A

Project Report

On

IMAGE PROCESSING USING VERILOG

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, KADAPA

in partial fulfilment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

S.GOWSIYA TAJ	**R170640**
M. SAI PRAKASH REDDY	**R170797**

Under the Guidance of

Mr. ABDUL MUNAF

Assistant Professor

Department Of Electronics and Communication Engineering

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

RGUKT, RK VALLEY
(RGUKT KADAPA is approved by UGC, AICTE, established in 2008, provide Education opportunities for rural people)
Vempalli, Kadapa-516330

2019-2023

CERTIFICATE

This is to certify that the project report entitled **"IMAGE PROCESSING USING VERILOG"** a bonafide record of the project work done and submitted by

S.GOWSIYA TAJ	**R170640**
M.SAI PRAKASH REDDY	**R170797**

for the partial fulfilment of the requirements for the award of B.Tech Degree in **ELECTRONICS AND COMMUNICATION ENGINEERING,** RGUKT , Kadapa.

PROJECT GUIDE
Mr. ABDUL MUNAF
Assistant Professor

Head of the Department
B. Madhan Mohan
Assistant Professor,
Head of Department of ECE

External Viva-Voice Exam Held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We here by declare that the project report entitled **“IMAGE PROCESSING USING VERILOG”** submitted to the Department of **ELECTRONICS AND COMMUNICATION ENGINNERING** in partial fulfillment of requirements for the award of the degree of **BACHELOR OF TECHNOLOGY.** This project is the result of our own effort and that it has not been submitted to any other university or institution for the award of any degree or diploma other than specied above.

S.GOWSIYA TAJ R170640
M.SAI PRAKASSH REDDY R170797

ACKNOWLEDGEMENT

We are thankful to our guide **Mr. ABDUL MUNAF** for his valuable guidance and encouragement . His helping attitude and suggestions have helped us in the successful completion of the project.

We would like to express our gratefulness and sincere thanks to **Prof. B.MADHAN MOHAN,** Head of the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING,** for the kind help and engouragement during the course of our study and in the successful completion of the project work.

We have great pleasure in expressing our hearty thanks to beloves Director Prof. **K.SANDHYA RANI** for spending her valuable time with us to complete this project.

Successful completion of any project cannot be done without proper support and encouragement.We sincerely thanks to the **Managament** for providing all the necessary facilities during the course of study.

We would like to thank to our parents and friends, who have the greatest contribution in all our achievements, for the great care and blessings in making us successful in all our endevours.

S.GOWSIYA TAJ **R170640**
M.SAI PRAKASH REDDY **R170797**

ABSTRACT

The field of digital image processing has experienced continuous and significant expansion in recent years.the useful of this technology is apparent in many different disciplines covering medcine through remote sensing .The advances and wide availability of image processing hardware has further enchanced the usefulness of image processing.FPGAs are widely used for video and image processing applications such as broadcast infrastructure,medical imaging ,HD videoconferencing ,video Surveilance and military imaging.

TABLE OF CONTENT **PAGE NO**

LIST OF IMAGES **PAGE NO**

CHAPTER 1

INTRODUCTION:

Image processing is a technique to enhance raw images received from cameras/sensors placed on satelities ,spaces probes and aircrafts or pictures taken in normal day to day life for life for various applications.Various techniques have been developed in image processing during the last four to five decades.Most of the techniques are developed for enhancing images obtained from unmanned space crafts ,space probes and military reconnaissance flights.Image proccesing systems are becoming popular due to easy availability of powerful personnel computers,large size memory devices ,graphics software etc.

There are two methods of image proccessing

1.Analog image processing

2.Digital image processing

CHAPTER 2 VERILOG

2.1.WHAT IS VERILOG?

Verilog is a hardware description language. It is a language used for describing a digital system like a network switch or a microprocessor or a memory or a flip-flop.It means,by using a HDL we can describe any digital hardware at any level . Designs ,which are described in HDL are independent of technology,very easy for designing and debugging ,and are normally more useful than schematics,particularly for large circuits.So,by all these reasons ,we took verilog HDL for designing purpose in software.

Verilog, standardized as **IEEE 1364**, is a hardware description language (HDL) used to model electronic systems. It is most commonly used in the design and verification of digital circuits.

It is also used in the verification of analog circuits and mixed-signal circuits, as well as in the design of genetic circuits.

