EVDK Based Object Counting Demonstration

User Guide

FPGA-UG-02050-1.2

August 2019

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

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CNN	Convolutional Neural Network
EVDK	Embedded Vision Development Kit
FPGA	Field-Programmable Gate Array
LED	Light-emitting diode
MLE	Machine Learning Engine
SDHC	Secure Digital High Capacity
SDXC	Secure Digital eXtended Capacity
SPI	Serial Peripheral Interface
VIP	Video Interface Platform
USB	Universal Serial Bus
NN	Neural Network

1. Introduction

This document provides technical information and instructions for setting up and running the EVDK Based Object (fruit) Counting Demo. This demo is designed to utilize the Lattice Machine Learning Engine (MLE) IP and implemented onto the Lattice Embedded Vision Development Kit (EVDK). The EVDK Based Object Counting Demo performs the fruit counting using the camera on the EVDK and feeds the video stream through the Convolutional Neural Network (CNN) inside Lattice MLE. Red and green dots identify the detected apples and oranges respectively while overlay text indicates the number of apples and oranges detected. The CNN in the Lattice MLE is trained with apple and orange images.

Refer to the following documents for detailed information on Lattice development boards and kit:

- Lattice Embedded Vision Development Kit User Guide (FPGA-UG-02015)
- CrossLink VIP Input Bridge Board Evaluation Board User Guide (FPGA-EB-02002)
- ECP5 VIP Processor Board Evaluation Board User Guide (FPGA-EB-02001)
- HDMI VIP Output Bridge Board Evaluation Board User Guide (FPGA-EB-02003)

2. Functional Description

The EVDK Based Fruit Counting Demo is designed to utilize the Lattice Embedded Vision Development Kit with MicroSD Card Adapter Board, as shown in Figure 2.1.

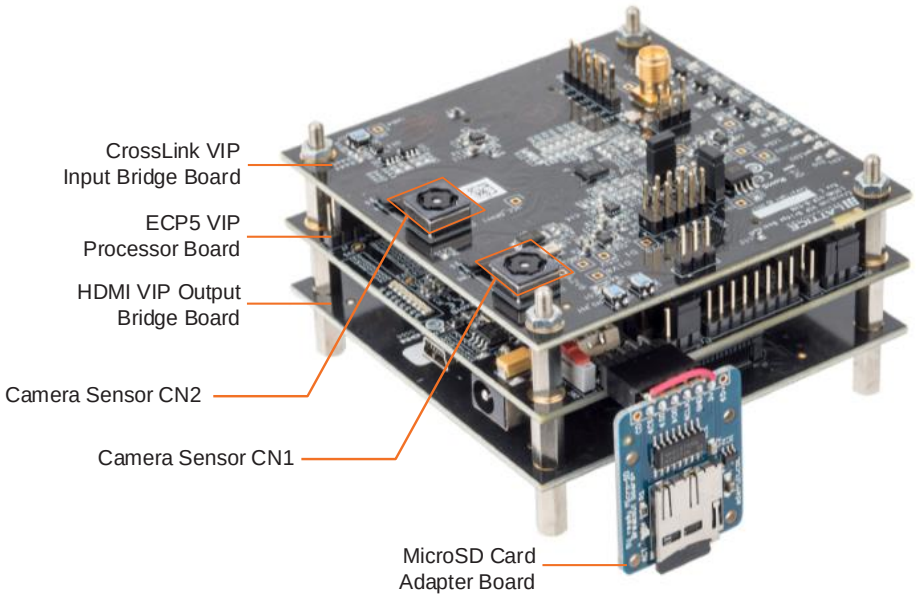

Figure 2.1. Lattice EVDK with MicroSD Card Adapter Board

The Lattice Embedded Vision Development Kit features a stackable modular architecture consisting of three boards:

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

Figure 2.1 shows Revision C of the Embedded Vision Development Kit. For earlier revisions, refer to the user guide of the specific evaluation board. For more information on the Embedded Vision Development Kit, visit the Lattice website Embedded Vision Development Kit page.

The firmware, which holds the CNN training results (from Caffe tool) is stored inside the SD card. The MLE detects the objects (fruit) and results are overlaid as red and green dots identifying the detected apples and oranges, respectively, and text indicating the number of apples and oranges detected.

As shown in Figure 2.2, the video data taken by the camera sensor (CN2) on the CrossLink VIP Input Bridge Board are fed into the ECP5 VIP Processor Board where the MLE processes the image data. This data, with weights and biases from the firmware, is used to create the dots and text overlay.

The implementation of this demo in ECP5-85 consists of 8 Neural Network engines (NN) engines. The implemented Neural Network allows a 224x224 RGB Input with 8 convolution layers.

