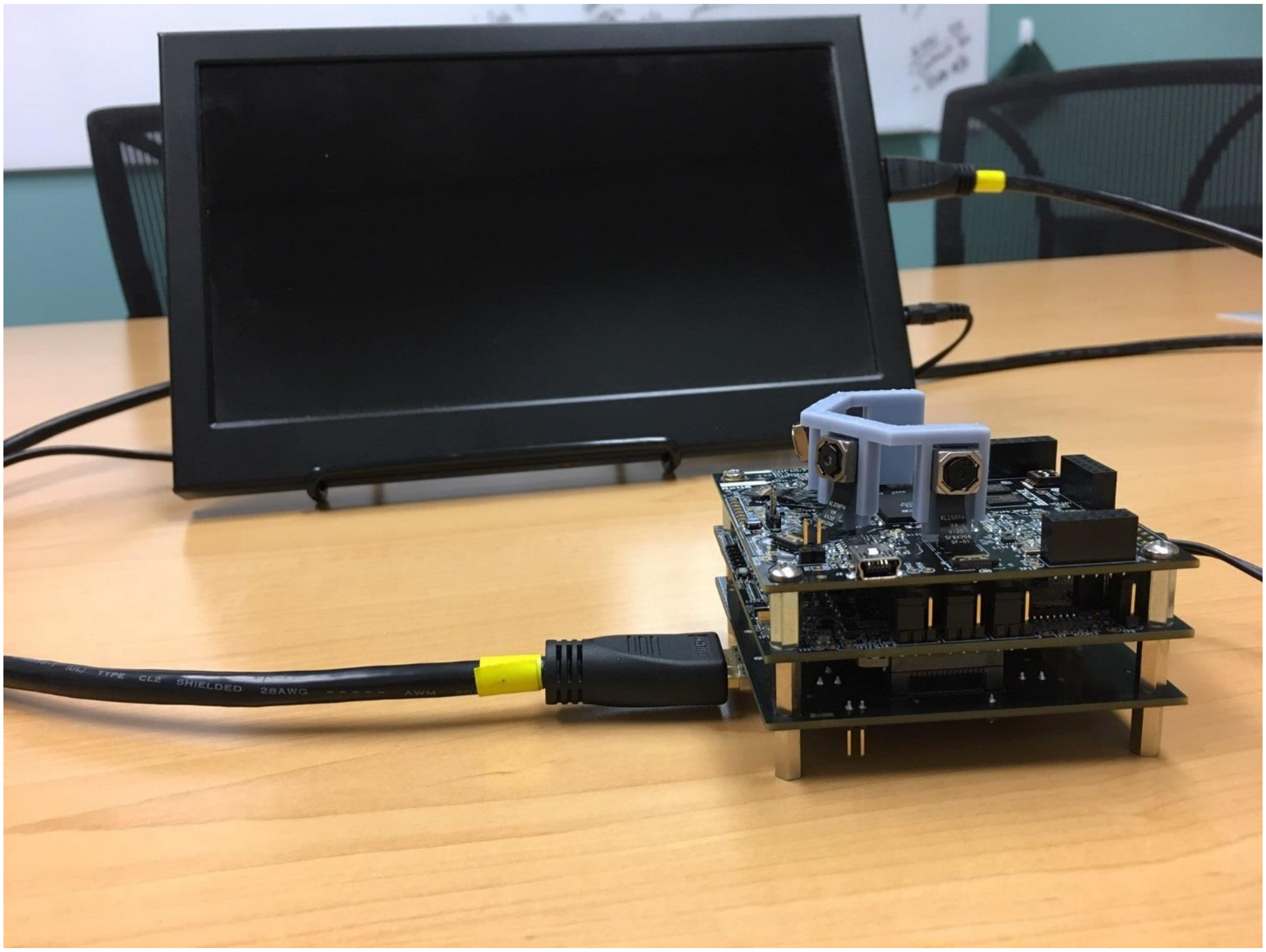

Figure 3.1. Quad Camera to HDMI Setup

3.1. CrossLink-NX VIP Sensor Input Board

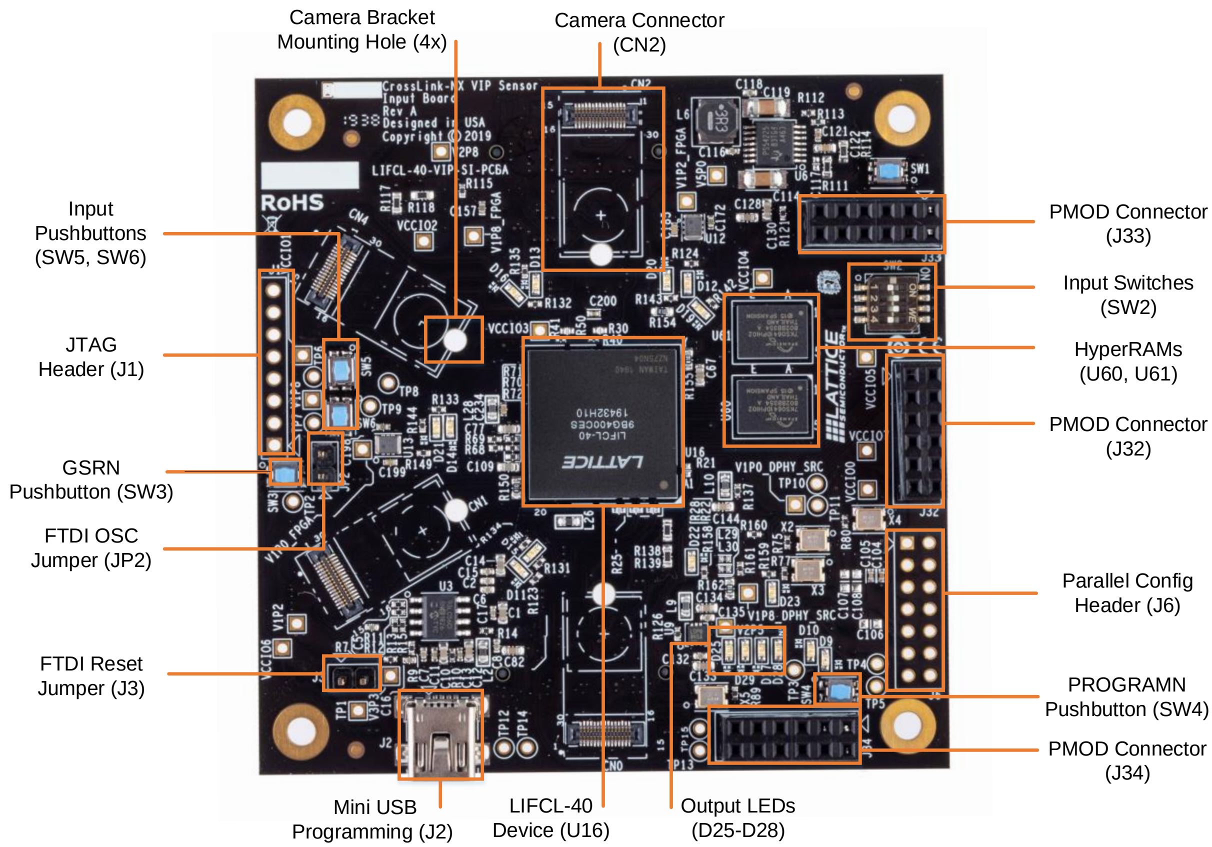

Figure 3.2.Top View of CrossLink-NX VIP Sensor Input Board

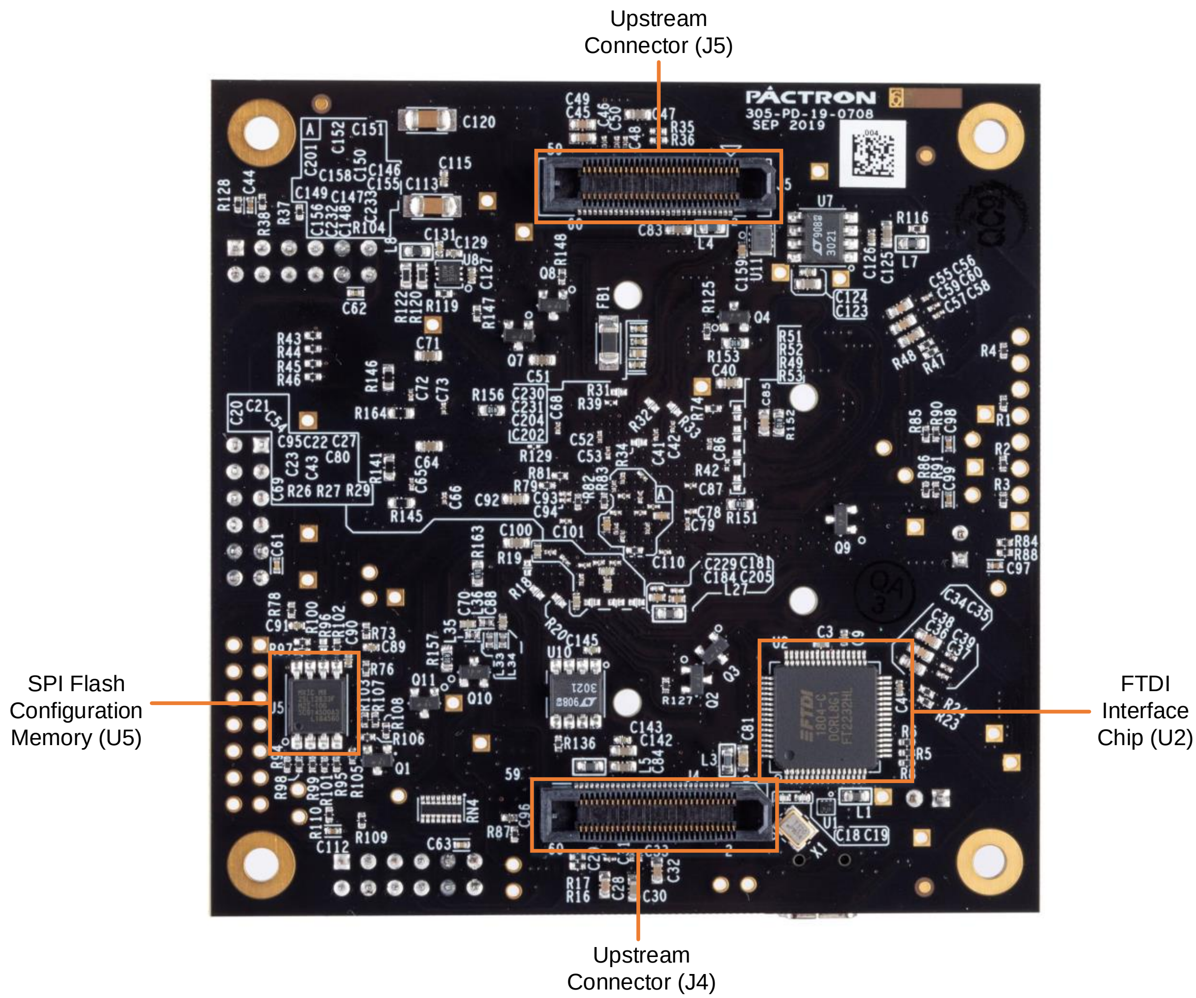

Figure 3.3. Bottom View of CrossLink-NX VIP Sensor Input Board

3.2. ECP5 VIP Processor Board

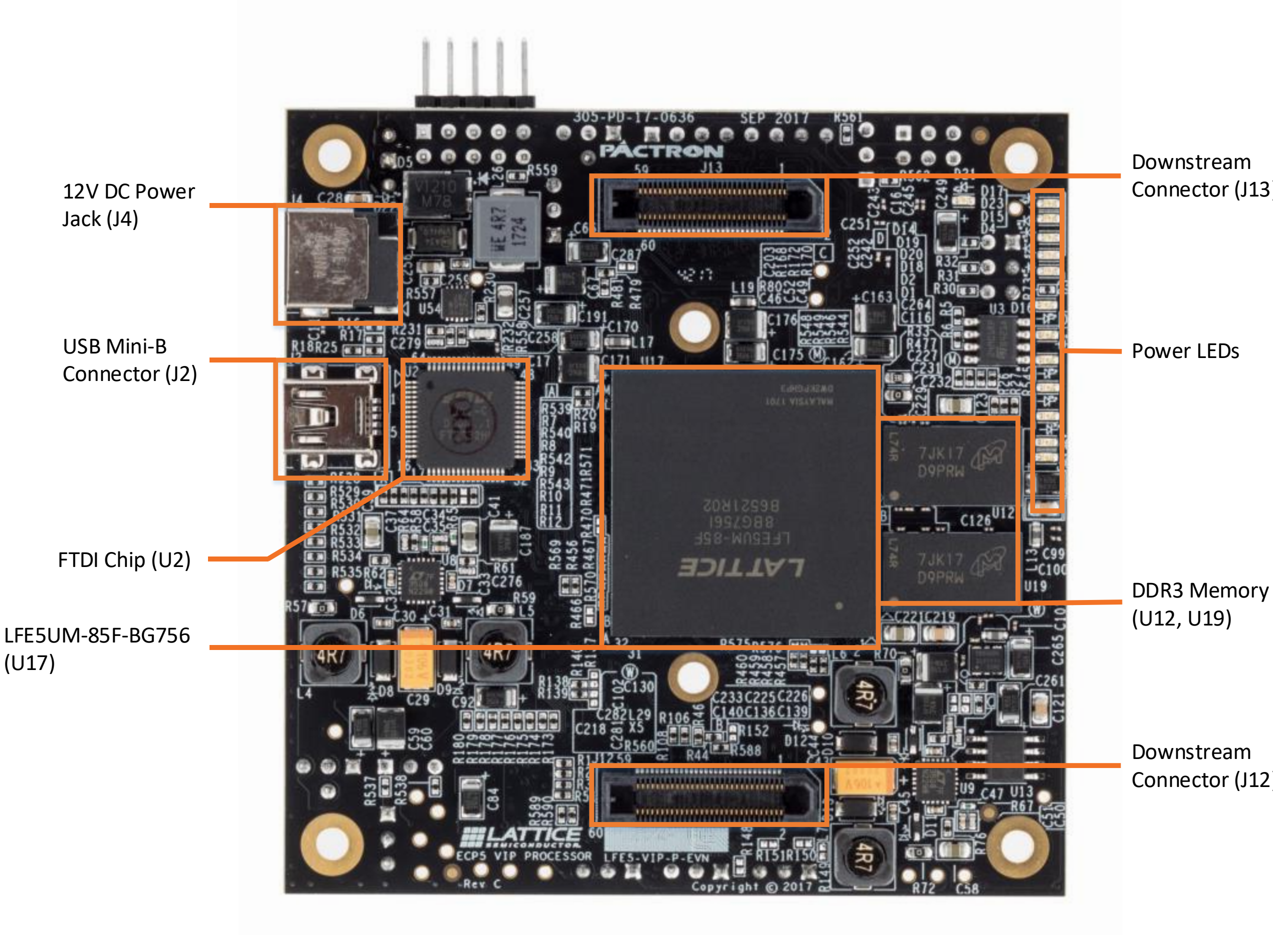

Figure 3.4. Top View of ECP5 VIP Processor Board

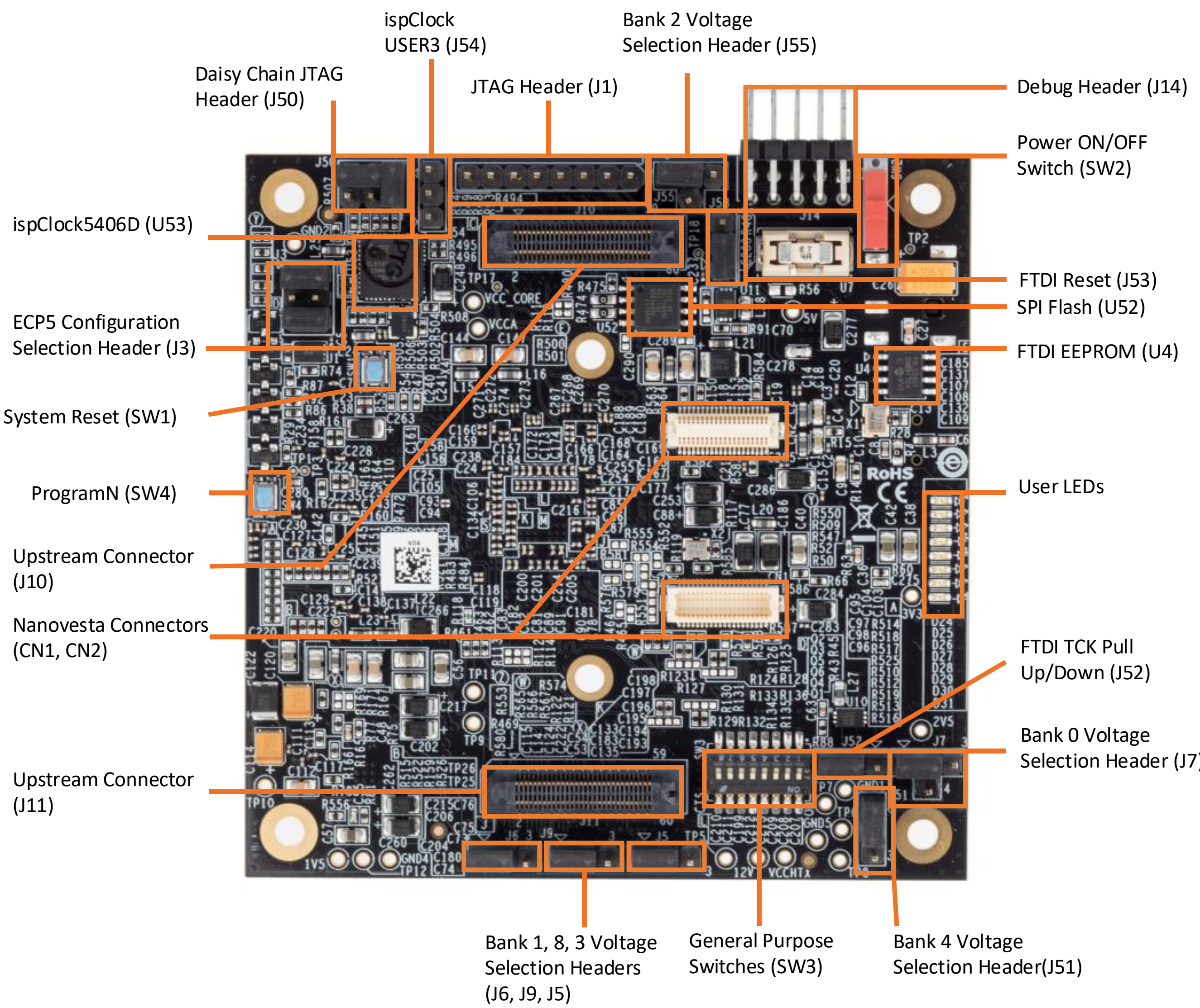

Figure 3.5. Bottom View of ECP5 VIP Processor Board

3.3. HDMI VIP Output Bridge Board

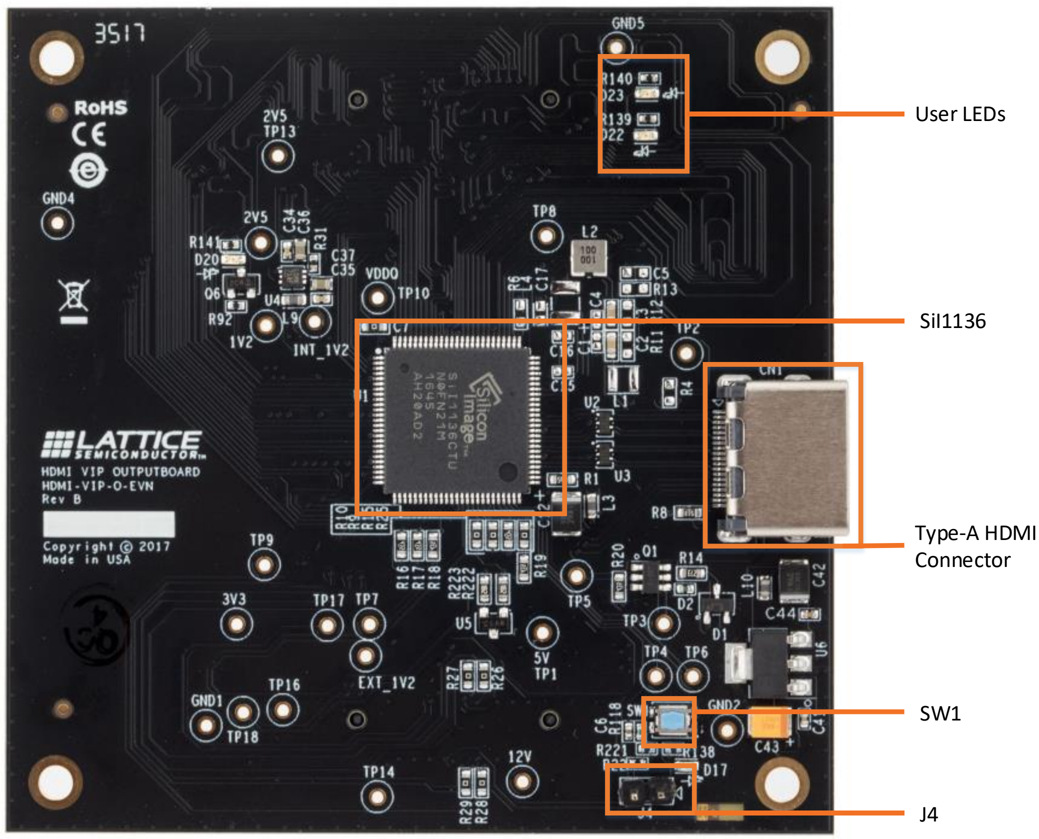

Figure 3.6. Top View of HDMI VIP Output Board

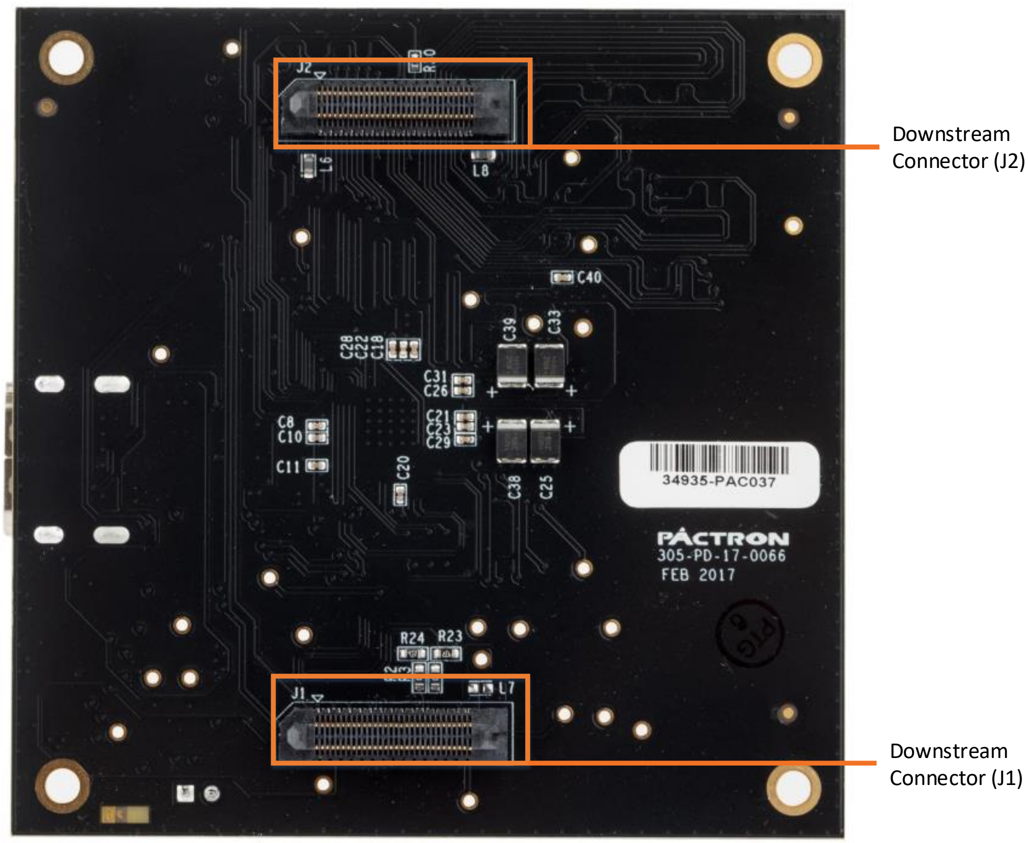

Figure 3.7. Bottom View of HDMI VIP Output Board