2.2 What is FPGA?

FPGA in VLSI stands for

'Field Programmable Gate Array', which is an integrated circuit that may be programmed to execute a tailored function for a particular purpose.FPGAs have become highly popular in the VLSI area.The code for FPGA programming is written in languages like VHDL and Verilog.FPGAs are integrated circuits in the category of programmable logic devices or PLDs. FPGAs are the highest performance, most flexible and also the most expensive of the PLD types.

A drawback of FPGAs compared to other PLDs is that they do not remember their design when the power is removed. An FPGA therefore needs a separate configuration memory chip that holds the FPGA design. When the power is given back to the FPGA a fixed part of the FPGA reads the configuration from the configuration memory chip.After the FPGA is configured it will be able to do the function it was given by the design.An FPGA is different from a microprocessor or microcontroller. An FPGA in its basic form is not able to run software. Only when the FPGA is given a configuration that contains a processor-architecture it has the ability to run software.

CHAPTER 3

IMAGE PROCESSING

Image processing is the process of transforming an image into a digital from and performing certain operations to get some useful information from it.The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.

There are five main types of image processing :

1.Visualization -Find objects that are not visible in the image.

2.Recognisation-Distinguish or detect objects in the image.

3.Sharpening and restoring-create an enhanced image from the original image.

4.Pattern Recognisation-Measure the various pattern around the objects in the image.

5.Retrieval-Browser and search images from a large database of digital images that are similar to the original image.

In the FPGA verilog project some simple processing operations are implemented in verilog such as inversion ,brightness control and threshold operations.the image writing is also extremely useful for testing as well when you want to set output image in BMP format.

CHAPTER 4

4.1. PROCESS:

This FPGA project is aimed to show in details how to process an image using Verilog from reading an input bitmap image (.bmp) in Verilog, processing and writing the processed result to an output bitmap image in Verilog.

In this FPGA Verilog project, some simple processing operations are implemented in Verilog such as inversion, brightness control and threshold operations.

The image processing operation is selected by a "parameter.v" file and then, the processed image data are written to a bitmap image output.bmp for verification purposes. The image reading Verilog code operates as a Verilog model of an image sensor/ camera, which can be really helpful for functional verifications in real-time FPGA image processing projects.

The image writing part is also extremely useful for testing as well when you want to see the output image in BMP format. In this project, I added some simple image processing code into the reading part to make an example of image processing, but you can easily remove it to get raw image data.

First of all, Verilog cannot read images directly. To read the .bmp image on in Verilog, the image is required to be converted from the bitmap format to the hexadecimal format. Below is a Matlab example code to convert a bitmap image to a .hex file. The input image size is 768x512 and the image .hex file includes R, G, B data of the bitmap image.

To read the image hexadecimal data file, Verilog uses this command: $readmemh or $readmemb if the image data is in a binary text file. After reading the image .hex file, the RGB image data are saved into memory and read out for processing.

The header data for the bitmap image is very important. If there is no header data, the written image could not be correctly displayed. In Verilog HDL, $fwrite command is used to write data to file.

Finally, we have everything to run a simulation to verify the image processing code. Let's use the following image as the input bitmap file.Run the simulation for 6ms, close the simulation and open the output image for checking the result.

Since the reading code is to model an image sensor/camera for simulation purposes, it is recommended not to synthesize the code. If you really want to synthesize the processing code and run this directly on FPGA, you need to replace these image arrays (total_memory, temp_BMP, org_R, org_B, org_G) in the code by block memory (RAMs) and design address generators to read image data from the block memory.

4.2. STEPS:

1. Run the simulation about 6ms and close the simulation, then you will be able to see the output image.
2. The reading part operates as a Verilog model of an image sensor/camera (output RGB data, HSYNC, VSYNC, HCLK). The Verilog image reading code is extremely useful for functional verification in real-time FPGA image/video projects.
3. In this project, I added the image processing part to make an example of image enhancement. You can easily remove the processing part to get only raw image data in the case that you want to use the image sensor model only for verifying your image processing design.
4. The image saving into three separate RGB mems: Because Matlab code writes the image hexadecimal file from the last row to the first row, the RGB saving codes (org_R, org_B, org_G) do the same in reading the temp_BMP memory to save RGB data correctly. You can change it accordingly if you want to do it differently.
5. You might find the following explanation for the BMP header useful if you want to change the image size:

