Human Presence Detection Using Compact CNN Accelerator IP

Reference Design

FPGA-RD-02045 Version 1.1

October 2018

Contents

Figures

1. Introduction

The Human Presence Detection Using Compact CNN Accelerator IP Reference Design document describes how to implement the Human Presence Detection design using the iCE40™ UltraPlus FPGA. This design utilizes the Lattice Radiant CNN Accelerator IP core, which is optimized for Convolutional Neural Network (CNN) implementations. The document discusses the design in detail on the top module blocks that show how Human Presence Detection is implemented.

2. Related Documentation

In addition to using this guide to help you get started developing CNN solutions on your device, you can refer to other applicable documents that may contain more detailed information that is beyond the scope of this guide.

The following documents can be obtained on the Lattice website:

- Lattice SensAI Neural Network Compiler Software (FPGA-UG-02052) - This document explains how to create the firmware file, which contains the command sequence as well as weights that go into the CNN Accelerator IP Core.
- Compact CNN Accelerator IP Core User Guide (FPGA-IPUG-02038) - This document provides additional details about the CNN IP Core contained in this design.
- Lattice Radiant Software User Guide - This document describes the main features, usage, and key concepts of the Lattice Radiant software, which is used in creating this reference design.

3. Software Requirements

The following software are required to design and use the Human Presence Detection Reference Design:

- Radiant Software Version 1.0 or higher - This is used to build the reference design.
- Radiant Programmer tool - This is used to program the bitstream to the External SPI Flash on the board.
- Compact CNN Accelerator IP Core - This is used to create the CNN Accelerator Engine.
- Lattice SensAI Neural Network Compiler Software – This is used to generate the firmware file from the TensorFlow/Caffe outputs.

4. Hardware Requirements

The reference design is targeted to support the Himax HM01B0 UPDuino Shield board, but it can be modified for use in other hardware as needed. This hardware requires a mini-USB cable in order to program.

5. Directory Structure

Figure 5.1 shows the directory structure of the **Human Presence Detection Using Compact CNN Accelerator IP Detect Low Power – Project Files**. The figure also details the files contained in each folder.

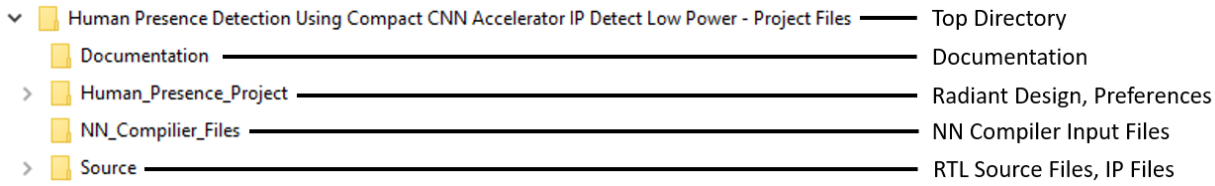

Figure 5.1. Human Presence Detection Directory Structure

6. Reference Design Overview

6.1. Block Diagram

This section describes the Human Presence Detection block diagram and each top block module, as shown in Figure 6.1.

Note: Different PAR iterations can give different results. Try more than one PAR iteration to acquire the best results.