4. Jumper Settings

Table 4.1. CrossLink-NX VIP Sensor Input Board

Jumper	Description	Settings
JP2	FTDI oscillator jumper	Default short (osc connected). Alternate open (osc unconnected).
J3	FTDI Reset Jumper	Default open (active FTDI). Alternate short (reset FTDI).
—	—	All other headers should be kept open.

Table 4.2. ECP5 VIP Processor Board

Jumper	Description	Settings
J3	ECP5 Configuration Selection	Three bit setting for CFG0-2. Default is 010 (jumper on first and third positions).
J5	Bank 3 Voltage Selection	Default connect 1-2 (3.3 V). Alternate connect 2-3 (2.5 V).
J6	Bank 1 Voltage Selection	Default connect 1-2 (3.3 V). Alternate connect 2-3 (2.5 V).
J7	Bank 0 Voltage Selection	Default connect 2-3 (3.3 V). Alternate connect 1-2 (2.5 V).
J9	Bank 8 Voltage Selection	Default connect 1-2 (3.3 V). Alternate connect 2-3 (2.5 V).
J50	JTAG Daisy Chain	Default connect 1-2 and 3-5 (ECP5 Only). Alternates 2-4 and 5-6 (isp Only), and 1-2, 3-4, and 5-6 (Daisy Chain).
J51	Bank 4 Voltage Selection	Default connect 1-2 (3.3 V). Alternate connect 2-3 (2.5 V).
J52	FTDI TCK Pull Up/Down	Default connect 2-3 (JTAG). Alternate connect 1-2 (SPI).
J53	FTDI Reset	Default connect 1-2 (active FTDI). Alternate connect 2-3 (reset FTDI).
J55	Bank 2 Voltage	Default connect 2-3 (3.3 V). Alternate connect 1-2 (2.5 V).
—	—	All other headers should be kept open.

5. Demo Procedure

To set up the demonstration:

1. Connect the ECP5 VIP processor board to the wall socket using 12 V power adapter.
2. Power up the demo kit by turning on SW2 on ECP5 VIP processor board.
3. Connect the HDMI cable from CN1 of HDMI VIP output board to the HDMI display/monitor. The monitor displays the quad camera merged image as shown in Figure 5.1. The camera image is not downsized to fit the display quadrant. Each quadrant is composed of the respective quadrant of the input camera image.
4. By pressing SW5, the image on the display changes cyclically in the following order:

   merged image from four cameras, image from CAM1, image from CAM2, image from CAM3, image from CAM4

Figure 5.1. Quad Camera Merged Image

6. Pinout Information

6.1. CrossLink-NX

Table 6.1 lists the CrossLink-NX pinouts used for the demo.