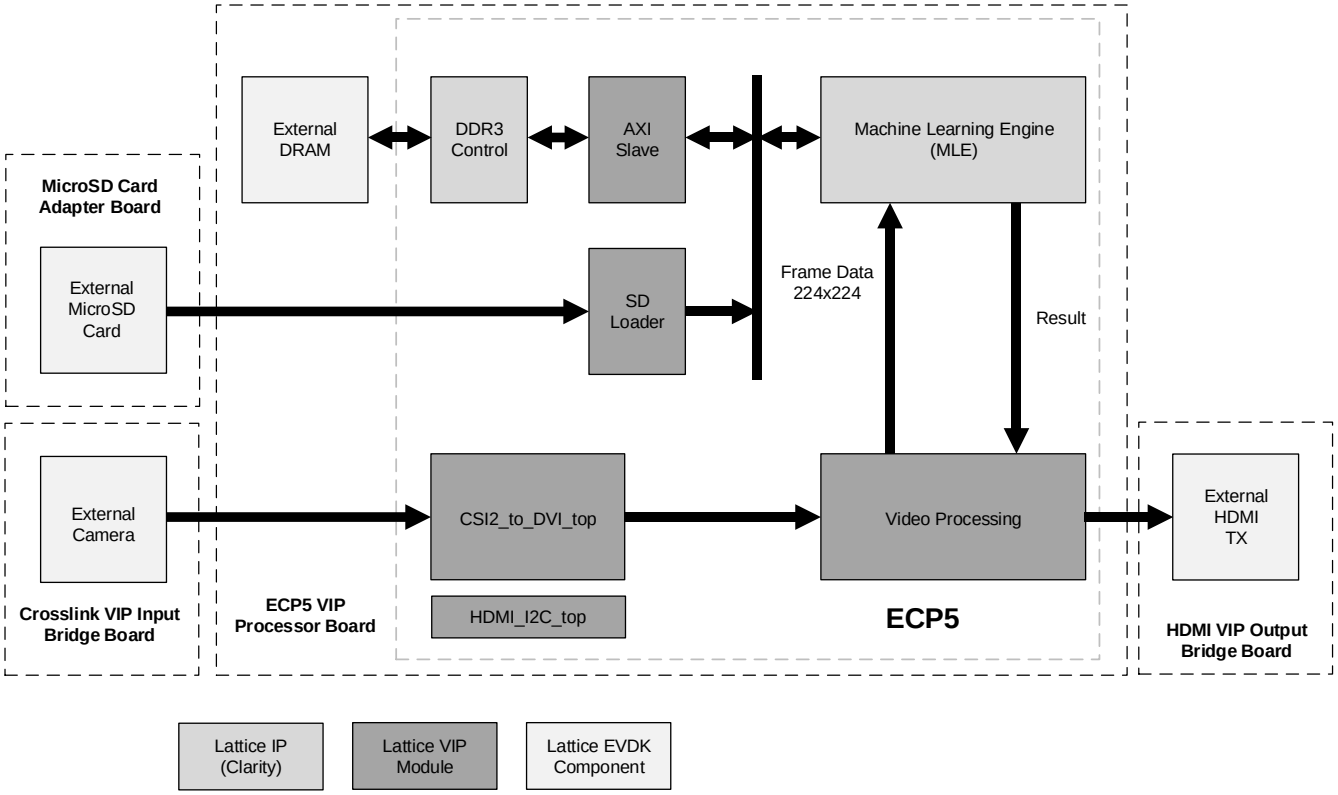

Figure 2.2. Object Counting Demo Diagram

3. Demo Setup

This section describes the demo setup.

3.1. Hardware Requirements

- Lattice Embedded Vision Development Kit (LF-EVDK1-EVN)
  - Mini-USB Cable (Included in the kit)
  - 12 V Power Supply (Included in the kit)
- HDMI Cable
- HDMI Monitor (1080p60)
- Machine Learning Adapter Card (ML-EVN-ADP)
  - The microSD Card Adapter (MICROSD-ADP-EVN) has been replaced by the Machine Learning Adapter Card (ML-EVN-ADP). The Machine Learning Adapter card includes an optional microphone input, but is otherwise a 100% functional replacement for the microSD Card Adapter.
- MicroSD Card (Standard only - less than 2 GB, not SDHC/SDXC and others)

3.2. Software and Firmware Requirements

- Diamond Programmer (Refer to www.latticesemi.com/programmer)
- Programming files for Embedded Vision Development Kit
  - Dual_Camera_to_Parallel_Crosslink.bit (targets CrossLink)
  - vip_objectcount_ecp5.bit (targets ECP5)
- MicroSD card Image writer software (Win32diskimager)
  - URL link: https://sourceforge.net/projects/win32diskimager/
- MicroSD card image
  - vip_objectcount.bin

3.3. Board Settings

Before programming the boards, perform the following steps:

1. On the ECP5 VIP Input Bridge Board, make sure the jumper settings are as shown in Figure 3.1.
2. On the CrossLink VIP Processor Board (see Figure 3.2), ensure that SW2 is ON to power the board (LEDs should be ON).
3. Connect the 12 V power supply to the barrel plug J4.
4. Connect the mini-USB cable from the PC to the mini-USB connector J2.