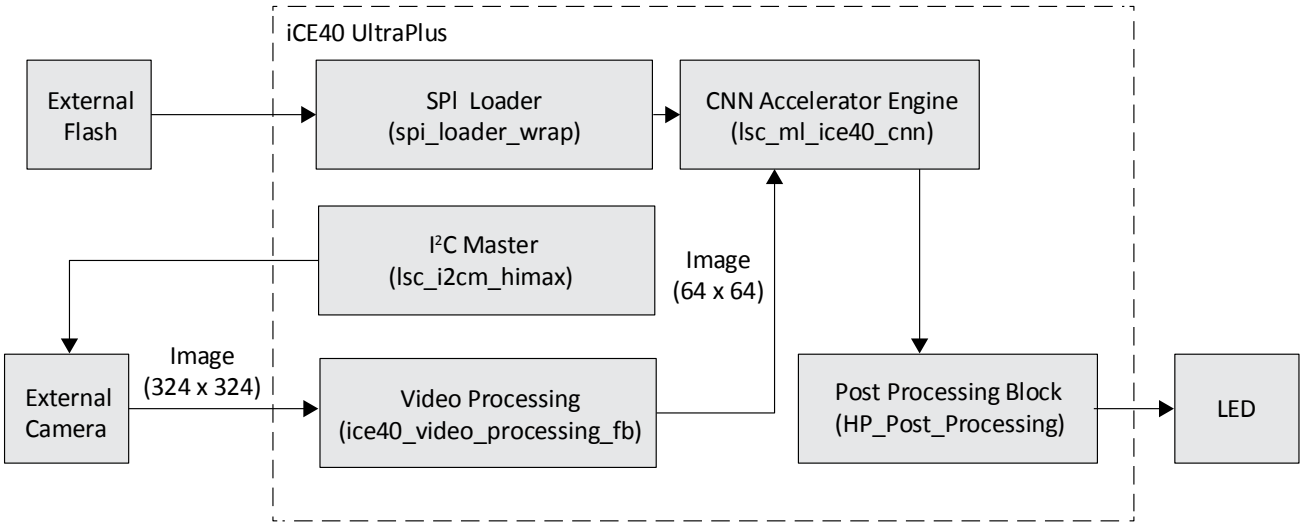

Figure 6.1. Human Presence Detection Top-Level Block Diagram

The reference design shows an implementation of a machine learning design using the Compact CNN Soft IP Engine. There are two main inputs that go into the CNN Accelerator Engine, which then outputs six values that are evaluated to determine whether a human is detected in a specific area.

6.2. Top-Level Blocks

This section discusses in detail the top-level modules that make up the Human Presence Detection Reference Design. Each top-level block includes the Verilog (.v) or the IP Catalog (.ipx) files. These files are in the source directory of the .zip file.

6.2.1. Top-Level Module – Human_Presence_Top.v

This is the top-level module that contains all the blocks and connections found in Figure 6.1.

6.2.2. CNN Accelerator Engine – lsc_ml_ice40_cnn.ipx

This block contains the CNN Accelerator Engine. In this design, the engine is configured to Memory Type: DUAL SPRAM, ML_TYPE: CNN, SCRATCH SIZE: 4K. Refer to the Compact CNN Accelerator IP Core User Guide (FPGA-IPUG-02038) for details. This block takes the firmware file via SPI_Loader_wrap.v and the processed image to output the desired result.

6.2.3. SPI Loader – spi_loader_wrap.v and spi_loader_spram.v

This module reads the external flash for the firmware file. The firmware file is then outputted into the CNN Accelerator IP Core. The SPI loader is configured to read the firmware file at address 20'h20000. The Lattice Neural Network compiler tool generates the firmware file.

Firmware files are located in the **NN_Compiler_Files** directory for you to generate your own firmware file.

6.2.4. I2S Master – i2cm_himax.v

This module communicates with the Himax Camera and configures it upon boot up to make sure that it outputs the correct image. The Himax Camera is configured to output a 3 x 324 x 324 image.

6.2.5. Video Preprocessing – ice40_himax_video_process_64.v

This module takes the Himax Camera image output and downscales it to 3 x 64 x 64. This downscaled image is then sent to the CNN Accelerator IP core when requested. Scaling the image down to 3 x 64 x 64 reduces the amount of data fed into the CNN Accelerator without significant loss in accuracy.

6.2.6. Post Processing Block/LED – HP_Post_Processing.v

This module captures the output of the CNN Accelerator Engine and post processes the data. The output is in six clock cycles, which represent the confidence value of a human appearing in the Upper Left, Upper Right, Lower Left, Lower Right, Center, and Full. The results are outputted to LED lights on the board. If any of these values go over a set threshold, a corresponding LED light turns on.

7. Generating the Firmware File

To generate the Human Presence Firmware file:

1. Using the files located in the **NN_Compiler_Files** directory, create a new project in SensAI and apply the following settings as shown in Figure 7.1.
   - **Framework — TensorFlow**
   - Device — UltraPlus
   - **Class — CNN**
   - **Network File — HumanPresence.pb**
   - **Image/Video/Audio Data — man2.jpg**
2. Click **Next**.