Table 6.1. CrossLink-NX Pinouts

Port Name	Pin	Bank	Buffer Type	Site	Properties
gsr_n_i	G13	1	LVCMOS33_IN	PR4B	Pull: Up, Clamp: On, Hysteresis: On
ref_clk_i	L5	6	LVCMOS18_IN	PL30A	
Camera Sensor Interface					
sensor_reset_n_o	T1	5	LVCMOS18_OUT	PB4B	
rx0_mclk_o	M3	6	LVCMOS18_OUT	PL34B	Slew: Fast
rx0_scl	N5	6	LVCMOS18_OUT	PL40B	Slew: Fast
rx0_sda	N4	6	LVCMOS18_OUT	PL40A	Slew: Fast
rx0_clk_p_i	A2	61	DPHY_BIDI	DPHY0_CKP	—
rx0_clk_n_i	B1	61	DPHY_BIDI	DPHY0_CKN	—
rx0_d0_p_i	B2	61	DPHY_BIDI	DPHY0_DP0	—
rx0_d0_n_i	C1	61	DPHY_BIDI	DPHY0_DN0	—
rx0_d1_p_i	A3	61	DPHY_BIDI	DPHY0_DP1	—
rx0_d1_n_i	B3	61	DPHY_BIDI	DPHY0_DN1	—
rx0_d2_p_i	C2	61	DPHY_BIDI	DPHY0_DP2	—
rx0_d2_n_i	D1	61	DPHY_BIDI	DPHY0_DN2	—
rx0_d3_p_i	A4	61	DPHY_BIDI	DPHY0_DP3	—
rx0_d3_n_i	B4	61	DPHY_BIDI	DPHY0_DN3	—
rx1_mclk_o	M4	6	LVCMOS18_OUT	PL36A	Slew: Fast
rx1_scl	N7	6	LVCMOS18_OUT	PL42B	Slew: Fast
rx1_sda	N6	6	LVCMOS18_OUT	PL42A	Slew: Fast
rx1_clk_p_i	A8	61	DPHY_BIDI	DPHY1_CKP	—
rx1_clk_n_i	B8	61	DPHY_BIDI	DPHY1_CKN	—
rx1_d0_p_i	A7	61	DPHY_BIDI	DPHY1_DP0	—
rx1_d0_n_i	B7	61	DPHY_BIDI	DPHY1_DN0	—
rx1_d1_p_i	A9	61	DPHY_BIDI	DPHY1_DP1	—
rx1_d1_n_i	B9	61	DPHY_BIDI	DPHY1_DN1	—
rx1_d2_p_i	A6	61	DPHY_BIDI	DPHY1_DP2	—
rx1_d2_n_i	B6	61	DPHY_BIDI	DPHY1_DN2	—
rx1_d3_p_i	A10	61	DPHY_BIDI	DPHY1_DP3	—
rx1_d3_n_i	B10	61	DPHY_BIDI	DPHY1_DN3	—
rx2_mclk_o	M5	6	LVCMOS18_OUT	PL36B	Slew: Fast
rx2_scl	P2	6	LVCMOS18_OUT	PL44B	Slew: Fast
rx2_sda	P1	6	LVCMOS18_OUT	PL44A	Slew: Fast
rx2_clk_p_i	W11	3	DPHY_BIDI	PB54A	—
rx2_clk_n_i	Y11	3	LVCMOS12H	PB54B	—
rx2_d0_p_i	V11	3	DPHY_BIDI	PB56A	—
rx2_d0_n_i	U11	3	LVCMOS12H	PB56B	—
rx2_d1_p_i	W13	3	DPHY_BIDI	PB60A	—
rx2_d1_n_i	V12	3	LVCMOS12H	PB60B	—
rx2_d2_p_i	U12	3	DPHY_BIDI	PB62A	—
rx2_d2_n_i	T12	3	LVCMOS12H	PB62B	—

Port Name	Pin	Bank	Buffer Type	Site	Properties
rx2_d3_p_i	R12	3	DPHY_BIDI	PB64A	—
rx2_d3_n_i	P12	3	LVCMOS12H	PB64B	—
rx3_mclk_o	M6	6	LVCMOS18_OUT	PL38A	Slew: Fast
rx3_scl	P6	6	LVCMOS18_OUT	PL46B	Slew: Fast
rx3_sda	P5	6	LVCMOS18_OUT	PL46A	Slew: Fast
rx3_clk_p_i	T13	3	DPHY_BIDI	PB74A	—
rx3_clk_n_i	T14	3	LVCMOS12H	PB74B	—
rx3_d0_p_i	Y15	3	DPHY_BIDI	PB76A	—
rx3_d0_n_i	Y16	3	LVCMOS12H	PB76B	—
rx3_d1_p_i	U15	3	DPHY_BIDI	PB78A	—
rx3_d1_n_i	V16	3	LVCMOS12H	PB78B	—
rx3_d2_p_i	V17	3	DPHY_BIDI	PB80A	—
rx3_d2_n_i	U16	3	LVCMOS12H	PB80B	—
rx3_d3_p_i	P13	3	DPHY_BIDI	PB70A	—
rx3_d3_n_i	R13	3	LVCMOS12H	PB70B	—
ctrl_i	L20	1	LVCMOS33_IN	PR20A	—
h_ctrl_i	L19	1	LVCMOS33_IN	PR20B	—
ECP5 Interface					
reset_n_i	F18	1	LVCMOS33_IN	PR10A	
pixel_clk	Y19	2	LVCMOS33_OUT	PR49B	Slew: Fast
vs_o	V18	2	LVCMOS33_OUT	PR47A	Slew: Fast
hs_o	V19	2	LVCMOS33_OUT	PR47B	Slew: Fast
de_o	W19	2	LVCMOS33_OUT	PR49A	Slew: Fast
rd_o[0]	R17	2	LVCMOS33_OUT	PR40A	Slew: Fast
rd_o[1]	R18	2	LVCMOS33_OUT	PR40B	Slew: Fast
rd_o[2]	U20	2	LVCMOS33_OUT	PR42A	Slew: Fast
rd_o[3]	T20	2	LVCMOS33_OUT	PR42B	Slew: Fast
rd_o[4]	W20	2	LVCMOS33_OUT	PR44A	Slew: Fast
rd_o[5]	V20	2	LVCMOS33_OUT	PR44B	Slew: Fast
rd_o[6]	T18	2	LVCMOS33_OUT	PR46A	Slew: Fast
rd_o[7]	U18	2	LVCMOS33_OUT	PR46B	Slew: Fast
gd_o[0]	M19	2	LVCMOS33_OUT	PR32A	Slew: Fast
gd_o[1]	M20	2	LVCMOS33_OUT	PR32B	Slew: Fast
gd_o[2]	N19	2	LVCMOS33_OUT	PR34A	Slew: Fast
gd_o[3]	N20	2	LVCMOS33_OUT	PR34B	Slew: Fast
gd_o[4]	P19	2	LVCMOS33_OUT	PR36A	Slew: Fast
gd_o[5]	P20	2	LVCMOS33_OUT	PR36B	Slew: Fast
gd_o[6]	P17	2	LVCMOS33_OUT	PR38A	Slew: Fast
gd_o[7]	P18	2	LVCMOS33_OUT	PR38B	Slew: Fast
bd_o[0]	N14	2	LVCMOS33_OUT	PR24A	Slew: Fast
bd_o[1]	M14	2	LVCMOS33_OUT	PR24B	Slew: Fast
bd_o[2]	M16	2	LVCMOS33_OUT	PR26A	Slew: Fast
bd_o[3]	M15	2	LVCMOS33_OUT	PR26B	Slew: Fast
bd_o[4]	N15	2	LVCMOS33_OUT	PR27A	Slew: Fast
bd_o[5]	N16	2	LVCMOS33_OUT	PR27B	Slew: Fast
bd_o[6]	M17	2	LVCMOS33_OUT	PR30A	Slew: Fast
bd_o[7]	M18	2	LVCMOS33_OUT	PR30B	Slew: Fast

6.2. ECP5

Table 6.2 lists the ECP5 pinouts used for the demo.