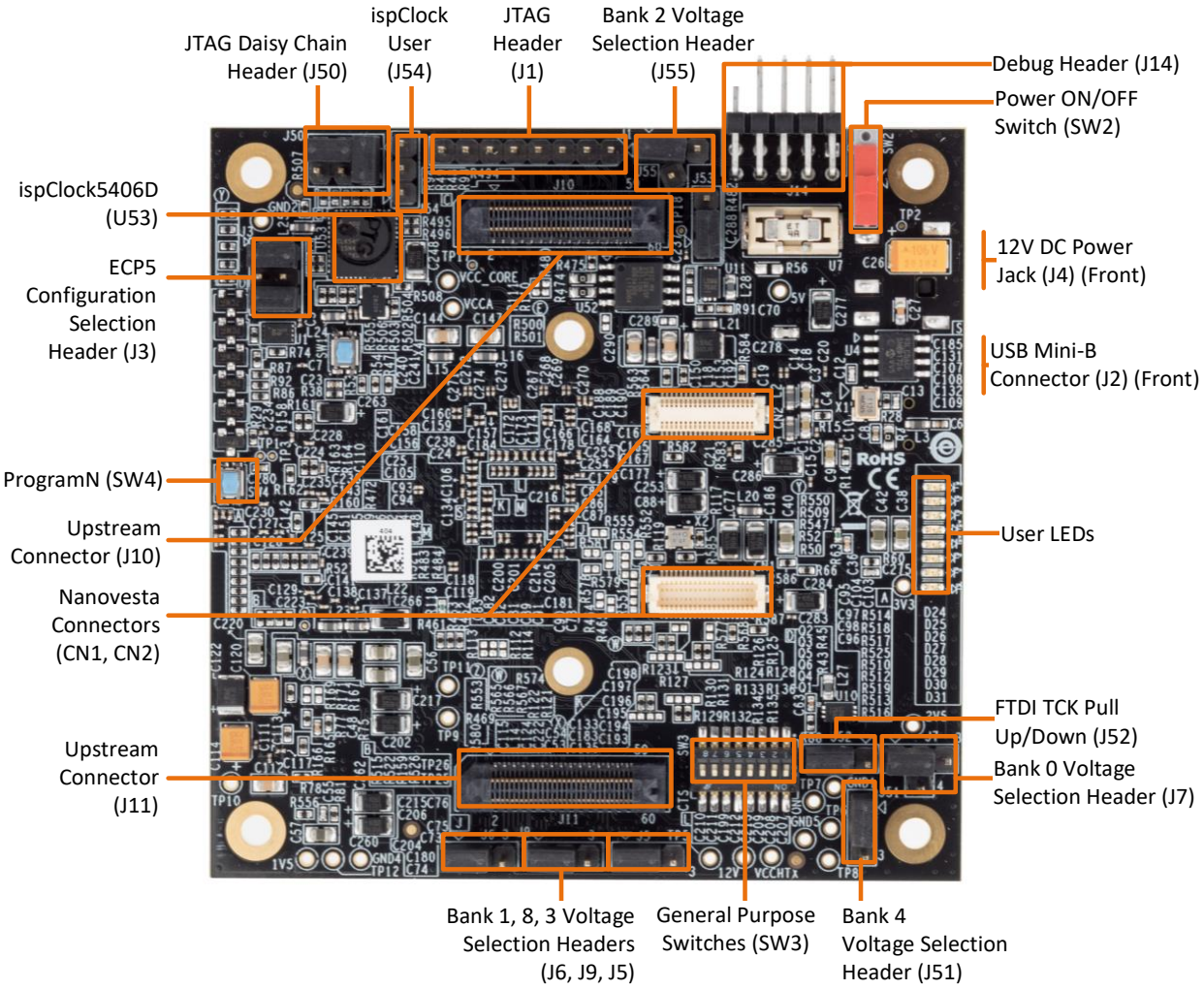

Figure 3.1. Back View of ECP5 VIP Input Bridge Board

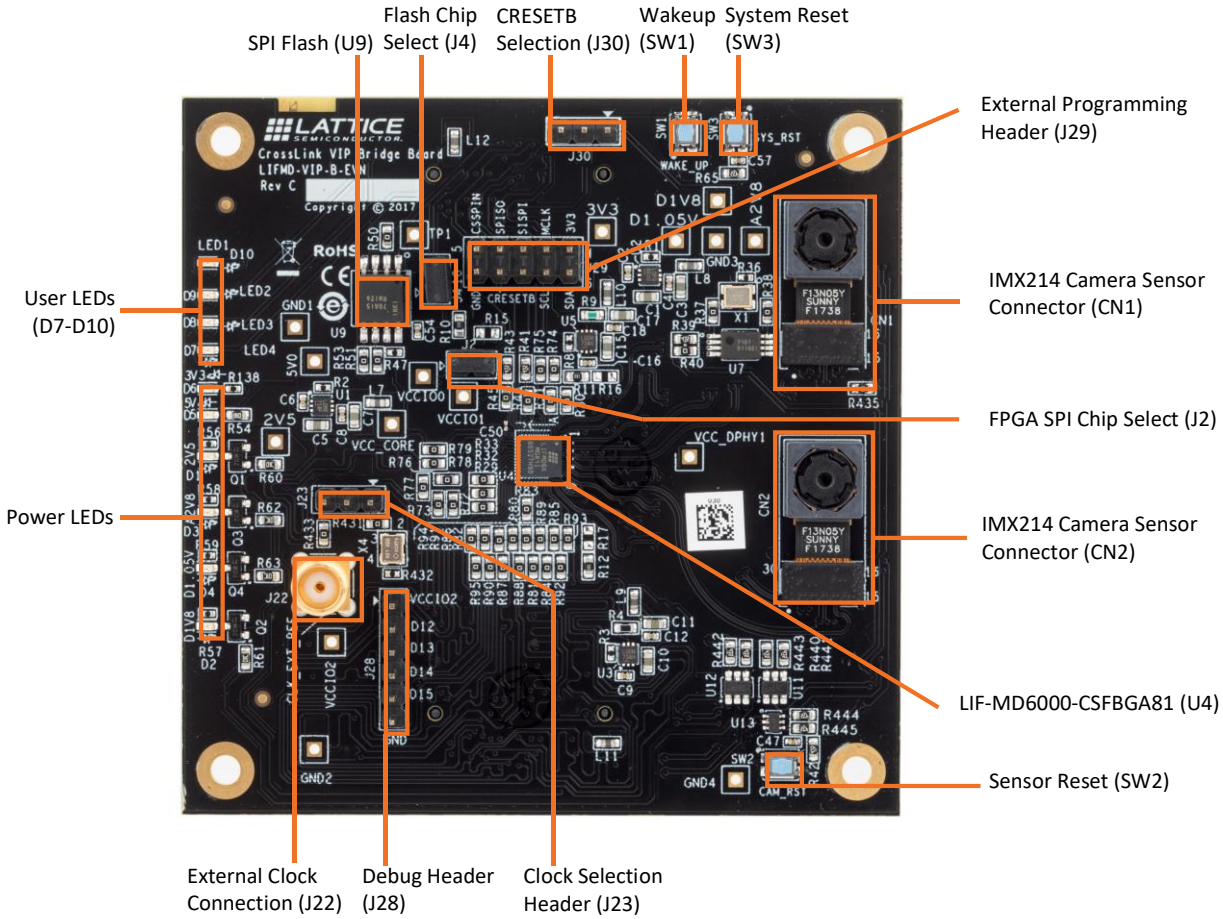

Figure 3.2. Top View of CrossLink VIP Input Bridge Board

4. Programming the Demo

Both the CrossLink VIP Input Bridge Board and the ECP5 VIP Processor Board must be configured and programmed. Also, the demo design firmware must be programmed onto the MicroSD card which is plugged into the MicroSD Card Adaptor Board.

4.1. Programming the ECP5 SPI Flash

4.1.1. Erasing the ECP5 SRAM Prior to Reprogramming

If the ECP5 device is already programmed (either directly or loaded from SPI Flash), erase the ECP5 SRAM