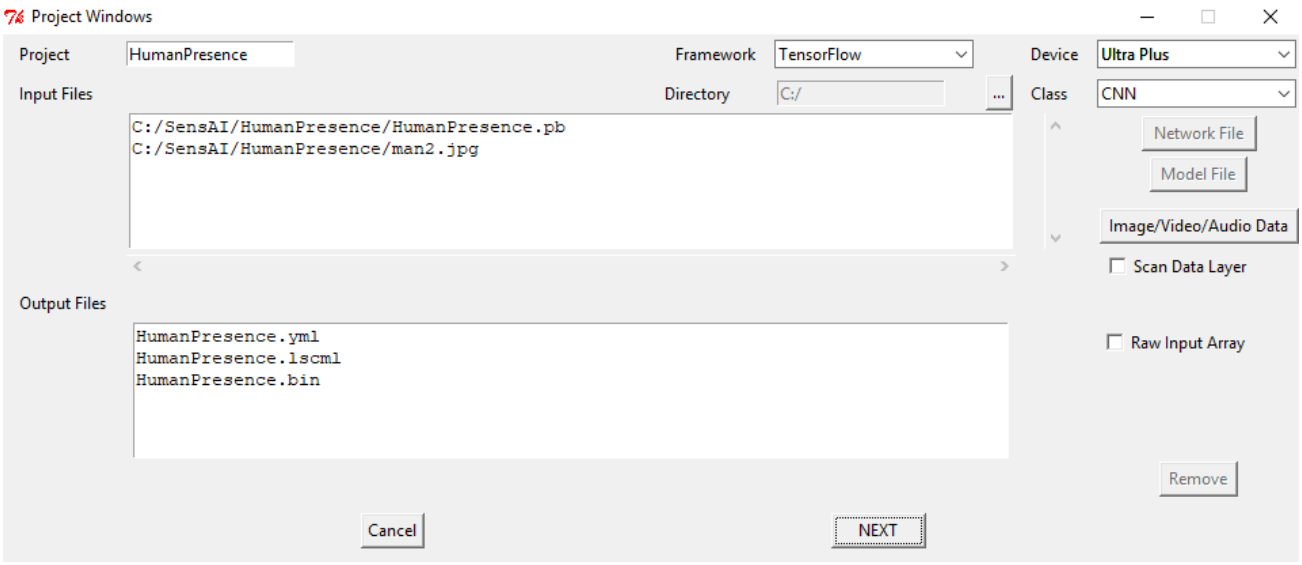

Figure 7.1. SensAI Project Settings Part 1

3. In the second section, apply the Neural Network engine settings as shown in Figure 7.2:
   - **On-Chip Memory Block Size — 65536**
   - **Mean Value — 0**
   - **Scale Value — 1.0**
4. Click **OK**.

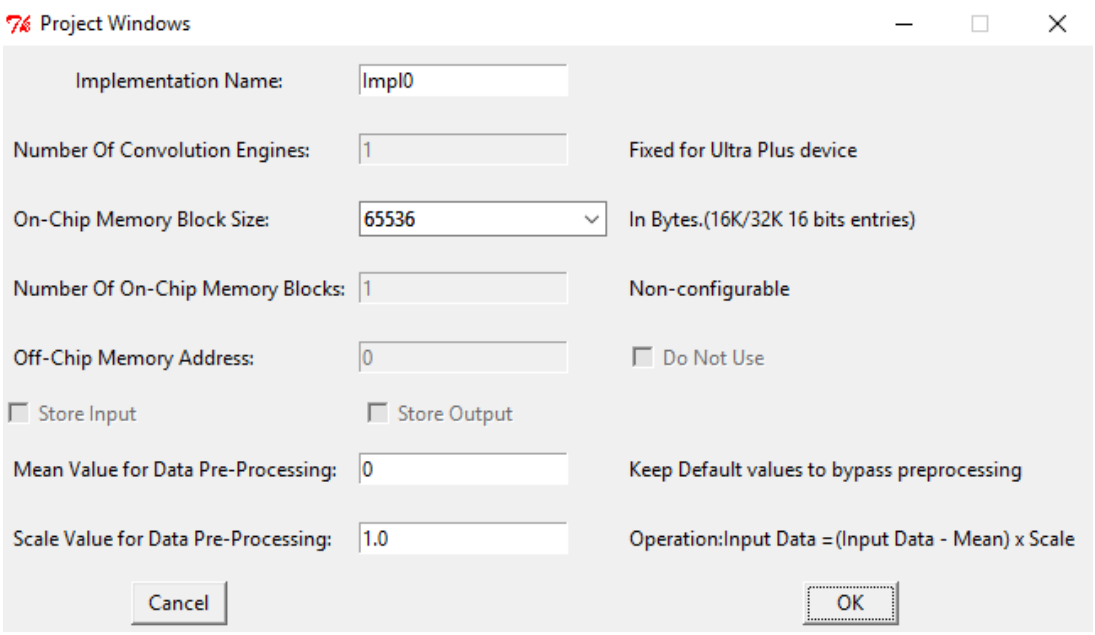

Figure 7.2. SensAI Project Settings Part 2

5. Note that this reference design requires fractional bit changes on every layer to fine-tune the accuracy of the design. After analyzing the network, edit the fractional bit for each layer. Figure 7.3 shows the change of the fractional bit.