Table 6.2. ECP5 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
q	AG30	4	LVCMOS33_OUT	PB114B	Drive:8 mA, Clamp: On, Slew: Slow
ddr_clk_in	C5	7	LVDS_IN	PL11A	Clamp: On
em_ddr_reset_n	C4	T	SSTL15_I_OUT	PL62B	Drive:8 mA, Clamp: On, Slew: Slow
CrossLink-NX Interface					
CLNX_pix_clk	P27	2	LVCMOS33_IN	PR44C	Pull: Down, Clamp: On, Hysteresis: On
CLNX_vs	P26	2	LVCMOS33_IN	PR38B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_de	D31	2	LVCMOS33_IN	PR38B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[0]	A13	0	LVCMOS33_IN	PT42B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[1]	A8	0	LVCMOS33_IN	PT20B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[2]	F9	0	LVCMOS33_IN	PT22A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[3]	D9	0	LVCMOS33_IN	PT22B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[4]	C9	0	LVCMOS33_IN	PT24A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[5]	A9	0	LVCMOS33_IN	PT24B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[6]	C10	0	LVCMOS33_IN	PT29B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_bd[7]	B10	0	LVCMOS33_IN	PT31A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[0]	A10	0	LVCMOS33_IN	PT31B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[1]	E11	0	LVCMOS33_IN	PT33B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[2]	D11	0	LVCMOS33_IN	PT36A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[3]	C11	0	LVCMOS33_IN	PT36B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[4]	F29	2	LVCMOS33_IN	PR14C	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[5]	F28	2	LVCMOS33_IN	PR14D	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[6]	C29	2	LVCMOS33_IN	PR11C	Pull: Down, Clamp: On, Hysteresis: On
CLNX_gd[7]	C30	2	LVCMOS33_IN	PR11D	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[0]	F32	2	LVCMOS33_IN	PR23C	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[1]	H32	2	LVCMOS33_IN	PR23D	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[2]	J29	2	LVCMOS33_IN	PR32C	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[3]	K29	2	LVCMOS33_IN	PR32D	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[4]	D13	0	LVCMOS33_IN	PT40B	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[5]	C13	0	LVCMOS33_IN	PT42A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[6]	K27	2	LVCMOS33_IN	PR38A	Pull: Down, Clamp: On, Hysteresis: On
CLNX_rd[7]	K26	2	LVCMOS33_IN	PR38B	Pull: Down, Clamp: On, Hysteresis: On
reset_crosslink	B14	0	LVCMOS33_OUT	PT49A	Drive:8 mA, Clamp: On, Slew: Slow
Camera Sensor Interface					
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
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

Port Name	Pin	Bank	Buffer Type	Site	Properties
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
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	3	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

7. Ordering Information

Table 7.1. Ordering Information

Description	Ordering Part Number
Lattice Embedded Vision Development Kit	LF-EVDK1-EVN

Appendix A. Lattice Embedded Vision Development Kit Setup

To set up the display demo boards:

1. Connect the J5 and J4 connectors of the CrossLink-NX VIP Sensor Input Board to the J10 and J11 connectors of the ECP5 VIP board.
2. Connect the J13 and J12 connectors of the ECP5 VIP board to the J2 and J1 connectors of the HDMI VIP output board.
3. Connect one end of the HDMI cable to the C1 connector of the HDMI VIP output board and the other end to the monitor.
5. Connect the 12 V wall power adapter cable to the J4 connector of the ECP5 VIP board.
6. The 4:1 CSI-2 camera to HDMI Bridge design should be programmed into the SPI Flash on the EVDK. This loads the reference design on power up. Refer to Appendix B. Programming the Lattice Embedded Vision Development Kit to update or change the FPGA or SPI Flash images.

Appendix B. Programming the Lattice Embedded Vision Development Kit

Using Lattice Radiant and Diamond Programmers with the EVDK

The EVDK has a built-in download controller for programming. It uses an FT2232H Future Technology Devices International (FTDI) part to convert USB to JTAG. To use the built-in download controller, connect the USB cable from J2 of the ECP5 VIP Processor Board to your PC (with Diamond programming software installed). A mini USB to USB-A cable is included in the EVDK. The USB hub on the PC detects the cable of the USB function on Port 0, making the built-in download controller available for use with the Diamond programming software.

During a JTAG scan, the Diamond Programmer sees the LFE5UM-85F. A JTAG scan also erases the ECP5 image, requiring you to reprogram the device. When using the Diamond Programmer, selecting **Create a new blank project** and manually selecting the device family and device prevents erasure.

The Crosslink-NX is programmed using the Lattice Radiant tool. The above procedure can be implemented using Lattice Radiant Programmer on the Crosslink-NX VIP Sensor Input Board.

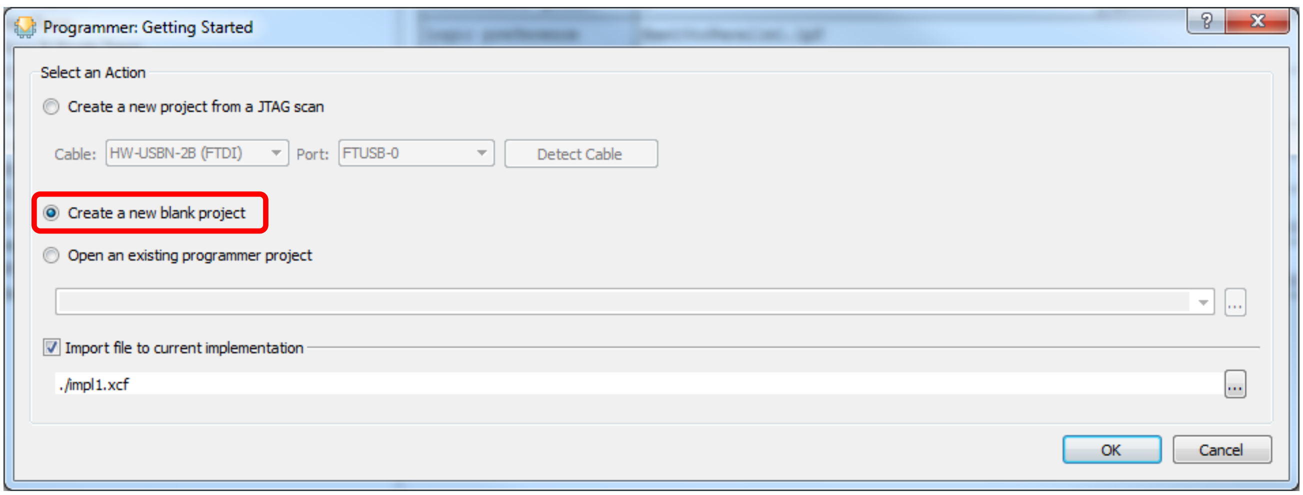

Figure B.1. Create a New Blank Project

ECP5 SPI Flash Programming

Erasing the ECP5 Prior to Reprogramming

If the ECP5 is already programmed (either directly, or loaded from SPI Flash), erase first the ECP5 SRAM memory, then program the ECP5's SPI Flash in the next section. Keep the board powered when re-programming the SPI Flash in the next section.