before reprogramming the ECP5 SPI Flash. Keep the board powered on to prevent reloading on reboot.

To erase the ECP5 SRAM:

1. Start Diamond Programmer. In the Getting Started dialog box, select **Create a new blank project**.
2. Click **OK**.
3. In the Diamond Programmer main interface, select **ECP5UM** in Device Family and **LFE5UM-85F** in Device as shown in Figure 4.1.

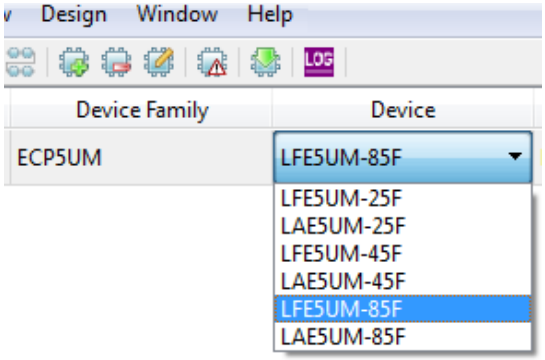

Figure 4.1. Device Selection

4. Click the ECP5 row and select **Edit > Device Properties**.
5. In the Device Properties dialog box, select **JTAG 1532 Mode** in Access mode and **Erase Only** in Operation (shown in Figure 4.2).