BMP_header[2]=54;
BMP_header[3]=0;
BMP_header[4]=18;
BMP_header[5]=0;
Image width = 768 => In hexadecimal: 0x0300.
The 4 bytes of the image width are 0, 3, 0, 0.Thats how you get the following values:

BMP_header[18]=0;
BMP_header[19]=3;
BMP_header[20]=0;
BMP_header[21]=0;
Image height = 512 => In hexadecimal: 0x0200. The 4 bytes of the image width are 0, 2, 0, 0.That's how we get the following values:
BMP_header[22]=0;
BMP_header[23]=2;
BMP_header[24]=0;
BMP_header[25]=0;
6. You should not synthesize this code because it is not designed for running on FPGA, *but rather for functional verification purposes*. If you really want to synthesize this code (read and process) and load the image into FPGA for processing directly on FPGA, replace all the temp. variables (org_R, org_B, org_G, tmp_BMP = total_memory) by block RAMs and generate addresses to read the image data (remove the always @(start) and all the "for loops" - these are for simulation purposes). There are two ways: 1. write a RAM code and initialize the image data into the memory using $readmemh; 2. generate a block memory using either Xilinx Core Generator or Altera MegaFunction and load the image data into the initial values of memory (.coe file for Xilinx Core Gen. and .mif for Altera MegaFunction), then read the image data from memory and process it (FSM Design).

7. In this project, two even and old pixels are read at the same time for speeding up the processing, but you can change the number of pixels being read depending on your design.
8. The writing Verilog code is also very helpful for testing purposes as you can see the output in BMP format.
9. If you want to do real-time image processing, you can check this for the camera interface code: **Basys 3 FPGA OV7670 Camera**

CHAPTER 5

5.1. OUTPUT IMAGES:

INPUT BITMAP IMAGE

.

OUTPUT BITMAP IMAGE

AFTER SUBTACTING BRIGHTNESS

OUTPUT BITMAP IMAGE AFTER INVERTING

OUTPUT BITMAP IMAGE AFTER THRESHOLDING

CHAPTER 6

6.1. APPLICATIONS

● Medical image Retrieval

●Beta – cells detection, Detection of cancer tissues

●Traffic Sensing Technologies

●Vehicle classification

●Video surveillance and tracking

●Image compression

●Image Reconstruction

●Face detectionImage processing in Astronomy

●Robotic vision

●Satellite imagery

●Face Detection

6.2. ADVANTAGES

●The digital image can be made available in any desired format (improved image, X-Ray, photo negative, etc)

●It helps to improve images for human interpretation.

●Information can be processed and extracted from images for machine interpretation.

●The pixels in the image can be manipulated to any desired density contrast.

●Images can be stored and retrieved easily.

●It allows for easy electronics transmission of images to third party providers.

6.3. DISADVANTAGES

- The initial cost can be high depending on the system used.
- If computer is crashes then pics that have not been printed and filed into book albums
- that are lost.
- Digital cameras which are used for digital image processing have some disadvantages like:
- Memory card problems
- Higher cost
- Battery consumption.

CHAPTER 7

CONCULSION

- When developing innovative communication systems, the quicker we have implementation of video and image processing.
- With the developing architecture, FPGAs are great fits for video and image processing applications, such as broadcast infrastructure, medical imaging.
- HD videoconferencing, video surveillance, and military imaging. Video and image processing solutions for FPGAs include optimized development tools and kits, reference designs, video compression IP, and interface and system IP, as well as video and image processing IP suite.
- These solutions can improve cost, performance, and productivity for many video and imaging applications in future.

CHAPTER 8

REFERENCE

[1] John C. Russ - “Image Processing Handbook (sixth edition)”, CRC Press, pp. 270-331, 2011

[2] Raman Maini, H. Aggarwal - “A Comprehensive Review of Image enhancement Techniques”, Journal of Computing, vol. 2, issue 3, ISSN 2151-9617, pp. 269-300,2010.

[3] Wilhelm Burger, Mark J. Burge - “Principles of Digital Image Processing – Fundamental Techniques”, Undergraduate Topics in Computer Science, DOI 10.1007/978-1-84800-191-6_4, Springer- Verlag London Limited, 2009.

[4] A. Zuloaga, J.L. Martin, U. Bidarte, J.A. Ezquerra - “VHDL test bench for digital image processing systems using a new image format”, ECSI, 2007 (http://mx.reocities.com/CapeCanaveral/8482/).