To erase the ECP5:

1. Launch Diamond Programmer with **Create a new blank project**.
2. Select **ECP5UM** for **Device Family** and **LFE5UM-85F** for **Device**.

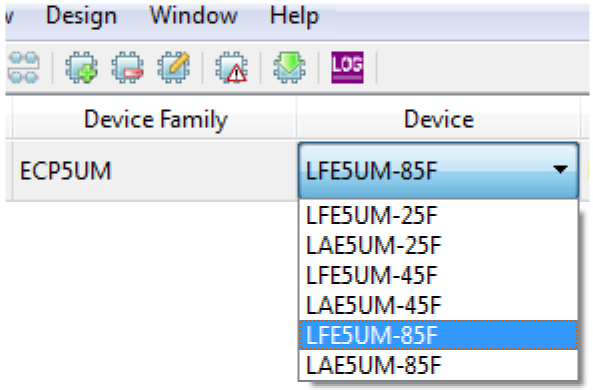

Figure B.2. Selecting Device

3. Right-click and select **Device Properties**.
4. Select **JTAG 1532 Mode** for **Access Mode** and **Erase Only** for **Operation**.

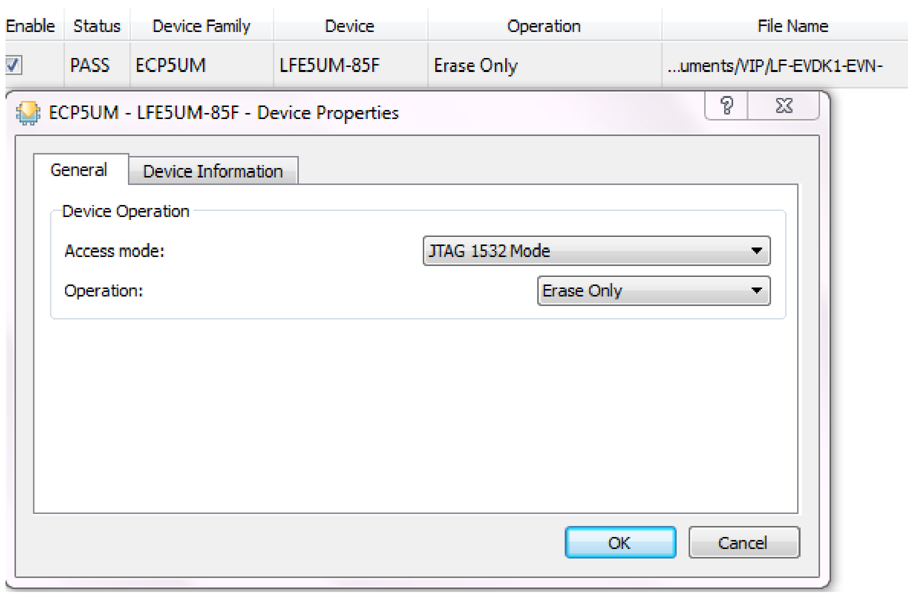

Figure B.3. Device Operation

5. Click **OK** to close the Device Properties window.
6. Click the **Program** button  in Diamond Programmer to start the Erase sequence.

Programming the SPI on the ECP5 VIP Processor Board

To program the SPI:

1. Ensure the ECP5 device is erased by performing Steps 1-6.
2. Right-click and select **Device Properties**.
3. Select **SPI Flash Background Programming** for **Access mode** and make the following selections:
   a. For **Programming File**, browse and select the **ECP5 bitfile** (*.bit)
   b. For **SPI Flash Options**, refer Table B.1.

Table B.1. SPI Flash Options Selection Guide

Item	Rev A/B	Rev C
Family	SPI Serial Flash	SPI Serial Flash
Vendor	Micron	Macronix
Device	SPI-N25Q128A	MX25L12835F
Comment	—	If the above device is not available in Diamond Programmer, select Macronix MX25L12805.

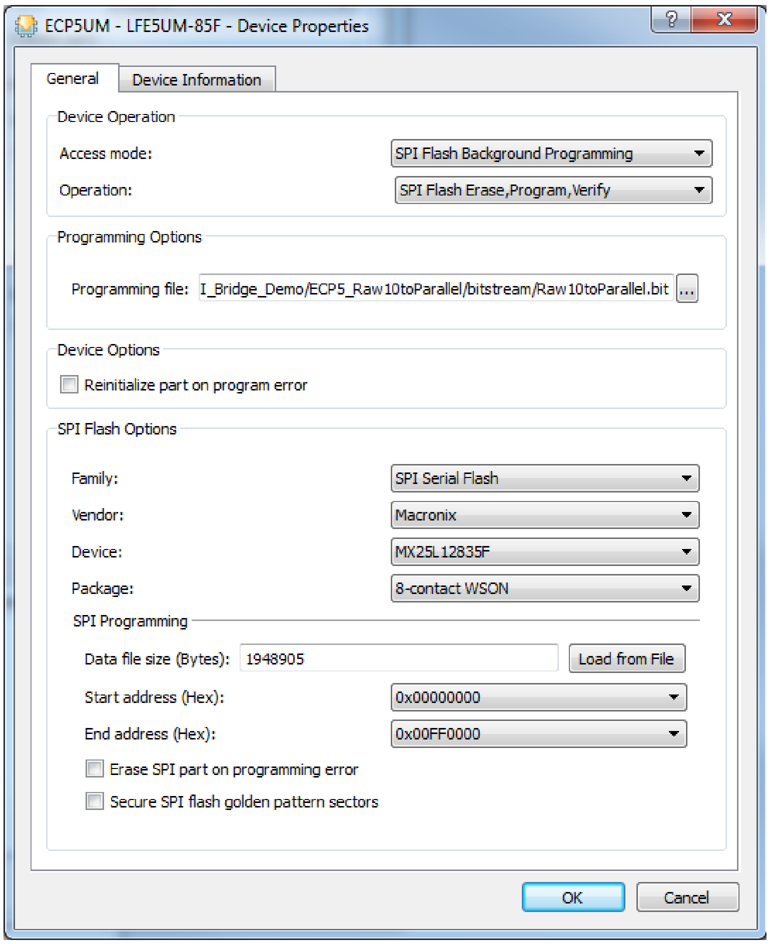

Figure B.4. Device Properties

4. Click **OK** to close the **Device Properties** window.
5. Click the **Program** button  in Diamond Programmer to start the programming sequence.
6. After successful programming, the **Output** console displays the results as shown in Figure B.5

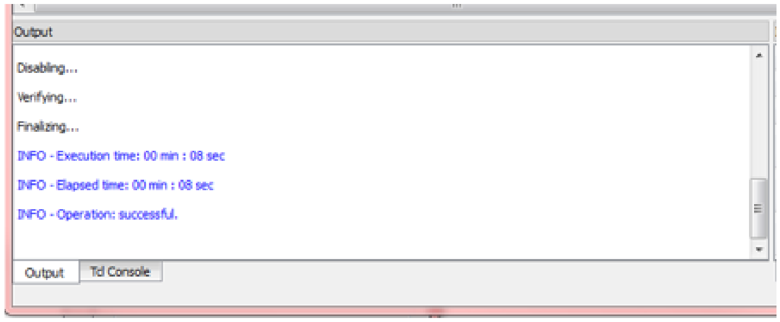

Figure B.5. Output Console

CrossLink-NX SPI Flash Programming

Erasing the CrossLink-NX FPGA Prior to Reprogramming

If the CrossLink-NX device is already programmed (either directly, or loaded from SPI Flash), follow this procedure to first erase the CrossLink-NX SRAM memory before re-programming the CrossLink-NX's SPI Flash. If you are doing this, keep the board powered when re-programming the SPI Flash (so it does not reload on reboot).