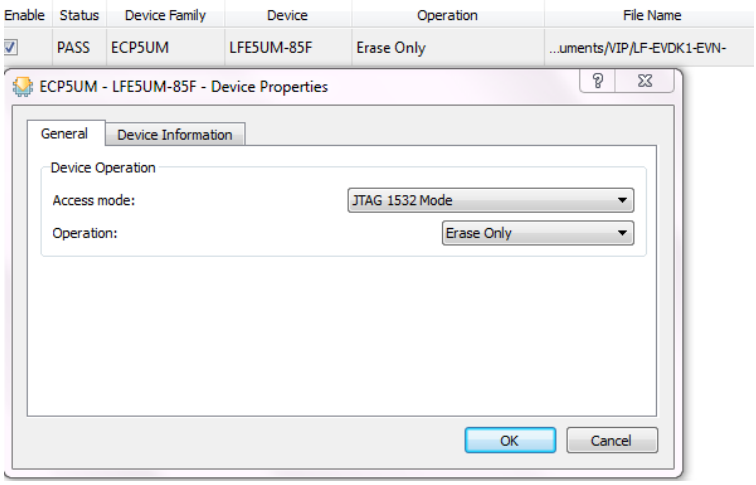

Figure 4.2. Device Operation

6. Click **OK** to close the Device Properties dialog box.
7. In the Diamond Programmer main interface, click the **Program** button to start the Erase operation.

Note: If you power OFF/ON the board, the SPI Flash reprograms the ECP5 device. In this case, you have to repeat steps 1 to 7.

4.1.2. Programming the ECP5 VIP Processor Board

To program the ECP5 VIP Processor Board:

1. Ensure that the ECP5 device is erased by performing the steps in Erasing the ECP5 SRAM Prior to Reprogramming.
2. In the Diamond Programmer main interface, click the ECP5 row and select **Edit > Device Properties** to open the Device Properties dialog box as shown in Figure 4.3.

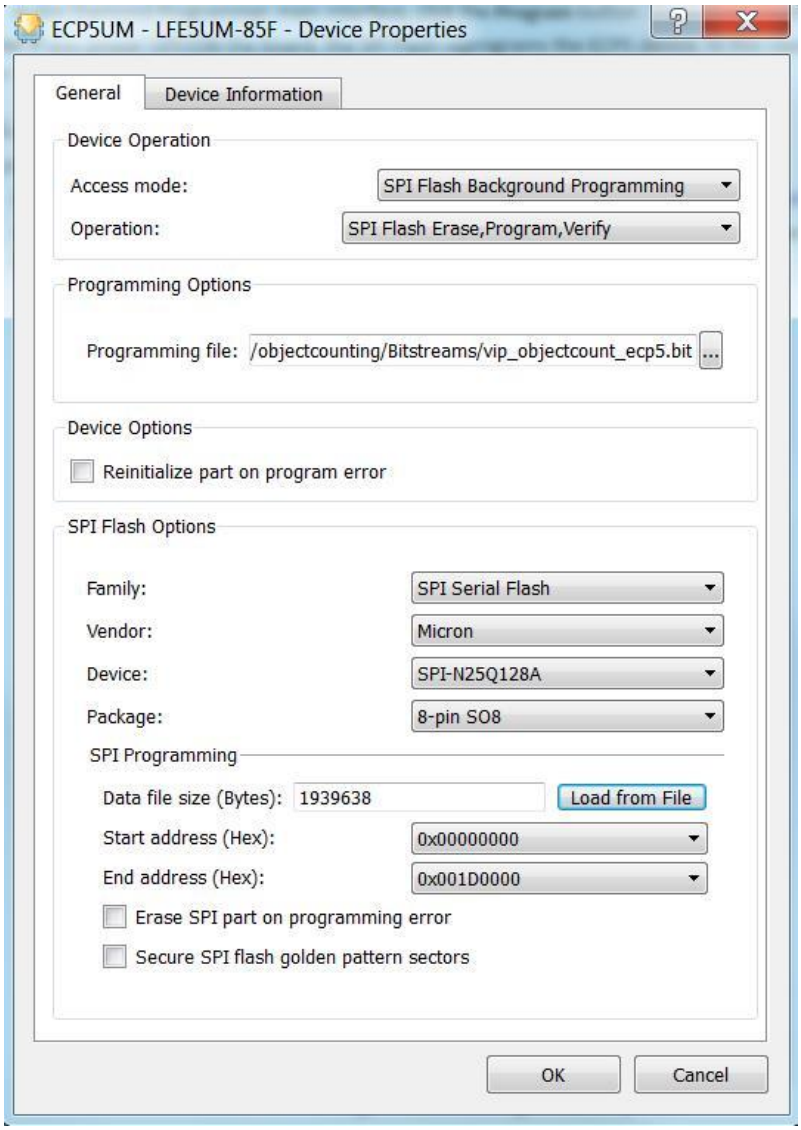

Figure 4.3. Selecting General Options

3. Apply the settings below:
   a. Under Device Operation, select the options below:
      - Access Mode: **SPI Flash Background Programming**
      - Operation: **Erase, Program, Verify**
   b. Under Programming Options, select the bitstream file **~/Demonstration/ vip_objectcount_ecp5.bit** in Programming file.
   c. Under **SPI Flash Options**, select the options below based on the board revision.