Impl0

Blobs	Signed Data Format(Analyzed)	Stored Data Format(User Edit)	Required Memory Bytes	MAE_Simulation
data	11.4	11.4	16384	
fire1/conv3x3/convolution	11.4	12.3	32768	
fire1/bn/batchnorm/Rsqrt	11.4	11.4	32768	
fire1/bn/batchnorm/Rsqrt/scale	11.4	11.4	32768	
fire1/Relu	11.4	11.4	32768	
fire1/pool/MaxPool	11.4	11.4	16384	
fire2/conv3x3/convolution	10.5	10.5	16384	
fire2/bn/batchnorm/Rsqrt	10.5	10.5	16384	
fire2/bn/batchnorm/Rsqrt/scale	10.5	10.5	16384	
fire2/Relu	10.5	10.5	16384	
fire3/conv3x3/convolution	5.10	5.10	16384	
fire3/bn/batchnorm/Rsqrt	5.10	5.10	16384	
fire3/bn/batchnorm/Rsqrt/scale	5.10	5.10	16384	
fire3/Relu	5.10	5.10	16384	
fire3/pool/MaxPool	5.10	5.10	16384	
fire4/conv3x3/convolution	5.10	5.10	16384	
fire4/bn/batchnorm/Rsqrt	5.10	5.10	16384	
fire4/bn/batchnorm/Rsqrt/scale	5.10	5.10	16384	
fire4/Relu	5.10	5.10	16384	
fire5/conv3x3/convolution	4.11	5.10	16384	
fire5/bn/batchnorm/Rsqrt	4.11	5.10	16384	
fire5/bn/batchnorm/Rsqrt/scale	4.11	4.11	16384	
fire5/Relu	4.11	4.11	16384	
fire5/pool/MaxPool	4.11	4.11	16384	
fire6/conv3x3/convolution	5.10	5.10	16384	
fire6/bn/batchnorm/Rsqrt	5.10	5.10	16384	
fire6/bn/batchnorm/Rsqrt/scale	5.10	5.10	16384	
fire6/Relu	5.10	5.10	16384	
fire6/pool/MaxPool	5.10	5.10	16384	
conv12/convolution	4.11	5.10	16384	

Figure 7.3. Fractional Bit Change

6. After changing the fractional bit, save the project file and reanalyze the design.

7. In addition to modifying the fractional bits, modify the **yml** file. Open the **yml** file located in your project directory. On top of the **yml** file, add **output_data_length: 96**. At the last convolution, add **weight_slice: 12**. Figure 7.4 and 7.5 shows the change.

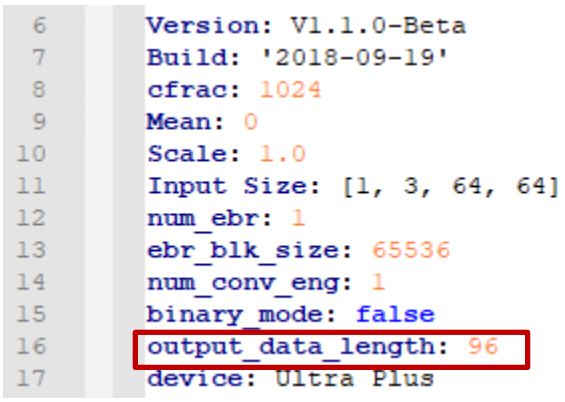

Figure 7.4. YML Modification Part 1

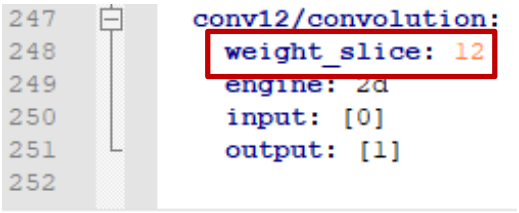

Figure 7.5. YML Modification Part 2

8. Click **Compile** to generate the Firmware file.

   Note: Do not click **Analyze**. This overwrites the **yml** file.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.1, October 2018

Section	Change Summary
Generating the Firmware File	Updated Figure 7.1. SensAI Project Settings Part 1.

Revision 1.0, September 2018

Section	Change Summary
All	Initial release.

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