To erase CrossLink-NX:

1. Launch Lattice Radiant Programmer with **Create a new blank project**.
2. Select **LIFMD** for **Device Family** and **LIFCL** for **Device Vendor** and **LIFCL-40** for **Device**.

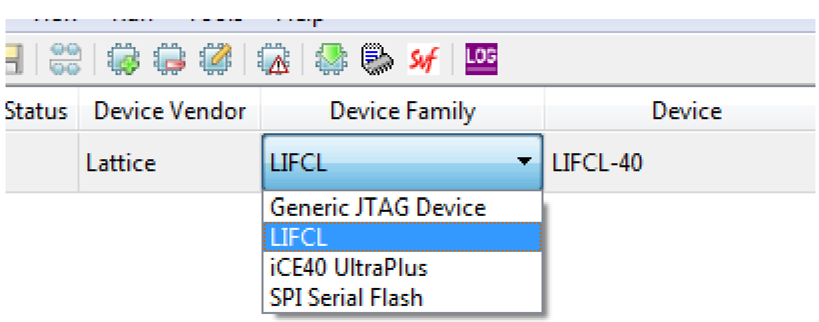

Figure B.6. Select Device

3. Right-click and select **Device Properties**.
4. Select JTAG for Port Interface, Direct Programming for Access Mode and Erase Only for Operation.

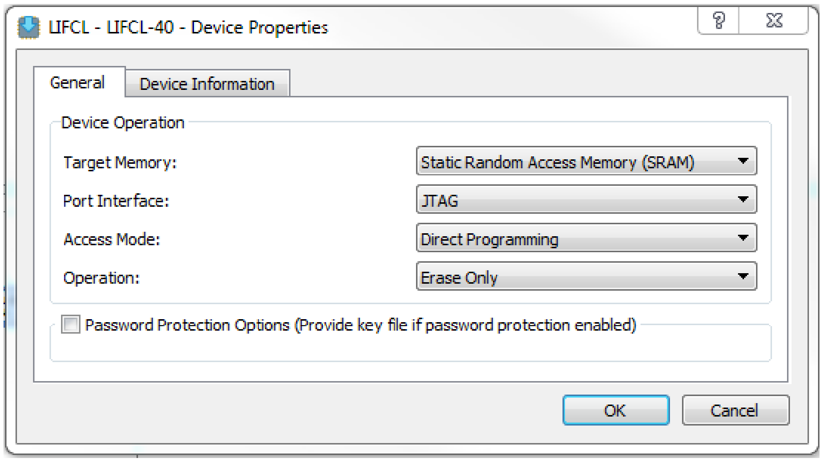

Figure B.7. Device Operation

5. Click **OK** to close the Device Properties window.
6. Click the **Program** button  in Lattice Radiant Programmer to start the Erase sequence.

Programming the SPI on the CrossLink-NX VIP Sensor Input Board

To program the SPI:

1. Ensure the CrossLink-NX device is erased by performing Steps 1-6.
2. Right-click and select **Device Properties**.
3. Select **SPI FLASH** for **Access mode, JTAG2SPI** for **Port Interface,** and **Direct Programming** for **Access Mode.**
4. For **Programming File**, browse and select the **CrossLink-NX bitfile** (*.bit).
5. For **SPI Flash Options**, make the selections in Figure B.8 below to slect the Macronix 25L12833F device.

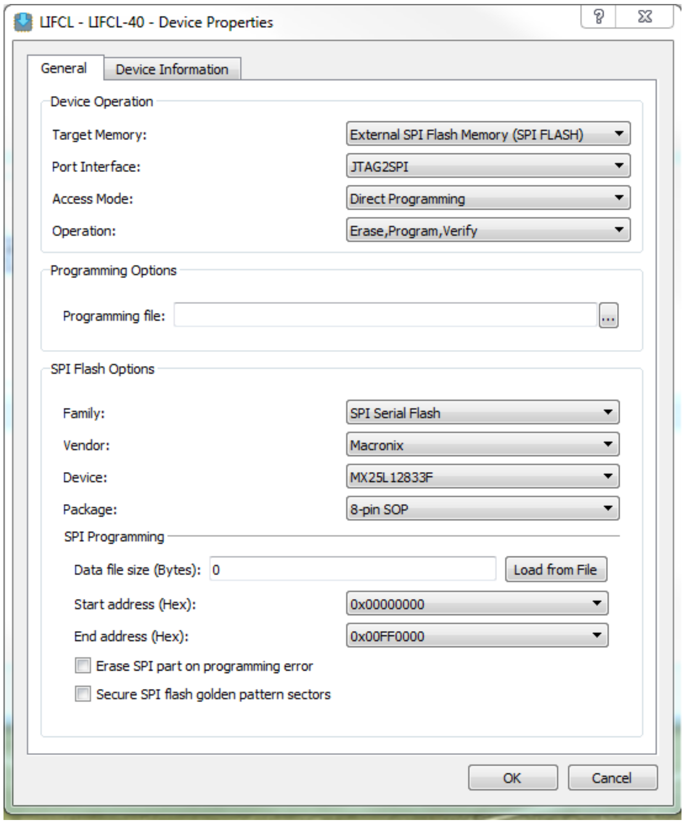

Figure B.8. Device Properties

6. Click **OK** to close the **Device Properties** window.
7. Click the **Program** button  in Lattice Radiant Programmer to start the programming sequence.

References

For more information, refer to:

- ECP5 and ECP5-5G Family Data Sheet (FPGA-DS-02012, previously DS1044)
- CrossLink-NX Family Data Sheet (FPGA-DS-02049)
- SiI9136-3/SiI1136 HDMI Deep Color Transmitter (SiI-DS-1084)

For schematics, refer to:

- ECP5 VIP Processor Board Evaluation Board User Guide (FPGA-EB-02001)
- CrossLink-NX VIP Sensor Input Board Evaluation Board User Guide (FPGA-EB-02029)
- HDMI VIP Output Bridge Board Evaluation Board User Guide (FPGA-EB-02003)

Technical Support

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

Revision History

Revision 1.1, June 2021

Section	Change Summary
Demo Requirements	Changed Lattice Radiant® Programmer version 2.X to version 3.0.

Revision 1.0, December 2019

Section	Change Summary
All	Initial release.

www.latticesemi.com