      Revision B
      - Family: **SPI Serial Flash**
      - Vendor: **Micron**
      - Device: **SPI-N25Q128A**
      - Package: **8-pin SO8**

      Revision C
      - Family: **SPI Serial Flash** (**SPI Serial Flash Beta** for Diamond 3.10 or earlier)
      - Vendor: **Macronix**
      - Device: **MX25L12835F**
      - Package: **8-Land WSON**
   d. Click **Load from File** to update the Data file size (Bytes) value.
   e. Ensure that the following addresses are correct:
      - Start Address (Hex): **0x00000000**
      - End Address (Hex): **0x001D0000**
4. Click **OK**.
5. In the Diamond Programmer main interface, click the **Program** button to start the programming operation. Successful programming is displayed in the Diamond Programmer Output console as shown in Figure 4.4.

Figure 4.4. Output Console

4.2. Programming the CrossLink SPI Flash

4.2.1. Erasing the CrossLink SRAM Prior to Reprogramming

If the CrossLink is already programmed (either directly or loaded from SPI Flash), erase the CrossLink SRAM before reprogramming the CrossLink SPI Flash. Keep the board powered on to prevent reloading on reboot.

To erase the CrossLink device SRAM:

1. Start Diamond Programmer. In the Getting Started dialog box, select **Create a new blank project**.
2. Click **OK**.
3. In the Diamond Programmer main interface, select **LIFMD** in Device Family and **LIF-MD6000** in Device as shown in Figure 4.5.

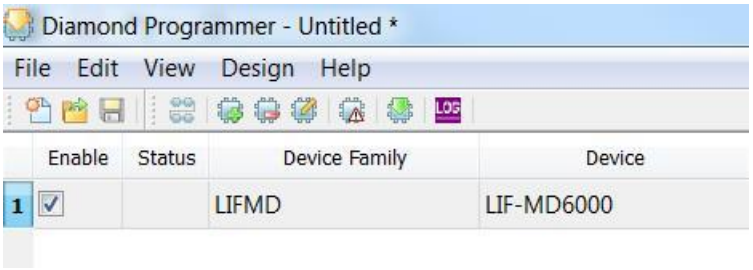

Figure 4.5. Device Selection

4. Click the CrossLink row and select **Edit > Device Properties**.
5. In the Device Properties dialog box, select **SSPI SRAM Programming** in Access mode and **Erase Only** in Operation as shown in Figure 4.6.

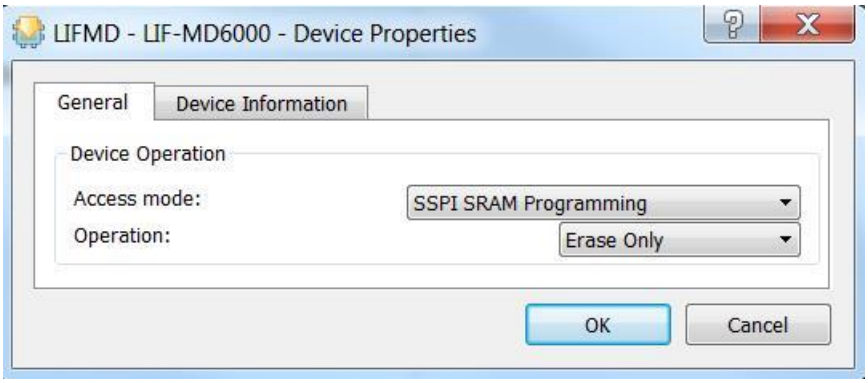

Figure 4.6. Device Operation

6. Click **OK** to close the Device Properties dialog box.
7. In the Diamond Programmer main interface, click the **Program** button to start the erase operation.

Note: If you power OFF/ON the board, the SPI Flash reprograms the CrossLink device. In this case, you have to repeat steps 1 to 7.

4.2.2. Programming the CrossLink VIP Input Bridge Board

To program the CrossLink VIP Input Bridge Board:

1. Ensure that the CrossLink device SRAM is erased by performing the steps in Erasing the CrossLink SRAM Prior to Reprogramming.
2. In the Diamond Programmer main interface, click the CrossLink row and select **Edit > Device Properties** to open the Device Properties dialog box as shown in Figure 4.7.

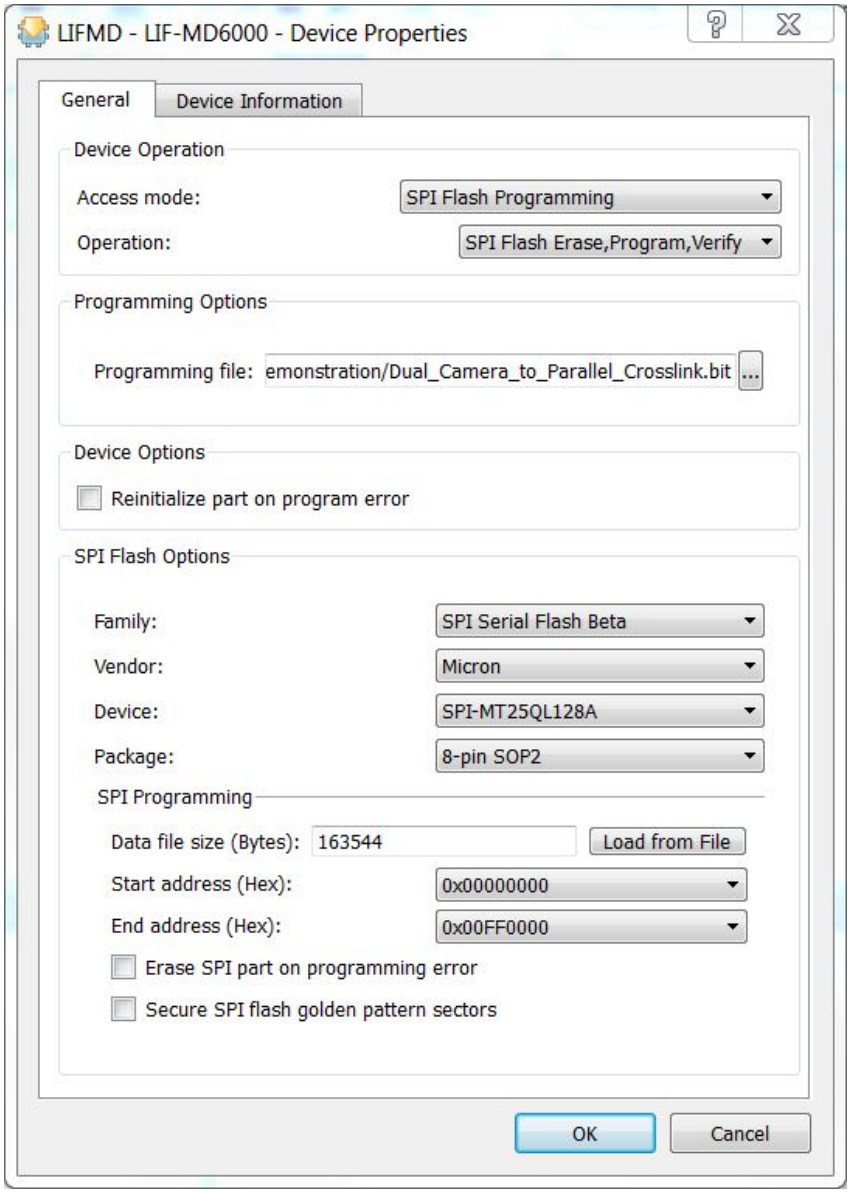

Figure 4.7. Selecting General Options

3. Apply the settings below.
   a. Under Device Operation, select the options below:
      - Access Mode: **SPI Flash Programming**
      - Operation: **SPI Flash Erase, Program, Verify**
   b. Under Programming Options, select the bitstream file **~/Demonstration/Dual_Camera_to_parallel_Crosslink.bit** in Programming file.
   c. Under **SPI Flash Options**, select the options below based on the board revision.

      Revision B
      - Family: **SPI Serial Flash**
      - Vendor: **Micron**
      - Device: **SPI-M25PX16**
      - Package: **8-pin S08W**

      Revision C

      Revision C board may be populated with one of the following devices:

      Option 1
      - Family: **SPI Serial Flash** (**SPI Serial Flash Beta** for Diamond 3.10 SP1 or earlier)
      - Vendor: **Micron**
      - Device: **SPI-MT25QL128A**
      - Package: **8-pin SOP2**

      Option 2
      - Family: **SPI Serial Flash** (**SPI Serial Flash Beta** for Diamond 3.10 SP1 or earlier)
      - Vendor: **Macronix**
      - Device: **MX25L12835F**
      - Package: **8-Land WSON**
   d. Click **Load from File** to update the Data file size (Bytes) value.
   e. Ensure that the following addresses are correct:
      - Start Address (Hex): **0x00000000**
      - End Address (Hex): **0x00020000**
4. Click **OK**.
5. In the Diamond Programmer main interface, click the **Program** button to start the programming operation. Successful programming is displayed in the Diamond Programmer **Output** console as shown in Figure 4.8.

Figure 4.8. Output Console

4.3. Programming the MicroSD Card Firmware

To write the image to the MicroSD Card:

1. Download and install the Win32diskimager Image Writer software from the following link: https://sourceforge.net/projects/win32diskimager/.
2. Use Win32diskimager to write the appropriate Flash image file to the SD memory card. Depending on your PC, you may need a separate adapter (not described in this document) to physically connect to the card. See the Programming the Demo section to determine the file for the specific demo.
3. Connect the MicroSD Card as shown in Figure 4.9.

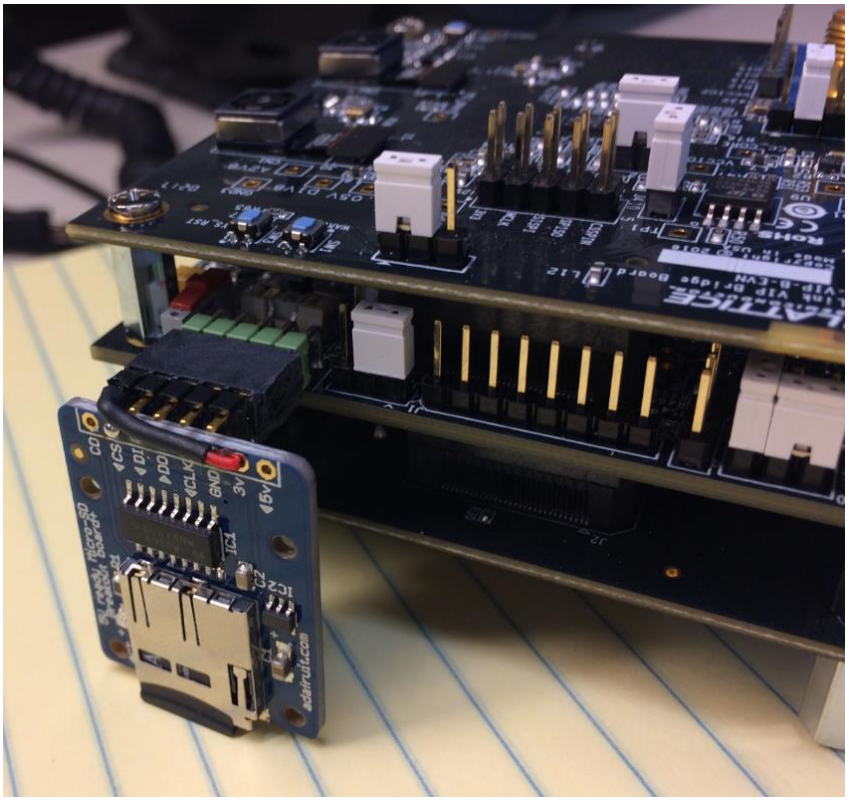

Figure 4.9. Connecting the MicroSD Card

4. In Win32 Disk Imager, select the image file **~/Demonstration/vip_objectcount.bin** as shown in Figure 4.10.
5. Select the card reader in **Device**.
6. Click **Write**.

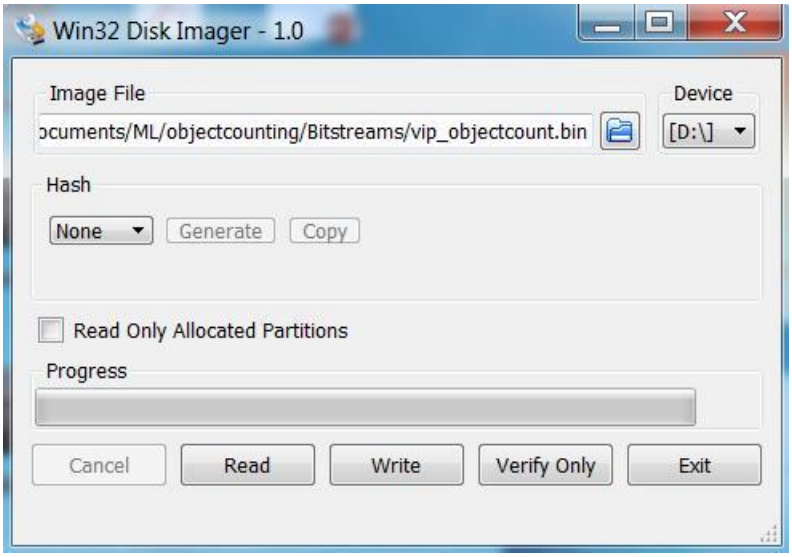

Figure 4.10. Win32 Disk Imager

5. Running the Demo

To run the demo:

1. Write the **~/Demonstration/vip_objectcount.bin** file to the MicroSD card.
2. Insert the configured MicroSD Card into the MicroSD Card Adapter, and connect it to the Embedded Vision Development Kit.
3. Cycle the power on the Embedded Vision Development Kit to allow ECP5 and CrossLink to be reconfigured from Flash.
4. Connect the Embedded Vision Development Kit to the HDMI monitor. The camera image should be displayed on monitor.

Note: Since demo firmware/information is written to non-volatile Flash memory, it runs at power-up.

This demo provides an example of object (fruit) counting application. Red and green dots identify the detected apples and oranges respectively, while overlay text indicates the number of apples and oranges detected as shown in Figure 5.1.

Figure 5.1. Fruit Counting Demo Results

Technical Support

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

Revision History

Revision 1.2, August 2019

Section	Change Summary
All	• Added Disclaimers page. • Updated the last page of this document.
Demo Setup	Added a sub-bullet point for Micro SD Card Adapter in Hardware Requirements section.

Revision 1.1, September 2018

Section	Change Summary
Demo Setup	Adjusted callouts in Figure 3.1. Back View of ECP5 VIP Input Bridge Board and Figure 3.2. Top View of CrossLink VIP Input Bridge Board.
Programming the Demo	• Updated the values for Revision C in Programming the ECP5 VIP Processor Board section. • Updated the values for Revision C in Programming the CrossLink VIP Input Bridge Board section.
Revision History	Updated revision history table to new template.

Revision 1.0, June 2018

Section	Change Summary
All	Initial release.

www.latticesemi.com